A Remeshing Method via Adaptive Multiple Original-Facet-Clipping and Centroidal Voronoi Tessellation

Yue Fei
Hefei University of Technology,
Hefei 230601, China

Jingjing Liu
Hefei University of Technology,
Hefei 230601, China

Yuyou Yao
School of Public Safety and Emergency Management, Anhui University of Science & Technology
Hefei 231131, China

Yusheng Peng
Hefei University of Technology,
Hefei 230601, China

Liping Zheng
Hefei University of Technology,
Hefei 230601, China
zhenglp@hfut.edu.cn

Abstract

CVT (Centroidal Voronoi Tessellation)-based remeshing optimizes mesh quality via the Voronoi-Delaunay framework, optimizing vertex distribution and generating regular triangles. Current CVT-based approaches can fall into two categories. The former are exact methods, such as Geodesic CVT and Restricted Voronoi Diagrams(RVD), which ensure high quality but require significant computation. The latter are approximate methods, which reduce computational complexity yet compromise quality. To address this trade-off, we propose a CVT-based surface remeshing method that balances optimization between quality and efficiency via curvature-adaptive multi-clipping of 3D Centroidal Voronoi cells using original surface facets. The core idea of the method is that we adaptively adjust the number of clipping times according to local curvature, and use the angular relationship between the normal vectors of neighboring facets to represent the magnitude of local curvature. Experimental results demonstrate the effectiveness of our method.

1. Introduction

Surface remeshing improves computational efficiency and model accuracy by refining mesh distribution and reducing complexity. It applied in animation, game development, physics simulations, and computational fluid dynamics, offering benefits like improved real-time rendering or numerical stability depending on the field. Recently, numerous methods have been proposed to address various objectives in surface remeshing, including local approaches [8, 10], anisotropic methods [31], optimization-based techniques [21], and CVT-based methods [28]. Khan et al. [11] provides a detailed review of these advancements, outlining their classification, strengths, limitations, and future research opportunities.

CVT [5] partitions space by positioning Voronoi sites at their cell centroids. Its dualization generates triangular meshes [4, 15]. To apply CVT for generating surface remeshing models, researchers compute the precise intersections between CVT cells and the original model [17], or extend the distance definitions in the Voronoi diagram. For example, Rong et al. [25] applied the CVT generation algorithm to spherical and hyperbolic spaces to improve the remeshing of 3D models. Liu et al. [19] used geodesic distances to construct the geodesic Voronoi diagram (GVD). Though these methods produce exact remeshing results, their computational efficiency is low due to the costly computations of intersections and distances.

Due to the inefficiency of exact surface remeshing methods, researchers are exploring approximate techniques to efficiently simulate 3D model surface. VoroCrust adds [1] boundary auxiliary points to create Voronoi cells simulating surfaces. The Restricted Tangent Face (RTF) method [38] uses CVT to approximate 3D surface boundary with tangent planes. The Power Diagram based Restricted Tan-

gent Face (PowerRTF) method [37] uses the capacity-constrained property of the Power diagram to adaptively implement surface remeshing. While well-designed and easy to implement, these algorithms struggle with highly complex models that feature intricate curvature variations and details, such as hair-like structures and sharp features, which may lead to topological errors like overlapping facets or non-manifold edges.

To balance between enhancing output mesh quality and computational complexity in CVT-based surface remeshing, we propose a method that clips 3D Centroidal Voronoi cells using curvature-adaptive original surface facets. It is an approximate method that performs multiple clips on individual Voronoi cells, enabling better approximation of the original model. This process allows for the selection of optimal clipping facets and determines the required number of clips per cell. Additionally, GPU acceleration achieves parallel computation across individual Voronoi cells. The main contributions of the proposed method are as follows:

- We propose a curvature-adaptive method that sets the number of clipping operations per cell based on the estimated local curvature. Through this adaptive multi-clipping process, the generated polygonal facets achieve high-quality model remeshing.
- We propose a neighborhood-ring-based search strategy that locates neighboring facets with the highest Voronoi cell clipping potential by initiating from vertices of the original facet hosting the sample.

The remainder of this paper is organized as follows: A brief review of several CVT-based surface remeshing methods is provided in Sec. 2. Fundamentals of CVT are introduced in Sec. 3. Our method is detailed in Sec. 4, and experimental results are presented in Sec. 5. Finally, conclusions and limitations are given in Sec. 6.

2. Related work

With the solid theoretical foundation and ability to improve mesh quality, CVT-based methods constitute a core approach in this field. For instance, NASM [16] proposes a GNN-driven CVT method using a high-dimensional normal metric, which avoids precomputed curvature and manual tagging. However, due to training data limitations, it struggles with sparse-vertex CAD models. CWF [32] uses CVT with normal anisotropy and a decaying weight scheme to balance accuracy, quality, and feature preservation, though it faces challenges in efficiently computing the anisotropic RVD. This section focuses on CVT-based remeshing algorithms. For a broader overview of remeshing advances, we refer readers to survey works such as [11].

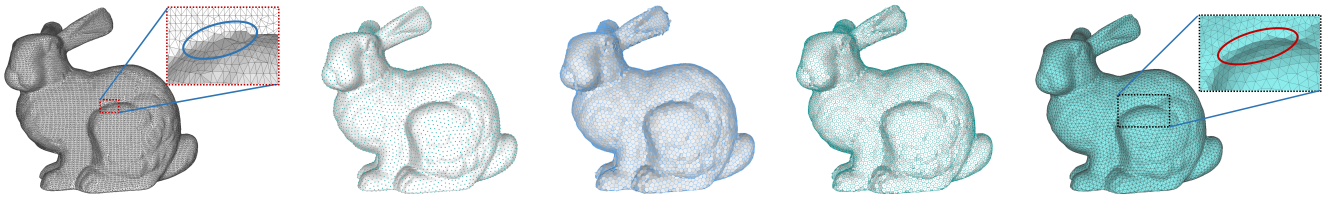
2.1. Exact CVT-based remeshing

The core challenge of exact CVT-based remeshing is the iterative computation of Voronoi diagrams on complex surfaces, requiring numerical optimization to minimize CVT energy for even sample distribution [17]. Leveraging the orthogonal relationship between meshes and their duals, high-quality triangular meshes can be generated from the duals of Voronoi diagrams over a set of sample points [23].

Existing methods for computing the exact Voronoi diagram on a surface can be broadly divided into two main categories. The first type computes the RVD [34], determining the intersection between Voronoi cells and the model surface to accurately capture boundaries for high-quality remeshing. Subsequent research has extensively utilized and extended RVD. For example, [13] applies it to high-dimensional surface remeshing, and Yan et al. [35] improves its computational efficiency for 3D closed models. To address non-contiguous regions, Yan et al. [36] introduced LRVD. Other advancements include adding constraints to minimize obtuse angles [33], optimizing degrees for mesh refinement [6], tailoring robust algorithms for thin-sheet models [27], and extending RVD with signed distance fields [9], demonstrating its versatility. The second type extends distance metrics for more accurate surface Voronoi diagrams, notably by computing geodesic distances for geodesic Voronoi diagrams. GCVT [39] and GVD [29][22] have been developed for this purpose. Researchers have employed GVD to construct Delaunay triangulations (DT) for remeshing applications, as demonstrated in [18] and [20]. Despite the superior results of exact CVT-based methods, they require substantial computational cost.

2.2. Approximative CVT-based remeshing

To reduce computational cost in CVT-based remeshing, approximate methods have been developed. These approximate methods can fall into two categories. The first type employs tangent plane methods. For instance, Zimmer et al. [40] introduces additional degrees of freedom and deformed intersection formulas to fit tangent planes to the model surface. Chen et al. [4] proposes an approach involving clipped planes and Restricted Voronoi Cell (RVC) construction via site position resampling, and further extends CVT by integrating density functions to achieve density-adapted RVC (D-RVC). Notably, Yao et al. [38] introduces the RTF algorithm, which employs plane cutting constrained by tangents to approximate model boundaries. Furthermore, Yao et al. [37] extends Voronoi diagrams to Power diagrams, proposing the PowerRTF algorithm for adaptive curvature-aware remeshing. These methods offer advantages in terms of implementation simplicity and robustness, but their applicability is limited to specific types of models. The second type uses auxiliary points, often based on ingenious geometric constructions. For instance, the RPF method [30] constructs Voronoi diagrams on input model boundaries via shadow points, enabling efficient computation and high-quality triangular mesh generation. VoroCrust [1] robustly constructs



(a) Input (b) Initial sampling (c) Initial clipped (d) Optimized clipped (e) Output

Figure 1. Remeshing workflow applied to a 3D bunny scan model. (a) Original model with 35.292k vertices and 70.580k facets, containing irregular and skinny triangles. (b) Initial sample points placed on the original model. (c) Clipped facets of model based on initial sample points. (d) Optimal clipped mesh after iteratively refining the sample point positions. (e) Output mesh with 7k vertices and 13.996k facets, showing significantly more uniform and regular triangles. Notably, geometrically complex regions, such as the leg-body junctions, exhibit much improved triangulation.

Voronoi cells along model boundaries by selecting circle centers and using their intersections as CVT sites. Overall, these methods are efficient and less complex algorithmically, but require significant geometric and mathematical expertise to implement.

Our method belongs to the approximate approaches, but it acts as a compromise between general approximate methods and exact methods, aiming to achieve high-quality remeshing while avoiding excessive computational time.

3. Centroidal Voronoi Tessellation

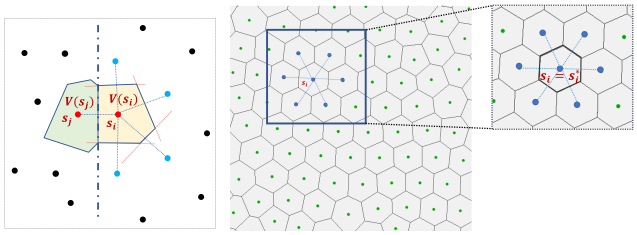
The Voronoi Tessellation [26] partitions a domain Ω around a set of sample points $V(S)$, where each region $V(s_i)$ contains points nearest to $s_i \in V(S)$(Fig. 2(a)):

$$V(s_i) = \{s \in \Omega | \|s - s_i\| \leq \|s - s_j\|, \forall j \neq i\} \quad (1)$$

Adding centroid constraints yields the Centroidal Voronoi Tessellation (CVT) [17](Fig. 2(b)):

$$s_i = \frac{\int_{V(s_i)} s\rho(s)ds}{\int_{V(s_i)} \rho(s)ds} \quad (2)$$

where s_i coincides with the cell centroid, and $\rho(\mathbf{s})$ is a C^1-smooth density function on Ω.



(a) Voronoi Diagram (b) CVT

Figure 2. Illustration of Voronoi Diagram and CVT.

4. Method

In this section, we present our method in detail, organized as follows: initial data processing is described in 4.2, followed by the adaptive clipping strategy in 4.3, centroid update computation in 4.4, and final mesh extraction in 4.5.

4.1. Overview

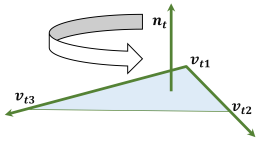
Let the input surface triangular mesh be defined as

$$\mathbf{M} = (\mathbf{V_M}, \mathbf{F_M})$$

$\mathbf{F_M} = \{f_t\}_{t=1}^{n_f}$ denotes the set of triangular faces, where each face f_t is represented by an ordered triplet of vertex indices. $\mathbf{V_M} = \{\boldsymbol{v}_{t,j} \mid t \in \{1, \ldots, n_f\},\ j \in \{1, 2, 3\}\}$ denotes the ordered vertex coordinates, with $\boldsymbol{v}_{t,j}$ being the j-th vertex of the t-th face. The workflow, which is further detailed in Alg. 1, consists of four phases: (1) **initial sampling**, (2) **compute cell and clipping**, (3) **update sites**, and (4) **mesh extraction**. It is also illustrated in Fig. 1.

4.2. Initialization

During this phase, we perform two primary operations: (1) generating a set of sample points $\mathbf{S} = \{\mathbf{s}_i\}_{i=1}^{n}$ via uniform surface sampling on $\mathbf{M}$ using the Geogram library [14], and (2) computing the normal vector $\boldsymbol{n}_t$ for each face f_t, the normal vector is computed by applying the right-hand rule Eq. 3 to its three vertex coordinates.



$$\boldsymbol{n}_t = (\boldsymbol{v_{t3}} - \boldsymbol{v_{t1}}) \times (\boldsymbol{v_{t2}} - \boldsymbol{v_{t1}}) \quad (3)$$

4.3. Clipping Strategy

We adopt the parallelized clipping framework from [24] for Voronoi cell $V(\mathbf{s}_i)$ computation and polyhedral clipping. Its uses key mechanisms: KNN-driven Voronoi computation, where the Voronoi cell of site $\mathbf{s}_i$ is iteratively clipped by bisecting planes $\Pi(\mathbf{s}_i, \mathbf{s}_k)$ generated by its K nearest neighbors $\{\mathbf{s}_k\}_{k=1}^{K}$; half-space partitioning, which defines the vertex set of the convex cell, identifies and removes vertices on the non-retained side, computes intersections between the half-plane and crossing edges, connects these points cyclically, and updates the vertex set (seen in Fig. 3).

For CVT computation, Lloyd's method is adopted over Newton's method. Newton's method requires geometric

Algorithm 1: Surface remeshing using Original-Facet-Clipping and Centroidal Voronoi Tesssellation

Input: original surface $\mathbf{M}$, number of sampling points n,error termination ϵ, maximum number of iterations N_m
Output: triangular meshes $\mathbf{M}'$
`// Phase 1: Initialization`
$\mathbf{S} = \{\mathbf{s}_i\}_{i=1}^{n}$ are uniformly sampled from $\mathbf{M}$
Calculate the normal $\boldsymbol{n}_t$ for each original mesh face
Set the counter n_{it} to 0
`// Phases 2-3: Iterative Optimization`
while $\delta > \epsilon$ *and* $n_{it} < N_m$ **do**
  for $\mathbf{s}_i \in \mathbf{S}$ *in parallel* **do**
    `// Compute and clip the cell`
    Compute the Centroid Voronoi Cell $V(\mathbf{s}_i)$ use Lloyd's method
    Construct the neighboring facet set F_{near}
    Determine the number of clipping counts
    Select original facets for clipping
    Clip the cell based on the frame work [24]
    `// Update position of site`
    Calculate the barycenter b_i
    Calculate the projection point b_i^p by projecting the barycenter b_i on $\mathbf{M}$
    $s_i \leftarrow b_i^p$
  $n_{it} \leftarrow n_{it} + 1$
`// Phase 4: Mesh Extraction`
Extract the triangular meshes

continuity between Voronoi cells, but clipping operations introduce discontinuous gaps that disrupt this continuity. Lloyd's method imposes no continuity constraints and provides superior robustness.

Constructing neighboring facet set. The number of clipping operations for each Voronoi cell $V(\mathbf{s}_i)$ is set adaptively between 1 and 3, based on the estimated local curvature at its site $\mathbf{s}_i$. This design directly addresses a trade-off in geometry processing. Exact calculation of curvature on discrete triangle meshes is not only computationally expensive but also intrinsically ill-posed, as it requires fitting local analytic surfaces to inherently noisy data, which amplifies noise and is highly sensitive to the specific tessellation. In contrast, computing face normals is stable and can be performed more efficiently. To align with the efficiency requirements of our CVT framework, we adopt a robust strategy where local curvature is approximated by the maximum dihedral angle between adjacent triangular faces, derived from precomputed face normals. This angle directly quantifies the rate of surface normal variation, providing a stable and computationally trivial geometric signal. By linking the number of clipping passes to this measure, our method maintains simplicity while achieving the overarching goal of generating high-quality, feature-aware tessellations.

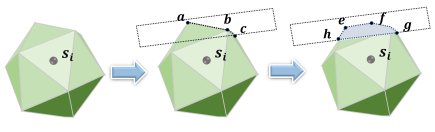
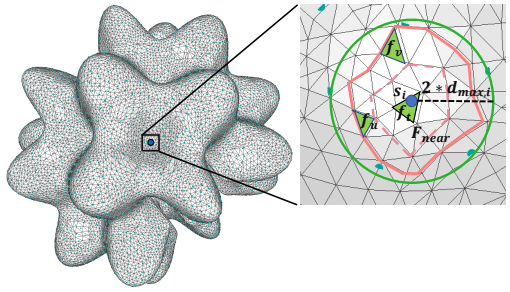
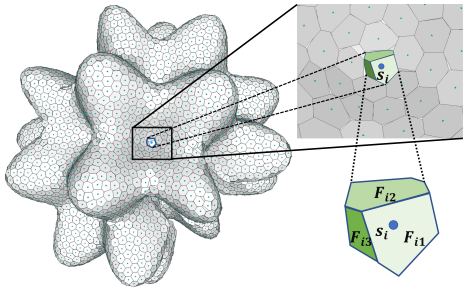


Figure 3. Half-space clipping removes vertices a, b, c and connected edges on the removal side, then computes intersections e, f, g, h between the clipping plane and cell edges to generate a new facet by connecting these points.

To achieve the curvature-adaptive clipping strategy, we first construct a neighboring facet set F_{near} containing original facets potentially enclosed by the cell $V(\mathbf{s}_i)$, enabling both curvature estimation (via dihedral angles) and clipping facet selection. To reduce algorithmic complexity, we include all facets within the twice-ring neighborhood of f_t. However, as elongated one-ring facets may position some twice-ring facets too far from f_t, we constrain the maximum distance between the centroid of any facet in the set and the centroid of f_t to be no more than twice the distance to the farthest neighboring site $\mathbf{s}_k$. This limits the search to original facets approximately enclosed by the cell, as shown in Fig. 4(a).



(a) Construction of neighboring facet set



(b) The model with clipped cell (by original-facets)

Figure 4. (a) Construction of neighboring facet set F_{near} for site $\mathbf{s}_i$ (blue point). Red dashed line indicates first-layer neighborhood-ring of facet $\mathbf{f}_t$ containing $\mathbf{s}_i$. Pink solid line indicates second layer neighborhood-ring. Green circle circle radius is $2 \times d_{\max,i}$ with $d_{\max,i} = \max_{\mathbf{s}_k \in N(\mathbf{s}_i)} |\mathbf{s}_k - \mathbf{s}_i|$. F_{near} contains facets within both rings and circle. Angle cosine computation relative to $\mathbf{f}_t$ determines three required clippings, identified using Eq. 4 and Eq. 5. (b) Resulting clipped cell with three sub-facets: F_{i1} (light green), F_{i2} (dark green), and F_{i3} (deepest green).

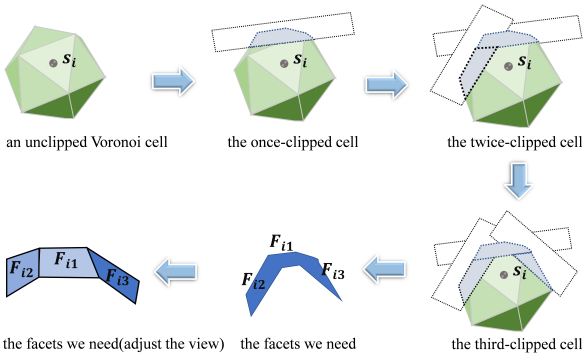


Figure 5. The Voronoi cell undergoes three sequential clipping operations along the arrow direction, starting from the unclipped state at the top-left and terminating at the bottom-right. The algorithm retains only the facets F_{i1}, F_{i2}, F_{i3} generated by these clips. The middle facet in the bottom row shows the final retained result, while the leftmost element provides an alternative view for clearer inspection of the facets.

Determining the number of clipping counts. The angles between facets in F_{near} and original facet f_t are assessed using face normals. Level 1 curvature (one clipping) occurs when absolute cosine values exceed $\boldsymbol{\alpha} = 0.8$. If any facet in F_{near} has an absolute cosine value of the angle with f_t below $\boldsymbol{\alpha}$, Level 2 curvature (at least two clippings) is identified.

When two clippings are required and f_u is the second clipping facet, Level 3 curvature (third clipping) is confirmed if any facet in F_{near} has absolute cosine values below $\boldsymbol{\beta} = 0.7$ relative to both f_t and f_u. The three-clipping process is shown in Fig. 5, with resulting surface facets in Fig. 4((b)).

Selecting original facets for clipping. When a cell $V(\mathbf{s}_i)$ requires a second clipping, eligible facets in F_{near} must satisfy $\cos\theta_A < \alpha$ where $\theta_A \in [0, \pi]$ is the angle between their normals and host facet f_t containing s_i. The second clipping facet f_u minimizes:

$$\text{scoreA} = |\cos A| + \frac{\text{disA}}{d_{\max,i}} \tag{4}$$

with disA being Euclidean distance to f_t's centroid, and $d_{\max,i} = \max_{\mathbf{s}_k \in N(\mathbf{s}_i)} \|\mathbf{s}_k - \mathbf{s}_i\|$.

For three clippings, eligible facets must have $\cos\theta_B > \beta$ and $\cos\theta_C > \beta$ for angles $\theta_B, \theta_C \in [0, \pi]$ relative to f_t and f_u. The third clipping facet f_v minimizes:

$$\text{scoreB} = \sum_{\theta \in \{\theta_A, \theta_B, \theta_C\}} \left(\cos\theta + \frac{\text{dis}_\theta}{d_{\max,i}} \right) \tag{5}$$

Cross-cell GPU parallelism is implemented, but intra-cell clipping remains sequential.

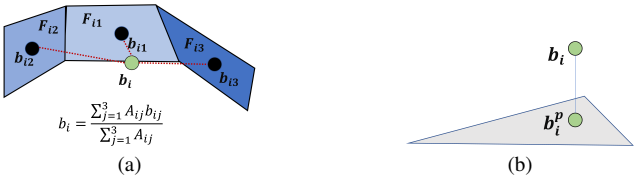


Figure 6. Centroid calculation: (a) Weighted centroid of three patches via Eq. 6; (b) Surface projection for external centroids following [38].

4.4. Calculating the centroid

During iterative optimization, centroids of clipped cells are recalculated each iteration as initial sites for subsequent CVT. Since our method targets surface remeshing, centroids are computed through area-weighted averaging of clipped facets rather than complete Voronoi cells:

$$\mathbf{b}_i = \frac{\sum_{i=1}^{m} A_{ij} \mathbf{b}_{ij}}{\sum_{j=1}^{m} A_{ij}} \tag{6}$$

where A_j and $\mathbf{b}_j$ are the area and centroid of facet j, with m being the facet count per Voronoi cell. This prioritizes surface geometry during updates, as shown in Fig. 6((a)).

Centroids $\mathbf{b}_i$ off the original surface project back using an RTF-inspired method: identify k nearest surface vertices $\mathbf{vm}_i^k$, compute projection distances to their containing triangles, then project to the closest triangle to preserve topological fidelity Fig. 6((b)).

4.5. Extracting meshes

The RVD-based method in [3], accessible via the Geogram library [14], is employed for mesh extraction. The optimized sampling points $\mathbf{S}$ serve as direct input to the RVD method, facilitating triangular mesh generation.

5. Experiments

We validate our method on representative models, comparing with CVT-based methods: RVD [34], PowerRTF [20], RVC [4], and MPS [7]. All experiments were performed on Windows 11 with 2.1 GHz i7-13700 CPU, 32GB RAM, NVIDIA RTX 4070 Ti (12GB), CUDA 12.1.

Remeshing quality is evaluated using geometric and efficiency metrics. Angular distribution is analyzed using the minimum ($\Theta_{\min}$) and maximum ($\Theta_{\max}$)angles, along with the percentage of angles smaller than 30° ($\Theta_{<30°}$) or exceeding 90° ($\Theta_{>90°}$). These criteria are based on fundamental angle conditions for finite element convergence, which ensure bounded interpolation error and prevent numerical instability [2, 12]. Following [11], we compute per triangle its area (A), semiperimeter (S), and longest edge length (E). Triangle quality is quantified as:

$$Q = \frac{6}{\sqrt{3}} \cdot \frac{A}{S \cdot E} \tag{7}$$

Global geometric accuracy is evaluated using:

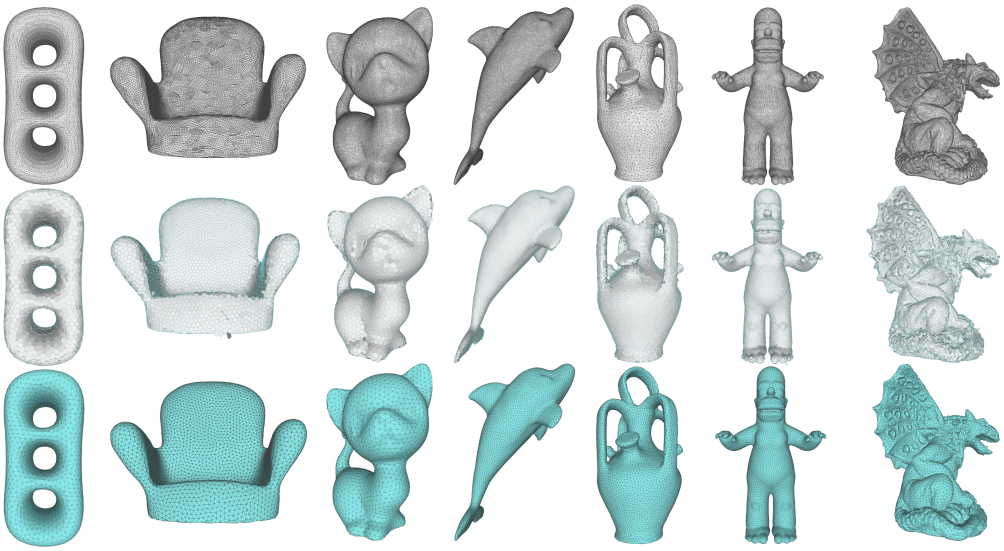
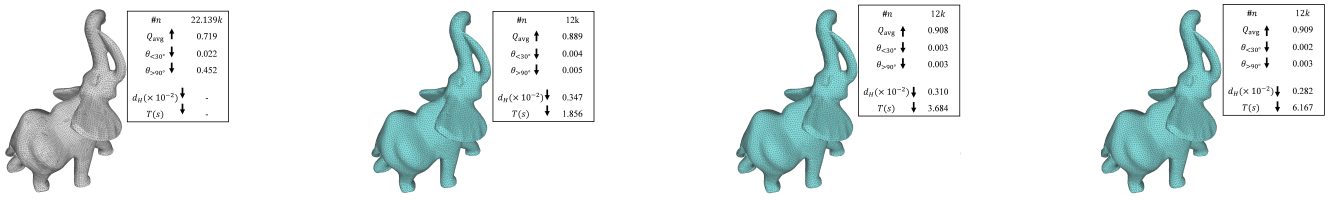


Figure 7. Results on seven organic models (Holes to Gargoyle). Columns: (Top) Input; (Middle) CVT-optimized with multi-step clipping; (Bottom) Output. Significant triangle reduction achieved while preserving features and quality, even for complex cases (metrics in Tab 1).

Table 1. *The quality and running time of various methods on different models correspond to Figure 7.*

Model	**In/Out**	**n**	$Q_{\min}\uparrow$	$Q_{\text{avg}}\uparrow$	$\Theta_{\min}\uparrow$	$\Theta_{\max}\downarrow$	$\Theta_{<30^\circ}\downarrow$	$\Theta_{>90^\circ}\downarrow$	$d_H(\times 10^{-2})\downarrow$	$T(s)\downarrow$	Q_{up} (%) $\uparrow$	$Q_{\text{up}}/T(s)\uparrow$
Holes	Input	24.503k	0.074	0.730	3.437	167.929	0.199	0.296	–	–	–	–
	Output	3k	0.728	0.924	38.869	88.738	0.000	0.000	0.296	6.500	26.575	4.088
Armchair	Input	39.524k	0.024	0.697	1.257	176.288	0.259	0.391	–	–	–	–
	Output	4k	0.718	0.922	38.128	89.537	0.000	0.000	0.480	6.772	32.281	4.767
Kitten	Input	55.099k	0.023	0.741	1.206	176.706	0.106	0.277	–	–	–	–
	Output	5k	0.674	0.921	39.255	94.888	0.000	0.000	0.255	9.810	24.291	2.476
Dolphin	Input	49.181k	0.245	0.781	13.949	147.587	0.064	0.255	–	–	–	
	Output	6k	0.676	0.924	37.923	93.667	0.000	0.000	0.260	7.239	18.310	2.529
Botijo	Input	10.858k	0.026	0.669	1.574	176.539	0.335	0.399	–	–	–	–
	Output	8k	0.338	0.901	15.832	129.728	0.001	0.005	0.452	10.060	34.679	3.447
Homer	Input	31.743k	0.009	0.712	0.510	178.835	0.267	0.309	–	–	–	–
	Output	9k	0.120	0.903	4.145	137.135	0.002	0.005	0.233	16.179	26.826	1.658
Gargolye	Input	100.001k	0.081	0.650	2.962	169.216	0.373	0.471	–	–	–	–
	Output	9k	0.251	0.881	9.200	125.742	0.008	0.023	0.586	19.834	35.538	1.792



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

Figure 8. (a) Input model and mesh quality. (b) Remeshed with max clipping times = 1; (c) Remeshed with max clipping times = 2; (d) Remeshed with max clipping times = 3. Increasing max clipping times from 1 to 2: computational time slightly increases while mesh quality significantly improves. Further increasing to 3: time substantially grows whereas quality improvement becomes marginal. Our algorithm selects 3 clippings as optimal balance.

- Hausdorff distance d_H $(\times 10^{-2})$ for maximum surface deviation.

To quantify efficiency, we propose two metrics:

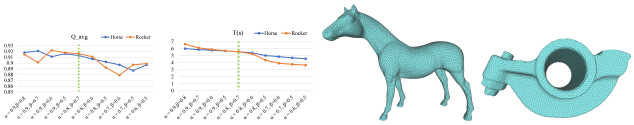
- Quality improvement rate:

$$Q_{\text{up}}\ (\%) = \left(\frac{Q_{\text{avg}}^{(\text{output})} - Q_{\text{avg}}^{(\text{input})}}{Q_{\text{avg}}^{(\text{input})}} \right) \times 100 \qquad (8)$$

- Quality improvement per unit time:

Table 2. *The quality and running time of various methods on different models correspond to Figure 10. Values in blue indicate the best results; values in cyan indicate the second best results.*

Model	Method	n	$Q_{\min}\uparrow$	$Q_{\text{avg}}\uparrow$	$\Theta_{\min}\uparrow$	$\Theta_{\max}\downarrow$	$\Theta_{<30^\circ}\downarrow$	$\Theta_{>90^\circ}\downarrow$	$d_H(\times 10^{-2})\downarrow$	$T(s)\downarrow$	Q_{up} (%) $\uparrow$	$Q_{\text{up}}/T(\% \cdot s^{-1})\uparrow$
Genus	Input	31.421k	0.053	0.771	3.429	127.991	0.078	0.289	–	–	–	–
	RVD	2.977k	0.550	0.851	30.184	109.487	**0.000**	0.065	1.251	73.484	10.376	0.141
	MPS	3.000k	0.472	0.800	27.332	119.027	0.006	0.158	1.008	**0.188**	3.761	**20.007**
	RVC	2.996k	**0.632**	0.913	**34.654**	**99.823**	**0.000**	0.063	1.459	3.69	18.418	4.991
	PowerRTF	3.000k	0.485	0.867	24.322	112.105	0.002	0.018	**0.604**	1.72	12.451	7.239
	Ours	3.000k	0.586	**0.916**	30.460	101.487	**0.000**	**0.001**	0.783	6.500	**18.807**	2.893
Bird	Input	2.431k	0.414	0.807	24.864	126.238	0.010	0.157	–	–	–	–
	RVD	3.043k	0.546	0.853	28.964	109.877	**0.000**	0.048	0.613	51.084	5.700	0.112
	MPS	3.003k	0.447	0.792	25.798	122.041	0.012	0.180	0.352	0.125	nan	nan
	RVC	2.423k	0.373	0.887	15.487	113.139	0.004	0.027	1.488	2.551	9.913	3.886
	PowerRTF	3.000k	0.499	0.865	23.968	110.545	0.003	0.023	0.336	1.646	7.187	4.369
	Ours	3.000k	**0.582**	**0.915**	**30.635**	**102.412**	**0.000**	**0.002**	**0.331**	1.169	**13.383**	**11.448**
Duck	Input	24.988k	0.030	0.756	1.583	175.713	0.140	0.259	–	–	–	–
	RVD	2.934k	0.530	0.849	28.884	111.857	**0.000**	0.063	0.693	69.274	12.302	0.178
	MPS	3.001k	0.473	0.800	27.705	118.789	0.004	0.165	0.795	**0.156**	5.820	**37.308**
	RVC	2.999k	0.583	0.908	**30.067**	105.584	0.001	0.009	0.755	2.084	20.106	9.648
	PowerRTF	3.000k	0.534	0.869	26.241	105.781	0.002	0.016	**0.432**	2.463	13.003	5.280
	Ours	3.000k	**0.633**	**0.917**	29.717	**96.923**	**0.000**	**0.001**	0.536	3.436	**21.296**	6.198
Fandisk	Input	7.223k	0.179	0.625	0.759	176.536	0.481	0.310	–	–	–	–
	RVD	3.156k	0.522	0.829	28.590	112.615	**0.000**	0.066	0.929	45.385	32.640	0.719
	MPS	3.006k	0.410	0.798	25.341	126.711	0.009	0.166	1.219	**0.125**	27.68	**221.44**
	RVC	2.993k	0.394	0.896	22.059	128.175	0.002	0.016	1.999	3.097	43.360	14.001
	PowerRTF	3.000k	0.538	0.864	25.226	110.61	0.001	0.022	**0.747**	2.315	38.240	16.518
	Ours	3.000k	**0.663**	**0.912**	**35.742**	**95.913**	**0.000**	**0.002**	0.824	2.672	**45.92**	17.186
David head	Input	108.333k	0.046	0.654	2.455	173.76	0.361	0.461	–	–	–	–
	RVD	6.114k	**0.518**	0.849	**25.369**	**112.322**	**0.000**	0.058	**0.602**	392.818	**29.817**	0.076
	MPS	–	–	–	–	–	–	–	–	–	–	–
	RVC	5.917k	0.336	0.890	13.456	116.225	0.002	0.024	2.114	7.647	36.086	4.719
	PowerRTF	6.000k	0.094	0.867	3.250	130.014	0.007	0.025	0.772	**5.316**	**32.569**	**6.127**
	Ours	6.000k	0.182	**0.896**	6.459	115.713	0.003	**0.014**	0.667	10.512	37.003	3.520
Elk	Input	10.716	0.019	0.821	0.863	176.633	0.051	0.101	–	–	–	–
	RVD	7.146k	**0.518**	0.849	25.369	**112.322**	**0.000**	0.058	0.567	226.232	**11.858**	0.052
	MPS	7.000k	0.416	0.795	25.363	125.963	0.008	0.178	0.637	**0.297**	nan	nan
	RVC	6.860k	0.022	0.895	0.746	174.673	0.013	0.027	2.998	24.922	9.013	0.362
	PowerRTF	7.000k	0.471	0.864	22.656	116.628	0.004	0.023	**0.516**	2.97	5.238	1.764
	Ours	7.000k	0.495	**0.916**	**27.483**	115.19	**0.000**	**0.002**	0.594	3.47	11.571	**3.335**
Lion	Input	40k	0.307	0.496	1.283	175.596	0.658	0.682	–	–	–	–
	RVD	20.266k	0.284	0.851	14.205	141.66	**0.006**	0.064	**0.559**	1348.96	71.573	0.053
	MPS	20.028k	**0.368**	0.790	**21.824**	**131.836**	0.020	0.181	0.632	**1.093**	59.274	**54.231**
	RVC	17.269k	0.103	0.858	5.295	166.344	0.015	0.056	5.114	6.265	72.984	11.649
	PowerRTF	20.000k	0.075	0.881	2.557	159.706	0.014	0.030	0.885	9.197	77.621	8.440
	Ours	20.000k	0.072	**0.891**	2.451	149.668	0.009	**0.023**	0.620	9.514	**79.637**	8.371



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

Figure 9. (a) Q_{avg} under different α-β combinations. (b) $T(s)$ under different α-β combinations. Experimental results of parameter combinations on different models: Horse (organic model) and Rocker (CAD model). (c) Output model for Horse with $\alpha = 0.8$, $\beta = 0.7$. (d) Output model for Rocker with $\alpha = 0.8$, $\beta = 0.7$. It can be observed that when $\alpha = 0.8$ and $\beta = 0.7$, the output models achieve relatively good quality with moderate time consumption.

$$Q_{\text{up}}/T\ (\% \cdot s^{-1}) = \frac{Q_{\text{up}}}{T} \tag{9}$$

where T denotes the computation time in seconds. These metrics collectively characterize the method's effectiveness and time-efficiency.

5.1. Surface Remeshing Results

Experimental validation on seven models (Holes to Gargoyle) confirms consistent performance. Visual results (Fig. 7) and quantitative metrics (Tab. 1) show simple models (Holes, Armchair, Kitten, Dolphin) achieve near-zero $\Theta_{<30^\circ}$ and $\Theta_{>90^\circ}$. Complex models maintain $\Theta_{<30^\circ} < 0.1\%$. Triangle quality exceeds $Q_{\text{avg}} > 0.9$ for all models except Gargoyle, which attains 35.54% improvement. Geometric fidelity demonstrates Hausdorff distance $d_H < 0.6\times 10^{-2}$ universally. Repeated clipping on high-curvature features increases time consumption for complex models,

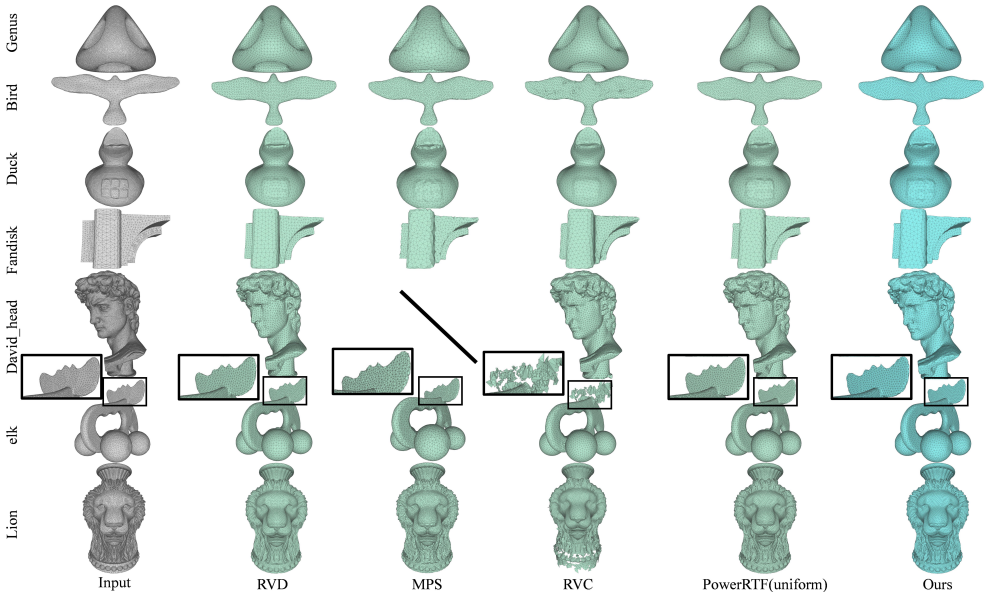


Figure 10. Our method demonstrates robust remeshing across models of varying complexity, where others fail (e.g., MPS on David_head). The result exhibits a closer approximation to the original geometry and higher mesh quality, as visible in the detailed elk antler structure.

reducing Q_{up}/T despite high quality gains. The method maintains effective geometric preservation, demonstrating robust handling of complex geometries with stable convergence.

5.2. Parameter Analysis

Maximum clipping times critically determine remeshing quality. Experiments reveal baseline quality at 1 clipping, modest improvement at 2 clippings, and further enhancement at 3 clippings (Fig.8). Single-clipping operation resembles PowerRTF's approximation [37], while higher clipping counts approach RVD's precision [34]. This adjustability enables efficiency-to-precision tradeoffs, reflecting framework flexibility.

Curvature estimation via adjacent face angles (cosines $\in [0, 1]$) governs multi-clip decisions. Thresholds $\alpha = 0.8$ and $\beta = 0.7$ balance quality and computation: smaller values reduce multi-clip cells lowering complexity but compromising quality; larger values increase multi-clip cells improving quality at higher computational cost. Optimal thresholds were experimentally determined (Fig. 9).

5.3. Comparison

Compared with recent CVT-based methods (Fig. 10 and Tab. 2), our approach achieves optimal Q_{avg} values, demonstrating significant quality improvement. It maintains optimal/suboptimal $\Theta_{<30^\circ}$ and $\Theta_{>90^\circ}$ values, confirming effective angle control. Superior d_{H} $(\times 10^{-2})$ results indicate precise geometric preservation. Robustness extends across diverse models from simple to complex structures. Compared to RVD [34] and PowerRTF [20] our adaptive clipping applies multiple clips in high curvature regions (sharp edges, textures) and fewer in low curvature areas. This yields faster processing for smooth surfaces and longer computation for rough surfaces, while both Q_{up} $(\%)$ and Q_{up}/T $(\% \cdot s^{-1})$ metrics confirm confirm stable efficiency in quality-time tradeoffs

6. Conclusion

We propose a CVT-based remeshing algorithm that dynamically balances computation and quality by adjusting clipping times based on facet curvature. Unlike precise intersection or plane approximation, it computes intersections between 3D CVT and original facets at sites. Experiments validate effectiveness on models

Limitations and future work Although our method performs well on various models, some limitations remain. First, it employs uniform remeshing, while practical models require denser sampling points in high-curvature areas (e.g., lion's mane) than flat regions (e.g., lion's base) to better fit details. A curvature-adaptive remeshing approach might yield improved results. Second, we do not preserve features, making it unsuitable for CAD models requiring strong feature retention. Finally, our results may depend on the original model quality. Next, we will incorporate curvature-based sampling for CVT site distribution and introduce feature preservation to better capture model details.

7. Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grants 62372152.

References

[1] Ahmed Abdelkader, Chandrajit L Bajaj, Mohamed S Ebeida, Ahmed H Mahmoud, Scott A Mitchell, John D Owens, and Ahmad A Rushdi. Vorocrust: Voronoi meshing without clipping. *ACM Transactions on Graphics (TOG)*, 39(3):1–16, 2020. 1, 2

[2] I. Babu?Ka and A. K. Aziz. On the angle condition in the finite element method. *Siam Journal on Numerical Analysis*, 13(2):214–226, 1976. 5

[3] Dobrina Boltcheva and Bruno Lévy. Surface reconstruction by computing restricted voronoi cells in parallel. *Computer-Aided Design*, 90:123–134, 2017. SI:SPM2017. 5

[4] Zhonggui Chen, Tieyi Zhang, Juan Cao, Yongjie Jessica Zhang, and Cheng Wang. Point cloud resampling using centroidal Voronoi tessellation methods. *Computer-Aided Design*, 102:12–21, 2018. 1, 2, 5

[5] Qiang Du, Vance Faber, and Max Gunzburger. Centroidal Voronoi tessellations: Applications and algorithms. *SIAM Review*, 41(4):637–676, 1999. 1

[6] Xingyi Du, Xiaohan Liu, Dongming Yan, Caigui Jiang, Juntao Ye, and Hui Zhang. Field-aligned isotropic surface remeshing. *Computer Graphics Forum*, 37(6):343–357, 2018. 2

[7] Jianwei Guo, Dongming Yan, Xiaohong Jia, and Xiaopeng Zhang. Efficient maximal Poisson-disk sampling and remeshing on surfaces. *Computers & Graphics*, 46:72–79, 2015. 5

[8] Jianwei Guo, Fan Ding, Xiaohong Jia, and Dongming Yan. Automatic and high-quality surface mesh generation for cad models. *Computer-Aided Design*, 109:49–59, 2019. 1

[9] Wenjuan Hou, Chen Zong, Pengfei Wang, Shiqing Xin, Shuangmin Chen, Guozhu Liu, Changhe Tu, and Wenping Wang. SDF-RVD: Restricted Voronoi diagram on signed distance field. *Computer-Aided Design*, 144:103166, 2022. 2

[10] Dawar Khan, Dongming Yan, Fan Ding, Yixin Zhuang, and Xiaopeng Zhang. Surface remeshing with robust user-guided segmentation. *Computational Visual Media*, 4(2):113–122, 2018. 1

[11] Dawar Khan, Alexander Plopski, Yuichiro Fujimoto, Masayuki Kanbara, Gul Jabeen, Yongjie Jessica Zhang, Xiaopeng Zhang, and Hirokazu Kato. Surface remeshing: A systematic literature review of methods and research directions. *IEEE Transactions on Visualization and Computer Graphics*, 28(3):1680–1713, 2020. 1, 2, 5

[12] Michal Křížek. On the maximum angle condition for linear tetrahedral elements. *SIAM J. Numer. Anal.*, 29(2):513–520, 1992. 5

[13] Bruno Levy and Nicolas Bonneel. Variational anisotropic surface meshing with voronoi parallel linear enumeration. *21st International Meshing Roundtable*, pages 349–366, 2013. 2

[14] Bruno Lévy and Alain Filbois. Geogram: a library for geometric algorithms. In *VII International Conference on Adaptive Modeling and Simulation (ADMOS 2015)*, page 45. CIMNE, 2015. 3, 5

[15] Bruno Lévy and Yang Liu. L_p centroidal Voronoi tessellation and its applications. *ACM Transactions on Graphics (TOG)*, 29(4):1–11, 2010. 1

[16] Hongbo Li, Haikuan Zhu, Sikai Zhong, Ningna Wang, Cheng Lin, Xiaohu Guo, Shiqing Xin, Wenping Wang, Jing Hua, and Zichun Zhong. Nasm: Neural anisotropic surface meshing. 2024. 2

[17] Yang Liu, Wenping Wang, Bruno Lévy, Feng Sun, Dongming Yan, Lin Lu, and Chenglei Yang. On centroidal Voronoi tessellation—energy smoothness and fast computation. *ACM Transactions on Graphics (TOG)*, 28(4):1–17, 2009. 1, 2, 3

[18] Yongjin Liu, Chunxu Xu, Ran Yi, Dian Fan, and Ying He. Manifold differential evolution (mde): a global optimization method for geodesic centroidal voronoi tessellations on meshes. 35(6), 2016. 2

[19] Yongjin Liu, Dian Fan, Chunxu Xu, and Ying He. Constructing intrinsic Delaunay triangulations from the dual of geodesic Voronoi diagrams. *ACM Transactions on Graphics (TOG)*, 36(2):1–15, 2017. 1

[20] Yong-Jin Liu, Dian Fan, Chunxu Xu, and Ying He. Constructing intrinsic delaunay triangulations from the dual of geodesic voronoi diagrams. *ACM Trans. Graph.*, 36(2), 2017. 2, 5, 8

[21] Esdras Medeiros and Marcelo Siqueira. Good random multitriangulation of surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 24(6):1983–1996, 2018. 1

[22] Wenlong Meng, Pengbo Bo, Xiaodong Zhang, Jixiang Hong, Shiqing Xin, and Changhe Tu. An efficient algorithm for approximate voronoi diagram construction on triangulated surfaces. *Computational Visual Media*, 9:443 – 459, 2023. 2

[23] Patrick Mullen, Pooran Memari, Fernando de Goes, and Mathieu Desbrun. HOT: Hodge-optimized triangulations. In *ACM SIGGRAPH 2011 papers*, pages 1–12, 2011. 2

[24] Nicolas Ray, Dmitry Sokolov, Sylvain Lefebvre, and Bruno Lévy. Meshless voronoi on the gpu. *ACM Trans. Graph.*, 37 (6), 2018. 3, 4

[25] Guodong Rong, Miao Jin, Liang Shuai, and Xiaohu Guo. Centroidal voronoi tessellation in universal covering space of manifold surfaces. *Computer Aided Geometric Design*, 28(8):475–496, 2011. Solid and Physical Modeling 2010. 1

[26] Alfred H. Thiessen. Precipitation averages for large areas. *Monthly Weather Review*, 39:1082–1089, 1911. 3

[27] Pengfei Wang, Shiqing Xin, Changhe Tu, Dongming Yan, Yuanfeng Zhou, and Caiming Zhang. Robustly computing restricted Voronoi diagrams (RVD) on thin-plate models. *Computer Aided Geometric Design*, 79:101848, 2020. 2

[28] Xiaoning Wang, Xiang Ying, Yongjin Liu, Shiqing Xin, Wenping Wang, Xianfeng Gu, Wolfgang Mueller-Wittig, and Ying He. Intrinsic computation of centroidal voronoi tessellation (cvt) on meshes. *Computer-Aided Design*, 58:51–61, 2015. Solid and Physical Modeling 2014. 1

[29] Yong Ge Xinqiao Duan, Lin Li and Bo Liu. Exact voronoi diagram for topographic spatial analysis. *GIScience & Remote Sensing*, 60(1):2171703, 2023. 2

[30] Minfeng Xu, Shiqing Xin, and Changhe Tu. An efficient surface remeshing algorithm based on centroidal power diagram. In *Proceedings of the 2019 7th International Confer-*

ence on Information Technology: IoT and Smart City, pages 536–542, 2019. 2

[31] Qun-Ce Xu, Dongming Yan, Wenbin Li, and Yongliang Yang. Anisotropic surface remeshing without obtuse angles. *Computer Graphics Forum*, 38(7):755–763, 2019. 1

[32] Rui Xu, Longdu Liu, Ningna Wang, Shuangmin Chen, Shiqing Xin, Xiaohu Guo, Zichun Zhong, Taku Komura, Wenping Wang, and Changhe Tu. Cwf: Consolidating weak features in high-quality mesh simplification. *ACM Trans. Graph.*, 43(4), 2024. 2

[33] Dongming Yan and Peter Wonka. Non-obtuse remeshing with centroidal Voronoi tessellation. *IEEE Transactions on Visualization and Computer Graphics*, 22(9):2136–2144, 2015. 2

[34] Dongming Yan, Bruno Lévy, Yang Liu, Feng Sun, and Wenping Wang. Isotropic remeshing with fast and exact computation of restricted Voronoi diagram. *Computer Graphics Forum*, 28(5):1445–1454, 2009. 2, 5, 8

[35] Dongming Yan, Wenping Wang, Bruno Lévy, and Yang Liu. Efficient computation of clipped voronoi diagram for mesh generation. *Computer-Aided Design*, 45(4):843–852, 2013. Geometric Modeling and Processing 2010. 2

[36] Dongming Yan, Guanbo Bao, Xiaopeng Zhang, and Peter Wonka. Low-resolution remeshing using the localized restricted Voronoi diagram. *IEEE Transactions on Visualization and Computer Graphics*, 20(10):1418–1427, 2014. 2

[37] Yuyou Yao, Jingjing Liu, Yue Fei, Wenming Wu, Gaofeng Zhang, Dongming Yan, and Liping Zheng. Powerrtf: Power diagram based restricted tangent face for surface remeshing. *Computer Graphics Forum*, 42(5):e14897, 2023. 2, 8

[38] Yuyou Yao, Jingjing Liu, Wenming Wu, Gaofeng Zhang, Benzhu Xu, and Liping Zheng. Accelerating surface remeshing through GPU-based computation of the restricted tangent face. *Computer Aided Geometric Design*, 104:102216, 2023. 1, 2, 5

[39] Zipeng Ye, Ran Yi, Minjing Yu, Yongjin Liu, and Ying He. Geodesic centroidal Voronoi tessellations: Theories, algorithms and applications. *arXiv preprint arXiv:1907.00523*, 2019. 2

[40] Henrik Zimmer, Marcel Campen, Ralf Herkrath, and Leif Kobbelt. Variational tangent plane intersection for planar polygonal meshing. In *Advances in Architectural Geometry 2012*, pages 319–332. Springer, 2013. 2