Construction of Iso-contours, Bisectors and Voronoi Diagrams on Triangulated Surfaces

Yong-Jin Liu, Zhan-Qing Chen, and Kai Tang

Abstract—In the research of computer vision and machine perception, three-dimensional objects are usually represented by 2-manifold triangular meshes $\mathcal{M}$. In this paper, we present practical and efficient algorithms to construct iso-contours, bisectors and Voronoi diagrams of point sites on $\mathcal{M}$, based on an exact geodesic metric. Compared to Euclidean metric spaces, the Voronoi diagrams on $\mathcal{M}$ exhibit many special properties that fail all the existing Euclidean Voronoi algorithms. To provide practical algorithms for constructing geodesic-metric-based Voronoi diagrams on $\mathcal{M}$, this paper studies the analytic structure of iso-contours, bisectors and Voronoi diagrams on $\mathcal{M}$. After a necessary preprocessing of model $\mathcal{M}$, practical algorithms are proposed for quickly obtaining full information about iso-contours, bisectors and Voronoi diagrams on $\mathcal{M}$. The complexity of the construction algorithms is also analyzed. Finally three interesting applications, surface sampling and reconstruction, 3D skeleton extraction and point pattern analysis are presented that show the potential power of the proposed algorithms in pattern analysis.

Index Terms—Shape, geometric transformations, triangular meshes, exact geodesic metrics, point patterns.

1 Introduction

Voronoi diagram is an elegant spatial structure which has found diverse applications in a variety of disciplines in natural science, including pattern recognition, motion planning, operational research, information retrieval, biological morphology, and so on. Various extensions and derived distance transforms make the Voronoi diagram a basic and appealing tool. In Euclidean space, medial axis transformations [34], [66], [20] and generalized Euclidean distance transformations [19], [41], [46] are widely studied for digital images and volume data. For spaces with non-Euclidean metrics, the domain of Voronoi diagrams has also been extended to spheres [4], [50], polyhedral surfaces [49], [30], [67], parametric surfaces [33], hyperbolic spaces [53] and the general Riemannian manifolds [65], [36], [52]. For detailed surveys, the reader is referred to [5], [51] and the references therein. In this paper, we study a class of Voronoi diagrams on a triangulated 2-manifold setting and propose practical and efficient algorithms to compute them.

Recently with the rapid development of remote sensing and laser scanning techniques, many complex 3D objects, terrains and scenes are modeled by dense triangular meshes [16], [31]. Shifting Voronoi diagrams from Euclidean space such as images to 2-manifold triangulated surfaces presents significant challenges and plays an important role in point pattern analysis and spatial optimization (see Figure 1 for an example). The first challenge concerns the distance metric. On 2-manifold surfaces, a natural and widely used metric is the geodesic distance. Research on geodesic computation of mesh models can be dated back to twenty years ago [45] and our presented work makes use of the MMP algorithm [45] that outputs exact geodesic paths on triangulated surfaces. The second challenge is regarding the special structures inherent in the Voronoi diagram on triangulated surfaces that make it unique and distinct from its Euclidean counterpart. E.g., in Euclidean plane three points not lying on a line uniquely determine a circumcircle. However, there may be no such geodesic circle or many geodesic circles existing on 2-manifolds. The special analytic structure of Voronoi diagram on triangulated surfaces is analyzed in this paper.

We make two contributions in this paper:

1) An analytic structure is analyzed and presented for Voronoi diagrams on triangulated surfaces $\mathcal{M}$. The relations between iso-contours, bisectors and Voronoi diagrams on $\mathcal{M}$ are also established. Details are presented in Section 4.
2) Efficient and practical algorithms are presented to compute iso-contours, bisectors and Voronoi diagrams with the proposed analytic structure. Details are presented in Section 5.

The distinct properties of Voronoi diagrams on triangulated surfaces $\mathcal{M}$ make them interesting and attractive in many pattern analysis applications. In Section 6, Three interesting applications, surface sampling and reconstruction, 3D skeleton extraction and point pattern analysis, are presented that show the potential power of applying Voronoi diagrams on $\mathcal{M}$ in pattern analysis.

2 Related Work

On a 2-manifold surface $\mathcal{M}$, the shortest path between two points on $\mathcal{M}$ is a geodesic on $\mathcal{M}$. While the gen-

- *Y.J. Liu is with the Tsinghua National Lab for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, Beijing, China. E-mail: liuyongjin@tsinghua.edu.cn*
- *Z.Q. Chen and K. Tang are with Hong Kong University of Science and Technology, Hong Kong. E-mail: {vitachen, mektang}@ust.hk*

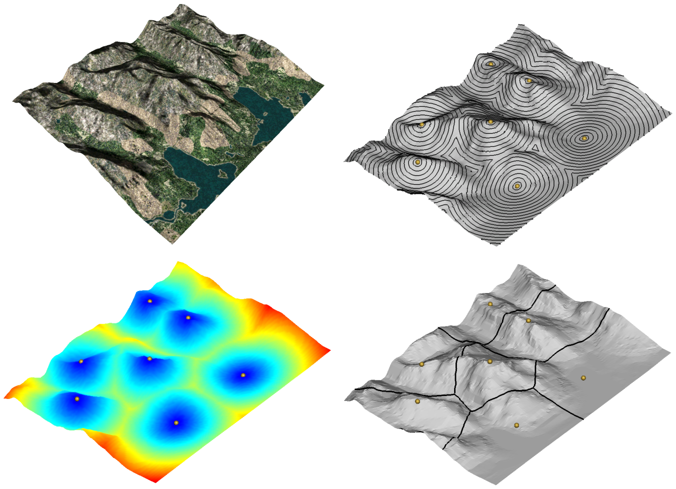


Fig. 1. Iso-contours, distance field and Voronoi diagram of seven point sites on a 2-manifold terrain model of 32,258 triangles. Top-left: the texture mapped model. Top-right: the iso-contours with seven point sources. Bottom-left: the distance field mapped by a color index. Bottom-right: the Voronoi diagram on the terrain surface.

eral problem of computing the shortest paths among polyhedral obstacles in $\mathbb{R}^3$ is $\mathcal{NP}$-hard [9], computing a geodesic on $\mathcal{M}$ can be solved in polynomial time. Notably two classes exist for geodesic computation on $\mathcal{M}$: approximation and exact algorithms. Approximation algorithms are characterized by the approximation ratio ϵ, i.e., the length of computed approximation solution is at most $1+\epsilon$ times the exact solution. Two typical $1+\epsilon$ approximation algorithms running in sub-cubic time are proposed in [1], [21]. A polyhedral surface can also be viewed as a linear approximation of an underlying smooth 2-manifold and thus numerical algorithms for solving the Eikonal equation on triangular or quadrilateral grids can be used. The work in this direction is exemplified by the fast marching methods [56], [29]. A survey on approximate geodesic computation is presented in [44].

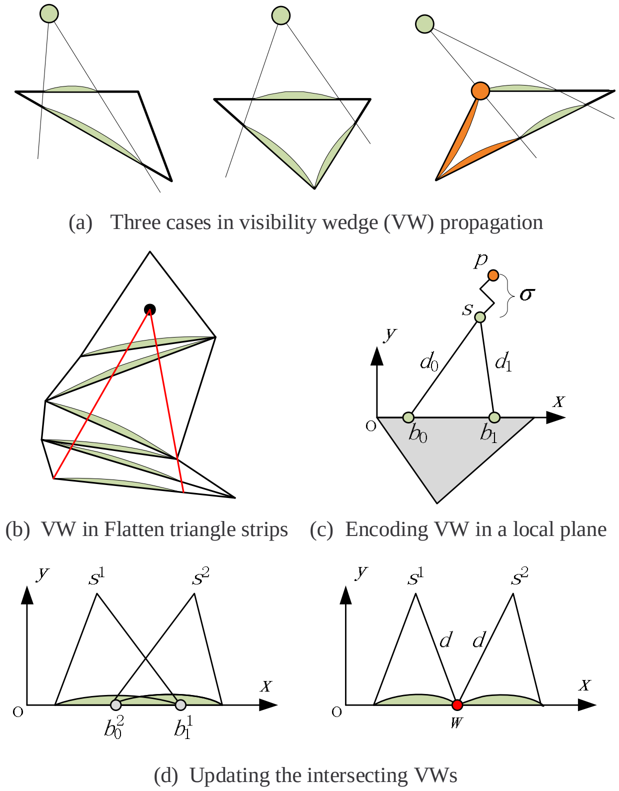
Exact geodesic computation on general polyhedral surfaces was first studied by Mitchell et al. [45], in which an $O(n^2 log^2 n)$ algorithm was proposed, where n is the face number in $\mathcal{M}$. Later, several researchers improved this bound to $O(n^2)$ [11], [27] and $O(nlog^2 n)$ [28]. Recently, Surazhsky et al. [59] presented a novel implementation of the MMP algorithm in [45] and showed that, in practice, it runs much faster than the other algorithms. Between two points on $\mathcal{M}$, there could be several geodesics connecting them. Balasubramanian et al. [7] proposed an LOS-Floyd algorithm that runs in cubic time and can report all geodesic paths between two arbitrary points on $\mathcal{M}$.

In this paper, we propose practical algorithms for computing the Voronoi diagrams on 2-manifold triangular meshes based on exact geodesic distance [45], [59]. Distinct from Euclidean cases, Voronoi diagrams on triangulated surfaces possess many unique properties. Mount [49] first studied some of these properties, showing that Voronoi diagrams on $\mathcal{M}$ with m point sites have the complexity $O(m(m+n))$. In the worst case, the bisector between two point sites on $\mathcal{M}$ has the complexity $\Omega(n^2)$ [48]. A recent study [10] reveals that the sum of the combinatorial complexities of the order-j Voronoi diagrams on S, for $j = 1, 2, \cdots, k$, is $O(k^2n^2 + k^2m + knm)$. Moet et al. [48] and Aronov et al. [3] studied a class of realistic terrains, which is a special kind of triangulated surface, showing that the worst-case complexity is $\Theta(n)$ for a bisector and $\Theta(n + m\sqrt{n})$ for a Voronoi diagram, respectively, on realistic terrains. Although rigorous constructive proofs are presented in [49], [48], [3], [10], they are nevertheless of greater theoretical than practical interest, because the constructions did not offer practical algorithms to explicitly build Voronoi diagrams on general triangulated surfaces $\mathcal{M}$ with concise data structures.

In terms of applications in pattern analysis for Voronoi diagrams on $\mathcal{M}$, little work exists since there has been no practical construction algorithms in previous work. Peyre and Cohen [54] use recursively farthest points [15], [47] to sample the surfaces and use Voronoi-Delaunay duality [36] to remesh and parameterize the triangulated surfaces. Approximate geodesic computation using the fast marching method [29] was adopted in [54] for sampling. Since the work in [54] uses the approximate geodesic distance and the work in [38] uses the Euclidean distance instead of geodesic distance, both methods can produce potentially large errors if the triangles in $\mathcal{M}$ are extremely slivered. In this work, we re-examine the uniform sampling strategy in Section 6.1 using the exact geodesic computation. Hilaga et al. [22] proposed a multiresolutional Reeb graph to estimate the similarities of 3D shapes by topological matching. In [22], single-source shortest paths along edges on $\mathcal{M}$, output from the classical Dijkstra's algorithm [12], are used as a rough approximation of geodesic paths and therefore, the meshes of shapes have to be uniformly densified, which also lead to a high computational load. In Section 6.2, with the tools of building Voronoi diagrams on $\mathcal{M}$, we propose a surface skeletonization method that simplifies the skeleton's topological structure from a mixed cell-complex in $\mathbb{R}^3$ [20] to a 1D axis structure akin to the planar smooth ones in [61], [6]. Voronoi diagrams on $\mathcal{M}$ can also be used in point pattern analysis [64] and we examine this case with examples in Section 6.3.

3 Exact Geodesic Metric on $\mathcal{M}$ based on MMP Algorithm in [45], [59]

To make the paper self-contained and more easily readable, in this section we briefly summarize the novel implementation in [59] of the MMP algorithm [45] to establish the exact geodesic metric on triangulated 2-manifold surfaces $\mathcal{M}$. The surfaces $\mathcal{M}$ studied in this



(a) Three cases in visibility wedge (VW) propagation

(b) VW in Flatten triangle strips (c) Encoding VW in a local plane

(d) Updating the intersecting VWs

Fig. 2. Geodesic paths encoding with visibility wedge propagation [59].

paper are compact piecewise flat surfaces. The Hopf-Rinow theorem [24] and its adaption to general piecewise flat surfaces [2] ensure that a minimal geodesic exists between two arbitrary points on this kind of surfaces. Denote the topology of a triangular mesh surface $\mathcal{M}$ by (V, E, F), where V, E, F, are the vertex, edge and face sets, respectively.

Given a surface $\mathcal{M}$ and one vertex $v \in V$, the MMP algorithm [45] establishes a distance function D_v on $\mathcal{M}$ such that for any point $q \in \mathcal{M}$, $D_v(q)$ is the exact geodesic distance from q to v on $\mathcal{M}$. The basic idea of the MMP algorithm is to partition all faces in $F \in \mathcal{M}$ into a 2D subdivision structure. To establish this structure, the following property is used. Inside every triangle in $\mathcal{M}$, the geodesics 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 (Figure 2b).

Definition 1: The vertices through which geodesics pass are called *pseudo-sources* in this paper. *Singular vertices* are those in V whose total surrounding angle is larger than or equal to 2π.

Definition 2: Given a source p and any strip of unfolded triangles starting at p, a *visibility wedge* is the set of points on the strip that are visible from p.

From the triangles containing one or multiple sources p, a set of initial visibility wedges (VWs) are identified. These VWs are propagated (Figure 2a) until all the edges E in $\mathcal{M}$ are covered. During the VW propagation, three different cases as shown in Figure 2a would arise. It is proved in [45] that the pseudo-source of each VW can only be the singular vertices in V. To store the VW information in the local plane defined by each triangle, a 8-tuple $(b_0, b_1, d_0, d_1, \tau, \sigma, Id_{nv}, Id_{pt})$ is used in this paper (Figure 2c), where b_0, b_1 are parameters measuring the distance along the edge, the 2D unfolded position of the nearest pseudo-source s is encoded by its distances d_0, d_1 to the endpoints b_0, b_1, respectively, Id_{nv} and Id_{pt} are the identifiers of s and the original point site p, respectively, τ specifies the side of edge on which s lies, and σ is the length of the geodesic path from $s = Id_{nv}$ back to the site $p = Id_{pt}$. During the VW propagation, the new emerging wedges may intersect some existing wedges. Any two intersected wedges $(b_0^i, b_1^i, d_0^i, d_1^i, \sigma^i)$, $i = 1, 2$, are updated by solving the equation with unknown w (Figure 2d):

$$\sqrt{(w - s^1.x)^2 + (s^1.y)^2} + \sigma^1 = \sqrt{(w - s^2.x)^2 + (s^2.y)^2} + \sigma^2$$

The solution is the intersection point of a branch of hyperbola with the x axis.

Given one source point p [1], the VW propagation builds a 2D subdivision structure $(D_1, D_2, \cdots, D_n)$ on $\mathcal{M}$ that satisfies $\bigcup_{i=1}^{n} D_i = \mathcal{M}$ and $D_i \cap D_j = \emptyset$, $i \neq j$, $i, j = 1, 2, \cdots, n$. Each subdivision D_i has a corresponding $Id_{nv}(i)$ that is stored as a local 2D projection nv_i on each D_i. Given an arbitrary target position q on $\mathcal{M}$, the geodesic path between p and q is computed as follows.

1) Find the subdivision cell D_q containing q. Set $D_l = D_q$, $r = q$.
2) Connect r and the 2D position of nv_l by a line l, in the plane defined by D_l.
3) If $nv_l \neq p$, 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).

In the Supplementary Material A submitted along with this paper, the complete 2D subdivision structure of a 3D star model is illustrated. Due to the extreme complexity of the 2D subdivision structure with curved boundaries on $\mathcal{M}$, we only store the 1D subdivision with VWs on each edge of $\mathcal{M}$ and propose in the following sections practical algorithms to compute the iso-contours, bisectors and Voronoi diagrams of multiple point sites on $\mathcal{M}$. It was shown in [40] that the 1D subdivision on edges of $\mathcal{M}$ can completely induce the correct 2D subdivision on faces of $\mathcal{M}$.

4 Structures of Iso-contour, Bisector and Voronoi Diagram on $\mathcal{M}$

Given a set of distinct point sources $P = (p_1, p_2, \cdots, p_m)$ on $\mathcal{M}$, the geodesic distance $D_P(x)$ for $x \in \mathcal{M}$ is defined as $\arg\min_i \{D_{p_i}(x), p_i \in P\}$. An iso-contour of the distance field D_P is the trace of those points on $\mathcal{M}$ that have the same value of distance. A bisector of two

[1] Multiple source points are handled in a similar way

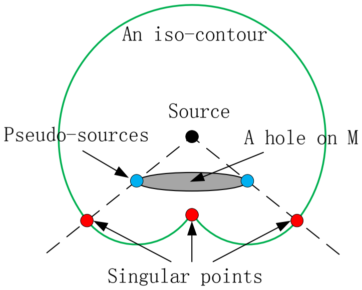


Fig. 3. The existence of singular points in an iso-contour. For closed surfaces, the hole shown in the figure can be a polygonal obstacle such as a prism with sufficient height.

points $p_i, p_j \in P$ is the trace of points q on $\mathcal{M}$ satisfying $D_{p_i}(q) = D_{p_j}(q)$. The Voronoi diagram of P on $\mathcal{M}$ is a set $VD(P) = (VC(p_1), VC(p_2), \cdots, VC(p_m))$, where $VC(p_i) = \{q | D_{p_i}(q) \leq D_{p_j}(q), q \in \mathcal{M}, i \neq j, j \in I_m\}$. In this section, we present four properties that show the inter-structures and relationships among iso-contours, bisectors and Voronoi diagrams on $\mathcal{M}$. Figure 1 illustrates an example of distance field (bottom-left), iso-contours (top-right) and Voronoi diagram with trimmed bisectors (bottom-right).

4.1 Structure of Iso-contours

Due to the exitance of pseudo-sources, the iso-contours on $\mathcal{M}$ have the following analytic structure. For a closed surface $\mathcal{M}$ without boundary, each iso-contour of the distance field on $\mathcal{M}$ consists of at least one closed curve. Each closed curve consists of circular arc segments joined at singular points.

Definition 3: The *singular points* are locations where the nearest pseudo-source is changing from one to another. The singular points can be grouped into segments: each segment is continuous on $\mathcal{M}$ and is called a *singular locus* in this paper.

The iso-contours can only be C^0-continuous at a singular locus. Definition 3 is based on the following observations. The exact geodesic path on $\mathcal{M}$ is a polyline and the only possibility of vertices in V (except for the source points) existing along a geodesic path is that they are singular vertices. Between each pair of sequential singular vertices, the path goes through a series of triangles which can be unfolded into a common plane without overlap (ref. Fig. 2b) and the geodesic path in the plane is a single straight line segment. So except for the locations of singular points, locally in each triangle an iso-contour is a circular arc. The existence of singular points is shown in Figure 3: it is readily seen that the iso-contour at the singular points can have C^0 or C^1 continuity.

Definition 4: A point $p \in \mathcal{M}$ is a *critical point* of the distance field function D, if the partial derivatives of D vanish at p. The index d of a critical point p is the number of negative eigenvalues of a Hessian matrix of D at p.

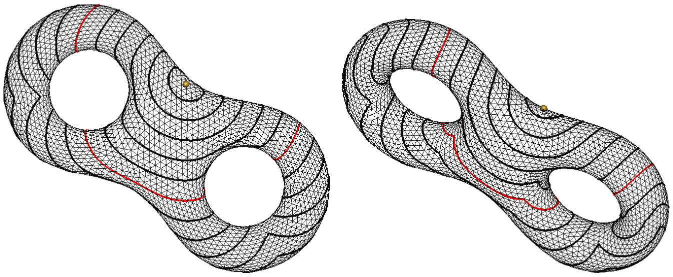


Fig. 4. The front and back views of iso-contours of a single source point on an eight model. The maximal geodesic distance on $\mathcal{M}$ is normalized to 1 and the iso-contour with value 0.5 is shown in red color which consists of three distinct closed curves.

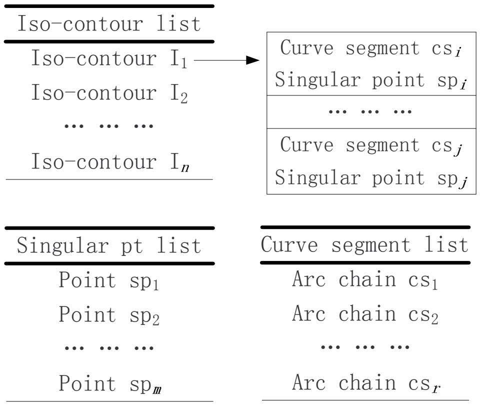


Fig. 5. The data structure of iso-contours on $\mathcal{M}$.

Property 1: The number of closed curves in an iso-contour of multiple sources on the surface of a general genus-r ($r \geq 0$) object depends on the indices of critical points of the distance field function on $\mathcal{M}$.

This property is drawn from Morse theory and algebraic topology [17]. At $d = 1$ critical point, a minimum increases and a maximum decreases the circle number of iso-contours by one. At $d = 3$ critical points, a saddle splits or merges circles in iso-contours.

A genus-2 model with ten iso-contours is shown in Figure 4. It clearly shows the tangent discontinuities at the singular points and an iso-contour that is separated into three disjoint closed segments. Based on the Definition 3 and Property 1, we propose the data structure listed in Figure 5 for iso-contours on $\mathcal{M}$.

4.2 Bisectors of Point Sites on $\mathcal{M}$

The bisector $B(p, q)$ defined by point sites p and q is the trace of points on $\mathcal{M}$ which have equal geodesic distance to p and q. The bisectors on $\mathcal{M}$ may not be one-dimensional as revealed in the following property:

Property 2: If a singular vertex of $\mathcal{M}$ lies on $B(p, q)$, then $B(p, q)$ contains a 2D region on $\mathcal{M}$.

To see Property 2, we develop the geodesic paths from p and q, respectively, onto a plane. The shaded 2D area shown in Figure 6 lies in $B(p, q)$. In this paper,

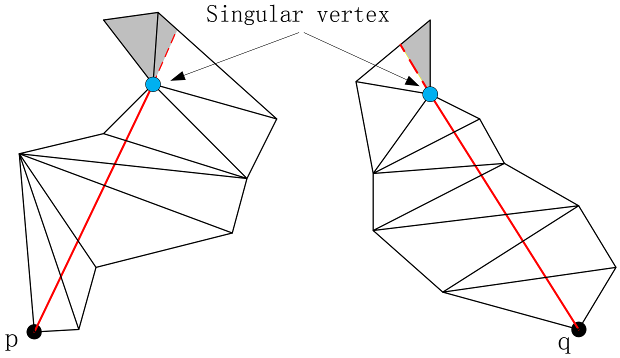


Fig. 6. Illustration of Property 2.

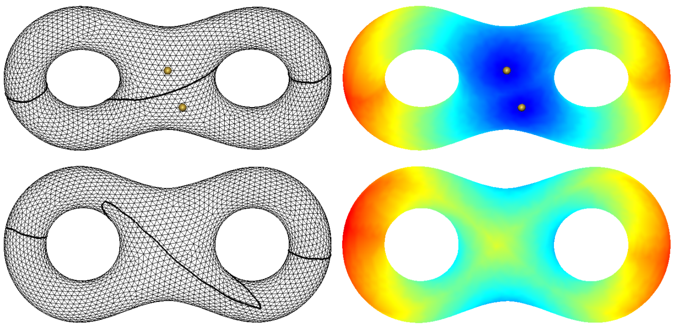


Fig. 7. The front and back views of a bisector of two point sites, with color-mapped distance field, respectively. Note that in the bottom-right figure, pseudo-bisectors exist on which the points have the same geodesic distance to an identical source point, but from different directions.

we assume that all source points are distinct from each other and no vertices of $\mathcal{M}$ have the same geodesic distance to two or more source points. So the bisectors of $\mathcal{M}$ consist of 1D curve segments. By Definition 3, the singular loci of iso-contours contain all the bisectors. In addition, singular loci also contain pseudo-bisectors on which the points have the same geodesic distance to an identical source point, but from different directions (Figure 7 shows an example). Bisectors of source points have the following structure.

Property 3: The bisector of two distinct source points on a genus-$r(r \geq 0)$ object's surface can have at most $r+1$ distinct closed curves.

Property 3 is based on the following observations. One complete bisector cuts the surface into two parts. Each part has shorter geodesic paths to one source point and thus must contain that source point. On a genus-r model, $r+2$ non-intersected closed curves cut the surface into at least three distinct parts and there are only two source points; a contradiction.

One example of a bisector on a genus-2 model consisting of three closed curves is shown in Figure 7.

Definition 5: Each distinct closed curve of a bisector can be decomposed at breakpoints. A *breakpoint* is the location at which the nearest pseudo-source is changing along the bisector from one side of a source point.

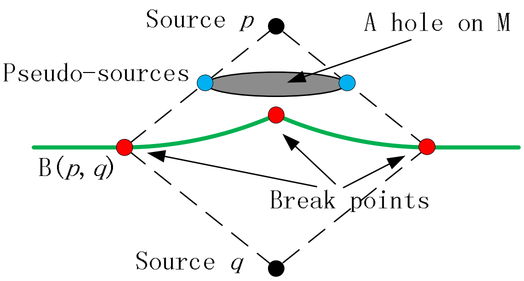


Fig. 8. Illustration of Definition 5.

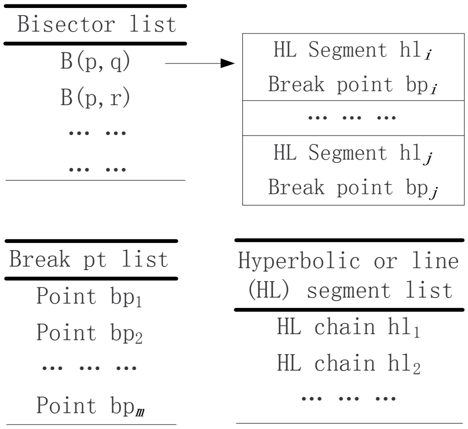


Fig. 9. The data structure of bisectors on $\mathcal{M}$.

Pseudo-sources make the bisector behavior as an additive weighted Voronoi diagram in a local 2D plane. So between breakpoints, a bisector consists of hyperbolic and line segments. The bisector is C^0 continuous at break points.

An illustration of Definition 5 is shown in Figure 8. Based on Property 3 and Definition 5, we propose the data structure listed in Figure 9 for bisectors on $\mathcal{M}$.

4.3 Voronoi Diagram of Point Sites on $\mathcal{M}$

Let $P = \{p_1, p_2, \cdots, p_m\} \subset S$ and $p_i \neq p_j$ for any $i \neq j$. The region defined by

$$VC(p_i) = \{q | D_{p_i}(q) \leq D_{p_j}(q),\ i \neq j, q \in \mathcal{M}\}$$

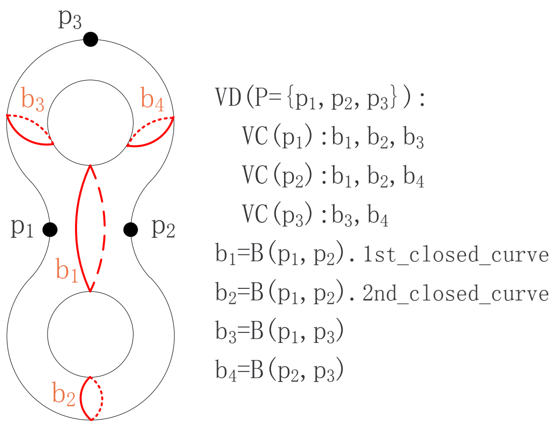
is called the Voronoi cell of p_i. For 2-manifold meshes without boundary, all Voronoi cells are bounded by bisectors, mutually exclusive or semi-exclusive, and $\bigcup_{i=1}^{n} VC(p_i) = \mathcal{M}$. The set given by

$$VD(P) = \{VC(p_1), VC(p_2), \cdots, VC(p_m)\}$$

is defined as the Voronoi diagram of point sites P on $\mathcal{M}$. Quite different from the Euclidean space cases, the Voronoi diagram on 2-manifold $\mathcal{M}$ possesses some unique properties.

Property 4: Each Voronoi cell on $\mathcal{M}$ is connected, but it may not be singly connected.

Property 4 is based on the following observation. By definition, each Voronoi cell $VC(p_i)$ must contain its point site p_i and $\forall p \in VC(p_i)$, the geodesic path between



VD(P={p1, p2, p3}):
  VC(p1):b1, b2, b3
  VC(p2):b1, b2, b4
  VC(p3):b3, b4
b1=B(p1, p2).1st_closed_curve
b2=B(p1, p2).2nd_closed_curve
b3=B(p1, p3)
b4=B(p2, p3)

Fig. 10. An illustration of a Voronoi diagram of three point sites on an eight model: each Voronoi cell has more than one closed curve on the boundaries.

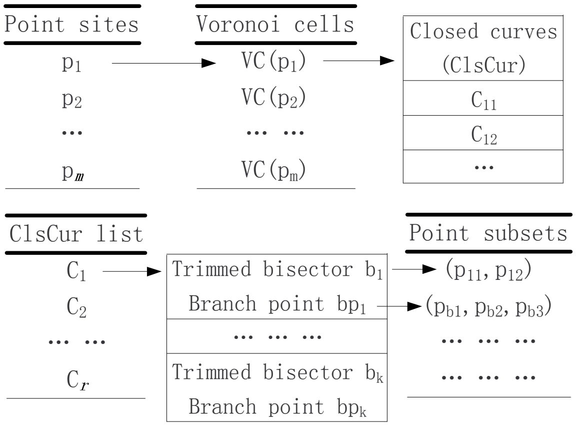


Fig. 11. The data structure of Voronoi diagrams of point sites on $\mathcal{M}$.

p and p_i must also be contained in $VC(p_i)$. So $VC(p_i)$ is connected. Since according to Property 3 the boundaries of the cell can have more than one closed curve, it could be multiple-connected: an example is shown in Figure 10. Based on Property 4, we have the following definition.

Definition 6: Each Voronoi cell in $VD(P)$ is bounded by one or more closed curves. Each closed curve consists of bisectors. The bisectors are trimmed and joined into closed curves at the branch points. A *branch point* is the location on $\mathcal{M}$ which has the same distance value to its three closest sites. The boundary of a Voronoi cell does not have to contain a branch point.

One example showing the existence of branch points is presented in Figure 1. For the model shown in Figure 10, none of the Voronoi cells have any branch points on the boundary. Given the Property 4 and Definition 6, we propose the data structure listed in Figure 11 for Voronoi diagrams of point sites on $\mathcal{M}$.

As a short summary, some of the definitions and properties presented in this section are not new: Property 1 is well explained in [17] but in a different form, Property 4 and Definition 6 have been given and studied in a much broader scope in Riemannian manifold [65], [66]. Nevertheless we present and study these definitions and properties systematically, persisting on triangulated 2-manifold $\mathcal{M}$, so they benefit use in constructing the practical algorithms proposed in the next section.

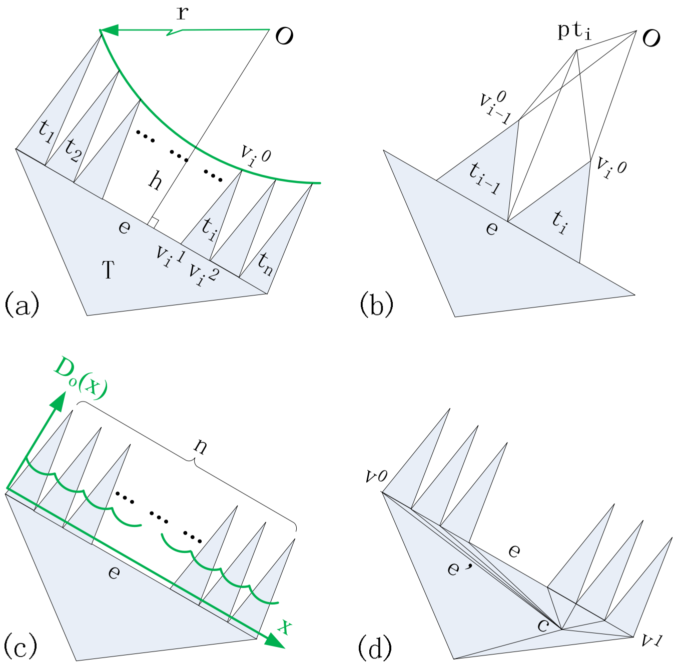


Fig. 12. A constructive example shows that the distance field values on e' can have worst-case $O(n)$ extrema.

5 Practical Computation Algorithms

Based on the properties and data structures stated in Section 4, in this section we propose efficient and practical algorithms to explicitly construct the iso-contours, bisectors and Voronoi diagrams of point sites on triangulated surface $\mathcal{M}$. Recall that we use a 8-tuple $(b_0, b_1, d_0, d_1, \tau, \sigma, Id_{nv}, Id_{pt})$ explained in Section 3 to represent a visibility wedge on an edge e of $\mathcal{M}$, where Id_{pt} is the ID of source point and Id_{nv} is the nearest pseudo-source which may or may not be identical to Id_{pt}. Throughout this section, n and m denote the number of triangles in $\mathcal{M}$ and the number of point sites in $VD(P)$, respectively.

The arrangement of triangles on $\mathcal{M}$ can exhibit various wild scenarios that lead to high-complexity iso-contours and bisectors. In the following subsections, we preprocess the triangular meshes such that pathological worst cases can be avoided in these more realistic models, which also make efficient and practical algorithms feasible.

5.1 Iso-contours

Property 5: For an edge e of $\mathcal{M}$, the distance function on e can have in the worst case $O(n)$ extrema.

This property is proved in [40] and an example is constructed in Figure 12. Assume that triangles $T, t_1, \cdots, t_n$

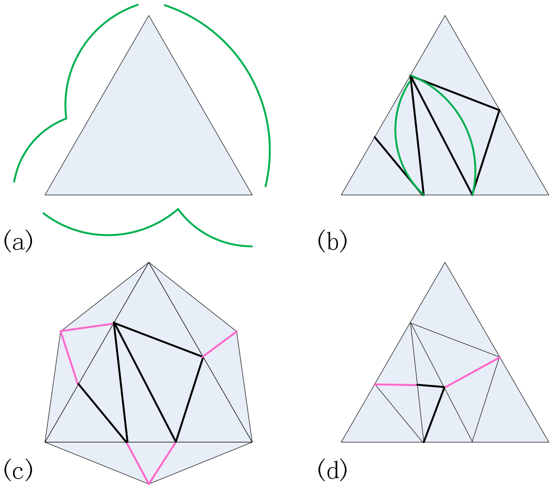


Fig. 13. Model preprocess. (a) A triangle with the distribution of distance field value shown in green for three edges. (b) Triangle is subdivided using bold black new edges. (c) The triangulation in neighbor triangles is completed using red edges. (d) New created edges with multiple extrema of distance field value is further subdivided.

lie in the same plane pl. Referring to Fig. 12a, each t_i has vertices v_i^0, v_i^1, v_i^2 and the edge e of T consists of the edges $(v_i^1, v_i^2), i = 1, 2, \cdots, n$. All the vertices v_i^0 sit on the same circle centered at O of radius r. Define the distance from O to edge e to be h, $h > r, r \to \infty$, so all the isosceles triangles t_i can be regarded congruent to each other. Referring to Fig. 12b, between each pair (t_{i-1}, t_i) of triangles in tandem, let point pt_i be out of plane pl with sufficient distance and edges $(pt_i, v_i^0), ((pt_i, v_{i-1}^0))$ be perpendicular to edges $(v_i^0, v_i^1), (v_{i-1}^0, v_{i-1}^2)$, respectively, so the geodesics from O to edge e can only go through triangles $t_1, \cdots, t_n$. It is readily seen that, as shown in Fig. 12c, the distance field function $D_O(x)$ along edge e has $O(n)$ extrema. To complete the triangulated 2-manifold setting, let triangle T be partitioned by an interior point c, c is very close to v^1 and the angle $\angle cv^0v^1$ is very close to zero (Fig. 12d). So the edge $e' = (c, v^0)$ has the same distribution of values of distance field function of e.

Preprocess. Since the distance field value on an edge e of $\mathcal{M}$ can have in the worst-case $O(n)$ extrema, we partition e into subedges such that the distance field value on each subedge is monotone and linear. For each triangle t containing partitioned edges, t is subdivided by constrained Delaunay triangulation and the geodesic visibility wedges on the new added edges in t are locally updated (Figure 13). Since the number of 2D subdivision regions (ref. Supplementary Material A) in each triangle is bounded by $O(n)$ [40], the complexity of the preprocess is $O(n^2)$ in the worst case, while in all our experiments, it runs in only linear time: an example is shown in Figure 14. Denote the number of triangles in a preprocessed mesh by n'.

After preprocessing, each triangle has three edges on which the geodesic distance is linear. Without loss of generality (subject to a shift by a scalar value), explicit construction of the iso-contour zero is considered below.

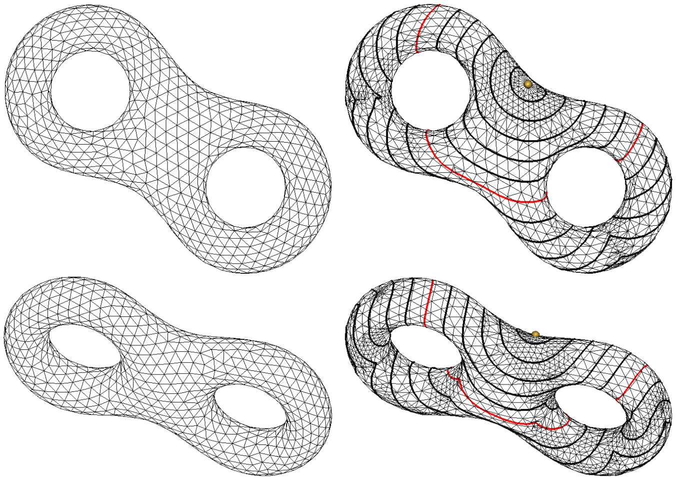


Fig. 14. Preprocess triangular meshes. Left column: a coarse mesh before preprocess. Right column: two views of the preprocessed mesh with the same iso-contours as those in Fig. 4.

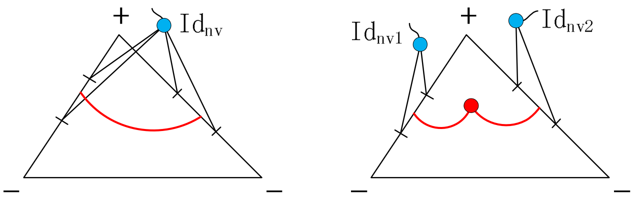

+
$\mathrm{Id}_{\mathrm{nv}}$
−
−
$\mathrm{Id}_{\mathrm{nv1}}$
+
$\mathrm{Id}_{\mathrm{nv2}}$
−
−

Fig. 15. Two situations of iso-contour in one triangle. Left: the triangle contains only one arc iso-contour. Right: Two arcs are intersected at a singular point and are made up of an iso-contour in a triangle.

Property 6: On a preprocessed mesh, the zero-value iso-contour only passes through the triangles that have opposite signs at two of its three vertices. If at two edges the two wedges, which contain zero geodesic distance value, have the same nearest pseudo-source Id_{nv}, then the triangle contains one single arc segment. Otherwise the triangle contains one singular point which is the intersection point of two arc segments.

Given Definition 3, a sketch illustrating Property 6 is shown in Figure 15. While in a carefully constructed artificial model (Fig. 12) an iso-contour can have $\Omega(n^2)$ complexity, it has only $O(n')$ complexity in a preprocessed model since each triangle can have at most one connected piece in an iso-contour.

Based on whether critical points exist or not, below we classify 2-manifold models into two classes and propose two corresponding algorithms. To determine the critical points and index number on $\mathcal{M}$, the $O(n)$ algorithm proposed by Takahashi et al. [60] is adopted. Their method scans the circular list of neighbors for each vertex v in $\mathcal{M}$ in counter-clockwise order and reduces the sequence of neighbors by computing the sums of all positive Δ_i and all negative Δ_i, respectively, where Δ_i

is the distance field value difference between neighbors nb_i and v. Finally the elements in a reduced vertex list L_c are the extracted critical points.

5.1.1 Models with Single-Piece Iso-Contour

If a triangulated surface $\mathcal{M}$ has no critical points with indices 1 or 3, by Property 1, any iso-contour on it is a single closed curve. To explicitly construct the iso-contour with any prescribed value on this kind of models, we preprocess the model as follows. Given a source point p, we compute the geodesic distance field on $\mathcal{M}$ and find the farthest point q on $\mathcal{M}$. Then we construct a geodesic path from p to q. The path goes through a set of triangles: each of them covers a distance interval and all the intervals in them are continuous in tandem. We sort these triangles into a binary tree indexed by the distance interval. To construct a particular iso-contour, we search the binary tree and find the triangle whose distance interval contains the iso-contour value.

Given a starting triangle, the algorithm runs in a marching process. Without loss of generality, we assume that the iso-contour value is zero. Given the first triangle through which the iso-contour passes, we trace the iso-contour using the edge which has the opposite signs at its two vertices. Each such edge guides the iso-contour from one triangle to another triangle. For each marched triangle, the inside iso-contour is determined by Property 6. If the iso-contour goes through a vertex v of $\mathcal{M}$, the 1-ring neighbor triangles of v is checked to find the next triangle to be marched. Our marching method is topology-oriented, since we first check the sign of vertices of each marched triangle. So our method is robust to degeneracy and is topologically consistent.

Property 7: The iso-contour construction algorithm for preprocessed models with single-piece iso-contours takes $O(\log n' + k)$ time, where k is the number of triangles through which the iso-contour passes.

To see the above property, note that the geodesic path and the iso-contour both have the $O(n')$ complexity in preprocessed models. So searching the binary tree takes $O(\log n')$ time to identify the starting triangle and the marching process takes $O(k)$ time. Finally the overall complexity is $O(\log n' + k)$.

5.1.2 Arbitrary Genus-r ($r \geq 0$) Models

The algorithm for arbitrary genus-r models is more complicated than that for models studied in Section 5.1.1. We preprocess the model as follows. We compute the geodesic distance field with a prescribed source p on $\mathcal{M}$. Each triangle in $\mathcal{M}$ covers a distance interval. We sort all triangles in $\mathcal{M}$ into an interval tree [13], [42] indexed by the distance interval of each triangle.

The genus-r iso-contour algorithm also runs in a marching process. Given a particular iso-contour value, we search the interval tree for stabbing query and sort all the triangles, whose distance interval covers the iso-contour value into a queue Que. The following algorithm reports the inquired iso-contour with the number of closed curves.

```
Algorithm 1: genus_r_isocontours($\mathcal{M}, c$)
Input. An iso-contour value $c$, a preprocessed mesh $\mathcal{M}$
        with constructed distance field and interval tree.
Output. The requested iso-contour with the number of
        closed curves.
1.    Stabbing query in the interval tree and output
      the result in a queue $Que$;
2.    Set curve number $cn = 0$;
3.    While ($Que$ is not empty)
3.1.      $cn = cn + 1$;
3.2.      Pop the first element $t$ of $Que$;
3.3.      Marching triangles with the initial triangle $t$;
3.4.      Remove all the marched triangles from $Que$.
```

Property 8: **Algorithm 1** for arbitrary genus-r models takes $O(\log n' + k \log k)$ time, where k is the number of triangles through which the iso-contour passed.

We sketch the proof of Property 8 as follows. An interval tree for a set of n' intervals reports all intervals that contain a query point in $O(\log n' + k)$ time, where k is the number of reported intervals [13], [42]. The Que can be built in $O(k)$ time and removing an element with a specified key from Que takes $O(\log k)$ time [12]. So **Algorithm 1** runs in $O(\log n' + k \log k)$ time.

Before running Algorithm 1, the distance field on $\mathcal{M}$ is constructed in $O(n^2 \log n)$ time [45] and an interval tree is constructed for all the triangles in $\mathcal{M}$ in $O(n' \log n')$ time. Since multiple point sources behave like pseudo-sources, the construction of iso-contours of multiple sources is identical to that of a single source.

5.2 Bisectors

According to Definition 3, the singular loci of a multi-source geodesic distance field contain the trimmed bisectors that contribute to the Vornoi diagram of the point source set P on $\mathcal{M}$. In this section, we explicitly construct the bisector of two source points p and q on $\mathcal{M}$.

Property 9: The bisector $B(p, q)$ goes through an edge e at the location that delimits two visibility wedges at e and the two wedges have the source point ID $Id_{pt} = p$ and $Id_{pt} = q$, respectively.

Preprocess. In the worst case, one bisector can go through a triangle as many as n times [48], [40]. Similar to iso-contour construction, the mesh model is also preprocessed to avoid this hypothetical worst-case complexity. We first examine each edge in $\mathcal{M}$ and guarantee that each edge contains at most one intersection point with the bisector: if this is not the case, the edge is subdivided. Denote the number of triangles in a preprocessed mesh by n'. Starting from p and q, we propagate the visibility wedges in a continuous-Dijkstra fashion. During propagation, we use a list to store those edges at which a visibility wedge is updated with a neighbor wedge belonging to a different source. To the end, the edge list is converted into a queue Que which contains all the triangles through which the bisector passes. Similar to

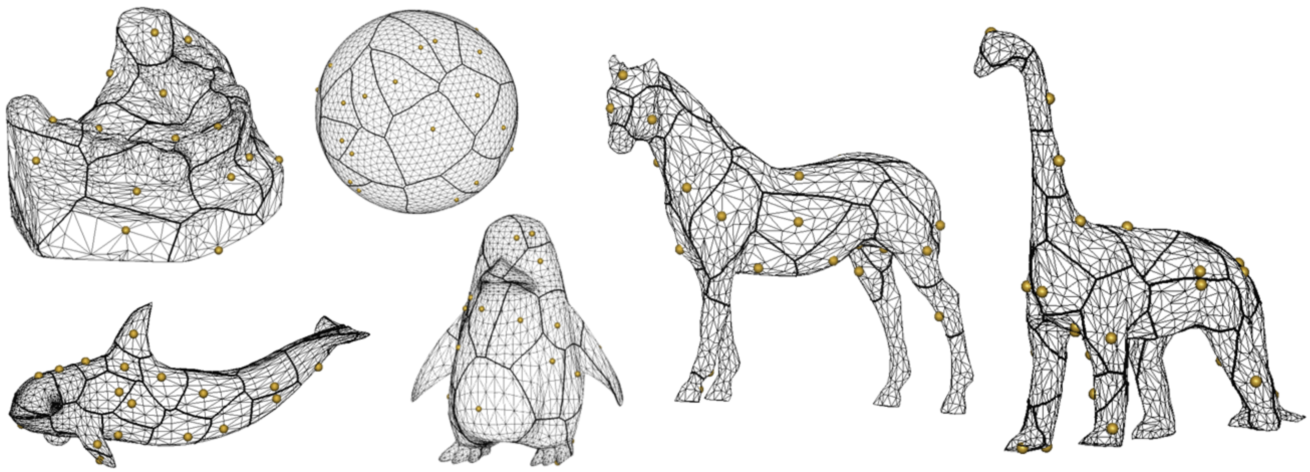


Fig. 16. The Voronoi diagrams of 30 randomly generated points on different models of genus-0. The statistical data is summarized in Table 1.

iso-contours, the following marching algorithm is used to construct the bisector on a general genus-r model.

```
Algorithm 2: genus_r_bisector($\mathcal{M}, p, q$)
Input. A preprocessed mesh $\mathcal{M}$ and two point sites $p, q$.
Output. The requested bisector with the number of
        closed curves.
1.    Build the distance field on $\mathcal{M}$ with point sites $p, q$
      and output the queue $Que$;
2.    Set curve number $bn = 0$;
3.    While ($Que$ is not empty)
3.1.      $bn = bn + 1$;
3.2.      Pop the first element $t$ of $Que$;
3.3.      Marching triangles with the initial triangle $t$;
3.4.      Remove all the marched triangles from $Que$.
```

Property 10: **Algorithm 2** runs in $O(n^2 \log n)$ time.

The proof of Property 10 is sketched as follows. It takes $O(n^2 \log n)$ time to compute the geodesic distance field [45] and report all triangles containing the bisector $B(p,q)$ with the triangle number $k < n' < n^2$. The marching process takes $O(k)$ time and the queue Que operations in $O(k \log k)$ time. So the total running time is $O(n^2 \log n)$.

5.3 Voronoi Diagrams

The preprocess step for constructing Voronoi diagrams on $\mathcal{M}$ is the same as that for the bisector construction. Also similar to the bisector case, we record a list to store those edges at which a visibility wedge is updated with a neighbor wedge belonging to a different source. After the geodesic distance field construction, the list of edges LE is converted into a list of triangles LT incident to LE. From Definition 6 and Property 9, the following property holds.

Property 11: If any triangle in LT has all its three edges contained in LE, then it contains a branch point. Each triangle in LT that does not contain a branch point is passed through by a single piece of a bisector.

Property 11 gives us a valuable means to compute the analytic structure of the Voronoi diagram on $\mathcal{M}$. We separate the list LT into two sub-lists. One is LBT whose elements contain branching points. The other is $LTT = LT \backslash LBT$. The following algorithm constructs the Voronoi diagram of point set P on $\mathcal{M}$ using the data structure as shown in Figures 9 and 11.

```
Algorithm 3: genus_r_Voronoi_diagram($\mathcal{M}, P$)
Input. A preprocessed mesh $\mathcal{M}$ and a point site set $P$.
Output. The requested Voronoi diagram using the data
        structure depicted in Figures 9 and 11.
1.    Build the distance field on $\mathcal{M}$ with the set $P$ and
      output the lists $LE$ and $LT$;
2.    Separate $LTT$ into $LBT$ and $LTT$;
3.    Create a branch-point list $BP$: each point $bp_i$ with
      sources $(s_i(1), s_i(2), s_i(3))$ corresponds to a triangle
      $t_i$ in $LBT$.
4.    For all the branch points $bp_i \in BP$
4.1.      For $m = 1$ to 3
4.1.1.        If the bisector $B(s_i(m), s_i((m+1)\%3))$ is not
              computed
4.1.1.1.          Marching $B(s_i(m), s_i((m+1)\%3))$ started
                  from $t_i$ and ended at another $t_j$ in $LBT$.
4.1.1.2.          Remove all marched triangles from $LTT$.
5.    While ($LTT$ is not empty) //some bisectors do not
                                 //have branch points by Definition 6
5.1.      Create a new entry in the bisector list;
5.2.      Pop one element $t$ in $LTT$;
5.3.      Marching triangles with the initial triangle $t$;
5.4.      Remove all the marched triangles from $LTT$.
```

Property 12: **Algorithm 3** runs in $O(n^2 \log n)$ time.

The proof of Property 12 is sketched as follows. It takes $O(n^2 \log n)$ time to compute the geodesic distance field [45] and report the set LE, LBT and LTT. Let k be the number of triangles passed by the boundaries of $VD(P)$. The two loops in Steps 4 and 5 take time $O(k \log k)$. So the total running time is $O(n^2 \log n)$.

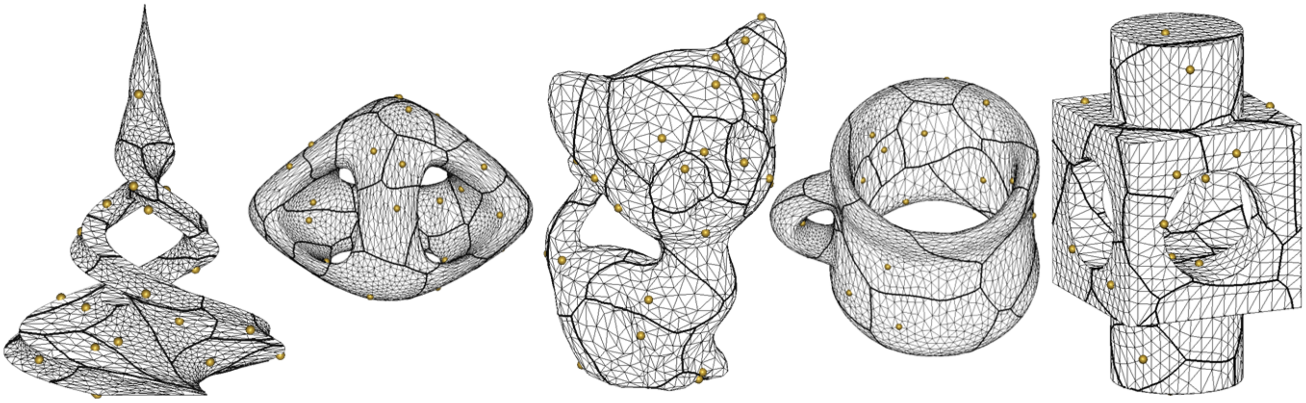


Fig. 17. The Voronoi diagrams of 30 randomly generated points on different models of genus-r, $r \geq 1$. The statistical data is summarized in Table 1.

5.4 Experiments

We test the proposed algorithms on diverse triangulated surfaces which are chosen from two classes. The first class contains shapes with simple topological types (Figure 16) and the second contains topologically complex shapes (Figure 17). All the shapes in two classes have the triangle numbers ranging from $3,000$ to $10,000$. For each shape, using the random point sampling method presented in Section 6.3, thirty points are sampled and used as the point set P to generate the Voronoi diagram $VD(P)$. The Voronoi diagrams of different numbers of samples on a genus-1 cat model, are shown in Figure 18. The performance data of output Voronoi diagrams are summarized in Tables 1 and 2. The running time is measured on a laptop with Intel Core 2 Duo CPU running at 2.13GHz.

Our first observation is drawn from the sphere model in Figure 16. Although the exact bisector of two spherical points is a great circle on an ideal sphere, triangulated spherical surfaces only provide a linear approximation: induced from Definition 5, each bisector on a triangulated surface consists of hyperbolic and line segments, and it may not be tangent continuous at break points.

Define the combinatorial complexity of the Voronoi diagram to be the total number of point sites, bisectors and branch points. If a sufficiently dense sampling P_{dense} on S is used, the $VD(P_{dense})$ will behave locally as for the Euclidean plane case in which the complexity is $\Theta(n)$ (See [52] for a detailed discussion on dense sampling and the linear complexity). If a mild sampling is used, Tables 1 and 2 empirically reveal that the $VD(P)$ complexity is linear: this can be explained by (1) each bisector can have at most $r+1$ distinct circles on a genus-r model[2]; (2) the boundary of a Voronoi cell may not contain a branch point (see Definition 6).

We measure the time complexity of the Voronoi diagram $VD(P)$ in an output-sensitive manner. The term $cplx$ in Tables 1 and 2 is defined as $1000 \times \frac{time_{sec}}{num_{ptri}}$, where $time_{sec}$ is the running time measured in second and num_{ptri} is the number of of triangles passed by the $VD(P)$ boundaries. In Table 2, exp is defined as $8 \times \frac{cplx}{\sqrt{s}}$, where s is the number of random sample points. The results show that our marching algorithm is empirically $O(num_{ptri}\sqrt{s})$ for preprocessed meshes, i.e., linear to the number of triangles passed by bisectors and increase with the number of samples with exponential rate 0.5.

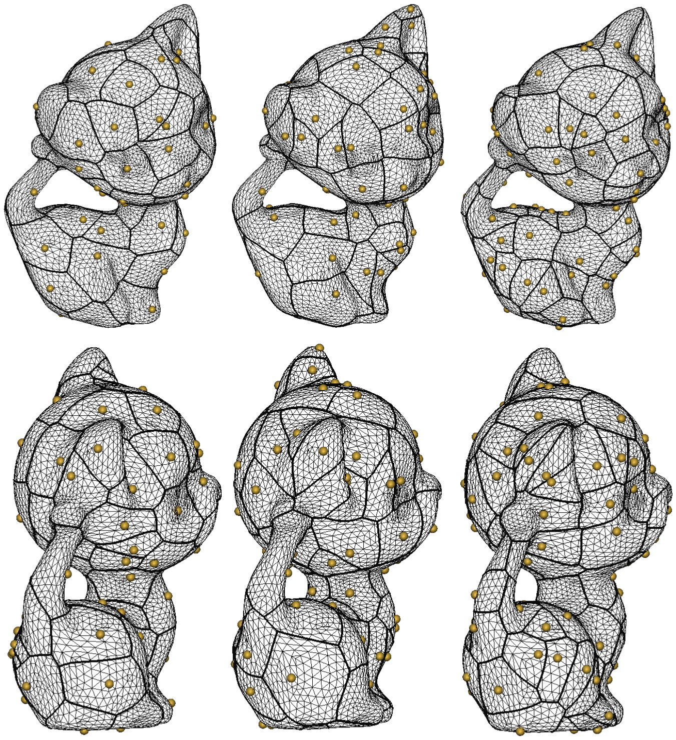


Fig. 18. The front and side views of Voronoi diagrams on the cat model with 60, 75 and 90 random samples, respectively. The statistic data is summarized in Table 2.

6 Applications

Geodesic-metric-based Voronoi diagrams reveal an intrinsic structure of point sites on triangulated surfaces

[2] When using different graphics models, r could be different and we assume that r is small and less than a fixed integer.

model	tri no	bh pt	bs no	bk pt	time(sec)	cplx
sphere	4,074	54	81	558	$11.9 + 0.55$	0.574
shark	3,412	53	81	523	$14.47 + 0.98$	0.592
tooth	5,036	50	75	648	$20.31 + 0.57$	0.604
penguin	4,416	56	84	613	$17.6 + 0.61$	0.596
dinosaur	3,433	48	74	570	$11.68 + 0.48$	0.556
horse	3,212	52	80	497	$9.89 + 0.48$	0.606
sharp	4,680	50	78	642	$20.11 + 0.54$	0.548
tri-dome	7,824	72	108	1,141	$42.25 + 1.17$	0.679
cat	10,960	58	87	1,117	$55.67 + 0.98$	0.611
cup	8,400	62	93	1,034	$52.37 + 0.95$	0.632
part	4,272	68	102	626	$14.73 + 0.70$	0.651

TABLE 1
The complexity (number of triangles, branch points, bisectors, break points and time in seconds) of the Voronoi diagrams on diverse shapes, shown in Figures 16 and 17. The time is measured in two parts, for distance field construction and Voronoi diagram construction, respectively. See Sec. 5.4 for $cplx$.

samples	branch pt	bisector no	break pt	cplx	exp
15	28	42	851	0.436	0.901
30	58	87	1,117	0.614	0.897
45	90	135	1,265	0.793	0.945
60	118	177	1,449	0.96	0.991
75	148	222	1,576	1.117	1.032
90	174	261	1,705	1.196	1.009

TABLE 2
The complexity of the Voronoi diagrams on the cat model shown in Figure 18, with different sample points. See Sec. 5.4 for exp.

$\mathcal{M}$. Below we present three applications that show the power of the Voronoi diagrams on $\mathcal{M}$ as a basic tool in pattern analysis.

6.1 Geodesic Remeshing

Nowadays 3D reconstruction from range data often produces dense triangle meshes with non-uniform triangle aspect ratio [16]. For many applications, partial differential equations need to be solved on these triangulated surfaces $\mathcal{M}$ [57]. To achieve better numerical precision, it is often required to remesh $\mathcal{M}$ into $\mathcal{M}'$ such that the triangles in $\mathcal{M}'$ are as close as possible to equilateral triangles.

To uniformly sample the surface $\mathcal{M}$, farthest point samples are used [15], [47]. Given a set of samples $P = \{p_1, p_2, \cdots, p_m\}$ on $\mathcal{M}$, we define the dispersion in P by

$$\delta(P) = \sup_{x \in S}\{\min_{p \in P} D_p(x)\}$$

where $D_p(x)$ is the geodesic distance between p and x. To find a new sample p_{m+1} that minimizes the dispersion $\delta(P \cup p_{m+1})$, the position of p_{m+1} must be at one of the branch points of $VD(P)$ or lie on the bisector which does not end at branch points. This property dramatically reduces the search space in $\mathcal{M}$. Starting from an arbitrary sample, more samples are added one by one by incrementally updating the Voronoi diagram. Leibon and Letscher [36] show that if the samples are sufficiently dense, the dual triangulation of the Voronoi diagram on $\mathcal{M}$ exists and thus offers us a solution to the geodesic remeshing problem. An example is shown in Figure 19.

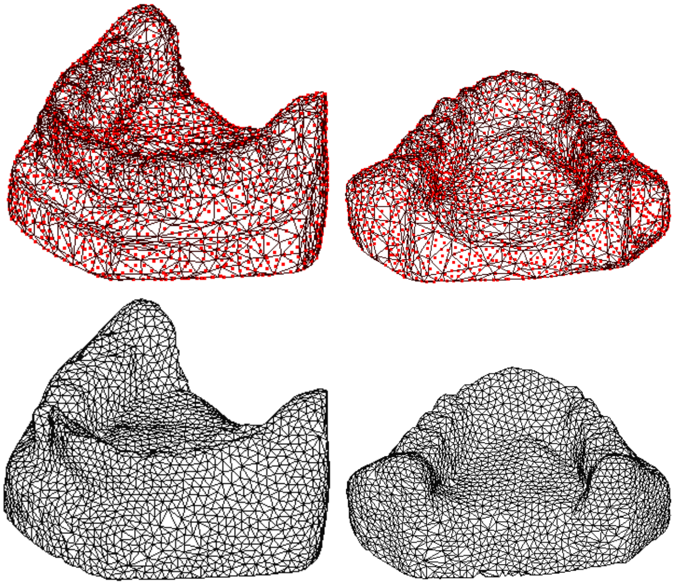


Fig. 19. Geodesic remesh of a tooth model. The first row shows two views of uniform samplings (red points) on an original mesh. The second row shows the remesh with the uniform samples.

Peyre and Cohen [54] presented an approach similar to ours, but used an approximate geodesic metric which is computed by Kimmel and Sethian's fast marching algorithm [29]. Since the original meshes can have extremely slivered triangles before remeshing, the numerical fast marching methods might be contaminated by numerical errors, while our method is more accurate and robust[3] since we use the exact geodesic metric.

6.2 Tree Skeleton Extraction and Classification

Skeletons of 3D articulated models reveal rich topological information and play an important role in pattern recognition and computer animation. Many elegant mathematical tools have been investigated for extracting skeletons from 3D models, including medial axis, shock graphs, Reeb graphs with Morse functions, etc [17], [58]. Despite the novelties in these tools, the resulting skeletons do not take the full advantage of vision perception and are not visually simple. E.g., mixed 1D and 2D cell types appear in the medial axis/surface of 3D objects and are sensitive to tiny noises on surfaces [20].

Observing that the human vision system is able to infer visually simple skeletons with full functionality,

[3] To robustly handle the degenerate cases arisen in numerical computation of our constructive algorithms, we use the toolkit proposed in [39], which classifies and handles degeneracies in two types: degeneracies on geometric intersection and degeneracies on geodesic discontinuities.

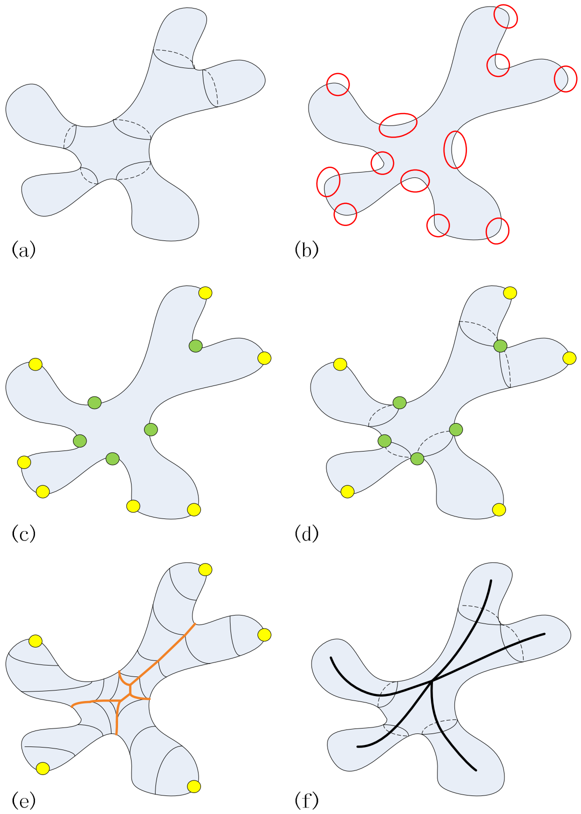


Fig. 20. The process of tree-skeleton extraction. (a) A branched 3D model; (b) Mesh saliency computation [35]: the most salient areas are circled in red; (c) Candidate critical point determination by clustering: saddle points (green) and extreme points (yellow); (d) Critical point filtering by protrusion part saliency [55]; (e) Geodesic distance field with extreme points: Voronoi diagram on surface (brown curves) and iso-contours (black curves); (f) Perceptual salient skeleton extracted from iso-contours.

	cat1	cat2	dolphin1	dolphin2	human1	human2
cat1	1.0	0.85	0.18	0.33	0.44	0.55
cat2	0.85	1.0	0.46	0.48	0.19	0.28
dolphin1	0.18	0.46	1.0	0.85	0.05	0.22
dolphin2	0.33	0.48	0.85	1.0	0.3	0.43
human1	0.44	0.19	0.05	0.3	1.0	0.97
human2	0.55	0.28	0.22	0.43	0.97	1.0

TABLE 3
The skeleton similarity of six models shown in Fig. 21.

regardless of noises or wrinkles on object surfaces [32][4], we use the following guidelines in human vision to design a computer program of skeleton extraction:

- Human perceives a global shape structure by integrating local pattern elements [8], [55];
- At the early visual process of human beings, the primate visual cortex selectively filters signals according to the spatial frequencies and orientations of local patterns [8].
- A theory called *minima rule* was examined in [23] in which human vision detects local patterns along negative minima of the principal curvatures on surfaces.

The overall algorithm is sketched in the following two steps and is illustrated in Figure 20.

- Step 1. A three-pass computation is used to simulate the filtering process in the primate visual cortex. First, the potential critical points (extrema and saddle points) are weighted by perceptual salience [35]. Only the most salient points[5] are used to identify local parts of a 3D shape. Secondly, each segmented local part is evaluated for its perceptual salience when compared to the overall shape. Thirdly, each most salient part is clustered and represented by one prototype to be used in Step 2.
- Step 2. A multi-source geodesic distance field on surface is established for all prototypes. The Reeb graph of the distance field provides the desired, perceptually salient tree skeleton. The Reeb graph can be efficiently constructed by tracing the changes in the number of closed curves in each iso-contour.

Step 2, using the iso-contour construction algorithm proposed in this paper, is illustrated in Figure 21. Our obtained 1D tree skeleton is similar to the medial scaffold in [37] and the skeletal curves in [63]. While the methods in [37], [63] extract skeletons from the general 3D point cloud, our proposed method is concentrated on triangulated 2-manifolds $\mathcal{M}$ and the obtained skeletons are perceptually simple.

Given the 1D tree skeletons, we use the graph matching method in [6] to measure the similarity between the 3D objects. We choose the method in [6] since it does not consider the topological structure of skeleton trees and is suitable in our application[6]. The skeletons of six articulated objects, two dolphins, two cats and two humans, with different poses, are extracted (Figure 21) and are used for shape similarity measures (Table 3). From the similarity values, the threshold 0.8 well classifies the objects into the three correct classes.

To test our approach in large databases, we use the McGill 3D Shape Benchmark [43]. 190 models are selected and categorized into 19 classes, each of which contains an equal number of models. Five representative matching methods are performed and compared with our approach: extended Gaussian images [25], spin images [26], D2 shape distribution [18], bending invariant signature [14], geometric moment invariants [68]. Two performance measures in [18] are used in our test: given an inquiry model in class C and a number K of top matches, *precision* is the ratio of the top K matches

[4] We emphasize that Kovacs et al.'s work [32] is mainly for 2D dynamic shape, and the related studies on 3D perception remain to be done.

[5] We use the standard deviation at top scale $\sigma = 2\varepsilon$ in the Gaussian filter in [35], where ε is 0.3% of the diagonal length of bounding box of the model.

[6] Visually similar skeletons may have different topological structures, as shown in Figure 21.

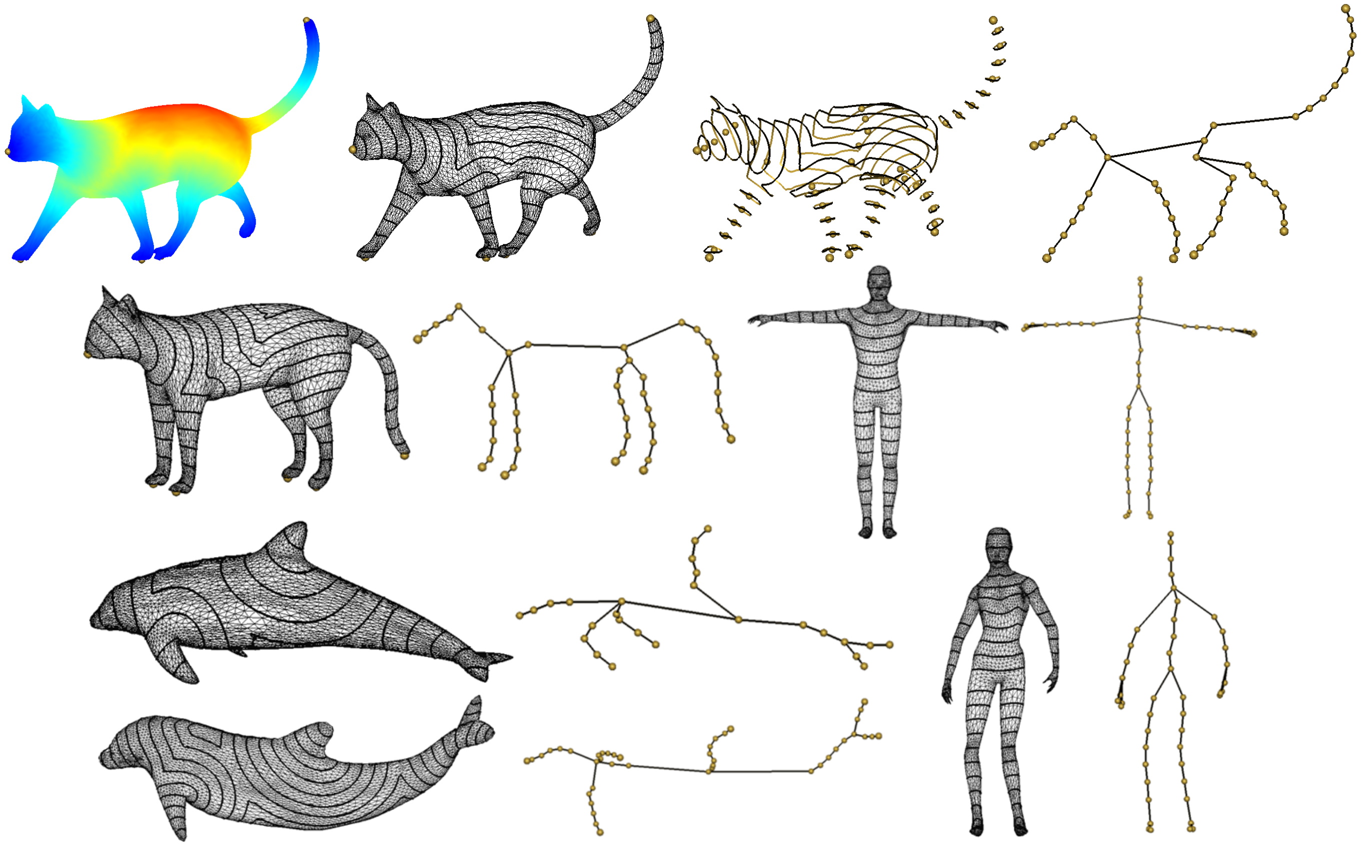


Fig. 21. Iso-contour-based tree skeleton extraction of six models: cat1, cat2, dolphin1, dolphin2, human1, human2.

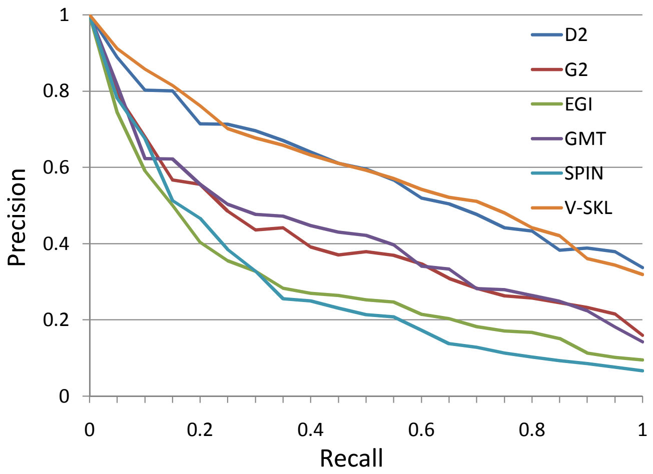


Fig. 22. Plots of precision versus recall of six approaches (D2 [18], G2 [14], EGI [25], GMT [68], SPIN [26], and the Voronoi-based skeleton match V-SKL).

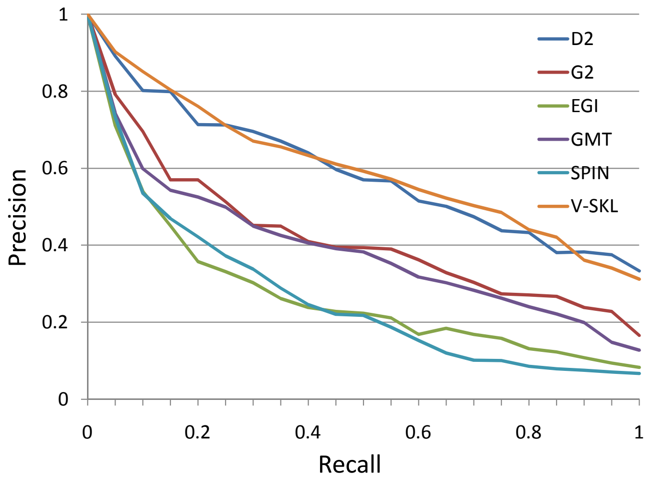


Fig. 23. Plots of precision versus recall of six approaches, testing with the noised database.

that are members of class C, and *recall* is the ratio of models in class C returned within the top K matches. The curves of precision versus recall (averaged over all models in the database) are plotted in Figure 22. Ideally a perfect matching result corresponds to a horizontal line at precision being 1 in the plot. Generally, the more area enclosed under the plot of precision versus recall, the better the matching performance is. The skeleton extraction and matching result using our approach is summarized in the Supplementary Material B. Observed from Figure 22, Our approach and D2 shape distribution [18] have better performances than other methods.

To assess the noise-insensitivity, we generate a noised version of database by disturbing each vertex along its normal direction: the magnitude of disturbance is randomly chosen between $(-L, L)$ with the zero mean, and L is 0.1 times the diagonal length of the bounding

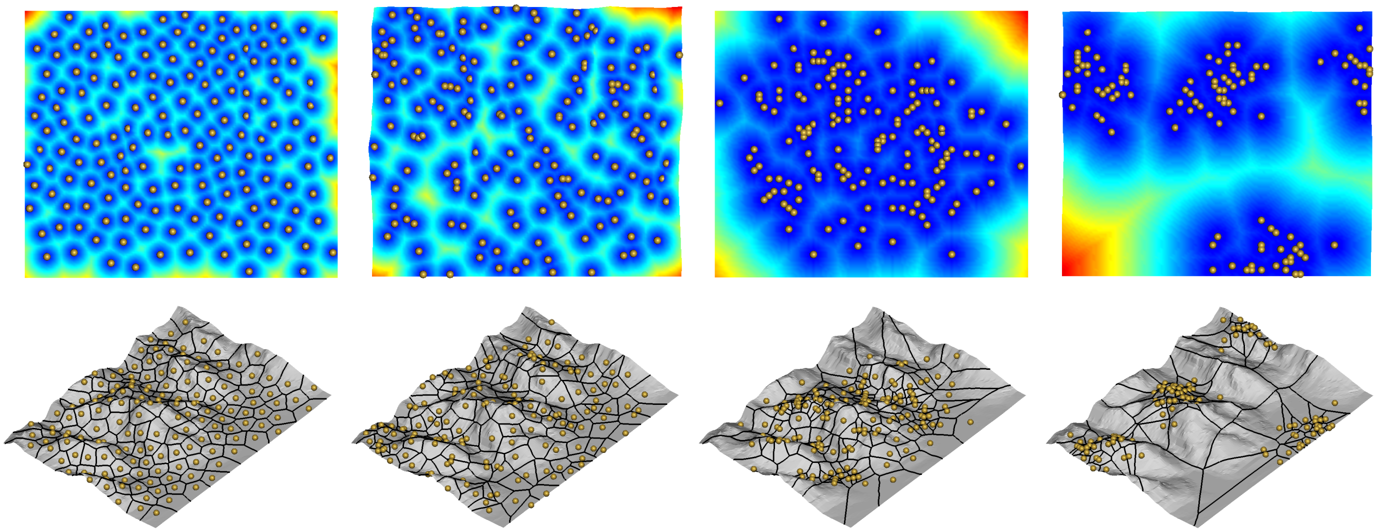


Fig. 24. Four simulated point patterns on a 2-manifold model. Top: top views of color-mapped distance field. Bottom: Voronoi diagrams of sample points. From left to right: uniform sampling, random sampling, small cluster sampling ($\mu = 12, \omega = 8$) and big cluster sampling ($\mu = 40, \omega = 20$).

box of the model. The plots of precision versus recall of the six approaches are shown in Fig. 23, from which we conclude that our voronoi-skeleton matching (V-SKEL), D2 shape distribution [18] and bending invariant signature G2 [14] are robust to noise, while geometric moment invariants GMT [68], extended Gaussian images EGI [25] and spin images [26] are more sensitive to noise. This can be interpreted by that noises heavily change the normals and areas of models, and GMT uses integral of area and EGIs and spin images use normals.

6.3 Point Pattern Analysis on $\mathcal{M}$

The Voronoi diagrams on triangulated 2-manifolds $\mathcal{M}$ can also be used to examine whether or not a pattern exists in a set of sampling points on $\mathcal{M}$. The sampling may represent the population of a state (geography), artefacts in a site (archaeology), subcellular localization in tissues (biology), etc. Using the Voronoi diagram construction algorithm proposed in this paper, the polygonal based method in [64] can be extended to the domain of 2-manifold surfaces $\mathcal{M}$.

We use the following methods to generate different point patterns on $\mathcal{M}$:

- *Random point sampling.* An array A is generated with the number of triangles in $\mathcal{M}$, i.e., $A[i]$ corresponds to the triangle t_i. Each element in A stores the triangle areas accumulated so far, i.e., $A[i] = \sum_{j=1}^{i} \Delta t_j$, where Δt_j is the area of triangle t_j. A random number generator is used to sample between 0 and $A[n]$. Each generated number x corresponds to a sample point on $\mathcal{M}$ which lies in the triangle t_k with $A[k-1] < x \le A[k]$.
- *Uniform point sampling.* The farthest point sampling method on $\mathcal{M}$ presented in Section 6.1 is used.
- *Clustering point sampling.* First the cluster origins o_i are randomly distributed. Secondly a number of points are generated for each cluster i from a random distribution with mean μ. Thirdly the points in a cluster i are distributed according to a Gaussian function centered at o_i and with standard deviation ω.

measure	uniform pattern	random pattern	small cluster pattern	big cluster pattern
ARF	0.7614	0.6926	0.6542	0.6311
RFH	0.5512	0.5106	0.5404	0.5371
AD	0.6835	0.4573	0.1244	−1.2184

TABLE 4
The mean of three measures in ten simulations for the four different point patterns in Figure 24.

measure	uniform pattern	random pattern	small cluster pattern	big cluster pattern
ARF	0.0055	0.007	0.0122	0.0172
RFH	0.0341	0.0412	0.0526	0.0547
AD	0.0465	0.0451	0.182	0.3757

TABLE 5
The standard deviation of three measures in ten simulations for the four different point patterns.

Four patterns (one random, one uniform, two cluster distributions with different μ, ω) are generated in a 2-manifold model and shown in Figure 24. We generate the Voronoi diagrams for different point samples. For each Voronoi cell $VC(p_i)$, denote its area by $A(i)$ and its perimeter by $L(i)$. Three measures are defined below (ARF and RFH are adopted from [64]) to test the pattern in the sampling:

$$ARF = \frac{1}{n}\sum_{i=1}^{n} RF(i), \quad RF(i) = \frac{4\pi A(i)}{L^2(i)}$$

$$RFH = 1 - \frac{\sigma_{RF}}{RF_{av}}$$
$$AD = 1 - \frac{\sigma_A}{A_{av}}$$

where σ_A is the area standard deviation, A_{av} is the mean area and σ_{RF} is the standard deviation of $RF(i)$. The performance data of the three measures on the patterns shown in Figure 24 is collected. To test the stability of the measured values, we run ten simulations with the four different patterns. The mean and the standard deviation of the measured values are listed in Tables 4 and 5, respectively. We also test the three measures ARF, RFH and AD in a large U.S. geological survey (USGS) database [62]. Ten geographic models, on each of which ten simulations are run with the four different patterns, are selected from [62]. These testing models and performance data (the mean and standard deviations of three measures) are summarized in the Supplementary Material C. The results are similar to the ones in Tables 4 and 5. From these results, it is observed that AD is the most significant measure to discriminate between the four patterns (ref. Table 4) and it is also very stable with small deviation (compared to the mean values) as observed in Table 5.

7 Conclusion

Analytical structures of the Voronoi diagram and practical algorithms to compute them are often desired in diverse pattern analysis applications. In this paper, we systematically study some important properties of isocontours, bisectors and Voronoi diagrams on triangulated 2-manifold surfaces $\mathcal{M}$. Based on these properties, a concise data structure is established to facilitate the explicit description of the Voronoi diagram and practical algorithms are proposed to efficiently construct the isocontours, bisectors and Voronoi diagrams of a set of point sites on $\mathcal{M}$.

Our proposed algorithms are based on the exact geodesic metric on $\mathcal{M}$ and thus, compared to previous work [30], [56], [57], are insensitive to triangle shape and triangle density in $\mathcal{M}$. Experiments and three selected applications are presented to demonstrate the effectiveness and novelty of the Voronoi diagram on $\mathcal{M}$ as a basic tool in pattern analysis. In future work, more applications of Voronoi diagrams on $\mathcal{M}$ should be explored, including the study of spatial-temporal processes of Voronoi diagrams on a time-varying 2-manifold $\mathcal{M}(t)$ and the locational optimization of observation points on $\mathcal{M}(t)$.

Acknowledgments

The authors thank the reviewers for their constructive comments that help improve this paper, and thank Mr. Lv Lu for providing the comparison results shown in Figs. 22 and 23. The authors also highly appreciate McGill and USGS for making their data publicly available. The first author was supported by the Natural Science Foundation of China (Project Number 60970099) and the National Basic Research Program of China (Project Number 2011CB302202), and the third author was supported by the Hong Kong RGC General Research Fund ENG.620409.

References

[1] P. Agarwal and K. Varadarajan, "Approximating Shortest Paths on An Noncovex Polyhedron," IEEE Symposium on Foundations of Computer Science, pp.182-191, 1997.

[2] A.D. Aleksandrov, V.A. Zalgaller, *Intrinsic Geometry of Surfaces*, AMS Publisher, 1967.

[3] B. Aronov, M. de Berg, and S. Thite, "The Complexity of Bisectors and Voronoi Diagrams on Realistic Terrains," Proc. 16th Annual European Symposium on Algorithms, pp. 100-111, 2008.

[4] J.M. Augenbaum and C.S. Peskin, "On the Construction of Voronoi Mesh on A Sphere," Journal of Computational Physics, vol. 59, pp. 177-192, 1985.

[5] F. Aurenhammer, "Voronoi Diagrams — A Survey of a Fundamental Geometric Data Structure," ACM Computing Surveys, vol. 23, no. 3, pp. 345-405, 1991.

[6] X. Bai and L.J. Latecki, "Path Similarity Skeleton Graph Matching," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 30, no. 7, pp. 1-11, 2008.

[7] M. Balasubramanian, J.R. Polimeni, E.L. Schwartz, "Exact geodesics and shortest paths on polyhedral surfaces," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 31, no. 6, pp. 1006-1015, 2009.

[8] I. Biederman, "Recognition-by-Components: A Theory of Human Image Understanding," Psychological Review, vol. 94, no. 2, pp. 115-147, 1987.

[9] J. Canny and J. Reif, "New Lower Bound Techniques for Robot Motion Planning Problems," IEEE Conf. Foundations of Computer Science, pp. 39-48, 1987.

[10] S. Cabello, M. Fort, and J.A. Sellares, "Higher-Order Voronoi Diagrams on Triangulated Surfaces," Information Processing Letters, vol. 109, pp. 440-445, 2009.

[11] J. Chen and Y. Han, "Shortest Paths on A Polyhedron: Part I: Computing Shortest Paths" Int. J. Comp. Geom. & Appl. vol. 6, no. 2, pp. 127-144, 1996.

[12] T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, *Introduction to Algorithms*, second ed. MIT Press, 2002.

[13] H. Edelsbrunner, "Dynamic Data Structures for Orthogonal Intersection Queries," Technical Report, no. F59, Tech. Univ. Graz., Austria, 1980.

[14] A.E. Elbaz and R. Kimmel, "On bending invariant signatures for surfaces," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 25, no. 10, pp. 1285-1295, 2003.

[15] Y. Eldar, M. Lindenbaum, M. Porat, and Y.Y. Zeevi, "The Farthest Point Strategy for Progressive Image Sampling," IEEE Trans. Image Processing, vol. 6, no. 9, pp. 1305-1315, 1997.

[16] O. Faugeras and R. Keriven, "Complete Dense Stereovision Using Level Set Methods," Proc. Fifth European Conf. Computer Vision, pp. 379-393, 1998.

[17] A.T. Fomenko and T.L. Kunii, *Topological Modeling for Visualization*, Springer, 1997.

[18] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, D. Jacobs, "A Search Engine for 3D Models," ACM Transactions on Graphics. 22(1): 83-105, 2003.

[19] Y. Ge and J. Fitzpatrick, "On the Generation of Skeletons from Discrete Euclidean Distance Maps," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 18, pp. 1055-1066, 1996.

[20] P. Giblin and B.B. Kimia, "A Formal Classification of 3D Medial Axis Points and Their Local Geometry," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 26, no. 2, pp. 238-251, Feb. 2004.

[21] S. Har-Peled, "Constructing Approximate Shortest Path Maps in Three Dimensions," SIAM Journal on Computing, vol. 28, no. 4, pp. 1182-1197, 1999.

[22] M. Hilaga, Y. Shinagawa, T. Kohmura, T.L. Kunii, "Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes," ACM Siggraph'01, pp. 203-212, 2001.

[23] D. Hoffman and M. Singh, "Salience of Visual Parts," Cognition, vol. 63, pp. 29-78, 1997.

[24] H. Hopf and W. Rinow, ”Über den Begriff der vollständigen differntial-geometrischen Flächen,” Comment. Math. Helv., vol. 3, pp. 209-225, 1931.

[25] B. Horn, ”Extended Guassian Images,” Proc. of the IEEE, 12: 1671-1686, 1984.

[26] A.E. Johnson, M. Hebert, ”Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 21, no. 5, pp. 433-4448, 1999.

[27] B. Kaneva and J. O’Rourke, ”An Implementation of Chen & Han’s Shortest Paths Algorithm,” Proc. 12th Canadian Conf. on Comput. Geom., pp. 139-146, 2000.

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

[29] R. Kimmel and J.A. Sethian, ”Computing Geodesic Paths on Manifolds,” Proc. of National Academy of Science, vol. 95, no. 15, pp. 8431-8435, 1998.

[30] R. Kimmel and J.A. Sethian, ”Fast Voronoi Diagrams and Offsets on Triangulated Surfaces,” Proc. of AFA Conf. on Curves and Surfaces, pp. 193-202, 1999.

[31] V. Kolmogorov and R. Zabih, ”Multi-Camera Scene Reconstruction via Graph Cuts,” Proc. Seventh European Conf. Computer Vision, pp. 82-96, 2002.

[32] I. Kovacs, A. Feher, B. Julesz, ”Medial-Point Description of Shape: a representation for action coding and its psychophysical correlates,” Vision Research, vol. 38, no. 15-16, pp. 2323-2333, 1998.

[33] R. Kunze, F. Wolter and T. Rausch, ”Geodesic Voronoi Diagrams on Parametric Surfaces,” Proc. of Computer Graphics International, pp. 230-237, 1997.

[34] D.T. Lee, ”Medial Axis Transformation of A Planar Shape,” IEEE Trans. Pattern Analysis and Machine Intellegence, vol. 4, no. 4, pp. 363-369, 1982.

[35] C.H. Lee, A. Varshney and D.W. Jacobs, ”Mesh Saliency,” Proc. ACM SIGGRAPH’05, pp. 659-666, 2005.

[36] G. Leibon and D. Letscher, ”Delaunay Triangulations and Voronoi Diagrams for Riemannian Manifolds,” Proc. 16th Annual Symposium on Computational Geometry, pp. 341-349, 2000.

[37] F. Leymarie, B. Kimia, ”The Medial Scaffold of 3D Unorganized Point clouds,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 29, no. 2, pp. 313-330, Feb. 2007.

[38] Y. Liu, W. Wang, B. Levy, F. Sun, D.M. Yan, L. Lu, and C.L. Yang, ”On Centroidal Voronoi Tessellation – Energy Smoothness and Fast Computation,” ACM Transactions on Graphics. 28(4), Article No. 101, 2009.

[39] Y.J. Liu, Q. Zhou, and S.M. Hu, ”Handling degenerate cases in exact geodesic computation on triangle meshes,” The Visual Computer, 23(9-11): 661-668, 2007.

[40] Y.J. Liu, W.Q. Zhang, and K. Tang, ”Some Properties of Exact Geodesics on Triangular Mesh Surfaces,” Technical Report, no. TR-090620, Tsinghua University, 2009, http://cg.cs.tsinghua.edu.cn/research_archive.htm.

[41] C.R. Maurer, R. Qi and V. Raghavan, ”A Linear Time Algorithm for Computing Exact Euclidean Distance Transforms of Binary Images in Arbitrary Dimensions,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 25, no. 2, pp. 265-270, Feb. 2003.

[42] E.M. McCreight, ”Efficient Algorithms for Enumerating Intersecting Intervals and Rectangles,” Technical Report, no. CSL-80-9, Xerox Palo Alto Research Center, USA, 1980.

[43] McGill 3D Shape Benchmark. http://www.cim.mcgill.ca/∼shape/benchMark/.

[44] J. Mitchell, ”Geometric Shortest Paths and Network Optimization,” in JR. Sack, J. Urrutia, (eds.) Handbook of Computational Geometry, Elsevier Science, pp. 633-702, 2000.

[45] J. Mitchell, D.M. Mount, and C.H. Papadimitriou, ”The Discrete Geodesic Problem,” SIAM Journal of Computing, vol. 16, no. 4, pp. 647-668, 1987.

[46] M. Miyazawa, P. Zeng, N. Iso, and T. Hirata, ”A Systolic Algorithm for Euclidean Distance Rransform,” IEEE Trans. Pattern Analysis and Machine Intellegence, vol. 28, no. 7, pp. 1127-1134, 2006.

[47] C. Moenning and N.A. Dodgson, ”Fast Marching Farthest Point Sampling,” Technical Report, no. 562, University of Cambridge, 2003.

[48] E. Moet, M. van Kreveld, and A.F. van der Stappen, ”On Realistic Terrains,” Proc. 22nd Annu. ACM Sympos. Comput. Geom., pp.177-186, 2006.

[49] D.M. Mount, ”Voronoi Diagrams on the Surface of A Polyhedron,” Technical Report, no. 1496, University of Maryland, 1985.

[50] H.S. Na, C.N. Lee, and O. Cheong, ”Voronoi Diagrams on the Sphere,” Computational Geometry: Theory and Applications, vol. 23, no. 2, pp. 183-194, 2002.

[51] A. Okabe, B. Boots, K. Sugihara, and S.N. Chiu, *Spatial Tessellations: Concept and Applications of Voronoi Diagrams*, second ed. Chichester : Wiley, 2000.

[52] K. Onishi and J. Itoh, ”Estimation of the necessary number of points in Riemannian Voronoi diagram,” Proc. 15th Canadian Conf. Comput. Geom., pp. 19-24, 2003.

[53] K. Onishi, and N. Takayama, ”Construction of Voronoi Diagram on the Upper Half-Plane,” IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences, vol. E79-A, no. 4, pp. 533-539, 1996.

[54] G. Peyre and L.D. Cohen, ”Geodesic Remeshing Using Front Propagation,” International Journal of Computer Vision, vol. 69, no. 1, pp. 145-156, 2006.

[55] J. Saarinen, D.M. Levi and B. Shen, ”Integration of Local Pattern Elements into A Global Shape in Human Vision,” Proc. of National Academy of Science, vol. 94, no. 15, pp. 8267-8271, 1997.

[56] J.A. Sethian, ”A Fast Marching Level Set Method for Monotonically Advancing Fronts,” Proc. of National Academy of Science, vol. 93, no. 4, pp. 1591-1595, 1996.

[57] J.A. Sethian, *Level Set Methods and Fast Marching Methods*, second ed. Cambridge University Press, 1999.

[58] K. Siddiqi and S. Pizer, *Medial Representations: Mathematics, Algorithms and Applications*, Springer, 2008.

[59] V. Surazhsky, T. Surazhsky, D. Kirsanov, S. Gortler, and H. Hoppe, ”Fast Exact and Approximate Geodesics on Meshes,” Proc. ACM SIGGRAPH’05, pp. 553-560, 2005.

[60] S. Takahashi, T. Ikeda, Y. Shinagawa, T.L. Kunii and M. Ueda, ”Algorithms for Extracting Correct Critical Points and Constructing Topological Graphs from Discrete Geographical Elevation Data,” Proc. Eurogrpahics’95, pp. 181-192, 1995.

[61] H. Tek, B. Kimia, ”Boundary smoothing via symmetry transforms,” Journal of Mathematical Imaging and Vision, 14(3): 211-223, 2001.

[62] U.S. geological survey (USGS) Geographic Data. http://edc2.usgs.gov/geodata/index.php.

[63] A. Verroust, F. Lazarus, ”Extracting Skeletal Curves from 3D Scattered Data,” The Visual Computer, 16(1): 15-25, 2000.

[64] F. Wallet and C. Dussert, ”Multifactorial Comparative Study of Spatial Point Pattern Analysis,” Journal of Theoretical Biology, vol. 187, no. 3, pp. 437-447, 1997.

[65] F.E. Wolter, *Cut Loci in Bordered and Unbordered Riemannian Manifolds*, PhD thesis, Technical University of Berlin, 1985.

[66] F.E. Wolter, ”Cut locus and medial axis in global shape interrogation and representation,” Technical Report 92-2, MIT, 1992.

[67] F.E. Wolter, K.L. Friese, ”Local and global geometric methods for analysis interrogation, reconstruction, modification and design of shape,” Proc. Computer Grapics International 2000, pp.137-151.

[68] D. Xu, H. Li, ”Geometric Moment Invariants,” Pattern Recognition, vol. 41, no. 1, pp. 240-249, 2008.