

DOI: 10.1111/cgf.14897

Eurographics Symposium on Geometry Processing 2023
 P. Memari and J. Solomon
 (Guest Editors)

COMPUTER GRAPHICS *forum*
 Volume 42 (2023), Number 5

PowerRTF: Power Diagram based Restricted Tangent Face for Surface Remeshing

Yuyou Yao¹ , Jingjing Liu¹, Yue Fei¹, Wenming Wu¹, Gaofeng Zhang¹ , Dong-Ming Yan^{2,3} and Liping Zheng^{1†}

¹ School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China

² State Key Laboratory of Multimodal Artificial Intelligence Systems (MAIS) & National Laboratory of Pattern Recognition (NLPR), Institute of Automation of the Chinese Academy of Sciences, Beijing, China

³ School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China

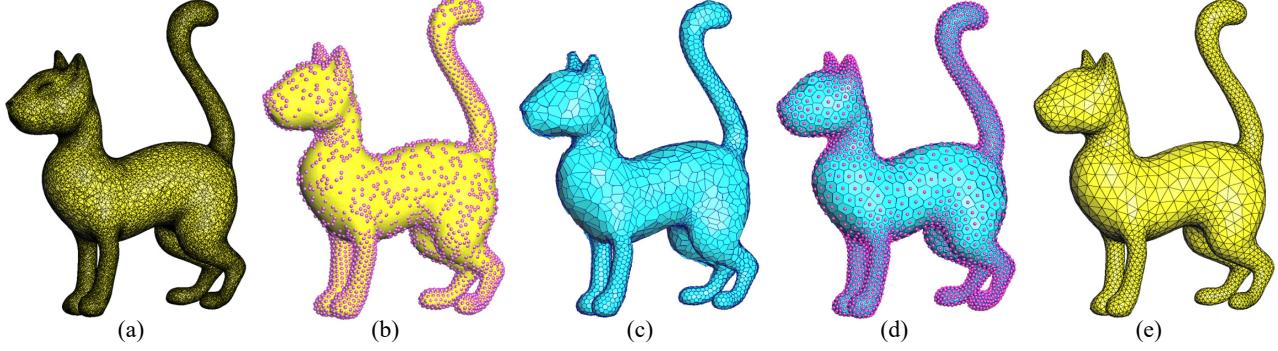


Figure 1: Overall remeshing procedure of our method. (a) input surface with 35.0k vertices and 70.0k facets; (b) initial sample points; (c) initial PowerRTF; (d) PowerRTF of optimized sample points; and (e) output triangular mesh with 3.0k vertices and 6.0k facets.

Abstract

Triangular meshes of superior quality are important for geometric processing in practical applications. Existing approximative CVT-based remeshing methodology uses planar polygonal facets to fit the original surface, simplifying the computational complexity. However, they usually do not consider surface curvature. Topological errors and outliers can also occur in the close sheet surface remeshing, resulting in wrong meshes. With this regard, we present a novel method named PowerRTF, an extension of the restricted tangent face (RTF) in conjunction with the power diagram, to better approximate the original surface with curvature adaption. The idea is to introduce a weight property to each sample point and compute the power diagram on the tangent face to produce area-controlled polygonal facets. Based on this, we impose the variable-capacity constraint and centroid constraint to the PowerRTF, providing the trade-off between mesh quality and computational efficiency. Moreover, we apply a normal verification-based inverse side point culling method to address the topological errors and outliers in close sheet surface remeshing. Our method independently computes and optimizes the PowerRTF per sample point, which is efficiently implemented in parallel on the GPU. Experimental results demonstrate the effectiveness, flexibility, and efficiency of our method.

CCS Concepts

- Computing methodologies → Computer graphics; Shared memory algorithms;

1. Introduction

Triangular meshes are most frequently utilized to represent three-dimensional (3D) digital models owing to their versatility and sim-

plicity. The raw meshes are typically generated using 3D modeling software [ER15] or reconstructed from sample points collected by scanners [CMZZ21]. However, the majority of raw meshes are of poor quality, e.g., small angles, short edges [LZ11, EI14], etc., making it impossible for them to be used directly in subsequent applications. Therefore, surface remeshing has received much attention.

† Corresponding author. E-mail: zhengl@hfut.edu.cn

tion, and numerous techniques have been proposed to improve the mesh quality, e.g., simplification-based [PJ19], local modification-based [HYB^{*}16], and segmentation-based methods [ME16], etc.

Centroidal Voronoi tessellation (CVT) [DFG99] is a powerful technique for surface remeshing. Specifically, given an input surface and the number of sample points, the CVT energy is iteratively minimized to yield the optimal sample point distribution [Bal09]. Thus, the duals of the Voronoi diagram of these optimal sample points could be utilized to produce triangular meshes of high quality. A significant problem in CVT-based surface remeshing techniques is the computation of the Voronoi diagram on the complex surface at each iteration. Previous work either construct an exact Voronoi diagram on the surface, such as the geodesic Voronoi diagram (GVD) [LFXH17] and the geodesic CVT (GCVT) [YYY^{*}19], or truncate a 3D Voronoi cell of the original surface, such as the restricted CVT (RCVT) [DGJ03] and the restricted Voronoi diagram (RVD) [YLL^{*}09], etc. As they calculate precise geodesic distances or intersections on the surface, they could be regarded as the exact CVT-based approaches. However, calculating the geodesic distances or intersections is very time-consuming, even taking several seconds per iteration for complex models.

Recently, the idea of approximation has been used for surface remeshing [BLM17]. Several strategies [CZC^{*}18, XXT19, ABE^{*}20] that could be considered approximative CVT-based methodologies have been put forth. Instead of calculating precise geodesic distances or intersections, they use a set of planar polygonal facets to approximate the surface and compute CVT on these planar facets [ZCHK13]. Thus, the computation per iteration is significantly simplified. The primary challenge with this sort of method is computing the approximative polygonal facets. One idea is to construct a 3D Voronoi diagram using a series of auxiliary points, where these facets of adjacent Voronoi cells close to the surface boundary are considered as these desired approximative facets, such as the restricted power face (RPF) [XXT19] and Voro-Crust [ABE^{*}20], etc. They are simple and straightforward, but the facet extraction relies on the 3D Voronoi diagram construction. Another option is to compute the clipped tangent planes of sample points to roughly resemble the surface, where auxiliary points and 3D Voronoi diagram construction are no longer necessary, such as the restricted Voronoi cell (RVC) [CZC^{*}18] and the restricted tangent face (RTF) [YLW^{*}23], etc. The computation of the clipped tangent planes for each sample point could be done independently, allowing for high parallelization and GPU implementation. However, curvature adaption is not considered in RTF, and topological errors and outliers may also occur in close sheet surface remeshing, as shown in Figure 13(b).

Therefore, we concentrate on and handle the limitations in earlier works. Motivated by the accurate capacity-constrained feature of the power diagram, we propose an effective power diagram based RTF (PowerRTF) to approximate the original surface, achieving curvature-adapted and quality-controlled meshes, as shown in Figure 1. Instead of building a Voronoi diagram on each tangent plane, we construct a power diagram by applying a weight characteristic to each sample point for curvature-adapted surface remeshing. That is, small (large) power cells with teensy (large) values of weights are applied for curved (flat) regions, as shown in Fig-

ure 1(d). Rather than capacity constrained centroidal power diagram (CCCPD) [XLC^{*}16], a variable-capacity constrained variant, called variable-capacity constrained centroidal power diagram (VCCCPD) [ZYW^{*}21] is further introduced to combine with PowerRTF, which imposes interval capacity constraint to each sample point to achieve the trade-off between mesh quality and computational efficiency. Moreover, to tackle the topological errors and outliers in close sheet surface remeshing, we apply an additional normal verification-based inverse side point culling (NVISPC) method, which avoids the impact of points from the opposing facets. Owing to the independent construction and optimization of PowerRTF for each sample point, we develop a GPU algorithm for surface remeshing via PowerRTF, yielding curvature-adapted sample points and meshes. We contribute the following:

- We present an extension of RTF in conjunction with the power diagram, namely PowerRTF, to approximate the original surface with better curvature adaptation than state-of-the-art methods. Instead of CCCPD [XLC^{*}16], a variable-capacity constrained variant, called VCCCPD [ZYW^{*}21], is further introduced to combine with PowerRTF, which imposes interval capacity constraint to each sample point to achieve the trade-off between mesh quality and computational efficiency.
- We present an NVISPC method, which prevents the impact of inverse side points to tackle the topological errors and outliers in close sheet surface remeshing with approximative CVT-based techniques.
- We develop a GPU-assisted construction algorithm to generate PowerRTF from sample points rapidly. Based on this, combined with an improved version of the weighted evaluated algorithm, we provide a GPU algorithm to optimize the PowerRTF with variable-capacity constraint.

The remainder of this paper is organized as follows. In Section 2, some surface remeshing techniques based on the CVT are briefly reviewed. Section 3 provides the preliminary of the power diagram, VCCCPD, and RTF. We presented a detailed explanation of our method for surface remeshing in Section 4. A variety of remeshing results of our method are shown in Section 5, and some conclusions are given in Section 6.

2. Related work

The goal of surface remeshing is to produce meshes of high quality by feeding the original mesh surface into a sequence of algorithms. An exhaustive review of surface remeshing methods that may be referred to [AUGA08, KPF^{*}20], is outside the scope of our work. In this section, we briefly review CVT-based surface remeshing.

2.1. Exact CVT-based remeshing

Iteratively computing the Voronoi diagram on a large and complex mesh surface is the main issue of the exact CVT-based remeshing. Integrated with numerical optimization, the CVT energy is minimized to generate regularly distributed sample points [LWL^{*}09]. Combined the relationship of meshes and their orthogonal duals, high-quality triangular meshes could be produced based on the duals of the Voronoi diagram of these sample points [MMdGD11].

Existing work on computing the exact Voronoi diagram on a surface could be summarized into two categories as follows.

Geodesic distance based methods. The first category constructs a Voronoi diagram based on the precise geodesic distance, e.g., GVD [LFXH17], GCVT [YYY*19], etc. GVD computation is firstly considered in parameter space and extrapolated the Euclidean space results to the original space. For instance, Alliez et al. [ADVDI03] turn GCVT computation into 2D CVT computation in a unit disk by applying a global conformal planar parameterization. Others directly compute the GVD on the original triangular meshes. The triangular mesh is incorporated in a tensor-based anisotropy metric in an iterative method [ZZCJ14] for mesh generation on the basis of anisotropic geodesic calculation. Wang et al. [WYL*15] provide two intrinsic techniques to accelerate the centroid calculation of GVD. Herholz et al. [HHA17] define Voronoi cells on the basis of heat diffusion and provide an approximation method to compute geodesic distances on surfaces. However, calculating precise geodesic distances on a complex surface is too complicated. Meanwhile, improving the quality of generated meshes and increasing the efficiency of geodesic distance calculation remains a challenge.

Clipped Voronoi cell based methods. Instead of calculating precise geodesic distances, the second category calculates the Voronoi diagram on a surface based on the intersections of a 3D Voronoi cell and the original surface, e.g., RVD [YLL*09], FRVD [DLY*18], etc. Du et al. [DGJ03] first propose a constrained CVT (CCVT), in which each sample point coincides with the mass center of its Voronoi cell. Yan et al. [YLL*09] offer a polyhedral meshing algorithm based on the clipped Voronoi diagram, called RVD, where an exact calculating algorithm is proposed to truncate a 3D Voronoi cell with the original surface. Later, they present several improvements to address the deficiencies in RVD, e.g., the clipped Voronoi diagram for closed 3D models [YWLL13], localized RVD (LRVD) for multiple disjoint surface patches [YBZW14], CVT extension for non-obtuse remeshing [YW15], and GPU version with efficient computation [HYWZ17], etc. Numerous extension versions of RVD have recently been developed for particular applications, e.g., L_p CVT for surface remeshing [LL10], RVD for close sheet models [WXT*20], RVD on a signed distance field [HZW*22], and the restricted power diagram (RPD) for medial axis transform with feature preservation [WWWG22]. However, RVD and its extensions require truncating a 3D Voronoi cell with a surface, necessitating expensive computation and consuming more time.

2.2. Approximative CVT-based remeshing

Contrary to the exact CVT-based method, the approximative ones use polygonal facets to fit the original surface [ZCHK13]. A regular distribution of sample points could be generated by combining the CVT optimization and the polygonal facet computation, reconstructing meshes of high quality with several algorithms [BL17, WWX*22]. By avoiding the calculation of precise geodesic distances or intersections on surfaces, the approximative CVT-based remeshing greatly reduces computational complexity. Thus, the computation of CVT on polygonal facets is the primary work.

Auxiliary points based methods. One class of methods for computing CVT on these approximative polygonal facets is on the basis

of auxiliary points, such as RPF [XXT19], VoroCrust [ABE*20], etc. They construct 3D Voronoi or power diagrams using sample points and auxiliary ones. The planes of adjacent Voronoi or power cells close to the surface boundary are identified as these desired polygonal facets. For instance, VoroCrust [ABE*20] utilizes pairs of seeds of spherical intersection to capture the surface boundary and some interior seeds to build a 3D Voronoi diagram. Xu et al. [XXT19] introduce weighted shadow points along the relevant normal direction, combined with the surface sample points, to construct 3D power diagram. Nevertheless, polygonal facet extraction relies on constructing 3D Voronoi or power diagrams, and these methods do not consider parallel computation.

Tangent plane based methods. Another option is to compute these approximative polygonal facets without auxiliary points directly [ZCHK13]. The tangent planes of sample points are utilized for surface approximation, e.g., RVC [CZC*18], RTF [YLW*23], etc. By restricting Voronoi cells on a set of best-fitting planes, Chen et al. [CZC*18] extend the CVT computation on point clouds and achieve partial surface curvature adaptation on the basis of a density-dependent energy function, that is the density-adapted RVC (D-RVC). More recently, Yao et al. [YLW*23] present an RPF simplification, namely RTF, where the requirement for these shadow points is eliminated. They provide a remeshing method based on a 3D CVT framework [RSLL18] to produce meshes of high quality. Owing to GPU acceleration and a well-designed projection strategy, this method delivers improved computation efficiency and more accurate approximation. However, RTF fails to capture surface curvature and struggles with close sheet surface remeshing, which is also a challenge for RVC, as shown in Figure 13(b).

3. Preliminary

In this section, we provide an overview of the definition of the power diagram, VCCPD, and RTF.

3.1. Power diagram

The Voronoi diagram [AK00], also known as the “Voronoi tessellation”, defines a set of disjoint subregions $\mathbf{V}(\mathbf{S}) = \{V(\mathbf{s}_i)\}_{i=1}^n$ based on a set of points (also called “sites” or “generators”) $\mathbf{S} = \{\mathbf{s}_i\}_{i=1}^n$ for a given domain $\Omega \subset \mathbb{R}^N$. Each Voronoi cell $V(\mathbf{s}_i)$, or subregion of the relevant point \mathbf{s}_i , is defined as:

$$V(\mathbf{s}_i) = \{\mathbf{s} \in \Omega \mid \|\mathbf{s} - \mathbf{s}_i\| \leq \|\mathbf{s} - \mathbf{s}_j\|, \forall j \neq i\} \quad (1)$$

where $\|\cdot\|$ denotes the Euclidean distance.

The power diagram [Aur87], as a significant extension of the Voronoi diagram, assigns weights $\mathbf{W} = \{w_i\}_{i=1}^n$ to points \mathbf{S} . That is, each point \mathbf{s}_i is associated with a weight parameter w_i . Each subregion $P(\mathbf{s}_i)$, called power cell, is redefined as:

$$P(\mathbf{s}_i) = \{\mathbf{s} \in \Omega \mid \|\mathbf{s} - \mathbf{s}_i\|^2 - w_i \leq \|\mathbf{s} - \mathbf{s}_j\|^2 - w_j, \forall j \neq i\} \quad (2)$$

The bisector of two adjacent power cells is the equal power distance line in two dimensional (2D) or the equal power distance plane in 3D. Notably, the power diagram degenerates to the Voronoi diagram when the weights of all points are equal [AHA98].

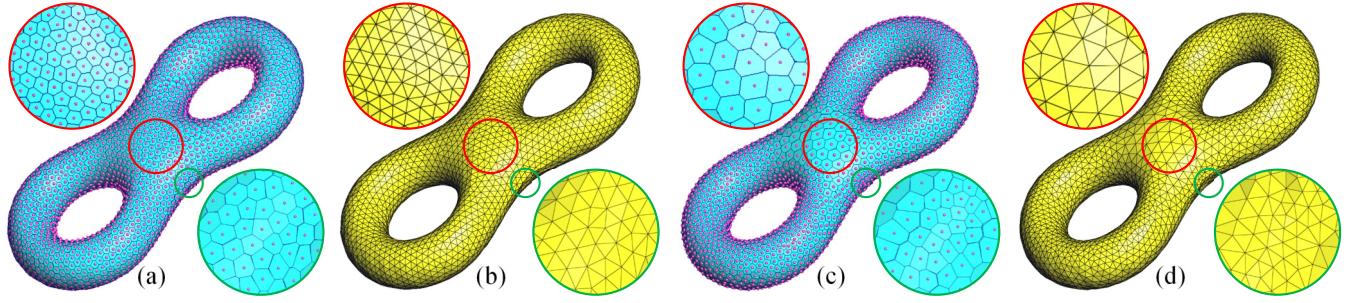


Figure 2: Exemplification of RTF and PowerRTF on the Eight model with same sample points, where purple dots represent the sample points on the surface. (a) RTFs of sample points in [YLW*23]; (b) triangular meshes generated by RTF [YLW*23]; (c) PowerRTFs of sample points to capture the surface curvature; and (d) triangular meshes obtained by PowerRTF.

3.2. VCCCPD

By applying the centroid constraint to the conventional power diagram, the centroidal power diagram (CPD) could be produced [DG-BOD12], where each point \mathbf{s}_i is positioned at the center-of-mass \mathbf{s}_i^* of the corresponding power cell $P(\mathbf{s}_i)$, that is,

$$\mathbf{s}_i = \mathbf{s}_i^* = \frac{\int_{P(\mathbf{s}_i)} \mathbf{s} \rho(\mathbf{s}) d\mathbf{s}}{\int_{P(\mathbf{s}_i)} \rho(\mathbf{s}) d\mathbf{s}} \quad (3)$$

where $\rho(\mathbf{s})$ is a C^1 -smooth density function on Ω .

Owing to the introduction of weight to each point, the power diagram has the property of accurate capacity constraint, and a CC-CPD is generated by adding an additional capacity constraint to the CPD. The capacity $m_i = \int_{P(\mathbf{s}_i)} \rho(\mathbf{s}) d\mathbf{s}$ (i.e., area or mass, etc.) of the power cell $P(\mathbf{s}_i)$ is equal to the preset fixed value c_i , that is, $m_i = c_i$.

Recently, Zheng et al. [ZYW*21] provide an extension of CC-CPD, namely VCCCPD, where the capacity constraint of each point is confined to a variable-capacity constraint, that is, a capacity-constrained interval $[c_i^{\min}, c_i^{\max}]$. They present a weight evaluated algorithm to optimize weights of points to satisfy the capacity-constrained interval, that is,

$$c_i^{\min} \leq m_i = \int_{P(\mathbf{s}_i)} \rho(\mathbf{s}) d\mathbf{s} \leq c_i^{\max} \quad (4)$$

Similarly, a VCCCPD degenerates into a CCCPD when the capacity-constrained interval of each point \mathbf{s}_i is a constant value, i.e., $c_i^{\min} = c_i^{\max} = c_i$.

3.3. RTF

Let $M = \{\mathbf{V}^M, \mathbf{F}^M\}$ be a mesh surface that consists of a set of vertices $\mathbf{V}^M = \{\mathbf{v}_i\}_{i=1}^{n_v}$ and a set of triangular facets $\mathbf{F}^M = \{t_i\}_{i=1}^{n_f}$, where n_v and n_f are the number of vertices and triangular facets. Each triangular facet t_i is represented by three ordered vertices, with indices 0, 1, and 2. Let $\mathbf{S} = \{\mathbf{s}_i\}_{i=1}^n$ denote n different sample points on the original surface M . The RTF [YLW*23] defines a set of approximative polygonal facets $\mathbf{F} = \{F_i\}_{i=1}^n$ to fit the original surface M . Each polygonal facet F_i is determined on the basis of the tangent plane of \mathbf{s}_i . To be specific, the RTF F_i of \mathbf{s}_i is composed by the intersections of the tangent plane τ_i and the vertical bisector planes of its neighbor points, that is, $F_i \subset \tau_i$, as shown in Figure 2.

Taking advantage of the independent computation, Yao et al. [YLW*23] present a GPU-based approach to compute the RTF of each sample point, and more regularly distributed sample points could be produced with a projection strategy, yielding high-quality meshes. In this paper, we combine the power diagram with RTF and provide a PowerRTF to capture the surface curvature.

4. PowerRTF based surface remeshing

In this section, we first introduce the concept of PowerRTF and the GPU-assisted PowerRTF construction method. Based on this, combining with the variable-capacity constraint and centroid constraint, a GPU optimization method is presented for surface remeshing, producing quality-controlled meshes.

4.1. PowerRTF and its construction

To compute the RTF F_i of sample point \mathbf{s}_i , Yao et al. [YLW*23] truncate a 3D Voronoi cell $V(\mathbf{s}_i)$ with the tangent plane τ_i , resulting in a polygonal facet. The Voronoi cell $V(\mathbf{s}_i)$ is clipped based on the k -nearest-neighbor points, as shown in the top row in Figure 3. In this paper, we offer the generalized RTF, namely PowerRTF, to approximate the given surface based on power diagrams, as illustrated in the bottom row in Figure 3. Without loss of generality, the PowerRTF of sample point \mathbf{s}_i is also recorded as F_i . Instead of the Voronoi cell in RTF, we apply a 3D power cell $P(\mathbf{s}_i)$, which is pre-clipped according to the k -nearest-neighbor points and their weights. Thus, the PowerRTF F_i of sample point \mathbf{s}_i could be obtained by intersecting the clipped power cell with the tangent plane τ_i .

We first recall some notions of PowerRTF as follows. Let $\mathbf{S} = \{\mathbf{s}_i\}_{i=1}^n$ represent a set of sample points on the original surface M , associated with a set of weights $\mathbf{W} = \{w_i\}_{i=1}^n$. e_{ij} denotes the regular edge between two adjacent sample points \mathbf{s}_i and \mathbf{s}_j , and \mathbf{s}_{ij} is the intersecting point of e_{ij} and the equal power distance plane. According to the definition of the power diagram, the value of $\mathbf{s}_{ij} = (x_{ij}^s, y_{ij}^s, z_{ij}^s)$ could be calculated as [DGBOD12, ZGC*19]:

$$\mathbf{s}_{ij} = \mathbf{s}_i + \frac{\|\mathbf{s}_j - \mathbf{s}_i\|^2 + (w_i - w_j)}{2\|\mathbf{s}_j - \mathbf{s}_i\|^2} \cdot (\mathbf{s}_j - \mathbf{s}_i) \quad (5)$$

As a result, the equation of the equal power distance plane could be calculated using the positions and weights of the sample points \mathbf{s}_i

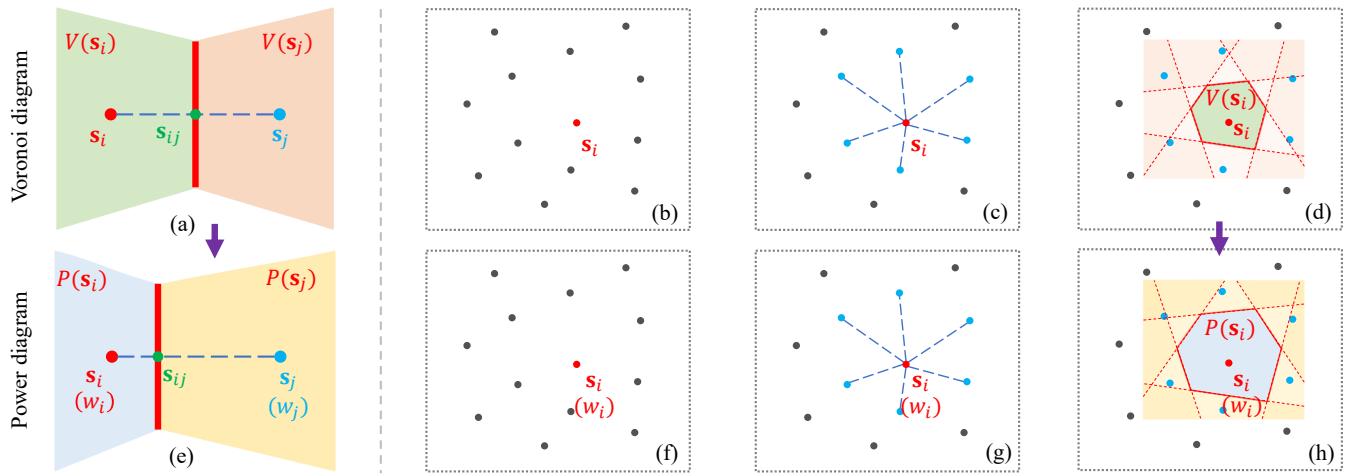


Figure 3: Exemplification of the pre-clipping process of RTF and PowerRTF in 2D space. Top: the pre-clipping process of RTF based on the Voronoi diagram in [YLW*23]. (a) the bisector (red line) of two points s_i and s_j ; (b) sample points; (c) k -nearest-neighbor points (blue dots) of s_i ; and (d) clipped Voronoi cell $V(s_i)$ with its k -nearest-neighbor points. Bottom: the pre-clipping process of PowerRTF on the basis of the power diagram in our work, where w_i is the weight of sample point s_i . (e) the equal power distance line (red line) of two points s_i and s_j ; (f) sample points; (g) k -nearest-neighbor points (blue dots) of s_i ; and (h) clipped power cell $P(s_i)$ with its k -nearest-neighbor points.

and s_j as follows:

$$\begin{cases} x_{ij}^s \cdot x + y_{ij}^s \cdot y + z_{ij}^s \cdot z - \mathbf{n}_{ij} \cdot \mathbf{s}_{ij} = 0 \\ \mathbf{n}_{ij} = \mathbf{s}_i - \mathbf{s}_j \end{cases} \quad (6)$$

where \mathbf{n}_{ij} is the normal of the equal power distance plane.

Similar to RTF computation in [YLW*23], we start with initializing a 3D power cell $P(s_i)$ of the sample point s_i by the bounding box of the input surface M . Based on Equation 6, the initialized power cell $P(s_i)$ is then clipped by the equal power distance facets of its k -nearest-neighbor sample points S_i^k , where k is the number of neighbor sample points, and the default value of k is set to 32 in our work. Thus, we could obtain a 3D clipped power cell $P(s_i)$ with suitable size. According to the position s_i and normal \mathbf{n}_i of the sample point, we could directly calculate the tangent plane τ_i equation, which is also known as RTF or PowerRTF plane equation. Consequently, the 3D clipped power cell $P(s_i)$ is further truncated with the tangent plane τ_i , yielding the PowerRTF F_i of s_i . Owing to the well-designed clipped power cell computation and PowerRTF clipping, the PowerRTF per sample point is constructed independently, which is friendly for GPU implementation. The GPU-assisted construction algorithm of PowerRTF is provided in algorithm 1. Theoretically, the running time complexity of algorithm 1 is $O(n)$ owing to the independence and parallelism of the PowerRTF construction for each sample point.

4.2. PowerRTF optimization

The fundamental principle of PowerRTF is to approximate the given surface by computing a set of polygonal facets in conjunction with the power diagram. In the curved (flat) regions, small (big) power cells are used to capture the surface curvature. However, due to the randomly distributed sample points in the ordinary PowerRTF, meshes of poor quality frequently appear. Besides, as shown in Figure 1(c), there are gaps between the original surface

Algorithm 1: GPU-assisted PowerRTF construction

```

Input: surface  $M$ , sample points  $\mathbf{S} = \{\mathbf{s}_i\}_{i=1}^n$ , weights  $\mathbf{W} = \{w_i\}_{i=1}^n$ , point normal  $\mathbf{N} = \{\mathbf{n}_i\}_{i=1}^n$ 
Output: PowerRTF  $\mathbf{F} = \{F_i\}_{i=1}^n$ 
1 for  $s_i \in S$  in parallel do
2   Initialize  $P(s_i)$  with  $\text{BoundingBox}(M)$ 
3   Search  $k$ -nearest-neighbor points set  $S_i^k$  from  $\mathbf{S}$ 
4   for  $s_j \in S_i^k$  do
5     | Clip  $P(s_i)$  based on Equation 6
6   end
7   Compute the tangent plane  $\tau_i$  based on  $s_i$  and  $\mathbf{n}_i$ 
8   Calculate  $F_i$  by truncating  $P(s_i)$  with  $\tau_i$ 
9 end
```

and the PowerRTFs. The total mass of PowerRTFs for all sample points is not constant but varies with their positions and weights.

Therefore, we impose the centroid constraint and interval capacity constraint to each sample point to produce adaptively distributed sample points, from which curvature-adapted meshes could be extracted. With the introduction of interval capacity constraint, we can vary the capacity-constrained interval of each sample point to achieve the trade-off between the mesh quality and computational efficiency, as shown in Figure 11. Moreover, the capacity constraint per sample point is relaxed from a value to an interval could accelerate the convergence of PowerRTF computation as the optimization process stabilizes and the total mass of PowerRTFs fluctuates within a certain range.

Without loss of generality, the proportion of the capacity-constrained interval is denoted as γ , and \bar{m}_i is the average PowerRTF capacity of all sample point \mathbf{S} . Thus, we introduce the capacity-constrained interval $[c_i^{\min}, c_i^{\max}]$ to each sample point s_i ,

which is defined as: $[(1 - \gamma) \cdot \bar{m}_i, (1 + \gamma) \cdot \bar{m}_i]$. Moreover, based on the density function $\rho(\mathbf{s}_i)$ of a given point \mathbf{s}_i on the original surface M , the mass of the PowerRTF F_i could be calculated as: $m_i = A(F_i) \cdot \rho(\mathbf{s}_i)$, where $A(F_i)$ is the area of F_i . To generate PowerRTF with interval-capacity constraint, we provide a GPU method by alternatively optimizing the weights and positions per iteration.

4.2.1. Weight optimization

According to the definition of power diagrams [DGBOD12], the borderlines of a power cell $P(\mathbf{s}_i)$ (or F_i) is affected by the weight w_i of associated sample point \mathbf{s}_i , which is strongly related to the mass m_i of $P(\mathbf{s}_i)$ (or F_i). Therefore, to produce PowerRTF with interval-capacity constraint, that is, $c_i^{\min} \leq m_i \leq c_i^{\max}$, we adopt the weight evaluated algorithm in [ZYW^{*}21] and provide an improved version in this work. As shown in the Figure 4, let l_{ij} (or l'_{ij} after weight changed) signify the distance between the equal power distance and the point \mathbf{s}_j , i.e., $l_{ij} = \|\mathbf{s}_{ij} - \mathbf{s}_i\|$. Taking the increasing of the weight w_i as an example, let $\Delta w_i = w'_i - w_i > 0$ represent the weight variation of point \mathbf{s}_i , and Δs_{ij} indicate the distance variation of the equal power distance lines (blue to purple lines), i.e., $\Delta s_{ij} = l'_{ij} - l_{ij}$. Based on the definition of power diagrams, Zheng et al. [ZGC^{*}19] provide the formulation of Δs_{ij} as follow:

$$\Delta s_{ij} = \left\| \frac{w'_i - w_i}{2\|\mathbf{s}_i - \mathbf{s}_j\|^2} \cdot (\mathbf{s}_j - \mathbf{s}_i) \right\| = \frac{\Delta w_i}{2d_{ij}} \quad (7)$$

where $d_{ij} = \|\mathbf{s}_i - \mathbf{s}_j\|$ is the distance of points \mathbf{s}_i and \mathbf{s}_j .

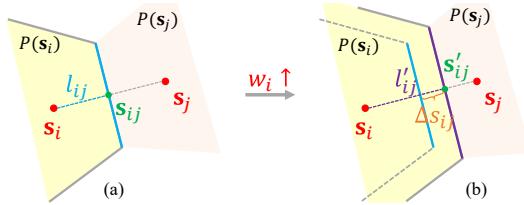


Figure 4: The equal power distance variation Δs_{ij} in 2D space.

Based on the Equation 7, the weight variation Δw_i is determined as: $\Delta w_i = 2d_{ij} \cdot \Delta s_{ij}$. However, the distance d_{ij} between the sample point \mathbf{s}_i and its neighboring sample points are not the same, making it difficult to determine the weight variation in the optimization process. Similar to [ZYW^{*}21], based on the distance D_i between the sample point \mathbf{s}_i and its nearest neighboring sample point and a fixed value of Δs_{ij} , the weight variation Δw_i could be re-calculated as: $\Delta w_i = 2D_i \cdot \Delta s_{ij}$, where Δs_{ij} is set to 0.20 during the optimization phase to control the area variation of PowerRTFs.

In our work, we apply a multiplicative operator $\alpha_i = \lfloor \frac{\Delta m_i}{\Delta c_i} - 1 \rfloor$ of each sample point \mathbf{s}_i to speedup the convergence of the weight evaluated algorithm, where $\Delta m_i = |m_i - 0.5(c_i^{\min} + c_i^{\max})|$ is the capacity difference and $\Delta c_i = 0.5(c_i^{\max} - c_i^{\min})$ is the half size of the capacity-constrained interval. Consequently, the weight w_i is increased as $w_i = w_i + \alpha_i \cdot \Delta w_i$ for the sample point \mathbf{s}_i whose mass m_i less than the left value of the respective capacity-constrained interval. Conversely, we decrease the weight w_i as $w_i = w_i - \alpha_i \cdot \Delta w_i$ when the mass m_i is greater than the right value of the capacity-constrained interval. This process is repeated until all masses of

Algorithm 2: Improved version of the weight evaluate method

```

Input: sample points  $\mathbf{S} = \{\mathbf{s}_i\}_{i=1}^n$ , associated with weights  $\mathbf{W} = \{w_i\}_{i=1}^n$ , capacity constraints  $\mathbf{C} = \{[c_i^{\min}, c_i^{\max}]\}_{i=1}^n$ , maximum weight iteration number  $N_m^w$ 
Output: PowerRTF  $\mathbf{F} = \{F_i\}_{i=1}^n$  with  $(\mathbf{S}', \mathbf{W}')$ 
1 Set the counter  $n_u^w$  to 0
2 while  $\exists s_j, m_j \notin [c_j^{\min}, c_j^{\max}]$  and  $n_u^w < N_m^w$  do
3   for  $s_i \in \mathbf{S}$  in parallel do
4     Compute the multiplicative operator  $\alpha_i$  and weight variation  $\Delta w_i$ 
5     if  $m_i < c_i^{\min}$  then
6       | Update weight  $w_i = w_i + \alpha_i \cdot \Delta w_i$ 
7     else if  $m_i > c_i^{\max}$  then
8       | Update weight  $w_i = w_i - \alpha_i \cdot \Delta w_i$ 
9     end
10   end
11   Call algorithm 1 to construct the PowerRTFs of  $\mathbf{S}$ 
12   Compute  $m_i$  of each sample point  $\mathbf{s}_i$  in parallel
13    $n_u^w \leftarrow n_u^w + 1$ 
14 end
```

the PowerRTFs are inside the relevant capacity-constrained interval, and the procedure is presented in **algorithm 2**. From a theoretical perspective, the running time complexity of **algorithm 2** is $O(n)$, which is similar to **algorithm 1**.

4.2.2. Position optimization

One prevalent method for position optimization is Lloyd's method. In our work, we follow a similar method that each sample point \mathbf{s}_i is relocated to the center-of-mass \mathbf{b}_i of the respective PowerRTF F_i in each iteration. However, the center-of-mass \mathbf{b}_i may be deviated from the surface, resulting in a large distance error (evaluated by the Hausdorff distance) between the output meshes and the original surface. With this regard, we utilize the KNN-based projection strategy in [YLW^{*}23] to restrict the movement of sample points on the original surface, as shown in Figure 5.

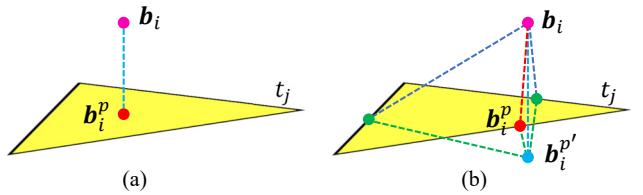


Figure 5: Exemplification of the KNN-based projection strategy, where pink dots represent the center-of-mass \mathbf{b}_i of the PowerRTF F_i , and red dots indicate the projected points \mathbf{b}_i^p of \mathbf{b}_i on the plane of a triangular facet t_j . (a) \mathbf{b}_i is projected inside the triangular facet t_j , and (b) \mathbf{b}_i is projected outside the triangular facet t_j (that is $\mathbf{b}_i^{p'}$), which is further projected onto the borderlines of t_j .

To be specific, for each sample point \mathbf{s}_i , we calculate the center-of-mass \mathbf{b}_i of the associated PowerRTF F_i and select the

Algorithm 3: PowerRTF for surface remeshing

Input: original surface M , number of sampling points n , the proportion γ , maximum number of iterations N_m

Output: triangular meshes M'

- 1 Calculate the normal \mathbf{n}_i^v and density $\rho(\mathbf{v}_i)$ of each vertex $\mathbf{v}_i \in \mathbf{V}^M$
- 2 Sample points $\mathbf{S} = \{\mathbf{s}_i\}_{i=1}^n$ from the input surface M
- 3 Set the initial weights $\mathbf{W} = \{w_i\}_{i=1}^n$ to 1 of \mathbf{S}
- 4 Set the counter n_{it} to 0
- 5 **while** $\exists \mathbf{s}_j \in \mathbf{S}, m_j \notin [c_j^{\min}, c_j^{\max}]$ and $n_{it} < N_m$ **do**
- 6 **for** $\mathbf{s}_i \in \mathbf{S}$ **in parallel do**
- 7 Estimate the normal \mathbf{n}_i and calculate the density $\rho(\mathbf{s}_i)$ of sample point \mathbf{s}_i
- 8 Call **algorithm 1** to construct the PowerRTF F_i
- 9 Calculate the mass m_i
- 10 **end**
- 11 Set the capacity-constrained interval $[c_i^{\min}, c_i^{\max}]$
- 12 **for** $\mathbf{s}_i \in \mathbf{S}$ **in parallel do** // weight
- 13 Call **algorithm 2** to optimize the weight w_i
- 14 **end**
- 15 **for** $\mathbf{s}_i \in \mathbf{S}$ **in parallel do** // position
- 16 Calculate the barycenter \mathbf{b}_i of F_i
- 17 Project the barycenter \mathbf{b}_i to \mathbf{b}_i^p on M
- 18 $\mathbf{s}_i \leftarrow \mathbf{b}_i^p$
- 19 **end**
- 20 $n_{it} \leftarrow n_{it} + 1$
- 21 **end**
- 22 Extract the triangular meshes M' from optimized \mathbf{S}

k -nearest-neighbor vertices \mathbf{N}_i^k from the original surface. Based on these neighboring vertices \mathbf{N}_i^k , the center-of-mass \mathbf{b}_i is projected onto the triangular facets with at least one vertex in \mathbf{N}_i^k , as shown in [Figure 5](#). Consequently, the sample point \mathbf{s}_i is restricted to the projected point with the smallest projection distance, achieving the optimization of sample points on the original surface.

4.3. Surface remeshing

The key idea of our work is to approximate the raw mesh surface using a set of PowerRTFs with multiple constraints, thereby generating adaptively distributed sample points. By varying the capacity-constrained interval of each sample point, a quality-controlled mesh could be extracted from these optimized sample points. Starting with the input raw meshes, our method proceeds as depicted in [Figure 1](#), and the pseudo-code for surface remeshing is provided in [algorithm 3](#). Note that the capacity-constrained interval of each sample point \mathbf{s}_i is set to $[(1 - \gamma) \cdot \bar{m}_i, (1 + \gamma) \cdot \bar{m}_i]$, where \bar{m}_i is the average PowerRTF mass of all sample points in each iteration. Theoretically, the running time of [algorithm 3](#) is $O(n^2)$. We present some implementation details of our method as follows.

Initial Sampling. A simple way to initialize the points \mathbf{S} is randomly sampling from the original surface M . Nevertheless, the resulting points are incongruent with the original surface curvature, which may have a negative impact on the stability and convergence

of the PowerRTF optimization. Therefore, we offer an initialization strategy to sample n points from the original surface with curvature adaptation as follow: (1) For an input surface M , we first estimate the normal \mathbf{n}_i^v of each vertex \mathbf{v}_i , and then calculate and normalize the curvature κ_i^v of \mathbf{v}_i as: $\kappa_i^v \in (0, 1]$; (2) The normalized curvature per vertex is converted to a density function f with positive correlation: $\rho(\mathbf{v}_i) = f(\kappa_i^v)$, e.g., proportional or user-specified functions; (3) From the original surface M , we randomly upsample m -times more points, that is $m \times n$ points \mathbf{S}^\uparrow and calculate their relevant density values; (4) From the point set \mathbf{S}^\uparrow , n pre-sampled points are chosen with density-dependent probability as the initial sample points \mathbf{S} of our method. In our work, m is set to 2. Notably, the initialization strategy could be replaced by other curvature-adapted sampling methods in Geogram [[LF15](#)] or PCL [[RC11](#)].

The initial sample points (purple dots) are shown in [Figure 1\(b\)](#) on a *StandingCat* mesh surface. Obviously, more points are positioned at curved regions, e.g., ears, feet, and tail, etc., and fewer points are located at flat regions, e.g., body, etc. To further improve the stability of PowerRTF construction, we apply ten iterations without weight optimization to these initialized sample points \mathbf{S} , producing more regularly distributed sample points, as shown in [Figure 1\(c\)](#).

Normal estimation and density calculation. The PowerRTF of each sample point is defined as the intersection of a 3D power cell with the tangent plane. Therefore, the normal estimation of each sample point is crucial to the PowerRTF construction in [algorithm 1](#). Unlike previous work [[CZC*18](#), [YLW*23](#)], we calculate the normal \mathbf{n}_i of sample point \mathbf{s}_i based on the k_n -nearest vertices from the original surface M : $\mathbf{n}_i = \frac{1}{k_n} \sum_{i=1}^{k_n} \mathbf{n}_i^v$. Similarly, we could calculate the density value $\rho(\mathbf{s}_i)$ of sample point \mathbf{s}_i as: $\rho(\mathbf{s}_i) = \frac{1}{k_d} \sum_{i=1}^{k_d} \rho(\mathbf{v}_i), \mathbf{v}_i \in \mathbf{V}^M$, where k_n and k_d are the number of nearest-neighbor vertices of \mathbf{s}_i from the original surface M .

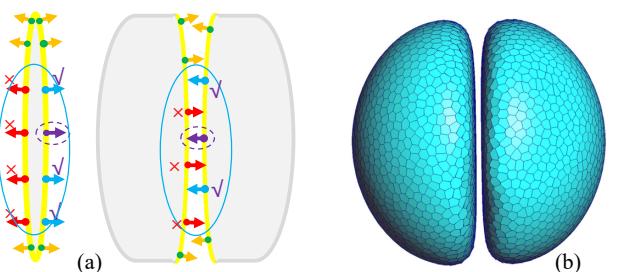


Figure 6: Exemplification of normal verification and remeshing results on a Close Hemisphere model. (a) normal verification of close sheet (left) or separation (right) surface remeshing, where purple dots are the current sample points, dots in blue circular are the k -nearest neighbors, blues dots are taken into account for computation while red dots are not considered; (b) generated PowerRTF.

NVSPC method. In view of approximating the original surface using polygonal facets, the topology of close sheet surface remeshing remains a challenge in earlier work [[CZC*18](#), [YLW*23](#)]. These methods mostly use the k -nearest-neighbor sample points to estimate the normal of each point, resulting in biased or inaccurate normal of points placed at close sheet regions. In our work, an additional NVSPC method is carried out to address this problem while

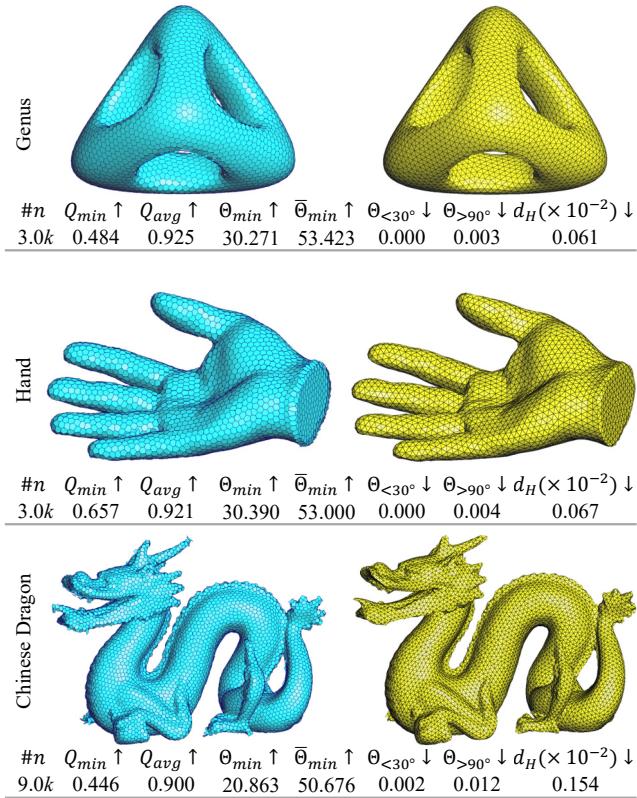


Figure 7: Remeshing results with uniform density on three different models with holes (top), multiple slender regions (middle), and complex topologies (bottom).

computing the normal, pre-clipped power cell, and position optimization of each sample point.

Taking the close sheet as an example, as shown in Figure 6(a), the normal of each sample point is determined based on its k -nearest-neighbor vertices (dots in blue circular) in the original surface M . For a point (purple dot) on the close-plate region, some vertices on the opposite side (red dots) are mistakenly treated as its k -nearest neighbors, leading to incorrect normal estimation. To solve it, we first calculate the angle between the basic normal (purple arrow, average normal of its three-nearest-neighboring vertices in normal estimation, otherwise its own normal) and the normal of its neighboring vertices. Then, these neighboring vertices (blue dots) are taken into consideration only if the angles lie within a given threshold $\Theta^{\#}$. Notably, the value of $\Theta^{\#}$ is set to 60° in our work. Similarly, this normal verification is taken for the 3D power cell pre-clipping in PowerRTF construction and the projection strategy in position optimization of each sample point. Consequently, our method can mostly address the topological errors or outliers in close sheet surface remeshing, producing high-quality meshes, as shown in Figure 6(b).

Mesh extraction. An adaptively distributed sample points could be generated after the PowerRTF optimization, from which the high-quality meshes could be produced. However, as shown in Figure 1(d), the misaligned facets in the PowerRTF optimization make it almost impossible to directly generate the desired triangular mesh

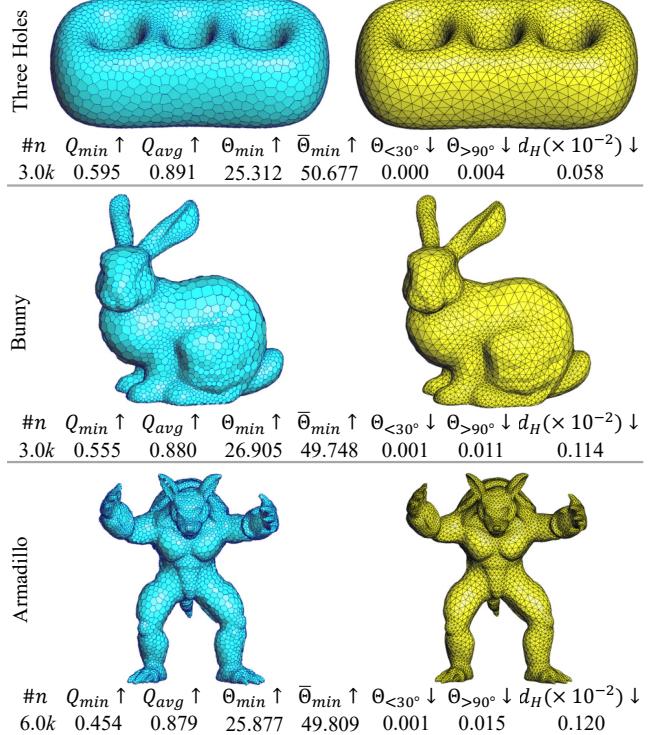


Figure 8: Remeshing results with non-uniform density on three different models with multiple holes (top), narrow regions (middle), and complex topologies (bottom).

since the connectivity is ambiguous. To achieve it, we utilize a robust technique, called the RVD-based mesh extraction method provided in [BL17], which reconnects the input points with triangles based on the calculation of their restricted Voronoi cells. For degenerate cases that could arise where there are four co-circular sample points on the same plane and the possibility of isolated holes, a manifold extraction method and a set of post-processing heuristics to recover a clean mesh, which could be referred to [BL17]. The source code of the RVD-based mesh extraction method [BL17] is provided by the authors in the Geogram library [LF15]. Notably, the optimized sample points are utilized as inputs of the RVD-based mesh extraction method in our work, generating triangular meshes without optimization operations.

5. Experiments

In this section, we demonstrate the capability of the proposed PowerRTF method for surface remeshing and investigate the impact of some parameters. In order to evaluate the remeshing results, we also compare our method with the state-of-the-art approximate CVT-based methods, i.e., the RVC [CZC^{*}18], RTF [YLW^{*}23], and density RVC (D-RVC) [CZC^{*}18], etc. All experiments are performed on a Windows 10 computer with 3.6 GHz Intel (R) Core (TM) i7-9700K CPU with 16 GB memory, and a NVIDIA GeForce RTX 2080 Ti with 11 GB memory, using CUDA version 10.0.

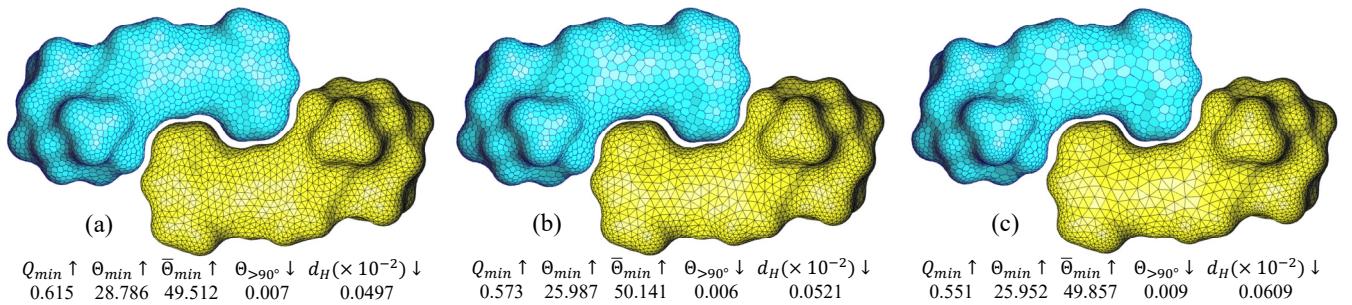


Figure 9: Results of our method with various densities on the Retinal model, including the PowerRTF (blue) and the corresponding triangular meshes (yellow). (a) results with $\rho = f$; (b) results with $\rho = f^2$; and (c) results with $\rho = f^3$.

GPU implementation and parameter setting. The ability of our method to compute and optimize each PowerRTF independently is friendly for a GPU implementation. Here we provide some implementation details of our method. Similar to the 3D clipped Voronoi diagram [RSLL18, LMGY22], the sets for vertices and plane equations are stored in shared memory arrays of constant size max 96 and 64, respectively. The number of thread by blocks is set to 16. The parameter k for the PowerRTF construction is set to 32, and 16 for the projection strategy to restrict the sampling points moving on the surface, that is similar to the RTF in [YLW*23]. Moreover, the parameters k_n and k_d are set to 12; the default value of the proportion of the capacity-constrained interval γ is set to 4%; the default values of N_m^w and N_m are set to 30 and 100; and the number of sampling points n is set to $3.0k$.

Mesh Quality metric. We use the criteria in [KPF*20] to assess the quality of meshes in our experiments. For a triangular t , the quality is determined as $Q_t = \frac{6}{\sqrt{3}} \cdot \frac{A_t}{S_t E_t}$, where A_t , S_t , and E_t are the area, half-perimeter, and the length of the longest edge, respectively. Therefore, the quality of a surface mesh is firstly measured by the minimum triangle quality Q_{min} and the average quality Q_{avg} . Additionally, the quality of a surface mesh is measured by the angles of triangles, including the minimal angles of all triangles Θ_{min} , the average of the minimal angles $\bar{\Theta}_{min}$, and the percentage of triangles with angles less than 30° , recorded as $\Theta_{<30^\circ}$, or greater than 90° , denoted by $\Theta_{>90^\circ}$. We also report the approximation error in view of the Hausdorff distance d_H , that is normalized by the diagonal of the bounding box, and the computational time $T(s)$ of surface remeshing. Notably, the symbols \uparrow or \downarrow indicate that the larger or smaller the value corresponds to a better quality result.

5.1. Uniform and adaptive remeshing

Owing to the capacity constraint characteristic of the power diagram, our method offers the capability of surface remeshing with uniform and adaptive densities (user-defined or surface curvature adaptation). Therefore, we start by evaluating our method on a variety of models exhibiting different challenges ranging from smooth surfaces, e.g., *Genus*, *Three Holes*, etc., to complex surfaces, e.g., *Bunny*, *Dragon*, and *Armadillo*, etc. The relevant results of our method with uniform density are shown in Figure 7, and these results with adaptive density are provided in Figure 8. The remeshing results prove the effectiveness of our method. In view of the quality of generated meshes, when the density of the model surface varies

widely, there are a few sample points with large power cells in the flat regions while more sample points with small power cells in the curved regions, such as the ears in the Bunny, etc. In this case, there may be a few triangular facets with small angles, leading to some slightly inferior values of the mesh quality metric, e.g., the minimal triangle quality Q_{min} and the minimal angle Θ_{min} in Figure 8.

5.2. Parameters analysis

The quality of generated results are controlled by the values of some parameters, i.e., the density ρ , the proportion of the capacity-constrained interval γ , and the number of sample points n . We now investigate the effects of these parameters on the generated results of the proposed PowerRTF.

Analysis of density. Firstly, we test the impact of the density ρ on the PowerRTF optimization and the generated triangular meshes. In this experiment, we select a relatively smooth surface, namely *Retinal* model, as the basis input of our method. In the *initial sampling* phase, three different positive correlation functions are utilized to calculate the normal curvature κ_i^v of each vertex v_i on the original surface M , from the primary, quadratic to cubic functions respectively. Thus, the density value $\rho(s_i)$ per sample point s_i is varying with the given functions based on the density calculation strategy in our method. Figure 9 demonstrate the impact of the density ρ on the *Retinal* model.

From the results in Figure 9, it can be observed that our method is effective in constructing and optimizing PowerRTFs of sample points at different densities. Simultaneously, these results also demonstrate the ability of PowerRTF to characterize the surface curvature, which is consistent with the motivation of our work. To be specific, more sample points with small power cells are used to approximate the curved regions, while less with large power cells are applied to capture the flat regions in the original surface. Moreover, the curvatures change more dramatically, i.e., the greater the density value, the greater the difference of the relevant PowerRTFs, which is a reflection of the impact of the density ρ on the generated triangular meshes, as shown in Figure 9. This behavior of our method in terms of ρ is consistent for different models.

Analysis of interval and number of sample points. Next, we study the impact of varying both the proportion γ of the capacity-constrained interval and the number n of sample points. The relatively smooth *Fertility* model is taken as input of PowerRTF. In this

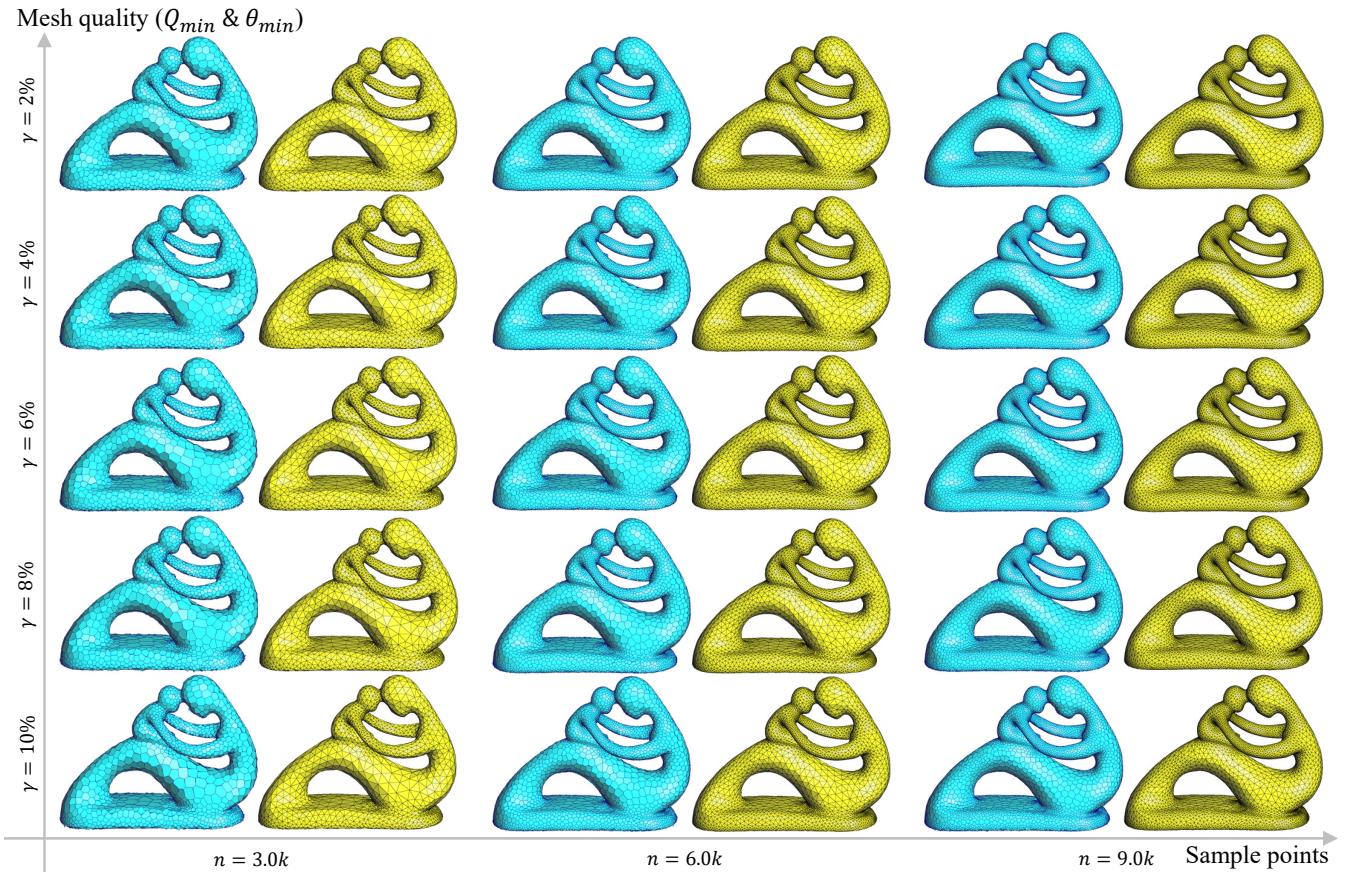


Figure 10: The impact of the proportion γ of the capacity-constrained interval and the number n of sampling points, demonstrated on the Fertility surface, where blue results represent the optimized PowerRTFs and yellow results indicates the generated triangular meshes.

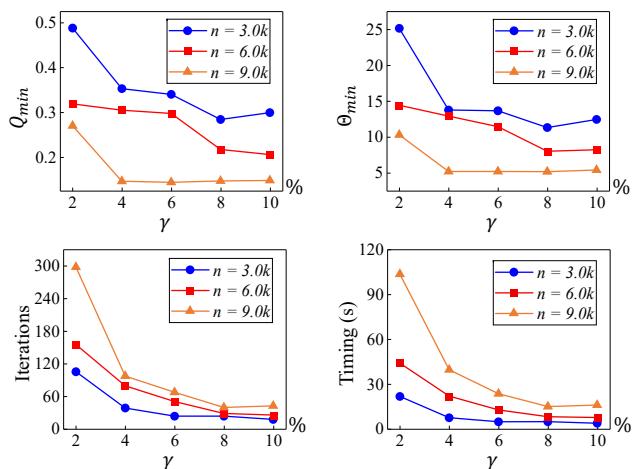


Figure 11: Impact of the proportion of the capacity-constrained interval γ and the number of sampling points n . Top: the minimum triangle quality Q_{min} , the minimum angle Θ_{min} . Bottom: number of iterations of our method, and running time of our method.

experiment, the proportion γ of the capacity-constrained interval is selected from a set $\{2\%, 4\%, 6\%, 8\%, 10\%\}$, and the number n of sample points is set to $\{3.0k, 6.0k, 9.0k\}$, respectively. Figure 10 il-

lustrates the visual comparisons of these results for a combination of 3×5 parameter settings, and the quality and running time of each combination in Figure 10 are reported in Figure 11. It should be mentioned that the value of N_m is set to 200 when the value of γ is 2%, because our method requires more iterations and time consumption under small capacity-constrained intervals.

From a qualitative perspective, as shown in Figure 10, the generated PowerRTFs and triangular meshes are influenced by the distribution of sample points, which varies with the capacity-constrained interval. Meanwhile, these triangular meshes are affected by the ability of surface curvature characterization, that is related to the number of sample points. Specifically, small value of the parameter γ brings about a more adaptively distributed sample points, from which we could obtain meshes of higher quality, as shown in these results with $\gamma = 2\%$ in Figure 10. Besides, large number of sample points has the better capability to capture the surface curvature, i.e., more sample points are located at the curved regions in the original surface, as shown in these results with $n = 9.0k$ in Figure 10.

From a quantitative perspective, as shown in Figure 11, the quality and the running time of generated meshes are affected by the parameters γ and n . More generally, a small value of γ , i.e., a small capacity-constrained interval, produces more regularly distributed sample points, thus yielding meshes of higher quality, as shown in

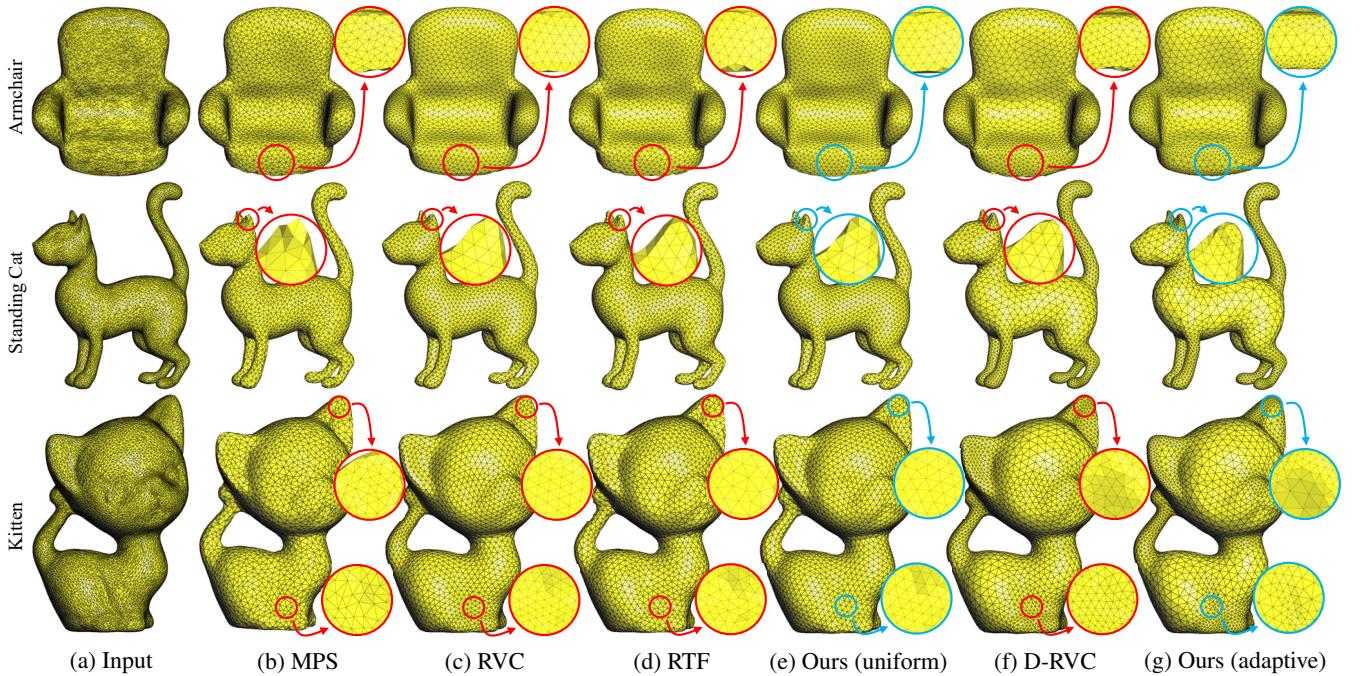


Figure 12: The visual comparison of various models on three models. The quality and relevant time of these results are reported in [Table 1](#).

the top row in [Figure 11](#). However, this typically requires more iterations and time consumption, as shown in the bottom row in [Figure 11](#). Moreover, more iterations and time consumption are also necessary for surface remeshing with more sample points. When more sample points are used for surface remeshing, PowerRTF could characterize curved regions in a better manner. Nevertheless, the density is discretized during the density calculation per sample point in our method, leading to a large density variability in flat-to-curved regions. Therefore, there may be a few slender and small-angle triangular facets in these results using more sample points.

5.3. Comparison

We compare against the RVC [[CZC^{*}18](#)] and the RTF [[YLW^{*}23](#)] as representatives of state-of-the-art surface approximation algorithms based on CVT without auxiliary points and the MPS [[GYJZ15](#)] as representative of the surface remeshing algorithm based on CVT computation in parameter space. Notably, the MPS is a CPU-based method, RVC is a multi-threading method on the CPU, and RTF is a GPU-assisted method. Besides, the RVD [[YLL^{*}09](#)], as a representation of the exact CVT-based method, is also taken for comparison in close sheet surface remeshing.

Comparison on normal models. Firstly, we select several methods for the comparison of uniform and adaptive remeshing, including the MPS [[GYJZ15](#)], RVC [[CZC^{*}18](#)], RTF [[YLW^{*}23](#)], and D-RVC [[CZC^{*}18](#)], respectively. Notably, the MPS, RVC and RTF apply CVT for optimizing the polygonal facets without taking into account the surface curvature to approximate the original surface. Conversely, the D-RVC is an improved version of the RVC proposed by Chen et al. [[CZC^{*}18](#)], which applies a density term in the energy function to capture the surface curvature. Therefore, the MPS, RVC and RTF are used for comparison with uniform remesh-

ing, whereas the D-RVC is employed for comparison with adaptive remeshing. Three normal models, including the *Armchair*, *StandingCat*, and *Kitten*, are taken as inputs for various methods, where the number of sample points n are set to $3.0k$ to maintain consistency. Besides, all parameters in our method are set to their default values. [Figure 12](#) illustrates the remeshing results of different methods, and the quality and running time are reported in [Table 1](#).

According to these results in [Figure 12](#) and [Table 1](#), it is clear that our method is successful in remeshing different simple models at both uniform and adaptive densities, producing relatively high-quality meshes. In view of uniform surface remeshing, our method produces almost the best results for three different inputs. Compared to MPS, our method generates meshes of higher quality, but the reported time of MPS is better than ours, as the MPS calculates the 2D CVT in the parameter space. Compared to the RVC and RTF, our method combines power diagrams and RTF to generate a more regularly distributed sample points based on the property of the precise capacity constraint in the power diagram. In contrast, the Voronoi diagram is used in RVC and RTF without the capability of capacity constraint. Moreover, similar to RTF, the sample points are projected onto the original surface rather than a best-fitted plane in RVC, achieving small Hausdorff distance errors, as shown in [Table 1](#). In terms of computational efficiency for uniform remeshing, the superiority of our method is attributed to two aspects. On one hand, the initial sampling strategy in this paper yields relatively regularly distributed sample points, which are fed as inputs for subsequent optimization phase. On the other hand, to achieve quality-controlled remeshing, the capacity constraint is relaxed to an interval in our method rather than a fixed value, which also improves the optimization efficiency, as shown in [Figure 11](#).

Considering adaptive surface remeshing, our method achieves

Table 1: The quality and running time of various methods on different models, where the red values indicate the best results in different types.

Model	Type	Method	$Q_{min} \uparrow$	$Q_{avg} \uparrow$	$\Theta_{min} \uparrow$	$\bar{\Theta}_{min} \uparrow$	$\Theta_{<30^\circ} \downarrow$	$\Theta_{>90^\circ} \downarrow$	$d_H (\times 10^{-2}) \downarrow$	$T(s) \downarrow$
Armchair	Uniform	Input	0.024	0.633	1.257	33.468	0.398	0.505	-	-
		MPS	0.443	0.798	26.974	44.421	0.006	0.166	0.083	0.472
		RVC	0.616	0.903	33.151	52.152	0.000	0.010	0.105	4.450
		RTF	0.484	0.883	26.149	50.185	0.001	0.015	0.087	6.882
		Ours	0.707	0.929	33.724	53.805	0.000	0.000	0.073	1.565
	Adaptive	D-RVC	0.529	0.880	21.242	50.951	0.001	0.022	0.135	5.731
		Ours	0.476	0.886	25.730	49.789	0.001	0.014	0.066	13.121
	-	Input	0.118	0.704	6.467	37.537	0.245	0.353	-	-
StandingCat	Uniform	MPS	0.477	0.800	27.350	44.665	0.004	0.159	0.107	0.561
		RVC	0.599	0.900	34.935	51.965	0.000	0.012	0.094	4.762
		RTF	0.530	0.889	27.618	50.651	0.001	0.009	0.066	7.140
		Ours	0.671	0.917	32.389	52.726	0.000	0.000	0.066	2.754
	Adaptive	D-RVC	0.559	0.907	31.819	52.513	0.000	0.007	0.099	6.541
		Ours	0.600	0.899	31.007	51.245	0.000	0.004	0.068	11.652
	-	Input	0.060	0.701	3.842	37.369	0.256	0.361	-	-
Kitten	Uniform	MPS	0.470	0.799	26.608	44.379	0.005	0.166	0.086	0.387
		RVC	0.646	0.901	35.509	51.936	0.000	0.009	0.067	4.615
		RTF	0.547	0.887	29.677	50.490	0.000	0.009	0.057	7.591
		Ours	0.687	0.925	36.532	53.468	0.000	0.000	0.054	1.658
	Adaptive	D-RVC	0.597	0.891	30.217	51.325	0.000	0.015	0.124	5.429
		Ours	0.581	0.891	29.494	50.686	0.000	0.006	0.052	8.856

comparable results with comparison to the D-RVC. From a qualitative perspective, our method is more consistent with the original surface in characterizing the surface curvature, e.g., the base of *Armchair*, the ear of *StandingCat* and *Kitten*, etc. In D-RVC, the density per sample point is calculated based on the two principle curvatures of the input points, and this yields density values similar in the curved regions. Besides, there is no guarantee in the Voronoi diagram that polygonal facets are of certain masses or capacities, so they are typical quite different. Unlike D-RVC, our method calculates the density per sample point based on the normal curvatures of vertices in original surface, and the masses of PowerRTFs in our work similar owing to the property of precise capacity constraint in power diagrams. Therefore, our method could better characterize the surface curvature, as shown in the red and blue circles in [Figure 12](#). From a quantitative perspective, the quality of meshes generated by our method is comparable to D-RVC in some metric, but inferior in others, e.g., Q_{min} and Θ_{min} . The main reason is the presence of some slender and small-angle triangular facets in the regions where curvatures change dramatically on the original surface. In terms of computational efficiency for adaptive remeshing, a weight optimization process is necessary in our method, which requires more time consumption. Overall, though there are some weakness, our work surpasses the counterparts in curvature adaptive ability.

Comparison on close sheet models. Close sheet topology models are particularly important for physical simulations. It has been a challenge for remeshing work, especially for algorithms that approximating the surfaces with polygonal facets. Next, we compare with two existing algorithms in remeshing a close sheet surface, including the RVD [[YLL*09](#)] as a representation of exact CVT-based method and the D-RVC [[CZC*18](#)] as a representation of approximate CVT-based technique. [Figure 13](#) presents the results gener-

ated different methods on the *Elk* model, where the number n of sample points is set to 6.0k for three methods.

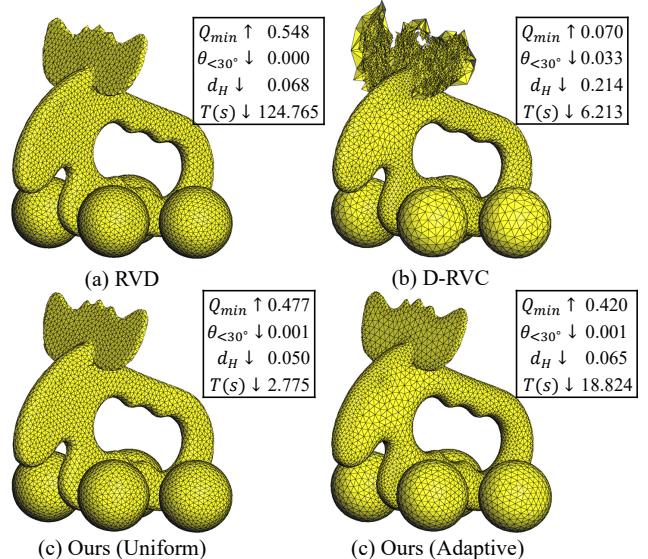


Figure 13: Comparison results of different methods on the *Elk* model with close sheet topology.

As shown in [Figure 13](#), RVD calculates the exact intersections between 3D Voronoi cells and surfaces, producing relatively better triangular meshes with no topological errors or outliers. However, the calculation of precise intersections is too complicated and time-consuming, and the surface curvatures are not considered in surface remeshing. Different from RVD, D-RVC and our method utilize planar polygonal facet to approximate the original surface with

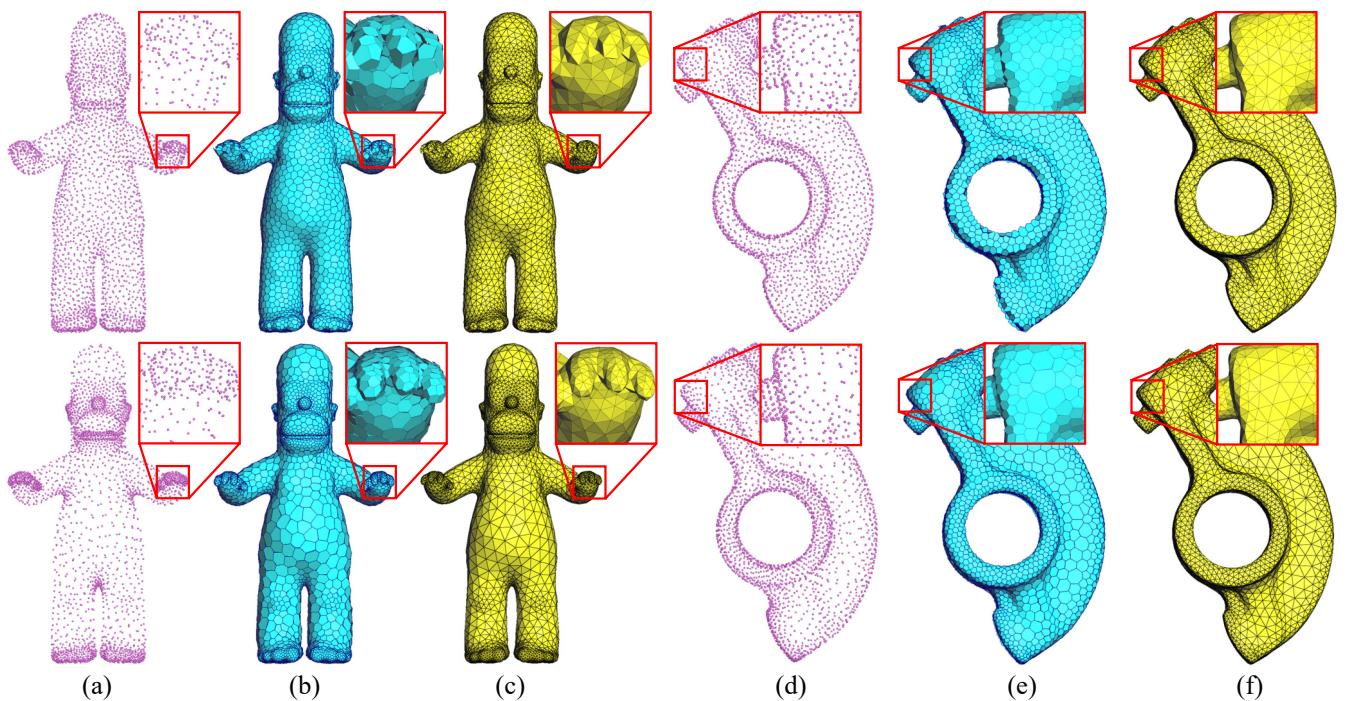


Figure 14: Experimental results for the weighted sample point ablation on the Homer and Rocker models. Top: results with unweighted sample points; bottom: results with weighted sample points. (a) Optimized sample points on the Homer model, (b) optimized PowerRTF on the Homer model, (c) generated triangular meshes on the Homer model, (d) optimized sample points on the Rocker model, (e) optimized PowerRTF on the Rocker model, and (f) generated triangular meshes on the Rocker model.

consideration of curvature adaption, greatly improving the computational efficiency. However, the sample points in D-RVC are projected onto a best-fitting plane using the k nearest neighbors, leading to the appearance of outliers and topological errors, as shown in Figure 13(b). Fortunately, our method restricts the movement of sample points on the original surface to avoid the outliers in optimizing stage. Moreover, a normal verification is applied during the normal estimation, PowerRTF construction, and position optimization phases in our method, which circumvents the impact of inverse side points, generating more accurate PowerRTFs and meshes. In terms of surface curvature adaption, the densities of sample points at close sheet regions are similarly high in D-RVC. Our method could capture the variation of densities more accurately in close sheet regions. More precisely, the densities at the center of close sheet regions are relatively low while densities at the corners or edges are mostly high, as shown in Figure 13(d).

5.4. More results

To further demonstrate the effectiveness and advantages of our method, we provide more experimental results, both from ablation experiments and on more complex models.

Ablation experiment. A major contribution of our work is the combination of the RTF with the power diagram. By introducing weight characteristic to each sample point, we compute the power cell on the corresponding tangent plane for each weighted sample point. To evaluate the advantage of weighted sample point for adaptive surface remeshing, we conduct an ablation experiment on the

Homer and Rocker models. By keeping everything else the same (including the initial strategy and NVISPC method described in Section 4.3), we present experimental results for surface remeshing with weighted and unweighted sample points, as shown in Figure 14, where the number of sample points is set to $3.0k$, and the Table 2 presents the quality of these generated triangular meshes. According to the results in Figure 14 and Table 2, we could observe that our method with weighted sample points could generate better curvature-adapted approximative facets and the relevant triangular meshes. In particular, for these high curvature regions, such as the finger regions in the Homer model, our method with weighted sample points could achieve better curvature adaption by adjusting the weights of sample points to control the size of approximative facets.

Furthermore, another contribution of our work is the proposed NVISPC method. The NVISPC method is used in computing the normal, pre-clipping power cell and position optimization of each sample point, achieving more accurate normal, PowerRTF, and projected point results, as described in Section 4.3. Similarly, to validate the importance of the NVISPC method in our method, we carry out another ablation experiment on two close sheet surfaces, i.e., the Disc and Fish models. In this experiment, we keep everything else the same (including the initial strategy, weighted sample points, etc., described in Section 4.3) and analyze the remeshing results with and without the NVISPC method. Figure 15 shows two groups of remeshing results on different models, and the quality of generated triangular meshes is provided in Table 2. The NVISPC method could produce more accurate PowerRTF and triangular meshes. By calculating the angle between the basic normal and the

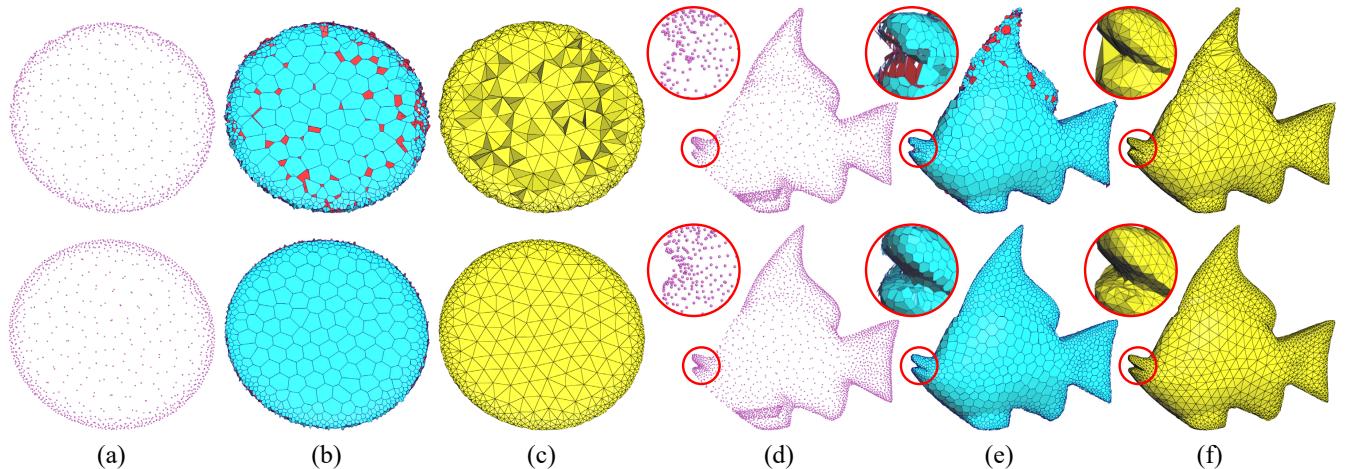


Figure 15: Experimental results for the NVISPC ablation on the Disc and Fish models. Top: results of our method without the NVISPC, bottom: results of our method with the NVISPC. (a) Optimized sample points on the Disc model, (b) optimized PowerRTF on the Disc model, (c) generated triangular meshes on the Disc model. (d) optimized sample points on the Fish model, (e) optimized PowerRTF on the Fish model, (f) generated triangular meshes on the Fish model.

Table 2: Triangle quality of meshes generated by our method, corresponding to these results in Figure 14, Figure 15, and Figure 16.

Model	Result	n	$Q_{min} \uparrow$	$Q_{avg} \uparrow$	$\Theta_{min} \uparrow$	$\bar{\Theta}_{min} \uparrow$	$\Theta_{<30^\circ} \downarrow$	$\Theta_{>90^\circ} \downarrow$	$d_H (\times 10^{-2}) \downarrow$
Homer	Figure 14(c)(Top)	3.0k	0.489	0.877	22.088	49.622	0.001	0.017	0.051
	Figure 14(c)(Bottom)	3.0k	0.460	0.882	22.438	50.020	0.001	0.015	0.030
Rocker	Figure 14(f)(Top)	3.0k	0.502	0.871	23.046	49.031	0.002	0.014	0.054
	Figure 14(f)(Bottom)	3.0k	0.507	0.884	26.902	50.013	0.001	0.012	0.043
Disc	Figure 15(c)(Top)	3.0k	0.066	0.710	2.260	38.072	0.256	0.316	0.049
	Figure 15(c)(Bottom)	3.0k	0.308	0.836	12.474	46.605	0.019	0.079	0.022
Fish	Figure 15(f)(Top)	3.0k	0.051	0.792	1.710	43.421	0.096	0.148	0.083
	Figure 15(f)(Bottom)	3.0k	0.394	0.871	14.939	49.029	0.004	0.020	0.052
David Head	Figure 16(b)(Top)	30.0k	0.350	0.875	14.376	49.605	0.001	0.021	0.036
	Figure 16(b)(Bottom)	10.0k	0.472	0.882	20.497	50.030	0.002	0.018	0.075
Gargoyle	Figure 16(d)(Top)	30.0k	0.275	0.883	13.172	50.231	0.001	0.016	0.043
	Figure 16(d)(Bottom)	10.0k	0.408	0.880	19.571	49.914	0.001	0.022	0.039

normal of its neighboring vertices, these adjacent vertices with significant angle variation are not considered in the pre-clipping process. Thus, we could obtain more accurate PowerRTFs, as shown in the *Disc* model in Figure 15(f). Similarly, the NVISPC method is also used in the position optimization process to circumvent the projection of sample points onto the inverse side, achieving better remeshing results, as shown in the *Fish* model in Figure 15(g). However, the NVISPC method is heuristic, and the quality of generated meshes may be degraded because of the frequency details on close sheets or even complex models, as shown in Table 2.

Results on more complex models. It is a very challenging task for approximative CVT-based remeshing techniques to produce remeshing results of more complex models, e.g., RTF [YLW*23], etc. To demonstrate the capability of our method, two extremely complicated models are taken as inputs of our algorithm, including the *David Head* model and the *Gargoyle* model. Due to the existence of more details in these complex models, we select a relatively large number of sampling points for uniform remeshing. To be specific, an comprehensive experiment is performed, where the

number of sampling points is set to 30.0k for uniform remeshing and 10.0k with surface curvature adaption remeshing. Figure 16 demonstrate the remeshing results of our method on the *David Head* and *Gargoyle* models, and Table 2 presents the quality of these generated meshes. The blue models represent the optimized PowerRTF results, and the yellow models indicate the triangular meshes generated by our method on different cases. Besides, the appendix video presents more remeshing results of our method. From these results in Figure 16 and the appendix video, we could observe that our method is also applicable for more complex surface remeshing, which reflects the superiority of our method. However, the quality of generated triangular meshes is degraded on more complex or close sheet with high frequency details.

6. Conclusion

In this paper, we introduce an improved technique of approximative CVT-based, called PowerRTF, for surface remeshing with curvature adaption. To achieve it, we present an extension of RTF in combination with the power diagram to curvature-adapted approx-

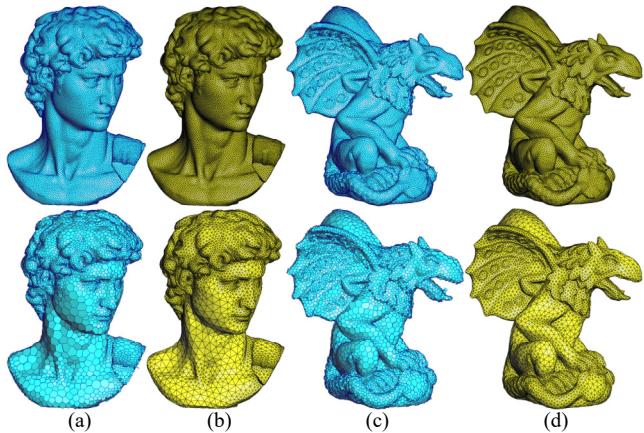


Figure 16: Remeshing results with our method on two more complex models, including the David Head model and the Gargoyle model. Top: uniform remeshing results with 30.0k sampling points; and bottom: non-uniform remeshing results with 10.0k sampling points. From left to right: (a) PowerRTF on the David Head model, (b) generated triangular meshes on the David Head model, (c) PowerRTF on the Gargoyle model, and (d) generated triangular meshes on the Gargoyle model.

imate the original surface. We impose the variable-capacity constraint and centroidal constraint each sample point by relaxing the capacity constraint to an interval, providing the trade-off between mesh quality and computational efficiency. What's more, an additional NVISPC method is provided to address the topological errors and outliers, generating topological correct triangular meshes in close sheet surface remeshing. Experimental results on various models, including those in this paper, the supplementary materials, demonstrate the effectiveness of our method.

Limitations and further work. There are still a few limitations in our work. Firstly, there is a potential for short edges in PowerRTFs and the triangular meshes, resulting in narrow or small-angle triangles in the regions where curvatures change dramatically, which affects the quality of generated meshes. This could be solved by applying a mesh improvement technique as postprocessing [YW15]. Another limitation is that remeshing with sharp feature models, e.g., *Fandisk*, *Block*, etc., remains a challenge for our method, as there is no guarantee that several sample points are moved on corners or edges. Finally, the efficiency of weight optimization has an impact on the convergence of our method. Consideration may be given to a numerical optimization technique that is similar to RVC [CZC*18], e.g., Newton's method, the quasi-Newton method, etc. What's more, there is no theory to prove its validity as there are gaps between the PowerRTF of each sample point.

Acknowledgments

The models used in this paper are from the AIM@Shape and courtesy of the Standford 3D Scanning Repository. We would like to thank Zhonggui Chen for providing their executable programs and Bruno Lévy for the open source code of the RVD-based mesh extraction method. This work was supported in part by

the National Natural Science of Foundation of China (61972128, 62172415) and the National Key Research and Development Plan (2022YFC3900800).

References

- [ABE*20] ABDELKADER A., BAJAJ C. L., EBEIDA M. S., MAHMOUD A. H., MITCHELL S. A., OWENS J. D., RUSHDI A. A.: Vorocrust: Voronoi meshing without clipping. *ACM Transactions on Graphics (TOG)* 39, 3 (2020), 1–16. [doi:10.1145/3337680.2.3](https://doi.org/10.1145/3337680.2.3)
- [ADVDI03] ALLIEZ P., DE VERDIRE E. C., DEVILLERS O., ISENBERG M.: Isotropic surface remeshing. In *2003 Shape Modeling International* (2003), IEEE, pp. 49–58. [doi:10.1109/SMI.2003.1199601.3](https://doi.org/10.1109/SMI.2003.1199601.3)
- [AHA98] AURENHAMMER F., HOFFMANN F., ARONOV B.: Minkowski-type theorems and least-squares clustering. *Algorithmica* 20, 1 (1998), 61–76. [doi:10.1007/PL00009187.3](https://doi.org/10.1007/PL00009187.3)
- [AK00] AURENHAMMER F., KLEIN R.: Voronoi diagrams. *Handbook of Computational Geometry* 5, 10 (2000), 201–290. [doi:10.1016/B978-044482537-7/50006-1.3](https://doi.org/10.1016/B978-044482537-7/50006-1.3)
- [AUGA08] ALLIEZ P., UCELLI G., GOTSMAN C., ATTENE M.: Recent advances in remeshing of surfaces. *Shape Analysis and Structuring* (2008), 53–82. [doi:10.1007/978-3-540-33265-7_2.2](https://doi.org/10.1007/978-3-540-33265-7_2.2)
- [Aur87] AURENHAMMER F.: Power diagrams: properties, algorithms and applications. *SIAM Journal on Computing* 16, 1 (1987), 78–96. [doi:10.1137/0216006.3](https://doi.org/10.1137/0216006.3)
- [Bal09] BALZER M.: Capacity-constrained Voronoi diagrams in continuous spaces. In *2009 sixth International Symposium on Voronoi Diagrams* (2009), IEEE, pp. 79–88. [doi:10.1109/ISVD.2009.28.2](https://doi.org/10.1109/ISVD.2009.28.2)
- [BL17] BOLTACHEVA D., LÉVY B.: Surface reconstruction by computing restricted Voronoi cells in parallel. *Computer-Aided Design* 90 (2017), 123–134. [doi:10.1016/j.cad.2017.05.011.3.8](https://doi.org/10.1016/j.cad.2017.05.011.3.8)
- [BLM17] BUET B., LEONARDI G. P., MASNOU S.: A varifold approach to surface approximation. *Archive for Rational Mechanics and Analysis* 226, 2 (2017), 639–694. [doi:10.1007/s00205-017-1141-0.2](https://doi.org/10.1007/s00205-017-1141-0.2)
- [CMZZ21] CHEN Y., MA Z., ZHOU H., ZHANG W.: 3D Print-Scan resilient localized mesh watermarking. In *2021 IEEE International Workshop on Information Forensics and Security (WIFS)* (2021), IEEE, pp. 1–6. [doi:10.1109/WIFS53200.2021.9648379.1](https://doi.org/10.1109/WIFS53200.2021.9648379.1)
- [CZC*18] CHEN Z., ZHANG T., CAO J., ZHANG Y. J., WANG C.: Point cloud resampling using centroidal Voronoi tessellation methods. *Computer-Aided Design* 102 (2018), 12–21. [doi:10.1016/j.cad.2018.04.010.2.3.7.8.11.12.15](https://doi.org/10.1016/j.cad.2018.04.010.2.3.7.8.11.12.15)
- [DFG99] DU Q., FABER V., GUNZBURGER M.: Centroidal Voronoi tessellations: Applications and algorithms. *SIAM Review* 41, 4 (1999), 637–676. [doi:10.1137/S0036144599352836.2](https://doi.org/10.1137/S0036144599352836.2)
- [DGBOD12] DE GOES F., BREEDEN K., OSTROMOUKHOV V., DESBRUN M.: Blue noise through optimal transport. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 1–11. [doi:10.1145/2366145.2366190.4.6](https://doi.org/10.1145/2366145.2366190.4.6)
- [DGJ03] DU Q., GUNZBURGER M. D., JU L.: Constrained centroidal Voronoi tessellations for surfaces. *SIAM Journal on Scientific Computing* 24, 5 (2003), 1488–1506. [doi:10.1137/S1064827501391576.2.3](https://doi.org/10.1137/S1064827501391576.2.3)
- [DLY*18] DU X., LIU X., YAN D.-M., JIANG C., YE J., ZHANG H.: Field-aligned isotropic surface remeshing. *Computer Graphics Forum* 37, 6 (2018), 343–357. [doi:10.1111/cgf.13329.3](https://doi.org/10.1111/cgf.13329.3)
- [EI14] ENGWIRDA D., IVERS D.: Face-centred Voronoi refinement for surface mesh generation. *Procedia Engineering* 82 (2014), 8–20. [doi:10.1016/j.proeng.2014.10.364.1](https://doi.org/10.1016/j.proeng.2014.10.364.1)
- [ER15] EULITZ M., REISS G.: 3D reconstruction of SEM images by use of optical photogrammetry software. *Journal of Structural Biology* 191, 2 (2015), 190–196. [doi:10.1016/j.jsb.2015.06.010.1](https://doi.org/10.1016/j.jsb.2015.06.010.1)

- [GYJZ15] GUO J., YAN D.-M., JIA X., ZHANG X.: Efficient maximal Poisson-disk sampling and remeshing on surfaces. *Computers & Graphics* 46 (2015), 72–79. doi:10.1016/j.cag.2014.09.015. 11
- [HHA17] HERHOLZ P., HAASE F., ALEXA M.: Diffusion diagrams: Voronoi cells and centroids from diffusion. *Computer Graphics Forum* 36, 2 (2017), 163–175. doi:10.1111/cgf.13116. 3
- [HYB*16] HU K., YAN D.-M., 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 (2016), 2560–2573. doi:10.1109/TVCG.2016.2632720. 2
- [HYWZ17] HAN J., YAN D.-M., WANG L., ZHAO Q.: Computing restricted Voronoi diagram on graphics hardware. In *Proceedings of the 25th Pacific Conference on Computer Graphics and Applications* (2017), pp. 23–26. doi:10.2312/pg.20171320. 3
- [HZW*22] HOU W., ZONG C., WANG P., XIN S., CHEN S., LIU G., TU C., WANG W.: SDF-RVD: Restricted Voronoi diagram on signed distance field. *Computer-Aided Design* 144 (2022), 103166. doi:10.1016/j.cad.2021.103166. 3
- [KPF*20] KHAN D., PLOPSKI A., FUJIMOTO Y., KANBARA M., JABEEN G., ZHANG Y. J., ZHANG X., KATO H.: Surface remeshing: A systematic literature review of methods and research directions. *IEEE Transactions on Visualization and Computer Graphics* 28, 3 (2020), 1680–1713. doi:10.1109/TVCG.2020.3016645. 2, 9
- [LF15] LÉVY B., FILBOIS A.: Geogram: a library for geometric algorithms. In *VII International Conference on Adaptive Modeling and Simulation (ADMOS 2015)* (2015), CIMNE, p. 45. 7, 8
- [LFXH17] LIU Y.-J., FAN D., XU C.-X., HE Y.: Constructing intrinsic Delaunay triangulations from the dual of geodesic Voronoi diagrams. *ACM Transactions on Graphics (TOG)* 36, 2 (2017), 1–15. doi:10.1145/2999532. 2, 3
- [LL10] LÉVY B., LIU Y.: L_p centroidal Voronoi tessellation and its applications. *ACM Transactions on Graphics (TOG)* 29, 4 (2010), 1–11. doi:10.1145/1778765.1778856. 3
- [LMGY22] LIU X., MA L., GUO J., YAN D.-M.: Parallel computation of 3D clipped Voronoi diagrams. *IEEE Transactions on Visualization and Computer Graphics* 28, 2 (2022), 1363–1372. doi:10.1109/TVCG.2020.3012288. 9
- [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 Transactions on Graphics (TOG)* 28, 4 (2009), 1–17. doi:10.1145/1559755.1559758. 2
- [LZ11] LIANG X., ZHANG Y.: Hexagon-based all-quadrilateral mesh generation with guaranteed angle bounds. *Computer Methods in Applied Mechanics and Engineering* 200, 23–24 (2011), 2005–2020. doi:10.1016/j.cma.2011.03.002. 1
- [ME16] MANSOURI S., EBRAHIMNEZHAD H.: Segmentation-based semi-regular remeshing of 3D models using curvature-adapted subdivision surface fitting. *Journal of Visualization* 19 (2016), 141–155. doi:10.1007/s12650-015-0288-8. 2
- [MMGD11] MULLEN P., MEMARI P., DE GOES F., DESBRUN M.: HOT: Hodge-optimized triangulations. In *ACM SIGGRAPH 2011 papers* (2011), pp. 1–12. doi:10.1145/1964921.1964998. 2
- [PJ19] PANCHAL D. M., JAYASWAL D. J.: Surface remeshing using quadric based mesh simplification and minimal angle improvement. In *2019 International Conference on Data Science and Engineering (ICDSE)* (2019), IEEE, pp. 90–95. doi:10.1109/ICDSE47409.2019.8971801. 2
- [RC11] RUSU R. B., COUSINS S.: 3D is here: Point Cloud Library (PCL). In *2011 IEEE International Conference on Robotics and Automation* (2011), IEEE, pp. 1–4. doi:10.1109/ICRA.2011.5980567. 7
- [RSLL18] RAY N., SOKOLOV D., LEFEBVRE S., LÉVY B.: Meshless Voronoi on the GPU. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–12. doi:10.1145/3272127.3275092. 3, 9
- [WWWG22] WANG N., WANG B., WANG W., GUO X.: Computing medial axis transform with feature preservation via restricted power diagram. *ACM Transactions on Graphics (TOG)* 41, 6 (2022), 1–18. doi:10.1145/3550454.3555465. 3
- [WWX*22] WANG P., WANG Z., XIN S., GAO X., WANG W., TU C.: Restricted Delaunay triangulation for explicit surface reconstruction. *ACM Transactions on Graphics (TOG)* (2022). doi:10.1145/3533768. 3
- [WXW*20] WANG P., XIN S., TU C., YAN D., ZHOU Y., ZHANG C.: Robustly computing restricted Voronoi diagrams (RVD) on thin-plate models. *Computer Aided Geometric Design* 79 (2020), 101848. doi:10.1016/j.cagd.2020.101848. 3
- [WYL*15] WANG X., YING X., LIU Y.-J., XIN S.-Q., WANG W., GU X., MUELLER-WITTIG W., HE Y.: Intrinsic computation of centroidal Voronoi tessellation (CVT) on meshes. *Computer-Aided Design* 58 (2015), 51–61. doi:10.1016/j.cad.2014.08.023. 3
- [XLC*16] XIN S.-Q., LÉVY B., CHEN Z., CHU L., YU Y., TU C., WANG W.: Centroidal power diagrams with capacity constraints: Computation, applications, and extension. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 1–12. doi:10.1145/2980179.2982428. 2
- [XXT19] XU M., XIN S., TU C.: An efficient surface remeshing algorithm based on centroidal power diagram. In *Proceedings of the 2019 7th International Conference on Information Technology: IoT and Smart City* (2019), pp. 536–542. doi:10.1145/3377170.3377200. 2, 3
- [YBZW14] YAN D.-M., BAO G., ZHANG X., WONKA P.: Low-resolution remeshing using the localized restricted Voronoi diagram. *IEEE Transactions on Visualization and Computer Graphics* 20, 10 (2014), 1418–1427. doi:10.1109/TVCG.2014.2330574. 3
- [YLL*09] YAN D.-M., 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. doi:10.1111/j.1467-8659.2009.01521.x. 2, 3, 11, 12
- [YLY*23] YAO Y., LIU J., WU W., ZHANG G., XU B., ZHENG L.: Accelerating surface remeshing through GPU-based computation of the restricted tangent face. *Computer Aided Geometric Design* 104 (2023), 102216. doi:10.1016/j.cagd.2023.102216. 2, 3, 4, 5, 6, 7, 8, 9, 11, 14
- [YW15] YAN D.-M., WONKA P.: Non-obtuse remeshing with centroidal Voronoi tessellation. *IEEE Transactions on Visualization and Computer Graphics* 22, 9 (2015), 2136–2144. doi:10.1109/TVCG.2015.2505279. 3, 15
- [YWLL13] YAN D.-M., WANG W., LÉVY B., LIU Y.: Efficient computation of clipped Voronoi diagram for mesh generation. *Computer-Aided Design* 45, 4 (2013), 843–852. doi:10.1016/j.cad.2011.09.004. 3
- [YYY*19] YE Z., YI R., YU M., LIU Y.-J., HE Y.: Geodesic centroidal Voronoi tessellations: Theories, algorithms and applications. *arXiv preprint arXiv:1907.00523* (2019). doi:10.48550/arXiv.1907.00523. 2, 3
- [ZCHK13] ZIMMER H., CAMPEN M., HERKRATH R., KOBELT L.: Variational tangent plane intersection for planar polygonal meshing. In *Advances in Architectural Geometry 2012*. Springer, 2013, pp. 319–332. doi:10.1007/978-3-7091-1251-9_26. 2, 3
- [ZGC*19] ZHENG L., GUI Z., CAI R., FEI Y., ZHANG G., XU B.: GPU-based efficient computation of power diagram. *Computers & Graphics* 80 (2019), 29–36. doi:10.1016/j.cag.2019.03.011. 4, 6
- [ZYW*21] ZHENG L., YAO Y., WU W., XU B., ZHANG G.: A novel computation method of hybrid capacity constrained centroidal power diagram. *Computers & Graphics* 97 (2021), 108–116. doi:10.1016/j.cag.2021.04.007. 2, 4, 6
- [ZZCJ14] ZHUANG Y., ZOU M., CARR N., JU T.: Anisotropic geodesics for live-wire mesh segmentation. *Computer Graphics Forum* 33, 7 (2014), 111–120. doi:10.1111/cgf.12479. 3