Non-obtuse Remeshing with Centroidal Voronoi Tessellation

Dong-Ming Yan, Peter Wonka

Abstract—We present a novel remeshing algorithm that avoids triangles with small (acute) angles and those with large (obtuse) angles. Our solution is based on an extension of *Centroidal Voronoi Tesselation* (CVT). We augment the original CVT formulation with a penalty term that penalizes short Voronoi edges, while the CVT term helps to avoid small angles. Our results show significant improvements in remeshing quality over the state of the art.

1 Introduction

Polygonal geometry processing has contributed many useful techniques to the generation, manipulation, and manufacture of three-dimensional (3D) models. Examples are mesh smoothing, mesh deformation, parameterization, surface approximation, and mesh segmentation [1]. A significant number of these techniques is based on solving PDEs with the Laplacian matrix involved in the computation. Many recent remeshing techniques therefore focus on eliminating triangles with small angles. The condition number of the Laplacian matrix is directly affected by such small angles and even a single bad triangle can have a big impact on the computation [2]. Somewhat surprisingly, there is very little work on triangular remeshing that avoids large (obtuse) angles, even though obtuse angles have a similar bad impact on the computation. For example, the correctness of the fast-marching method depends on the quality of the mesh. Obtuse angles in the mesh might violate the order of processing the nodes. The defacto standard to set up the Laplacian matrix is to use the cotan Laplacian [3]. The cotan Laplacian is guaranteed to yield positive values if no obtuse triangles are present. Otherwise, negative values might occur, requiring special treatment in many common algorithms, e.g., [4].

In this paper, we present a novel remeshing technique that avoids both small (acute) angles ($< 30^\circ$) and large (obtuse) angles ($\geq 90^\circ$). Our main idea is to add an additional penalty term to improve state-of-the-art *Centroidal Voronoi Tesselation* (CVT)-based remeshing. To prevent obtuse angles, we penalize short edges in the dual mesh, the Voronoi diagram. Our results show that we can produce very good remeshing results on a wide range of models and significantly improve upon the state of the art.

- *D.-M. Yan is with King Abdullah University of Science and Technology, Thuwal 23955-6900, Saudi Arabia, and the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China. E-mail: yandongming@gmail.com.*
- *P. Wonka is with King Abdullah University of Science and Technology, Thuwal 23955-6900, Saudi Arabia, and Arizona State University, Tempe, AZ 85287-8809, USA. E-mail: pwonka@gmail.com.*

2 Related Work

There are many techniques to remesh a surface, e.g., mesh simplification [5], mesh optimization with local operations [6], the active-front based method [7], Delaunay insertion algorithms [8] and variational approaches. The remeshing techniques can also be classified as isotropic or anisotropic [9], [10]. In this paper, we focus on isotropic remeshing. A complete survey of all remeshing techniques is beyond the scope of this paper. We concentrate on approaches to isotropic remeshing that are based on CVT. We refer the reader to a survey paper [11] for more details about remeshing and a text book [12] for the fundamentals of the Delaunay triangulation and the Voronoi diagram.

Centroidal Voronoi tessellation [13] has been used as starting point for many remeshing techniques. The difference between the available methods is in how the Voronoi diagram on surfaces is approximated. Parameterization-based approaches [14], [15], [16], [17] first parameterize the 3D mesh locally or globally, and then perform Lloyd iterations [18] in the two-dimensional (2D) parameter domain and finally map the 2D samples back to the original 3D surface. Rong et al. [19] propose the use of hyperbolic space as the parameter domain. Parametrization-based approaches are fast and efficient, but they suffer from the distortion introduced by the parameterization. Instead of parametrizing the 3D surface, Valette et al. [20] directly cluster triangles of a 3D mesh and compute an approximated CVT on the mesh surface.

Another way of computing CVT on surfaces is to use the centroidal *Geodesic Voronoi Diagram* (GVD) [21], [22], [23]. Although the centroidal GVD can sometimes generate good-quality meshes, there is no direct link between the meshing quality and the centroidal GVD, since the quality of the remeshed surfaces is measured with the Euclidean metric. While

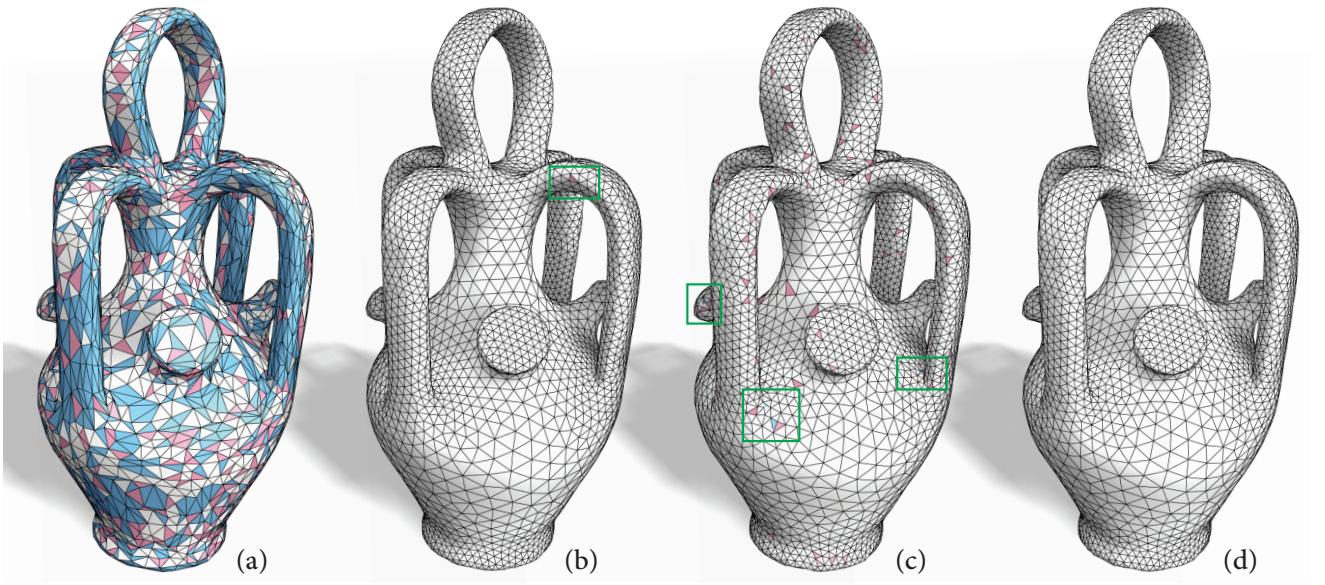


Fig. 1. Non-obtuse remeshing of the Botijo model with 5K vertices and 10K triangles. The pink triangles are obtuse and the blue triangles have small angles, i.e., less than 30 degrees. (a) Input mesh with 6K triangles; (b) result of CVT after 300 iterations (8 obtuse triangles); (c) result of feature-sensitive CVT after 200 iterations (231 obtuse triangles and 6 triangles with small angles); and (d) our non-obtuse remeshing result with 100 iterations. Our result has no obtuse and no small angles.

future work might generate good remeshing results using geodesic distances, currently the methods based on Euclidean distances create superior results. Moreover, we are not aware of any existing work that uses a geodesic metric to generate non-obtuse remeshings on surfaces.

To build a state-of-the-art remeshing technique, we believe that it is best to compute the exact *Restricted Voronoi Diagram* (RVD) [24] and not to rely on approximations. For efficient optimization of CVT energy, Liu et al. [25] propose to use a Newton-like solver, e.g., L-BFGS. Based on these two techniques, an efficient remeshing framework, which overcomes the drawbacks of previous parameterization-based and discrete clustering-based methods, can be developed. Lévy and Bonneel [26] propose the Voronoi parallel linear enumeration that uses *Approximate Nearest Neighbor* (ANN) search instead of Delaunay triangulation to improve the efficiency of the RVD computation. Lévy [27] further improves the robustness of RVD computation using a combination of arithmetic filters, expansion arithmetics and symbolic perturbation. More recently, Yan et al. [28] propose the localized RVD that enables remeshing with a low number of polygons. The RVD framework is also used for blue-noise sampling and remeshing on surfaces [29], [30], [31], [32], polynomial surface fitting [33] and the modeling of minimal surfaces [34].

A related concept is "well-centered" triangulation in which circumscribed centers have to be located inside the corresponding triangles. Note that "well-centered" and "non-obtuse" triangle meshes are equivalent if vertex weights are not considered. Sieger et al. [35] propose to remove the short Voronoi edges by minimizing the summed squared distances between inscribed and circumscribed centers of each triangle to improve the condition number of the Laplacian matrix. Vanderzee et al. [36] study the theoretical aspect of well-centered triangulations in Euclidean space. Sun et al. [37] propose to reduce the number of obtuse triangles in anisotropic remeshing by replacing the Euclidean metric with a hexagonal Minkowski metric. In the context of weighted triangulation optimization, Mullen et al. [38] propose to use the power diagram and the regular triangulation to compute well-centered primal/dual meshes. More recently, de Goes et al. [4] improve primal/dual meshes using weighted triangulation. However, both methods cannot guarantee the non-obtuse property of the optimized meshes.

However, none of the methods discussed above is able to generate a mesh without obtuse triangles, which is very important in mesh processing, e.g., FEM simulation [39]. Non-obtuse meshing has been investigated in the 2D plane [40], [41], [42], [43], but this work does not directly apply to our 3D remeshing problem. The only approach for isotropic surface remeshing we are aware of is [44], which uses the quadric-based mesh decimation algorithm to generate non-obtuse meshes. Unfortunately, this method cannot prevent small angles. Here, we address the problem of generating high-quality meshes, while avoiding both small angles and large angles.

3 Remeshing Framework

Considering the huge amount of literature on remeshing, it is a daunting task to determine the state of the art. In this section, we propose a combination of multiple published techniques that we believe gives the best results among the currently published methods and we explain why obtuse triangles cannot be avoided. The building blocks we propose to use are: 1) CVT 2) The exact Restricted Voronoi Diagram (RVD) 3) The Feature-Sensitive CVT extension and 4) Optimization based on L-BFGS.

The input is a mesh surface, $\mathcal{M} = \{F, V\}$, and a set of samples, $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$, generated on $\mathcal{M}$. $\mathcal{M}$ is a two-manifold triangular mesh consisting of a set of vertices, V, and faces, F. The goal is to compute the CVT of the samples on the surface and to extract a primal mesh from the optimized samples that satisfies certain properties.

3.1 Centroidal Voronoi tessellation

Centroidal Voronoi tessellation of n distinct points (or *seeds*), $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ in R^3, is a special kind of Voronoi tessellation that minimizes the following energy function [13]:

$$F_{CVT}(\mathbf{X}) = \sum_{i=1}^{n} \int_{\Omega_i} \rho(\mathbf{x}) \|\mathbf{x} - \mathbf{x}_i\|^2 \, d\sigma, \tag{1}$$

where $\Omega_i = \{\mathbf{x} \in R^3 \mid \|\mathbf{x} - \mathbf{x}_i\| \leq \|\mathbf{x} - \mathbf{x}_j\|, \forall j \neq i\}$ is the Voronoi cell of point $\mathbf{x}_i$, and $\rho(\mathbf{x}) \geq 0$ is a density function defined over the domain.

3.2 Restricted Voronoi diagram

To compute the CVT on mesh surfaces, we have to restrict the Voronoi diagram on surfaces, called a restricted Voronoi diagram (RVD) [45]. It is defined as the intersection of the 3D Voronoi diagram, $\Omega = \{\Omega_i\}$, and the mesh surface, i.e., $\Omega_{|\mathcal{M}} = \Omega \cap \mathcal{M} = \{\Omega_{i|\mathcal{M}}\}$. Each triangle in the input surface is split and assigned to the incident Voronoi cells. Fig. 2 shows an example of RVD. The dual of RVD is the primal triangulation, called the restricted Delaunay triangulation (RDT). RDT is the actual output of the remeshing framework.

3.3 Feature-sensitive CVT extension

Feature-sensitive CVT is proposed by [10] to inject normal anisotropy into the original CVT energy function as follows:

$$F_{CVT_{fs}}(\mathbf{X}) = \sum_{i=1}^{n} \sum_{f \in \Omega_{i|\mathcal{M}}} \int_f \rho(\mathbf{x}) \|\mathbf{A}_s(\mathbf{N}^f)(\mathbf{x} - \mathbf{x}_i)\|^2 \, d\sigma, \tag{2}$$

where $\mathbf{N}^f$ is the normal of a triangle, f, used for integration (see Fig. 2) and $\mathbf{A}_s(\mathbf{N}) = (s-1)\mathbf{N}\mathbf{N}^t + \mathbf{I}_{3\times 3}$. The parameter $s \geq 1$ controls the weight of normal anisotropy. We set $s = 5$ in all our experimental results as suggested in [10].

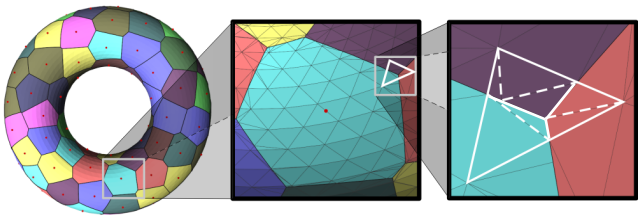


Fig. 2. Illustration of the RVD computation. All input triangles are assigned to their incident Voronoi cell(s). Some triangles are split in the process (middle). Splitting such a triangle produces multiple smaller triangles used for computing the integral of the energy function (dashed white triangles in the right).

3.4 L-BFGS optimization

A straightforward CVT implementation using Lloyd iterations moves vertices to the centroids of their corresponding Voronoi cells. This is a gradient-decent algorithm with linear convergence [13]. Liu et al. [25] prove that the energy term has C^2 smoothness and can therefore be optimized using a quasi-Newton-like solver, such as L-BFGS [46]. Since the newly introduced primal penalty term in Eqn. 3 is also quadratic, our objective function can be minimized efficiently by quasi-Newton solvers.

4 Methodology

In this section, we first describe our novel penalty term and give further implementation details on how the penalty term is embedded into the remeshing framework.

4.1 Penalty term

As discussed in the previous section, the minimization of CVT energy is not sufficient to remove obtuse triangles, especially when using non-uniform density functions. By analyzing the configurations of CVT, we can observe that the longest edge of an obtuse triangle is always dual to a short Voronoi edge. An example is shown in Fig. 3. Inspired by this simple observation, we propose to add a penalty term in the CVT energy function for the primal triangulation RDT. This term penalizes the short Voronoi edges. The modified CVT energy function for eliminating obtuse triangles is defined as follows:

$$F(\mathbf{X}) = F_{CVT_{fs}}(\mathbf{X}) + \lambda \, R(\mathbf{X}), \tag{3}$$

where $R(\mathbf{X})$ is the penalty term designed for preventing short Voronoi edges. It is defined as

$$R(\mathbf{X}) = \sum_{i=1}^{n} \| \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in RDT} w_{i,j}(\mathbf{x}_i - \mathbf{x}_j)\|^2, \tag{4}$$

where $w_{i,j} = \frac{\|(\mathbf{x}_i, \mathbf{x}_j)\|}{\|dual(\mathbf{x}_i, \mathbf{x}_j)\| + \epsilon}$. Here, a small value, ϵ, is used to avoid the denominator becoming zero. Also,

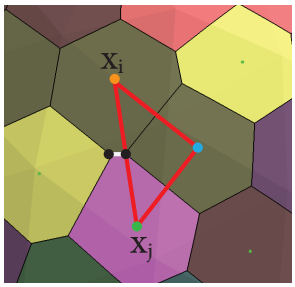


Fig. 3. Illustration of the relationship between a short Voronoi edge and an obtuse triangle. The obtuse triangle is shown in red, and the Voronoi edge that is dual to the longest edge is shown in white.

$(\mathbf{x}_i, \mathbf{x}_j)$ is an edge in the triangulation and $dual(\mathbf{x}_i, \mathbf{x}_j)$ is its dual edge.

In this formulation, the short Voronoi edges will contribute more energy and tend to shrink the corresponding primal edges to make the triangle more uniform. Note that a restricted Voronoi edge, $dual(\mathbf{x}_i, \mathbf{x}_j)$, might not be a single segment. It might consist of a sequence of segments of the neighboring restricted Voronoi cells over the mesh surface. We found that the weight, $\lambda \in [0.1, 1]$, in Eqn. 3 gives satisfying results in all experiments. Fig. 4 shows the effect of varying parameter λ. Note that in a recent paper [47], Renka proposed a simple method that minimizes the variance in the triangle area, which has a similar effect with the CVT formulation. However, as shown in that paper, the obtuse triangles cannot be completely eliminated.

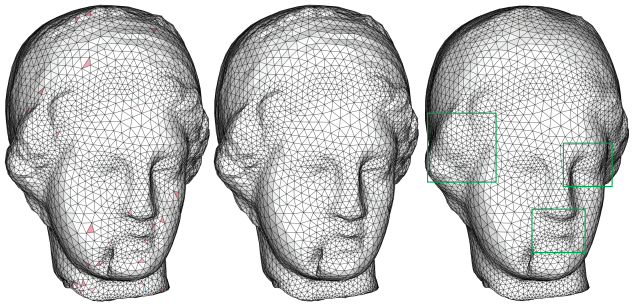


Fig. 4. Illustration of the effectiveness of the weight, λ. Left: $\lambda = 0$ is identical to CVT; middle: an appropriate value of $\lambda = 1$ eliminates obtuse triangles while approximating the input mesh well; right: a larger value of $\lambda = 10$ smoothes out the details.

4.2 Implementation details

We use feature-sensitive CVT introduced in Sec. 3.3 together with our newly introduced penalty term (see Sec 4.1), which improves both the mesh quality and the approximation fidelity. In general, the feature-sensitive CVT extension results in better Hausdorff distance, but sacrifices some of the angle quality of CVT. This can also be observed in our results described in Sec. 5. For the RVD computation, we use the latest improved version presented in [28], which is able to handle surfaces containing thin sheets and self-intersections. There are four main steps in our framework.

Initial sampling. In this step, we randomly generate n initial points on the mesh surface, with respect to a given density function, $\rho(\mathbf{x})$, defined on the mesh vertices and piecewise-linearly interpolated over the mesh triangles. For non-uniform remeshing, we use the local feature size (lfs) [48], which is a popular choice in the literature.

CVT optimization. We optimize the initial point set with the CVT optimizer without the penalty term and the feature-sensitive constraint, typically for 30-50 iterations.

Valence optimization. The CVT energy function is nonconvex and has many local minima. Empirally, we can observe that vertices with valences smaller than 5 or larger than 7, which are called bad-valence vertices in this paper, almost always correpsond to undesirable local minima. It has been proven that the global minimum of the CVT energy function is a regular hexagonal pattern [49] in the plane. While no similar proof exists for surfaces, other authors observed that removing bad-valence vertices by perturbation reduces the CVT energy [50]. We solve this problem by perturbing the bad-valence vertices locally. For each bad-valence vertex, we randomly sample a new vertex inside the triangles in the one-ring neighborhood. The newly sampled vertex is projected onto the input mesh and inserted into the Delaunay triangulation. The old bad-valence vertex is removed. Then Lloyd optimization is performed locally around these vertices (e.g., 3-ring neighborhood), while the others are fixed. Fig. 5 illustrates this process. This simple method works well since neither high valences (> 7) nor low valences (< 5) are stable configurations. In all our tests, this simple perturbation can remove the bad-valence vertices without adding/removing/teleporting other vertices, which helps the global convergence of the optimization.

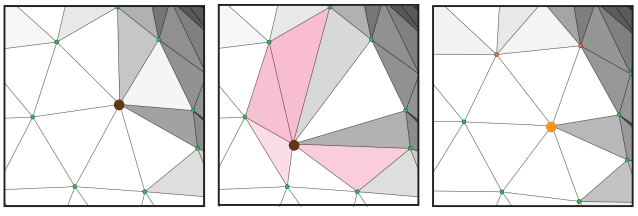


Fig. 5. Localized random optimization of bad-valence vertices. Green: valence 6; Orange: valence 7 and Blue: valence 5. The dark color stands for high/low valences. Left: current configuration. Middle: a valence-8 vertex is perturbed. Right: after 3 steps of local Lloyd optimization, the high-valence vertex is removed.

Obtuse triangle removal. Once we have a well-distributed point set, we add the penalty term in the CVT energy function and start the second stage of the optimization. This removes the obtuse triangles while keeping the good global distribution of the vertices. Finally, we optimize the points again with the feature-sensitive CVT (Eq. 3) and the penalty term for improving the approximation quality.

5 Experimental Results

In this section, we first analyze the remeshing quality of the state-of-the-art CVT framework. Next, we evaluate our algorithm and compare our method with other remeshing techniques. The proposed approach is implemented in C++. The results shown in this section are generated on an Intel X5680 Dual-Core 3.33GHz CPU with 4GB memory and a 64-bit Windows 7 operating system.

Problem analysis. We analyze the remeshing quality of the proposed state-of-the-art remeshing framework with and without the feature-sensitive extension. We observe that the feature-sensitive extension always has significantly more obtuse triangles, so we show only the results without the feature-sensitive extension here. We let the optimizer run until convergence. Fig. 6 shows the number of obtuse triangles in relation to the number of iterations. We can see that CVT cannot eliminate obtuse triangles even after hundreds of iterations, especially in non-uniform remeshing.

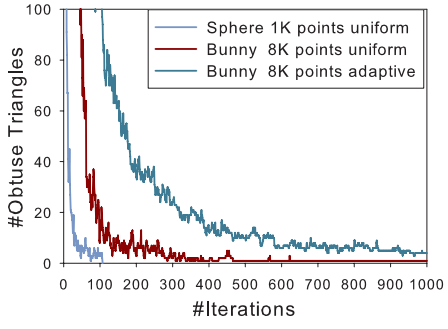


Fig. 6. The number of obtuse triangles in relation to the number of iterations in CVT optimization. We can see that CVT is able to get rid of obtuse triangles in a smooth domain with a small number of points, e.g., the sphere with 1K points. However, if we increase the complexity of the domain, as well as the number of points, CVT is not able to generate non-obtuse remeshing even with thousands of iterations.

Evaluation. We use the names CVT and CVT_{fs} to denote the methods described in Sec. 3 and CVT_{nob} to refer to our new method. We also compare our results with results from *Approximate Centroidal Voronoi Diagram* (ACVD) [20], *Capacity Constrained CVT* (CVT_{cap}) [29], *Maximal Poisson-disk Sampling* (MPS) [30], [53] and *Farthest Point Optimization* (FPO) [31]. In the first comparison, we choose two example models for both uniform remeshing and adaptive remeshing. The comparison is summarized in Table 1. We can see that our algorithm is almost as good as CVT_{fs} in approximating the original mesh, while having significantly better angle quality than any other method, assuming that both small and large angles are considered undesirable. In the second comparison, we only compare our method with CVT and CVT_{fs} (See Table 2). Selected meshing results are visualized in Fig. 7 and more results are shown in the additional materials.

Parameter analysis. There are two main parameters used in our approach. One is the coefficient λ of the penalty term (Eq. 2) and the other is the parameter s that controls the scale of feature-sensitive CVT. Assume that in the asymptotical setting, each restricted Voronoi cell becomes a regular hexagon, and the density function is a constant inside each cell. Denote r as the edge length of a Voronoi edge. Then, CVT energy is proportional to r^4, while the penalty term is proportional to r^2. Since $r^2 \propto \frac{|M|}{n}$ ($|M|$ is the area of the mesh surface), we multiply the penalty term by $\frac{|M|}{n}$ for normalization.

Another important parameter is the feature-sensitive parameter, s, in Eq. 2. Fig. 8 shows the influence of this parameter. In this example, we set $\lambda = 1$ and increase the value of s. As shown in the figure, a larger s will introduce more badly shaped triangles, but with smaller approximation errors, i.e., smaller Hausdorff distances.

Sharp feature handling. We provide an alternative way to handle the sharp features of CAD models or boundaries. We assume that the sharp features are tagged by the user or given as input. We restrict the vertices whose restricted Voronoi cells intersect with feature curves strictly on the feature, similar to previous approaches [24], [28]. The vertices of the input mesh that are incident to (more than) three feature edges are identified as corners and are sampled and kept unchanged during the optimization process. Fig. 9 shows an example of boundary handling of curved surfaces, and Fig. 11 shows another example of handling CAD models with structured feature skeletons.

Comparison. We first compare our results with decimation-based, non-obtuse remeshing [44] in Table 3. Our results always have minimal angles larger than 30° while the results of [44] do not have this property. Further, our approximation quality is better.

Model	Method	$\lvert\mathbf{X}\rvert$	$\lvert\triangle\rvert$	θ_{min}	$\theta_{<30^0}\%$	$d_H(\times 10^{-2})$
Horse	[44]	9944	19880	15.03	0.18	1.31
	Ours	9944	19884	**38.4**	**0**	**0.46**

TABLE 3
A comparison with non-obtuse mesh decimation [44].

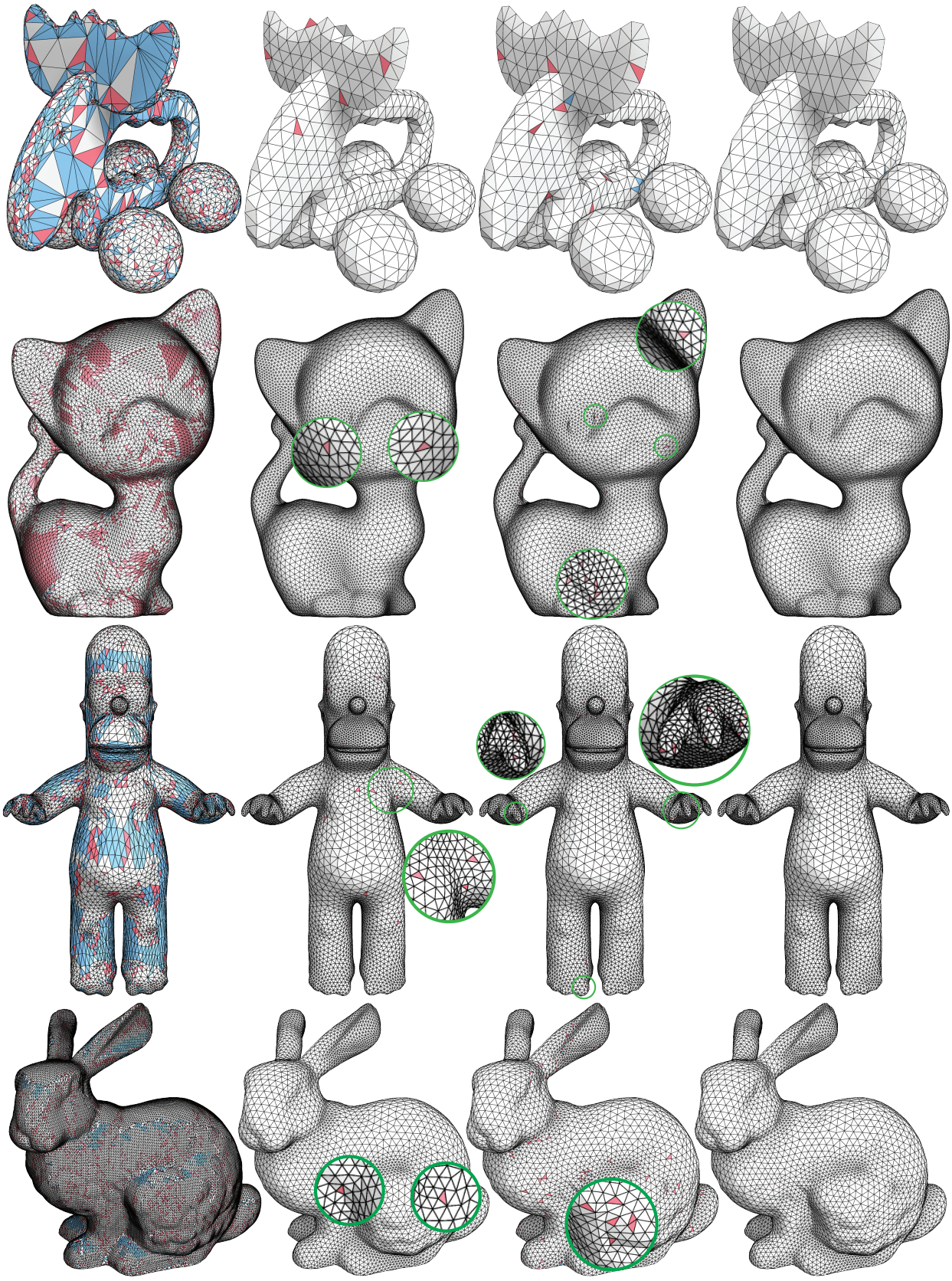


Fig. 7. Comparison of remeshing with CVT, CVT$_{fs}$ and our method: (top to bottom) uniform remeshing of the Elk model (1K vertices); and adaptive remeshing of Kitten (9.8K vertices); Homer (7.5K vertices), and Bunny (8K vertices). The obtuse triangles are shown in pink, and triangles with $\theta_{min} \leq 30^{\circ}$ are shown in blue. From left to right: input meshes, results of CVT, CVT$_{fs}$ and CVT_{nob} (ours).

Model	Method	$\|\mathbf{X}\|$	$\|\triangle\|$	$\|\triangle_{obt}\|$	Q_{min}	Q_{avg}	θ_{min}	$\bar{\theta}_{min}$	θ_{max}	$\theta_{<30^o}\%$	$\theta_{>90^o}\%$	$V_{567}\%$	$d_{RMS}(\times 10^{-3})$	$d_H(\times 10^{-2})$
Venus	ACVD	3.0K	6.0K	466	0.25	0.84	14.4	47.6	146.3	0.36	7.77	98.4	0.76	0.74
	MPS	3.0K	6.0K	**0**	0.67	0.85	32.7	48.6	90.0	**0**	**0**	**100**	0.67	0.68
	FPO	3.0K	6.0K	377	0.57	0.85	34.2	50.8	107.1	**0**	6.29	99.7	0.64	0.71
	CVT_{cap}	3.0K	6.0K	1.0K	0.39	0.78	20.5	43.3	128.9	4.41	17.7	98.8	0.68	0.61
	CVT	3.0K	6.0K	15	0.65	0.93	**39.5**	54.5	97.3	**0**	0.25	**100**	0.76	0.59
	CVT_{fs}	3.0K	6.0K	49	0.61	0.91	27.2	53.0	102.7	0.03	0.82	99.8	**0.32**	**0.31**
	CVT_{nob}	3.0K	6.0K	**0**	**0.73**	**0.96**	37.4	**55.9**	**86.1**	**0**	**0**	**100**	0.62	0.56
Genus	ACVD	6.5K	13K	913	0.47	0.85	29.0	48.1	118.6	0.06	7.02	98.1	0.45	0.44
	MPS	6.5K	13K	**0**	0.66	0.85	31.3	48.5	90.0	**0**	**0**	**100**	0.41	0.65
	FPO	6.5K	13K	737	0.55	0.86	33.1	50.9	109.3	**0**	6.14	99.6	0.42	0.59
	CVT_{cap}	6.5K	13K	2.3K	0.38	0.78	16.8	42.6	130.0	5.91	18.3	98.6	0.44	0.45
	CVT	6.5K	13K	27	0.66	0.94	**38.7**	54.6	96.8	**0**	0.21	**100**	0.47	0.52
	CVT_{fs}	6.5K	13K	50	0.62	0.93	34.1	54.5	100.3	**0**	0.38	**100**	**0.21**	**0.22**
	CVT_{nob}	6.5K	13K	**0**	**0.72**	**0.95**	36.0	**55.6**	**88.9**	**0**	**0**	**100**	0.23	0.31
Homer	ACVD	7.5K	15K	2.7K	0.04	0.80	2.11	44.4	174.1	6.18	17.7	94.4	0.69	0.33
	MPS	7.5K	15K	1.6K	0.56	0.82	30.1	46.4	105.8	0	10.8	100	0.52	0.30
	FPO	7.5K	15K	1.2K	0.50	0.85	28.2	49.9	116.0	0.01	7.52	98.9	0.43	0.30
	CVT_{cap}	7,5K	15K	1.6K	0.37	0.83	21.3	46.2	131.1	0.98	10.66	95.8	0.37	0.20
	CVT	7.5K	15K	56	0.65	0.93	**38.1**	54.0	98.2	0	0.37	100	0.49	0.20
	CVT_{fs}	7.5K	15K	108	0.50	0.91	27.4	52.5	113.1	0.01	7.21	99.9	**0.24**	**0.16**
	CVT_{nob}	7.5K	15K	**0**	**0.69**	**0.94**	33.7	**54.5**	**85.4**	**0**	**0**	**100**	0.31	0.17
Bunny	ACVD	8.0K	16K	3.3K	0.10	0.77	4.76	42.4	165.9	9.96	20.5	92.2	0.61	0.59
	MPS	8.3K	16.6K	1.3K	0.400	0.83	33.9	53.9	103.0	0	0.29	99.8	0.47	0.34
	FPO	8.0K	16K	1.6K	0.39	0.84	22.6	48.7	128.5	0.56	0.29	98.0	0.45	0.34
	CVT_{cap}	8.0K	16K	1.4K	0.39	0.84	15.6	47.3	125.1	1.04	8.46	98.7	**0.43**	**0.24**
	CVT	8.0K	16K	10	0.64	0.93	34.8	54.2	98.8	**0**	0.06	99.9	0.53	0.37
	CVT_{fs}	8.0K	16K	492	0.32	0.89	12.5	50.6	128.8	0.43	3.06	99.5	**0.43**	0.26
	CVT_{nob}	8.0K	16K	**0**	**0.75**	**0.94**	**35.5**	**55.0**	**84.2**	**0**	**0**	**100**	0.44	0.49

TABLE 1

Comparison of remeshing quality with previous techniques. The best result of each measurement is marked in **bold** font. $|\mathbf{X}|$ is the number of vertices; $|\triangle|$ is the number of triangles; $|\triangle_{obt}|$ is the number of obtuse triangles; Q_{min} is the minimal triangle quality, where the quality of a triangle is $Q(\mathbf{t}) = \frac{6}{\sqrt{3}} \frac{S_\mathbf{t}}{p_\mathbf{t} h_\mathbf{t}}$, where $S_\mathbf{t}$ is the area of $\mathbf{t}$, $p_\mathbf{t}$ is the half-perimeter of $\mathbf{t}$ and $h_\mathbf{t}$ the the longest edge length of $\mathbf{t}$ [51]; Q_{avg} is the average of the triangle qualities; θ_{min} is the minimal angle; $\bar{\theta}_{min}$ is the average of minimal angle of each triangle; θ_{max} is the maximal angle, $\theta_{<30^o}\%$ is the percentage of triangles with angles smaller than 30^o; $\theta_{>90^o}\%$ is the percentage of obtuse triangles; $V_{567}\%$ is the percentage of the valence 5, 6, and 7 vertices; d_{RMS} is the root mean square distance, and d_H is the Hausdorff distance between the remesh and the input surface, which is measured by Metro [52].

Model	Method	$\|\mathbf{X}\|$	$\|\triangle\|$	$\|\triangle_{obt}\|$	Q_{min}	Q_{avg}	θ_{min}	$\bar{\theta}_{min}$	θ_{max}	$\theta_{<30^o}\%$	$\theta_{>90^o}\%$	$V_{567}\%$	$d_{RMS}(\times 10^{-3})$	$d_H(\times 10^{-2})$
Elk	CVT	1.0K	2.0K	8	0.66	0.92	**36.9**	53.7	95.0	**0**	0.40	97.3	3.64	1.48
	CVT_{fs}	1.0K	2.0K	29	0.50	0.90	22.6	51.6	104.0	0.70	1.45	97.5	**1.58**	**0.75**
	CVT_{nob}	1.0K	2.0K	**0**	0.70	**0.93**	36.2	**53.9**	**88.2**	**0**	**0**	**97.8**	1.93	1.14
Botijo	CVT	5.0K	10K	8	0.67	0.93	**38.2**	**54.2**	94.9	**0**	0.08	100	0.74	0.33
	CVT_{fs}	5.0K	10K	231	0.51	0.89	26.5	51.1	114.8	0.06	2.31	99.7	**0.36**	0.33
	CVT_{nob}	5.0K	10K	**0**	**0.70**	**0.94**	31.1	**54.2**	**88.4**	**0**	**0**	**100**	0.48	**0.32**
Rockerarm	CVT	5.8K	11.6K	5	0.67	**0.94**	**37.9**	54.7	94.7	**0**	0.04	100	0.79	0.46
	CVT_{fs}	5.8K	11.6K	391	0.28	0.88	10.5	50.5	117.5	0.49	3.37	99.3	**0.28**	**0.31**
	CVT_{nob}	5.8K	11.6K	**0**	**0.74**	**0.94**	31.7	**54.9**	**89.9**	**0**	**0**	**100**	0.53	0.36
Kitten	CVT	9.8K	19.6K	6	0.67	0.94	**39.2**	54.8	95.6	**0**	0.03	99.9	0.32	0.22
	CVT_{fs}	9.8K	19.6K	119	0.52	0.93	26.4	53.9	112.6	0.03	0.61	99.9	**0.17**	**0.14**
	CVT_{nob}	9.8K	19.6K	**0**	**0.71**	**0.95**	33.6	**55.3**	**87.1**	**0**	**0**	**100**	0.23	0.21

TABLE 2

Comparison of remeshing quality with other CVT-based approaches. Definitions of all symbols are given in the legend to Table 1.

While the results of MPS [30], [53] do not show an optimization for non-obtuse triangles, the method can be configured to eliminate non-obtuse triangles by random resampling. We can observe that MPS is able

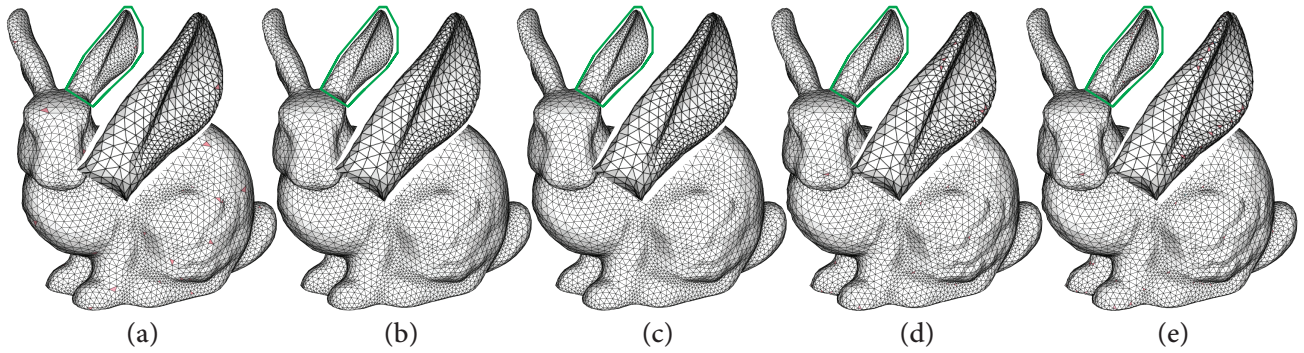


Fig. 8. Illustration of the remeshing results with different parameters for feature-sensitive CVT. (a) result of CVT: $s = 1, \lambda = 0, |\triangle_{obt}| = 79, d_H = 0.41\%$; (b) result with CVT + penalty term: $s = 1, \lambda = 1, |\triangle_{obt}| = 0, d_H = 0.36\%$, (c, d, e) result of CVT$_{fs}$ + penalty term: $s = 5, \lambda = 1, |\triangle_{obt}| = 0, d_H = 0.33\%$; $s = 8, \lambda = 1, |\triangle_{obt}| = 113, d_H = 0.20\%$; $s = 10, \lambda = 1, |\triangle_{obt}| = 210, d_H = 0.18\%$.

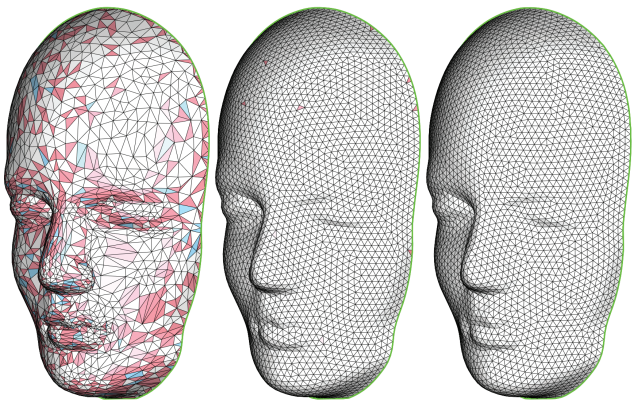


Fig. 9. Boundary and feature handling. Left: input mesh (the boundary curve is shown in green); middle: result of CVT with 14 obtuse triangles; right: our non-obtuse remeshing.

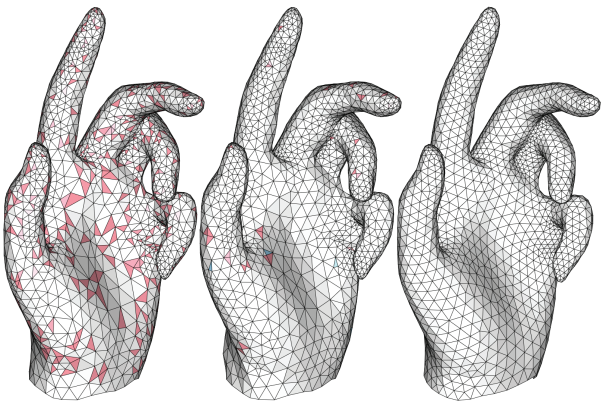


Fig. 10. Comparison with well-centered meshes generated by weighted triangulation [4]. The same number of vertices are used in the remeshing. Left: input mesh with 687 obtuse triangles (16.0%); middle: result of the weighted triangulation [4] with 17 triangles with small angles ($< 30^o$) and 93 obtuse triangles (2.17%); right: result of our non-obtuse remeshing.

to generate non-obtuse meshes for uniform remeshing problems. However, it never converges for adaptive remeshing.

Finally, we compare our results with the results of recent work using weighted triangulations that aims at generating well-centered meshes [4], which is an improvement over primal/dual *Hodge-Optimized Triangulation* (HOT) [38]. As shown in Fig. 10, although the weighted triangulation approach creates meshes with elegant primal/dual structures, it is not able to eliminate all the obtuse triangles. Since the weighted center of each triangle is not the circumcenter anymore, the well-centered mesh is typically not a non-obtuse mesh in the context of weighted triangulations.

Limitations. Our algorithm might not be successful in the presence of noise or when the density function changes dramatically (see Fig. 11 (top)). Our algorithm cannot eliminate all obtuse triangles along the sharp features. One example is the uniform remeshing of the Joint model as shown in Fig. 11 (bottom left). This model contains sharp features and planar regions on each side of the features. In this case, once the valence of a vertex on a sharp feature is less than six, there will be an obtuse angle on the side where the vertex has only one neighbor. Another failure of adaptive remeshing of the Sculpt models is shown in the bottom right of Fig. 11. Some of these limitations could be dealt with in future work, but certain configurations of sharp features inherently lead to small undesirable angles (e.g., two feature curves joining with a small angle). Another limitation is that we do not provide a theoretical proof of the convergence analysis of the proposed primal penalty term that could guarantee the max/min angle bounds of the resulting meshes. We hope to address these limitations in future work.

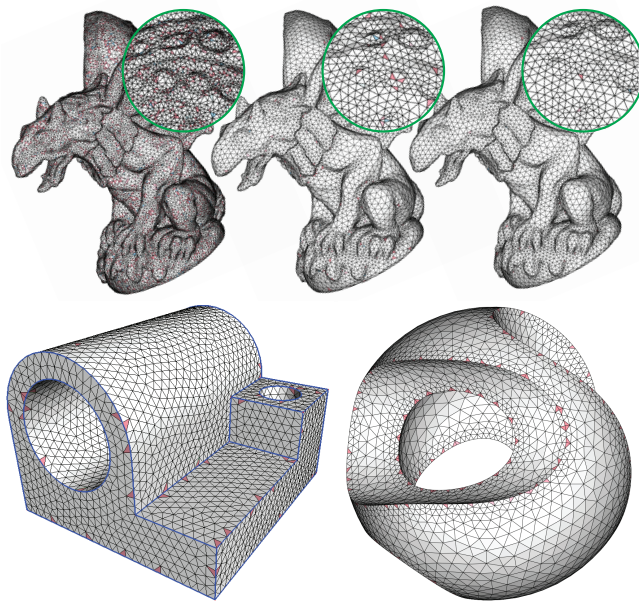


Fig. 11. Unsatisfactory examples. Top row: our algorithm fails to eliminate all the obtuse triangles of the Gargoyle model, which has dramatically changed density functions (left to right: input, result of CVT with 591 bad triangles and final result with 143 bad triangles). As shown in the top right, an obtuse angle appears in the transitional region where the size of the triangles changes quickly. Bottom row: Uniform and adaptive remeshing of the Joint and Sculpt models with 4K vertices, respectively. Our algorithm cannot eliminate all the obtuse triangles that are adjacent to sharp features. The angle bounds are $[35.4^o, 105.0^o]$ and $[31.5^o, 101.1^o]$ for the Joint and Sculpt models, respectively.

6 Conclusion and Future Work

We present a novel remeshing algorithm that reduces the number of triangles with small angles (less than 30 degrees) and triangles with large (obtuse) angles. We augment the original CVT formulation by a primal penalty term preventing short Voronoi edges. In this formulation, the dual term helps us to avoid small angles and the primal term helps us to avoid large angles. In future work, we plan to look for new applications of our non-obtuse remeshing technique and extend our concept of primal/dual optimization to volumetric tetrahedral mesh generation, which is preferred in applications such as fluid simulation.

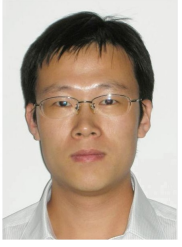
Acknowledgements

We would like to thank Zhonggui Chen, Fernando de Goes and Sebastien Valette for providing the data and executables, Bruno Lévy for sharing the code for feature-sensitive CVT, and Lubin Fan for the rendering. This work was partially supported by the KAUST Visual Computing Center, the National Natural Science Foundation of China (Nos. 61372168, 61331018, 61571439, and 61572502), the China National 863 Program (No. 2015AA016402), the Scientific Research Foundation for the Returned Overseas Chinese Scholars of State Education Ministry of China, the open funding project of the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University (No. BUAA-VR-15KF-06).

References

[1] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Lévy, *Polygon Mesh Processing*. AK Peters, 2010.
[2] Q. Du, D. Wang, and L. Zhu, "On mesh geometry and stiffness matrix conditioning for general finite element spaces," *SIAM J. Numer. Anal.*, vol. 47, no. 2, pp. 1421–1444, 2009.
[3] K. Crane, F. de Goes, M. Desbrun, and P. Schröder, "Digital geometry processing with discrete exterior calculus," in *ACM SIGGRAPH 2013 Courses*, 2013, pp. 7:1–7:126.
[4] F. de Goes, P. Memari, P. Mullen, and M. Desbrun, "Weighted triangulations for geometry processing," *ACM Trans. on Graphics*, vol. 33, no. 3, pp. 28:1–28:13, 2014.
[5] P. Heckbert and M. Garland, "Survey of polygonal surface simplification algorithms," in *SIGGRAPH 97 Course Notes: Multiresolution Surface Modeling*, 1997.
[6] M. Botsch and L. Kobbelt, "A remeshing approach to multiresolution modeling," in *Proc. of Symp. of Geometry Processing*, 2004, pp. 189–196.
[7] J. Schreiner, C. E. Scheidegger, S. Fleishman, and C. T. Silva, "Direct (re)meshing for efficient surface processing," *Computer Graphics Forum (Proc. EUROGRAPHICS)*, vol. 25, no. 3, pp. 527–536, 2006.
[8] S.-W. Cheng, T. K. Dey, and J. R. Shewchuk, *Delaunay Mesh Generation*. CRC Press, 2012.
[9] P. Alliez, D. Cohen-Steiner, O. Devillers, B. Lévy, and M. Desbrun, "Anisotropic polygonal remeshing," *ACM Trans. on Graphics (Proc. SIGGRAPH)*, vol. 22, no. 3, pp. 485–493, 2003.
[10] B. Lévy and Y. Liu, "L_p centroidal Voronoi tesselation and its applications," *ACM Trans. on Graphics (Proc. SIGGRAPH)*, vol. 29, no. 4, pp. 119:1–11, 2010.
[11] P. Alliez, G. Ucelli, C. Gotsman, and M. Attene, "Recent advances in remeshing of surfaces," in *Shape Analysis and Structuring*, 2008, pp. 53–82.
[12] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, 2nd ed. Wiley, 2000.
[13] Q. Du, V. Faber, and M. Gunzburger, "Centroidal Voronoi tessellations: applications and algorithms," *SIAM Review*, vol. 41, pp. 637–676, 1999.
[14] P. Alliez, M. Meyer, and M. Desbrun, "Interactive Geometry Remeshing," *ACM Trans. on Graphics (Proc. SIGGRAPH)*, vol. 21(3), pp. 347–354, 2002.
[15] P. Alliez, É. C. d. Verdiére, O. Devillers, and M. Isenburg, "Isotropic surface remeshing," in *Shape Modeling International - SMI*, 2003, pp. 49–58.
[16] V. Surazhsky, P. Alliez, and C. Gotsman, "Isotropic remeshing of surfaces: a local parameterization approach," in *12th Intl. Meshing Roundtable*, 2003, pp. 204–231.
[17] M. Nieser, J. Palacios, K. Polthier, and E. Zhang, "Hexagonal global parameterization of arbitrary surfaces," *IEEE Trans. on Vis. and Comp. Graphics*, vol. 18, no. 6, pp. 865–878, 2011.
[18] S. A. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.
[19] G. Rong, M. Jin, L. Shuai, and X. Guo, "Centroidal Voronoi tessellation in universal covering space of manifold surfaces," *Comp. Aided Geom. Design*, vol. 28, no. 8, pp. 475–496, 2011.
[20] S. Valette, J.-M. Chassery, and R. Prost, "Generic remeshing of 3D triangular meshes with metric-dependent discrete Voronoi diagrams," *IEEE Trans. on Vis. and Comp. Graphics*, vol. 14, no. 2, pp. 369–381, 2008.
[21] G. Peyré and L. D. Cohen, "Geodesic remeshing using front propagation," *Int. J. Comput. Vision*, vol. 69, no. 1, pp. 145–156, 2006.

[22] Y. Fu and B. Zhou, "Direct sampling on surfaces for high quality remeshing," in *ACM symposium on Solid and physical modeling*, 2008, pp. 115–124.
[23] X. Wang, X. Ying, Y.-J. Liu, S.-Q. Xin, W. Wang, X. Gu, W. Mueller-Wittig, and Y. He, "Intrinsic computation of centroidal Voronoi tessellation (CVT) on meshes," *Computer-Aided Design*, vol. 58, no. 0, pp. 51 – 61, 2015.
[24] D.-M. Yan, B. Lévy, Y. Liu, F. Sun, and W. Wang, "Isotropic remeshing with fast and exact computation of restricted Voronoi diagram," *Computer Graphics Forum (Proc. SGP)*, vol. 28, no. 5, pp. 1445–1454, 2009.
[25] Y. Liu, W. Wang, B. Lévy, F. Sun, D.-M. Yan, L. Lu, and C. Yang, "On centroidal Voronoi tessellation - energy smoothness and fast computation," *ACM Trans. on Graphics*, vol. 28, no. 4, pp. 101:1–101:11, 2009.
[26] B. Lévy and N. Bonneel, "Variational anisotropic surface meshing with Voronoi parallel linear enumeration," in *Proceedings of the 21st International Meshing Roundtable*, 2012, pp. 349–366.
[27] B. Lévy, "Restricted voronoi diagrams for (re)-meshing surfaces and volumes," in *Curves and Surfaces*, 2014, code download: http://gforge.inria.fr/projects/geogram/.
[28] D.-M. Yan, G. Bao, X. Zhang, and P. Wonka, "Low-resolution remeshing using the localized restricted Voronoi diagram," *IEEE Trans. on Vis. and Comp. Graphics*, vol. 20, no. 10, pp. 418–1427, 2014.
[29] Z. Chen, Z. Yuan, Y.-K. Choi, L. Liu, and W. Wang, "Variational blue noise sampling," *IEEE Trans. on Vis. and Comp. Graphics*, vol. 18, no. 10, pp. 1784–1796, 2012.
[30] D.-M. Yan and P. Wonka, "Gap processing for adaptive maximal Poisson-disk sampling," *ACM Trans. on Graphics*, vol. 32, no. 5, pp. 148:1–148:15, 2013.
[31] D.-M. Yan, J. Guo, X. Jia, X. Zhang, and P. Wonka, "Blue-noise remeshing with farthest point optimization," *Computer Graphics Forum (Proc. SGP)*, vol. 33, no. 5, pp. 167–176, 2014.
[32] S. Zhang, J. Guo, H. Zhang, X. Jia, D.-M. Yan, J.-H. Yong, and P. Wonka, "Capacity constrained blue-noise sampling on surfaces," *Computers & Graphics*, 2015, accepted.
[33] V. Nivoliers, D.-M. Yan, and B. Lévy, "Fitting polynomial surfaces to triangular meshes with Voronoi squared distance minimization," *Eng. with Comput.*, vol. 30, no. 3, pp. 289–300, 2014.
[34] H. Pan, Y.-K. Choi, Y. Liu, W. Hu, Q. Du, K. Polthier, C. Zhang, and W. Wang, "Robust modeling of constant mean curvature surfaces," *ACM Trans. on Graphics (Proc. SIGGRAPH)*, vol. 31, no. 4, p. 85, 2012.
[35] D. Sieger, P. Alliez, and M. Botsch, "Optimizing Voronoi diagrams for polygonal finite element computations," in *Proceedings of the 19th International Meshing Roundtable, IMR 2010*, 2010, pp. 335–350.
[36] E. Vanderzee, A. N. Hirani, D. Guoy, and E. Ramos, "Well-centered triangulation," *SIAM J. Sci. Comput.*, vol. 31, no. 6, p. 4497C4523, 2010.
[37] F. Sun, Y.-K. Choi, W. Wang, D.-M. Yan, Y. Liu, and B. Lévy, "Obtuse triangle suppression in anisotropic meshes," *Comp. Aided Geom. Design*, vol. 28, no. 9, pp. 537–548, 2011.
[38] P. Mullen, P. Memari, F. de Goes, and M. Desbrun, "Hot: Hodge-optimized triangulations," *ACM Trans. on Graphics (Proc. SIGGRAPH)*, vol. 30, no. 4, pp. 103:1–103:12, July 2011.
[39] J. R. Shewchuk, "What is a good linear element? interpolation, conditioning, and quality measures," in *11th Intl. Meshing Roundtable*, 2002, pp. 115–126.
[40] B. S. Baker, E. Grosse, and C. S. Raffery, "Nonobtuse triangulation of polygons," *Disc. & Comp. Geom.*, vol. 3, pp. 147–168, 1988.
[41] M. Bern, S. Mitchell, and J. Ruppert, "Linear-size nonobtuse triangulation of polygons," in *ACM Symp. on Comp. Geom.*, 1994, pp. 221–230.
[42] H. Erten and A. Üngör, "Computing acute and non-obtuse triangulations," in *CCCG 2007*, 2007, pp. 205–208.
[43] H. Erten and A. Üngör, "Computing triangulations without small and large angles," in *Sixth International Symposium on Voronoi Diagrams, ISVD 2009*, 2009, pp. 192–201.
[44] J. Li and H. Zhang, "Nonobtuse remeshing and decimation," in *Proc. of Eurographics Symposium on Geometry Processing*, 2006, p. Proc. of Symp. of Geometry Processing.
[45] H. Edelsbrunner and N. R. Shah, "Triangulating topological spaces," *IJCGA*, vol. 7, no. 4, pp. 365–378, 1997.
[46] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Math. Program.*, vol. 45, no. 3, pp. 503–528, 1989.
[47] R. J. Renka, "Two simple methods for improving a triangle mesh surface," *Computer Graphics Forum*, p. to appear, 2015.
[48] N. Amenta, M. Bern, and M. Kamvysselis, "A new Voronoi-based surface reconstruction algorithm," in *Proc. ACM SIGGRAPH*, 1998, pp. 415–421.
[49] G. F. Tóth, "A stability criterion to the moment theorem," *Studia Scientiarum Mathematicarum Hungarica*, vol. 38, p. 209C224, 2001.
[50] L. Lu, F. Sun, H. Pan, and W. Wang, "Global optimization of centroidal voronoi tessellation with monte carlo approach," *IEEE Trans. on Vis. and Comp. Graphics*, vol. 18, no. 11, pp. 1880–1890, 2012.
[51] P. Frey and H. Borouchaki, "Surface mesh evaluation," in *6th Intl. Meshing Roundtable*, 1997, pp. 363–374.
[52] P. Cignoni, C. Rocchini, and R. Scopigno, "Metro: measuring error on simplified surfaces," *Computer Graphics Forum*, vol. 17, no. 2, pp. 167–174, 1998.
[53] J. Guo, D.-M. Yan, X. Jia, and X. Zhang, "Efficient maximal Poisson-disk sampling and remeshing on surfaces," *Computers & Graphics*, vol. 46, no. 6-8, pp. 72–79, 2015.



Dong-Ming Yan received his PhD degree from Hong Kong University in 2010, and his Bachelor's and Master's degrees from Tsinghua University in 2002 and 2005, respectively. He is a research scientist at King Abdullah University of Science and Technology (KAUST) and an associate professor at the National Laboratory of Pattern Recognition of the Institute of Automation, Chinese Academy of Sciences. His research interests include computer graphics, geometric processing and visualization.

Peter Wonka received his PhD in computer science and his MS in urban planning from the Technical University of Vienna, Vienna, Austria, in 2001 and 2002, respectively. He was a postdoctoral researcher at the Georgia Institute of Technology, Atlanta, GA, for two years. He is currently a professor in the Computer, Electrical and Mathematical Science and Engineering Division, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia, and also an associate professor at Arizona State University, Tempe, AZ, USA. His research interests include computer graphics, visualization, computer vision, remote sensing, image processing, and machine learning.