Exact Geodesic Metric in 2-Manifold Triangle Meshes Using Edge-based Data Structures

Yong-Jin Liu

Tsinghua National Lab for Information Science and Technology,
Department of Computer Science and Technology,
Tsinghua University,
Beijing, P. R. China
liuyongjin@tsinghua.edu.cn

Abstract

A natural metric in 2-manifold surfaces is to use geodesic distance. If a 2-manifold surface is represented by a triangle mesh T, the geodesic metric on T can be computed exactly using computational geometry methods. Previous work for establishing the geodesic metric on T only supports using half-edge data structures; i.e., each edge e in T is split into two halves (he_1, he_2) and each half-edge corresponds to one of two faces incident to e. In this paper, we prove that the exact-geodesic structures on two half-edges of e can be merged into one structure associated with e. Four merits are achieved based on the properties which are studied in this paper: (1) Existing CAD systems that use edge-based data structures can directly add the geodesic distance function without changing the kernel to a half-edge data structure; (2) To find the geodesic path from inquiry points to the source, the MMP algorithm can be run in an on-the-fly fashion such that the inquiry points are covered by correct wedges; (3) The MMP algorithm is speeded up by pruning unnecessary wedges during the wedge propagation process; (4) The storage of the MMP algorithm is reduced since less wedges need to be stored in an edge-based data structure. Experimental results show that when compared to the classic half-edge data structure, the edge-based implementation of the MMP algorithm reduces 44% running time and 29% storage on average.

Keywords: Exact geodesic, triangle meshes, 2-manifold, edge-based data structure

1 Introduction

In many geometric problems in industry, solution spaces (such as object spaces or configuration spaces) are usually given in the form of polygonal meshes in $\mathbb{R}^3$ which represent 2-manifold surfaces. On curved 2-manifold surfaces, a natural metric is to use geodesic distance. Computing geodesic metric has found a widely range of applications in natural science and engineering. In this paper, we study the computation of exact geodesic metric on 2-manifold triangle meshes.

A geodesic path between two points p and q on a 2-manifold M is a local shortest path on M which connects p and q. A geodesic metric on M is a real function $d : M \times M \to \mathbb{R}$ such that $\forall p, q \in M$, $d(p, q)$ is the length of the shortest path between p and q. Let T be a 2-manifold triangle mesh. If a geodesic path on T does not go through any vertex in T except for its two endpoints, we can always unfold the triangles, which are passed through by the geodesic path, one by one along their shared edges into a 2D plane and then the geodesic path becomes a straight line in the plane. Fast algorithms have been proposed for both approximation and exact geodesic computation on T. We briefly summarize some representative works below. The reader is referred to [1] for a detailed survey.

A path is called a $1 + \epsilon$ approximation of a shortest path on T if its length is at most $1 + \epsilon$ times the length of the shortest path. Let n be the number of triangles in T. Hershberger and Suri [2] presented a simple linear algorithm to compute an approximate shortest path ($\epsilon = 1$) on convex polytopes. For $\epsilon < 1$, Agarwal et al. [3] presented an algorithm of $O(n \log \frac{1}{\epsilon} + \frac{1}{\epsilon^3})$ time complexity, which had been further improved to $O(n + \frac{\log n}{\epsilon^{1.5}} + \frac{1}{\epsilon^3})$ [4] and $O(\frac{n}{\sqrt{\epsilon}} + \frac{1}{\epsilon^4})$ [5], respectively. These algorithms [2, 3, 4, 5] are all only applicable for convex polytopes. For non-convex polytopes, Har-Peled [6] presented an $1+\epsilon$ approximation algorithm of $O(n^2 \log n + \frac{n}{\epsilon} \log \frac{1}{\epsilon} \log \frac{n}{\epsilon})$ time complexity. Interpreting a triangulated surface as a linear approximation of a smooth 2-manifold, geodesic paths can also be computed approximately using numerical methods [7] which use the fast marching method to solve the Eikonal equation on T.

Exact geodesic metric can be computed on T using computational geometry methods. Sharir and A. Schorr [8] proposed an $O(n^3 \log n)$ algorithm to compute shortest paths on convex polyhedrons. A breakthrough is achieved in [9], called *MMP algorithm* below, in which the shortest path between a source point and any destination point on T is determined in $O(n^2 \log n)$ time. The running time of MMP algorithm was further improved to $O(n^2)$ [10] and $O(n \log^2 n)$ [11], respectively. Surazhsky et al. [12] and Qin and Wang [13] presented novel implementations of MMP algorithm [9] and CH algorithm [10], re-

spectively, and reported that their implementations runs fast in practice. The degenerate cases in the implementation [12] are handled in [14].

All these work [9, 10, 12, 13] for computing exact geodesic metric on T relies on a half-edge (also called directed edge in literature) representation [15, 16] of T; i.e., every edge in T is represented by two half-edges, each of which corresponds to one of two faces incident to that edge. This property, however, prevents the algorithms to apply with other widely used data structures in engineering such as doubly-connected edge list [17], winged-edge data structure [18] and its variants for sold modeling [19], in which each edge in T is recorded exactly once and we call these data structures [17, 18, 19] as edge-based data structures, compared to the half-edge data structures [15, 16]. The comparisons of half-edge and edge-based data structures are studied in [20].

In this paper, we make the following contributions:

- We prove that the exact geodesic metric structure in MMP algorithm [9] based on half edges can be merged for each edge in T. This makes the algorithm applicable for edge-based data structures. Furthermore, we show that the MMP algorithm can be performed in an on-the-fly fashion; i.e., when we start at a mesh point p to propagate a structure for find the geodesic path to another mesh point q, the MMP algorithm need not to be performed over the entire mesh surface, but can be terminated when a correct structure covers q.

- The core of the MMP algorithm is to propagate a set of wedges over all edges in T. By merging the wedges on two incident half-edges into one set of wedges on an edge, unnecessary wedges can be efficiently pruned during the wedge propagation process and the number of total wedges in the MMP algorithm is reduced. We show that when compared to the half-edge data structures, edge-based implementation of the MMP algorithm reduces 29% storage and 44% running time on average.

2 A short summary of the MMP algorithm [9, 12]

The key idea of the MMP algorithm [9] is that between two inquiry points p and $q \in T$, there exists a set of triangles $T_{pq} = (t_1, t_2, \cdots, t_m) \in T$, $p \in t_1$, $q \in t_m$, which satisfies:

1. Each two sequential triangles t_i and t_{i+1} are adjacent and share a common edge;

2. If all these triangles are unfolded into a plane $\mathbb{R}^2$ along the shared edges one by one, in the unfolded image of T_{pq} in $\mathbb{R}^2$, the shortest path between p, q on T becomes a line $\overline{pq}$ or or a polyline $\overline{pv_1v_2\cdots q}$, where $v_1, v_2, \cdots$ are saddle vertices in T_{pq}.

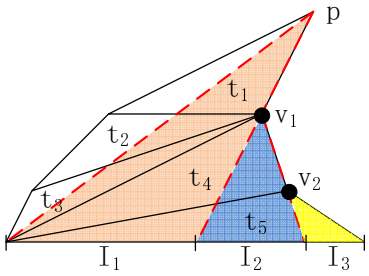

Figure 1: Shortest paths inside triangles $(t_1, t_2, t_3, t_4, t_5)$ unfolded into a plane. For the source point p, if the inquiry point $q \in I_1$, then the shortest path is a line $\overline{pq}$; if $q \in I_2$ (or $q \in I_3$), the shortest path is a polyline $\overline{pv_1q}$ (or $\overline{pv_1v_2q}$).

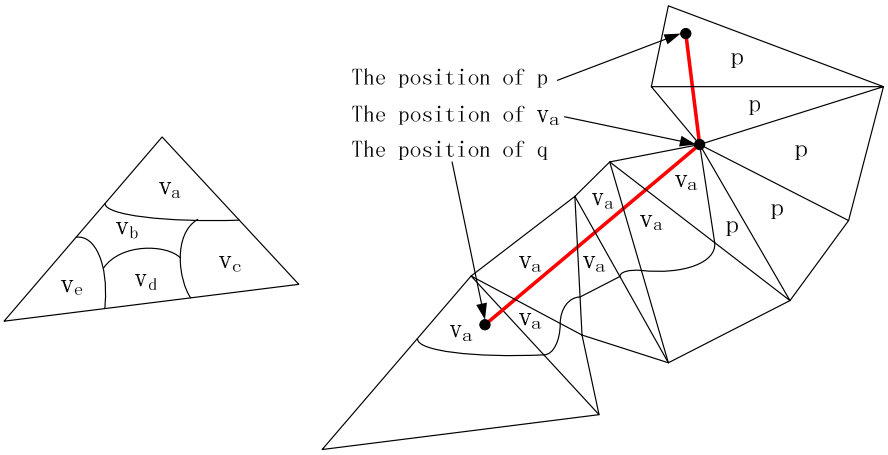

Figure 2: 2D subdivision structure in a triangle mesh T for shortest path computation. Left: the subdivision of a triangle and each subdivision cell D_i is assigned a vertex v_i. Right: tracing the shortest path using the subdivision structure and the assigned vertices.

In above, the saddle vertex is defined as the vertex for which the sum of incident angles is greater than or equal to 2π. One example of the shortest paths in a triangle set $(t_1, t_2, t_3, t_4, t_5)$ is illustrated in Figure 1. For the source point $p \in t_1$, the lines connecting p to vertices v_1, v_2 partition the bottom edge into three intervals I_1, I_2, I_3. If the inquiry point $q \in t_5$ is in the interval of I_1, the shortest path inbetween is a line $\overline{pq}$. If $q \in I_2$ (or $q \in I_3$), the shortest path is a polyline $\overline{pv_1q}$ (or $\overline{pv_1v_2q}$).

Based on the observation of plane unfolding, the MMP algorithm computes a 2D subdivision structure $(D_1, D_2, \cdots)$ on T, which is exhaustiveness $(\bigcup_i D_i = T)$ and semi-mutual exclusion $(X_{ij}^{\circ} = \emptyset, X_{ij} = D_i \cap D_j$, for any $i \neq j$, where X° is the interior of set $X)$. Each 2D subdivision cell D_i is assigned with a point v_i that is the 2D image of either the source point p or a saddle vertex projected onto the plane containing D_i. As illustrated in Figure 2, given such a subdivision structure, the shortest path between source p and an inquiry point q can be efficiently achieved using the following steps:

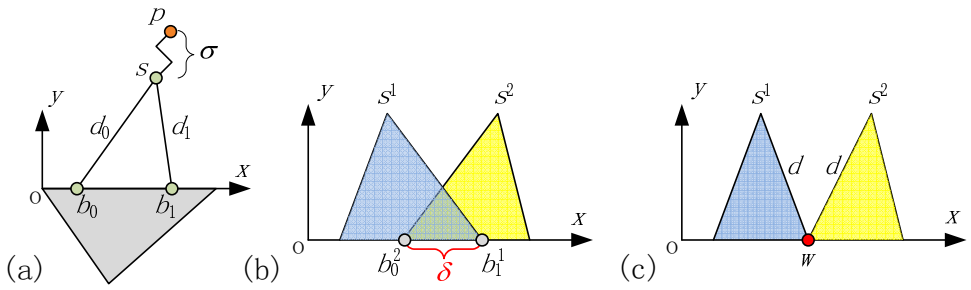

Figure 3: The wedge representation and the intersection of two wedges [12].

S1. Find the subdivision cell D_s containing point q. Set $D_c = D_s$, $r = q$.

S2. Connect r and v_c (the assigned vertex of D_c) by a line l in the plane defined by D_c.

S3. If $r \neq p$, find the intersection x of l and D_c; otherwise stop;

S4. Find the adjacent subdivision D_a of D_c shared the same boundary point x. Set $D_c = D_a$, $r = x$. Go to step S2.

Due to the extreme complexity of the 2D subdivision structure with curved boundaries for each subdivision cell, it is impractical to explicitly build and store this 2D structure on T. Mitchell et al. [9] circumvented this difficulty by only building a 1D subdivision structure on the half edges of T and Surazhskey et al. [12] propose a novel implementation of this 1D structure as summarized below.

Definition 1. *A wedge is an interval on a half edge he of T, defined by 6-tuple $(b_0, b_1, d_0, d_1, \sigma, \tau)$, where b_0 and b_1 are parameters measuring distance along he, d_0 and d_1 are distance from the assigned vertex s to the endpoints b_0 and b_1, respectively, which encodes the 2D position of s, σ is the distance of the shortest path from the source p to s, τ is a binary direction specifying to which side of he the vertex s lies.*

Figure 3(a) illustrates the wedge definition. To compute the 1D edge subdivision, from the triangle containing the source p, initial wedges can be identified and propagated. During the wedge propagation, the new derived wedges may intersect some existing wedges. Let w_1, w_2 be two intersected wedges with overlap δ (Figure 3(b)). Surazhsky et al. [12] use the following rules to update w_1, w_2:

C1. If one of the wedges has a larger distance value entirely over δ, then simply cut δ from that wedge;

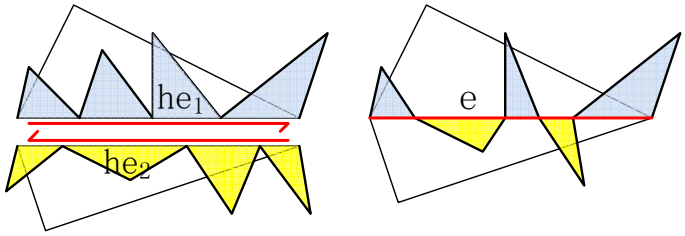

Figure 4: Merge the wedges in two half-edges he_1, he_2 into one set of wedges in an edge e.

C2. If the case C1 does not hold, then update two wedges using the new separating point w (Figure 3(c)) computed by

$$\sqrt{(q - s_{1x})^2 + s_{1y}^2} + \sigma_1 = \sqrt{(q - s_{2x})^2 + s_{2y}^2} + \sigma_2 \tag{1}$$

The solution of Equation (1) is the intersection points of a branch of hyperbola with the edge. The wedge propagation and updating can be performed using a priority queue in a continuous Dijkstra fashion. Mitchell et al. [9] proved that

- The propagation algorithm will generate correct solutions that all half edges of T are completely covered by wedges.
- The total number of wedges in T is bounded by $O(n^2)$, where n is the number of triangles in T.

The MMP algorithm solves a single-source all-destinations geodesic problem on T. It can be naturally extended to a multiple-sources geodesic field solution by propagating initial wedges simultaneously from multiple source points on T [21].

3 Edge-based exact geodesic metric on T

The wedge structure on half edges of T provides the exact geodesic metric on T. To establish and store the exact geodesic metric via wedge structure, the MMP algorithm [9] split each edge e of T into two halves and each half edge corresponds to one of two incident triangles of e. Let q be a point on e incident to triangles t_1, t_2. The necessity of using half edges is that the shortest path from source p to q that approaches q by crossing t_1 or t_2 must be treated separately[1] in the MMP algorithm.

If wedges are constructed using a half-edge data structure, all wedges on one half edge must have the assigned saddle vertices lying on the same side of that half edge (ref. Figure 4 left). In this case the binary direction τ of assigned saddle vertex in Definition 1 can be

[1]The two paths are called t_1-free and t_2-free paths in [9].

ignored. However, Surazhsky et al.'s implementation [12] use a half-edge data structure and a direction τ in their wedge definition simultaneously: this reveals a deficiency in their work. Surazhsky et al. [12] presented that for wedge updating, if the case C2 happens, then a single solution exists in the intersection region δ by solving Equation (1). This claim is only true for the half edge's wedge intersection and two intersected wedges must satisfy certain conditions. We re-examine this claim in Property 6 in Section 4.

In this work, we study wedges in two stages:

- Wedge propagation stage. In this stage, much more wedges than those in the final stage will be generated, updated or possibly deleted later.
- Final stage. In this stage, only those necessary wedges for answering single-source all-destinations inquiries are kept.

In this section, we show that wedges on two incident half-edges can be merged into one set of wedges on an edge (ref. Figure 4 right) and the exact geodesic metric structure still holds. In particular, we show that this half-edge merging is correct at both wedge propagation and final stages. We prove the correctness using the following three steps:

- Step 1. Based on the correctness of the MMP algorithm [9], at the final stage, each half edge of T is completely covered by wedges.
- Step 2. Based on the correctness of Step 1, we prove that wedges on two incident half-edges can be merged into one set of wedges on an edge at the final stage (Properties 1 and 2).
- Step 3. Based on the correctness of Step 2, we prove that at the wedge propagation stage, wedges can be propagated and updated using an edge-based data structure (Properties 3 and 4).

The merit of wedge merging at Step 2 is to save data storage for geodesic path inquiries. The merit of wedge propagation using an edge-based data structure at Step 3 is to reduce the number of unnecessary wedges to be propagated at the wedge propagation stage, and thus speed up the MMP algorithm.

Property 1. *The set of wedges on an edge, by merging the wedges on two incident half-edges at the final stage with intersection updating using rules C1 and C2, offer a correct 1D subdivision on all edges in T.*

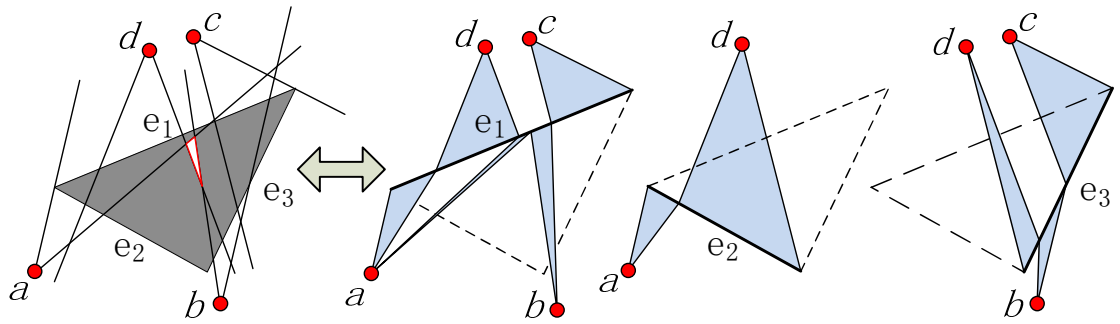

Figure 5: A paradox of 2D face subdivision induced from 1D edge subdivision using wedges.

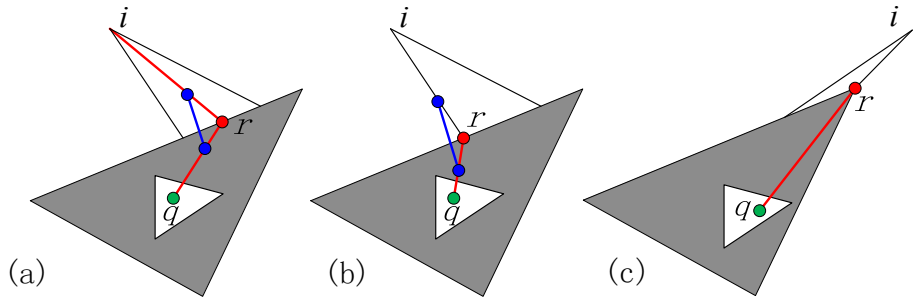

Figure 6: Proof of Property 2: r lies inside the wedge (a) or sit on the boundary of the wedge (b) or sit on a saddle vertex (c).

Proof. Refer to Figure 4. Based on the correctness of the MMP algorithm [9], at the final stage the wedges on two incident half-edges he_1 and he_2 completely cover the two half-edges exhaustively and semi-mutual exclusively. Then merging these wedges on he_1 and he_2 into one set of wedges on edge e completely cover e. For any point q on an edge in T, its shortest path to source p is clearly indicated by a wedge containing q on that edge. For any point q inside a triangle t in T, its shortest path $\overline{pq}$ to source $p \notin t$ must cross t. Let the intersection point of $\overline{pq}$ and t be r. Since $\overline{pq}$ is a shortest path, $\overline{pr}$ must be a shortest path too. Then r is properly covered by a wedge which indicates the short path $\overline{pr}$. □

The exact geodesic metric on T relies on a 2D subdivision as illustrated in Figure 2. However, a complete covering of 1D edges in T using wedges may not induce a complete 2D subdivision of triangles in T. Figure 5 shows such an example. Let w_a, w_b, w_c, w_d denote the wedges with assigned saddle vertices a, b, c, d, respectively. For 1D subdivision on the edges, edge e_1 is completely covered by wedges (w_a, w_b, w_c, w_d), e_2 by (w_a, w_d), e_3 by (w_b, w_c, w_d). For the induced 2D subdivision inside the triangle, there is a vacuum area (shown in white color in Figure 5 left) which is not covered by any wedges. To answer this puzzle, the following property is in order.

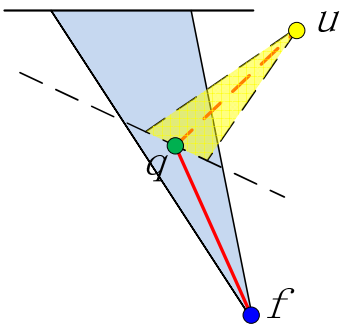

Figure 7: Proof of Property 3: an edge-based wedge w_f at the final stage (shown in blue color) is updated by a wedge w_u (shown in yellow color) during the edge-based wedge propagation process.

Property 2. *The 1D subdivision on edges in T, resulted from merging wedges on half edges at the final stage, induces a correct 2D subdivision inside all triangles in T. Furthermore, each planar 2D subdivision area induced from a wedge w_v is star-shaped with respect to the assigned saddle vertex v in the same plane.*

Proof. Assume that a vacuum area exists inside the triangle t from the induced 2D subdivision, as shown in Figure 5. Let q be a point inside this vacuum area. Without loss of generality, assume that the source p is outside t. The shortest path from p to q must intersect the boundary edges of t and let r be the intersection point. Since the edges are completely covered by wedges, point r must be inside a wedge w_i or in the boundary of w_i, as shown in Figure 6. By assumption, q cannot be in the extended line from i to r. So in the cases shown in Figure 6, the shortest path from p to q at point r form an angle which is not equal to π in the unfolded plane. If r is not a mesh vertex, then a shortcut must exist that offers a shorter path (Figure 6(a) and (b)), a contradiction. If r is a vertex (Figure 6(c)), then r is a saddle vertex and q is star-shaped with respect to r. □

Property 3. *At the wedge propagation stage, the wedges on two incident half-edges can be merged into one set of wedges on an edge. I.e., the resulting edge-based wedges at the final stage induce a correct 2D subdivision (and a correct geodesic metric structure) inside all triangles in T.*

Proof. Based on Property 2, the edge-based wedges at the final stage by merging half-edge-based wedges induce a correct 2D subdivision in T. If Property 3 does not hold, there exists at least one wedge in the edge-based wedges at the final stage which is modified or removed during the edge-based wedge propagation process. Let w_f be such a wedge (shown in blue in Figure 7) at the final stage. If w_f is updated (modified or deleted) by

a wedge w_u during the edge-based wedge propagation process, let q be a point sitting on the updated area (green point in 7). The updating means that the path from q to source p through wedge w_u is shorter than the path through wedge w_f. However, since w_f is a wedge existing at the final stage, the path from q to source p through wedge w_f should be shortest, a contradiction. □

In addition to its theoretical value, we found that Property 3 is very useful in practice, since in our experiments more than 29% unnecessary wedges can be pruned during the edge-based wedge propagation process and the MMP algorithm is speeded up by reducing 44% running time on average. More details are presented in Section 5.

The 2D subdivision structure inferred from edge-based wedge information offers a geodesic metric $d : T \times T \to \mathbb{R}$. Let A be a nonempty subset of T. The geodesic distance function for a point $x \in T$ to A is defined as $d_A(x) = \inf_{y \in A} d(x, y)$, which found a wide range of applications in industry [21, 22]. We conclude this section with the following property.

Property 4. *Assume A is a nonempty subset of T. The map $x \mapsto d_A(x)$, $x \in T$, is uniformly Lipschitz continuous in T, i.e.,*

$$\forall x, y \in T, \quad |d_A(x) - d_A(y)| \leq d(x, y)$$

Proof. Let x, y, z be three arbitrary points in T. Denote $\overline{xz}$, $\overline{xy}$, $\overline{yz}$ as the shortest paths that from x to z, x to y, y to z in T, respectively. Since $\overline{xz}$ is the shortest path between x and z, we have $d(x, z) \leq d(x, y) + d(y, z)$. Now, $\forall z \in A, \forall x, y \in T$, let $z^* \in \overline{A}$ such that $\inf_{z \in A} d(y, z) = d(y, z^*)$. Without loss of generality, let $d_A(x) \geq d_A(y)$. Then $d_A(x) = \inf_{z \in A} d(x, z) \leq d(x, z^*) \leq d(x, y) + d(y, z^*) = d(x, y) + d_A(y)$. □

4 Edge-based wedge intersection updating

We have shown that the wedges on half edges can be merged for each edge in T at the wedge propagation stage and the exact geodesic metric still holds at the final stage. These properties (Properties 1-4) have two merits for practical applications:

- Unnecessary wedges are pruned efficiently during the wedge propagation process and the MMP algorithm is speeded up;
- For an existing CAD system which use classic edge-based data structures, the system need not change to a half-edge data structure for adding a new exact geodesic function, but just use the existing edge-based data structure.

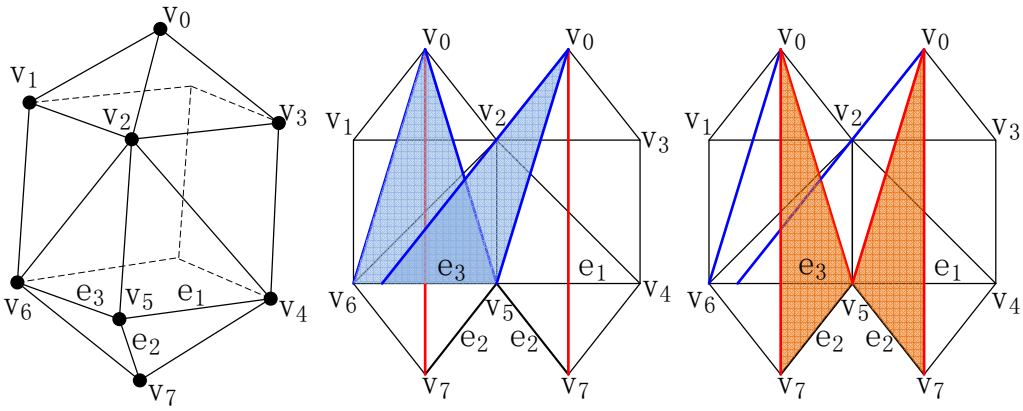

Figure 8: Example 1. Middle: on edge $e_3 = (v_5, v_6)$, source point v_0 contributes two wedges w_1 and w_2 (shown in blue color). Right: on edge $e_2 = (v_5, v_7)$, source point v_0 contributes two wedges w_3 and w_4 (shown in red color).

Compared to the half-edge data structure, special attentions must be paid for edge-based wedge intersection at the wedge propagation stage. We start with the following example.

Example 1. Figure 8 shows a symmetric mesh model. The vertices' positions 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 $\triangle ijk$ denote the triangle formed by vertices i, j, k. Consider edges $e_2 = (v_5, v_7)$ and $e_3 = (v_5, v_6)$. On e_3, source v_0 contributes two wedges w_1, w_2:

- w_1 goes through $\triangle v_0v_1v_2$, $\triangle v_1v_6v_2$, $\triangle v_2v_6v_5$.
- w_2 goes through $\triangle v_0v_2v_3$, $\triangle v_2v_4v_3$, $\triangle v_2v_5v_4$, $\triangle v_2v_6v_5$.

On e_2, source v_0 contributes two wedges w_3, w_4:

- w_3 goes through $\triangle v_0v_1v_2$, $\triangle v_1v_6v_2$, $\triangle v_2v_6v_5$, $\triangle v_5v_6v_7$.
- w_4 goes through $\triangle v_0v_2v_3$, $\triangle v_2v_4v_3$, $\triangle v_2v_5v_4$, $\triangle v_4v_5v_7$.

It is readily seen that in this example,

- The intersection of w_1 and w_2 on e_3 is in the case C1, for which Equation (1) has no solution in the intersection region δ.
- The intersection of w_3 and w_4 on e_2 is in the case C2, for which Equation (1) has an infinite number of solutions in δ.

Definition 2. *At the wedge propagation stage, for a wedge $w = (b_0, b_1, d_0, d_1, \sigma, \tau)$ at an edge e, w is called correct if w offers a true geodesic metric for the interval (b_0, b_1) at e.*

In the Surazhsky et al.'s implementation [12] of the MMP algorithm, for any moment at the wedge propagation stage, there are some wedges existed on the edges and those wedges to be propagated are stored in a priority queue Q. The priority queue Q sorts the wedges to be propagated by their shortest distances back to the source point. Each time the first wedge in Q is popped off and propagated outward across a triangle face. If a wedge w_p is propagated to produce a new wedge w_c, we denote $w_p = Parent(w_c)$ and $w_c = Child(w_p)$. The following property is readily seen.

Property 5. *At the wedge propagation stage, if the longest geodesic distance in an existing wedge w_e is shorter than the shortest geodesic distance in the first wedge w_f in the propagation priority queue Q, then w_e is correct. This property is true for both half-edge and edge-based data structures.*

Proof. Since the propagation of w_f to $Child(w_f)$ will increase the shortest geodesic distance in $Child(w_f)$ to the source point, $Child(w_f)$ cannot update w_e. Meanwhile, after popping off and propagation of w_f, the new wedges that enter into Q can never has their shortest geodesic distances shorter than the shortest geodesic distance in w_f. That completes the proof. □

We find Property 5 is valuable in practice. For previous implementations of the MMP algorithm, to find the geodesic path from one inquiry point q to the source point p, the wedge propagation has to be completely performed over the entire mesh surface. Based on Property 5, once we determine there is a wedge that covers q and is correct, the wedge propagation process can be terminated and accordingly the MMP algorithm is performed in an on-the-fly fashion. This on-the-fly fashion is particularly efficient for those inquiry points near the source p.

For the number of solutions of Equation (1) in the intersection region δ, we have the following property.

Property 6. *In the intersection region δ of two intersected wedges w_1 and w_2 at the wedge propagation stage, the number of solutions of Equation (1) can be zero, one, two or infinite. If both $Parent(w_1)$ and $Parent(w_2)$ are correct and the number of solutions is two or infinite, the two assigned saddle vertices of wedges w_1 and w_2 must be from different sides of the edge.*

Proof. If the intersection of two wedges is in the case C1, then Equation (1) has no solution in δ. For the case C2, the existence of one single solution is given in [12] and the existence

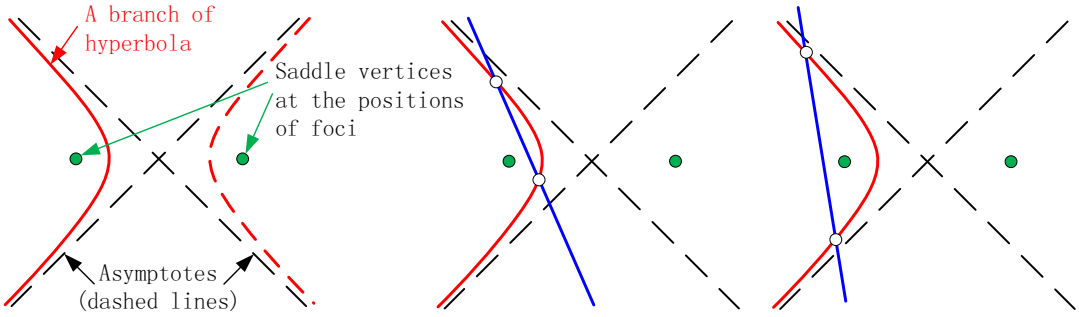

Figure 9: The relation of hyperbola and wedges' saddle vertices. The solution of Equation (1) is the intersection of a branch of hyperbola with the edge, where the wedges' saddle vertices are treated as the foci of the hyperbola (left). Two solutions exist where two saddle vertices lie in different sides of the edge (middle) or lie in the same side of the edge (right).

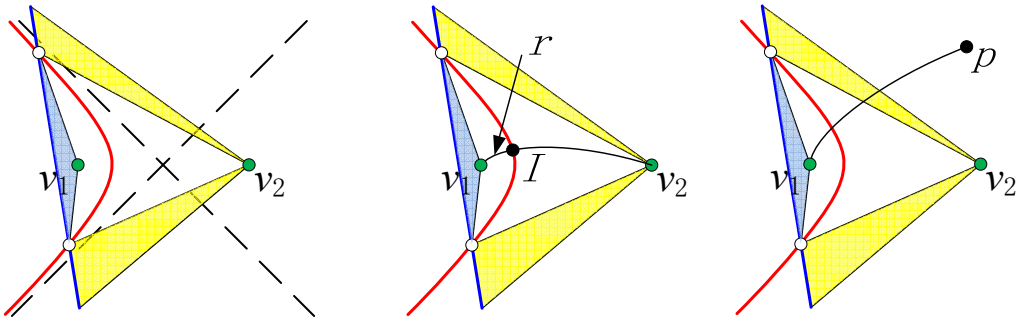

Figure 10: Proof of Property 6. Left: the update of two intersected wedges w_1 and w_2 for the case shown in Figure 9 right. Middle: The shortest path from the saddle vertex V_1 to the source point p goes through the saddle vertex V_2. Right: The shortest path from the saddle vertex V_1 to the source point p does not go through the saddle vertex V_2

of infinite solutions is given in Example 1. There are two possibilities of existing two solutions of Equation (1) in δ. The first case is that two assigned saddle vertices of w_1 and w_2 lie in the different sides of the edge (Figure 9 middle) and the second case is two assigned saddle vertices in the same side of the edge (Figure 9 right). Both cases can occur at the wedge propagation stage. Below we prove that if both $Parent(w_1)$ and $Parent(w_2)$ are correct, the second case does not exist. For the second case, the update of wedges can only be the case shown in Figure 10 left. Let v_1 and v_2 be the saddle vertices of w_1 and w_2, respectively. Since $Parent(w_1)$ is correct, the shortest path from v_1 back to the source is indicated by $Parent(w_1)$. If this shortest path from v_1 passes through v_2 as shown in Figure 10 middle, let this path intersects the branch of hyperbola at the point I. For any point r lying in segment $\overline{v_1 I}$ of the path, r has a shorter path from r to the source by going through v_2 than the path from r to the source through v_1. However, since r lies on the left hand of the branch of hyperbola, the path from r to the source through v_1 should

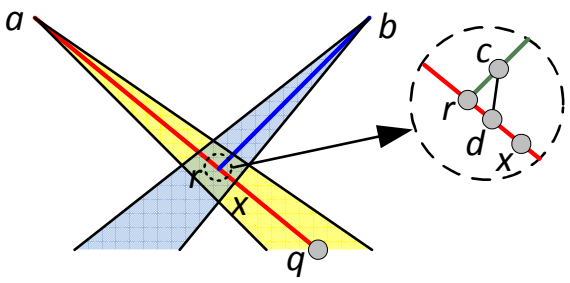

Figure 11: Proof of Property 7: a and b are two assigned saddle vertices of two intersected correct wedges w_a and w_b, respectively, and $\overline{cd}$ is a shortcut of path $r(b) + \overline{rx}$.

be shorter than the path r to the source through v_2, a contradiction. So the shortest path from v_1 to the source cannot pass through v_2, but intersect the wedge $Parent(w_2)$ as shown in Figure 10 right. However, since the shortest paths cannot intersect (see the proof of Property 7), this case does not exist neither. So if the number of solutions is two and $Parent(w_1)$ and $Parent(w_2)$ are correct, the two assigned saddle vertices of wedges w_1 and w_2 must be from different sides of the edge.

If the number of solutions is infinite, Then either the two wedges are identical or the hyperbola solution degenerates to a bisector (that coincides with the mesh edge) between two assigned saddle vertices. The fist case cannot occur at the wedge propagation stage. For the second case, the two assigned saddle vertices lie in the different sides of the edge. □

In the proof of Property 6, we use the following property.

Property 7. *Two correct wedges cannot intersect.*

Proof. Let a and b be two assigned saddle vertices of correct wedges w_a and w_b, respectively. If w_a and w_b intersects, let r be a point inside the intersection area as shown in Figure 11. Denote the shortest path distance from source p to r through a by $r(a)$, and $r(b)$ is similarly defined for the saddle vertex b. In the wedge w_a, we extend the line segment $\overline{ar}$ to some point x which is also in the intersection area. If $r(a) = r(b)$, then there are two equal-distance shortest paths from p to x, i.e., $r(a) + \overline{rx} = r(b) + \overline{rx}$. However, since both r and x are interior points in the nonempty intersection area $w_a \cap w_b$, there exists a shortcut (shown in line segment $\overline{cd}$ in Figure 11) in the path $r(b) + \overline{rx}$; thus contradicts to the assumption $r(a) + \overline{rx}$ is a shortest path. Now without loss of generality, assume $r(a) > r(b)$. In the wedge w_a, we extend the line segment $\overline{ar}$ to a point q which is inside w_a but not in w_b. Then the inequality $r(a) + \overline{rq} > r(b) + \overline{rq}$ means $q(a)$ in w_a is not a shortest path distance; a contradiction. □

5 Experimental results

We have proved several properties which show the MMP algorithm [9] can be performed in an edge-based data structure and the original MMP algorithm can be improved in the following three aspects:

- On-the-fly implementation. If the user specifies any two points p and q on the mesh and running the MMP algorithm by propagating wedges initialized from p, the MMP algorithm needs not to be run over the entire mesh surface: According to Property 5, once a correct wedge covers q, the MMP algorithm can be terminated.

- Time efficiency. According to Property 3, at the wedge propagation stage, the wedges on two incident half-edges can be merged into one set of wedges on an edge. Then some unnecessary wedges can be efficiently pruned during the propagation process and thus the algorithm is speeded up. Our experiments show that using edge-based data structure reduces 44% running time on average when compared to using half-edge data structure.

- Space efficiency. According to Properties 2 and 3, less wedges are stored in an edge-based data structure when compared to those wedges stored in half-edge data structures. Our experiments show that at the wedge propagation stage, the number of total produced wedges is reduced by 29% on average, and at the final stage, the number of stored wedges is reduced by 34% on average.

Below we present the experiments and summarize the experimental results, which demonstrate the time and space efficiency achieved by using the edge-based data structure.

5.1 Benchmark test models

To test the robustness and stability of coding of geodesic metric computation on T, we build a set of benchmark models utilizing visibility graphs [15]. Refer to Figure12 (a) and (b). 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 connecting two nodes v and w if they can see each other. Given two points p, q in S with n nodes, $VG(S \cup \{p, q\})$ can be constructed in $O(n^2 \log n)$ and the shortest path between p, q can be found in $O(n \log n)$ time [15].

We use the shortest planar paths computed from visibility graphs as the ground truth to construct the benchmark models. Let the obstacles S be bounded in a sufficient large

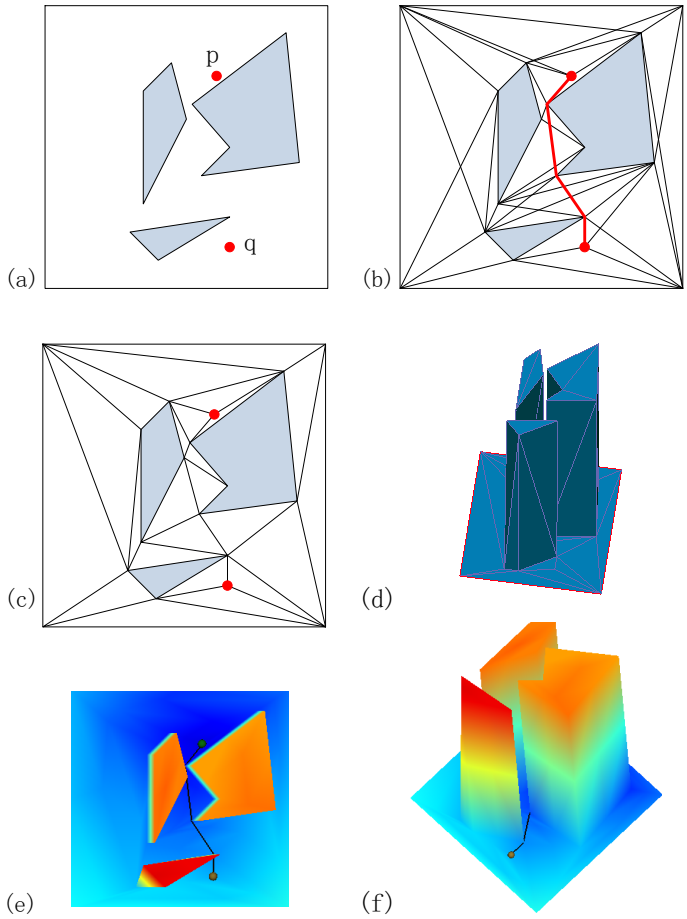

Figure 12: The benchmark model utilizing visibility graphs. (a) A set of obstacles and two source points p, q inside a rectangle. (b) The visibility graph of (a) and the shortest path between p and q. (c) A Triangulation of (a). (d) The benchmark triangle mesh model by lifting each planar obstacle as a polygonal cylinder with a sufficient height. (e, f) Two different views of the shortest path between p and q in the benchmark model, computed by the presented geodesic metric algorithm. The distance field on the model is colored by one-to-one mapping the geodesic distance to an index color map.

rectangle R (Figure 12(a)). A triangulation of p, q, R and the boundaries of S can be constructed in $O(n \log n)$ time (Figure 12(c)). At the place of each obstacle, a polygonal cylinder is lifted and sewed with the planar triangulation along the obstacle's boundary (Figure 12(d)). If the height of obstacle cylinders is sufficient high, the shortest path between p and q will not climb the cylinders; instead, the shortest path will go around the boundary of planar obstacles (Figure 12(e) and (f)). We use both convex and concave obstacles in the benchmark models. By using mesh refinement and simplification, the number of triangles in the benchmark models can be adjusted.

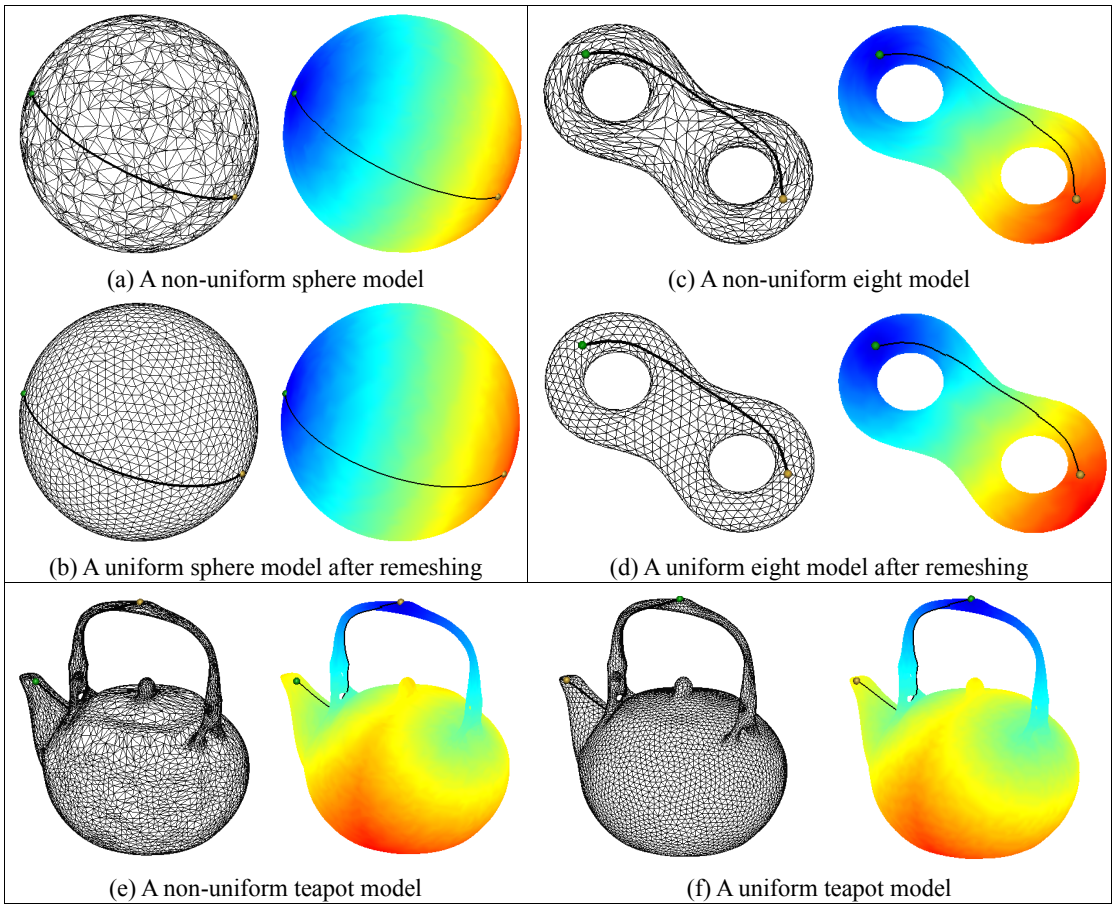

Figure 13: Test models with moderate triangle numbers: sphere (genus-0), teapot (genus-1) and 8-shape (genus-2) models. Each model is shown with two versions: non-uniform and uniform. The distance field on models is shown using color index.

5.2 Comparison of half-edge and edge-based data structures

We have implemented the geodesic metric algorithm on the platform of Visual C++.net in Microsoft Window operating system. The code is available at [2]. We test the code with the following models:

- Ten benchmark models in various complexities: the number of obstacles ranged from 10 to 100 and the number of triangles ranged from 1k to 20k.

- Ten 3D engineering models with moderate triangle numbers (2k to 20k). These models are with various complexities in both geometry (i.e., different curvature distributions) and topology (i.e., different genus numbers). Each model is provided with two types: uniform and non-uniform. Figure 13 shows three examples: On each model, a geodesic path is presented with source point shown in green and tar-

[2] http://cg.cs.tsinghua.edu.cn/people/∼Yongjin/Yongjin.htm

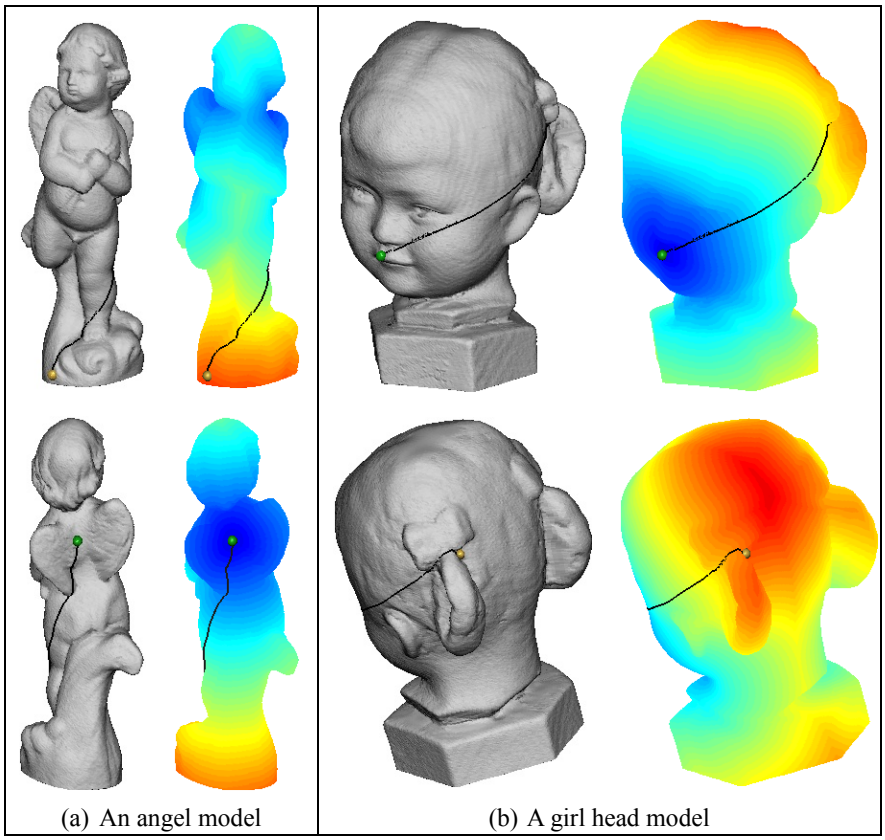

Figure 14: Two large 3D models: an angel and a girl head model. The angel model has 181,148 triangles and 543,444 edges. The girl head model has 384,902 triangles and 1,154,706 edges. On each model, a geodesic path is presented with source point shown in green and target point shown in yellow. By mapping the geodesic distances to a color index, the distance field on each model is also shown with colors.

get point shown in yellow. By mapping the geodesic distances to a color index, the distance field on each model is also shown with colors.

- Five large 3D mesh models (>150k triangles). All models are bounded in a $1.0 \times 1.0 \times 1.0$ cubes. Figure 14 shows two examples.

We randomly sample source points on models and compare the geodesic distances output from following three methods:

- the shortest paths utilizing the visibility graphs (only applicable for benchmark models).
- The GeodesicLib implementation [3] provided by the authors in [12].
- Our implementation using an edge-based data structure.

[3] http://www.cs.technion.ac.il/~vitus/geolib.html

	Face No.	Edge No.	$\overline{l(e)}$	$\sigma[l(e)]$	$\max_e l(e)$	$\min_e l(e)$
Non-uniform eight	2022	6,066	0.079710	0.032766	0.260810	0.020274
Uniform eight	2022	6,066	0.068804	0.007349	0.103375	0.042040
Non-uniform sphere	3996	11,988	0.089625	0.044386	0.369710	0.001317
Uniform sphere	4074	12,222	0.085278	0.007969	0.148602	0.073768
Non-uniform teapot	11666	34,998	0.044716	0.027207	0.571824	0.000994
Uniform teapot	11667	35,001	0.042054	0.012402	0.103145	0.002287
Angel	181147	543,444	0.007005	0.002620	0.086559	0.000011
Girl head	384902	1,154,706	0.007360	0.002791	0.082696	0.000001

Table 1: The statistic data of test models shown in Figures 13 and 14. All these models are normalized in a $1.0 \times 1.0 \times 1.0$ cube: $\overline{l(e)}$ is the average edge length of each model, $\sigma[l(e)]$ is the standard deviation of $l(e)$ which gives a measure of the uniformness of the model, $\max_e l(e)$ is the maximal edge length in the model, and $\min_e l(e)$ is the minimal edge length in the model.

Our test results show that among more than one hundred thousands of computed geodesic values, the results output from three methods are the same upon the floating point machine precision. These results experimentally demonstrate that the coding of our edge-based wedge intersection is correct.

Since the GeodesicLib for the implementation [12] outputs geodesic paths in a text file without the wedge information, below we use our implementation of the MMP algorithm to compare the performance between edge-based and half-edge data structures. The performance data given below are all tested with an off-the-shelf PC with INTEL I7-2600K CPU, running at 3.40GHz with 4GB RAM.

To evaluate the statistic data of wedge information, we use the 3D engineering models shown in Figure 13 that are presented in two formats: non-uniform and uniform. The statistic data of these mesh models is summarized in Table 1, which also include the two large models shown in Figure 14. All models are normalized in a $1.0 \times 1.0 \times 1.0$ cube. In Table 1, by regarding the length of each edge in a mesh model as a random variable, the statistic data includes the mean $\overline{l(e)}$ of $l(e)$ (i.e., the average edge length), the standard deviation $\sigma[l(e)]$ (which gives a measure of the edge uniformness in the model), the maximal edge length $\max_e l(e)$ in the model and the minimal edge length $\min_e l(e)$ in the model.

We first compare the space efficiency between half-edge and edge-based data structures, using two measures:

- The number of total wedges produced at the wedge propagation stage. It gives the upper bound of storage for running the MMP algorithm. Table 2 summarizes

	Wedges produced at the wedge propagation stage		Percentage of
	Half-edge structure	Edge-based structure	wedge reduction
Non-uniform eight	21059	12643	40.0%
Uniform eight	19289	11809	38.8%
Non-uniform sphere	117045	101227	13.5%
Uniform sphere	116253	103108	11.3%
Non-uniform teapot	172111	129810	24.6%
Uniform teapot	165996	128409	22.6%
Angel	1530506	935655	38.9%
Girl head	3271066	1972283	39.7%

Table 2: The comparison of the number of total wedges produced at the wedge propagation stage for using half-edge and edge-based data structures. These data is generated by averaging on the data over 100 trials, each time a random point on the model is selected to initialize the MMP algorithm.

	The number of wedges existed at the final stage		Percentage of
	Half-edge structure	Edge-based structure	wedge reduction
Non-uniform eight	11846	6481	45.3%
Uniform eight	11513	6578	42.9%
Non-uniform sphere	53167	43539	18.1%
Uniform sphere	53371	44468	16.7%
Non-uniform teapot	80732	55866	30.8%
Uniform teapot	76830	58304	24.1%
Angel	947923	504873	46.7%
Girl head	1995957	1056117	47.1%

Table 3: The comparison of the number of wedges existed at the final stage for using half-edge and edge-based data structures. These data is generated by averaging on the data over 100 trials, each time a random point on the model is selected to initialize the MMP algorithm.

this type of data for eight models listed in Table 1. These data shows that using edge-based data structure can on average reduce 29% wedges produced at the wedge propagation stage, when compared to the use of half-edge data structure.

- The number of wedges existed at the final stage. It gives the upper bound of storage for computing geodesic paths using the MMP algorithm. Table 3 summarizes this type of data for eight models listed in Table 1. These data shows that using edge-based data structure can on average reduce 34% wedges existed at the final stage, when compared to the use of half-edge data structure.

Secondly we compare the time efficiency between half-edge and edge-based data structures, using the measure of running time that is spent for all wedge computation and propagation. Table 4 summarizes this type of data for eight models listed in Table 1. These data shows that using edge-based data structure can on average reduce 44% run-

	Running time of the MMP algorithm (Sec.)		Percentage of
	Half-edge structure	Edge-based structure	time reduction
Non-uniform eight	0.03976	0.01985	50.1%
Uniform eight	0.03533	0.01756	50.3%
Non-uniform sphere	0.18345	0.11741	36.0%
Uniform sphere	0.17651	0.11464	35.1%
Non-uniform teapot	0.30973	0.17957	42.0%
Uniform teapot	0.28969	0.16996	41.3%
Angel	3.06657	1.57997	48.5%
Girl head	6.70950	3.47284	48.2%

Table 4: The comparison of running time spent for all wedge computation and propagation, for using half-edge and edge-based data structures. These data is generated by averaging on the data over 100 trials, each time a random point on the model is selected to initialize the MMP algorithm.

ning time of the MMP algorithm, when compared to the use of half-edge data structure.

6 Conclusions

Previous work [9, 10, 12, 13] that computes the exact geodesic metric on a 2-manifold mesh T is only applicable in half-edge data structures. In this paper, we show that the computation of exact geodesic metric is also applicable in edge-based data structures. A direct merit is that some existing CAD systems which use edge-based data structures [19] can simply add a new function for computing exact geodesic without changing the kernel completely to a half-edge data structure. We also show that edge-based implementation of the MMP algorithm can achieve three merits: (1) To find the geodesic path between any two points on T, the MMP algorithm can be run in an on-the-fly fashion; (2) unnecessary wedges can be efficiently pruned during the edge-based wedge propagation process and the MMP algorithm is speeded up; (3) Less wedges need to be stored for establishing the exact geodesic metric on T and thus improving the space efficiency. Experimental results are presented showing that when compared to the half-edge data structure, the edge-based implementation can reduce 44% running time and 29% storage on average.

7 Acknowledgement

The author thanks the reviewers for their constructive comments that help improve this paper. The author also thanks Mr. Wen-Qi Zhang and Chun-Xu Xu for their hard work to code the presented algorithm and summarize the experimental results. This work is supported by the National Basic Research Program of China (2011CB302202), the Natural Science Foundation of China (61272228) and the 863 program of China (2012AA011801).

This work is also partially supported by Program for NCET and TNList Cross-discipline Foundation.

References

[1] J. Mitchell. Geometric shortest paths and network optimization. In J.R. Sack, J. Urrutia, (eds.) Handbook of Computational Geometry, Elsevier Science, pp. 633-702, 2000.

[2] J. Hershberger and S. Suri. Practical methods for approximating shortest paths on a convex polytope. Proc. 6th ACM-SIAM Symposium on Discrete Aglorithms, pp.447-456, 1995.

[3] P.K. Agarwal, S. Har-Peled, M. Sharir and K. Varadarajan. Approximating shortest paths on a convex polytope in three dimensions. Journal of the ACM, Vol. 44 No. 4, pp. 567-584, 1997.

[4] S. Har-Peled. Approximate shortest paths and geodesic diameters on convex polytopes in three dimensions. Proc. 3rd Annual Symposium on Computational Geometry, pp. 359-365, 1997.

[5] P.K. Agarwal, S. Har-Peled and M. Karia. Computing approximate shortest paths on convex polytopes. Proc. 6th Annual Symposium on Computational Geometry, pp. 270-279, 2000.

[6] S. Har-Peled. Constructing approximate shortest path maps in three dimensions. SIAM Journal on Computing, Vol. 28, No. 4, pp. 1182-1197, 1999.

[7] R. Kimmel and J.A. Sethian. Computing geodesic paths on manifolds. Proceedings of National Academy of Science, Vol. 95, No. 15, pp. 8431-8435, 1998.

[8] M. Sharir and A. Schorr. On shortest paths in polyhedral spaces. SIAM Journal on Computing, Vol. 15, No. 1, pp. 193-215, 1986.

[9] J. Mitchell, D. Mount, C. Papadimitriou. The discrete geodesic problem. SIAM Journal on Computing, Vol. 16, No. 4, pp. 647-668, 1987.

[10] J. Chen and Y. Han. Shortest paths on a polyhedron. Proc. 6th Annual Symposium on Computational Geometry, pp. 360-369, 1990.

[11] S. Kapoor. Efficient computation of geodesic shortest paths. Proc. 31st Annual ACM Symposium on Theory of Computing, pp. 770-779, 1999.

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

[13] S.H. Qin and G.J. Wang. Improving Chen and Hans algorithm on the discrete geodesic problem. ACM Transactions on Graphics, Vol. 28, No. 4, Article 104, 2009.

[14] Y.J. Liu, Q.Y. Zhou and S.M. Hu. Handling degenerate cases in exact geodesic computation on triangle meshes. The Visual Computer, Vol. 23, No.9-11: 661-668, 2007.

[15] M. de Berg, M. van Kreveld, M. Overmars and O. Schwarzkopf. Computational geometry: algorithms and applications. Springer, 1997.

[16] M. Botsch, S. Steinberg, S. Bischoff and L. Kobbelt. OpenMesh – a generic and efficient polygon mesh data structure. 1st OpenSG Symposium, 2002.

[17] F.P. Preparata and M.I. Shamos. Computational geometry: an introduction. Springer-Verlag, 1985.

[18] B.G. Baumgart. Winged-edge polyhedron representation. Technical Report, STAN-CS-320, Stanford University, 1972.

[19] K. Weiler. Edge-based data structures for solid modeling in curved-surface environments, IEEE Computer Graphics and Applications, Vol. 5, No. 1, pp. 21-40, 1985.

[20] L. Kettner. Designing a data structure for polyhedral surfaces. Proc. 14th Annual Symposium on Computational Geometry (SCG'98), pp.146-154.

[21] Y.J. Liu, Z.Q. Chen and K. Tang. Construction of iso-contours, bisectors and Voronoi diagrams on triangulated surfaces. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 33, No. 8, pp. 1502-1517, 2011.

[22] H. Pottmann, S. Leopoldseder, M. Hofer, T. Steiner and W. Wang. Industrial geometry: recent advances and applications in CAD. Computer-Aided Design, Vol. 37, No. 7, pp. 751-766, 2005.