Anisotropic centroidal voronoi tessellations and their applications

Wang, Desheng; Du, Qiang

2005

Wang, D., & Du, Q. (2005). Anisotropic centroidal voronoi tessellations and their applications. Siam Journal on Scientific Computing, 26(3), 737–761.

https://hdl.handle.net/10356/98741

https://doi.org/10.1137/S1064827503428527

SIAM J. SCI. COMPUT.
Vol. 26, No. 3, pp. 737–761

ANISOTROPIC CENTROIDAL VORONOI TESSELLATIONS AND THEIR APPLICATIONS*

QIANG DU† AND DESHENG WANG‡

Abstract. In this paper, we introduce a novel definition of the anisotropic centroidal Voronoi tessellation (ACVT) corresponding to a given Riemann metric tensor. A directional distance function is used in the definition to simplify the computation. We provide algorithms to approximate the ACVT using the Lloyd iteration and the construction of anisotropic Delaunay triangulation under the given Riemannian metric. The ACVT is applied to the optimization of two-dimensional anisotropic Delaunay triangulation, to the generation of surface CVT, and high-quality triangular mesh on general surfaces. Various numerical examples demonstrating the effectiveness of the proposed method are presented.

Key words. Voronoi tessellations, anisotropy, Riemannian metric, anisotropic Delaunay triangulation, optimal tessellations, optimal mesh, surface mesh, surface triangulation

AMS subject classifications. 65D18, 65D17, 65N50, 65Y20

DOI. 10.1137/S1064827503428527

1. Introduction.

Anisotropic triangulations, in particular anisotropic Delaunay triangulations, have attracted the attention of many researchers (see [6, 9, 10, 11, 22, 25, 28, 29, 31, 34] and the references cited therein) due to their various applications ranging from volume and surface mesh generation to surface representation and image morphing. For best-performing triangulations, from the approximation theory point of view, it is well known that the aspect ratios and orientations of the triangles or tetrahedra should depend on the problems whose solutions are to be approximated. For problems such as the boundary layer in fluid flow, the solutions often display anisotropic behavior and are best resolved with anisotropic meshes.

The quality of an anisotropic triangulation, the same as in the isotropic case [1, 19], is closely related to the distribution of the vertices. In [12], a methodology for optimal points placement in regions, i.e., volumes, in $\mathbb{R}^d$ has been developed, based on the notion of centroidal Voronoi tessellations (CVTs). A CVT is a Voronoi tessellation whose generating points are the centroids (centers of mass) of the corresponding Voronoi regions associated with a given density. CVTs enjoy an optimization characteristic so that they themselves turn out to be useful in many applications such as image and data analysis, vector quantization, resource optimizations, statistics, and meshless computing; see, e.g., [12, 14, 15]. The concept of CVT has also been successfully applied to high-quality mesh generation and optimization [13, 17, 18, 19] and to the numerical solution of partial differential equations.

The basic definition of the CVT can be generalized to very broad settings ranging from abstract spaces to discrete point sets [12]. The purpose of this paper is to

*Received by the editors May 23, 2003; accepted for publication (in revised form) March 24, 2004; published electronically January 12, 2005. This project was supported in part by U.S. NSF-DMS 0196522, NSF-ITR 0205232, and China State Major Basic Research Project G199903280. Part of this work was completed while the authors were at the Lab for Scientific and Engineering Computing, Academy of Sciences, Beijing, China.

http://www.siam.org/journals/sisc/26-3/42852.html

†Department of Mathematics, Penn State University, University Park, PA 16802 (qdu@math.psu.edu).

‡Civil and Computational Engineering Centre, School of Engineering, University of Wales Swansea, Singleton Park, Swansea SA2 8PP, UK (desheng.wang@swansea.ac.uk).

introduce a new and consistent definition for anisotropic centroidal Voronoi tessellations in the Euclidean space but related to a given Riemannian metric tensor which possesses anisotropy and to develop computational algorithms for their efficient construction. By introducing a directional distance definition for any two points as a significant simplification of the classical Riemannian distance measure, the notion of anisotropic Voronoi region (AVR), anisotropic Voronoi tessellation (AVT), and the corresponding anisotropic Delaunay triangulation (ADT) can be suitably defined. Their definitions are different from the standard ones in [12], and they also differ from other popular definitions used in the literature [25, 28]. As our definition leads to a straightforward definition of the mass centroids, it thus provides a consistent definition of the anisotropic centroidal Voronoi tessellations (ACVTs), which is the main concept to be discussed in this paper. The ACVTs enjoy useful optimization properties that are naturally tied to the basic function approximation theory, and they reduce to the standard CVTs for isotropic Riemannian tensors. When applied to surface tessellation and triangulation, our definition is also different from the notion of constrained CVTs discussed in [14], where the distance remains to be measured in the Euclidean metric and only the definition of the mass centroids reflects the surface geometry.

Even with the simplified notion of directional distances, the direct construction of AVT is still computationally challenging due to the generality and the complexity of the Riemannian metric. We employ the method of unit meshing proposed in [6, 22] to give an approximate construction of the ADT and subsequently the AVT. A key observation based on our computational experience is that the AVTs can often be well approximated by their visibility regions. Finally, to compute the ACVT, we extend the classical Lloyd method which iterates between the computation of the AVT with given generators and the computation of mass centroids with a given AVT. Our proposed algorithm is shown to be very effective for various examples.

A direct application of ACVT is the two-dimensional ADT via the optimal anisotropic centroidal Voronoi–Delaunay triangulation (ACVDT). Under the Riemannian metric, the density function of ACVT is defined to be unit, which means that, asymptotically speaking, the dual triangulation ACVDT of the final converged ACVT has approximately unit-length edges, and, accordingly, the triangulation is an almost regular anisotropic triangulation under the Riemannian metric. This is similar to our previous work in isotropic meshing [19]. Numerous anisotropic examples vindicate the above assertion. Another direct application of the ACVT constructed through the Lloyd iteration is the surface tessellation and triangulation, similar to the work in [14]. Various ACVTs and the resulting high-quality Delaunay meshes on general surfaces demonstrate the effectiveness of the proposed method. In this regard, this paper serves as a companion work of [14].

The remainder of the paper is organized as follows. First, in section 2, we review the basic notion of the CVT and introduce our definition of ACVT. Then, in section 3, we describe the algorithm for computing the ADT under general Riemannian metric tensor. In section 4, we address the main point of the paper: the generation of ACVT. We also give some ACVT examples. In section 5, we discuss the applications of ACVT: optimization for two-dimensional ADT and high-quality surface meshing. Some concluding remarks are given in section 6.

2. CVT and ACVT. For detailed discussions on CVT, we refer the reader to [12]. For comparison purposes, some basic CVT terminologies in the Euclidean metric are briefly reviewed here.

2.1. CVT in Euclidean metric. Let $|\cdot|$ denote the Euclidean norm in $\mathbb{R}^d$. Given a bounded open set $\Omega \subset \mathbb{R}^d$ and a set of points $\{Z_i\}_{i=1}^n$ belonging to $\bar{\Omega}$, let

$$V_i = \{x \in \Omega \,|\, |x - z_i| < |x - z_j| \text{ for } j = 1, \ldots, n, j \neq i\}, \quad i = 1, \ldots, n.$$

Clearly, we have $V_i \cap V_j = \emptyset$ for $i \neq j$ and $\bigcup_{i=1}^n \bar{V}_i = \bar{\Omega}$. The set $\{V_i\}_{i=1}^n$ is referred to as a *Voronoi tessellation* (VT) or a Voronoi diagram of Ω, the members of the set $\{z_i\}_{i=1}^n$ are referred to as *generating points* or generators, and each V_i is referred to as the Voronoi region or Voronoi cell corresponding to z_i. The VTs are very useful in many applications [12].

Given a density function $\rho = \rho(x)$ defined on $\bar{\Omega}$ and positive and continuous almost everywhere, for each Voronoi region V_i, we define its mass centroid z_i^* by

$$z_i^* = \frac{\int_{V_i} x\rho(x)dx}{\int_{V_i} \rho(x)dx} \quad \text{for} \quad i = 1, \ldots, n.$$

We call the tessellation defined as above a CVT if and only if

$$Z_i = z_i^* \quad \text{for} \quad i = 1, \ldots, n;$$

i.e., the points Z_i's which serve as the generators associated with the Voronoi region V_i's are the mass centroids of those regions. The existence of CVTs for a given set has been proved, but note that, in general, they are not uniquely defined [12]. The dual Delaunay triangulation of CVT is referred to as the centroidal Voronoi Delaunay triangulation (CVDT) [13, 19].

For any tessellation $\{V_i\}_1^n$ of the domain Ω and a set of points $\{z_i\}_1^n$ in Ω, we can define the following cost (or error or energy) functional:

$$F(\{V_i\}_{i=1}^n, \{z_i\}_{i=1}^n) = \sum_{i=1}^n \int_{V_i} \rho(x)||x - z_i||^2 dx. \tag{2.1}$$

The standard CVTs along with their generators are critical points of this cost functional. In practice, the positions of the generators may be limited by various constraints, for instance to be confined to surfaces. Hence, the notion of constrained CVT (CCVT) and its dual triangulation (CCVDT) have been introduced in [13, 14]. Simply, the CCVT is defined as the minimum of F under some specified constraints.

The CVT provides, in some sense, an optimal distribution of generating points for meshing purposes. We have applied this tool to both two-dimensional and three-dimensional optimal Delaunay mesh generation and optimization [13, 19]. Indeed, we have shown that the construction of CVDT generalizes many existing local smoothing techniques. The construction of CVT and CVDT is performed with respect to a given density function which in principle should reflect the behavior of problems whose solutions are to be sought after. The density function may be related to the sizing function of the triangulation, and the CVT (or CVDT), by minimizing the cost functional F, turns to improve element quality and reduce the sizing distortion. The celebrated Gersho's conjecture [23, 24] and extensive numerical studies concerning the asymptotic property of the optimal Voronoi cells provide additional support for the above optimal Delaunay triangulation framework.

There are several algorithms known for constructing CVT of a given set [12]. They can be classified as probabilistic and deterministic approaches. A representative probabilistic algorithm given by Macqueen is a very elegant random sequential

method which divides sampling points into several subsets or clusters by sequentially taking means of sampling points. The deterministic Lloyd algorithm is the obvious iteration between the computation of the Voronoi diagrams and mass centroids. In quantization and clustering literature, one can also find the related h-means and k-means algorithms for the construction of discrete CVTs.

The notions of Voronoi regions and centroids, and therefore the CVTs, can be generalized to more abstract spaces and to metrics other than the Euclidean L^2 norm, since there are many applications in computer science, arts, archaeology, astronomy, biology, crystallography, and physics that are related to generalized CVTs [12, 32]. In this paper, we are interested in the special two-dimensional space setting with a positive definite Riemannian metric tensor.

2.2. The Riemannian structure and metric tensor. Although the discussions here on the Riemannian structure can be easily extended to higher-dimensional spaces, to simplify the notation and illustration, we merely present the two-dimensional versions which are more directly related to the two-dimensional triangulation and surface meshing applications considered in this paper.

Let P be a point of the planar domain Ω. A metric tensor at P refers to the specification of a positive definite matrix (tensor)

$$M(p) = \begin{pmatrix} a(p) & b(p) \\ b(p) & c(p) \end{pmatrix},$$

where $a(p) > 0$, $c(p) > 0$, and $a(p)c(p) - b(p)^2 > 0$.

The metric field $(M(p))_{p\in\Omega}$ induces a *Riemannian structure* on Ω, and we denote it by $(\Omega, M(p))$. If M is constant, i.e., is independent of the position, we refer to it as a constant or *uniform* Riemannian metric. Otherwise, it is *nonuniform*. If for all points the metric tensor remains identity, or a constant multiple of the identity, it is simplified (or equivalent) to the standard Euclidean structure.

Since $M(P)$ is positive definite, through diagonalization, we have

$$M(P) = E^T U E \quad \text{with } E = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix},\ U = \begin{pmatrix} \lambda_1(P) & 0 \\ 0 & \lambda_2(P) \end{pmatrix}. \tag{2.2}$$

Here, $\lambda_1(P)$ and $\lambda_2(P)$ are the two eigenvalues, and E are the corresponding eigenvectors of $M(P)$. Let $h_1(P) = \sqrt{1/\lambda_1(P)}$, and $h_2(P) = \sqrt{1/\lambda_2(P)}$; then $h_1(P)$, $h_2(P)$, and θ can be interpreted as the lengths of the two radii and the rotation angle of the ellipse shown in Figure 2.1, and the metric $M = M(P)$ corresponds to such an ellipse with the specified orientation and aspect ratio [4, 6, 7, 22, 26, 27, 35]. If $h_1(P) = h_2(P)$ for all points P, i.e., the ellipses become conventional circles, then the metric is called an *isotropic* metric; otherwise, we have an anisotropic metric field.

Let P, Q be two points in Ω, and let $S = S(t)$ be a path connecting the two points and parameterized by $t \in [0, 1]$. The length of $S = S(t)$ corresponding to M can be calculated by

$$L_M(S) = \int_0^1 \sqrt{\overrightarrow{S'(t)}^T M(S(t)) \overrightarrow{S'(t)}}\, dt$$

with $\overrightarrow{S'(t)} = (u_1(t), u_2(t))^T$ and denote $M(S(t))$ by

$$M(S(t)) = \begin{pmatrix} a_s(t) & b_s(t) \\ b_s(t) & c_s(t) \end{pmatrix};$$

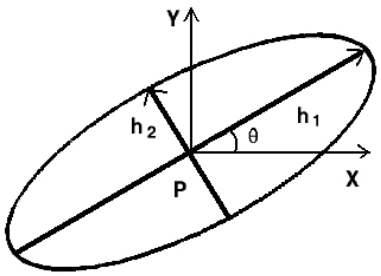

FIG. 2.1. *Anisotropy: Approximation by ellipse.*

then

$$L_M(S) = \int_0^1 \sqrt{a_s(t)u_1^2(t) + 2b_s(t)u_1(t)u_2(t) + c_s(t)u_2^2(t)}\, dt,$$

and the minimum of $L_M(S)$ gives the conventional Riemannian distance $d_M(P, Q)$.

As pointed out in [25], it is often difficult to compute the above distance for a general Riemannian metric. Moreover, even with the above distance definition, the VT and its dual Delaunay triangulation are not always well defined unless some conditions on the vertices are met [28]. We thus consider an alternative definition.

2.3. AVT in Riemannian metric. One crucial ingredient in the definition of the general CVT in the Euclidean space is the minimization of the corresponding cost functional. Such an optimization property is key to the success of CVT in various applications. To preserve such a property, a modified notion of distances and the mass centroids with respect to a given Riemannian metric need to be introduced.

We introduce first the *directional distance function*, the *AVR*, and the *AVT*; afterwards, we present the definitions of the cost functional, mass centers, and the ACVT.

DEFINITION 2.1. *Let P, Q be two points in Ω, and let M be a Riemannian metric defined as* (2.2). *We define the following simplified distance measure as the* directional distance *from Q to P:*

$$d_P(P, Q) = \sqrt{\overrightarrow{PQ}^T M(P) \overrightarrow{PQ}}\,.$$

The above directional distance between P and Q can be interpreted as a Riemannian distance between the two points, with the metric being a constant metric given by $M(P)$. Obviously, in contrast to the usual distance definitions, the above directional distance is not symmetric. This, however, is well suited to our definition of AVTs.

DEFINITION 2.2. *Given a set of points $\{Z_i\}_{i=1}^n$ in the domain Ω and a positive definite Riemannian metric M smoothly defined on Ω as in the above, the following set is defined as the* anisotropic Voronoi region *(AVR) of a point P in Ω:*

$$V(P) = \{x \in \Omega \mid d_x(x, P) < d_x(x, Z_i)\ \forall\ Z_i \neq P\}.$$

Points having equal directional distances to any two generators Z_i, Z_j are called their bisectors.

Clearly, the above definition is an extension of the classical definition in the isotropic case. The AVRs of any two distinct points are nonoverlapping, and the union of all the AVRs of $\{Z_i\}_{i=1}^n$ covers the domain. The smoothness of the Riemannian metric assures that the *bisectors* have measure zero and consist of piecewise smooth curves, though in practice the smoothness of the Riemannian metric is not always a necessity.

Given two points P, Q and their midpoint $X_m = (P+Q)/2$, since $\overrightarrow{X_m Q} = -\overrightarrow{X_m P}$, we easily see that $d_{X_m}(X_m, Q) = d_{X_m}(X_m, P)$. This leads to the following interesting property.

LEMMA 2.1. *The* bisector *of two generators defined using the directional distance always passes through their midpoint.*

We may attribute the above property as the midpoint of two generators being *blind* to the anisotropy with respect to the two generators. Note that AVRs defined as above are not necessarily connected. This is not surprising, as it is also the case for the AVRs defined in [25, 28]. We refer to the subregion of an AVR which contains the corresponding generator as its *main subregion*, while the other subregions that do not contain the generator we refer to as *orphan subregions*, just like in [25]. However, in most practical applications and with a sufficient number of generators, the appearance of orphan subregions is very rare. The part of the main subregion which is visible from the corresponding generator is called the *visibility region* in the AVR. The visibility regions are star-shaped with respect to the corresponding generators, and they serve as good approximations to the AVRs in practice.

With well-defined AVRs and bisectors, the VT corresponding to the directional distance can be defined.

DEFINITION 2.3. *Given a set of points $\{Z_i\}_{i=1}^n$ in the domain Ω and a positive definite Riemann metric M defined as above on Ω, the tessellation of Ω by the set of anisotropic Voronoi regions $\{V_i = V(Z_i)\}_{i=1}^n$ is called the* anisotropic Voronoi tessellation *(AVT). The dual triangulation obtained by joining generators whose main subregions share common bisectors is called the* anisotropic Delaunay triangulation *(ADT).*

Some examples of the AVRs and AVTs in a two-dimensional square $[-1, 1]^2$ with various Riemannian metric are illustrated in Figure 2.2. Five generators are taken, with one at the center and four on the vertices. Note that for $M = I$ we get the standard isotropic VTs. For $M = \mathrm{diag}(4, 1)$, we see the symmetry breaking due to the anisotropy and similarly for $M(x, y) = \mathrm{diag}(7 - 6.3|x|, 1)$. In the latter case, we also get two *orphan* subregions near points $(-1, 0)$ and $(1, 0)$ belonging to the AVR associated with the point at the center.

Since our notion of AVT differs significantly from the conventional definition, some comments are in order.

First of all, the use of a directional distance is not new. In fact, as we are finishing up this paper, it comes to our attention that such a notion is also used in a recent work [25]. There, the AVR is defined as

$$V(P) = \{x \in \Omega \mid d_P(x, P) < d_{Z_i}(x, Z_i)\ \forall Z_i \neq P\}.$$

Note that, to determine the membership of the Voronoi cells, the distances used in the above are in the *opposite* directions to that in our definition. Historically, the multiplicative weighted Voronoi tessellation (MWVT) may also serve as an earlier application of directional distances [3]. The definition of the multiplicative weighted

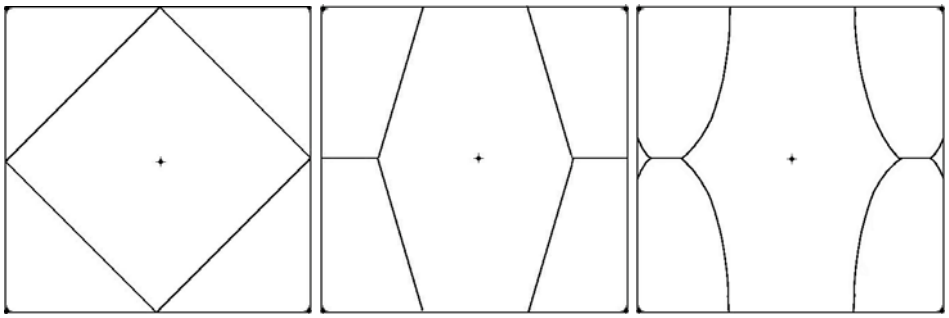

FIG. 2.2. *AVTs of* $[-1,1]^2$ *for* $M = I$, $M = \mathrm{diag}(4,1)$, *and* $M = \mathrm{diag}(7 - 6.3|x|, 1)$.

Voronoi region [3, 32] is defined as

$$V_i = \{x \in \Omega \mid w_i d(x, Z_i) < w_j d(x, Z_j)\ \forall Z_j \neq Z_i\}$$

with d being the standard Euclidean distance and $\{w_i\}$ being a set of predetermined positive weights. As stated in [25], if we set $d_{Z_i}(x, Z_i) = w_i d(x, Z_i)$, then the MWVT is a special case of the AVR given in [25].

We are motivated by our interpretation of the space and distance distortion effected by the Riemannian metric tensor: to determine the membership of a point x in a tessellation, the distance and space distortion should be generally viewed through the Riemannian metric tensor at the point x rather than through the different tensors at the respective generators. That is, locally at a given point x, the space and distance distortion is fixed when measuring distances from it to different generators while the distortion is only sensed when varying the location x. It will be seen later that, in the context when the set of generators is viewed as variables of some optimization process, our definition leads to an optimization property. Moreover, the following simple proposition implies that our definition offers a consistent generalization to the standard VT in the isotropic metric.

PROPOSITION 2.1. *If the Riemannian metric tensor* $M(P)$ *is isotropic, that is,* $\lambda_1(P) = \lambda_2(P) = \rho(P)$, *or, equivalently,* $M(P) = \rho(P)I$ *at all points, with* I *being the identity and* $\rho = \rho(P)$ *being a scalar field, then the anisotropic Voronoi region reduces to the conventional Voronoi region in the Euclidean metric taking the form of a convex polyhedra.*

On the other hand, even for an isotropic Riemannian tensor, the AVR given in [25] does not reduce to the conventional Voronoi region unless the tensor is uniform, that is, $\rho(P)$ is a constant field. Of course, for the truly anisotropic case, an AVR defined in this paper may have very complicated geometry, and, in comparison, the definition in [25] leads to an AVR with a relatively less complicated boundary. In fact, the boundary curves of the AVRs there consist of piecewise quadratic curves (quadratic surfaces in higher dimension), and the AVRs can be viewed as the projection of the lower envelope of the lifted paraboloids $\{p_i(x) = d_{Z_i}(x, Z_i)\}$. In our case, the lifted functions become $\{p_i(x) = d_x(x, Z_i)\}$ whose graphs are generally not paraboloids. Note, however, that, unlike in our case, the bisectors of the generators in the AVR definition of [25] do not pass through the midpoints of the generators.

Even with our definition of AVRs and the dependence on the local Riemannian metric tensor, it remains simple to determine the membership of a given point in an AVT as the computation of the directional distance is very straightforward. Of

course, the bisector of any two points in the above anisotropic case may be very complicated for a general metric. It is generally impossible to solve the bisectors exactly, not to say with polynomial complexity. Thus, finding efficient algorithms for a direct construction of the AVT remains nontrivial. In practice, however, one is more interested in finding good approximations, so some methods for the detailed constructions and approximations of the AVT and the ADT are given in the following sections.

2.4. Anisotropic mass center (centroids). To define the *anisotropic mass center* for the AVR of a generator, we recall that the mass center of a standard Voronoi region with respect to the Euclidean metric is the point that minimizes the cost functional (the mean square Euclidean distance to such a point) in the given region. Thus, we first define the following *anisotropic cost functional*.

DEFINITION 2.4. *Let $V(P)$ be the* anisotropic Voronoi region *of point P; the following integral is called the* anisotropic cost functional *or* anisotropic energy functional *of $V(P)$:*

$$F(Y) = \int_{V(P)} d_X^2(X, Y)dX.$$

Here, different from [12, 13, 19], the density function is simplified to be the identity, as its role is often reflected in the underlying Riemannian metric. Also, the integral can be interpreted either as an integration with respect to the Riemannian metric [4, 22] or the usual Lebesgue integral corresponding to the standard Lebesgue measure. Of course, for a given problem, only one interpretation is allowed. The details are to be discussed in section 4.

Minimizing the above quadratic functional with respect to Y, we are led to the following.

DEFINITION 2.5. *The* anisotropic mass center *of $V(P)$ is given by*

$$Y_c = \Big[\int_{V(P)} M(X)dX\Big]^{-1}\Big[\int_{V(P)} (M(X), X)dX\Big].$$

Here, $(M(X), X)$ is the product of $M(X)$ and X; A^{-1} denotes the inverse of matrix A. It is easy to see that the following lemma is true.

LEMMA 2.2. *If $M = M(P)$ is a smoothly defined positive definite Riemannian metric, then $\int_{V(P)} M(X)dX$ is a positive definite matrix for any $V(P)$ with nonempty interior.*

Thus, the notion of the anisotropic mass center given earlier is well defined. Clearly, it is a natural generalization of the corresponding definition in the Euclidean case. Moreover, once the AVR is given, the explicit formula makes the computation of the mass centers fairly straightforward. Had the functional $F(Y)$ been replaced by

$$\hat{F}(Y) = \int_{V(P)} d_Y^2(X, Y)dX,$$

there would be more issues to be addressed in defining and computing the mass centers.

2.5. Anisotropic centroidal Voronoi tessellations. Once the AVRs and their mass centers are defined, the *anisotropic centroidal Voronoi tessellation* (ACVT) and its dual ACVDT can then be defined.

DEFINITION 2.6. *Given the set of points* $\{Z_i\}_{i=1}^n$ *in the domain* Ω *and a smoothly defined positive definite Riemannian metric* M *defined as above on* Ω*, a Voronoi tessellation under the directional distance function is called an* anisotropic centroidal Voronoi tessellation (ACVT) *if*

$$z_i = z_i^*, \quad i = 1, \ldots, k;$$

i.e., the points $\{z_i\}$ *that serve as generators for the anisotropic Voronoi regions* $\{V_i\}$ *are themselves the anisotropic mass centers (centroids) of those regions. The corresponding dual triangulation is referred to as the* anisotropic centroidal Voronoi Delaunay triangulation *(ACVDT).*

We first observe the following trivial but important fact.

PROPOSITION 2.2. *If the Riemannian metric tensor* $M(P)$ *is isotropic, that is,* $\lambda_1(P) = \lambda_2(P) = \rho(P)$ *for all points* P*, or, equivalently,* $M(P) = \rho(P)I$ *with* I *being the identity and* $\rho = \rho(P)$ *being a scalar positive density function, then the anisotropic centroidal Voronoi tessellation reduces to the standard centroidal Voronoi tessellation corresponding to the Euclidean distance and the density function* $\rho = \rho(P)$*.*

The above proposition concludes that our definition of ACVT is a consistent generalization of the conventional CVT. If other definitions of directional distances were used, such a conclusion may not hold.

Similar to the standard CVT, the above-defined ACVT enjoys the following optimality property.

PROPOSITION 2.3. *Given a compact set* $\Omega \subset \mathbb{R}^d$*, a positive integer* n*, and a smoothly defined positive definite Riemannian metric* $M(\cdot)$*, let* $\{Z_i\}_{i=1}^n$ *denote any set of* n *points belonging to* Ω*, and let* $\{V_i\}_{i=1}^n$ *denote any tessellation of* Ω *into* n *regions. Define the* energy (cost) functional *for* $\{(Z_i, V_i)\}_{i=1}^n$ *by*

$$\mathcal{F}(\{(Z_i, V_i)\}_{i=1}^n) = \sum_{i=1}^n \int_{X \in V_i} d_X^2(X, Z_i)\, dX. \tag{2.3}$$

A necessary condition for $\mathcal{F}$ *to be minimized is that the* V_i*'s are the anisotropic Voronoi regions corresponding to the* Z_i*'s, and, simultaneously, the* Z_i*'s are also the anisotropic mass centroids of the corresponding* V_i*'s; i.e.,* $\{(Z_i, V_i)\}_{i=1}^n$ *is an anisotropic centroidal Voronoi tessellation of* Ω*.*

The proof follows easily from the definitions of the AVR and the anisotropic mass centroids. The functional $\mathcal{F}$ is often also called *distortion value* or *total variance* in different applications. In practice, it can be related to the errors of numerical approximation, being errors of surface representation or errors of numerical solution of some partial differential equations [13].

2.6. Relation to function approximation. Given a function f smoothly defined in the compact set $\Omega \subset \mathbb{R}^d$, a set of n distinct points $\{Z_i\}_{i=1}^n$ in Ω, and a tessellation of Ω denoted by $\{V_i\}_{i=1}^n$, we consider how to approximate the function f by a piecewise constant function f_n defined by $f_n(x) = f(Z_i)\chi_i(x)$ for each i with χ_i being the characteristic function of the set V_i.

We may measure the error of the approximation by

$$\int_\Omega |f_n(X) - f(X)|^2\, dX = \sum_{i=1}^n \int_{V_i} |f_n(X) - f(Z_i)|^2\, dX,$$

which, by Taylor expansion, can be approximated to the leading order by

$$\int_\Omega |f_n(X) - f(X)|^2 dX \approx \sum_{i=1}^n \int_{V_i} \overrightarrow{XZ_i}^T \nabla f(X) \nabla^T f(X) \overrightarrow{XZ_i}\, dX$$

$$\leq \sum_{i=1}^n \int_{V_i} d_X^2(X, Z_i)\, dX, \tag{2.4}$$

where $d = d_X(X, Y)$ can be any directional distance between X and Y corresponding to a Riemannian tensor $M = M(X)$ that bounds $\nabla f \nabla^T f$ (e.g., $M = \alpha I + \nabla f \nabla^T f$ with a small constant $\alpha > 0$).

Clearly, if the elements of $\{Z_i, V_i\}_{i=1}^n$ are allowed to vary in order to reduce the error of approximation, we may choose to minimize the error bound given on the right-hand side of (2.4) which is precisely in the form of the functional $\mathcal{F}(\{(Z_i, V_i)\}_{i=1}^n)$. This observation, among others, is one of the motivations for the definition of the energy functional. Hence, we see that, under the above metric (related to ∇f), the ACVT defined in the paper turns to provide the optimal error bound on the piecewise constant approximations of the given function f. Historically, such an idea has been used by Thiessen in one of the earliest applications of the Voronoi diagrams to get estimation of precipitation data in a given geographical region [32].

The relevant metric is obviously dependent on the quantities to be approximated and on how the errors are measured. If one is interested in approximating the derivatives ∇f by a piecewise vector-valued function

$$\vec{v}_n(x) \approx \sum \nabla f(Z_i) \chi_i(x)$$

and estimating the errors with

$$\int_\Omega |\vec{v}_n(X) - \nabla f(X)|^2\, dX,$$

then the corresponding metric would be tied to the Hessian matrix of f. Of course, the approximation error considered here is very generic, and in practical applications such as the numerical solution of partial differential equations, more sophisticated error estimators are desirable in order to ensure the optimal resolution. Nevertheless, the above discussion provides a natural link between the problem of optimal tessellation and the problem of optimal function (as well as surfaces to be discussed later) representation and approximation.

2.7. On the construction of ACVT. The construction of ACVT can be done with several different methods. Here, we mainly discuss the Lloyd algorithm, which is a fixed-point iteration between the generators and the mass centers of the AVRs [11, 12].

Given the generators, as mentioned before, the precise determination of the AVRs and their boundary can be computationally cumbersome, and there is little control in theory on the complexity of such procedures. However, in many applications, the construction of ACVT can often be viewed as an optimization procedure; thus it is of practical interest to compute good approximations of ACVT. Due to the complexity of the AVRs and the existence of orphan regions, we make an effective approximation in our construction of the ACVT: computing the visibility regions for the generators and their mass centroids instead of the full AVRs. The generation of the visibility

regions is helped by the classical ADT constructed by methods introduced in [4, 6, 22] (here we use the term *classical* to differentiate from the definition of ADT given in this paper).

In the next section, a systematic description of the algorithm for two-dimensional classical ADT is provided; then, in section 4, the detailed description of the ACVT construction is provided.

3. Classical ADT. In this section, we recall the technique of two-dimensional classical ADT for any planar domain Ω, with respect to a given Riemannian structure $(\Omega, M(P)_{P\in\Omega})$ defined in Ω [4, 6, 22]. The concepts of Delaunay measure and the generalized constrained Delaunay kernel are briefly reviewed; then, the traditional anisotropic two-dimensional Delaunay triangulation and the so-called unit mesh generation procedure are discussed.

3.1. Generalized constrained Delaunay kernel. The constrained Delaunay kernel, including the constructions of Base, Cavity, and Ball, is a procedure for inserting a new point into an existing Delaunay triangulation; see, e.g., [5, 6, 8, 21, 22]. The Delaunay criterion for insertion is based on the empty-circle (sphere for three dimensions) test. When adding a point (say P), the mesh updating kernel takes on the form $T = T - C(P) + B(P)$, where $C(P)$ is the Cavity associated with P, i.e., the set of existing triangles whose circumdiscs contain P, and $B(P)$ is the local updated triangulation of $C(P)$ enclosing P as a vertex, which is called the Ball of P; T denotes both the existing and newly updated Delaunay meshes. The Cavity can be constructed by either recursive neighboring searching or other methods [5, 6, 8, 21, 22], based on a proximity criterion, i.e., the empty-circle test.

The classical proximity criterion is not applicable to the anisotropic cases, and a generalized kernel definition for the Riemannian metric is needed. The generalization consists mainly of a redefinition of the Cavity $C(P)$. The Ball $B(P)$ can be constructed similarly as in the classical case, provided that the Cavity is star-shaped to P.

First, we recall the *Delaunay measures* α_M with respect to a given metric M associated with the pair (P, k), where P is a vertex and k a triangle (detailed discussions may be found in [6, 22]). The measures α_M vary with a reference point Q, which is taken to be either P or any of the three vertices of k. First, let O_Q and r_Q be the circum-disc center and the radius of k with respect to the (uniform) tensor $M = M(Q)$, i.e., $\overrightarrow{vO_Q}^T M(Q)\overrightarrow{vO_Q} = r_Q^2$, for any vertex v of k. Then

$$\alpha_M(Q)(P,k) = \sqrt{\overrightarrow{PO_Q}^T M(Q)\overrightarrow{PO_Q}}/r_Q,$$

which means that for each pair (P, k), four such α's may be defined [6, 22]. Then, the Cavity $C(P)$ is redefined as

$$C(P) = C_1(P)\bigcup C_2(P),$$

where

$$C_1(P) = \{k \in T \mid k \text{ contains } P\},$$

i.e., the Base of P; and

$$C_2(P) = \{k \in T \mid \text{if } \exists\, K^* \in C(P), k \text{ is adjacent to } K^*, \text{ and } \alpha_{M(P)}(P,k)\prod_v \alpha_{M(v)}(P,k) < 1,\ v \text{ is any vertex of } k \text{ and } P \text{ is visible from the vertices of } k\}.$$

Other variants of the definition of $C_2(P)$ can also be found in [6, 22]. The above $C(P)$ can be constructed via the recursive adjacent search, beginning from the Base of P, i.e., $C_1(P)$. Like the classical cases, this generalized Cavity is star-shaped with respect to P, and, accordingly, the updated triangulation remains a valid triangulation. The generalized Delaunay kernel still retains the form $T = T - C(P) + B(P)$ but with a generalized Cavity $C(P)$ defined as above.

3.2. Unit Delaunay meshing. The construction of the unit Delaunay mesh consists of meshing the domain Ω with mesh edges being as close to the unit length size as possible. Here, the edge length is evaluated with respect to the given Riemannian metric M, and it can be obtained by numerical integration. That is, the goal is to obtain a unit mesh with respect to M and such that every mesh edge PX connecting P satisfies $L_M(\overrightarrow{PX}) \approx 1$ [6, 22].

The unit meshing process of Ω, with respect to (Ω, M) involves two steps:

1. The generation of the boundary unit mesh.
2. The generation of the unit mesh of Ω based on the boundary unit mesh.

For completeness, in the following, we briefly describe the main features of the above two steps as given in [6, 22].

Unit meshing of the boundary. Assume that the input of the boundary consists of a set of curved segments; a unit mesh of the boundary with respect to the given Riemannian metric field is a discretization of these curved segments such that the obtained segments have unit length with respect to the control space or the Riemannian metric. Several methods have been discussed in [6, 22]. Here, a simple approach is adopted: apply the adaptive Simpson quadrature formula to subdivide each curved segment into a set of smaller curved segments with lengths less than a given threshold δ (usually we let it be 0.1 for accuracy); then approximate the length of curved segment by summing up the lengths of all these smaller segments and follow the approach in [6, 22] to get the final unit-length subdivisions of the curved segment.

Unit meshing of the domain. The unit meshing of the boundary of Ω described in the above provides a set of constrained edges having a set of end points, denoted by $BP(\Omega)$. First, an initial constrained Delaunay mesh of Ω is generated whose vertices are only the members of $BP(\Omega)$ with the constrained edges being preserved. This so-called empty mesh of Ω can be generated using the classical Delaunay kernel [6, 22]. Then, a new mesh is constructed by inserting iteratively field points into this empty mesh as follows:

(a) Generate interior field points by the method of interior edges unit subdivision or by the method of advancing front points generation [7, 21, 6, 22]; then, apply filtering to the newly generated points so that the distance measured in the Riemannian metric between any two points is no less than a given threshold (say, $\sqrt{2}/2$).
(b) The remaining generated field points after the filtering are inserted into the current mesh via the generalized constrained Delaunay kernel defined before.

The above iteration is repeated as long as the current mesh can be modified. The iteration is terminated if no new field points get generated. The generated mesh has the following property: most of the edges are with lengths (with respect to the given Riemannian metric) close to 1.0. However, some edges may not have ideal lengths, and an optimization of the mesh is often performed through both edges swapping and optimal vertex smoothing [6, 22] to make the resulting edges much closer to the unit length.

The mesh generated in the above method is in good conformity with the given Riemannian metric, and it is used as a good approximation to the dual of the generalized AVT.

4. An algorithm for approximating the ACVT. Next, the Lloyd iteration method is extended to construct the ACVT defined in section 2. The complete process goes as follows:

0. Construct an initial constrained ADT of the given domain in the method described in section 3. The triangulation is close to having a unit mesh with respect to the given Riemannian metric field M. Store all the data of the constrained boundary triangulation, i.e., the empty mesh.
1. Construct the approximate AVR for each interior point that is allowed to change its position and compute the approximate mass center of each Voronoi region. Here, the AVT and the mass centers are as defined in section 2. The approximate construction of the AVT is done using the existing anisotropic Voronoi triangulation as an auxiliary tool.
2. Insert the computed mass centers into the stored empty mesh by the generalized Delaunay insertion procedure described in section 3 and construct an updated ADT.
3. Compute the difference $D = \sum_{i=1}^{k} |P_i - P_{ic}|^2$: $\{P_i\}$ is the set of interior points allowed to change; $\{P_{ic}\}$ is the set of corresponding computed mass centroids.
4. If D is less than a given tolerance, terminate; otherwise, return to step 1.

Naturally, other stopping criteria can also be applied, such as using the measurement on the reduction of the energy functional.

The success of the above algorithm relies on the resolution of two issues: first, the efficient approximation of AVR with the help of the constructed or updated ADT and then the approximation of the mass center for each AVR. We address these two critical issues next.

4.1. The approximation of AVR. For the classic (isotropic) Delaunay triangulation, the Voronoi region of each interior point can be computed easily, as the relationship between the VT and the Delaunay triangulation is straightforward [32, 2, 10]. But, for the AVT and the ADT, due to the generality and the complexity of the metric and the derived distance function, it is difficult to characterize their duality in simple terms. Hence, numerical approximations are needed based on the definitions given in section 2.

Recall that for any interior point P, under the Riemannian metric $M = M(P)$, its AVR defined in section 2 is given by

$$V(P) = \{Q \in \Omega \mid d_Q(Q, P) < d_Q(Q, P_i), Q \in \Omega, P_i \neq P, P_i \in VT\},$$

where VT is the set of *input generators*. To compute $V(P)$, two special sets are defined. The first set, denoted by Ω^*, is the *testing region*, which basically can be any region that contains $V(P)$. In another word, only the points in $\Omega^*(P)$ can possibly be in $V(P)$. It is obvious from an algorithmic point of view that the smaller the $\Omega^*(P)$, the more efficient the determination of $V(P)$ becomes. The second set $V^*(P)$ is called the set of *testing vertices* defined by

$$V^*(P) = \{P^* \in VT \mid \exists\, Q \in \Omega^*(P),\ d_Q(Q, P^*) < d_Q(Q, P) \text{ and } P^* \neq P\}.$$

Clearly, $V^*(P)$ is dependent on the testing region $\Omega^*(P)$.

The following proposition can be easily verified.

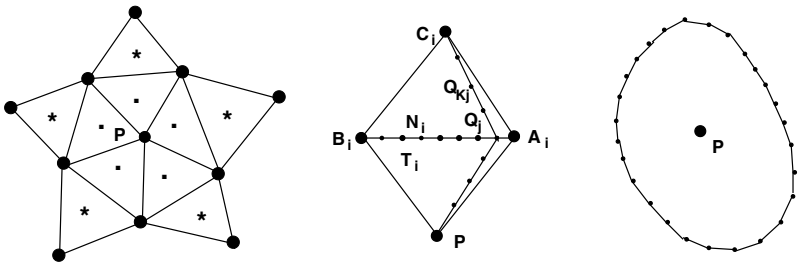

FIG. 4.1. *Approximation of the AVR.*

PROPOSITION 4.1. *The anisotropic Voronoi region of P can be redefined as*

$$V(P) = \{Q \mid d_Q(Q,P) < d_Q(Q,P_i),\ Q \in \Omega^*(P),\ P_i \neq P,\ P_i \in V^*(P)\}. \tag{4.1}$$

Computationally, it is important to define appropriate sets $\Omega^*(P)$ and $V^*(P)$ in order to construct the AVRs efficiently. The approach adopted by us is to define $\Omega^*(P)$ and $V^*(P)$ with the help of the traditional ADT discussed earlier.

Denote the triangles connecting P by $B^*(P)$, and define the set $C^*(P)$ as

$$C^*(P) = \{t \mid \exists\, t^* \in B^*(P), t \text{ connects } t^* \text{ by an edge of } t^*\}.$$

$C^*(P)$ includes the triangles which are neighbors of $B^*(P)$ sharing common edges. In Figure 4.1, $B^*(P)$ is the set of dotted triangles and $C^*(P)$ is the set of triangles marked with stars. We then choose $B^*(P) \bigcup C^*(P)$ for $\Omega^*(P)$.

Let $\bar{V}$ contain the vertices of the triangles $B^*(P) \bigcup C^*(P)$ except P. In Figure 4.1, $\bar{V}$ is the set of circled vertices. $\bar{V}$ is then chosen as $V^*(P)$.

Intuitively, such a choice implies to some degree that, instead of constructing the full AVR, we try to identify the visibility region of the corresponding generator in the AVR. Such a region is inside the main subregion of the AVR and is also star-shaped. The approximate visibility regions are taken as the approximate AVRs and are also used in the computation of the mass centroids. Though orphan regions cannot be eliminated, for practical applications and for sufficient large number of generators, the visibility regions computed served as good approximations to the whole AVRs. Indeed, our numerical examples show that the choices for $V^*(P)$ and $\Omega^*(P)$ are well suited for the approximation of $V(P)$.

With the above preparations, we can describe the approximate construction of the AVR $V(P)$ restricted by (4.1). The procedure in the pseudocode form is given as follows:

(a) Find the sets of triangles $B^*(P)$ and $C^*(P)$, and order them CCW (counterclockwise). Denote the degree of $B^*(P)$ by N_T. The number of elements of $C^*(P)$ is equal to that of $B^*(P)$, except near the boundary.

(b) Set $\mathrm{VB}(P) = \phi$.

(c) For $i = 1$ to N_T, do the following:
denote the ith triangle in $B^*(P)$ and $C^*(P)$ by T_i and N_i, respectively;
denote the other three vertices by A_i, B_i, C_i (see Figure 4.1);
divide the edge A_iB_i into N_D equal subedges and denote their end points in order as $\{Q_j, j = 0, \ldots, N_D\}$ where $Q_0 = A_i$.
For $j = 0$ to $N_D - 1$, do the following:

connect PQ_j and C_iQ_j, and divide the poly-segment PQ_jC_i into M_D subsegments; denote the interior division points in CCW order as $\{Q_{kj}, k = 1, \ldots, M_D, Q_{1j} = P\}$.
For $k = 2$ to M_D, do the following:
Compare the distance $d_{Q_{kj}}(Q_{kj}, P)$ with $\{d_{Q_{kj}}(Q_{kj}, q), q \in V^*(P)\}$.
If there exists $\widehat{Q} \in V^*(P)$,$d_{Q_{kj}}(Q_{kj}, P) > d_{Q_{kj}}(Q_{kj}, \widehat{Q})$, then
VB(P) = VB$(P) \bigcup \{Q_{kj}\}$
Exit this inner loop; move to the next poly-segment.
Endif
End Do
End Do
End Do

(d) As the points in VB(P) are in CCW order, we can connect them in such an order and get a closed polygon denoted by $\widehat{V(P)}$ as shown in Figure 4.1. Then, $\widehat{V(P)}$ is an approximation for $V(P)$.

From the above construction, it is obvious that $\widehat{V(P)}$ produced by the algorithm is star-shaped with respect to P. It is taken as an approximation of the visibility region and in fact the AVR corresponding to P.

Note that due to Lemma 2.1, we actually can shrink the domain $C^*(P)$ even more by considering the convex hull formed by the vertices of $B^*(P)$ (except P) and the midpoints of P and the vertices in $C^*(P)$. The rest of the algorithm is similar to the one given above. The resulting test regions and testing vertices still offer good approximations. Obviously, if M_D and N_D go to infinity, the approximation error goes to zero. In practice, we set them to be around 10, and various numerical examples show that such a choice is enough to achieve good accuracy.

With the AVR $V(P)$ being approximated as above, the computation of the mass center of $V(P)$ defined in section 2 is discussed next.

4.2. Computation of mass centers. Naturally, with the AVR $V(P)$ being approximately decomposed into $N_{TD} = N_T \cdot N_D$ triangles of the form $\{\triangle PQ_i^*Q_{i+1}^*, Q_i^* \in VB(P), i = 1, \ldots, N_{TD}\}$, the mass center of $V(P)$ may be expressed as

$$\left[\int_{V(P)} M(Y)dY\right]^{-1}\left[\int_{V(P)} (M(Y), Y)dY\right] \approx \left[\sum_{i=1}^{N_{TD}} \int_{\triangle_i} M(Y)dY\right]^{-1}\left[\sum_{i=1}^{N_{TD}} \int_{\triangle_i} (M(Y), Y)dY\right],$$

where $\triangle_i$ is the triangle $\triangle PQ_i^*Q_{i+1}^*$.

Now, the remaining issue is how to compute $\int_{\triangle_i}(M(Y), Y)dY$ and $\int_{\triangle_i} M(Y)dY$ through numerical quadratures.

The area of a triangle $\triangle ABC$ under a given Riemann metric M can be approximated by

$$\left[\int_{\triangle ABC} ds\right] \approx \sqrt{\det(M)}S(\triangle ABC),$$

where $S(\triangle ABC)$ is the triangle area in the usual Euclidean metric [4, 22, 20] and $\det(M)$ is the determinant of the metric tensor M at the geometric center of $\triangle ABC$.

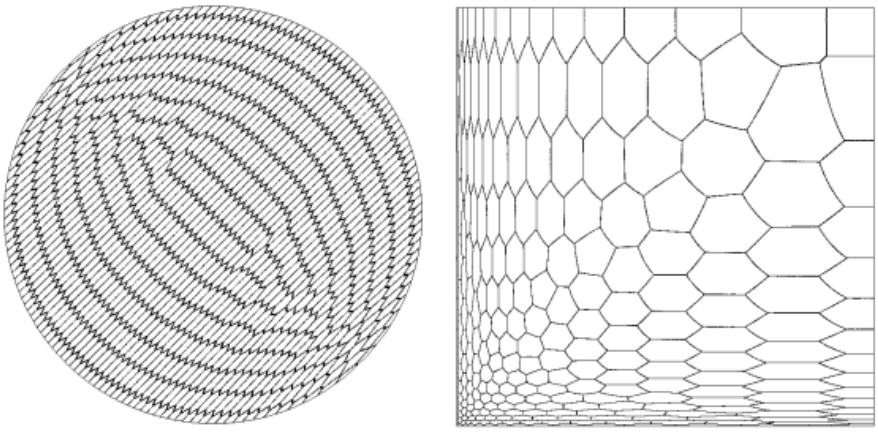

FIG. 4.2. *ACVTs in a disc with a uniform metric and in a square with a nonuniform metric.*

For standard Lebesgue measures and integrations, the above quadrature is simplified by taking $\det(M) = 1$.

To approximate the integral $[\int_{\triangle_i}(M(Y), Y)dY]$ (or $[\int_{\triangle_i} M(Y)dY]$), we divide the triangle $\triangle_i$ into m_i subtriangles. The division is made along the direction of the radius $\overrightarrow{PQ_i^*}$, since the triangle $\triangle PQ_i^*Q_{i+1}^*$ is long only in the $\overrightarrow{PQ_i^*}$ direction and thin in other directions. Then, $[\int_{\triangle_i}(M(Y), Y)dY]$ can be approximated by

$$\begin{aligned}\int_{\triangle_i}(M(Y), Y)dY &= \sum_{j=1}^{m_i}\left[\int_{\triangle_{ij}}(M(Y), Y)dY\right] \\ &\approx \sum_{j=1}^{m_i}\left[\sqrt{\det(M_{ij})}(M(Y_{ij}), Y_{ij})S(\triangle_{ij})\right],\end{aligned}$$

where Y_{ij} is the geometric centroid of $\triangle_{ij}$, M_{ij} is the metric of Y_{ij}, and $S(\triangle_{ij})$ is the Euclidean area of $\triangle_{ij}$. A similar formula can be given for $[\int_{\triangle_i} M(Y)dY]$.

Finally, the mass center Y of $V(P)$ can be approximated by

$$Y \approx \left[\sum_{i=1}^{N_{TD}}\sum_{j=1}^{m_i}\sqrt{\det(M_{ij})}M_{i,j}S(\triangle_{ij})\right]^{-1}\left[\sum_{i=1}^{N_{TD}}\sum_{j=1}^{m_i}\sqrt{\det(M_{ij})}(M_{ij}, Y_{ij})S(\triangle_{ij})\right].$$

Again, if we use the standard Lebesgue measures and integrations in defining the mass centers, then the above quadrature is applied with $\det(M) = 1$. When computing the mass centers, similar to that in [13], if a quadrature point is outside the given domain, its contribution is not included. For the examples given in the paper, in the construction of ACVT, the boundary points are fixed. In more general settings, we can apply the projection and merging techniques developed in [13, 19] to allow redistributions of boundary points while conforming to boundary constraints. For a two-dimensional domain with a simple boundary, the vertices of the unit boundary mesh are often well positioned with respect to the given Riemannian metric. In this simple case, there is seldom need for using the projection and merging techniques.

To end this section, we give a couple of numerical examples in Figure 4.2 for the approximations of ACVTs computed by our approach: one is for a circle of radius

10 with a constant metric given by $\theta = \pi/4$, $h_1 = 1.5$, and $h_2 = 0.3$; and the other is for a quadrilateral domain $[0, 4]^2$ corresponding to the metric with $\theta = 0$, $h_1(x, y) = 0.031 + x/4.0$, and $h_2(x, y) = 0.031 + y/4.0$. The resulting ACVTs clearly reflect the features of their respective metrics.

5. Application of ACVTs. Through a series of papers [13, 14, 15, 16, 17, 18, 19], it has been demonstrated that the CVT and their variants such as those satisfying certain constraints are very useful tools in mesh generation and optimization, surface representation, and other applications.

For instance, the CVDT concept provides a good theoretical explanation to the mesh smoothing process: by successively moving generators to the mass centers (of the Voronoi regions), the cost functional as defined in (2.1) is reduced. Here, smoothing means both node movement and node reconnection. With a suitable choice of the density function, the cost functional may be assumed to be related to the distortion of element quality and sizing conformity. The resulting CVDT turns out to have minimum distortion of element quality and sizing conformity, i.e., provides optimal mesh quality. Furthermore, for the multidimensional CVT, the celebrated Gersho's conjecture [23] implies that asymptotically, as the number of generators becomes large, the total cost (error) is approximately equipartitioned by congruent Voronoi cells. It was shown that the congruent cells are regular hexagons in two dimensions, with its dual Delaunay triangulation consisting of equilateral triangles. This gives strong substantiation to the claim that CVDT provides a high-quality mesh.

Based on these properties of CVDT, we have generated high-quality Delaunay triangular mesh and tetrahedral mesh through the construction of CVDT [13, 19]. The density function $\rho = \rho(P)$ is generally related to the sizing function $H(P)$ by $\rho(P) = c\,H(P)^{-2-d}$, where c is a scaling constant and d is the space dimensions.

As the anisotropy often appears in many practical problems, it is natural to expect broad applications of the ACVTs studied in this paper. In this section, a few of these representative applications are briefly outlined.

5.1. Optimization for two-dimensional ADT. Along with the constructed ACVT, its dual tessellation, the ACVDT, is obtained as a by-product of the Lloyd iteration. Similar to the conventional CVT case, the construction of ACVDT through the Lloyd iteration can be viewed from a different angle as a smoothing and optimization process of an initial mesh [19].

For a two-dimensional initial ADT, it is desirable that the resulting mesh has the unit mesh edge property, or, equivalently speaking, the mesh conforms with unit sizing in the Riemannian metric. This property guarantees the optimal element quality as measured using the Riemannian metric. However, in most cases, the numerical data on the edge lengths indicate that it is very hard to achieve near perfect unit sizing, even with a combination of postprocessing and optimization techniques such as edge swappings. Thus, the construction of two-dimensional ACVT and ACVDT provides a natural optimization for the initial two-dimensional ADT, similar to the two-dimensional and three-dimensional isotropic cases considered in [13, 19]. The final ACVDT gives a high-quality anisotropic Delaunay mesh.

As numerical examples, we first present the ACVT and ACVDT for a parallelogram domain with a constant metric that corresponds to $\theta = 0.0$, $h_1 = 2.0$, and $h_2 = 0.5$. To illustrate the optimization effect of the method, a points distribution is generated with half of them clustered to the center point of the domain. Then, 30 Lloyd iterations are applied to generate the approximate ACVT and its dual ACVDT. The AVTs and ADTs corresponding to the initial stage, after 5 and 30 Lloyd itera-

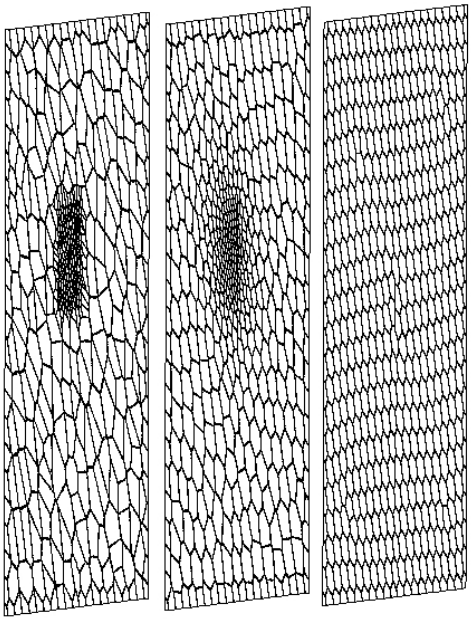

FIG. 5.1. *Iterations of AVTs for a parallelogram with uniform metric.*

tions, are shown in Figures 5.1 and 5.2.

Obviously, the ADT is more and more in conformity with the metric, and it gradually evolves into a high-quality mesh. To measure the anisotropic triangular element quality, the formulae discussed in [6, 22, 4] are used. The average values of element quality for these three meshes are $0.432, 0.852, 0.987$, respectively. The minimum values of the element quality for these three meshes are $0.0003, 0.131, 0.852$. The average values of the edge length measured under the given metric for the three meshes are $0.234, 0.422, 0.972$; that is to say, they become closer and closer to unit edge length. These element-quality and edge-length statistics show that our method for the ADT optimization is effective.

Another example (shown in Figure 5.3) is for a square domain $[0, 10]^2$ with a given nonuniform metric corresponding to $\theta = 0$, $h_1(x, y) = |y - 5|/5 + 0.04$, and $h_2(x, y) = \big|5 - 4|y - 5|/5\big| \cdot h_1(x, y)$. This means that a line refinement at $y = 5$ is to be introduced. Different from example 1, we start with an almost even initial point distribution which gives an ADT badly conformed with the given metric. The results of the initial configurations and those after 10 and 60 Lloyds iterations are shown in Figure 5.3. With 60 iterations, the final approximate ACVDT becomes very well conformed to the given metric. The average element quality is up to 0.982, and the minimum quality is 0.781. The average edge length is 1.15. This further shows that the optimization is also effective in nonuniform metric cases.

In the above, we have not performed any classical optimizations for the initial meshes, such as edges swapping and optimal vertex repositioning, before the construction of ACVDT. Of course, they can be combined with the Lloyd iteration to speed up the construction of ACVDT, an idea to be explored in our future works.

5.2. Application of ACVT to quality surface meshing. There are many instances in which optimal points distributions lying on surfaces or triangulated

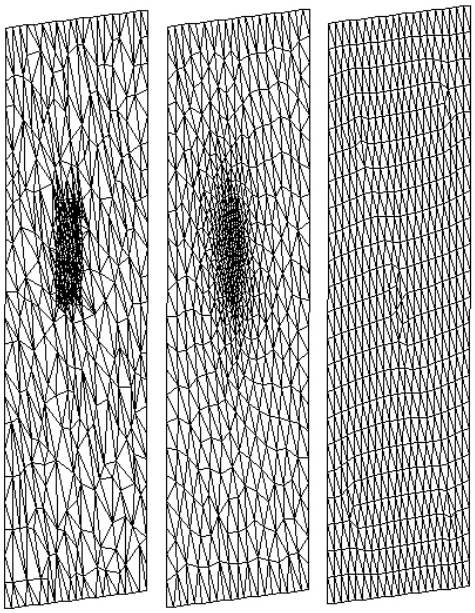

FIG. 5.2. *Corresponding dual ADTs for a parallelogram with uniform metric.*

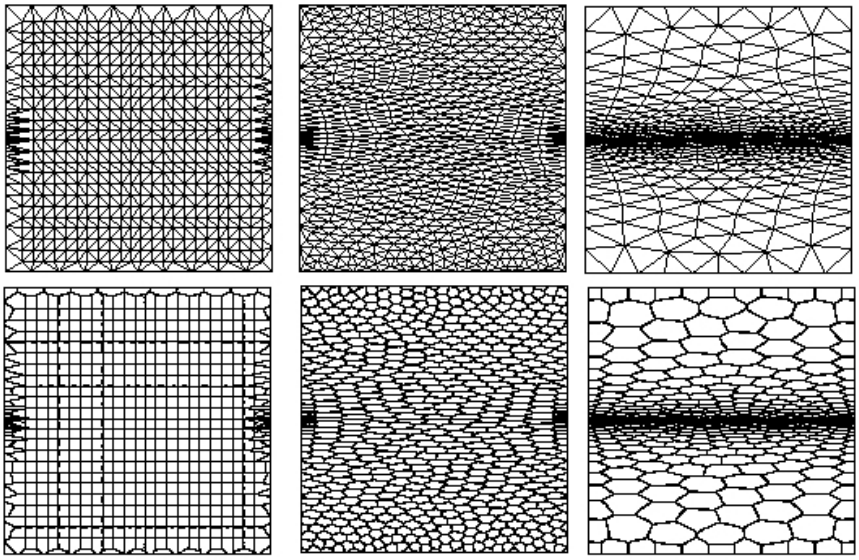

FIG. 5.3. *Optimizing ADT example* 2 *with nonuniform metric.*

surfaces and surface meshing or more general subdivisions of surfaces are needed. Examples include spherical meshes for geophysical calculations on the surface of the earth, collocation or nodal points for boundary finite element methods, and the geometric representation of surfaces by panels or other simple objects. In [14], a precise definition of the CCVT on general surfaces is given and a number of properties

are derived, including their characterization as minimizers of an "energy," and the generated optimal CCVT point sets are applied to polynomial interpolation and numerical integration on the sphere. The CCVT defined on general surfaces mainly involves constructing the CVT with respect to the isotropic Euclidean metric but constraining the mass centers on the surfaces. Thus, the distance used does not reflect the general landscape of the surfaces which are best described by Riemannian metrics. A direct application of the ACVT is to construct CVT on general surfaces which would naturally incorporate the information of the Riemannian metric associated with the surface.

Let Ψ be a parametric surface defined by $\sigma : (u, v) \in \Omega \to \sigma(u, v) \in \Psi$ with Ω being a planar domain, and let σ be a function of class C^2 (it is sufficient to have just class C^1). It is assumed that Ω is closed and bounded. Let M_{3D} be a three-dimensional metric tensor which defines directional sizings requirement on the surface Ψ. Then, for a given point (u, v) of Ω, i.e., the two-dimensional parametric domain, a Riemannian metric $M^*_{2D}(u, v)$ can be deduced as

$$M^*_{2D}(u, v) = (\sigma_u, \sigma_v)^T M_{3D}(\sigma_u, \sigma_v),$$

where the vectors σ_u and σ_v are the first-order derivatives or tangents to the constant parameter lines on the surface.

If $M_{3D} = H(P)^{-2} I$, this is the so-called isotropic controlling metric, which means that at each point P of Ψ, a length size $H(P)$ is specified which indicates that the ideal distance between other points and P is $H(P)$ in every direction [7, 22, 26, 27]; otherwise, the surface mesh will be anisotropic [7, 20, 22]. There are two typical types of M_{3D} for anisotropic cases. The first is

$$M_{3D}(P) = \mathrm{diag}(1/H_1(P)^2, 1/H_2(P)^2, 1/H_3(P)^2), \tag{5.1}$$

where $\{H_i(P)\}$ specifies the sizings in the coordinate directions. The second is for the geometric surface meshing where the metric tensor is related to the curvatures of the surface. In this case, M_{3D} takes the form

$$M_{3D}(P) = [\overrightarrow{v_{\min}}, \overrightarrow{v_{\max}}, \vec{n}]\mathrm{diag}\left(\frac{K^2_{\min}}{g(\epsilon)^2}, \frac{K^2_{\max}}{g(\epsilon)^2}, L\right)[\overrightarrow{v_{\min}}, \overrightarrow{v_{\max}}, \vec{n}]^T, \tag{5.2}$$

where $K_{\max}$ and $K_{\min}$ are the principal curvatures at P, $\vec{v}_{\max}$ and $\vec{v}_{\min}$ are the two corresponding principal directions, and $\vec{n}$ is the unit normal direction, all evaluated at the point P. Here, $g = g(\epsilon)$ is an auxiliary function of the tolerance ϵ used in the approximation, and L can be any constant; see [7, 20, 26, 27] for details of the above definitions. The metric (5.2) is used to prescribe the geometric anisotropic meshing.

In [7, 22], the above Riemannian metric M^*_{2D} is used to generate a unit anisotropic mesh on Ω and subsequently map it to generate a surface mesh on Ψ. It is very natural for us to use M^*_{2D} to construct an ACVT on Ω by the method given in section 4; then the ACVT is mapped back to Ψ. The resulting tessellation on Ψ is accordingly a CVT on the surface well conformed with the prescribed metric: it is either isotropic or anisotropic. The corresponding ACVDT provides a high-quality surface Delaunay triangular mesh on Ψ measured under the given metric tensor; see [20] for details of surface mesh evaluation.

To illustrate this application, three isotropic examples are presented here: ACVT and ACVDT on an ellipsoid, a paraboloid, and a torus. For each surface, a nonuniform sizing with either a point, a line, or a circle refinement is considered. That is, more

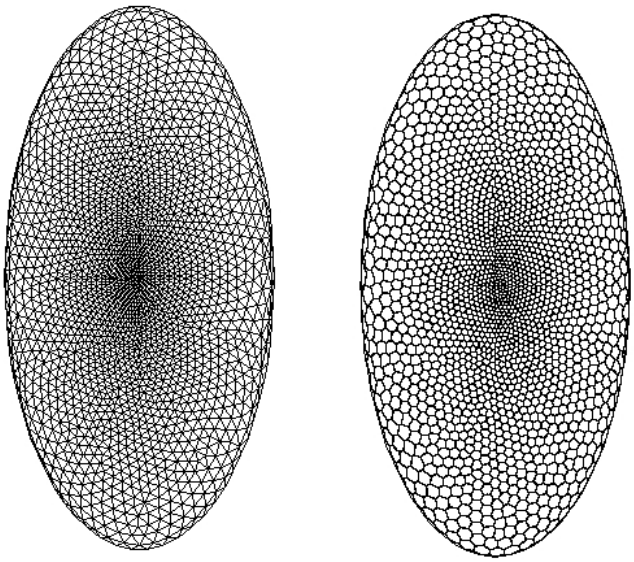

FIG. 5.4. *ACVDT and ACVT on an ellipsoid: with point refinement, top view.*

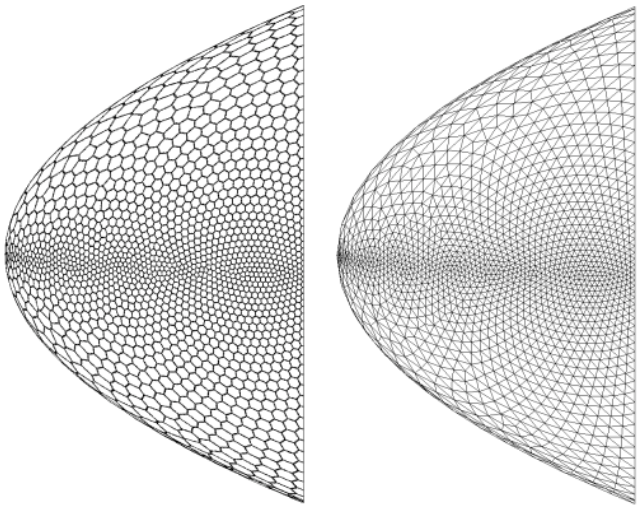

FIG. 5.5. *ACVT and ACVDT on a paraboloid: with line refinement, side view.*

refined meshes are generated near those objects. The results are shown in Figures 5.4, 5.5, and 5.6, respectively. An anisotropic example is also given for the ACVT and ACVDT on a saddle surface $z = 2x^2 - 4y^2$; the results are given in Figures 5.7 and 5.8.

In the isotropic cases, the average elements quality of the three ACVDTs are $0.984, 0.979, 0.978$, respectively; and the minimum elements quality are all above 0.50. Here, the surface element quality measure is in the range $[0, 1]$ as defined in [20], with 1 being the optimal value. These statistics demonstrate that the generated ACVTs on the surfaces provide well-distributed vertices, which in turn produces high-quality isotropic triangular surface meshes.

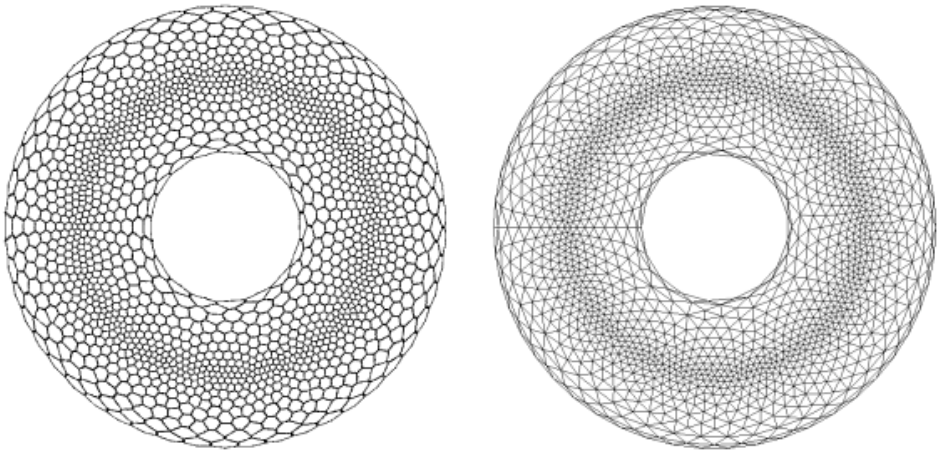

FIG. 5.6. *ACVT and CVDT on a torus: with circle refinement, top view.*

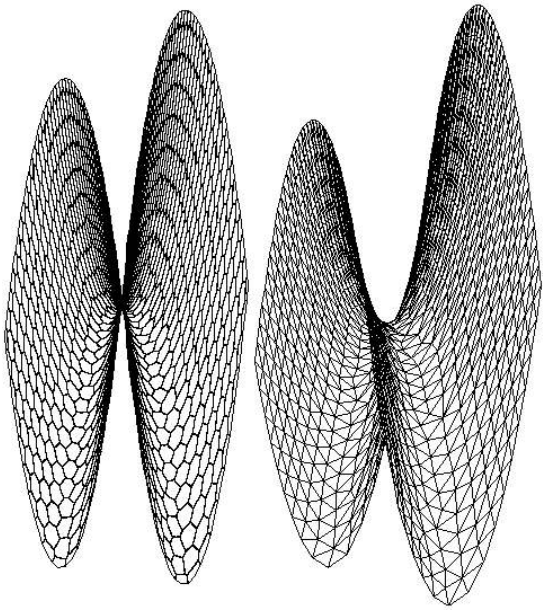

FIG. 5.7. *ACVT and CVDT on a saddle: with the metric tensor of type* (5.1).

In the saddle surface example, two cases are considered: one of type (5.1) with $H_1 : H_2 : H_3 = 1 : 10 : 1$ and the other based on a curvature-related metric of type (5.2). The ACVTs and ACVDTs of the two cases are shown in Figures 5.7 and 5.8, and the average elements quality of the two ACVDTs are 0.971 and 0.968, respectively; also, both have minimum element qualities above 0.45. These elements quality statistics are measured under the given metric, and they demonstrate that our proposed method is indeed effective in anisotropic cases.

6. Conclusion and future work. In this paper, a novel definition of ACVTs with respect to a given Riemannian metric is introduced. It is a consistent general-

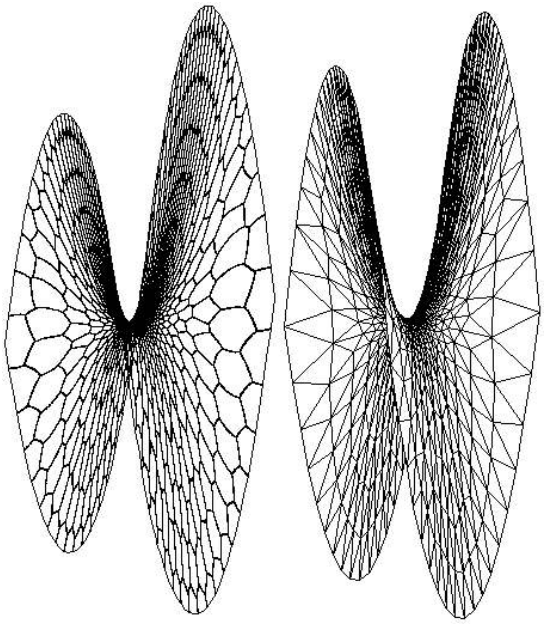

FIG. 5.8. *ACVT and CVDT on a saddle: with a curvature controlled tensor of type* (5.2).

ization of the standard CVTs in the Euclidean metric, and the ACVTs share similar optimization and best approximation properties as their conventional CVT counterparts in the Euclidean metric which make them valuable for high-quality mesh generation and optimization. Comparisons with related concepts previously studied in the literature are also made.

To approximate the ACVT, we apply the Lloyd iterations to move the generators to their desired positions. An effective and efficient approximation of the AVR is proposed using the ADT governed by the Riemannian metric. The ACVT is then applied to the two-dimensional anisotropic Delaunay mesh optimization and to quality meshing on general surfaces. Numerical examples demonstrate that the proposed definition of ACVT and its construction are computationally effective and lead to successful applications.

There are a number of issues to be studied further. In particular, more precise theoretical analysis are needed in identifying conditions on the metric, geometry, and the generator distributions that guarantee the well-defined duality between the AVTs and the ADTs. Systematic studies on the errors associated with the algorithms proposed here for the approximate constructions of the ACVTs are also worthwhile. Computationally, the acceleration of the Lloyd iteration and various other complexity issues remain to be investigated. The three-dimensional realizations may also be explored. In addition, we are also working on the applications of the ACVTs and the corresponding ACVDT meshes to numerical partial differential equations in both Euclidean spaces and on Riemannian manifolds [18], especially in the context of adaptive algorithms.

Acknowledgments. The authors would like to thank the referees for their valuable comments and suggestions that helped us to improve the presentation of the paper.

REFERENCES

[1] N. Amenta, M. Bern, and D. Eppstein, *Optimal point placement for mesh smoothing*, J. Algorithms, 30 (1999), pp. 302–322.

[2] F. Aurenhammer, *Voronoi diagrams—a survey of a fundamental geometric data structure*, Comput. Surveys, 23 (1991), pp. 345–405.

[3] F. Aurenhammer and H. Edelsbrunner, *An optimal algorithm for constructing the weighted Voronoi diagram*, Pattern Recognition, 17 (1984), pp. 251–257.

[4] H. Borouchaki and P. Frey, *Adaptive triangular-quadrilateral mesh generation*, Internat. J. Numer. Methods Engrg., 41 (1998), pp. 915–934.

[5] H. Borouchaki and P. George, *Optimal Delaunay point insertion*, Internat. J. Numer. Methods Engrg., 39 (1996), pp. 3407–3437.

[6] H. Borouchaki, P. George, F. Hecht, P. Laug, and E. Saltel, *Delaunay mesh generation governed by metric specifications. Part* I. *Algorithms*, Finite Elem. Anal. Des., 25 (1997), pp. 61–83.

[7] H. Borouchaki, P. Laug, and P. George, *Parametric surface meshing using a combined advancing-front generalized Delaunay approach*, Internat. J. Numer. Methods Engrg., 49 (2000), pp. 233–259.

[8] H. Borouchaki and S. Lo, *Fast Delaunay triangulation in three dimensions*, Comput. Methods Appl. Mech. Engrg., 128 (1995), pp. 153–167.

[9] F. Bossen and P. Heckbert, *A pliant method for anisotropic mesh generation*, in Proceedings of the Fifth International Meshing Roundtable, Sandia National Laboratories, Albuquerque, NM, 1996, pp. 63–74. Available online at http://www.andrew.cmu.edu/user/sowen/imr5.html

[10] A. Bowyer, *Computing Dirichlet tessellations*, Comput. J., 24 (1981), pp. 162–166.

[11] G. Buscaglia and E. Dari, *Anisotropic mesh optimization and its application in adaptivity*, Internat. J. Numer. Methods Engrg., 40 (1997), pp. 4119–4136.

[12] Q. Du, V. Faber, and M. Gunzburger, *Centroidal Voronoi tessellations: Applications and algorithms*, SIAM Rev., 41 (1999), pp. 637–676.

[13] Q. Du and M. Gunzburger, *Grid generation and optimization based on centroidal Voronoi tessellations*, Appl. Math. Comput., 133 (2002), pp. 591–607.

[14] Q. Du, M. D. Gunzburger, and L. Ju, *Constrained centroidal Voronoi tessellations for surfaces*, SIAM J. Sci. Comput., 24 (2003), pp. 1488–1506.

[15] Q. Du, M. Gunzburger, and L. Ju, *Probabilistic methods for centroidal Voronoi tessellations and their parallel implementations*, J. Parallel Comput., 28 (2002), pp. 1477–1500.

[16] Q. Du, M. Gunzburger, and L. Ju, *Meshfree, probabilistic determination of point sets and support regions for meshless computing*, Comput. Methods Appl. Mech. Engrg., 191 (2002), pp. 1349–1366.

[17] Q. Du, M. Gunzburger, L. Ju, and X. Wang, *Centroidal Voronoi Tessellation Algorithms for Image Compression and Segmentation*, preprint, Pennsylvania State University, University Park, PA, 2003.

[18] Q. Du and L. Ju, *Finite Volume Methods on Spheres and Spherical Centroidal Voronoi Meshes*, IMA preprint 1918, University of Minnesota, Minneapolis, MN, 2003. Available online at http://www.ima.umn.edu/preprints/apr2003/apr2003.html#1918

[19] Q. Du and D. Wang, *Tetrahedral mesh generation and optimization based on centroidal Voronoi tessellation*, Internat. J. Numer. Methods Engrg., 56 (2003), pp. 1355–1373.

[20] P. Frey and H. Borouchaki, *Surface mesh quality evaluation*, Internat. J. Numer. Methods Engrg., 45 (1998), pp. 101–118.

[21] P. Frey, H. Borouchaki, and P. George, 3*D Delaunay mesh generation coupled with an advancing-front approach*, Comput. Methods Appl. Mech. Engrg., 157 (1998), pp. 115–131.

[22] P. George and H. Borouchaki, *Delaunay Triangulation and Meshing: Application to Finite Elements*, Editions Hermès, Paris, 1998.

[23] A. Gersho, *Asymptotically optimal block quantization*, IEEE Trans. Inform. Theory, 25 (1979), pp. 373–380.

[24] R. Gray and D. Neuhoff, *Quantization*, IEEE Trans. Inform. Theory, 44 (1998), pp. 2325–2383.

[25] F. Labelle and J. Shewchuk, *Anisotropic Voronoi diagrams and guaranteed-quality anisotropic mesh generation*, in Proceedings of the 19th ACM Symposium on Computational Geometry, San Diego, CA, 2003, pp. 191–200.

[26] C. Lee, *Automatic metric advancing front triangulation over curved surfaces*, Engrg. Comput., 17 (2000), pp. 48–74.

[27] C. Lee, *On curvature element-size control in metric surface mesh generation*, Internat. J. Numer. Methods Engrg., 50 (2001), pp. 787–807.

[28] G. Leibon and D. Letscher, *Delaunay triangulations and Voronoi diagrams for Riemannian manifolds*, in Proceedings of the Sixteenth Annual Symposium on Computational Geometry, Hong Kong, 2000, ACM, New York, 2000, pp. 341–349.

[29] X. Li, S. Teng, and A. Ungor, *Biting ellipses to generate anisotropic mesh*, in Proceedings of the 8th International Meshing Roundtable, Sandia National Laboratories, Albuquerque, NM, 1999, pp. 97–109. Available online at http://www.andrew.cmu.edu/user/sowen/imr8.html

[30] S. Lloyd, *Least square quantization in PCM*, IEEE Trans. Inform. Theory, 28 (1982), pp. 129–137.

[31] R. Lohner, *Regridding surface triangulation*, J. Comput. Phys., 126 (1996), pp. 1–10.

[32] A. Okabe, B. Boots, and K. Sugihara, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, Wiley, Chichester, UK, 1992.

[33] M. Shephard and M. Georges, *Automatic three-dimensional mesh generation technique by the finite element octree techniques*, Internat. J. Numer. Methods Engrg., 32 (1991), pp. 709–749.

[34] K. Shimada, a. Yamada, and T. Itoh, *Anisotropic triangular meshing of parametric surfaces via close packing of ellipsoidal bubbles*, in Proceedings of the Sixth International Meshing Roundtable, Sandia National Laboratories, Albuquerque, NM, 1997, pp. 63–74. Available online at http://www.andrew.cmu.edu/user/sowen/imr6.html

[35] J. Tristano, S. Owen, and S. Canann, *Advancing front surface mesh generation in parametric space using a Riemannian surface definition*, in Proceedings of the Seventh International Meshing Roundtable, Sandia National Laboratories, Albuquerque, NM, 1998, pp. 429–445. Available online at http://www.andrew.cmu.edu/user/sowen/imr7.html