Pacific Graphics 2019
C. Theobalt, J. Lee, and G. Wetzstein
(Guest Editors)

*Volume 38 (2019), Number 7*

# Anisotropic Surface Remeshing without Obtuse Angles

Qun-Ce Xu[1] Dong-Ming Yan[2†] Wenbin Li[1] Yong-Liang Yang[1]

[1]University of Bath, UK [2]NLPR, Institute of Automation, CAS, China

**Abstract**

*We present a novel anisotropic surface remeshing method that can efficiently eliminate obtuse angles. Unlike previous work that can only suppress obtuse angles with expensive resampling and Lloyd-type iterations, our method relies on a simple yet efficient connectivity and geometry refinement, which can not only remove all the obtuse angles, but also preserves the original mesh connectivity as much as possible. Our method can be directly used as a post-processing step for anisotropic meshes generated from existing algorithms to improve mesh quality. We evaluate our method by testing on a variety of meshes with different geometry and topology, and comparing with representative prior work. The results demonstrate the effectiveness and efficiency of our approach.*

## 1. Introduction

With the recent development of 3D data acquisition and modeling techniques, triangle mesh becomes the most popular shape representation due to its simplicity, efficiency, and flexibility [BKP*10], benefiting enormous scientific and engineering applications such as scientific visualization, physical simulation, digital entertainment, just to name a few. Mathematically, triangle mesh composed by a set of mesh vertices and edges (faces are enclosed by neighboring edges) is a discrete shape representation, in which the geometric information is embedded by vertex coordinates, and the topological information is encoded by edges connecting incident vertices.

Given the underlying shape, how to generate meshes with desirable geometry and connectivity according to different application requirements has attracted research attention for many years, leading to an active research area called surface remeshing [AUGA08]. The majority of work in this area focuses on isotropic remeshing which aims to achieve meshes with equilateral triangles. Recent works [YW16, HYB*17, WYL*19] further investigate how to suppress large and small angles (especially obtuse angles) on isotropic meshes to favor applications that require robust numerical computation, in particular those relying on non-negative cotangent weights to form Laplacian-Beltrami operator [PP93, CdGDS13, WLY*16].

However, how to prevent obtuse angles for anisotropic meshes adaptive to surface features is much less explored. The most relevant is [SCW*11] that leverages a different (hexagonal Minkowski) metric which is computationally more expensive, and can only suppress but not eliminate obtuse angles. Moreover, due to the reliance on a new metric, the method is not compatible with the widely used Euclidean metric based anisotropic remeshing techniques, restricting its scope for refining existing anisotropic meshes that are slightly sub-optimal.

In this paper, we propose a novel anisotropic remeshing method that directly operates on anisotropic meshes generated from existing remeshing algorithms. Our method consists of two key components, including a *connectivity optimization* which breaks the connectivity barrier (a special mesh structure with an obtuse angle in each triangle due to the defining elliptic metric for remeshing is highly anisotropic, see explanation in Section 3.2.2 and illustration in Figure 2a) for obtuse angle removal, and a *geometry optimization* which refines the distribution of mesh vertices when necessary to ensure acute angles. We test our method on diverse shapes with varying geometry and topology and also compare with prior work. The results and comparisons validate the advantages of our approach.

Overall our work makes two major contributions as follows:

- A novel remeshing method that can eliminate obtuse angles for anisotropic meshes.
- An efficient connectivity and geometry optimization approach that can improve mesh quality without resampling geometry and rebuilding connectivity.

## 2. Related Works

As remeshing has been extensively studied in the geometry processing field for decades, a comprehensive overview on all types of remeshing techinques (e.g., isotropic/anisotropic, uniform/adaptive, triangle/quadrilangle, surface/volume) is beyond the scope of this paper. Interested readers may refer to [AUGA08] for a broad understanding of remeshing techniques, and [BLP*13] for a later survey on quad remeshing. Here we mainly discuss remeshing approaches that are most relevant to ours, including anisotropic triangle remeshing, and angle-specific remeshing.

† Corresponding Author

**Anisotropic triangle remeshing.** Unlike isotropic remeshing aiming at regular face shape (equilateral triangle) and vertex valence (ideally six) of the resultant mesh, anisotropic triangle remeshing focuses on generating triangular face elements that are adaptive to local surface anisotropy. The key problem here is how to evenly distribute new mesh vertices and then regularly construct mesh connectivity according to an anisotropic metric (in contrast to the isotropic Euclidean metric) on the original surface. Similar to the isotropic counterpart, recent anisotropic remeshing works can be classified into three categories according to the strategy for optimizing mesh vertex locations, including *particle optimization* (PO) based, *anisotropic centroidal Voronoi diagram* (ACVD) based, and *anisotropic optimal Delaunay triangulation* (AODT) based methods, respectively.

PO-based methods treat mesh vertices as particles with repulsive forces, and update mesh vertex distribution by particle repulsion until the state of force equilibrium is reached. Lai et al. [LZH*07] utilized an anisotropic feature sensitive (FS) metric in a six dimensional space (vertex position and normal) to define repulsive forces and perform particle repulsion. Then a local parameterization based remeshing technique [SAG03] was adopted to first embed local surface patches in 2D under anisotropic metric, and then construct mesh connectivity using Delaunay triangulation. Connectivity conflicts between patches need to be resolved using Delaunay refinement. Zhong et al. [ZGW*13] reformulated the particle-based anisotropic remeshing in terms of inner products and thus is able to naturally extend to using a given Riemannian metric. *Anisotropic Voronoi diagram* (AVD) [DW05] and restricted Voronoi diagram (RVD) [YLL*09] were used to generate mesh connectivity. In a very recent work, Zhong et al. [ZWL*18] computed a self-intersection free high-dimensional Euclidean embedding from an arbitrary smooth Riemannian metric. Then the problem is converted to isotropic remeshing that can be efficiently solved in the embedding space.

ACVD-based methods rely on explicitly constructing Voronoi diagram under an anisotropic metric and performing Lloyd-type iteration [Llo82] to distribute mesh vertices. Du and Wang [DW05] presented ACVT by generalizing *centroidal Voronoi tessellation* (CVT) to the anisotropic case, and then computed its dual mesh. Valette et al. [VCP08] computed the discrete metric-dependent Voronoi diagram directly on mesh surfaces for anisotropic surface remeshing. Zhong et al. [ZSJG14] conducted global conformal parameterization on surface with simple topology first, and then perform remeshing in 2D using CVT. To overcome the expensive computation of ACVD for each iteration, a novel surface embedding based approach was proposed in [LB13, NLG15]. The problem is first transferred to a 6D space consisting of vertex position and normal (similar to [LZH*07]). Then CVT is performed isotropically in this high-dimensional space and mapped back to 3D to generate the anisotropic mesh. Cai et al. [CGL*17] proposed to use *Principle Component Analysis* (PCA) for asymptotic clustering, which naturally leads to high-quality anisotropic remeshing.

AODT-based approach extends optimal Delaunay triangulation in the isotropic case for anisotropic remeshing [CX04], where the anisotropy is encoded by the Hessian of a convex function. Loseille and Alauzet [LA09] proposed to use local convex functions to cope with an arbitrary anisotropic metric. Fu et al. [FLSG14] constructed a convex function whose Hessian matches the anisotropic metric for each mesh face, and iteratively optimized mesh geometry and connectivity by minimizing the difference between the target convex functions and their piecewise-linear interpolation over the mesh. Budninskiy et al. [BLdG*16] also employed Hessian-based anisotropy to solve the dual problem of ODT, leading to optimal Voronoi tessellation (OVT) that only contains convex cells with straight edges, and admits an embedded dual triangulation that is combinatorially-regular. Xiao et al. [XCC*18] generalized the idea of OVT by *optimal power diagram* (OPD) via functional approximation for anisotropic remeshing.

Although anisotropic surface remeshing has been explored from different aspects, none of the above approaches can completely eliminate all the obtuse triangles after remeshing. Our method can be performed as an efficient post-processing step for anisotropic meshes generated from existing approaches. It can further improve mesh quality by eliminating all the obtuse angles within mesh faces.

**Angle-specific optimization for remeshing.** Almost all prior works on angle-specific optimization for remeshing are specific to isotropic meshes, for which various approaches have been proposed to improve mesh quality, including CVT [LWL*09], ODT [CX04], centroidal Delaunay triangulation [CCW12], etc. The difference among these approaches is how to measure mesh quality and how to reconstruct mesh geometry and connectivity accordingly.

In recent years, a number of approaches focused on further improving isotropic mesh quality by diminishing or even eliminating angles that are too large or too small. Yan and Wonka [YW16] augmented the original CVT formulation with a penalty term that penalizes short Voronoi edges, which in turn avoids obtuse angles in the dual triangulation. Hu et al. [HYB*17] repetitively performed edge collapsing, vertex relocation, and edge splitting to increase the smallest angle of the mesh while bounding the approximation errors and implicitly preserving features. Although the smallest angle could be elevated to a pre-defined threshold (up to $40^\circ$), this method introduces too many obtuse triangles at the same time. Wang et al. [WYL*19] proposed a method which progressively eliminates obtuse triangles and improves small angles at the same time. The former is based on a simple vertex insertion scheme, and the latter is through a vertex removal operation that improves the distribution of small angles. Connectivity optimization and local smoothing further improves the quality of the mesh.

The most relevant work in the anisotropic case is [SCW*11]. It relied on a hexagonal Minkowski metric rather than elliptic metric in place of the Euclidean space to form a suitable triangle layout, which suppresses obtuse angles but has no guarantee of elimination. Additionally, due to the change of metric, this method is computationally expensive and is not compatible to anisotropic meshes generated from elliptic metric by our method.

Different from previous work, our method is based on efficient connectivity and geometry optimization without resampling mesh geometry and rebuilding connectivity. Moreover, all the obtuse angles in the anisotropic mesh can be effectively removed.

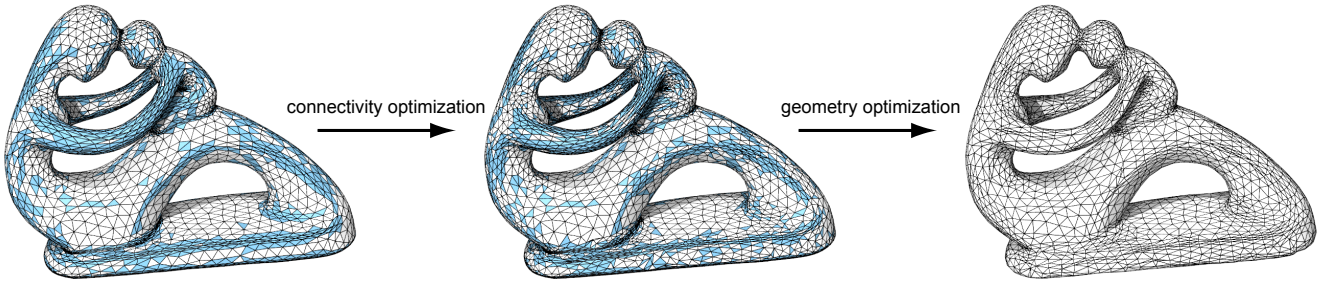



**Figure 1:** *Overview of our method which has two major stages, including connectivity optimization and geometry optimization. From left to right: the input anisotropic mesh generated from [NLG15] with normal lifting weight 0.1, the intermediate mesh after connectivity optimization, the final output mesh with all obtuse angles (highlighted in blue) eliminated.*

## 3. Algorithm

The input to our method is an anisotropic triangle mesh $\mathcal{M}(\mathcal{V},\mathcal{E},\mathcal{F})$ generated from an existing remeshing method, where $\mathcal{V}$, $\mathcal{E}$, $\mathcal{F}$ are the vertex, edge, and face set of mesh $\mathcal{M}$, respectively. Our goal is to output a refined mesh with all obtuse angles eliminated. Note that unlike isotropic meshes that favor angles close to $60^\circ$, for anisotropic meshes, small acute angles commonly exist due to skinny triangles in highly anisotropic regions, thus need to be admitted rather than avoided.

### 3.1. Algorithm overview

As shown in Figure 1, our method contains two major stages. The first stage is connectivity optimization. The basic idea is to adjust the connectivity of the input mesh, such that the mesh connectivity will not be a bottleneck for obtuse angle removal (Section 3.2). Based on the intermediate connectivity optimization result, the second stage is geometry optimization. It aims at optimizing the mesh vertex locations such that all obtuse angles can be removed (Section 3.3).

### 3.2. Connectivity optimization

This subsection describes the algorithmic details of the connectivity optimization stage.

#### 3.2.1. Valence optimization

Given an input anisotropic mesh $\mathcal{M}$, we first perform valence optimization [BK04] using a sequence of edge flipping to improve the regularity of the mesh. The basic idea is to minimize the squared sum of the difference between the valence of each vertex and its corresponding optimal valence, which is 6 for inner vertex, and 4 for boundary vertex, respectively. Note that the edge flipping is only performed if it does not affect the angle quality, i.e., introduce any new obtuse angles. We also perform a sanity check afterwards to see if any inner vertex with valence 4 is left. If so, we further apply an edge collapse to snap it to one of its neighbors that minimizes the largest angle in its one-ring neighborhood. This post process is very useful, as valence 4 vertex could easily lead to obtuse angle incident to that vertex.

#### 3.2.2. Connectivity adjustment

In this step, we efficiently adjust the connectivity of the mesh to break its limitation for obtuse angle removal, making it suitable for geometry optimization in the next stage while preserving the original mesh structure as much as possible.

As discussed in [SCW*11], according to the elliptic metric used in the CVT-based anisotropic remeshing approach, the resultant mesh would have a large number of triangles with obtuse angles in the anisotropic region (see Figure 2a). This is caused by the fact that the defining elliptic metric for surface anisotropy is highly anisotropic in such region. In other words, the metric changes very differently in different directions due to factors such as varying surface normals [NLG15] and normal curvatures [ZGW*13]. As such, the remeshing yields a special mesh structure with an obtuse angle in each triangle bounded by the metric ellipse (iso-distance curve under the elliptic metric). And given such connectivity constraint over the anisotropic region, it is difficult to diminish obtuse angles by geometry optimization on mesh vertices only.

Instead, a different metric called hexagonal Minkowski metric (see Appendix A for more background) is used within the CVT framework to generate mesh with a suitable connectivity to suppress obtuse angles (see Figure 2b). However, the cost is to start the expensive remeshing process all over again without preserving the good triangles with only acute angles. Moreover, the obtuse angles cannot be fully removed but only suppressed due to the existence of many skewed triangles even under the new metric (see Figure 2c).

In this work, we present a novel optimization that locally adjusts the connectivity of the mesh when needed, while avoiding

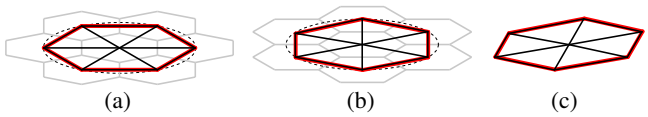



**Figure 2:** *(a) CVT-based remeshing result using elliptic metric. (b) CVT-based remeshing result using Minkowski metric. (c) Skewed triangles with obtuse angles from (b). We show anisotropic mesh (black), its dual Voronoi diagram (gray), metric ellipse (dash black), and metric hexagon (red) with six triangles.*

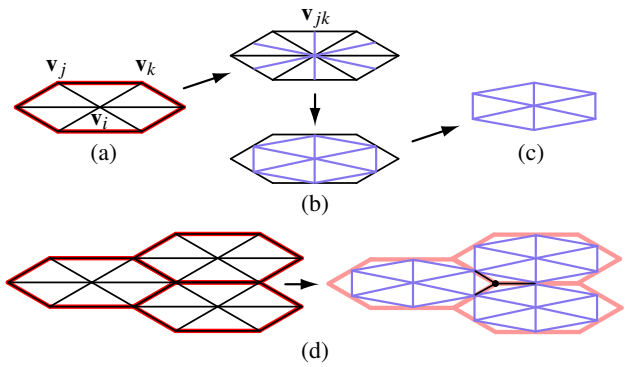



**Figure 3:** *(a) Problematic metric hexagon (p-Hex) whose constituent triangles have obtuse angles. (b) We perform vertex insertion (e.g., insert $\mathbf{v}_{jk}$ between $\mathbf{v}_j$ and $\mathbf{v}_k$), and edge flipping (e.g., flip $e_{ij}$) to construct inscribed hexagon (i-Hex). (c) Inscribed hexagon (i-Hex) with acute triangles. (d) Valence 3 vertex (black dot) will be deleted afterwards.*

expensive Lloyd-type iterations. The basic idea is shown in Figure 3. Inspired by the mesh structure from hexagon Minkowski metric in [SCW*11], we propose to first detect problematic metric hexagons (called p-Hex as in Figure 3a), whose individual constituent triangles have obtuse angles and are hard to optimize directly on geometry. This is done by first marking all triangles with obtuse angles, then grouping neighboring triangles to form hexagons, finally check if each hexagon is a p-Hex. For the last step, we estimate the major and minor axis of the underlying ellipse as follows. We first project the one-ring neighborhood bounded by the hexagon $\mathcal{H}_i$ to the tangent plane of the center vertex $\mathbf{v}_i \in \mathcal{V}$, then we estimate the major and minor axis of the underlying metric ellipse by simply applying PCA of the projected vertices. Finally we check if the incident edges $\{e_{ij}, e_{ik}, \ldots\} \subset \mathcal{E}$ of $\mathbf{v}_i$ are better aligned with the major or minor axis. If it is the former case, we mark $\mathcal{H}_i$ as p-Hex, otherwise $\mathcal{H}_i$ is more like Minkowski hexagon and no connectivity adjustment will be applied.

With p-Hex all marked, we then locally optimize mesh connectivity by constructing an inscribed hexagon (called i-Hex) for each p-Hex (see Figure 3c). More specifically, as shown in Figure 3b, we perform edge splitting by inserting a new vertex $\mathbf{v}_{jk}$ at the middle of each boundary edge $e_{jk}$ of p-Hex $\mathcal{H}_i$. Then we flip $\{e_{ij}, e_{ik}, \ldots\}$ within the one-ring neighborhood bounded by $\mathcal{H}_i$ to form i-Hex $\mathcal{H}_i'$. Note that after all the edge flipping, there would be valence 3 and valence 4 vertices generated depending on different spatial relation between neighboring i-Hexes. Then we simply delete valence 3 vertices (see Figure 3d), and perform valence optimization as in Section 3.2.1 to remove valence 4 vertices and other non-optimal valences.

In this way, we are able to construct Minkowski-like metric hexagons without explicitly performing the CVT-based optimization with Minkowski metric, which is very efficient. Meanwhile, the original mesh structure with all acute angles are also preserved. Note that after connectivity optimization, obtuse angles might still exist in the refined hexagonal structure due to skew distortion (see Figure 2c). But the obtuse angles are usually close to 90° and can be handled by the geometry optimization in the next stage, which is not possible otherwise.

### 3.3. Geometry optimization

The optimized mesh connectivity from Section 3.2 serves as an initialization for geometry optimization, which optimizes the location of mesh vertices on the input mesh in order to remove obtuse angles. However, due to inserting new vertices at the edges of a p-Hex, new triangles with irregular shape can be introduced at the boundary of an i-Hex region (region formed by multiple i-Hexes), making geometry optimization hard to converge. To improve the initialization, for the vertices at the boundary of i-Hex regions and also their neighbors, we perform a fairness refinement based on graph Laplacian, resulting in a better initialization for geometry optimization.

Given a good initialization, a straightforward strategy would be to formulate an optimization problem with mesh vertices as unknowns, where the objective function penalizes obtuse angles while keeping all vertices on the original input mesh. However, due to angles are non-linear w.r.t. vertex coordinates and non-obtuse angle is an inequality constraint (i.e., $\leq 90°$), such objective function would be highly non-linear and strongly constrained, making the formulation and optimization highly challenging.

On the other hand, it is obvious that face angle (enclosed by two edges with three incident vertices) is a local property and only involves three defining vertices. This allows us to employ an efficient local-global optimization approach [BDS*12], where local projections and global fitting are performed alternately to reach the optimum. In each local-global iteration, the local projections take into account the constraints acting on individual mesh elements (e.g., mesh face with non-obtuse angles in our scenario), then the global fitting reconstructs the whole mesh that best approximates the local projected elements in a least squares manner.

More specifically, as our goal is to remove obtuse angles, the local projection should project each mesh face into a constrained space $\overline{\mathcal{T}_o}$ consisting of all triangles without obtuse angles (suppose obtuse triangle space is $\mathcal{T}_o$). If a mesh face has no obtuse angle, it simply projects to itself as it is already in $\overline{\mathcal{T}_o}$. Otherwise a non-linear constrained optimization specific to that face can be performed, which would be inefficient considering the number of faces and the local-global iterations. And one can imagine that the closest point in $\overline{\mathcal{T}_o}$ would be a right triangle which is less desirable than an acute triangle. As such, we propose a simple yet efficient, and also geometrically meaningful way, to perform the local projection.

Our geometric projection is inspired by the typical obtuse triangle

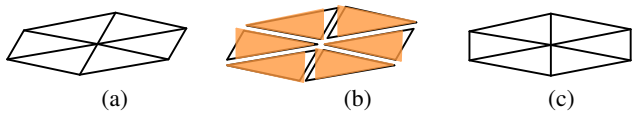



**Figure 4:** *(a) Typical obtuse triangle layout in a metric hexgon. (b) Each obtuse triangle is locally projected to an isosceles triangle with acute angles (in orange). (c) The global fitting step reconstructs the mesh that best approximates the local projections.*

layout in a metric hexagon (see Figure 4a). This is due to skewed distortion of a regular metric hexagon, in which each triangle should be an isosceles triangle (a triangle with at least two equal sides) whose vertex angle is the smallest acute angle. As such, we define the projected counterpart of an obtuse triangle $f_{ijk}$ as an isosceles triangle $f'_{ijk}$ whose vertex angle coincides with the smallest angle of $f_{ijk}$, two legs are with the average length of the corresponding two sides (see Figure 4b). In this way, each of the mesh face $f_{ijk}$ is projected to $f'_{ijk} \in \overline{\mathcal{T}_o}$, where $\mathbf{v}'^{jk}_i$ is the projection of $\mathbf{v}_i$.

In the global fitting step (see Figure 4c), the following objective function (in a least squares form) is optimized to reconstruct the whole mesh that best approximates the projected (non-obtuse) triangles:

$$F_o(\mathcal{V}) = \sum_i \sum_{f_{ijk} \in \mathcal{N}_i} ||\mathbf{v}_i - \mathbf{v}'^{jk}_i||^2, \tag{1}$$

where $\mathcal{N}_i$ is the one-ring neighborhood of $\mathbf{v}_i$.

In practice, the geometry optimization with the above global energy can effectively remove obtuse angles. However, the fairness of the mesh may not be well preserved, leading to unevenly distributed vertices in a local region, especially for those regions between two mesh patches with original and adjusted connectivity, respectively. Hence we also add a regularity term to take into account the fairness of the mesh as follows:

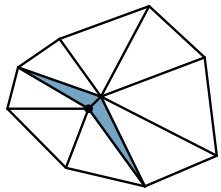
$$F_f(\mathcal{V}) = \sum_i \sum_{\mathbf{v}_j \in \mathcal{N}_i} ||\mathbf{v}_i - \mathbf{v}_j||^2. \tag{2}$$

As a result, the overall global energy has the following form:

$$F(\mathcal{V}) = w_o F_o + w_f F_f, \tag{3}$$

where $w_o$ and $w_f$ are weights that balance the two energy terms. Here we adopt a dynamic weighting scheme that applies $w_o = 1, w_f = 0.01$ at the beginning to regularize more of the optimization, and decay $w_f$ to 0 when getting close to the minimum (only 10% of obtuse angles remain) to add more geometric freedom and ensure convergence (i.e., only non-obtuse angles are left). This is similar to non-rigid registration that enforces more rigidity at first, and allows more shape deformation later [TCL*13]. Also, same as Llyod-type surface remeshing, after the global fitting step, we project the optimized mesh vertices back onto the original mesh to ensure mesh fidelity for each local-global iteration. And the iterative geometry optimization ends when no obtuse angle exists, which is true for all the models tested in our experiments (see Section 4). To demonstrate the effectiveness of $F_o$ and $F_f$ in Eqn. 3, we also run geometry optimization by excluding one of the two terms as shown in Figure 5. It can be seen that obtuse angles cannot be effectively eliminated without $F_o$, and the fairness of the mesh is much worse without $F_f$.

**Remarks.** In practice, we find that the geometry optimization may lead to skinny triangles with very short edges. We simply collapse these edges as in [DVBB13], when the length is less than $1/3$ of the minimum edge of the input mesh. Also, as shown in the inset figure, the only special case where the optimization is trapped into local minimum is when two obtuse angles (highlighted in blue) share the same vertex rather than being alternate as in Figure 4a. To solve this issue, we check such case after each local-global iteration, and perturb the common vertex (highlighted as the black dot) of the two obtuse angles to the barycenter of its one-ring neighborhood to escape from local minimum (as this avoids two neighboring obtuse angles).



## 4. Experimental Results

We evaluate our method by testing on various anisotropic meshes with different geometry and topology. The anisotropic meshes are generated using the available codes provided by the authors in [NLG15]. The results can be found in Figure 10. It can be seen that all the obtuse angles are removed from the input mesh. At the same time, our algorithm preserves the original connectivity without obtuse angles as much as possible. Table 1 presents the statistics of our method based on the standard evaluation criteria for surface remeshing. $\theta_{max}$ is the maximal angle in a mesh. $\theta_{>90^\circ}\%$ indicates

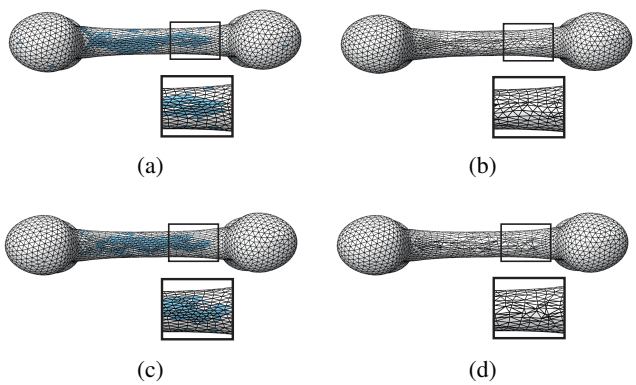



**Figure 5:** *(a) Input anisotropic meshes with obtuse angles. (b) Remeshing result by optimizing both $F_o$ and $F_f$. (c) Result without $F_o$. (d) Result without $F_f$.*

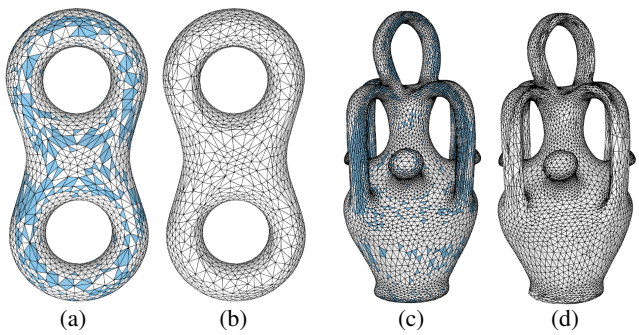



**Figure 6:** *(a) Input anisotropic meshes with obtuse angles (highlighted in blue) generated from [ZGW*13]. (b) Output of our method by eliminating obtuse angles from (a). (c) Input anisotropic meshes with obtuse angles generated from [FLSG14]. (d) Output of our method by eliminating obtuse angles from (c).*

the percentages of the obtuse angles. $Q_{min}$ and $\bar{Q}_{ave}$ measure the regularity of mesh faces given the underlying anisotropic metric of the input mesh. An equilateral triangle (under anisotropic metric here) has $Q(t) = 1$ while a degenerate triangle has $Q(t) = 0$. And the larger of $Q$, the better regularity of the anisotropic triangle.

Note that our method does not rely on any parameter subject to a specific anisotropic remeshing algorithm. Thus anisotropic meshes from other approaches can also serve as input to our method. Figure 6 shows some results by optimizing meshes from a PO-based method [ZGW*13] and an ODT-based method [FLSG14]. Note that the latter already results in Minkowski-type metric hexagons. Hence we do not perform connectivity optimization as the initialization is already good for geometry optimization.

To demonstrate the applicability of our method, we also test on anisotropic meshes with different emphasis on surface anisotropy.

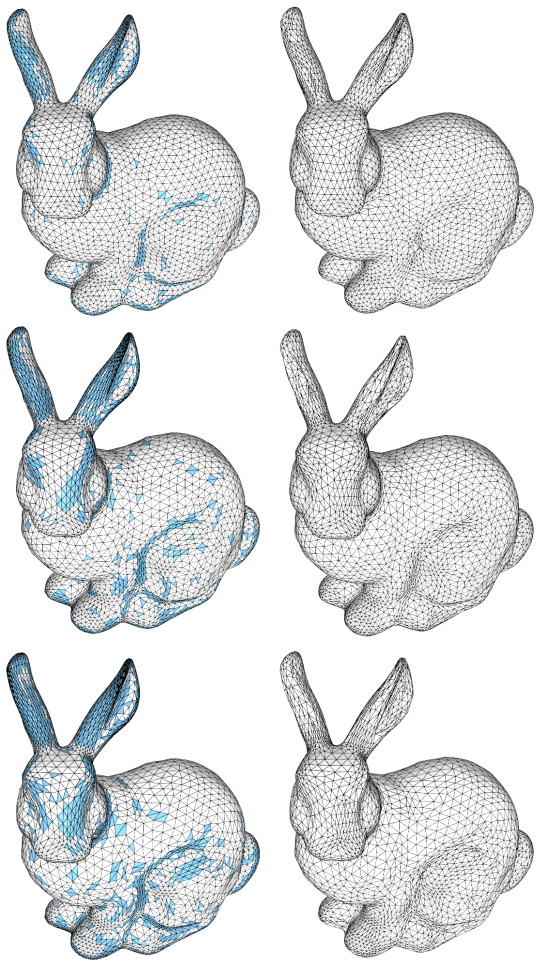
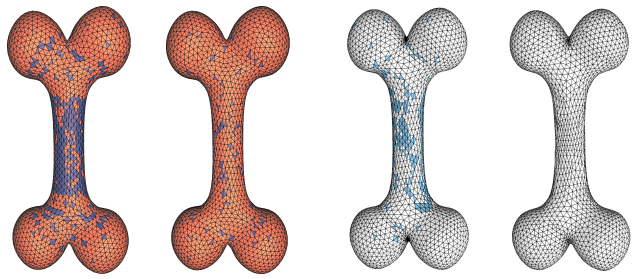


**Figure 7:** *Our results on anisotropic meshes with different level of face anisotropy. From top to bottom: meshes with approximately 5k vertices generated using [NLG15] with normal lifting weight 0.06, 0.1, 0.14, respectively. Left: input anisotropic meshes with obtuse angles highlighted in blue. Right: output of our method with all obtuse angles removed.*



(a) (b) (c) (d)

**Figure 8:** *Comparison between our method and [SCW*11] on the bone model. (a) Input mesh with 3000 vertices. (b) Output mesh with obtuse angles only suppressed but not eliminated. (c) Input mesh with 3000 vertices from [NLG15] with normal lifting weight $w = 0.1$. (d) Our result with all obtuse angles removed. (a) and (b) are adopted from [SCW*11].*

In other words, meshes with high anisotropy allows more skinny triangles in anisotropic regions. This can be easily adjusted by tuning the weight of normal lifting in [NLG15], where larger weight yields more highly isotropic mesh. We use ascending weights to remesh the same model while keeping the number of vertices fixed, resulting three anisotropic meshes. Our method can effectively eliminate obtuse angles on all of them (see Figure 7).

We further compare our method with the only work [SCW*11] that can suppress obtuse angles on anisotropic meshes. We use the only mesh model used in that work to make the comparison as in Figure 8. Note that as we cannot get available resources (i.e., codes, binaries, models) from [SCW*11] , we use [NLG15] to generate a similar anisotropic mesh with the same number of vertices. Our method can effectively remove all obtuse angles, while prior work

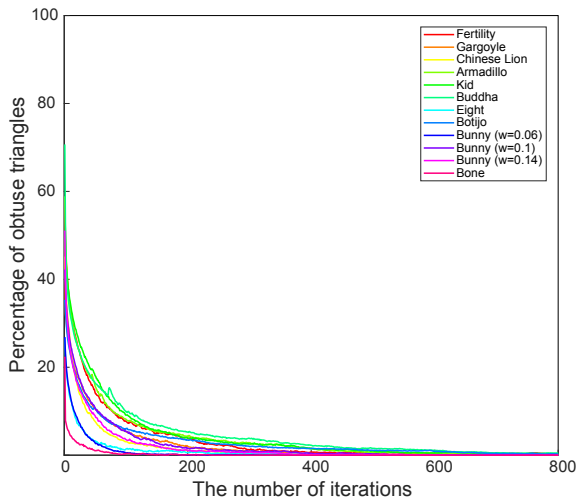



**Figure 9:** *Our geometry optimization converges within hundreds of local-global iterations on different models with thousands of obtuse triangles.*

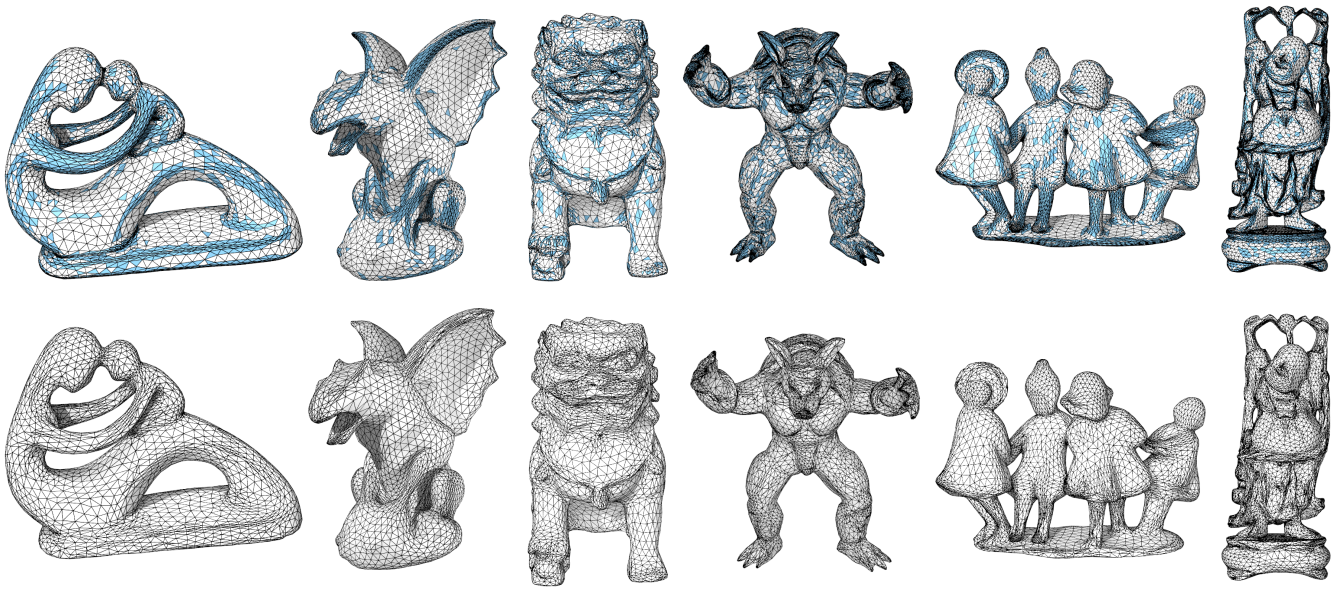


**Figure 10:** *Our results on various anisotropic meshes generated by normal lifting [NLG15] (with fixed lifting weight $w = 0.1$). Top row: the input anisotropic meshes with obtuse angles highlighted in blue. Bottom row: the output of our method with all obtuse angles removed.*

| Model | $\|\mathcal{V}\|$ | $\|\mathcal{F}\|$ | $Q_{min}$ | $Q_{avg}$ | $\theta_{max}$ | $\theta_{>90^\circ}\%$ | time |
|---|---|---|---|---|---|---|---|
| Fertility (Figure 1, 10) | 4000/4262 | 8012/8536 | 0.3055/0.1521 | 0.7237/0.7273 | 171.18/90.00 | 36.75/0.00 | 20 |
| Gargoyle (Figure 10) | 4988/5164 | 9964/10305 | 0.0096/0.0976 | 0.7004/0.6881 | 178.21/89.99 | 45.04/0.00 | 34 |
| ChineseLion (Figure 10) | 8234/8238 | 16464/16472 | 0.2279/0.1059 | 0.7084/0.6816 | 176.59/90.00 | 42.60/0.00 | 42 |
| Armadillo (Figure 10) | 9782/7778 | 19560/15552 | 0.1058/0.0781 | 0.6648/0.6434 | 179.56/89.99 | 56.18/0.00 | 87 |
| Kid (Figure 10) | 8004/8403 | 16036/16834 | 0.0999/0.0840 | 0.6949/0.6602 | 175.54/90.00 | 46.67/0.00 | 75 |
| Buddha (Figure 10) | 17055/12442 | 34142/24904 | 0.0042/0.0785 | 0.6666/0.6295 | 179.92/90.00 | 68.24/0.00 | 186 |
| Eight (Figure 6) | 2000/2114 | 4004/4232 | – / – | – / – | 152.80/89.99 | 25.19/0.00 | 6 |
| Botijo (Figure 6) | 11118/9986 | 22252/19988 | – / – | – / – | 160.31/90.00 | 24.85/0.00 | 71 |
| Bunny (Figure 7 top) | 5000/5306 | 9996/10608 | 0.3061/0.1852 | 0.7723/0.7427 | 160.77/90.00 | 23.33/0.00 | 11 |
| Bunny (Figure 7 middle) | 5000/5116 | 9996/10228 | 0.2166/0.1196 | 0.7198/0.6605 | 163.78/89.99 | 37.41/0.00 | 20 |
| Bunny (Figure 7 bottom) | 4991/4917 | 9978/9830 | 0.1585/0.1453 | 0.6227/0.6215 | 174.59/90.00 | 46.67/0.00 | 23 |
| Bone (Figure 8) | 3000/3001 | 5996/5998 | 0.4409/0.2765 | 0.8119/0.8252 | 169.25/89.99 | 17.91/0.00 | 14 |

**Table 1:** *The statistics of our method on different models. For simplicity, n1/n2 are measurements on input/out mesh. $|\mathcal{V}|$ is the number of vertices; $|\mathcal{F}|$ represents the number of triangles. Computational time is measured in seconds. Other measurements are explained in the first paragraph of Section 4. Note that ' – / – ' is due to the measurement cannot be made (see Discussion and Limitation at the end of Section 4), and '/' is not for division.*

can only reduce the number to a relatively small amount (4.6% obtuse triangles as reported in the paper). Also, the efficiency is discussed as a limitation in [SCW*11] as several minutes are needed to compute a mesh of several thousand vertices. In contrast, our method only takes 14 seconds on mesh with the same size.

**Performance.** We implement our method using C++ on a Windows laptop with 2.8 GHz CPU and 8 GB RAM. The ANN library [MA10] is used for closest point query. The majority of the computational time is due to the geometry optimization in Section 3.3, which is primarily affected by the number of obtuse angles but not the total number of resampled vertices. The typical computational time is less than one minute for meshes with several thousands of vertices, which is much faster than [SCW*11] (even on a more powerful PC with a 3.16 GHz Xeon CPU and 8 GB RAM). Figure 9 shows the convergence curves of geometry optimization on different models. It can be seen that for models with thousands of obtuse triangles, the optimization typically converges within hundreds of local-global iterations.

**Discussion and limitation.** Our method is very general as it is not specific to a certain type of anisotropic meshes (e.g., PO-based, CVT-based, ODT-based). Thus it can be applied directly as a post-processing step with only an isotropic mesh as input. No additional prior knowledge is required, such as anisotropic metric, remeshing strategy, etc. The limitation is that the face anisotropy is hard to

preserve or even evaluate after applying our method. We can evaluate the face anisotropy only if we know under what anisotropic metric the input anisotropic mesh is generated, which is often not available. This is different from isotropic mesh where the face isotropy can be easily evaluated based on Euclidean metric. Since our method only optimizes the general fairness using graph Laplacian, the face anisotropy is slightly worse after removing obtuse angles (see Table 1), which in turn affects the feature preservation in highly anisotropic region with lots of obtuse angles initially.

Another limitation is that we cannot explicitly control the number of vertices on the resultant mesh. This is due to the fact that our method involves not just local connectivity change such as edge flip, vertex collapse/insertion, but more global change during connectivity optimization (see Section 3.2.2). Short edges also need to be collapsed during geometry optimization to avoid nearly degenerated triangles.

Also, similar to previous works on isotropic meshes [YW16, HYB*17, WYL*19], our method has no hard guarantee for complete obtuse angle removal. However, with carefully designed connectivity and geometry optimization, it is empirically effective to eliminate all obtuse angles (see Table 1).

## 5. Conclusion and Future Work

In this work, we present a novel anisotropic remeshing method that can effectively improve the mesh quality by removing all the obtuse angles. This is based on an efficient connectivity and geometry optimization that better preserves the original mesh structure without resampling mesh vertices and fully rebuilding the mesh connectivity. The proposed method is very general and can be used as an efficient post-processing for anisotropic meshes generated by existing algorithms. We evaluate our method on various models with different geometry and topology. The results and the comparison with prior work demonstrate the advantages of our method.

In the future, we would like to extend the current method to handle models with sharp features (e.g., mechanical parts) by first detecting the feature vertices and then fixing them during optimization. We are also interested in further improving mesh quality by involving more prior knowledge of the input mesh, such as the induced anisotropic metric. By taking into account such prior knowledge during geometry optimization (e.g., using anisotropic surface Laplacian instead of graph Laplacian), we would better preserve the face anisotropy at highly anisotropic regions.

## Acknowledgments

This work is partially funded by CAMERA, the RCUK Centre for the Analysis of Motion, Entertainment Research and Applications (EP/M023281/1), the National Natural Science Foundation of China (61772523), and the Beijing Natural Science Foundation (L182059).

## Appendix A:

Given an ellipse $E$ in 2D Euclidean space centered at the origin $\mathbf{o}$, let $H$ be an affinely regular hexagon inscribed in $E$ (see Figure 11b ). The hexagon $H$ defines a vector norm (denoted by $||\mathbf{v}||_H$) in the way that $||\mathbf{v}||_H = 1$ for any vector $\mathbf{v}$ on $H$, otherwise $||k\mathbf{v}||_H = |k| \cdot ||\mathbf{v}||_H$ ($\forall k \in \mathbb{R}$). Then $H$ is called a unit hexagon. The norm $||\mathbf{v}||_H$ is called the *hexagonal Minkowski metric*, under which the iso-distance curves from the origin are hexagons. This is different from the elliptic metric $||\mathbf{v}||_E = \mathbf{v}^T\mathbf{M}\mathbf{v}$ ($\mathbf{M}$ is a positive definite matrix) with iso-distance curves as ellipses (see Figure 11a).

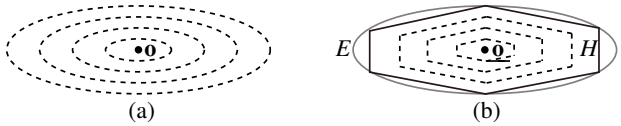



**Figure 11:** *(a) The iso-distance curves of the elliptic metric from the origin* **o**. *(b) The hexagonal Minkowski metric defined by an affinely regular hexagon H (black) inscribed in an ellipse E (gray), and its iso-distance curves (dash black).*

## References

[AUGA08] ALLIEZ P., UCELLI G., GOTSMAN C., ATTENE M.: *Recent Advances in Remeshing of Surfaces*. Springer Berlin Heidelberg, 2008, pp. 53–82. 1

[BDS*12] BOUAZIZ S., DEUSS M., SCHWARTZBURG Y., WEISE T., PAULY M.: Shape-up: Shaping discrete geometry with projections. *Comput. Graph. Forum 31*, 5 (2012), 1657–1667. 4

[BK04] BOTSCH M., KOBBELT L.: A remeshing approach to multiresolution modeling. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing* (2004), SGP '04, pp. 185–192. 3

[BKP*10] BOTSCH M., KOBBELT L., PAULY M., ALLIEZ P., LÉVY B.: *Polygon Mesh Processing*. AK Peters / CRC Press, 2010. 1

[BLdG*16] BUDNINSKIY M., LIU B., DE GOES F., TONG Y., ALLIEZ P., DESBRUN M.: Optimal voronoi tessellations with hessian-based anisotropy. *ACM Trans. Graph. 35*, 6 (2016), 242:1–242:12. 2

[BLP*13] BOMMES D., LÉVY B., PIETRONI N., PUPPO E., SILVA C., TARINI M., ZORIN D.: Quad-mesh generation and processing: A survey. *Computer Graphics Forum 32*, 6 (2013), 51–76. 1

[CCW12] CHEN Z., CAO J., WANG W.: Isotropic surface remeshing using constrained centroidal delaunay mesh. *Computer Graphics Forum 31*, 7 (2012), 2077–2085. 2

[CdGDS13] CRANE K., DE GOES F., DESBRUN M., SCHRÖDER P.: Digital geometry processing with discrete exterior calculus. In *ACM SIGGRAPH 2013 Courses* (2013), pp. 7:1–7:126. 1

[CGL*17] CAI Y., GUO X., LIU Y., WANG W., MAO W., ZHONG Z.: Surface approximation via asymptotic optimal geometric partition. *IEEE Trans. on Vis. and Comp. Graphics 23*, 12 (2017), 2613–2626. 2

[CX04] CHEN L., XU J.-C.: Optimal delaunay triangulations. *Journal of Computational Mathematics 22*, 2 (2004), 299–308. 2

[DVBB13] DUNYACH M., VANDERHAEGHE D., BARTHE L., BOTSCH M.: Adaptive Remeshing for Real-Time Mesh Deformation. In *Eurographics Short Papers Proceedings* (2013), pp. 29–32. 5

[DW05] DU Q., WANG D.: Anisotropic centroidal voronoi tessellations and their applications. *SIAM Journal on Scientific Computing 26*, 3 (2005), 737–761. 2

[FLSG14] FU X.-M., LIU Y., SNYDER J., GUO B.: Anisotropic simplicial meshing using local convex functions. *ACM Trans. Graph. 33*, 6 (2014), 182:1–182:11. 2, 5, 6

[HYB*17] HU K., YAN D., BOMMES D., ALLIEZ P., BENES B.: Error-bounded and feature preserving surface remeshing with minimal angle improvement. *IEEE Transactions on Visualization and Computer Graphics 23*, 12 (2017), 2560–2573. 1, 2, 8

[LA09] LOSEILLE A., ALAUZET F.: Optimal 3d highly anisotropic mesh adaptation based on the continuous mesh framework. In *Proceedings of the 18th International Meshing Roundtable* (2009), pp. 575–594. 2

[LB13] LÉVY B., BONNEEL N.: Variational anisotropic surface meshing with voronoi parallel linear enumeration. In *International Meshing Roundtable* (2013), pp. 349–366. 2

[Llo82] LLOYD S.: Least squares quantization in pcm. *IEEE Transactions on Information Theory 28*, 2 (1982), 129–137. 2

[LWL*09] LIU Y., WANG W., LÉVY B., SUN F., YAN D.-M., LU L., YANG C.: On centroidal voronoi tessellation—energy smoothness and fast computation. *ACM Trans. Graph. 28*, 4 (2009), 101:1–101:17. 2

[LZH*07] LAI Y.-K., ZHOU Q.-Y., HU S.-M., WALLNER J., POTTMANN H.: Robust feature classification and editing. *IEEE Transactions on Visualization and Computer Graphics 13*, 1 (2007), 34–45. 2

[MA10] MOUNT D., ARYA S.: ANN: Approximate nearest neighbors, 2010. 7

[NLG15] NIVOLIERS V., LÉVY B., GEUZAINE C.: Anisotropic and feature sensitive triangular remeshing using normal lifting. *Journal of Computational and Applied Mathematics 289* (2015), 225 – 240. 2, 3, 5, 6, 7

[PP93] PINKALL U., POLTHIER K.: Computing discrete minimal surfaces and their conjugates. *EXPERIMENTAL MATHEMATICS 2* (1993), 15–36. 1

[SAG03] SURAZHSKY V., ALLIEZ P., GOTSMAN C.: Isotropic Remeshing of Surfaces: a Local Parameterization Approach. In *International Meshing Roundtable* (2003), pp. 215–224. 2

[SCW*11] SUN F., CHOI Y.-K., WANG W., YAN D.-M., LIU Y., LÉVY B.: Obtuse triangle suppression in anisotropic meshes. *Computer Aided Geometric Design 28*, 9 (2011), 537 – 548. 1, 2, 3, 4, 6, 7

[TCL*13] TAM G. K., CHENG Z.-Q., LAI Y.-K., LANGBEIN F., LIU Y., MARSHALL A. D., MARTIN R., SUN X., ROSIN P.: Registration of 3d point clouds and meshes: A survey from rigid to nonrigid. *IEEE Transactions on Visualization and Computer Graphics 19*, 7 (2013), 1199–1217. 5

[VCP08] VALETTE S., CHASSERY J.-M., PROST R.: Generic remeshing of 3D triangular meshes with metric-dependent discrete Voronoi diagrams. *IEEE Trans. on Vis. and Comp. Graphics 14*, 2 (2008), 369–381. 2

[WLY*16] WANG X., LE T. H., YING X., SUN Q., HE Y.: User controllable anisotropic shape distribution on 3d meshes. *Computational Visual Media 2*, 4 (2016), 305–319. 1

[WYL*19] WANG Y., YAN D., LIU X., TANG C., GUO J., ZHANG X., WONKA P.: Isotropic surface remeshing without large and small angles. *IEEE Trans. on Vis. and Comp. Graphics 25*, 7 (2019), 2430–2442. 1, 2, 8

[XCC*18] XIAO Y., CHEN Z., CAO J., ZHANG Y. J., WANG C.: Optimal power diagrams via function approximation. *Computer-Aided Design 102* (2018), 52–60. 2

[YLL*09] YAN D., LÉVY B., LIU Y., SUN F., WANG W.: Isotropic remeshing with fast and exact computation of restricted voronoi diagram. *Computer Graphics Forum 28*, 5 (2009), 1445–1454. 2

[YW16] YAN D., WONKA P.: Non-obtuse remeshing with centroidal voronoi tessellation. *IEEE Transactions on Visualization and Computer Graphics 22*, 9 (2016), 2136–2144. 1, 2, 8

[ZGW*13] ZHONG Z., GUO X., WANG W., LÉVY B., SUN F., LIU Y., MAO W.: Particle-based anisotropic surface meshing. *ACM Trans. Graph. 32*, 4 (2013), 99:1–99:14. 2, 3, 5, 6

[ZSJG14] ZHONG Z., SHUAI L., JIN M., GUO X.: Anisotropic surface meshing with conformal embedding. *Graphical Models 76*, 5 (2014), 468–483. 2

[ZWL*18] ZHONG Z., WANG W., LÉVY B., HUA J., GUO X.: Computing a high-dimensional euclidean embedding from an arbitrary smooth riemannian metric. *ACM Trans. Graph. 37*, 4 (2018), 62:1–62:16. 2