

---

# GEODESIC CENTROIDAL VORONOI TESSELLATIONS: THEORIES, ALGORITHMS AND APPLICATIONS

---

A PREPRINT

**Zipeng Ye, Ran Yi, Minjing Yu, Yong-Jin Liu\***  
 Department of Computer Science and Technology,  
 Tsinghua University, China

**Ying He**  
 School of Computer Engineering,  
 Nanyang Technological University, Singapore

July 2, 2019

## ABSTRACT

Nowadays, big data of digital media (including images, videos and 3D graphical models) are frequently modeled as low-dimensional manifold meshes embedded in a high-dimensional feature space. In this paper, we summarized our recent work on geodesic centroidal Voronoi tessellations (GCVTs), which are intrinsic geometric structures on manifold meshes. We show that GCVT can find a widely range of interesting applications in computer vision and graphics, due to the efficiency of search, location and indexing inherent in these intrinsic geometric structures. Then we present the challenging issues of how to build the combinatorial structures of GCVTs and establish their time and space complexities, including both theoretical and algorithmic results.

**Keywords** Voronoi tessellation · Geodesic · Computational Geometry

## 1 Introduction

Let  $(X, d)$  be a metric space, where  $X$  is a point set and  $d : X \times X \rightarrow \mathbb{R}$  is a metric. Given an open subset  $\Omega \subseteq X$ , a set  $\{V_i\}_{i=1}^k$  is called a *tessellation* of  $\Omega$  if  $V_i \cap V_j = \emptyset$  for  $i \neq j$  and  $\cup_{i=1}^k \overline{V_i} = \overline{\Omega}$ , where  $\overline{A}$  is the closure of  $A$ . Given a set of points  $\{g_i\}_{i=1}^k$  in  $\Omega$ , the *Voronoi cell* corresponding to the point  $g_i$  is defined as

$$\hat{V}_i = \{x \in \Omega \mid d(x, g_i) < d(x, g_j) \text{ for } j = 1, \dots, k, j \neq i\}. \quad (1)$$

Elements of  $\{g_i\}_{i=1}^k$  are called *generators*.

Since 1644 (Part III of *Principia Philosophiae*, written by Descartes), Voronoi tessellations had been well studied in the Euclidean space  $\mathbb{R}^n$ ,  $n \in \mathbb{Z}^+$  [1], in which Voronoi tessellations (as well as their dual structures, well known as Delaunay triangulations) had played a central role as fundamental geometric structures. Voronoi tessellations had also been studied in spaces with non-Euclidean metrics, including spheres [2], hyperbolic spaces [3] and the general Riemannian manifolds [4]. Recently, due to the flourishes of big media data from digital sampling, more and more data are appearing in the form of manifold meshes [5]. Quite different from parametrized 2-manifolds or general Riemannian manifolds that are generally  $C^\infty$  smooth (or at least  $C^2$  smooth) [6], manifold meshes are only  $C^0$ . In our study, we adopt the discrete geodesic metric [7].

In this paper, we summarize our recent work on geodesic centroidal Voronoi tessellations (GCVT) — which are provable uniform tessellations on manifold meshes — and we show that they can be used to generate uniform remeshing in computer graphics and build content-sensitive superpixels/supervoxels for images and video in computer vision applications.

Before we introduce GCVTs, we present two close concepts of CVT and RCVT in related work.

---

\*Corresponding author

## 2 Related Work

### 2.1 Centroidal Voronoi Tessellations (CVT)

CVT had been well studied in science and engineering, with a wide range of applications including data compression in digital image processing, optimal quadrature in numerical methods, quantization and clustering in machine learning, finite difference methods in solid mechanics and fluid dynamics, distribution of resources in operational research, cellular patterns in biology, and the territorial behavior of animals; see [8] for an excellent survey.

Let  $V$  be a finite region in  $\mathbb{R}^n$ . The mass centroid  $m$  of  $V$  is defined by

$$m \triangleq \frac{\int_{x \in V} x \rho(x) dx}{\int_{x \in V} \rho(x) dx}, \quad (2)$$

where  $\rho$  is a density function defined in  $V$ . Given  $k$  points  $\{g_i\}_{i=1}^k$  in a domain  $\Omega \subseteq \mathbb{R}^n$ , we can define the Voronoi region  $\hat{V}_i$  corresponding to each  $g_i$  based on the Euclidean metric  $d_E$  and the Voronoi tessellation  $\{\hat{V}_i\}_{i=1}^k$  of  $\Omega$ . Let  $m_i$  be the mass centroid of each Voronoi region  $\hat{V}_i$ . A Voronoi tessellation is called CVT if all the generators are mass centroids, i.e.

$$g_i = m_i, \quad i = 1, \dots, k. \quad (3)$$

Arbitrarily chosen generators are usually not the mass centroids of their associated Voronoi regions so an arbitrary Voronoi tessellation cannot be a CVT. It can be shown that CVT minimizes the following energy (a.k.a. CVT energy functional):

$$\varepsilon_E(\{(p_i, V_i)\}_{i=1}^k) = \sum_{i=1}^k \int_{x \in V_i} \rho(x) d_E^2(x, p_i) dx, \quad (4)$$

where  $\rho$  is a density function defined in  $\Omega$ ,  $\{V_i\}_{i=1}^k$  is an arbitrary tessellation and  $\{p_i\}_{i=1}^k$  is any set of  $k$  points in  $\Omega$ . To compute CVT, various local methods had been surveyed in [8]. In particular, the Lloyd method [9] is a simple yet effective local method that iteratively computes mass centroids and Voronoi tessellation.

### 2.2 Constrained CVT for Surfaces

To extend the domain partitioning of CVT from Euclidean space to general spaces, Du et al. [10] proposed the constrained CVT (CCVT) that works on a compact and continuous surface  $S \subset \mathbb{R}^N$  defined by

$$S \triangleq \{x \in \mathbb{R}^N : g_0(x) = 0 \text{ and } g_i(x) \leq 0, \text{ for } i = 1, 2, \dots, m\} \quad (5)$$

where  $\{g_i\}_{i=0}^m$  are some continuous functions.

Given a set of points  $\{g_i\}_{i=1}^k \in S$ , the constrained Voronoi region  $\hat{V}_i$  corresponding to each  $g_i$  based on the Euclidean metric  $d_E$  and restricted in  $S$  is defined as

$$\hat{V}_i \triangleq \{x \in S \mid d_E(x, g_i) < d_E(x, g_j) \text{ for } j = 1, \dots, k, j \neq i\}. \quad (6)$$

The constrained mass centroid  $m_i$  of  $\hat{V}_i$  on  $S$  is defined to be the solution of the following problem:

$$\min_{x \in S} \varepsilon_i(z), \quad \text{where } \varepsilon_i(z) = \int_{x \in \hat{V}_i} \rho(y) d_E^2(x, z) dy \quad (7)$$

A tessellation on  $S$  is called CCVT if and only if the set  $\{g_i\}_{i=1}^k \in S$  are both the generators and constrained mass centroids of the tessellation  $\{\hat{V}_i\}_{i=1}^k$ .

The applications of CCVT including polynomial interpolation and numerical integration on the sphere are illustrated in [10].

## 3 Geodesic Centroidal Voronoi Tessellations (GCVT)

Given a  $k$ -dimensional compact differentiable manifold  $M$ , a set  $\{V_i\}_{i=1}^k$  is called a *tessellation* of  $M$  if  $V_i \cap V_j = \emptyset$  for  $i \neq j$  and  $\cup_{i=1}^k \bar{V}_i = \bar{M}$ . Denote by  $d_g(a, b)$  the geodesic distance between  $a$  and  $b$  in  $M$ . Given a set of points  $\{g_i\}_{i=1}^k$  in  $M$ , the *geodesic Voronoi region* corresponding to the point  $g_i$  is defined as

$$\hat{V}_i = \{x \in M \mid d_g(x, g_i) < d_g(x, g_j) \text{ for } j = 1, \dots, k, j \neq i\}. \quad (8)$$

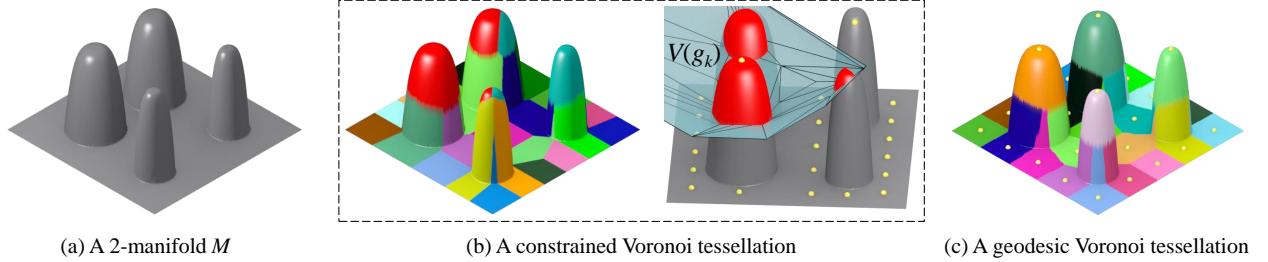


Figure 1: (b) shows a constrained Voronoi tessellation of 30 point generators on the 2-manifold (a) and its Voronoi cell  $V(g_k)$  is disconnected; i.e., it consists of three disjoint components (shown in red areas). Given the same set of point generators, each cell in geodesic Voronoi tessellation (c) is guaranteed to be connected.

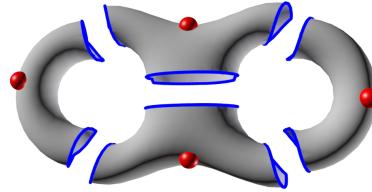


Figure 2: A geodesic Voronoi tessellation on a genus-2 manifold whose cells are multiply connected.

Elements of  $\{g_i\}_{i=1}^k$  are called *generators*. The set of geodesic Voronoi regions  $\{\hat{V}_i\}_{i=1}^k$  is called geodesic Voronoi tessellation of  $M$ . A geodesic Voronoi region is a connected domain and is a non-empty compact set [11]. For each geodesic Voronoi region  $\hat{V}_i$ , the *nominal mass centroid*  $m_i$  of  $\hat{V}_i$  on  $M$  is defined to be the solution of the following problem

$$\min_{z \in M} \varepsilon_i(z), \text{ where } \varepsilon_i(z) = \int_{x \in \hat{V}_i} \rho(x) d_g(x, z) dx. \quad (9)$$

It can be shown that  $\varepsilon_i(z)$  is continuous and the domain  $M$  is compact so  $\varepsilon_i(z)$  has at least one global minimum in  $M$ . Therefore, the nominal mass centroid exists.

Given  $k$  points  $\{g_i\}_{i=1}^k$ , we can define the geodesic Voronoi region  $\hat{V}_i$  corresponding to each  $g_i$  and geodesic Voronoi tessellation  $\{\hat{V}_i\}_{i=1}^k$  on  $M$ . For each geodesic Voronoi region  $\hat{V}_i$ , its nominal mass centroid  $m_i$  is defined by Eq.(9). We call the geodesic Voronoi tessellation as *geodesic centroidal Voronoi tessellation* (GCVT) if all the generators are nominal mass centroids, i.e.

$$g_i = m_i, \quad i = 1, \dots, k. \quad (10)$$

We define an energy function  $\varepsilon_g$  from any tessellation  $\{V_i\}_{i=1}^k$  of  $M$  and  $k$  points  $\{p_i\}_{i=1}^k \in M$ :

$$\varepsilon_g(\{(g_i, V_i)\}_{i=1}^k) = \sum_{i=1}^k \int_{x \in V_i} \rho(x) d_g^2(x, g_i) dx. \quad (11)$$

We call  $\varepsilon_g$  the GCVT *energy functional*. It can be shown that the necessary condition for  $\varepsilon_g$  being minimized is that  $\{(g_i, \hat{V}_i)\}_{i=1}^k$  is a GCVT [12] so we can obtain a GCVT by optimizing the GCVT energy functional. The theoretical results for the combinatorial structures of geodesic Voronoi tessellations will be presented in Section 5 and the algorithm for finding a GCVT will be introduced in Section 6.

### 3.1 Comparison of CVT, CCVT and GCVT

At the first glance, CVT, CCVT and GCVT are all uniform tessellations with similar formulations and we call the Voronoi regions of any tessellation as *cells*. CVT is defined in Euclidean spaces and its cells are convex polygon/polyhedra that are the intersection of half spaces. Both CCVT and GCVT are defined on manifold domains. However, CCVT needs that the manifold is embedded in a Euclidean space such that the Euclidean metric  $d_E$  can be used in Eq.(6). Then we say CCVT is *extrinsic*, which could also be interpreted as the intersection between the manifold and the Voronoi

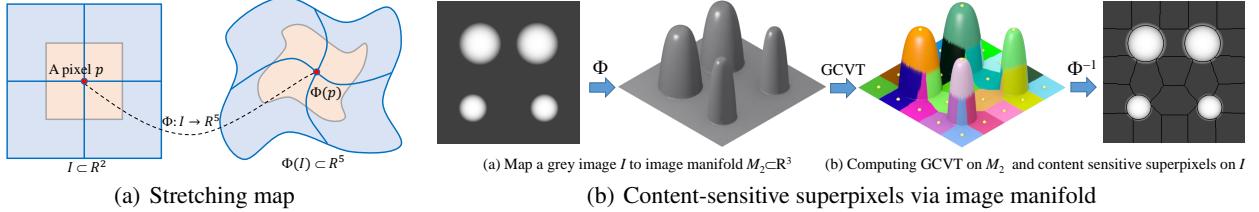


Figure 3: (a) The map  $\Phi$  that stretches an image  $I \subset \mathbb{R}^2$  to an image manifold  $M_2 \subset \mathbb{R}^5$ . (b) A uniform tessellation on  $M_2$  such as GCVT naturally induces content-sensitive superpixels in  $I$  via  $\Phi^{-1}$ ; for an easy illustration, a gray image is used for a mapping in  $\mathbb{R}^3$ .

tessellation in Euclidean space. Therefore, the cells in CCVT may be disconnected or multiply-connected<sup>2</sup>; see Figures 1 and 2 for an illustration.

GCVT is defined on a compact differentiable manifold without referring to an embedded Euclidean space. Since GCVT only relies on the geodesic metric, it is *intrinsic*. Compared to CCVT, all the cells in GCVT are guaranteed to be connected.

## 4 Applications

Before we present theoretic and algorithmic results of GCVT, we state three representative applications, showing that GCVT is a useful tool in computer vision and graphics.

### 4.1 Content-Sensitive Superpixels

Image pixels are only the units of image capturing device, but not optimized for image content presentation. Superpixels are a dense over-segmentation of image, which capture well image features and can serve as perceptually meaningful atomic regions for images. Superpixels can be used as a preprocessing for reducing the complexity of subsequent image processing tasks, which includes segmentation[13], contour closure[14], object location [15], object tracking [16], stereo 3D reconstruction [17], and many others. See [18] for a comprehensive survey.

To serve as perceptually meaningful atomic regions, superpixels generally have the following characteristics [12]:

- (1) *Partition*: each pixel in the image is assigned to exactly one superpixel so superpixels are a partition of the image;
- (2) *Connectivity*: each region of superpixel is simply connected;
- (3) *Compactness*: in the non-feature region, superpixels are regular in shape and uniform in size;
- (4) *Feature preservation*: superpixels should adhere well to image boundaries for preserving feature;
- (5) *Content sensitivity*: the density of superpixels is adaptive to the variety of image contents;
- (6) *Performance*: superpixels should be computed in a low cost of time and space.

In [19, 12], we propose an image manifold that maps a color image  $I$  from  $\mathbb{R}^2$  to a 2-manifold  $M_2$  embedded in the 5-dimensional combined image and colour space  $\mathbb{R}^5$ :

$$\Phi(u, v) = (\lambda_1 p, \lambda_2 \mathbf{c}) = (\lambda_1 u, \lambda_1 v, \lambda_2 l, \lambda_2 a, \lambda_2 b), \quad (12)$$

where  $I(u, v)$  is a color image with pixel positions  $p = (u, v)$ ,  $\mathbf{c}(p) = (l(u, v), a(u, v), b(u, v))$  is the color at the pixel  $p$  in CIELAB color space,  $\lambda_1$  and  $\lambda_2$  are global stretching factors. The area elements in the image manifold  $M_2$  are a good measure of the content density in the image  $I$ . Then a uniform tessellation such as GCVT on  $M_2$  naturally induce good content-sensitive superpixels in  $I$ . See Figure 3 for an illustration.

GCVT is a powerful tool for superpixels due to the following reasons:

- (1) *Partition*: GCVT is a tessellation of  $M_2$  (and also  $I$  due to the one-to-one mapping  $\Phi$ );
- (2) *Connectivity*: each cell in GCVT is guaranteed to be connected;

<sup>2</sup>A region is *simply connected* if any simple closed curve in it can be continuously shrunk into a point without leaving the region. A connected region that is not simply connected is called *multiply connected*.

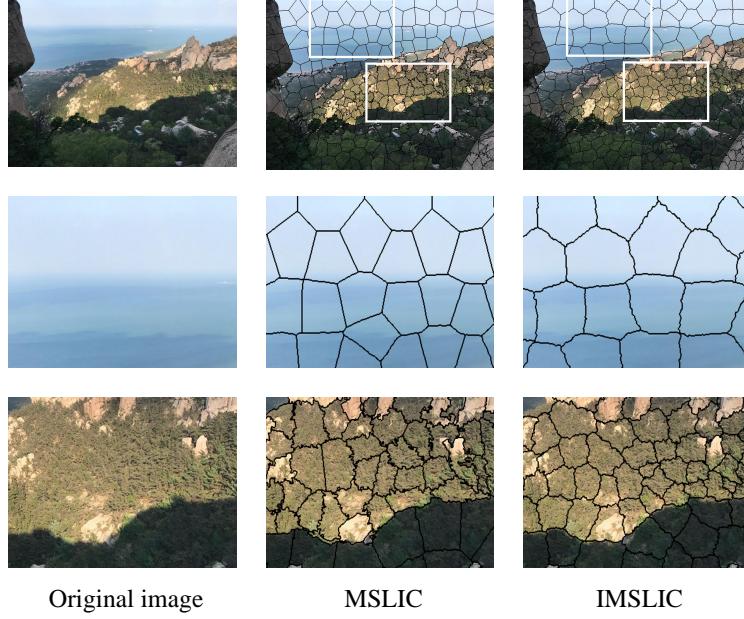


Figure 4: Some examples of two content-sensitive superpixels (MSLIC [19] and IMSLIC [12]) based on the uniform tessellation on the image manifold.

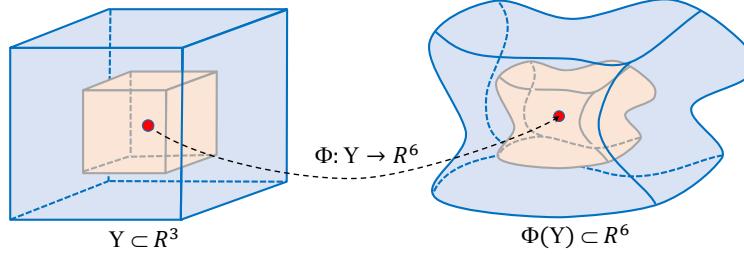


Figure 5: The stretching map  $\Phi : \Upsilon \rightarrow M_3 \subset \mathbb{R}^6$  maps a video  $\Upsilon \subset \mathbb{R}^3$  into a 3-manifold  $M_3 \subset \mathbb{R}^6$ .

- (3) *Compactness*: cells in GCVT are regular and uniform in non-feature regions;
- (4) *Feature preservation*: the feature regions (such as object boundary) have a large color variation and therefore lead to a large stretching/area on  $M_2$ . The larger the area in  $M_2$ , the higher possibility that a cell boundary passes through it;
- (5) *Content sensitivity*: content-dense regions have high intensity or color variation, and then larger area on  $M_2$ . Given uniform tessellation on  $M_2$ , the superpixels will be smaller in content-dense regions. Similarly, content-sparse regions have large superpixels;
- (6) *Performance*: we propose efficient computation methods in Section x that can quickly approximate GCVTs.

See Figure 4 for some qualitative results of content-sensitive superpixels.

## 4.2 Content-Sensitive Supervoxels

Akin to superpixels for images, Supervoxels are perceptually meaningful atomic regions in videos, obtained by grouping similar voxels that exhibit coherence in both appearance and motion. Superpixels over-segment a video in the spatiotemporal domain while well preserving its structural content.

To compute content-sensitive supervoxels, Yi et al. [20] extend the image manifold concept to the video manifold using the stretching map  $\Phi : \Upsilon \rightarrow M_3 \subset \mathbb{R}^6$  (Figure 5):

$$\Phi(\Upsilon) = \Phi(u, v, t) = (\lambda_1 u, \lambda_1 v, \lambda_2 t, \lambda_3 \mathbf{c}), \quad (13)$$

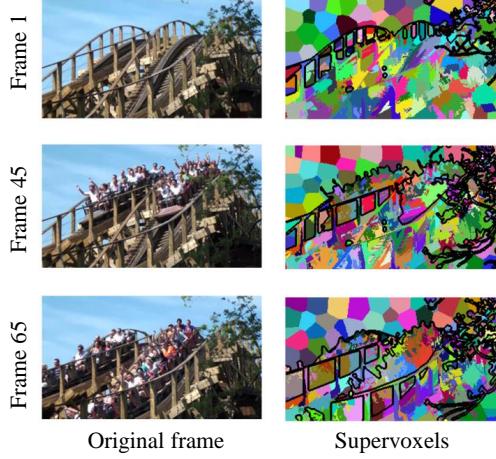


Figure 6: Examples of content-sensitive supervoxels [20] based on the uniform tessellation on the video manifold. For an easy illustration, supervoxels are clipped in each frame and shown as cross-sectional superpixels.

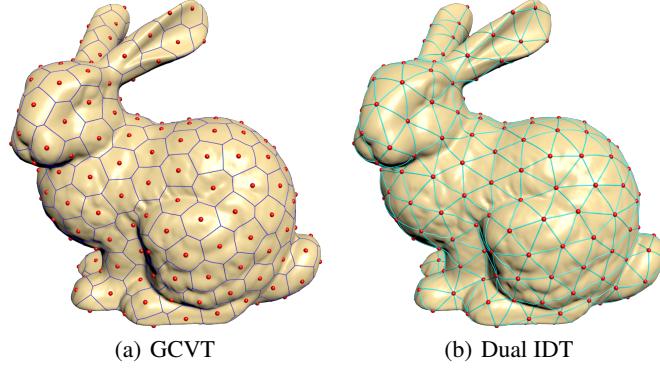


Figure 7: (a) GCVT with a small number of generators [21]. (b) Its dual intrinsic Delaunay triangulation (IDT) [22].

where the 3-manifold  $M_3$  is embedded in the 6-dimensional combined video and colour space  $\mathbb{R}^6$ ,  $\Upsilon$  is a video with  $N$  voxels,  $v(u, v, t) \in \Upsilon$  is a voxel with frame index  $t$  and the pixel position  $(u, v)$  in the frame,  $\mathbf{c}(v) = (l(u, v, t), a(u, v, t), b(u, v, t))$  is the color of  $v(u, v, t)$  in CIELAB color space  $\lambda_1, \lambda_2$  and  $\lambda_3$  are global stretching factors.

At the place where the color variation is large in  $\Upsilon$ ,  $\Phi$  maps a unit voxel into a large volume in  $M_3$ . Therefore, the volume elements in  $M_3$  offer a good measure of the content density in  $\Upsilon$ . In a similar way to content-sensitive superpixels, a uniform tessellation such as GCVT on  $M_3$  will naturally induce content-sensitive supervoxels in  $\Upsilon$ , i.e., supervoxels are typically larger and longer in content-sparse regions (i.e., with homogeneous appearance and motion), and smaller and shorter in content-dense regions (i.e., with high variation of appearance and/or motion). See Figure 6 for an illustration.

### 4.3 Low-Resolution Remeshing

In computer graphics, 3D shapes are usually represented by triangular 2-manifold. In many engineering applications such as finite element analysis, a high quality mesh with almost congruent triangles is desired. To convert an arbitrary triangular mesh into a high quality mesh while preserving geometric shapes, remeshing techniques are developed. Low-resolution remeshing is to generate a mesh with a small number of vertices and the vertex size approaching the feature size of the original high-resolution mesh. Due to the existence of thin-shell structures, Voronoi tessellations based on Euclidean metric frequently results in disjoint fragments in a Voronoi cell. See Figure 1b for an illustration.

GCVT can guarantee that each Voronoi cell is connected and thus is suitable for this low-resolution remeshing task. A globally optimized GCVT [21] can generate uniform tessellations with regular cells on a given high-resolution mesh.

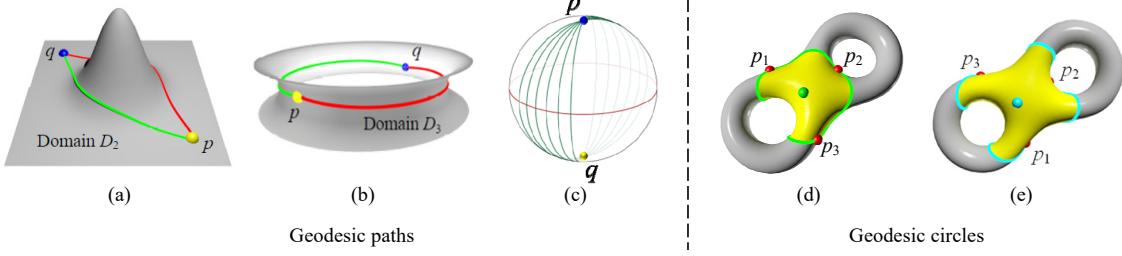


Figure 8: Geodesic paths and circles on curved manifolds. There are two different geodesic circles passing through three points  $p_1, p_2$  and  $p_3$  on  $M$ : one is in the front view (d) and the other is in the back view (e).

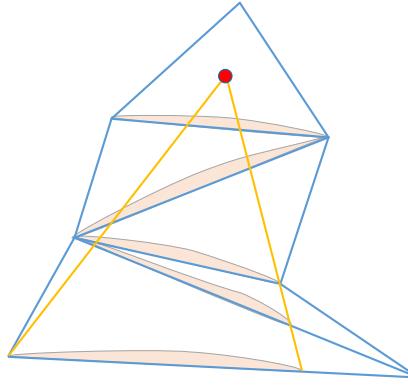


Figure 9: A visibility wedge (VW) on the bottom mesh edge.

We propose an efficient sampling criterion such that the intrinsic Delaunay triangulation due to the GVT exists [22]. Therefore, it provides an efficient solution to low-resolution remeshing. See Figure 7 for an illustration.

## 5 Theories

Geodesic paths are locally shortest paths between any two points on the manifold. Due to the bending of non-zero curvatures on curved manifolds  $M$ , the distance field on  $M$  characterized by geodesic distances/paths have quite different structures from that in Euclidean space  $\mathbb{R}^n$ , such as:

- Between any two non-duplicated points in  $\mathbb{R}^n$ , there is one and only one shortest path. However, between any two non-duplicated points on  $M$ , there may be one, two or infinite shortest paths; See Figure 8 (a-c) for an example. Only in a smooth, simply connected 2-manifold with negative Gaussian curvature everywhere, the geodesic path between any two points on  $M$  is unique.
- Given three points not lying on the same line in  $\mathbb{R}^n$ , there is a unique circle passing through them. However, given three points not lying on the same geodesic path on  $M$ , there may be no or more than one geodesic circles passing through them; See Figure 8 (d-e) for an example.

Therefore, the combinatorial structure of geodesic Voronoi tessellations on  $M$  are quite different from those in  $\mathbb{R}^n$ . Below we summarize the study of combinatorial structures on 2-manifold meshes  $M_2$  in hierarchical way [11, 23, 22].

### 5.1 Discrete geodesics

Mitchell et al. [7] establish the discrete geodesic structure on  $M_2$  (ref. Figure 9):

- Inside every triangle in  $M_2$ , geodesic paths are straight line segments;
- When crossing a mesh edge  $e$ , geodesic paths are straight lines if two adjacent faces of  $e$  are unfolded in the same plane along  $e$ ;

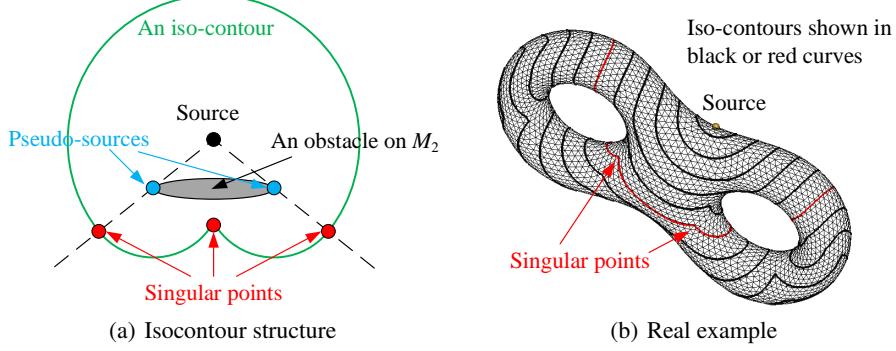


Figure 10: Each iso-contour of the distance field on a closed  $M_2$  consists of one or more closed curves and each closed curve consists of circular arc segments joined at singular points. The obstacle in (a) can be a mountain shape with a sufficient height.

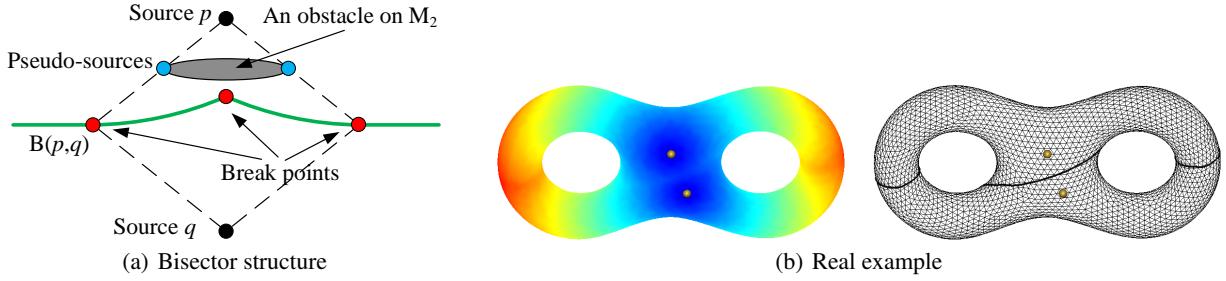


Figure 11: (a) If all vertices do not have the same geodesic distance to a given pair of source points, their bisector consists of 1D curve segments,  $C^0$  jointed at breakpoints. Between two breakpoints, the bisector portion is a line or hyperbolic segment. (b) A real example: the geodesic distance field (left) and bisectors between two points (right).

- Starting from the triangle that contain the source point, a visibility wedge (VW) can be initialized and propagated across edges until all edges in  $M_2$  are covered;
- The mesh vertices lying on any geodesics (i.e., the apexes of any VWs) are called *pseudo-sources*, which can only be saddle vertices, i.e., the vertices in  $M_2$  whose sum of surrounding angles is not smaller than  $2\pi$ .

The VW structure proposed in [7] can efficiently answer the single-source-all-destination discrete geodesic problem.

## 5.2 Iso-contour structure

Given one or more source points  $P = \{p_i\}_{i=1}^K$ , the geodesic distance is  $d_g(x) = \min_i\{d_g(x, p_i), p_i \in P\}, \forall x \in M_2$ . An iso-contour (a.k.a. level set) of the distance field  $d_g(x)$  is the trace of all points on  $M_2$  that have the same distance value. Iso-contours had drawn considerable attention in literature. On 2-manifold meshes  $M_2$ , their analytical structures are studied in [11] (ref. Figure 10):

- Due to the existance of pseudo-sources, each iso-contour of the distance field on a closed  $M_2$  consists of one or more closed curves;
- Each closed curve consists of circular arc segments joined at *singular* points, which are locations where the nearest pseudo-source is changing from one to another;
- The number of closed curves in an isocontour depends on the indices of critical points of the distance field function, where a point  $c \in M_2$  is a critical point of the distance field function  $D$ , if the partial derivatives of  $D$  vanish at  $c$ . The index of a critical point  $c$  is the number of negative eigenvalues of a Hessian matrix of  $D$  at  $c$ .

## 5.3 Bisector structure

The bisector between any two points on  $M_2$  is the trace of all points that have equal geodesic distance to these two points. On 2-manifold meshes  $M_2$ , the analytical structure of bisectors are studied in [11] (ref. Figure 11):

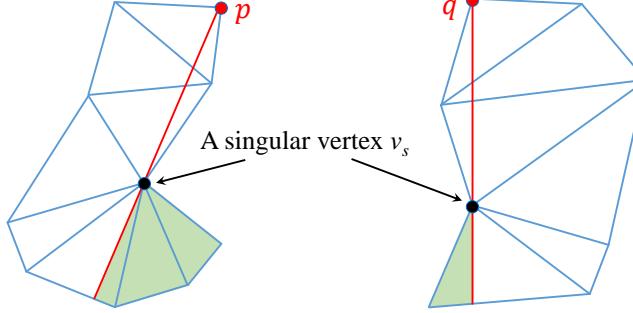


Figure 12: If a saddle vertex  $v_s$  lies on the bisector of two points  $p$  and  $q$ , i.e.,  $d_g(p, v_s) = d_g(q, v_s)$ , this bisector contains a 2D regions (shaded area).

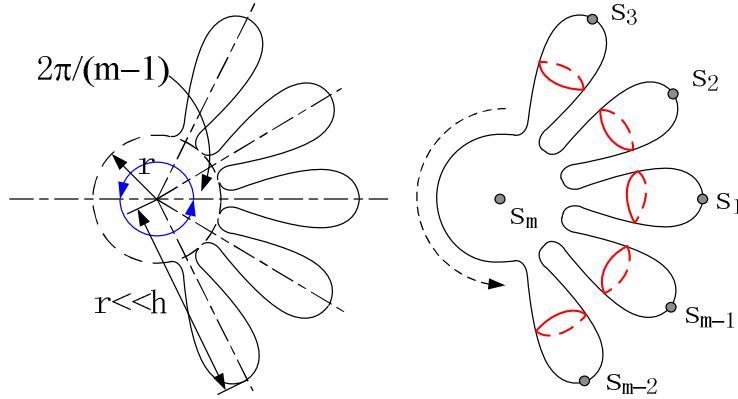


Figure 13: The geodesic Voronoi tessellation of the generator set  $\{s_i\}_{i=1}^m$ . The geodesic Voronoi cell of  $s_m$  has  $m - 1$  closed Voronoi edges, each of which is a bisector between  $s_m$  and  $s_i$ ,  $i = 1, 2, \dots, m - 1$ .

- The bisector between any two points  $p$  and  $q$  on  $M_2$  may not be 1D. If a saddle vertex lies on the bisector, then this bisector contains a 2D region on  $M_2$ . See Figure 12. Another example can be found in [24] (Figure 4);
- Upon small perturbation on the vertices of  $M_2$ , we can assume all vertices do not have the same geodesic distance to a given pair of source points, and then their bisector consists of 1D curve segments;
- If the bisector of two points is 1D, this bisector can have at most  $g + 1$  disjoint closed curves, where  $g$  is the genus of  $M_2$ ;
- For each closed curve in a bisector, it can be decomposed at breakpoints, which are the locations where the nearest pseudo-source is changing along the bisector. The bisector is only  $C^0$  at breakpoints. Between two adjacent breakpoints, the bisector portion can only be line or hyperbolic segment.

#### 5.4 Geodesic Voronoi Tessellation

Given a set of generators, the trimmed bisectors among them partition  $M_2$  into geodesic Voronoi cells. Their analytical structures are studied in [11] (ref. Figures 13 and 14):

- Each geodesic Voronoi cell is connected, but may not be singly connected;
- Each geodesic Voronoi cell is bounded by one or more closed curves called Voronoi edges. Each Voronoi edge consists of trimmed bisectors. Trimmed bisectors are joined at branch points that are locations on  $M_2$  having the same geodesic distance to its three closest generators. One Voronoi edge does not have to contain a branch point.

It is well known that in  $\mathbb{R}^2$ , given a set of  $n$  generators, there are at most  $3n - 6$  Voronoi edges and  $2n - 5$  branch points (also called *Voronoi vertices*) in Voronoi tessellation. The combinatorial structure of geodesic Voronoi tessellation on  $M_2$  is studied in [23]:

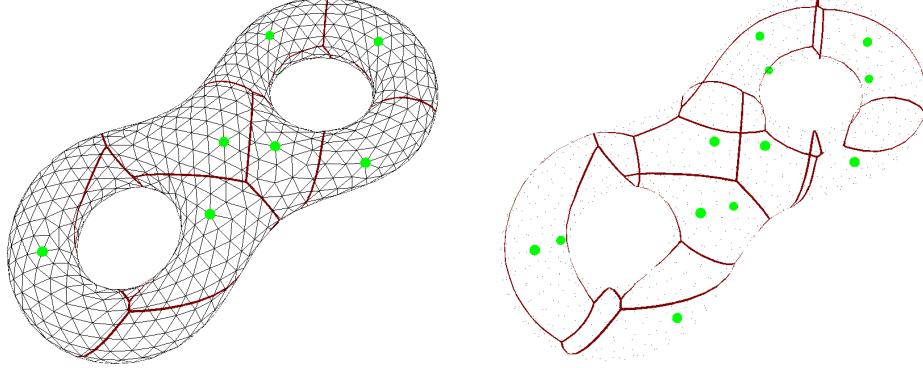


Figure 14: A real example of geodesic Voronoi tessellation of 12 point generators on a 2-manifold mesh with 2022 faces. Left is the geodesic Voronoi tessellation on the mesh. Right shows the boundary of each geodesic Voronoi cell with transparent surface rendering.

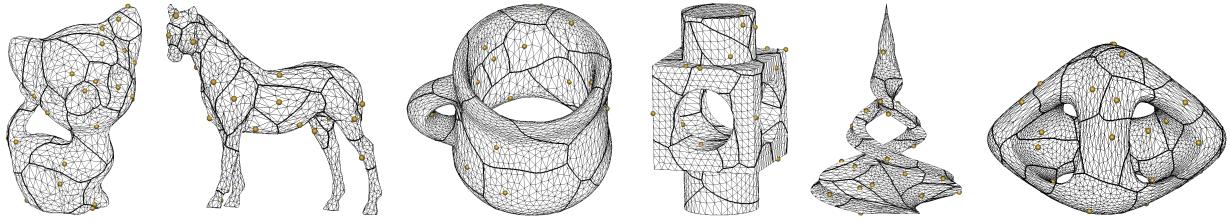


Figure 15: Some examples of geodesic Voronoi tessellations.

- On a genus-0  $M_2$ , the number of Voronoi vertices and Voronoi edges is  $O(m)$ , where  $m$  is the number of generators;
- The combinatorial complexity of geodesic Voronoi tessellation is defined to be the total number of breakpoints, Voronoi vertices, Voronoi edges and Voronoi cells. On a genus-0  $M_2$ , the combinatorial complexity of geodesic Voronoi tessellation is  $O(mk)$ , where  $k$  is the number of faces in  $M_2$ ;
- On a genus- $g$   $M_2$ , the number of Voronoi vertices and Voronoi edges is  $O(m + g)$ , where  $g$  is the genus of  $M_2$ ;
- If the set of generators is dense and the geodesic Voronoi tessellation satisfies the closed ball property [25], the combinatorial complexity of geodesic Voronoi tessellation on a genus- $g$   $M_2$  is  $O((n + g)k)$ .

Some real examples of geodesic Voronoi tessellations are illustrated in Figure 15.

## 6 Algorithms

Given the number  $N$  of point generators on a  $d$ -manifold mesh, the GCVT can be computed by finding a tessellation that minimizes the GCVT energy in (11). In this section, we present algorithms focusing on  $d = 2$ . Some methods summarized in this section (e.g., the RCVT approximation method in Section 6.2.2) can be extended to arbitrary  $d \geq 2$  dimensions.

The energy in (11) can be minimized globally or locally. There are two existing global optimization methods for CVT: one is the Monte Carlo with minimization (MCM) framework [26] that only deals with CVT in  $\mathbb{R}^2$  and the other is the manifold differential evolution (MDE) method [21] that is general to deal with GCVT on  $M_2$  ( $\mathbb{R}^2$  is a special case of  $M_2$ ). MCM is a heuristic method without theoretical guarantee, while MDE has a provable probabilistic convergence to the global optimum. In Section 6.1, we summarize the MDE global method.

Although the global method can achieve high-quality GCVT results and is insensitive to the initial position of generators, it is very time-consuming. Therefore, several fast, local optimization methods had proposed and we summarize two local methods in Section 6.2.

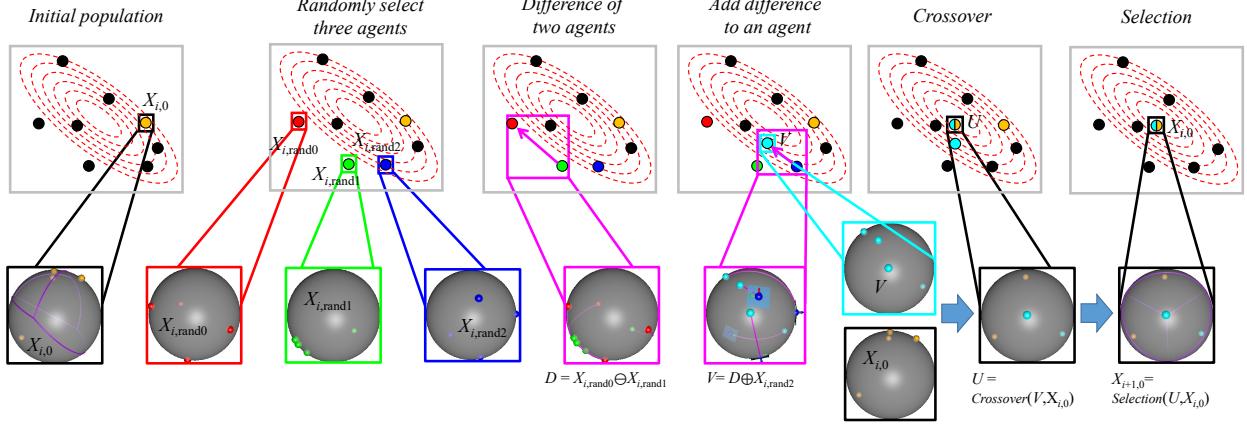


Figure 16: MDE pipeline.

### 6.1 MDE Global Method

MDE is a stochastic global optimization method, which extends the classic differential evolution [27] to the manifold setting.

Classic differential evolution applies agents that have operations of addition, subtraction and scala multiplication, all defined in a vector space. To use these operations on a manifold, MDE assigned an order to the generators in an agent for encoding them into a vector representation such that different agents can be matched akin to matching vectors. The pipeline of MDE is illustrated in Figure 16. It initializes a group of agents by random sampling and improves the quality of the agents iteratively. There are three steps in each iteration of MDE, i.e. mutation, crossover and selection.

*Vector representation and agent matching.* Given any two agents  $G = \{g_i\}_{i=1}^n$  and  $G' = \{g'_i\}_{i=1}^n$ , MDE builds a complete bipartite graph  $K_{n,n}$  whose vertices are  $G \cup G'$  and edges weights are the geodesic distances between the pair of any two  $g_i$  and  $g'_j$ . MDE finds a perfect matching in  $K_{n,n}$  by solving the minimum-weight perfect matching problem using the Hungarian algorithm [28], which runs in  $O(n^3)$  time. An order of generators can be induced by this matching, and thus the subscripts of generators can be corresponding by rearrangement. Let  $X_{k,j} = \{x_{k,j,i}\}_{i=1}^N$ ,  $j = 1 \dots M$  be the  $j$ -th agent of  $k$ -th generation, where  $N$  is the number of generators and  $M$  is the number of agents.

*Mutation operator.* MDE produces  $M$  new *mutative* agents by the mutation operator. The  $j$ -th mutative agent can be obtained by randomly selecting three agents and adding a scaled difference between two agents to another agent, i.e.

$$V_{k,j} = X_{k,rand3} \oplus (\lambda \otimes (X_{k,rand1} \ominus X_{k,rand2})), \quad (14)$$

where  $\ominus(y, x) : M \times M \rightarrow T_x M$  outputs a tangent vector at  $x$  whose direction is the starting direction of the geodesic path from  $x$  to  $y$  and whose magnitude is length of the geodesic path,  $\otimes$  is the scala multiplication in the tangent space and  $\oplus(z, v) : M \times T(M) \rightarrow M$  obtains a point on the manifold by (1) parallel transporting  $v$  to  $z$  along the geodesic path from the point of  $v$  to  $z$ , and denote the resulting vector as  $v'$ ; (2) computing a geodesic path from  $z$  with a initial direction  $v'$  and the length is equal to  $v'$ , and the end point of this geodesic path is the output.

*Crossover operator.* For each agent  $X_{k,j}$ , MDE produces a *competitor*  $U_{k,j} = \{u_{k,j,i}\}_{i=1}^N$  using the corresponding mutative agent  $V_{k,j}$  by crossover operator.  $u_{k,j,i}$  randomly uses the corresponding component of  $X_{k,j}$  or  $V_{k,j}$ , i.e.,

$$u_{k,j,i} = \begin{cases} v_{k,j,i} & \text{rand}_{(0,1)} < C_r \\ x_{k,j,i} & \text{otherwise} \end{cases}, \quad (15)$$

where  $C_r$  is *crossover rate*.

*Selection operator.* MDE selects a better agent from  $X_{k,j}$  and  $U_{k,j}$  and puts it into the new generation, i.e.,

$$X_{k+1,j} = \begin{cases} X_{k,j} & \varepsilon(X_{k,j}) < \varepsilon(U_{k,j}) \\ U_{k,j} & \text{otherwise} \end{cases}, \quad (16)$$

where  $\varepsilon$  is the GCVT energy of the agent computed by Equation 11.

The terminate condition of MDE is meeting one of the three conditions: (1) the iteration number exceeds the parameter specified by user; (2) the solution does not improve in successive several iterations; and (3) the GCVT energy reaches the prescribed value.

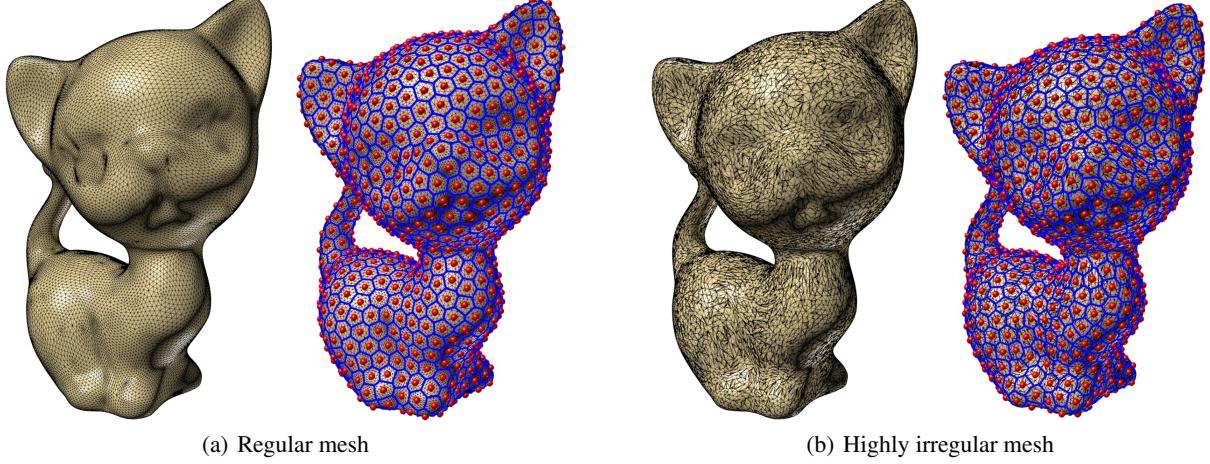


Figure 17: The regular mesh in (a) left and the irregular mesh in (b) left represent the same geometry. By applying the globally optimized MDE solution, the GCVTs on both meshes with the same number of point generators are the same.

It was shown in [21] that under some mild assumptions, the MDE solution converges to the global optimization with probability 1.

The globally optimized MDE solution is insensitive to the underlying mesh quality. See Figure 17 for an example.

## 6.2 Two Local Methods

### 6.2.1 Approximate nominal mass centroid

The Lloyd method [9] is a classic algorithm that can be used to efficiently compute the cluster problem, including the construction of CVT and GCVT. The Lloyd method locally minimizes the GCVT energy by iteratively moving the generators to the corresponding nominal mass centroids and updating the GVT of these generators. I.e., in each iteration, there are two steps in Lloyd method: one is fixing the tessellation and the moving generators to the corresponding nominal mass centroids and the other is fixing the generators and updating the GVT. Both of the two steps reduce the GCVT energy [12], which ensures the convergence of the Lloyd algorithm.

The bottleneck of this Lloyd method lies on the computation of the nominal mass centroids, which requires to solve the optimization problem in (9). It is difficult or even impossible to solve this problem analytically and thus approximation has to be used. Wang et al. [29] propose an approximation method that computes the Riemannian center instead of solving the problem (9). Let  $v_1, v_2, \dots, v_m$  be the Voronoi vertices of a geodesic Voronoi cell on  $M_2$ . The Riemannian center is defined as the local minima of the following function:

$$E(x) = \sum_{i=1}^m d_g^2(x, v_i) \quad (17)$$

Based on the properties studied in [30, 31], an iterative method utilizing the exponential map is proposed in [29] to quickly find an approximation to the Riemannian center. Another approximation to the nominal mass centroid is proposed in [12] that makes use of the landmark MDS (LMDS) [32] to quickly unfold a geodesic Voronoi cell into  $\mathbb{R}^2$  in a way such that the total distance distortion defined by a graph embedding is minimized. Then the nominal mass centroid is approximated by the mass centroid (ref. Eq.(2)) of unfolded geodesic Voronoi cell in  $\mathbb{R}^2$ .

### 6.2.2 RCVT approximation

The two local methods summarized in Section 6.2.1 make use of geodesic distance, which is time-consuming to compute for updating GVT in each Lloyd iteration.

Restricted centroidal Voronoi tessellation(RCVT) — which utilizes Euclidean distance in embedded Euclidean space — is proposed in [19, 20] as a fast approximation to GCVT. Different from the two local methods in Section 6.2.1 that can only handle the 2-manifold meshes, the RCVT summarized in this section can deal with any  $d$ -dimensional triangulated meshes,  $d \geq 2$ .

Given a  $d$ -manifold  $M$  embedded in  $\mathbb{R}^n$ ,  $d < n$ , and a set of point generators  $\{g_i\}_{i=1}^k \in \mathbb{R}^n$  (not necessary on  $M$ ), the *restricted Voronoi region* corresponding to the point  $g_i$  is defined as

$$\hat{V}_i = \{x \in M \mid d_E(x, g_i) < d_E(x, g_j) \text{ for } j = 1, \dots, k, j \neq i\}. \quad (18)$$

Elements of  $\{g_i\}_{i=1}^k$  are called *generators*. The set of *restricted Voronoi regions*  $\{\hat{V}_i\}_{i=1}^k$  forms a *restricted Voronoi tessellation* of  $M$ . The *mass centroid*  $m_i$  of  $\hat{V}_i$  is defined by

$$m_i = \frac{\int_{x \in \hat{V}_i} x dx}{\int_{x \in \hat{V}_i} dx}. \quad (19)$$

Similar to point generators  $\{g_i\}_{i=1}^k$ , the mass centroid  $m_i$  does not need to be on  $M$ . The restricted Voronoi tessellation is called *restricted centroidal Voronoi tessellation(RCVT)* if all the generators are mass centroids, i.e.

$$g_i = m_i, \quad i = 1, \dots, k. \quad (20)$$

Due to the following properties:

- Compared to GCVT, RCVT uses Euclidean distance and
- Compared to CCVT, the mass centroids do not need to be on  $M$ ,

RCVT behaves as a natural bridge between GCVT and CCVT. Finally, RCVT is easy to compute using the Lloyd method with Euclidean distance.

### 6.3 Algorithm Analysis

In this section, we provide an analysis on the time complexity and space complexity of the above methods for constructing GCVT.

In MDE, the agent matching runs in  $O(m^3)$  time and computing the exact geodesic between two points runs in  $O(n^2 \log n)$  time, where  $m$  is the number of generators and  $n$  is the number of vertices. Therefore, generating an agent takes  $O(m^3 + n^2 \log n)$  time and MDE runs in  $O(N_p K(m^3 + n^2 \log n))$ , where  $N_p$  is population size and  $K$  is iteration number. It takes  $O(N_p m)$  space for  $N_p$  agents which have  $N_p m$  generators in total.

IMSLIC [12] and MSLIC [19] construct GCVT and RCVT for an image using Lloyd method. It alternately constructs GVT and computes mass centroids. We analyze the complexities of the two steps separately. Fixing generators, IMSLIC can construct a GVT corresponding to the generators using Dijkstra's algorithm in  $O(n \log n)$  time and  $O(n)$  space. Adopting a label correcting method that maintains a bucket data structure can reduce the complexity and it only takes  $O(n)$  time. Fixing generators, MSLIC can construct a RVT corresponding to the generators by going through all pixels in  $O(n)$  time and  $O(n)$  space. Fixing the tessellation, IMSLIC computes approximate nominal mass using LMDS in  $O(n)$  time and  $O(n)$  space and MSLIC computes mass centroids by going through all pixels in  $O(n)$  time and  $O(1)$  space. Therefore, we can obtain an approximate GCVT in  $O(nK)$  time and  $O(n)$  space and obtain an RCVT in  $O(nK)$  time and  $O(n)$  space for an image, where  $K$  is the number of iteration.

## 7 Conclusion

Geodesic centroidal Voronoi tessellations (GCVTs) are intrinsic geometric structure inherent in manifolds. In this paper, we summarized our recent work on GCVTs on triangulated manifold. We show our results from both theoretical and algorithmic aspects. Their applications in computer graphics and vision are also presented. We hope that this paper can provide some insights for researchers to review the past developments and identify directions for future research on the study of intrinsic geometric structures in intelligent media data processing.

## References

- [1] Atsuyuki Okabe, Barry Boots, Kokichi Sugihara, and Sung Nok Chiu. *Spatial Tessellations: Concept and Applications of Voronoi Diagrams*. Wiley, 2000.
- [2] Jeffrey M Augenbaum and Charles S Peskin. On the construction of the voronoi mesh on a sphere. *Journal of Computational Physics*, 59(2):177–192, 1985.
- [3] Kensuke Onishi and Nobuki Takayama. Construction of voronoi diagram on the upper half-plane. *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, E79-A(4):533–539, 1996.

- [4] Greg Leibon and David Letscher. Delaunay triangulations and voronoi diagrams for riemannian manifolds. In *Proceedings of the Sixteenth Annual Symposium on Computational Geometry*, SCG '00, pages 341–349. ACM, 2000.
- [5] H. Sebastian Seung and Daniel D. Lee. The manifold ways of perception. *Science*, 290(5500):2268–2269, 2000.
- [6] William M Boothby. *An introduction to differentiable manifolds and Riemannian geometry*. Academic Press, 1975.
- [7] Joseph SB Mitchell, David M Mount, and Christos H Papadimitriou. The discrete geodesic problem. *SIAM Journal on Computing*, 16(4):647–668, 1987.
- [8] Qiang Du, Vance Faber, and Max Gunzburger. Centroidal voronoi tessellations: Applications and algorithms. *Siam Review*, 41(4):637–676, 1999.
- [9] Stuart P. Lloyd. Least squares quantization in pcm's. *IEEE Transactions on Information Theory*, 28:129–136, 03 1982.
- [10] Qiang Du, Max D. Gunzburger, and Lili Ju. Constrained centroidal voronoi tessellations for surfaces. *SIAM Journal on Scientific Computing*, 24(5):1488–1506, 2003.
- [11] Liu Yong-Jin, Chen Zhan-Qing, and Tang Kai. Construction of iso-contours, bisectors, and voronoi diagrams on triangulated surfaces. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 33(8):1502–1517, 2011.
- [12] Yong-Jin Liu, Minjing Yu, Bing-Jun Li, and Ying He. Intrinsic manifold SLIC: A simple and efficient method for computing content-sensitive superpixels. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(3):653–666, 2018.
- [13] L. Ming Yu, O. Tuzel, S. Ramalingam, and R. Chellappa. Entropy rate superpixel segmentation. In *Computer Vision & Pattern Recognition*, 2011.
- [14] Alex Levinstein, Cristian Sminchisescu, and Sven Dickinson. Optimal contour closure by superpixel grouping. In *European Conference on Computer Vision*, 2010.
- [15] Brian Fulkerson, Andrea Vedaldi, and Stefano Soatto. Class segmentation and object localization with superpixel neighborhoods. In *IEEE International Conference on Computer Vision*, 2009.
- [16] Wang Shu, Huchuan Lu, Yang Fan, and Ming Hsuan Yang. Superpixel tracking. In *International Conference on Computer Vision*, 2011.
- [17] Branislav Mičušík and Jana Košecká. Multi-view superpixel stereo in urban environments. *International Journal of Computer Vision*, 89(1):106–119, 2010.
- [18] David Stutz, Alexander Hermans, and Bastian Leibe. Superpixels: An evaluation of the state-of-the-art. *Computer Vision and Image Understanding*, 166:1–27, 2018.
- [19] Yong-Jin Liu, Chengchi Yu, Minjing Yu, and Ying He. Manifold SLIC: a fast method to compute content-sensitive superpixels. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '16, pages 651–659, 2016.
- [20] Ran Yi, Yong-Jin Liu, and Yu-Kun Lai. Content-sensitive supervoxels via uniform tessellations on video manifolds. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '18, pages 646–655, 2018.
- [21] Yong-Jin Liu, Chun-Xu Xu, Ran Yi, Dian Fan, and Ying He. Manifold differential evolution (MDE): A global optimization method for geodesic centroidal voronoi tessellations on meshes. *ACM Transactions on Graphics (SIGGRAPH ASIA 2016)*, 35(6):243:1–243:10, 2016.
- [22] Yong-Jin Liu, Dian Fan, Chun-Xu Xu, and Ying He. Constructing intrinsic delaunay triangulations from the dual of geodesic voronoi diagrams. *ACM Trans. Graph.*, 36(2):15:1–15:15, 2017.
- [23] Yong-Jin Liu and Kai Tang. The complexity of geodesic voronoi diagrams on triangulated 2-manifold surfaces. *Information Processing Letters*, 113(4):132–136, 2013.
- [24] Yong-Jin Liu. Semi-continuity of skeletons in 2-manifold and discrete voronoi approximation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1938–1944, 2015.
- [25] Herbert Edelsbrunner and Nimish R. Shah. Triangulating topological spaces. *International Journal of Computational Geometry and Applications*, 7(4):365–378, 1997.
- [26] Lin Lu, Feng Sun, Hao Pan, and Wenping Wang. Global optimization of centroidal voronoi tessellation with monte carlo approach. *IEEE Trans. Vis. Comput. Graph.*, 18(11):1880–1890, 2012.
- [27] Swagatam Das and Ponnuthurai Nagarathnam Suganthan. Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, 15(1):4–31, 2011.
- [28] B B. H. Korte and Jens Vygen. *Combinatorial Optimization: Theory and Algorithms*. Springer, 2012.

- [29] Xiaoning Wang, Xiang Ying, Yong-Jin Liu, Shi-Qing Xin, Wenping Wang, Xianfeng Gu, Wolfgang Müller-Wittig, and Ying He. Intrinsic computation of centroidal voronoi tessellation (CVT) on meshes. *Computer-Aided Design*, 58:51–61, 2015.
- [30] Xavier Pennec. Intrinsic statistics on riemannian manifolds: Basic tools for geometric measurements. *Journal of Mathematical Imaging and Vision*, 25(1):127–154, 2006.
- [31] Raif M. Rustamov. Barycentric coordinates on surfaces. *Comput. Graph. Forum*, 29(5):1507–1516, 2010.
- [32] Vin de Silva and Joshua B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In *Advances in Neural Information Processing Systems (NIPS '02)*, pages 705–712, 2002.