HKU CS Tech Report TR-2017-01

Shape-Preserving Meshes

Feng Sun*[1], Yi-King Choi†[2], Xiaohu Guo‡[3] and Wenping Wang§[4]

[1]Department of Computer Science, The University of Hong Kong
[2]Department of Computer Science, The University of Texas at Dallas

Abstract

Smooth surfaces are approximated by discrete triangle meshes for applications in computer graphics. Various discrete operators have been proposed for estimating differential quantities of triangle meshes, such as curvatures, for geometric processing tasks. Since a smooth surface can be approximated by many different triangle meshes, we propose to investigate which triangle mesh yields an estimation of differential quantities with optimal accuracy and how to compute such an optimal triangle mesh approximating the given smooth surface. Specifically, we study a special type of triangle meshes, called *shape-preserving meshes*, that preserve the local shapes of the smooth surface they represent, and characterize *optimal shape-preserving meshes*. We present an efficient method for computing the so called *optimal shape-preserving meshes*, and prove the convergence of several discrete differential operators on *optimal shape-preserving meshes*, an important property that does not hold for general triangle meshes. We also show that shape-preserving meshes lead to more accurate estimation of surface differential quantities as compared with other general triangle meshes obtained by commonly used surface meshing methods for the same smooth surface.

Keywords: Shape-preserving meshes, anisotropic meshes, convergence, discrete differential operators

1 Introduction

Smooth surfaces are ubiquitous in computer graphics, solid modeling and fluid dynamics. For better computational efficiency, a smooth surface is often discretized as a triangle mesh for processing in geometry computing, computer simulation, and rendering. A high-quality triangle mesh that accurately approximates the underlying smooth surface is critical to the numerical stability and fast convergence of computation. The quality measures of a triangle mesh normally include the approximation to the underlying surface in terms of distance, normal or tangent plane (first order derivatives), and curvature tensors (second order derivatives). Faithful estimation of differential quantities is important to many applications, ranging from mesh denoising [15], anisotropic remeshing [2] to shape modeling [31, 39].

Since a smooth surfaces can be approximated by many different triangle meshes, it is natural to ask which gives the best approximation in terms of the accuracy in computing differential quantities of the surface. It is well known that a triangle mesh with nearly regular triangles provides the best approximation of an isotropic region with positive Gaussian curvature where the two principal curvatures are roughly of the same magnitude. For anisotropic regions with principal curvatures of different magnitudes, while an anisotropic triangulation with triangle elements elongated along some specific directions appears suitable to capture such anisotropy accurately, it is not clear exactly what kind of an anisotropic triangulation is the most suitable for estimating surface differential quantities. Indeed, we shall see that some instances of anisotropic meshes in the literature do not lead to acceptable estimation of differential quantities, though they may approximate the underlying smooth surface quite well if measured by the distance errors.

Shape preservation On a smooth surface, a point is *elliptic* if the Gaussian curvature at the point is positive, and it is *hyperbolic* if the Gaussian curvature is negative. The local shape of a smooth surface is characterized accordingly. A surface region is called an *elliptic region* if all of its points are elliptic; similarly, a hyperbolic region can be defined. We propose

*qijun.sun@gmail.com
†ykchoi@cs.hku.hk
‡xguo@utdallas.edu
§wenping@cs.hku.hk

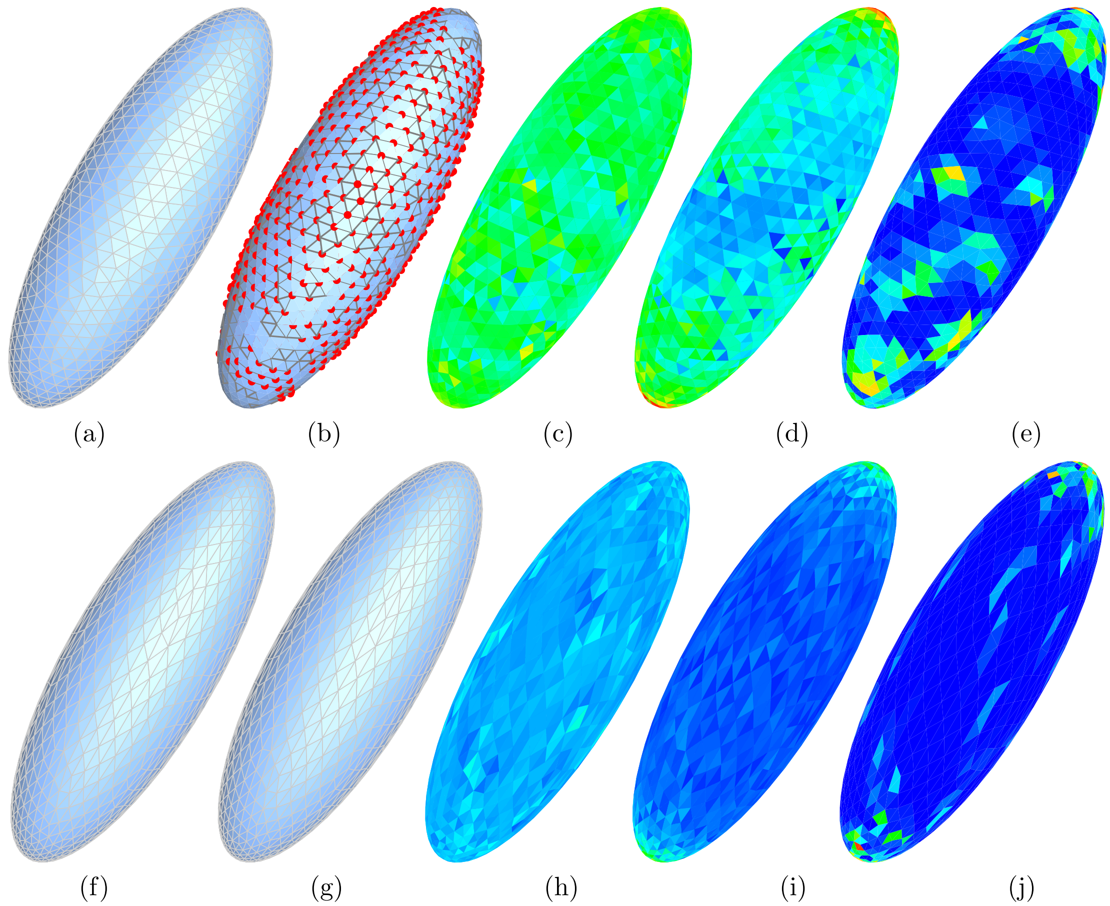


Figure 1: Two meshes approximating the ellipsoid ($x^2 + y^2 + z^2/9 = 1$). The top row shows the isotropic mesh generated by Yan et al.'s algorithm [50], the bottom row is generated by our algorithm. In each row, the first two figures show the mesh and highlight discrete edges and vertices that are not shape-preserving, and the last three figures show the errors of the distance, the normal vector and the Gaussian curvature with respect to the ellipsoid, respectively. In (b) and (g), those edges not on the convex hull are shown in black, and those vertices incident to at least one such edges are not shape-preserving and are highlighted in red. Clearly, our method produces shape-preserving meshes of better approximation in terms of the three error measures.

that the local surface shape of a smooth surface can be preserved when using a triangle mesh to approximate the smooth surface.

For a triangle mesh we define elliptic and hyperbolic vertices analogously. Roughly speaking, a vertex is elliptic if its neighborhood on the triangle mesh is locally convex; otherwise it is hyperbolic. (The rigorous definition will be given later in Section 3.) A triangle mesh $\mathcal{M}$ approximating a smooth surface $\mathcal{S}$ is said to be *shape preserving* if every vertex of $\mathcal{M}$ in an elliptic (resp. hyperbolic) region of the surface $\mathcal{S}$ is an elliptic (resp. hyperbolic) vertex.

We now use the ellipsoidal surface ($x^2+y^2+z^2/9=1$) in Figure 1 to illustrate the idea of shape-preserving meshes. Two different triangle meshes both having 1,024 vertices and approximating the same ellipsoid are shown in the two rows of Figure 1. Since the ellipsoid is convex, we expect a good approximating mesh of it to be convex as well. Figure 1(a) shows an isotropic mesh generated by the algorithm by Yan et al. [50]. Although each triangle is nearly regular, the mesh is, however, not convex. Those edges not belonging to the convex hull of the mesh are highlighted in black. Out of 1,024 vertices, there are 723 vertices that are not shape-preserving (i.e., incident to at least one non-convex edge) and are highlighted in red. The anisotropic mesh in the second row is generated by our algorithm for computing shape-preserving meshes. It is a convex polyhedral surface and thus shape-preserving. This simple example involves only a surface comprising entirely elliptic regions entirely, therefore simply computing the convex hull of the sample gives the shape-preserving mesh for the surface, However, in general, we shall see that computing a shape-preserving triangle mesh for a free-form surface with both elliptic and hyperbolic regions is a non-trivial task. Furthermore, we point out that an optimal shape preserving meshes entails not only the optimal selection of the connectivity of mesh edges but also optimal placement of mesh vertices on a given smooth surface to be discretized.

Contributions This work for the first time characterizes an optimal triangle mesh for approximating a free-form smooth surface for shape preservation and optimal estimation of differential quantities. We propose the notion of *optimal shape-preserving* meshes and prove the convergence of the discrete normal, curvature and Laplacian operators on these meshes. We also devise an algorithm for effectively computing optimal shape-preserving meshes.

2 Review

In this section, we will review the methods for estimating differential quantities on meshes and introduce the centroidal Voronoi tessellation (CVT) meshing framework.

2.1 Estimation of Differential Quantities on Meshes

Differential operators are versatile tools in geometry processing [43] and have been well studied for smooth surfaces [40, 10]. Discrete differential operators play a key role in many geometric processing tasks [48]. Various methods have been proposed for computing differential quantities on mesh surfaces [34, 20, 7, 19, 41, 4].

The methods for estimating differential quantities on meshes can be classified into two types: those based on geometric fitting and those based on discrete differential geometry. The methods based on geometric fitting [7, 19] fit locally a low degree surface patch around a mesh vertex and use the differential quantities of the surface patch at the vertex as an estimation. These geometric fitting methods generally do not use the information of mesh connectivity; when the sampling points are fixed, the estimated differential quantities are the same for different mesh edge connections.

The methods based on discrete differential geometry utilize some identities involving the differential quantities defined on smooth geometric shapes. To estimate the differential quantities which are undefined on C^0 triangle meshes, these methods use some quantities that are well defined on triangle meshes, such as areas and angles, and combine these quantities to produce the estimated differential quantities [34, 24, 41, 4]. For example, an approximation to the Gaussian curvature at a point on a triangle mesh is made by taking a weighted difference between 2π and the sum of angles at the vertex, making use of the Gauss-Bonnet theorem [34].

Next we will review several methods based on discrete differential geometry. A complete review of these methods is beyond the scope of the present paper, so we will discuss some that are closely related to our subsequent discussions on shape-preserving meshes.

Discrete Normal Vector A typical method for approximating the normal vector at a vertex of a tri-

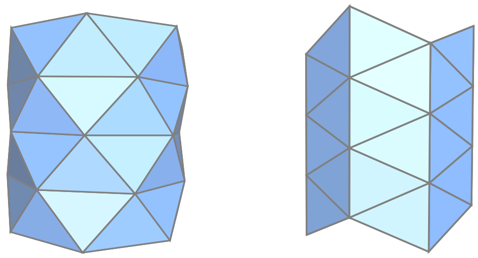


Figure 2: Two different meshes approximating a cylinder. While the Gaussian curvature on a cylinder is zero everywhere, all mesh vertices on the left are hyperbolic. In contrast, the mesh vertices on the right maintain the local shape properties.

angle mesh is averaging the normals of the incident faces of the vertex [19]. Morvan and Thibert propose [35] a discrete scheme to compute the normals and the area of a smooth surface with its approximated triangle mesh and prove [36] that if the Hausdorff distance of a sequence of meshes to a smooth surface converges, the convergence of the normal curvature guarantees the convergence of the area. Hildebrandt et al. [23] further show that if the Hausdorff distance converges, the convergences of normals, the area and Laplace-Beltrami operators are equivalent.

However, we stress that the convergence of the Hausdorff distance does not imply the convergence of normals (or the area or the Laplace-Beltrami operators), as demonstrated by the classical example of the Schwarz lantern [42] shown in Figure 2(left). The mesh is an approximation of a cylinder, on which its vertices lie. This mesh can be refined by increasing the number of layers of vertices along the generating line of the cylinder and increasing the number vertices in each layer. With more vertices added, such refinement reduces the Hausdorff distance between the mesh and the cylinder to zero. However, it can be shown that the normal error does not converge. Hence, the convergence of the Hausdorff distance alone does not guarantee the convergence of the derivatives on triangle meshes. Here the shapes of the triangles in the mesh make a critical difference.

Discrete Curvatures Cohen-Steiner and Morvan [9] present an estimate of curvatures based on normal cycle and restricted Delaunay triangulations. Meyer et al. [34] estimate normals and curvatures using finite element method. Interestingly, Borrelli et al. [6] show that the commonly used normalized angular defect in approximating the point-wise Gaussian curvature converges only on very specific meshes.

Discrete Laplace Operators The *cotan formula* [38] is widely used to compute the discrete Laplace operator on a triangle mesh surface. Glickenstein [18] considers a triangle mesh as the limiting case in the definition of the derivatives to compute discrete Laplacians.

Apart from the various discrete differential operators discussed above, the consistency of these operators is also a concern [25, 47], especially when several discrete operators are involved in one application at the same time. The convergence of discrete differential operators is considered in [23, 49]. Meek and Walton [33] give asymptotic analysis of several approximating methods of the normal and Gaussian curvatures. As we have seen, the quality of the estimated differential quantities of a mesh surface depends not only on the differential operators, but also on the underlying mesh representation. We will show the importance of a shape-preserving mesh to the convergence of several common discrete differential operators.

2.2 Centroidal Voronoi Tessellation

As an optimization-based method, the centroidal Voronoi tessellation (CVT) framework is widely applied in generating both isotropic and anisotropic meshes [12, 46, 37]. We give a brief introduction to the CVT framework and review its generalization for anisotropic mesh generation.

Isotropic CVT Let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ be a set of points, called *seeds*, in Ω. The Voronoi cell Ω_i of a seed $\mathbf{x}_i$ is

$$\Omega_i = \{\mathbf{x} \in \Omega \mid d(\mathbf{x}, \mathbf{x}_i) \leq d(\mathbf{x}, \mathbf{x}_j), \forall j \neq i, j = 1, 2, \ldots, n\},$$

where $d(\mathbf{x}, \mathbf{y})$ is the Euclidean distance between the points $\mathbf{x}$ and $\mathbf{y}$. The Voronoi cells of all the seeds form a Voronoi tessellation of the domain Ω. If every seed is also the centroid of its Voronoi cell, the special Voronoi tessellation thus defined is called a *centroidal Voronoi tessellation* (CVT) and the Voronoi cells are called the *CVT cells*. A CVT is also a critical point of the following isotropic CVT function ([11])

$$F(\mathbf{X}) = \sum_{i=1}^{n} F_i(\mathbf{X}) = \sum_{i=1}^{n} \int_{\Omega_i} d^2(\mathbf{x}, \mathbf{x}_i)\, \mathrm{d}\sigma \qquad (1)$$

where $\mathrm{d}\sigma$ is the differential area element of Ω. Furthermore, a local minimizer of $F(\mathbf{X})$, also known as a *stable CVT*, is more desirable.

The state-of-the-art method for computing a CVT is the L-BFGS method, which is a quasi-Newton method [30, 32]. Gersho's conjecture [17], which is proved for simple two-dimensional domains [22], states that asymptotically the Voronoi cells $\{\Omega_i\}$ of an optimal CVT (i.e., the global optimizer of Eq. (1)) converge to congruent regular hexagons in 2D, as the number of seeds approaches infinity. Therefore, the energy values $F_i(\mathbf{X})$ of all seeds are asymptotically equal in an optimal CVT. The dual of such a hexagonal configuration of seeds is an isotropic triangle mesh with nearly regular triangles and with most vertices having valence six.

Anisotropic CVT Given a domain Ω equipped with a Riemannian metric g, we have a Riemannian manifold $M = (\Omega, g)$. An anisotropic CVT (ACVT) function is defined as follows [12]:

$$F_R(\mathbf{X}) = \sum_{i=1}^{n} \int_{\Omega_{Ri}} d_R^2(\mathbf{x}, \mathbf{x}_i)\, \mathrm{d}\sigma, \tag{2}$$

where $\Omega_{Ri} = \{\mathbf{x} \in \Omega \mid d_R(\mathbf{x}, \mathbf{x}_i) \le d_R(\mathbf{x}, \mathbf{x}_j), \forall j \ne i, j = 1, 2, \ldots, n\}$ is the anisotropic Voronoi cell on M and d_R measures the geodesic distance on M.

The metric tensor $\mathbf{M} = \mathrm{diag}(\kappa_1^2, \kappa_2^2)$ has been proposed for advancing front anisotropic mesh generation [27] and has been adopted in [12], where κ_1 and κ_2 are the principal curvatures at a point. It is clear that the shape and size of the triangle faces are controlled by the metric in the ACVT function. We will see that the anisotropy of a shape-preserving mesh over a surface is governed by a tensor field whose eigenvalues are proportional to the absolute values of principal curvatures of the target surface, that is, $\mathbf{M} = \rho\, \mathrm{diag}(|\kappa_1|, |\kappa_2|)$, where ρ is some density function controlling the triangle size. Furthermore, we will show that this metric should be used in the ACVT function in order to generate a shape-preserving mesh.

The metric $\mathbf{M}$ is also used by Valette et al. [46] for variational anisotropic remeshing, without revealing its important connection with the shape preserving properties. Their method first computes, in each iteration, the anisotropic Voronoi cells of the seeds on a surface equipped with the metric $\mathbf{M}$ and then updates each seed to minimize the Quadratic Error Metrics energy [16] of its Voronoi cell. A discrete scheme clusters triangles in the input mesh to approximate the Voronoi cells, thus generating a result often far from a local minimizer, which can be shown by our experiments later. Recently, Lévy and Liu [29] generalize the CVT framework to adopt general anisotropic metrics. However, to achieve computational efficiency, anisotropic Voronoi cells are approximated by Euclidean Voronoi cells. Our empirical results also show that such an approximation in general does not generate shape-preserving meshes.

3 Shape-Preserving Meshes: Definition

The following characterization of local shape of continuous surfaces is well known in classical differential geometry [10]. Let $\mathcal{S}$ be a surface in $\mathbb{R}^3$ and let $N : \mathcal{S} \to \mathbf{S}^2$ be the Gauss map from $\mathcal{S}$ to the unit sphere $\mathbf{S}^2$. An orientation defined at $\mathbf{p} \in \mathcal{S}$ induces an orientation at $N(\mathbf{p})$ on $\mathbf{S}^2$. We have

$$dN_{\mathbf{p}}(\mathbf{v}_1) \times dN_{\mathbf{p}}(\mathbf{v}_2) = K \cdot \mathbf{v}_1 \times \mathbf{v}_2,$$

where $\{\mathbf{v}_1, \mathbf{v}_2\}$ is the basis in the tangent plane $\mathcal{T}_{\mathbf{p}}$ at $\mathbf{p}$, $dN_{\mathbf{p}}(\mathbf{v})$ is the derivative of the normal vector along the direction $\mathbf{v}$ and K is the Gaussian curvature at $\mathbf{p}$.

Another observation is that the orientation of $\mathcal{T}_{\mathbf{p}}$ induces the orientation of a small closed curve $\mathcal{C}$ in $\mathcal{S}$ around $\mathbf{p}$. The image, $N(\mathcal{C})$, of $\mathcal{C}$ on $\mathbf{S}^2$ has the same orientation if K is positive (i.e., $\mathbf{p}$ is an elliptic point) and has different orientation if K is negative (i.e., $\mathbf{p}$ is a hyperbolic point). Six normal vectors are shown around each of the elliptic and hyperbolic points in Figure 3(a) and (c). As viewed from the top, these six normals are traversed counterclockwise. Figure 3(b) shows the Gaussian images of the small closed curve of (a). We can see that at an elliptic point, the orientation of the Gaussian image of the normals around an elliptic point remains the same as its preimage. Similarly, by comparing (c) and (d), it can be seen that at a hyperbolic point, the orientation on the Gaussian image is reversed.

The local shape of a triangle mesh $\mathcal{M}$ can be characterized in a similar way. Define a *ring* $\mathcal{R}(\mathbf{v}) = \{t_i, i = 1, 2, \ldots, k\}$ as the set of the triangle faces of $\mathcal{M}$ incident to an interior vertex $\mathbf{v}$. Without loss of generality, we assume that, after choosing the orientation of $\mathcal{M}$, the k triangles in $\mathcal{R}(\mathbf{v})$ are arranged in counterclockwise order. Denote the normal of t_i as $\mathbf{n}_i$. On the Gaussian sphere, we get a spherical polygon $\mathcal{P}$ with edges $\{\overline{\mathbf{n}_i\mathbf{n}_{i+1}}, i = 1, 2, \ldots, k-1; \overline{\mathbf{n}_k\mathbf{n}_1}\}$, where $\overline{\mathbf{n}_i\mathbf{n}_j}$ is the minor arc of the great circle on the Gaussian sphere passing through $\mathbf{n}_i$ and $\mathbf{n}_j$.

If $\mathcal{P}$ is a star-shaped spherical polygon and the points $\mathbf{n}_i, i = 1, 2, \ldots, k$ are traversed in the same (resp. reverse) orientation as is on $\mathcal{M}$, then $\mathbf{v}$ is an *elliptic* (resp. *hyperbolic*) vertex. These two types of mesh

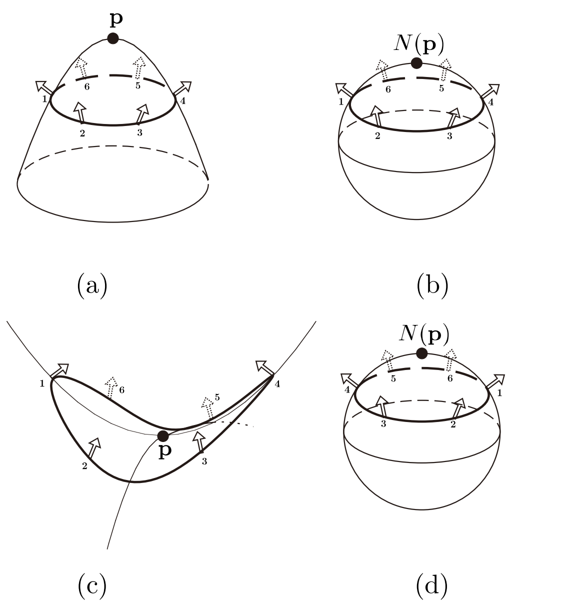


Figure 3: Two types of points and the orientation of a closed curve around a point. In a local canonical coordinate system, (a): an elliptic point and its second order Monge's form $z = -(x^2 + y^2)$. The orientation of the curve is counterclockwise. (b): the Gaussian image of the curve in (a), having the same counterclockwise orientation. (c): a hyperbolic point and its second order Monge's form $z = x^2 - y^2$. The orientation of the curve is counterclockwise. (d): the Gaussian image of the curve in (c) is, however, clockwise oriented.

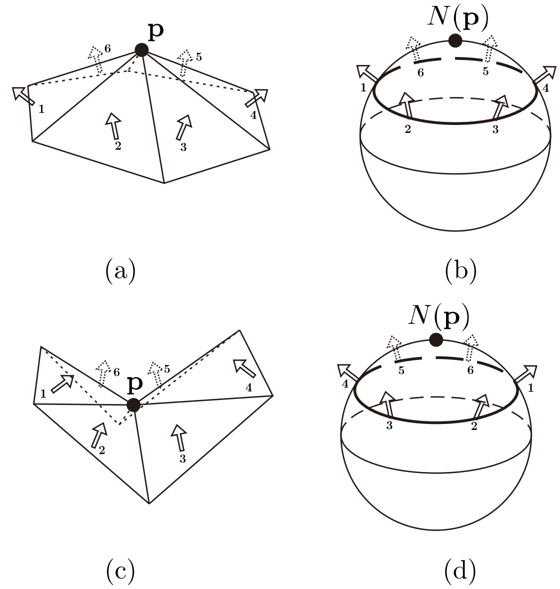


Figure 4: Elliptic vertices and hyperbolic vertices on a triangle mesh surface. (a): A counterclockwise order of visit to the triangles at an elliptic point $\mathbf{p}$. (b): The Gaussian image of the normals of the adjacent triangles at $\mathbf{p}$. The order of traversal of the normals on the mesh is retained on Gaussian sphere. (c): A counterclockwise order of visit to the triangles at a hyperbolic point $\mathbf{p}$. (d): The Gaussian image of the normals of the adjacent triangles at $\mathbf{p}$. In this case, the order of traversal of the normals on the mesh is reversed on the Gaussian sphere.

vertices are also referred to as *convex* vertices and *saddle* vertices, respectively, in [1]. See Figure 4 for an illustration of the two types of mesh vertices. Similar to the continuous case, the orientation at a point $\mathbf{p}$ on the mesh is counterclockwise. If $\mathbf{p}$ is an elliptic point, the Gaussian image of the normals is traversed in counterclockwise order, as shown in Fig. 4(a) and (b). If $\mathbf{p}$ is a hyperbolic point, the Gaussian image of the normals is traversed in clockwise order, as illustrated in Fig. 4(c) and (d).

We now consider a smooth surface $\mathcal{S}$ and its *interpolating mesh* $\mathcal{M}$ with all the vertices of $\mathcal{M}$ lying on $\mathcal{S}$. We will study the differential properties of a vertex $\mathbf{v}$ on $\mathcal{M}$ as well as that of the corresponding point where $\mathbf{v}$ is on $\mathcal{S}$. For brevity, we will also refer to the latter by saying "a certain property of $\mathbf{v}$ on $\mathcal{S}$" when the context is clear.

Definition 3.1 (Shape-Preserving Vertex) *Given a smooth surface $\mathcal{S}$ and an interpolating mesh $\mathcal{M}$ of $\mathcal{S}$, a vertex $\mathbf{v}$ of $\mathcal{M}$ is* shape-preserving *if either one of the following cases holds:*

- $\mathbf{v}$ *is an elliptic vertex* and *its adjacent vertices and* $\mathbf{v}$ *itself lie in an elliptic region of* $\mathcal{S}$*; or*
- $\mathbf{v}$ *is a hyperbolic vertex* and *its adjacent vertices and* $\mathbf{v}$ *itself lie in a hyperbolic region of* $\mathcal{S}$.

Remark 3.2 *There may exist two adjacent vertices* $\mathbf{v}$ *and* $\mathbf{w}$ *on* $\mathcal{M}$*, where* $\mathbf{v}$ *lies in an elliptic region and* $\mathbf{w}$ *lies in a hyperbolic region of* $\mathcal{S}$*. In this case, the edge* $(\mathbf{v}, \mathbf{w})$ *passes through the parabolic curve separating the elliptic and the hyperbolic regions. Since the one-ring triangles of* $\mathbf{v}$ *encapsulate both elliptic and hyperbolic regions, the orientation preservation of a local curve around* $\mathbf{v}$ *is not guaranteed due to the discretization. Therefore, in Definition 3.1, we consider only the cases when both* $\mathbf{v}$ *and its one-ring neighbor vertices lie in an elliptic (or hyperbolic) region.*

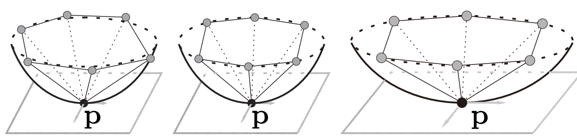


Figure 5: (a) A shape-preserving vertex $\mathbf{p}$ at the origin of the surface $z = x^2 + y^2$; (b) An optimal shape-preserving vertex $\mathbf{p}$ at the origin of the surface $z = x^2 + y^2$; all six vertices incident to $\mathbf{p}$ have the same z-coordinate and form a regular hexagon; (c) another optimal shape-preserving vertex $\mathbf{p}$ at the origin of the surface $z = x^2/4 + y^2$. The six vertices incident to $\mathbf{p}$ have the same z-coordinate and they form an affinely regular hexagon.

Definition 3.3 (Shape-Preserving Mesh) *Let* $\mathcal{M}$ *be an interpolating mesh of a surface* $\mathcal{S}$*. Define a subset* $\mathcal{Q}$ *of the mesh* $\mathcal{M}$ *such that every vertex in* $\mathcal{Q}$ *and its adjacent vertices are in the same elliptic region or in the same hyperbolic region of* $\mathcal{S}$*. Then* $\mathcal{M}$ *is called a* shape-preserving mesh *of* $\mathcal{S}$ *if every vertex in* $\mathcal{Q}$ *is a shape-preserving vertex.*

The shape-preserving mesh of a given surface is not unique, which is illustrated by the following example. Let $\mathbf{p}$ be an umbilical point with equal principal curvatures κ_1 and κ_2 on a surface $\mathcal{S}$. Without loss of generality, we assume that the surface in the local coordinate system around $\mathbf{p}$ takes the form $z = x^2 + y^2$. We now consider the one-ring triangles around a vertex at $\mathbf{p}$ on a shape-preserving mesh of $\mathcal{S}$. Figures 5(a) and (b) show two typical shape-preserving vertices, both with six adjacent vertices. The shape-preserving vertex in (b) is special in that its six adjacent vertices all have the same z-coordinate, denoted by z_0, and they are distributed evenly and form a regular hexagon on the curve $z_0 = x^2 + y^2$.

Figure 5(c) depicts a shape-preserving vertex at a point $\mathbf{p}$ where the local surface is given by $z = x^2/4 + y^2$. Both the local surface and the incident triangles can be obtained by a scaling of factor 2 along the x-axis from that in (b), and the adjacent vertices of $\mathbf{p}$ now form an affinely regular hexagon. Indeed, on shape-preserving meshes with vertex configurations as shown in (b) and (c), several discrete differential operators enjoy better convergence, which we will discuss in the next section.

To distinguish such special shape-preserving vertices from the general ones, we define an *optimal* shape-preserving vertex and an *optimal* shape-preserving mesh as follows.

Definition 3.4 (Optimal Shape-Preserving Vertex) *A shape-preserving vertex* $\mathbf{p}$ *is* optimal *if*

- *the valence of* $\mathbf{p}$ *is* 6*;*
- *in the local Monge's form* $z = \frac{1}{2}(\kappa_1 x^2 + \kappa_2 y^2)$ *defined at* $\mathbf{p}$*, the six one-ring neighbor vertices of* $\mathbf{p}$ *are represented by*

$$\Big(\frac{s_i}{\sqrt{|\kappa_1|}}, \frac{t_i}{\sqrt{|\kappa_2|}}, \frac{1}{2}(\mathrm{sign}(\kappa_1)s_i^2 + \mathrm{sign}(\kappa_2)t_i^2) + O(h^3)\Big)^T,$$

where $s_i = \cos(\theta + \frac{i\pi}{3})h$*,* $t_i = \sin(\theta + \frac{i\pi}{3})h$*,* $i = 1, \ldots, 6$*, and* $h > 0$.

Definition 3.5 (Optimal Shape-preserving Meshes) *A shape-preserving mesh is an* optimal *shape-preserving mesh if all its shape-preserving vertices are optimal shape-preserving vertices.*

Remark 3.6 *The definition of optimal shape-preserving vertices can be interpreted intuitively as follows. The one-ring neighbor vertices around an optimal shape-preserving vertex* $\mathbf{v}$ *are mapped to normal vectors of these vertices. The differences between these normal vectors from the normal of* $\mathbf{v}$ *(the Gauss map of* $\mathbf{v}$*) are roughly the same. In other words, a circle around the normal of* $\mathbf{v}$ *is roughly formed on the Gauss sphere. Therefore, given a vertex* $\mathbf{v}$*, the normal vectors of points in its neighbor region are close to the normal vector of* $\mathbf{v}$*. This is a more faithful approximation than an arbitrary choice of one-ring neighbor vertices, leading to improved convergence properties of discrete differential operators.*

4 Shape-preserving Meshes: Properties

We now discuss the convergence of several fundamental discrete geometric quantities and discrete differential operators on optimal shape-preserving meshes.

4.1 Discrete Normal Vector

The average of the normals of adjacent faces [19] is often used to approximate the normal vector at a vertex. Let h denote the mesh size, i.e., the length of the longest edge incident to an optimal shape-preserving vertex $\mathbf{v}$. Let $Q_{\mathbf{v}}$ denote the second order approximating surface at $\mathbf{v}$, which is the second order Monge's form in the local canonical coordinate system. The displacement of the adjacent vertices of $\mathbf{v}$ from $Q_{\mathbf{v}}$ is then $O(h^3)$, which is the remainder of the second order Taylor expansion. Based on this observation, we have the following theorem.

Theorem 4.1 *On an optimal shape-preserving mesh, if the length of the edges incident to a shape-preserving vertex $\mathbf{v}$ is $O(h)$, the error of the estimated normal vector is $O(h^2)$.*

Proof 1 *By Definition 3.4, the local second order Monge's form at the vertex $\mathbf{v}$ is $z = \frac{1}{2}(\kappa_1 x^2 + \kappa_2 y^2)$; the one-ring neighbor vertices of $\mathbf{v}$ are given by*

$$\mathbf{v}_i = \Big(\frac{s_i}{\sqrt{|\kappa_1|}}, \frac{t_i}{\sqrt{|\kappa_2|}}, \frac{1}{2}(\mathrm{sign}(\kappa_1)s_i^2 + \mathrm{sign}(\kappa_2)t_i^2) + O(h^3)\Big)^T,$$

where $s_i = \cos(\theta + \frac{i\pi}{3})h$, $t_i = \sin(\theta + \frac{i\pi}{3})h$, $i = 1, \ldots, 6$, and $h > 0$.

Hence, the normal direction of the triangle $\mathbf{v}\mathbf{v}_i\mathbf{v}_{i+1}$ incident to $\mathbf{v}$ is

$$\Big(\frac{\mathrm{sign}(\kappa_1)\alpha_1 + \mathrm{sign}(\kappa_2)\alpha_2}{\sqrt{|\kappa_2|}} + O(h^4), \frac{\mathrm{sign}(\kappa_1)\alpha_3 + \mathrm{sign}(\kappa_2)\alpha_4}{\sqrt{|\kappa_1|}} + O(h^4), \frac{\sqrt{3}h^2}{2\sqrt{|\kappa_1\kappa_2|}}\Big)^T,$$

where $\alpha_1 = (s_{i+1}^2 t_i - s_i^2 t_{i+1})$, $\alpha_2 = (t_i t_{i+1}^2 - t_i^2 t_{i+1})$, $\alpha_3 = (s_i s_{i+1}^2 - s_i^2 s_{i+1})$ and $\alpha_4 = (s_i t_{i+1}^2 - s_{i+1} t_i^2)$. Here, $i+1$ is 1 when $i = 6$.

Summing up the six normals, we get the estimated normal at $\mathbf{v}$ as $\big(O(h^2), O(h^2), 1\big)^T$ after normalization. Since the actual normal at $\mathbf{p}$ is $(0,0,1)^T$, we conclude that the error of the estimated normal is $O(h^2)$.

Remark 4.2 *It is noted in [33] that on general meshes, the approximation order of a discrete normal vector is $O(h)$ only. Hence, optimal shape-preserving meshes improve the accuracy by one order of magnitude.*

4.2 Discrete Curvatures

A typical method for estimating the principal curvatures and principal directions of a surface represented by a triangle mesh is to estimate the normal curvatures along the edges incident to a vertex and then find the second fundamental form in the least squares sense by fitting the normal curvatures along these edge directions [34, 19]. More specifically, given a vertex $\mathbf{v}$ and its k one-ring neighbor vertices $\mathbf{v}_i, i = 1, \ldots, k$, one first estimates the normal at $\mathbf{v}$, denoted by $\mathbf{n}$, and the tangent plane at $\mathbf{v}$ is thus defined. The quadratic curve passing through $\mathbf{v}$ and $\mathbf{v}_i$ with $\mathbf{n}$ as the normal at $\mathbf{v}$ is computed. The estimated normal curvature along the projected direction of $\mathbf{v}\mathbf{v}_i$ into the tangent plane is the curvature of the osculating circle. An illustration is shown in Figure 6, where the magnitude of the Gaussian curvature is shown in the tangent plane. The formula to compute the estimated normal curvature along the edge $\mathbf{v}\mathbf{v}_i$ is $\hat{\kappa}_i = \frac{2\mathbf{v}\mathbf{v}_i \cdot \mathbf{n}}{-\mathbf{v}\mathbf{v}_i^2}$. On the other hand, the normal curvature at $\mathbf{v}$ along a direction denoted by the unit vector $\mathbf{u} = (u_x, u_y)^T$ in the tangent plane can also be computed as: $\kappa_{\mathbf{u}} = \mathbf{u}^T \mathrm{I\!I} \mathbf{u}$, where $\mathrm{I\!I} = \begin{bmatrix} e & f \\ f & g \end{bmatrix}$ is the second fundamental form at the point $\mathbf{v}$.

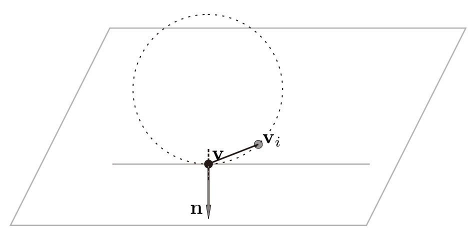


Figure 6: A point $\mathbf{v}$, its adjacent vertex $\mathbf{v}_i$ and the osculating circle along the shown principal direction.

We now have an identity involving $\hat{\kappa}_i$ induced by $\mathbf{v}_i$:

$$\mathbf{u}_i^T \mathrm{I\!I} \mathbf{u}_i = \frac{2\mathbf{v}\mathbf{v}_i \cdot \mathbf{n}}{-\mathbf{v}\mathbf{v}_i^2} \tag{3}$$

for each adjacent vertex $\mathbf{v}_i$ of $\mathbf{v}$, where $\mathbf{u}_i = (u_{ix}, u_{iy})^T$ is the unit vector of $\mathbf{v}\mathbf{v}_i$ projected on the tangent plane of $\mathbf{v}$. Hence, we have a linear system of equations with $\mathrm{I\!I}$ as the unknown. By solving this linear system in a linear least squares sense, we obtain the second fundamental form and can then estimate the principal curvatures and principal directions accordingly [45]. Since the valence of an internal vertex is at least three, the solution of the linear system is unique. The next theorem states the convergence of discrete curvatures using the above estimation on optimal shape-preserving meshes.

Theorem 4.3 *On an optimal shape-preserving mesh, if the length of the edges incident to a vertex* $\mathbf{v}$ *is* $O(h)$*, the error of the estimated principal curvatures and principal directions is* $O(h)$*.*

Proof 2 *By Definition 3.4, the one-ring neighbor vertices of* $\mathbf{v}$ *can be represented by* $\mathbf{v}_i = \left(\frac{s_i}{\sqrt{|\kappa_1|}}, \frac{t_i}{\sqrt{|\kappa_2|}}, \frac{1}{2}(\mathrm{sign}(\kappa_1)s_i^2 + \mathrm{sign}(\kappa_2)t_i^2) + O(h^3)\right)^T$*, where* $h > 0$*,* $s_i = \cos(\theta + \frac{i\pi}{3})h$ *and* $t_i = \sin(\theta + \frac{i\pi}{3})h$*,* $i = 1, \ldots, 6$*. The estimated normal curvature at* $\mathbf{v}$ *along the edge* $(\mathbf{v}, \mathbf{v}_i)$ *is*

$$\frac{\kappa_1\kappa_2\left(\mathrm{sign}(\kappa_1)\cos^2(\theta + \frac{i\pi}{3}) + \mathrm{sign}(\kappa_2)\sin^2(\theta + \frac{i\pi}{3})\right)}{\kappa_2\cos^2(\theta + \frac{i\pi}{3}) + \kappa_1\sin^2(\theta + \frac{i\pi}{3})} + O(h).$$

We denote the exact normal curvature at $\mathbf{v}$ *along the edge* $(\mathbf{v}, \mathbf{v}_i)$ *as* $\hat{c}_i$*. Hence, the estimated normal curvature has an* $O(h)$ *error. From Eq. (3), we have the following linear system of equations:*

$$\mathbf{u}_i^T \mathbb{I}\mathbf{u}_i = \hat{c}_i + O(h),\ i = 1, \ldots, 6. \qquad (4)$$

Denote II *as a vector* $(e, f, g)^T$*. We rewrite the linear system in the matrix form as* $\mathbf{U}II = \mathbf{C} + O(h)$*, where* $\mathbf{U}$ *is a* 6×3 *matrix with the six rows vectors* $(u_{ix}^2, 2u_{ix}u_{iy}, u_{iy}^2), i = 1, \ldots, 6$*, and* $\mathbf{C}$ *is the vector* $(\hat{c}_1, \ldots, \hat{c}_6)^T$*. Solving this linear system, we get* $II = \mathbf{U}^{-1}\mathbf{C} + O(h)$*, where* $\mathbf{U}^{-1}$ *is the generalized inverse of* $\mathbf{U}$*. We see that there is an* $O(h)$ *error in the estimated second fundamental form. Then, we compute the eigenvalues and eigenvectors of the matrix* $\mathbb{I}$ *to get the principal curvatures and principal directions. For a* 2×2 *matrix, the computation of its eigenvalues* λ_i *is equivalent to solving a quadratic equation* $\lambda^2 - m\lambda + n = 0$*, where* m *is the trace of the matrix and* n *is the determinant of the matrix. The* $O(h)$ *error in the coefficients of the quadratic equation yields an* $O(h)$ *error in its roots, which are the principal curvatures in our case. We conclude that the error in the estimated principal curvatures and principal directions is* $O(h)$*.*

Remark 4.4 *To the best of our knowledge, we are not aware of any previous literature on convergence analysis of this method for estimating principle curvature and principal directions on general meshes. Here, we prove that on optimal shape-preserving meshes, the estimation of principal curvature and principal directions converges.*

Corollary 1 *On an optimal shape-preserving mesh, if the length of the edges incident to a vertex* $\mathbf{v}$ *is* $O(h)$*, the error of the estimated Gaussian curvature and mean curvature is* $O(h)$*.*

4.3 Discrete Laplace-Beltrami Operators

Next, we show that the discrete Laplace-Beltrami operators converge on optimal shape-preserving meshes.

Theorem 4.5 *The Laplace-Beltrami operators converge on optimal shape-preserving meshes.*

Proof 3 *It is shown in [23] that the convergence of the normal vector and the convergence of the Laplace-Beltrami operator are equivalent, assuming the convergence of the Hausdorff distance. Our proof follows from Theorem 4.1 and the claim that the Hausdorff distance converges on an optimal shape-preserving mesh; the proof of the latter is simple and is thus omitted here.*

Remark 4.6 *Approximately the Hausdorff distance between a smooth surface and an interpolating mesh (i.e., its piecewise linear approximation) is the interpolation error of the mesh.*

5 Shape-preserving Meshes: Computation

In this section, we adopt the metric $M = \mathrm{diag}(\kappa_1, \kappa_2)$ [46] in the ACVT framework to generate optimal shape-preserving meshes. We first discuss the intrinsic relation between the metric M and shape-preserving meshes.

5.1 Riemannian Metric for Shape-preserving Mesh Generation

It is noted in [44] that there is a correspondence between an anisotropic CVT of a two-dimensional manifold and a CVT in a higher dimensional Euclidean space. According to the Gersho's conjecture [17, 21], the shape of a Voronoi cell in a CVT of a two-dimensional domain is asymptotically a regular hexagon. Therefore, the shape of a Voronoi cell in an anisotropic CVT can be inferred from the metric $\mathbf{M}$ defined on that domain. Given the metric $\mathbf{M}$, there is a mapping $\mathbf{G}$ from the high dimensional Euclidean space to the Riemannian manifold that satisfies $\mathbf{M} = (\mathbf{G}^{-1})^T \mathbf{G}^{-1}$. The shape of a Voronoi cell in an anisotropic CVT is then the image of a regular hexagon under the transformation $\mathbf{G}$.

We now derive the metric tensor $\mathbf{M}$ required for generating shape-preserving meshes with the CVT framework. As illustrated in Figure 5, in an isotropic region where the two principal curvatures are of similar magnitude, the triangles incident to an optimal shape-preserving vertex are regular triangles asymptotically. According to the definition of optimal shape-preserving vertices, the triangles incident to an optimal shape-preserving vertex can be obtained by transforming the triangles at an optimal shape-preserving vertex at an umbilical point. For each principal direction, this transformation applies a scaling along that direction. The triangles incident to an optimal shape-preserving vertex are thus scaled from the regular triangles by a factor of $\frac{1}{\sqrt{|\kappa_i|}}$ along $\mathbf{d}_i$, where κ_i is the principal curvature along $\mathbf{d}_i, i = 1, 2$, the two principal directions. Therefore, we can see that the mapping required to generate the appropriate Voronoi cells and hence the triangles for shape-preserving meshes is $\mathbf{G} = c\,\mathrm{diag}(\frac{1}{\sqrt{|\kappa_1|}}, \frac{1}{\sqrt{|\kappa_2|}})$, where c is a constant. We have thus obtained $\mathbf{M} = \rho\,\mathrm{diag}(|\kappa_1|, |\kappa_2|)$, where ρ is considered a density function which will be discussed in the next section.

5.2 Optimal Approximation

The density function ρ in the metric $\mathbf{M} = \rho\,\mathrm{diag}(|\kappa_1|, |\kappa_2|)$ is yet to be determined. Since the L_∞ distance error is a good approximation of the Hausdorff distance, which in turn is related to the convergence of the Laplace-Beltrami operator, we exploit the density function ρ to achieve minimal L_∞ distance error. Using the observation that the energies of all seeds are equal asymptotically [17] after convergence of the ACVT energy function (Eq. (2)), one may derive that the density function ρ should be taken as $\sqrt{|\kappa_1 \kappa_2|}$. In other words, the ACVT energy function that we shall use for generating shape-preserving meshes is given by Eq. (2) with the metric being $\mathbf{M} = \rho\,\mathrm{diag}(|\kappa_1|, |\kappa_2|)$, where $\rho = \sqrt{|\kappa_1 \kappa_2|}$.

5.3 L-BFGS Method

We use the limited memory BFGS method [30], or L-BFGS method for short, to minimize the ACVT function $F(\mathbf{X})$ in Eq. (2). The L-BFGS method is an iterative quasi-Newton method, which approximates the inverse of the Hessian matrix by accumulating gradients of the previous iterations. As demonstrated in [32], the L-BFGS method is an efficient method for large-scale problems on which Newton methods are too costly. On the other hand, the L-BFGS method is also much more effective than the gradient descent methods with nearly the same running time.

Given initial seeds $\mathbf{X} = (\mathbf{x}_i)$, the ACVT optimization involves the following steps in each iteration: (a) computing the anisotropic Voronoi tessellation of $\mathbf{X}$; (b) evaluating the ACVT function $F(\mathbf{X})$ and its gradient $\nabla F(\mathbf{X})$; (c) updating the seeds by the L-BFGS method. If the termination condition is met, the algorithm terminates; otherwise, we call the L-BFGS method to find the new positions of the seeds $\mathbf{X}$.

5.4 Post-processing

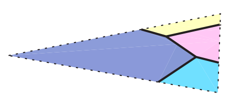
Due to its nature as an optimization framework, the L-BFGS method cannot reach the global minimium in general and stops at a local minimum in practice. Some non shape-preserving vertices still persist after the optimization by using the L-BFGS method. An observation is that in a convex area with positive Gaussian curvature, a *local* convex hull of the vertices within this area is shape-preserving (although not necessarily optimal). Thus, to further optimize the result generated by the L-BFGS method, we check the non shape-preserving vertices in a locally convex area and check whether two neighboring non shape-preserving vertices can be modified to shape-preserving vertices by an edge flip; and if so, perform the edge flip. This post-processing step further reduces the number of non shape-preserving vertices. In Figures 7, 8, 9 and 10, the number of non shape-preserving vertices is reduced from 59 to 31, from 13 to 7, from 29 to 21 and from 16 to 12, respectively, by post-processing.

6 Implementation Issues

In this section, we will discuss several implementation issues. We start with the computation of the anisotropic Voronoi tessellation of a set of seeds $\mathbf{X}$, which is essential for evaluating the ACVT function $F(\mathbf{X})$ and its gradient $\nabla F(\mathbf{X})$.

6.1 Computing Anisotropic Voronoi Tessellation

The existence of the Voronoi diagram (VD) of a given set of sufficiently dense seeds on a Riemannian manifold has been established by Leibon and Letscher [28]. However, the computation of the corresponding restricted anisotropic Voronoi diagram (in which the Voronoi cells are restricted to a given manifold) [13] is not easy, even for many applications in which a manifold is endowed with a piecewise constant or a piecewise linear metric. Different algorithms for computing anisotropic VD have been presented in [8, 26, 5]. However, no practical implementation of their methods are known for mesh surfaces. There are several existing methods for approximating anisotropic VDs, assuming that the manifold is triangulated. In the work by Valette et al. [46], an approximation of the anisotropic VD is computed by clustering of triangles, that is, assigning each triangle in its entirety to *one* seed. This results in a very crude approximation, especially when the number of seeds is comparable to the number of mesh triangles. It may even lead the optimization for a CVT to getting stuck at a bad minimizer. The right figure illustrates a typical situation in which a triangle is covered by more than one (four in this example) Voronoi cells and therefore should be tessellated into parts belonging to different seeds.



On the other hand, in order to achieve efficient computation of the anisotropic CVT, Lévy and Liu [29] replace the anisotropic VDs by standard VDs. We will show later in our experiments that this method is not effective for shape-preserving mesh generation.

Inspired by the exact computation of restricted isotropic VDs on mesh surfaces [50], we devise an efficient method for restricted anisotropic VD which also splits a triangle that should belong to more than one seeds. We define a constant metric on each triangle for robust computation. We observe that when the seeds assume a good distribution and the number of seeds is a fraction of the number of triangles in the domain, most triangles on the border of an anisotropic Voronoi cell belong to only two Voronoi cells. We note also that near a Voronoi vertex, one triangle almost always belongs to three or four Voronoi cells in practice, although it may belong to arbitrarily many Voronoi cells theoretically. Based on this observation, for each triangle, we first find the nearest seeds of its vertices. If the entire triangle lies within the Voronoi cell of a seed, it is assigned to the seed. Otherwise, we split the triangle into at most four parts and assign them to different seeds. The approximated anisotropic Voronoi cell is thus found. Our experiments show that with the number of triangles being several times of the number of seeds, such an approximation is accurate in the converging stage when the the seeds are distributed evenly.

6.2 Initialization of Seeds

It can also be shown that on a two-dimensional Riemannian manifold, the density of seeds at convergence is proportional to $\sqrt{\|\mathbf{M}\|_2}$, where $\|\mathbf{M}\|_2$ is the 2-norm of the metric $\mathbf{M}$ of the domain. Therefore, in order to accelerate computation, we initialize the seeds to conform to a desired density accordingly. We achieve this by using the error diffusion method as in [3] with a region growing approach.

6.3 Handling Degenerate Case

It is noted that when a region is nearly planar or cylindrical, one or both principal curvatures can be very small. To ensure a robust computation in the computation of anisotropic VDs, we set a positive tolerance value δ. If the absolute value of the input principal curvature is less than δ, we replace it by δ. This improves the numerical robustness when either κ_1 or κ_2 vanishes or is very small.

7 Empirical Validation

In this section, we demonstrate the effectiveness of our algorithm for shape-preserving mesh generation by several experiments. Our implementation uses *CGAL 4.0* [14] to compute Delaunay triangulations. All experiments are performed on a computer with a 2.3 GHz Intel Core i5 CPU and 4 GB RAM.

First we show the experiments on the convergence properties of optimal shape-preserving meshes. To compare with the ground truth, we choose several analytic surfaces with known exact normal and curvature information. Second, we demonstrate the effectiveness of our algorithm on free-form surfaces.

7.1 Experiments on the Convergence of Discrete Differential Operators

To show the convergence properties on analytic surfaces we generate meshes with different resolutions using the prevailing mesh decimation algorithm in [16] based on quadratic error minimization (QEM). The convergence statistics on these meshes are shown in Table 1. Table 2 shows the statistics on nearly optimal shape-preserving meshes generated by our method.

We choose four analytic surfaces whose exact normal vectors and Gaussian curvatures are known prior to the comparison. On a surface, increasing the number of vertices fourfold halves the triangle size of the mesh. Therefore, for each surface, we generate meshes with 200, 800, 3,200 and 12,800 vertices. We see that the convergence of both the normal vector and the Gaussian curvature is improved on nearly shape-preserving meshes, compared with meshes generated by using the QEM algorithm [16]. This is in conformity with our discussion in Section 4.

7.2 Experiments on Free-form Surfaces

We evaluate our algorithm for generating shape-preserving meshes against both isotropic and anisotropic remeshing methods, which are most widely used for mesh generation. Specifically, we compare against the implementation by Yan et al. [50] for isotopic remeshing and the discrete anisotropic CVT method by Valette et al. [46] which uses the same metric $\mathbf{M}$ as ours in their quadratic error minimization (QEM).

The first example is conducted on a duck model with 50,000 triangles. A total of 2,048 seeds is used. The results of the three methods are shown in Figure 7, with non shape-preserving vertices highlighted in red. For the isotropic mesh (left), the mesh is of bad quality in terms of shape-preservation although each triangle is nearly regular. There are 341 vertices which are not shape-preserving. The mesh in the middle is generated by the discrete anisotropic CVT. The result is better than isotropic remeshing, but still 237 vertices are not shape-preserving, due to both the discrete computation and updating of seeds by QEM.

Our algorithm, on the other hand, reduces the non shape-preserving vertices dramatically to 31 vertices. In our resulting mesh, the anisotropic triangles around the neck of the duck capture the local shape faithfully. As a side note, given a free-form shape, it is hard to generate a mesh with only shape-preserving vertices, due to the combinatorics of the vertices and the difficulty of local optimization in searching for a good minimizer.

The second example is a double-torus model with 50,000 triangles (Figure 8). The number of seeds is 1,024. Again, our algorithm generates much better result (with only 7 non shape-preserving vertices) than the other two methods. The third example is a human face with 115,876 triangles (Figure 9). The number of seeds is 2,048. Note that on the nose and the cheek, the non shape-preserving vertices are much reduced in our result. The last example is a pig with 56,960 triangles (Figure 10) with 1,024 seeds. On this model, there are both elliptic and hyperbolic regions. Near the region border, adjacent vertices may fall onto different regions. According to Definition 3.1 of shape-preserving vertices, these vertices are not considered when classifying whether a vertex is shape-preserving or not.

The computation time of both isotropic CVT [50] and our algorithm is less than 10 minutes for all these examples. The computation of discrete ACVT [46] is faster (in less than 1 minute), since no triangle clipping operation is involved.

8 Conclusion

In this paper, we define a shape-preserving mesh whose vertices represent faithfully the local shape of a surface. We demonstrate the importance of shape-preserving meshes by proving the convergence of several widely used discrete differential operators on these meshes. Moreover, we propose an effective algorithm for computing shape-preserving meshes. The new algorithm is based on the centroidal Voronoi tessellation framework with a carefully derived metric. Experimental results show that our method performs much better than the existing methods in generating shape-preserving meshes.

While the optimal shape-preserving mesh is well defined, its efficient computation is nontrivial because it involves determination of optimal layout of mesh vertices as well as optimal mesh connectivity. We have proposed a numerical method based on anisotropic centroidal Voronoi diagram. Due to its nature of local optimization, our method produces satisfactory but not perfect results, which means that non shape-preserving vertices cannot be removed completely for

	Elliptic Paraboloid		Hyperbolic Paraboloid		Hyperboloid		Torus	
#vertices	e_n	e_{GC}	e_n	e_{GC}	e_n	e_{GC}	e_n	e_{GC}
200	2.23×10^{-2}	2.54×10^{-2}	1.15×10^{-2}	2.29×10^{-2}	1.33×10^{-2}	1.51×10^{-2}	3.40×10^{-2}	2.76×10^{-2}
800	7.92×10^{-3}	1.03×10^{-2}	5.94×10^{-3}	1.31×10^{-2}	7.26×10^{-3}	3.85×10^{-3}	1.85×10^{-2}	8.27×10^{-3}
$3,200$	5.56×10^{-3}	7.39×10^{-3}	3.87×10^{-3}	1.27×10^{-2}	5.18×10^{-3}	2.97×10^{-3}	1.12×10^{-2}	7.92×10^{-3}
$12,800$	4.77×10^{-3}	5.76×10^{-3}	1.96×10^{-3}	7.39×10^{-3}	4.01×10^{-3}	1.89×10^{-3}	8.71×10^{-3}	4.28×10^{-3}

Table 1: On meshes generated by using the algorithm in [16], the convergence of the normal vector and the discrete Gaussian curvature [34] is shown. The error e_n is the angle between the estimated normal vector and the exact one. The error e_{GC} is the difference between the estimated Gaussian curvature and the exact one. The slow convergence is observed. The equations of the elliptic paraboloid, the hyperboloid paraboloid and the hyperboloid are $z = x^2 + y^2$, $z = x^2 - y^2$ and $x^2 + y^2 - z^2 = 1$,respectively. The equation of the torus is $(\sqrt{x^2 + y^2} - 1)^2 + z^2 = 1$.

	Elliptic Paraboloid		Hyperbolic Paraboloid		Hyperboloid		Torus	
#vertices	e_n	e_{GC}	e_n	e_{GC}	e_n	e_{GC}	e_n	e_{GC}
200	2.91×10^{-3}	9.73×10^{-3}	5.17×10^{-3}	1.33×10^{-2}	5.27×10^{-3}	6.55×10^{-3}	2.68×10^{-2}	1.31×10^{-2}
800	1.42×10^{-3}	4.06×10^{-3}	2.02×10^{-3}	1.09×10^{-2}	3.08×10^{-3}	2.96×10^{-3}	6.38×10^{-3}	5.58×10^{-3}
$3,200$	9.96×10^{-4}	3.47×10^{-3}	1.19×10^{-3}	7.61×10^{-3}	2.11×10^{-3}	1.81×10^{-3}	3.87×10^{-3}	3.80×10^{-3}
$12,800$	3.81×10^{-4}	2.81×10^{-3}	4.51×10^{-4}	5.90×10^{-3}	9.17×10^{-4}	9.61×10^{-4}	1.54×10^{-3}	2.82×10^{-3}

Table 2: On nearly optimal shape-preserving meshes generated by our method, the convergence of the normal vector and the discrete Gaussian curvature [34] is shown. The error e_n is the angle between the estimated normal vector and the exact one. The error e_{GC} is the difference between the estimated Gaussian curvature and the exact one. We see that the convergence of the discrete differential operators is improved on these meshes, as compared with the convergence of the operators on the meshes given in Table 1.

complex surfaces. The improvement on this aspect needs further research.

References

[1] Lyuba Alboul, Gertjan Kloosterman, Cornelis Traas, and Ruud van Damme. Best data-dependent triangulations. *Journal of Computational and Applied Mathematics*, 119(1-2):1–12, 2000.

[2] Pierre Alliez, David Cohen-Steiner, Olivier Devillers, Bruno Lévy, and Mathieu Desbrun. Anisotropic polygonal remeshing. *ACM Trans. Graph.*, 22(3):485–493, July 2003.

[3] Pierre Alliez, Éric Colin de Verdière, Olivier Devillers, and Martin Isenburg. Centroidal Voronoi diagrams for isotropic surface remeshing. *Graphical Models*, 67(3):204–231, 2003.

[4] Alexander I. Bobenko and Yuri B. Suris. *Discrete differential geometry*, volume 98 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2008. Integrable structure.

[5] J-D. Boissonnat, C. Wormser, and M. Yvinec. Locally uniform anisotropic meshing. In *Symposium on Computational Geometry (SOCG)*, pages 270–277, 2008.

[6] V. Borrelli, F. Cazals, and J. M. Morvan. On the angular defect of triangulations and the pointwise approximation of curvatures. *Computer Aided Geometric Design*, 20(6):319 – 341, 2003.

[7] F. Cazals and M. Pouget. Estimating differential quantities using polynomial fitting of osculating jets. *Comput. Aided Geom. Des.*, 22:121–146, February 2005.

[8] Siu Win Cheng, Tamal K Dey, Edgar A Romas, and Rephael Wengar. Anisotropic surface meshing. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 202–211, 2006.

[9] David Cohen-Steiner and Jean-Marie Morvan. Restricted Delaunay triangulations and normal cycle. In *Proceedings of the nineteenth annual symposium on Computational geometry*, SCG '03, pages 312–321, New York, NY, USA, 2003. ACM.

[10] Manfredo P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice Hall, 1976.

[11] Qiang Du, Vance Faber, and Max Gunzburger. Centroidal Voronoi tessellations: applications and algorithms. *SIAM Review*, 41:637–676, 1999.

[12] Qiang Du and Desheng Wang. Anisotropic centroidal Voronoi tessellations and their applica-

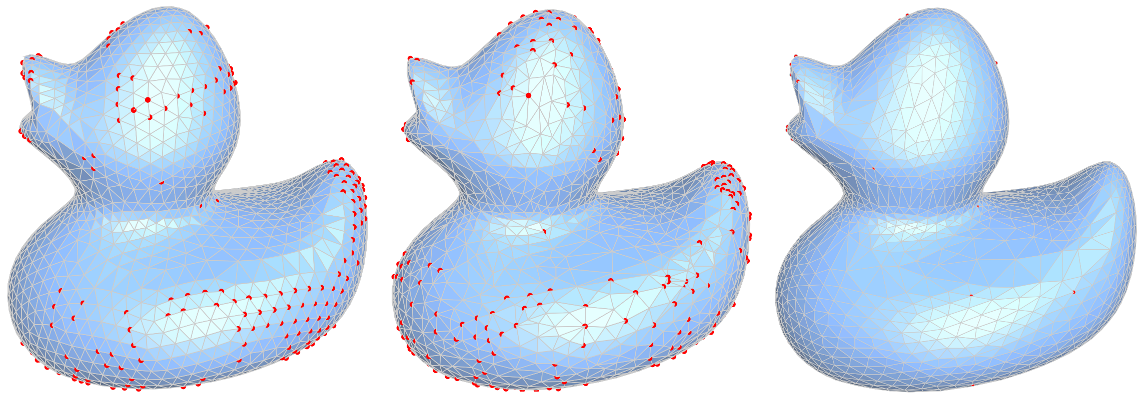


Figure 7: A duck model: meshing result with 2,048 vertices. From left to right: isotropic CVT [50], discrete anisotropic CVT [46] and our method. The number of non shape-preserving vertices are 341, 237 and 31, respectively.

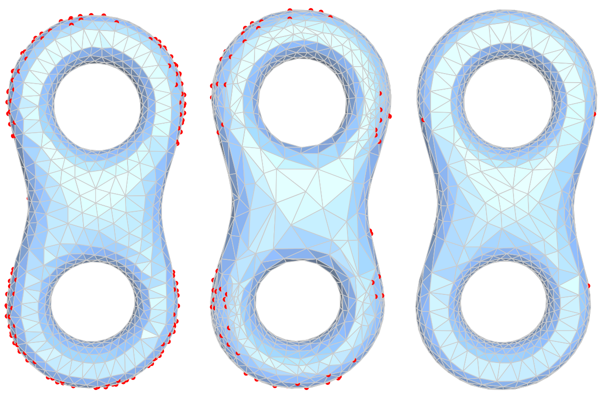


Figure 8: Eight shape model: meshing results with 1,024 vertices. From left to right: isotropic CVT [50], discrete anisotropic CVT [46] and our method. The number of non shape-preserving vertices are 213, 104 and 7, respectively.

tions. *SIAM J. Sci. Comput.*, 26(3):737–761, 2005.

[13] Herbert Edelsbrunner and Nimish R. Shah. Triangulating topological spaces. In *Proceedings of the tenth annual symposium on Computational geometry*, SCG '94, pages 285–292, New York, NY, USA, 1994. ACM.

[14] A. Fabri. CGAL-the computational geometry algorithm library. In *Proceedings of 10th International Meshing Roundtable*, pages 137–142, 2001.

[15] Shachar Fleishman, Iddo Drori, and Daniel Cohen-Or. Bilateral mesh denoising. *ACM Trans. Graph.*, 22(3):950–953, July 2003.

[16] M. Garland and P. Heckbert. Surface simplification using quadric error metrics. In *Proceedings of SIGGRAPH 97*, pages 209–216, 1997.

[17] A. Gersho. Asymptotically optimal block quantization. *Information Theory, IEEE Transactions on*, 25(4):373–380, Jul 1979.

[18] D. Glickenstein. Geometric triangulations and discrete Laplacians on manifolds. *ArXiv Mathematics e-prints*, August 2005.

[19] Jack Goldfeather and Victoria Interrante. A novel cubic-order algorithm for approximating principal direction vectors. *ACM Trans. Graph.*, 23:45–63, January 2004.

[20] Eitan Grinspun, Yotam Gingold, Jason Reisman, and Denis Zorin. Computing discrete shape operators on general meshes. *Computer Graphics Forum*, 25:547–556, 2006.

[21] Peter Gruber. Optimal configurations of finite sets in Riemannian 2-manifolds. *Geometriae Dedicata*, 84:271–320, 2001. 10.1023/A:1010358407868.

[22] Peter M. Gruber. A short analytic proof of Fejes Tóth's theorem on sums of moments. *Aequationes Mathematicae*, 58:291–295, 1999.

[23] Klaus Hildebrandt, Konrad Polthier, and Wardetzky Max. On the convergence of metric and geometric properties of polyhedral surfaces. *Geometriae Dedicata*, 123:89–112, 2006.

[24] Anil Nirmal Hirani. *Discrete exterior calculus.* PhD thesis, California Institute of Technology, Pasadena, CA, USA, 2003. AAI3086864.

[25] Xiangmin Jiao and Hongyuan Zha. Consistent computation of first- and second-order differen-

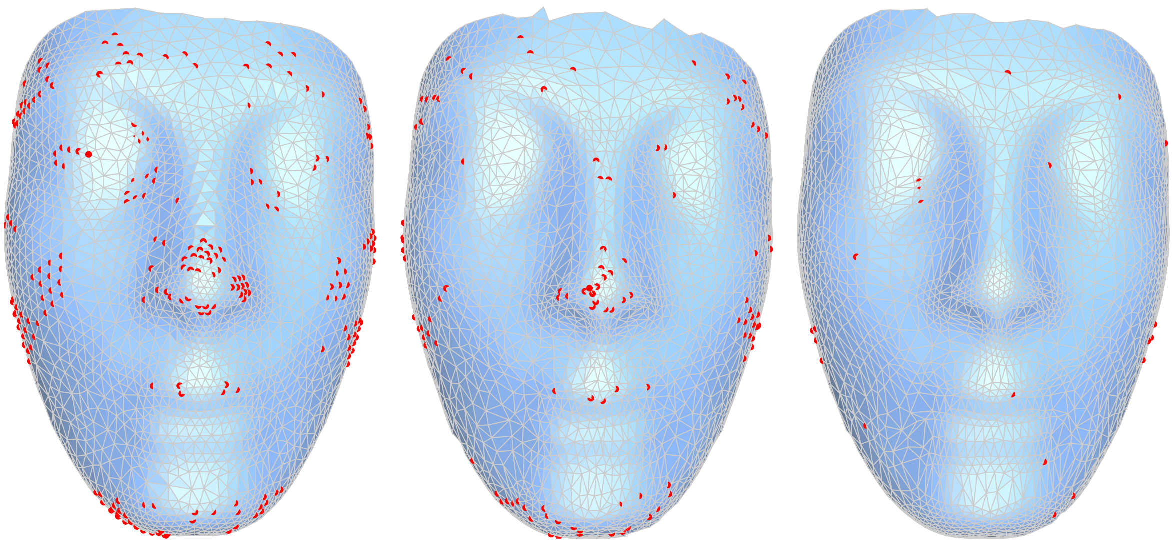


Figure 9: Nefertiti model: meshing results with 2,048 vertices. From left to right: isotropic CVT [50], discrete anisotropic CVT [46] and our method. The number of non shape-preserving vertices are 225, 112 and 21, respectively.

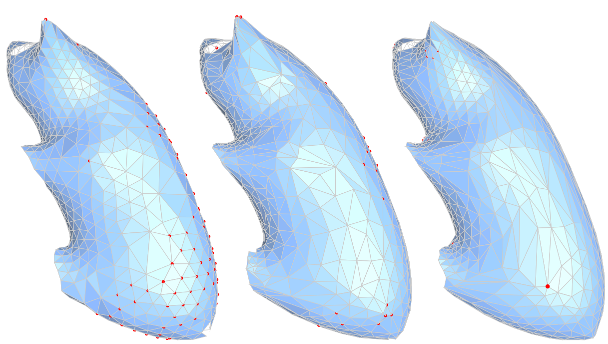


Figure 10: Pig model: meshing results with 1,024 vertices. From left to right: isotropic CVT [50], discrete anisotropic CVT [46] and our method. The number of non shape-preserving vertices are 120, 30 and 12, respectively.

tial quantities for surface meshes. In *Proceedings of the 2008 ACM symposium on Solid and physical modeling*, SPM '08, pages 159–170, New York, NY, USA, 2008. ACM.

[26] Francois Labelle and Jonathan Richard Shewchuk. Anisotropic Voronoi diagrams and guaranteed-quality anisotropic mesh generation. In *Proceedings of the nineteenth annual symposium on Computational geometry (SCG 03)*, pages 191–200, 2003.

[27] C.K. Lee. Automatic metric advancing front triangulation over curved surfaces. *Engineering Computations*, 17(1):48–74, 2000.

[28] Greg Leibon and David Letscher. Delaunay triangulations and Voronoi diagrams for Riemannian manifolds. In *SCG '00: Proceedings of the sixteenth annual symposium on Computational geometry*, pages 341–349, 2000.

[29] Bruno Lévy and Yang Liu. L_p centroidal Voronoi tessellation and its applications. *ACM Trans. Graph.*, 29:119:1–119:11, July 2010.

[30] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming: Series A and B*, 45(3):503–528, 1989.

[31] Yang Liu, Helmut Pottmann, Johannes Wallner, Yong-Liang Yang, and Wenping Wang. Geometric modeling with conical meshes and devel-

opable surfaces. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, pages 681–689, New York, NY, USA, 2006. ACM.

[32] Yang Liu, Wenping Wang, Bruno Lévy, Feng Sun, Dong-Ming Yan, Lin Lu, and Chenglei Yang. On centroidal Voronoi tessellation – energy smoothness and fast computation. *ACM Trans. Graph.*, 28:101:1–101:17, September 2009.

[33] D. S. Meek and D. J. Walton. On surface normal and Gaussian curvature approximations given data sampled from a smooth surface. *Comput. Aided Geom. Des.*, 17:521–543, July 2000.

[34] M. Meyer, M. Desbrun, P. Schröder, and A. Barr. Discrete differential geometry operators for triangulated 2-manifolds. In *International Workshop on Visualization and Mathematics*, pages 35–57, 2002.

[35] J. M. Morvan and B. Thibert. On the approximation of a smooth surface with a triangulated mesh. *Computational Geometry*, 23(3):337–352, 2002.

[36] J. M. Morvan and B. Thibert. On the approximation of the area of a surface. Technical Report RR4375, INRIA, 2002.

[37] Hoa Nguyen, John Burkardt, Max Gunzburger nad Lili Ju, and Yuki Saka. Constrained CVT meshes and a comparison of triangular mesh generators. *Computational Geometry: Theory and Applications*, 42:1–19, 2009.

[38] Ulrich Pinkall, Strasse D. Juni, and Konrad Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2:15–36, 1993.

[39] Helmut Pottmann, Alexander Schiftner, Pengbo Bo, Heinz Schmiedhofer, Wenping Wang, Niccolo Baldassini, and Johannes Wallner. Freeform surfaces from single curved panels. *ACM Trans. Graph.*, 27(3):76:1–76:10, August 2008.

[40] Steven Rosenberg. *The Laplacian on a Riemannian Manifold: an Introduction to Analysis on Manifolds*. Cambridge University Press, Cambridge, 1997.

[41] S. Rusinkiewicz. Estimating curvatures and their derivatives on triangle meshes. In *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on*, pages 486 – 493, sept. 2004.

[42] Hermann Amandus Schwarz. Sur une définition erronée de l'aire d'une surface rourbe. *Gesammelte Mathematische Abhandlungen*, 2:309–311, 1890.

[43] Olga Sorkine. Differential representations for mesh processing. *Computer Graphics Forum*, 25(4):789–807, December 2006.

[44] Feng Sun, Yi-King Choi, Wenping Wang, Dong-Ming Yan, Yang Liu, and Bruno Lévy. Obtuse triangle suppression in anisotropic meshes. *Computer Aided Geometric Design*, 28(9):537 – 548, 2011.

[45] G. Taubin. Estimating the tensor of curvature of a surface from a polyhedral approximation. In *Proceedings of the Fifth International Conference on Computer Vision*, ICCV '95, pages 902–907, Washington, DC, USA, 1995. IEEE Computer Society.

[46] Sébastien Valette, Jean-Marc Chassery, and Rémy Prost. Generic remeshing of 3D triangular meshes with metric-dependent discrete Voronoi diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 14(2):369–381, 2008.

[47] Duo Wang, Bryan Clark, and Xiangmin Jiao. An analysis and comparison of parameterization-based computation of differential quantities for discrete surfaces. *Computer Aided Geometric Design*, 26(5):510 – 527, 2009.

[48] Max Wardetzky, Miklós Bergou, David Harmon, Denis Zorin, and Eitan Grinspun. Discrete quadratic curvature energies. *Comput. Aided Geom. Des.*, 24:499–518, November 2007.

[49] Guoliang Xu. Discrete Laplace-Beltrami operators and their convergence. *Comput. Aided Geom. Des.*, 21:767–784, October 2004.

[50] Dong-Ming Yan, Bruno Lévy, Yang Liu, Feng Sun, and Wenping Wang. Isotropic remeshing with fast and exact computation of restricted Voronoi diagram. *Computer Graphics Forum*, 28(5):1445–1454, 2009.