

CWF: Consolidating Weak Features in High-quality Mesh Simplification

RUI XU*, Shandong University, China

LONGDU LIU*, Shandong University, China

NINGNA WANG, The University of Texas at Dallas, USA

SHUANGMIN CHEN, Qingdao University of Science and Technology, China

SHIQING XIN[†], Shandong University, China

XIAOHU GUO, The University of Texas at Dallas, USA

ZICHUN ZHONG, Wayne State University, USA

TAKU KOMURA, The University of Hong Kong, China

WENPING WANG, Texas A&M University, USA

CHANGHE TU[†], Shandong University, China



Fig. 1. In this paper, we propose a functional for aligning both strong and weak feature lines during mesh simplification while maintaining high-quality triangulations. The left side shows organic models, and the right side displays CAD models. It's important to mention that the original versions of the organic models are heavily smoothed.

In mesh simplification, common requirements like accuracy, triangle quality, and feature alignment are often considered as a trade-off. Existing algorithms concentrate on just one or a few specific aspects of these requirements. For example, the well-known Quadric Error Metrics (QEM) approach [Garland and Heckbert 1997] prioritizes accuracy and can preserve strong feature lines/points as well, but falls short in ensuring high triangle quality and may degrade weak features that are not as distinctive as strong ones. In this paper, we propose a smooth functional that simultaneously considers all of these requirements. The functional comprises a normal anisotropy term and a Centroidal Voronoi Tessellation (CVT) [Du et al. 1999] energy term, with the variables being a set of movable points lying on the surface. The former inherits the spirit of QEM but operates in a continuous setting, while the latter encourages even point distribution, allowing various surface metrics. We further introduce a decaying weight to automatically balance the two terms. We selected 100 CAD models from the ABC dataset [Koch et al. 2019], along with 21 organic models, to compare the existing mesh simplification

algorithms with ours. Experimental results reveal an important observation: the introduction of a decaying weight effectively reduces the conflict between the two terms and enables the alignment of weak features. This distinctive feature sets our approach apart from most existing mesh simplification methods and demonstrates significant potential in shape understanding. Please refer to the teaser figure for illustration.

CCS Concepts: • Computing methodologies → Mesh geometry models.

Additional Key Words and Phrases: mesh simplification, weak feature, centroidal Voronoi tessellation, feature consolidation

1 INTRODUCTION

Mesh simplification aims to reduce complexity to balance the need of visual fidelity with computational efficiency. It has been proved useful in various applications. First, a simplified mesh can be considered as a more compact representation [Dou et al. 2020; Li and Nan 2021] w.r.t the original mesh, which can facilitate real-time transmission over networks [Cabiddu and Attene 2015; Ko and Choy 2002], and rendering in mobile applications or AR/VR environments [Bahirat et al. 2018; Fu et al. 2022]. Second, a simplified mesh serves as a proxy of the original shape, requiring fewer computational resources and enhancing speed in interactive applications [Zhang et al. 2023] and simulations.

* Equal contribution.

† Corresponding authors.

Authors' addresses: Rui Xu, Shandong University, China, xrvtd@163.com; Longdu Liu, Shandong University, China, liulongdu@163.com; Ningna Wang, The University of Texas at Dallas, USA, ningna.wang@utdallas.edu; Shuangmin Chen, Qingdao University of Science and Technology, China, csmqq@163.com; Shiqing Xin, Shandong University, China, xinshiqing@sdu.edu.cn; Xiaohu Guo, The University of Texas at Dallas, USA, xguo@utdallas.edu; Zichun Zhong, Wayne State University, USA, zichunzhong@wayne.edu; Taku Komura, The University of Hong Kong, China, taku@cs.hku.hk; Wenping Wang, Texas A&M University, USA, wenping@tamu.edu; Changhe Tu, Shandong University, China, chtu@sdu.edu.cn.

Common requirements in the field of mesh simplification include accuracy [Lescoat et al. 2020; Liu et al. 2023], triangle quality [Abdelkader et al. 2020; Du et al. 1999; Wang et al. 2018], and feature alignment [Chen et al. 2023; Garland and Heckbert 1997; Lévy and Liu 2010]. Accuracy is typically measured based on the difference between the simplified and original versions. Good triangle quality not only leads to more accurate results in physics simulations but also increases the stability of numerical computations [Dou et al. 2022; Wang et al. 2024b, 2022a, 2024a]. Feature alignment helps provide crucial visual clues for effective shape recognition and involves aligning the mesh edges with strong and weak feature lines. However, weak features, less distinctive than strong ones, are challenging to preserve or even consolidate during mesh simplification.

In past research, requirements such as accuracy, triangle quality, and feature alignment were often considered trade-offs. Existing algorithms, including Quadric Error Metrics (QEM) [Garland and Heckbert 1997] and Centroidal Voronoi Tessellation (CVT) [Du et al. 1999], focus on a few specific aspects but struggle to achieve a good balance. For instance, QEM, the most popular approach for mesh simplification, involves a sequence of basic operations such as edge contraction. Despite its advantages in accuracy and alignment with strong features, it suffers from the degradation of weak features. In contrast, CVT focuses on maintaining an even point distribution by optimizing a set of movable points, yet it falls short in aligning with any feature lines, whether strong or weak.

In this paper, we propose a smooth functional that concurrently addresses the aforementioned requirements. This functional includes a normal anisotropy term and a CVT energy term, applied to a set of movable points on the surface. The normal anisotropy term, drawing inspiration from QEM, operates in a continuous setting, while the CVT term promotes an even distribution of points, accommodating various surface metrics. We also introduce a decaying weight to automatically balance these two terms. Optimization typically concludes within tens of iterations, with each iteration decomposing the surface into Voronoi cells. Considering that existing tools for computing Restricted Voronoi Diagrams (RVDs) cannot ensure the “one site, one region” property on thin-plate models, we generate an inward counterpart for each movable point and develop a simple yet effective technique to facilitate the computation of RVD on thin-plate models.

We tested our functional on 100 CAD models from the ABC dataset [Koch et al. 2019] and 21 organic models. The experimental results not only confirm that the normal anisotropy term effectively preserves accuracy and aligns strong features, while the CVT term promotes even point distribution, but they also demonstrate that the introduction of a decaying weight effectively reduces the conflict between the two terms, achieving a good balance among multiple requirements. This advantageous property makes it useful for the lightweight representation of CAD models. Additionally, it can be observed that our algorithm is particularly effective in consolidating weak features of organic shapes during mesh simplification.

Our contributions are three-fold:

- (1) We propose a functional that unifies the requirements of accuracy, triangle quality, and feature alignment, effectively achieving a good balance among these key aspects.

- (2) We introduce a decaying weight to gradually reduce the impact of the CVT energy, thereby naturally coordinating the two terms.
- (3) We develop a simple yet effective technique to facilitate the computation of the Restricted Voronoi Diagram (RVD) on thin-plate models.

2 RELATED WORK

Triangle mesh surfaces have become increasingly popular and widely used. Given that dense triangulations can lead to high computational costs, mesh simplification is essential for balancing computational efficiency with an acceptable level of visual or geometric fidelity across various applications. Key considerations typically include accuracy, triangle quality, and feature alignment. In the following, we will review related works on mesh simplification.

2.1 CVT and Its Variants

By minimizing the Centroidal Voronoi Tessellation (CVT) energy [Du et al. 1999], one can achieve an even distribution of movable points on a surface. There is a substantial body of literature on using CVT to generate high-quality mesh surfaces [Valette and Chassery 2004; Valette et al. 2008]. A notable implementation combines Restricted Voronoi Diagrams (RVD) and L-BFGS optimization to accomplish high-quality meshing, as proposed in [Yan et al. 2009]. Additionally, numerous variants of CVT [Du et al. 2018; Edwards et al. 2013; Sun et al. 2011; Wang et al. 2018; Yan and Wonka 2015] have been developed to enhance triangulations in specific scenarios. However, most of these variants struggle with feature alignment.

In the work of LpCVT [Lévy and Liu 2010], normal anisotropy is incorporated into the objective function to facilitate feature alignment during remeshing. However, the balancing weight between the two terms requires careful case-by-case adjustment according to the specific input shape. Typically, the normal anisotropy term approaches zero for CAD models, but remains significant for organic models. This variability leads to LpCVT’s inconsistency in preserving both strong and weak features.

2.2 Accuracy Guided Mesh Simplification

Local decimation schemes are widely favored in mesh simplification for their ease of implementation and near-linear scalability [Hoppe et al. 1993]. To minimize accuracy loss, Garland and Heckbert [1997] introduced a fast, high-quality edge collapse-based simplification algorithm using the Quadric Error Metric (QEM). This algorithm is proficient at predicting target positions during edge contraction into a vertex, utilizing the normal vectors of the incident triangles. Our tests confirm that QEM offers an accuracy advantage and can align with strong features. Due to its popularity, various QEM variants have been developed. For example, Ozaki et al. [2015] suggested dividing the original surface into multiple patches and performing QEM-guided simplification on each to enhance parallelism and ensure algorithmic feasibility. Wei and Lou [2010] expanded the classic QEM into a constrained high-dimensional space with specific considerations for singular scenarios. Additional improvements to QEM are discussed in [Chen et al. 2023; Liu et al. 2023; Panchal and Jayaswal 2022].

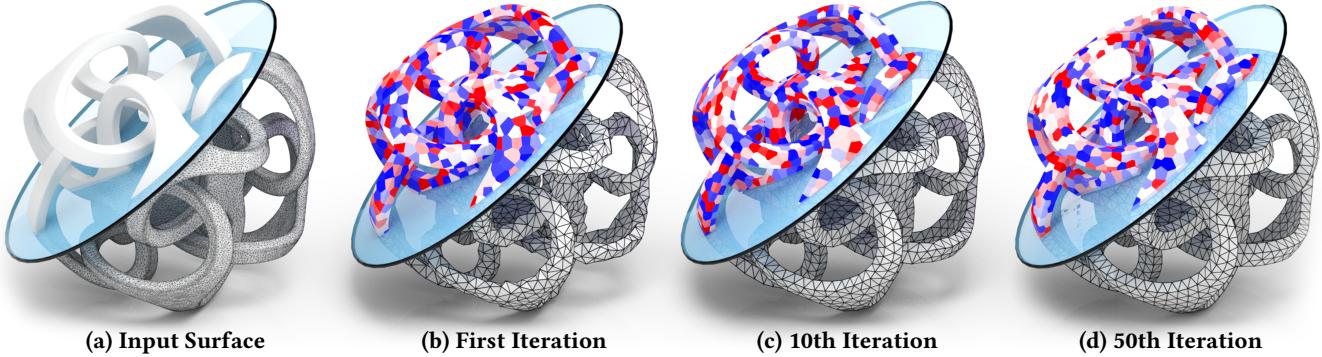


Fig. 2. We present a functional that integrates the demands of accuracy, triangle quality, and feature alignment. The impact of the CVT energy gradually diminishes, thanks to the decaying weight, achieving a harmonious balance between the two terms. It is noted that a Restricted Voronoi Diagram (RVD) must be computed during each iteration. In this example, a total of 50 iterations activated our termination condition.

The primary goal of preserving accuracy is to minimize the Hausdorff distance between the original and simplified surfaces, ensuring geometric fidelity [Ma et al. 2012]. Alternatives include defining an isotropic density function or an anisotropic metric to guide mesh simplification. For instance, Du and Wang [2005] integrated the traditional Lloyd-CVT method with anisotropic Riemannian metrics, naming it ACVT. Zhong et al. [2013] achieved anisotropic meshing by mapping the anisotropic space to a higher-dimensional isotropic space. Xu et al. [2019] worked directly with anisotropic meshes generated from existing remeshing algorithms, focusing on eliminating obtuse angles. Moreover, curvature has been utilized to generate adaptive and isotropic mesh results [Lv et al. 2022; Su et al. 2019]. However, these algorithms primarily improve accuracy rather than focusing on feature alignment.

2.3 Feature Preserving Mesh Simplification

Feature preservation is a critical aspect of mesh simplification. Several approaches [Xie et al. 2012; Yan et al. 2014; Yan and Wonka 2013; Zhong et al. 2013] begin by pre-detecting features and subsequently conducting re-meshing while retaining these features. For instance, VoroCrust [Abdelkader et al. 2020] suggests explicitly placing points symmetrically along pre-detected feature lines. While pre-detecting features seems reasonable, it becomes impractical for organic models where identifying weak features is challenging compared to strong features.

A more promising approach involves ensuring that movable points naturally align with underlying features. Most existing algorithms leverage the inherent property of QEM, which naturally captures strong features. For instance, Valette et al. [2008] used QEM to guide the generation of Voronoi vertices, ensuring feature alignment. Similarly, Chiang et al. [2011] proposed mesh quality improvement through quadratic error-based mesh relaxation. Additionally, Gao et al. [2013] extended optimal Delaunay triangulation (ODT) to surface meshes by solving a quadratic optimization problem. Although recent works [Panchal and Jayaswal 2022; Xu et al. 2022; Zhao et al. 2023] have utilized QEM’s ability to preserve sharp features, it must be noted that weak features remain a challenge for QEM.

Various approaches utilize an anisotropy field or curvature information to optimize point placement. Lévy and Liu [2010] employ

an anisotropy field to guide point placement without the need for tagging. In contrast, Jakob et al. [2015] introduce a field-aligned meshing method for isotropic triangular/quad-dominant remeshing, incorporating local smoothing and control over sharp features. Additionally, Cai et al. [2016] propose a novel PCA-based energy for face-based clustering, resulting in an anisotropic mesh surface. However, these methods face challenges in achieving high-quality isotropic triangulations and struggle to preserve weak features effectively.

3 FORMULATION

In this paper, we introduce a unified functional designed to meet three essential requirements: accuracy, triangle quality, and feature alignment. Utilizing a set of movable points $\{\mathbf{x}_i\}_{i=1}^N$ as inputs, our approach alternates between decomposing the surface into distinct regions based on proximity and optimizing point distribution. This effectively decouples point placement from their connections, a characteristic reminiscent of the principles underlying CVT.

3.1 Objective Function

Our objective function is constructed as follows:

$$E(\{\mathbf{x}_i\}_{i=1}^N) = \lambda_{NA} E_{NA} + \lambda_{CVT} E_{CVT}, \quad (1)$$

where E_{NA} and E_{CVT} respectively denote the normal anisotropic term and the CVT term. The normal anisotropic term [Lévy and Liu 2010] is defined as:

$$E_{NA} = \sum_i \int_{\Omega_i} \left((\mathbf{x} - \mathbf{x}_i)^T \mathbf{n}_x \right)^2 ds = \sum_i \int_{\Omega_i} (\mathbf{x} - \mathbf{x}_i)^T \mathbf{n}_x \mathbf{n}_x^T (\mathbf{x} - \mathbf{x}_i) ds, \quad (2)$$

where $\{\Omega_i\}_{i=1}^N$ defines the surface decomposition, commonly computed as a Restricted Voronoi Diagram (RVD). The CVT term is expressed as:

$$E_{CVT} = \sum_{i=1}^N \int_{\Omega_i} (\mathbf{x} - \mathbf{x}_i)^T M_{CVT} (\mathbf{x} - \mathbf{x}_i) ds. \quad (3)$$

Typical choices for M_{CVT} include:

- (1) An identity matrix, defining the Euclidean-line distance.
- (2) An isotropic matrix, encoding a density function on the surface.

(3) An anisotropic matrix, utilizing the anisotropic tensor to capture and represent directional variations.

In summary, our objective is represented as:

$$E(\{\mathbf{x}_i\}_{i=1}^N) = \sum_{i=1}^N \int_{\Omega_i} (\mathbf{x} - \mathbf{x}_i)^T M(\mathbf{x} - \mathbf{x}_i) ds, \quad (4)$$

where the kernel matrix M is defined as:

$$M = \lambda_{NA} M_{NA} + \lambda_{CVT} M_{CVT}. \quad (5)$$

3.2 Links to Existing Approaches

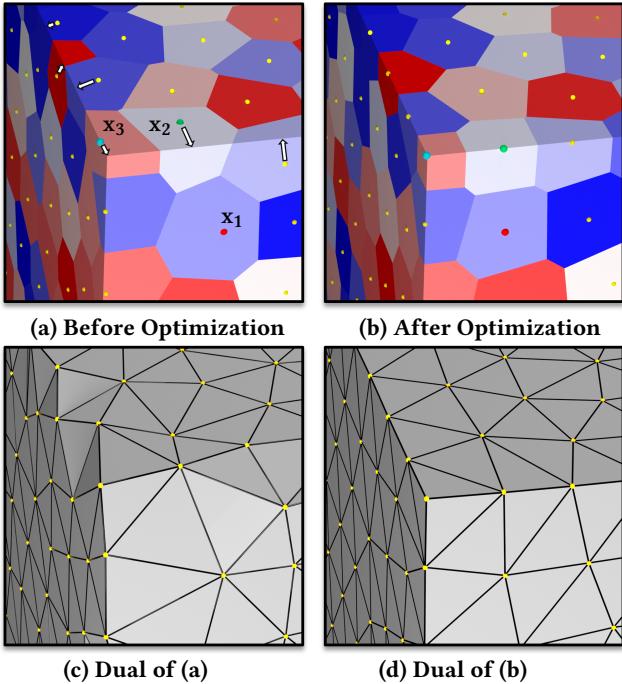


Fig. 3. The RVD after optimizing the positions by minimizing our objective function, points near the features are naturally drawn toward the nearby points/lines. In this figure, x_1 (red) controls a planar region and remains static, x_2 (green) dominates an area across the feature line and is shifted to the feature line, and x_3 (blue) governs a corner point and is relocated to the corner.

Normal Anisotropy. As mentioned in [Lévy and Liu 2010], the normal anisotropy matrix $\mathbf{n}_x \mathbf{n}_x^T$ quantifies the extent to which the vector $\mathbf{x} - \mathbf{x}_i$ is orthogonal to the normal vector at \mathbf{x} , effectively positioning \mathbf{x}_i with respect to all the points in its dominant region Ω_i .

It is evident that the normal anisotropy term inherits the essence of QEM [Garland and Heckbert 1997]. The key difference is that while QEM assesses the quadratic error in relation to triangles incident to a mesh edge and its two endpoints, our approach evaluates it within the region of Ω_i . Fig. 3 (a) illustrates the initial positions and the surface decomposition. The dual of the surface decomposition, shown in Fig. 3 (c), generates a triangle mesh, but it does not align well with feature points or lines. After optimizing the positions by minimizing our objective function, as demonstrated in Fig. 3 (b, d), points near features are naturally pulled towards nearby

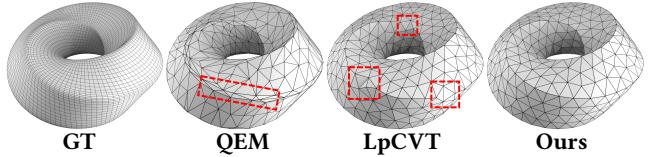


Fig. 4. By setting the target number of vertices to 400, we compare the ability to consolidate weak features among QEM, LpCVT, and our method. It's noted that the original Mobius-ring model has a smoothly transitioning shape with 7000 vertices.

points or lines. To be more specific, x_1 (red) controls a planar region and remains static, x_2 (green) dominates an area across the feature line and shifts towards it, and x_3 (blue) governs a corner point and relocates to that corner.

It is worth noting that the normal anisotropy term can also achieve its minimum when Voronoi cell boundaries coincide with the model's feature lines. However, such Voronoi diagrams must have degree-4 vertices, which are not stable configurations. Given a random initial point placement, it is unlikely to terminate at such an unstable configuration, as explained in Fig. 2 of [Liu et al. 2009]. We provide an example of this initialization in Section 5.

Our Considerations. As discussed in Section 1, the essential requirements for mesh simplification include accuracy, triangle quality, and feature alignment. It is clear that the normal anisotropy term primarily preserves accuracy and enforces feature alignment, while the CVT term tends to enhance triangle quality. Achieving a harmonious balance between these aspects is not straightforward. Our analysis identifies two main challenges.

First, the magnitudes of the normal anisotropy term and the CVT term can differ significantly. For example, the normal anisotropy term can be reduced to zero in a standard cube model, but the CVT term cannot be completely eliminated. Conversely, in the case of an organic model, such as the Bunny model, the normal anisotropy term cannot be reduced to zero. This discrepancy underscores the need for a more effective mechanism to balance these terms.

Second, while strong features are marked by drastic changes in normals within a localized area, weak features involve subtle shape variations over larger scales. For instance, in the Mobius-ring model visualized in Fig. 4, the original model exhibits a smoothly transitioning shape with dense triangulation. Our tests indicate that traditional methods like QEM and LpCVT struggle to capture these weak feature lines, unlike our algorithm, which demonstrates a superior capability in this regard.

4 ALGORITHM

Minimizing Eq. (1) resembles solving the CVT problem, which alternately performs surface decomposition and point re-location. The algorithm paradigm decouples point placement from their connections. Similar to CVT, we also utilize the L-BFGS solver to minimize Eq. (1) until some convergence criterion is satisfied. In the following, we elaborate on (1) the computation of the objective function and gradients, (2) the utilization of decaying weight, and (3) a more effective technique for computing RVDs on thin-plate models.

4.1 Optimization

As mentioned above, the entire energy includes two parts denoted by E_{NA} and E_{CVT} , respectively. The computation of the objective function and gradients relies on surface decomposition and numerical quadrature.

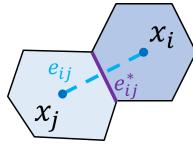
Surface Decomposition and Quadrature. Given a set of points $\{\mathbf{x}_i\}_{i=1}^N$, the RVD typically decomposes the surface into N regions, with each point \mathbf{x}_i being associated with one region. However, the RVD may encounter challenges with thin-plate models, as discussed in Section 4.3. Recall that the input is represented by a triangular surface S . Each region, Ω_i , consists of a collection of planar convex polygons. These polygons may include triangles from S or sub-triangles that are sliced by the walls of the Voronoi diagram of the points $\{\mathbf{x}_i\}_{i=1}^N$. To compute the contribution of a polygon with k vertices, the polygon is partitioned into $k - 2$ triangles. The Albrecht-Collatz quadrature is then employed to evaluate the integral over each of these triangles. Notably, the Albrecht-Collatz quadrature utilizes six points per triangle: three points are positioned at the midpoints of the edges, and the remaining three points are situated inside the triangle.

Gradients. Like that achieved in CVT [Liu et al. 2009],

$$\nabla_{\mathbf{x}_i} E_{CVT} = \int_{\Omega_i} -2M_{CVT}(\mathbf{x} - \mathbf{x}_i) ds. \quad (6)$$

However, $\nabla_{\mathbf{x}_i} E_{NA}$ is not as straightforward as $\nabla_{\mathbf{x}_i} E_{CVT}$. According to Reynolds transport theorem, we have

$$\begin{aligned} \nabla_{\mathbf{x}_i} E_{NA} &= \int_{\Omega_i} -2((\mathbf{x} - \mathbf{x}_i) \cdot \mathbf{n}_x) \mathbf{n}_x ds \\ &+ \sum_{j \in \text{Nei}(i)} \int_{e_{ij}^*} ((\mathbf{x} - \mathbf{x}_i) \cdot \mathbf{n}_x)^2 (\nabla_{\mathbf{x}_i} \mathbf{x} \cdot \frac{\mathbf{e}_{ij}}{\|\mathbf{e}_{ij}\|}) dx \\ &- \sum_{j \in \text{Nei}(i)} \int_{e_{ij}^*} ((\mathbf{x} - \mathbf{x}_j) \cdot \mathbf{n}_x)^2 (\nabla_{\mathbf{x}_i} \mathbf{x} \cdot \frac{\mathbf{e}_{ij}}{\|\mathbf{e}_{ij}\|}) dx, \end{aligned} \quad (7)$$



where $\{\mathbf{x}_j\}_{j \in \text{Nei}(i)}$ is the point set neighboring to \mathbf{x}_i , $\mathbf{e}_{ij} = \mathbf{x}_j - \mathbf{x}_i$, and \mathbf{e}_{ij}^* is the common boundary between \mathbf{x}_i 's cell and \mathbf{x}_j 's cell (see the purple line in the wrapped figure). According to [De Goes et al. 2012], we further have:

$$\nabla_{\mathbf{x}_i} \mathbf{x} \cdot \frac{\mathbf{e}_{ij}}{\|\mathbf{e}_{ij}\|} = \frac{\mathbf{e}_{ij}}{2\|\mathbf{e}_{ij}\|}. \quad (8)$$

However, in our experiments, we observe that the sum of the second and third terms in Eq. (7) is negligible compared to the first term. Therefore, we only consider the first term to approximate the gradients:

$$\nabla_{\mathbf{x}_i} E_{NA} \approx \int_{\Omega_i} -2((\mathbf{x} - \mathbf{x}_i) \cdot \mathbf{n}_x) \mathbf{n}_x ds. \quad (9)$$

4.2 Decaying Weight

Suppose that the total surface area is normalized to 1.0. E_{CVT} does not vary significantly for either CAD models or organic models. However, E_{NA} can be reduced to nearly 0 for most CAD models, making the values of E_{NA} and E_{CVT} not of the same order of magnitude (to be more precise, the changes in the two terms occur at the

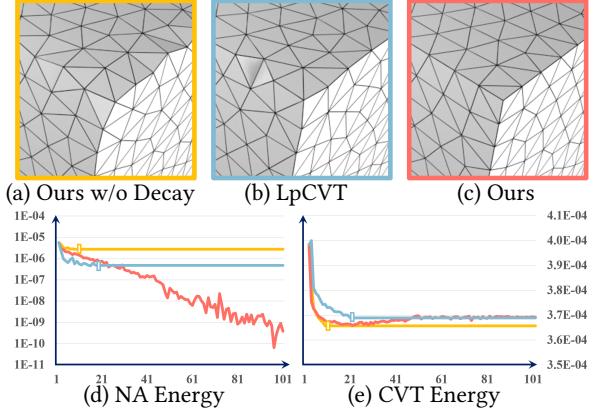


Fig. 5. Resulting triangle mesh surfaces. For all three approaches, we maintain a constant total of 100 iterations and monitor changes in the normal anisotropy term (d) and the CVT energy term (e). The plots for these approaches are colored yellow, blue, and red, respectively. Please note that the vertical axis of the normal anisotropy term is displayed on a $\log 10$ scale, whereas the vertical axis for the CVT energy term uses a linear scale.

same rate in a balanced state). For example, with the Cube model (see Fig. 3), E_{NA} can be reduced to 0, which makes E_{CVT} the dominant term. In this situation, E_{CVT} prevents E_{NA} from vanishing. Therefore, we need to prioritize minimizing E_{CVT} at the start of optimization and gradually reduce its influence. To achieve this, we keep λ_{NA} unchanged throughout the optimization while allowing λ_{CVT} to decay according to:

$$\lambda_{CVT}^{(i)} = \lambda_{CVT}^{(i-1)} \times \tau, \quad (10)$$

where $\tau = 0.95$ by default. This approach prioritizes even point placement in the initial iterations and shifts focus to the feature alignment requirement subsequently. Refer to Fig. 5 for plots showing the changes in these terms with and without weight decay.

The optimization generally requires tens of iterations for CAD models based on our tests. Even with up to 100 iterations, the resulting triangulation on a CAD model can still satisfy high triangle quality and feature alignment simultaneously. However, we observe that for an organic model, the triangle quality may diminish significantly as the number of iterations increases. To prevent the deterioration of triangle quality, we need to stop the optimization if the CVT energy term E_{CVT} experiences an increase, described by:

$$E_{CVT}^{(i)} \geq \mu \min_{j=0}^{i-1} E_{CVT}^{(j)}, \quad (11)$$

where μ is set at 1.05.

To summarize, our optimization terminates if one of the following conditions is met:

- The gradient norm is less than 10^{-8} .
- $E_{CVT}^{(i)} \geq 1.05 \times \min_{j=0}^{i-1} E_{CVT}^{(j)}$.

Using the model in Fig. 2 as an example, we set the target number of movable points to 3000 and execute our algorithm. The optimization process involves 50 iterations, as the second termination condition is triggered.

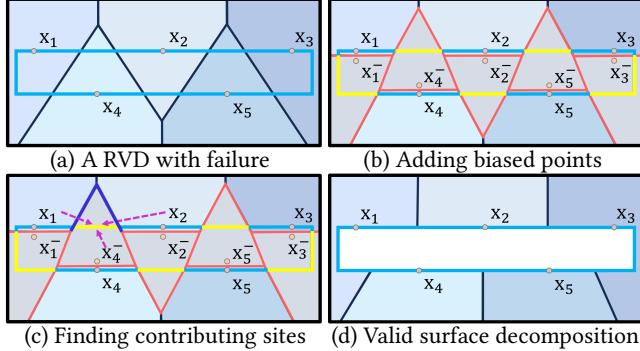


Fig. 6. A simple yet effective technique for fixing RVDs by introducing an additional set of biased points. (a) RVDs may fail on a thin-plate model with a set of sparse sites. (b) Rather than fix the problematic RVD, we bias each point at a negligible distance along the inward normal direction. (c) We need to find the contributing sites for each yellow region that is dominated by a biased site. (d) After the yellow regions are partitioned and assigned to the contributing sites, we get a valid surface decomposition.

4.3 RVDs on Thin-plate Models

When the input model is a thin-plate and the number of movable points is insufficient, RVDs may encounter an issue where a single site dominates two or more regions. Some research works [Wang et al. 2022b] focus on identifying the ownerless regions and further analyzing which sites can partition these regions, which requires tedious computation. In this paper, we present a simple yet effective technique to address this issue.

We use a 2D example shown in Fig. 6 to illustrate our idea, where the blue rectangle is very thin compared to the number of sites. If one directly computes the RVD, x_2 , x_4 , and x_5 dominate two regions, violating the rule of “one site, one region”; See Fig. 6(a). Rather than correcting the problematic RVD, we bias each point at a negligible distance (e.g., 0.01) along the inward normal direction, yielding a point set $\{x_i^-\}_{i=1}^N$ of equal size. After that, it requires two stages to accomplish the surface decomposition.

Stage 1. We compute the Voronoi diagram of $\{x_i\}_{i=1}^N \cup \{x_i^-\}_{i=1}^N$. The entire surface is thus decomposed into N or more regions. The regions dominated by x_i^- , $i = 1, 2, \dots, N$, colored in yellow in Fig. 6(b), need to be partitioned and re-assigned.

Stage 2. For each yellow region, we identify the “contributing” sites. Taking the bottom-left yellow region as an example, it is determined by two bisectors: one given by x_1 and x_1^- , and the other given by x_1^- and x_4 . We define the contributing sites of this region as x_1 , x_1^- , and x_4 . By disregarding the biased site x_1^- , we partition this region using x_1 and x_4 . (Note that x_4 does not gain more dominating area in this operation.) Similarly, for the top-left yellow region, the contributing sites are x_1 , x_2 , and x_4^- . By disregarding the biased site x_4^- , this region can be re-partitioned and assigned to x_1 and x_2 . See Fig. 6(c,d).

It’s worth noting that the regions dominated by biased sites are distinct from the ownerless regions in [Wang et al. 2022b]. For instance, the bottom-left yellow region in Fig. 6 (b) is not an ownerless region but necessitates additional handling in our algorithm. After

identifying the yellow regions, we can process them separately and in parallel. See the fixed surface decomposition in Fig. 6 (d).

5 EXPERIMENTAL RESULTS

All of our experiments were conducted on a computer equipped with an AMD Ryzen 9 5950X CPU and 64 GB of memory. The ABC dataset [Koch et al. 2019] includes models with self-intersections, non-manifold vertices/edges, and open boundaries. Therefore, we randomly selected 100 watertight manifold meshes, similar to those in RFEPS [Xu et al. 2022]. See Fig. 7 for a gallery of the selected models. Additionally, we selected 21 organic models with weak features.

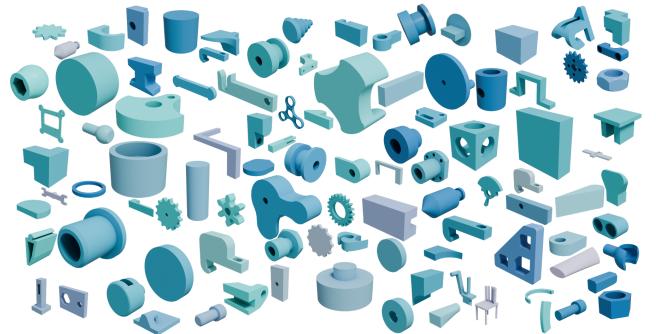


Fig. 7. 100 models selected from the ABC dataset.

For each CAD model, we set the target number of vertices to be 500, 1000, and 2000, respectively. For organic models, we sampled various target numbers of vertices, ranging from 100 to 8000.

Evaluation Metrics. To measure the difference between the simplified surface and the original version, we use four indicators: *Chamfer Distance* (CD), *F-score* (F1), *Normal Consistency* (NC), and *Hausdorff Distance* (HD).

To facilitate the definition of metrics, we define M_g and M_p as the ground truth model and the simplified model, respectively. Let $\mathcal{X} \subset M_g$ and $\mathcal{Y} \subset M_p$ denote the randomly sampled points from each model, respectively. We sample 100K points for all evaluations. The nearest point can be found as follows:

$$\begin{aligned} \mathcal{P}_{g2p}(\mathbf{x}) &= \arg \min_{\mathbf{y} \in \mathcal{Y}} \|\mathbf{x} - \mathbf{y}\|_2, \\ \mathcal{P}_{p2g}(\mathbf{y}) &= \arg \min_{\mathbf{x} \in \mathcal{X}} \|\mathbf{x} - \mathbf{y}\|_2. \end{aligned} \quad (12)$$

Chamfer Distance (CD). To measure the average nearest squared distance between \mathcal{Y} and \mathcal{X} , we introduce the chamfer distance metric as follows:

$$\begin{aligned} \text{CD}(\mathcal{X}, \mathcal{Y}) &= \frac{1}{2N_g} \sum_{i=1}^{N_g} \|\mathbf{x}_i - \mathcal{P}_{g2p}(\mathbf{x}_i)\|_2 + \\ &\quad \frac{1}{2N_p} \sum_{i=1}^{N_p} \|\mathbf{y}_i - \mathcal{P}_{p2g}(\mathbf{y}_i)\|_2. \end{aligned} \quad (13)$$

F-score (F1). F-score is the harmonic mean of precision and recall. Precision is the ratio of true positives to the sum of true positives

and false positives, while recall is the ratio of true positives to the sum of true positives and false negatives.

$$F1(\mathcal{X}, \mathcal{Y}) = \frac{2 * \text{precision} * \text{Recall}}{\text{precision} + \text{Recall}}. \quad (14)$$

Normal Consistency (NC). To measure the alignment of normals, we define a normal consistency score by calculating the average of the absolute dot products between pairs of normals.

$$\begin{aligned} \text{NC}(\mathcal{X}, \mathcal{Y}) = & \frac{1}{2N_g} \sum_{i=1}^{N_g} |\mathbf{n}(\mathbf{x}_i) \cdot \mathbf{n}(\mathcal{P}_{g2p}(\mathbf{x}_i))| + \\ & \frac{1}{2N_p} \sum_{i=1}^{N_p} |\mathbf{n}(\mathbf{y}_i) \cdot \mathbf{n}(\mathcal{P}_{p2g}(\mathbf{y}_i))| \end{aligned} \quad (15)$$

Hausdorff Distance (HD). The Hausdorff distance is a distance metric that can measure the distance between two subsets within the same metric space. $\overrightarrow{\text{HD}}(\mathcal{X}, \mathcal{Y})$ defines the one-sided Hausdorff distance from \mathcal{X} to \mathcal{Y} , and $\text{HD}(\mathcal{X}, \mathcal{Y})$ is the two-sided version.

$$\begin{aligned} \overrightarrow{\text{HD}}(\mathcal{X}, \mathcal{Y}) &= \max_{\mathbf{x} \in \mathcal{X}} \min_{\mathbf{y} \in \mathcal{Y}} \|\mathbf{x} - \mathbf{y}\|_2, \\ \text{HD}(\mathcal{X}, \mathcal{Y}) &= \max \left(\overrightarrow{\text{HD}}(\mathcal{X}, \mathcal{Y}), \overrightarrow{\text{HD}}(\mathcal{Y}, \mathcal{X}) \right). \end{aligned} \quad (16)$$

To evaluate triangle quality, we use the *TriangleQ* indicator [Frey and Borouchaki 1999]. A value closer to 1.0 indicates that the triangle is nearly equilateral.

$$\text{TriangleQ}(t) = \frac{6}{\sqrt{3}} \frac{S_t}{p_t h_t} \quad (17)$$

where S_t , p_t , and h_t represent the area, half-perimeter, and the longest edge length of the triangle t , respectively. Furthermore, we use *OpenB* to represent the number of open mesh edges in the simplification result, and *NMV* to denote the number of non-manifold vertices. For CAD models, we also employ *Edge Chamfer Distance* (ECD) and *Edge F-score* (EF1), proposed by NMC [Chen and Zhang 2021], to measure the extent to which the feature lines are preserved.

Parameters. In our experiments, we use the same parameter settings for all the models used in this paper, whether CAD models or organic models. We set $\lambda_{\text{NA}} = \lambda_{\text{CVT}} = 1.0$ at the beginning of our optimization. λ_{NA} remains unchanged throughout the optimization, but λ_{CVT} undergoes a decaying process at the rate of $\tau = 0.95$. The parameter μ , which controls the extent of the rise in the CVT energy, is set to 1.05. We employ the L-BFGS solver to solve the optimization, and the termination condition is set by referring to the gradient norm, with a tolerance of $1e^{-8}$. We show an ablation study of these parameters in Section 5.5.

5.1 Comparison Methods

We compared our method with nine state-of-the-art (SOTA) methods: CVT [Du et al. 1999], LpCVT [Lévy and Liu 2010], QEM [Garland and Heckbert 1997], SMS [Lescoat et al. 2020], IEM [Liu et al. 2023], LPM [Chen et al. 2023], PQ [Trettner and Kobbelt 2020], MD [Kobbelt et al. 1998] and ERB [Hu et al. 2016]. In the following, we will briefly introduce each method and its parameter settings.

Table 1. Quantitative comparison on 100 CAD models, taken from the ABC dataset [Koch et al. 2019]. Each CAD model has strong features. The **best** scores are emphasized in bold with underlining, while the **second best** scores are highlighted only in bold.

	#V	CVT	LpCVT	QEM	SMS	IEM	LPM	PQT	PQP	MD	ERB	Ours
CD ($\times 10^4$) ↓	500	0.349	0.148	0.119	0.335	2.876	0.126	0.573	0.099	16.650	0.098	0.115
	1000	0.174	0.098	0.108	0.146	0.881	0.127	0.262	0.085	16.339	0.098	0.093
	2000	0.110	0.080	0.096	0.123	0.267	0.116	0.155	0.078	13.816	0.098	0.076
F1 ↑	500	0.743	0.893	0.933	0.804	0.784	0.908	0.605	0.908	0.675	0.920	0.929
	1000	0.823	0.923	0.940	0.878	0.863	0.907	0.750	0.930	0.695	0.920	0.938
	2000	0.884	0.938	0.943	0.90	0.914	0.923	0.842	0.943	0.738	0.920	0.944
NC ↑	500	0.947	0.978	0.985	0.982	0.928	0.982	0.941	0.985	0.841	0.986	0.987
	1000	0.964	0.986	0.986	0.987	0.967	0.988	0.958	0.988	0.847	0.986	0.989
	2000	0.975	0.989	0.984	0.989	0.982	0.989	0.969	0.989	0.867	0.986	0.990
ECD ($\times 10^2$) ↓	500	4.247	0.120	0.105	3.716	1.577	0.839	11.340	2.762	10.250	0.137	0.101
	1000	2.600	0.080	0.088	0.539	1.360	0.431	11.598	0.198	10.664	0.137	0.079
	2000	2.560	0.061	0.045	0.152	0.765	0.448	11.115	0.135	9.001	0.137	0.055
EF1 ↑	500	0.020	0.461	0.573	0.208	0.383	0.392	0.004	0.563	0.161	0.560	0.566
	1000	0.056	0.553	0.607	0.387	0.444	0.468	0.006	0.586	0.186	0.560	0.601
	2000	0.108	0.593	0.622	0.487	0.510	0.520	0.011	0.596	0.230	0.560	0.615
<i>TriangleQ</i> ↑	500	0.879	0.840	0.525	0.767	0.483	0.261	0.761	0.579	0.407	0.766	0.857
	1000	0.892	0.859	0.573	0.766	0.538	0.274	0.754	0.602	0.468	0.766	0.883
	2000	0.892	0.872	0.641	0.769	0.626	0.305	0.740	0.632	0.534	0.766	0.893
HD ($\times 10^3$) ↓	500	5.542	4.461	0.243	2.064	3.279	1.103	0.543	0.036	2.261	0.083	0.201
	1000	2.487	3.706	0.042	0.838	1.397	0.952	0.312	0.021	2.160	0.083	0.062
	2000	1.296	1.923	0.034	0.442	0.436	0.790	0.184	0.011	1.811	0.083	0.020
<i>OpenB</i> ↓	500	0	0	346	247	285	24	4	5	683	166	0
	1000	0	0	509	384	467	43	2	9	1213	166	0
	2000	0	0	660	576	638	38	2	10	2198	166	0
<i>NMV</i> ↓	500	0	0	312	69	720	0	2	1	238	243	0
	1000	0	0	661	242	1348	0	2	1	321	243	0
	2000	0	0	1035	696	2342	0	2	2	440	243	0

Table 2. Quantitative comparison on 21 organic models with weak features.

	CVT	LpCVT	QEM	SMS	IEM	LPM	PQT	PQP	MD	ERB	Ours
CD ($\times 10^4$) ↓	0.201	0.106	0.063	0.301	0.268	0.093	0.332	0.105	0.198	0.069	0.090
	0.776	0.876	0.948	0.748	0.850	0.894	0.754	0.934	0.901	0.942	0.902
	0.829	0.575	0.110	0.259	0.710	0.540	0.344	0.098	0.170	0.103	0.147
F1 ↑	0.962	0.973	0.935	0.822	0.973	0.976	0.966	0.960	0.962	0.972	0.976
	0.892	0.847	0.531	0.611	0.684	0.478	0.674	0.518	0.486	0.781	0.861
	0	0	18	1	478	3	2	69	38	0	0
<i>OpenB</i> ↓	0	0	2	3	3	0	4	4	2	1	0
	0	0	2	3	3	0	4	4	2	1	0
	0	0	2	3	3	0	4	4	2	1	0

CVT [Du et al. 1999] optimizes site placement from an energy perspective, achieving high-quality triangulations. LpCVT [Lévy and Liu 2010] extends CVT to better preserve mesh features. In the default setting, the coefficient of normal anisotropy is set to 4. QEM [Garland and Heckbert 1997] simplifies input meshes by minimizing the quadratic error metric. QEM supports the user-specified number of faces. SMS [Lescoat et al. 2020] is a spectrum-based meshing method that takes the number of preserved eigenvalues as input, which we set to 50 after experimentation. IEM [Liu et al. 2023] extends QEM to preserve input model features from an intrinsic perspective. LPM [Chen et al. 2023] extracts a low-resolution isosurface with features and progressively guides the simplification of the original surface using QEM to align with the input. LPM requires the resolution parameter to extract the isosurface, set to 100 in our experiments. PQ [Trettner and Kobbelt 2020] revolutionizes error quadric minimization by embedding it in a probabilistic framework, enabling robust and efficient solutions that are up to 50 times faster than traditional SVD methods and enhance mesh processing tasks with greater uniformity and noise resilience. In the implementation of PQ [Trettner and Kobbelt 2020], two distinct configurations are offered: ‘prob_triangle’ and ‘prob_plane’, which we denote as PQT and PQP, respectively, to differentiate their applications in probabilistic quadric error minimization. MD [Kobbelt et al. 1998] establishes a generalized framework for mesh reduction, similar to greedy heuristic optimization algorithms, providing a

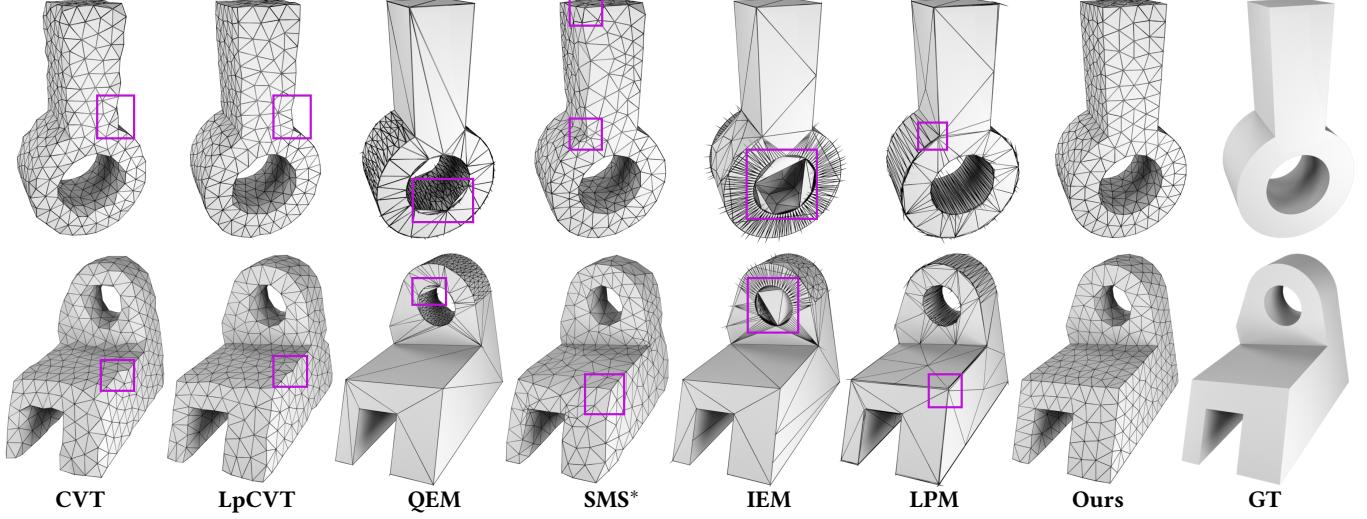


Fig. 8. Comparison with state-of-the-art methods on two CAD inputs. Our method excels in both accuracy and manifoldness. The target number of vertices for both inputs is set at 500. Note that SMS deteriorates when the input surface contains open boundaries, and we show their results on manifold inputs.

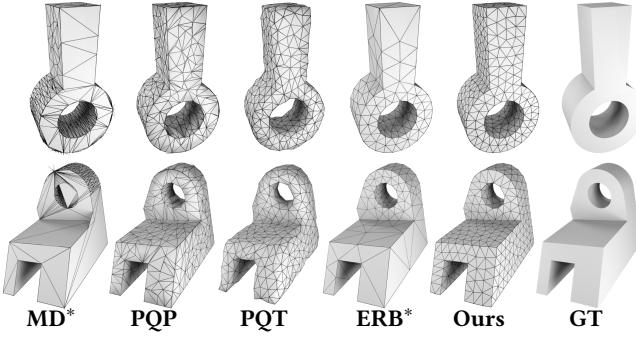


Fig. 9. Comparison with state-of-the-art methods on two CAD inputs. The target number of vertices for both inputs is set at 500. Note that MD and ERB deteriorate when the input surface contains open boundaries, and we show their results on manifold inputs.

versatile template that adapts to the specific needs of various target applications. ERB [Hu et al. 2016] introduces a novel surface remeshing algorithm that adeptly balances geometric fidelity, minimal complexity, and quality by simultaneously optimizing for an exact approximation error bound, minimal interior angles, and vertex count, resulting in superior meshes for geometry processing applications. It is important to note that ERB [Hu et al. 2016] does not support the specification of target points, which sets it apart from other methods.

5.2 Comparisons on CAD Models

We present the statistics of quantitative comparisons for the 100 CAD models in Table 1. The qualitative comparisons are illustrated in Fig. 8 and Fig. 9. It is evident that the CVT method excels in triangulation quality but falls short in preserving features. The LpCVT outperforms CVT in terms of the ability to preserve sharp features. In comparison with CVT and LpCVT, our method consistently demonstrates lower CD/HD scores and higher F1/NC scores,

regardless of the number of input sites (500, 1000, and 2000), which validates the advantage of our method in preserving sharp features.

As the number of points increases, QEM excels in accuracy preservation, as indicated by the ECD metric in Table 1. However, the *TriangleQ* indicator reveals that QEM cannot ensure high triangulation quality, with a significant number of obtuse triangles. SMS, IEM, LPM, PQP and ERB also demonstrate an accuracy advantage in terms of the CD, F1 and NC scores but may suffer from various artifacts, such as open mesh edges and non-manifold vertices. In comparison with these methods, our approach simultaneously addresses the requirements of accuracy, triangle quality, and feature alignment.

Note that PQT (referred to as ‘prob_triangle’ in PQ [Trettner and Kobbelt 2020]) demonstrates significant enhancement in triangulation quality over PQP (referred to as ‘prob_plane’ in PQ [Trettner and Kobbelt 2020]), albeit at the expense of accuracy, particularly for sharp feature indicators like ECD and EF1. ERB [Hu et al. 2016] excels across various metrics; however, its inability to specify point count often results in a higher average number of points used, approximately 874 in our analysis. Additionally, ERB is prone to issues with open boundaries and non-manifolds.

5.3 Comparisons on Organic Models

We also test our method on 21 organic models with weak features and show the quantitative statistics in Table 2. Similar to CAD model comparisons discussed in Section 5.2, CVT and LpCVT achieve excellent triangulation quality but exhibit lower accuracy. QEM, SMS, IEM, LPM, PQP, MD and ERB are better than CVT and LpCVT in terms of simplification accuracy, by the cost of diminishing triangle quality. It can be seen from Table 2 that our approach consistently achieves optimal or near-optimal scores, whether in accuracy or triangulation quality.

Due to the nature of organic models, some inherent weak features are not as distinctive as strong features. These weak features are

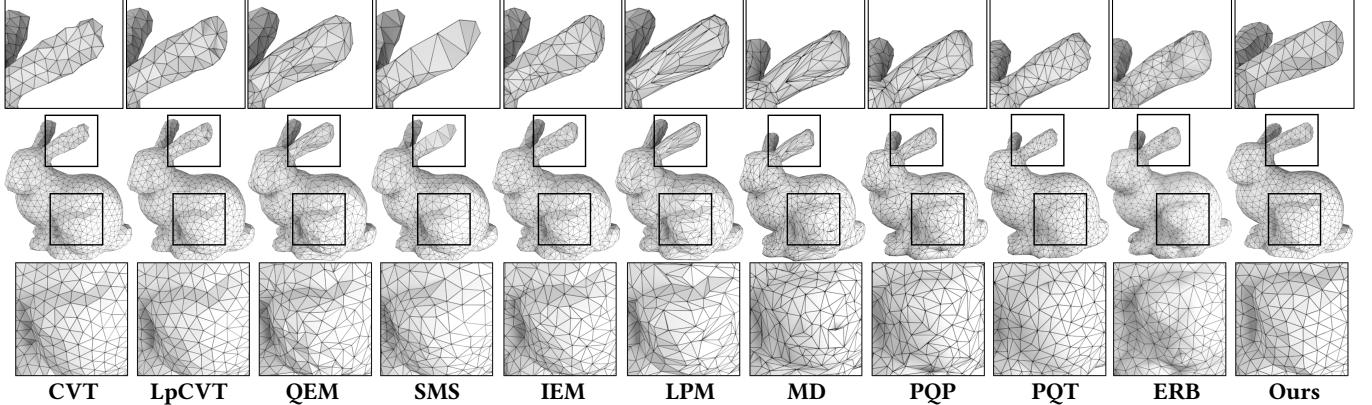


Fig. 10. Comparison with state-of-the-art methods on the bunny model using 1000 sample points, demonstrating that our method can more effectively consolidate weak features.

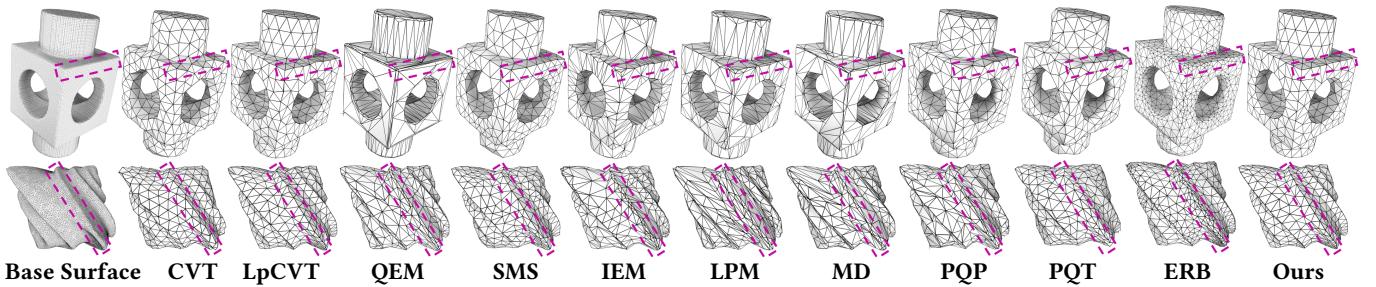


Fig. 11. Comparison with state-of-the-art methods on two smoothed inputs. The target number of points is set at 500. Our method more effectively consolidates weak features, whereas the QEM family may result in a chamfer along the feature line.

prone to being erased in the simplification result. In Fig. 10 and Fig. 11, it is evident that our method can consolidate weak features while maintaining high-quality triangulation, like the ears, legs, and neck of the *bunny*. With this advantageous property, our algorithm has the potential to convey visual shape clues through a simplified representation, distinguishing itself from other mesh simplification algorithms.

Additionally, VSR [Zhao et al. 2023] can be employed for mesh simplification by transforming the input mesh into a set of points. VSR alternates between point clustering and relocation. As the point relocation operation follows the spirit of QEM, the resulting point distribution may not be even when the number of points is limited. In Fig. 12, with the number of points set to 500 (the maximum supported by VSR), VSR produces a simplified mesh with poor triangulation. Furthermore, VSR involves a mixed-integer programming step, and practical success in finding a valid solution is not guaranteed. Consequently, numerous open mesh edges exist in this example due to the failure to satisfy manifold-edge constraints.

5.4 Run-time Performance

We present the run-time performance statistics in Table 3. The tests were conducted on the block model and bunny model, each with varying resolutions ranging from 0.5K points to 10K points. The total running time primarily includes the construction of RVDs and the optimization, with the computation of RVDs being the most time-consuming operation. Additionally, it can be observed

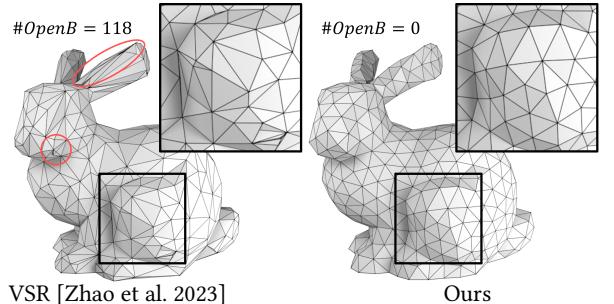


Fig. 12. Comparison with VSR [Zhao et al. 2023]. Due to VSR is a QEM-based method that involves a mixed-integer programming step and practical success in finding a valid solution is not guaranteed. Numerous open mesh edges exist in this example due to the failure to satisfy manifold-edge constraints.

that the optimization typically requires 40 to 50 iterations. The practical timing cost is approximately $O(k(m+n))$, where m is the complexity of the base mesh, n is the target number of vertices, and k is the number of optimization iterations.

5.5 Ablation Study

In the following, we present an ablation study regarding the weighting coefficients τ , λ_{NA} , λ_{CVT} and the termination tolerance μ .

Table 3. Running time (in seconds) with respect to the number of sampling points (#V). The input surface mesh of these two models contains both 40K faces. Note that CVT, LpCVT, and our method are based on RVD computation, so we show the running time as L-BFGS time / RVD time.

	#V	CVT	LpCVT	QEM	SMS	IEM	LPM	Ours	PQT	PQB	MD	ERB
block	0.5K	3.566/5.015	3.839/5.887	0.750	33.397	4.798	112.789	4.642/7.772	9.246	9.112	0.872	1692.910
	1K	3.720/5.812	6.520/8.511	0.759	33.818	4.593	112.163	3.821/7.473	9.644	9.784	0.755	1692.910
	3K	2.980/5.564	3.467/6.209	0.698	32.482	4.311	118.774	6.568/9.344	9.233	9.739	0.759	1692.910
	5K	3.834/7.053	9.722/15.047	0.582	33.172	4.158	118.615	6.415/11.013	9.720	9.845	0.744	1692.910
	10K	7.108/11.872	6.831/10.730	0.594	28.692	3.245	134.092	11.67/14.423	10.379	10.760	0.770	1692.910
bunny	0.5K	2.417/3.681	2.347/3.618	0.346	18.339	2.897	111.043	2.963/7.477	5.762	6.334	0.536	1542.080
	1K	3.622/5.722	3.79/5.203	0.476	17.729	2.929	113.652	5.085/7.301	5.936	5.821	0.580	1542.080
	3K	6.839/12.335	2.836/5.672	0.392	17.127	2.664	110.438	6.896/12.300	5.489	5.918	0.508	1542.080
	5K	6.516/13.238	2.569/5.724	0.449	15.615	2.350	114.986	7.587/13.183	6.203	6.523	0.541	1542.080
	10K	5.324/16.906	6.829/14.073	0.377	13.176	2.275	143.400	10.588/19.440	6.839	6.958	0.489	1542.080

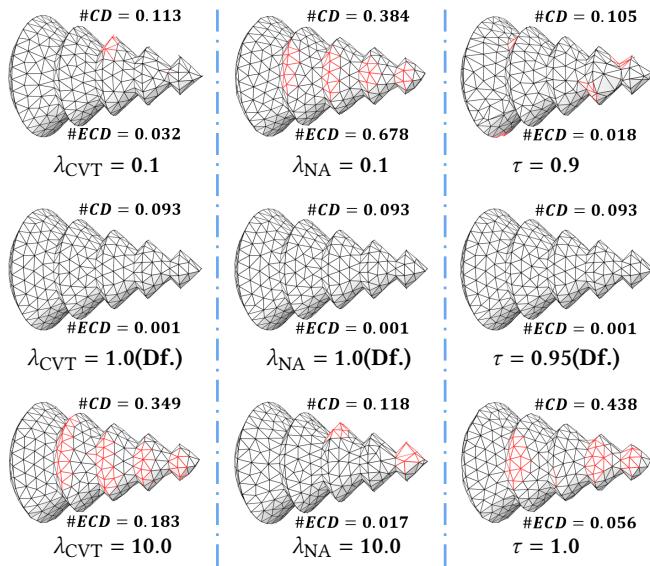


Fig. 13. Ablation study of the weighting coefficients τ , λ_{NA} and λ_{CVT} , where bad edges are colored in red. We select $\tau = 0.95$, $\lambda_{NA} = 1.0$, $\lambda_{CVT} = 1.0$ as the favorite combination, which is used in all of our experiments.

Decaying Rate. The decaying weight τ determines the rate of diminishing the CVT energy during the iteration process. When τ is set to 1.0 (see the bottom-right part of Fig. 13), the optimization process excessively prioritizes the CVT term, leading to the degradation of some weak feature edges. If τ is slightly smaller than 1.0, the contribution of the normal anisotropy term is increasingly emphasized with the growing number of iterations, while the CVT energy term is progressively suppressed. For example, by setting $\tau = 0.95$, the weighting influence of the CVT energy reduces from 1.0 to about 0.07 after 50 iterations. It can be imagined that if τ is too small (top right in Fig. 13), the point placement fails to achieve a good distribution since the CVT energy cannot sufficiently play its role before its influence disappears. And when multiple feature lines are in close proximity, the algorithm necessitates uniformity to achieve an ideal outcome. In the absence of uniformity, the control

region of a single Voronoi cell could encompass several feature lines, leading to the misalignment of features. We show the numerical curves of the two energy terms E_{CVT} and E_{NA} under different decay rates in Fig. 14.

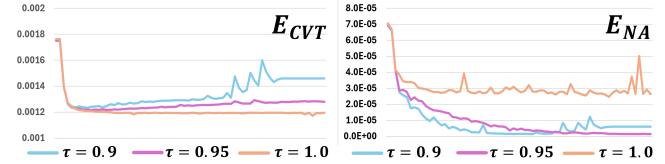


Fig. 14. Plot curves about how the two energy terms E_{CVT} and E_{NA} decrease under various decay rates on the model depicted in Fig. 13.

Parameter λ_{NA} . The parameter λ_{NA} in the main paper defines the influence of the normal anisotropy term during optimization. As we increase λ_{NA} from 0.1 to 10, the influence of the normal anisotropy term increases, and the meshing results resemble QEM. When λ_{NA} is set as large as 10.0, the triangle quality degrades. Moreover, while accuracy appears to improve, this is not the case when the points are insufficient. See the red edges in the bottom-middle part of Fig. 13.

Parameter λ_{CVT} . As Fig. 13 illustrates, with the increasing value of λ_{CVT} from 0.1 to 1.0, the contribution of the CVT energy term is enlarged, and the energy function gradually approaches the standard CVT. This makes the results resemble CVT, leading to an overall improvement in the triangulation quality of the model. However, it is observed that in the red region, the feature edges are damaged due to an excessive pursuit of triangulation quality.

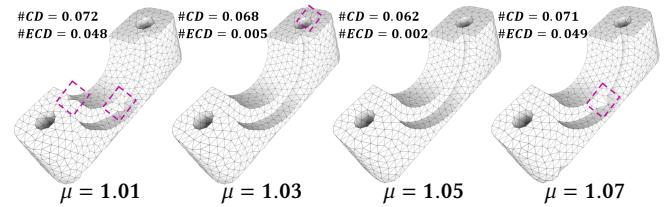


Fig. 15. Ablation study on the termination tolerance μ . We use $\mu = 1.05$ in all experiments.

Parameter μ . We present the ablation study on the termination tolerance μ in Fig. 15. It can be observed that a smaller μ , such as 1.01 or 1.03, requires fewer iterations to terminate, but the feature alignment ability is not fully realized. Conversely, a larger μ value, such as 1.07, requires hundreds of iterations to reach termination. Furthermore, the triangle quality may deteriorate at termination. This is why we use $\mu = 1.05$ in all experiments.

5.6 Non-Euclidean Metric

It is worth noting that we use Euclidean metric-based RVD in our experiments, and we can also support non-Euclidean metrics.

Anisotropic Metric. As Eq. (5) demonstrates, our formulation supports non-Euclidean metrics. For instance, one can utilize curvature to define an anisotropic field on the surface, establishing curvature-aware point-to-point distances. An example is illustrated in Fig. 16, where the surface decomposition is computed based on plane cutting [Sun et al. 2011]. The clipping technique, based on our tests, can only handle convex shapes (e.g., an ellipsoid). To the best of our knowledge, there is currently no mature solver available for the anisotropic Restricted Voronoi Diagrams.

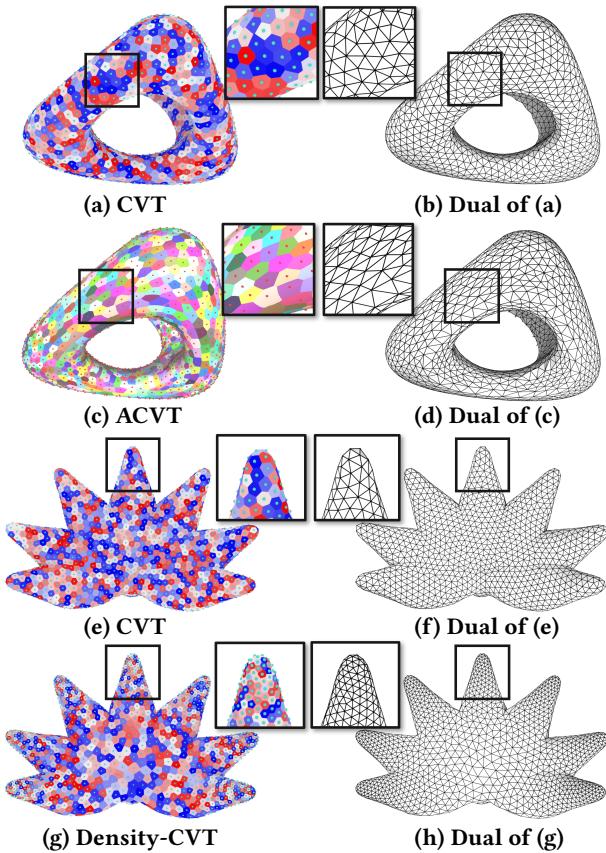


Fig. 16. Examples of non-Euclidean distance, (a-d) is the anisotropic metrics and RVDs, (e-h) is the density-based metrics and RVDs.

Density Function. It's natural that our algorithm also supports density-adaptive remeshing. The typical density field is derived from the local feature size (LFS), adapting the triangle sizes to the curvature. As depicted in Fig. 16, the utilization of density enables the concentration of mesh elements around features.

5.7 Robustness and Scalability

Noise. To test our noise-resistant ability, we add 0.25% and 0.50% Gaussian noise to the surface vertices of the *curve-ball* model, as shown in Fig. 17. As the noise level increases, the triangle quality of the mesh is diminished, and the surface becomes very bumpy. It can be observed from Fig. 17 that our method exhibits strong resistance to noise, consistently producing robust outputs.

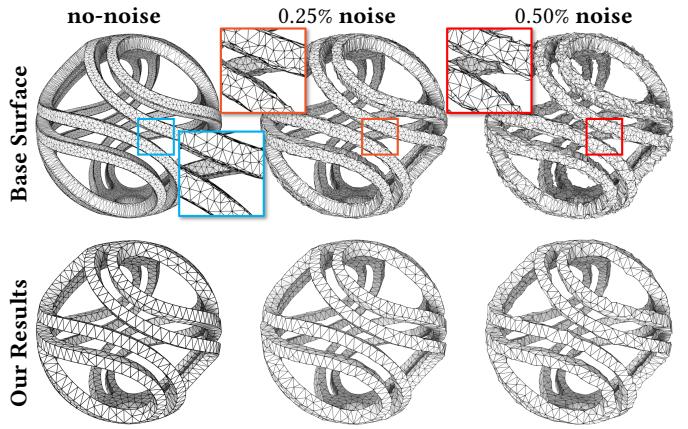


Fig. 17. Ablation study of our noise-resistant ability. We add 0.25% and 0.50% Gaussian noise to the surface vertices of the model. More noises may results in bad surface connectives such as self-intersections. Our method shows strong resistance to noise to some extent.

Poorly Triangulated Inputs. Our algorithm optimizes a set of movable points on the surface. It repeatedly decomposes the surface into regions during optimization. Therefore, it depends only on the geometry and works well for poorly triangulated inputs. Fig. 18 validate the robustness of our algorithm. We conducted tests on the *artwork* model, which has poor mesh quality on its side.

Initialization. The “normal anisotropy” term can also reach its minimum when the Voronoi cell edges align with the feature lines of the model, which is an unstable state, as mentioned in Section 3.2. We construct such an unstable example in the first column of Fig. 19. Our method can still optimize it into a stable point placement.

Additionally, we explore two additional point placement initialization strategies. The first assumes the points are distributed on a spherical surface, and the second assumes the points gather around a single point. In the second case, it can be seen that the points encounter certain resistance when crossing a feature line. Therefore, we recommend the Poisson-disk sampling strategy for initialization in all our experiments.

More Tests. Generally, our method takes a closed surface as the input. However, it must be noted that our method can also be applied

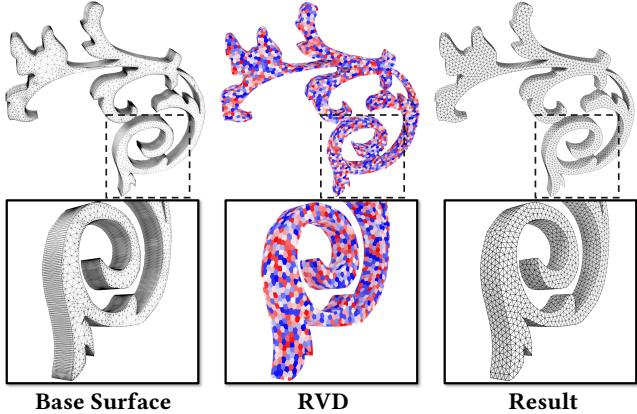


Fig. 18. Ablation study on the low triangle quality of the base surface. The input mesh contains thin and sharp triangles, which are normally undesired. Our method demonstrates resistance to these low-quality meshes as input and outputs a high-quality simplification result.

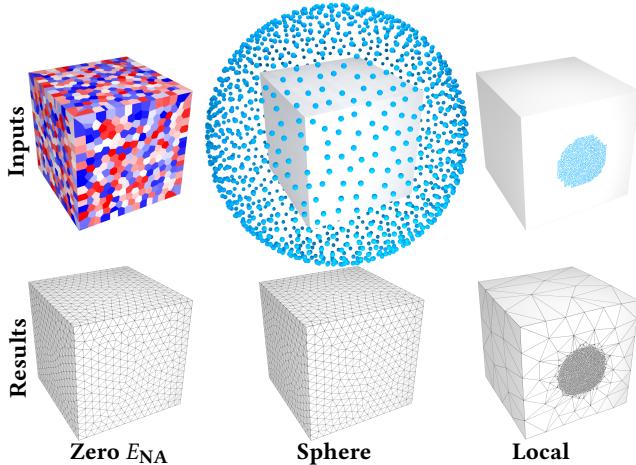


Fig. 19. Our results with different point placement initialization strategies.

to meshes with open boundaries, as shown in the top of Fig. 20. In addition, users can specify the target number of vertices. Despite the disparity in meshing resolution (200, 500, and 1000), our algorithm can consistently meet the requirements of accuracy, triangle quality, and feature alignment; see the bottom of Fig. 20.

5.8 Potential Applications

Mesh Segmentation. Past research indicates that feature lines are crucial for mesh segmentation. However, existing segmentation algorithms struggle to align with weak feature lines. It can be imagined that by simplifying the mesh while preserving its weak features, segmentation becomes more manageable.

Using [Shapira et al. 2008] as the segmentation solver, it is evident from Fig. 21 (a,b) that the segmentation results are significantly improved. This enhancement is due to our algorithm’s ability to consolidate weak features during mesh simplification and reduce the impact of intricate details.

In summary, our algorithm can serve as a preprocessing step for the mesh segmentation task. It not only eases the difficulty of

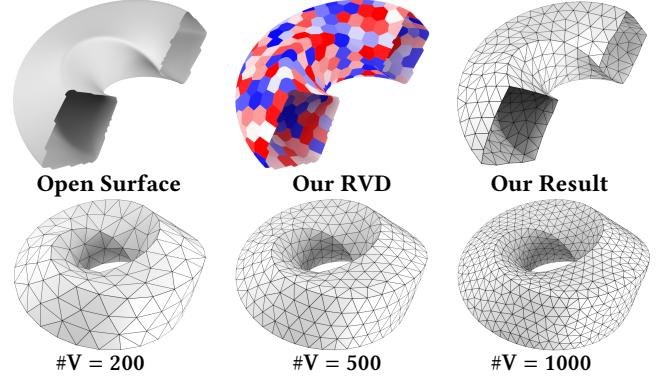


Fig. 20. Ablation study on an open surface (top) and three different target numbers of vertices, $\#V$, (bottom).

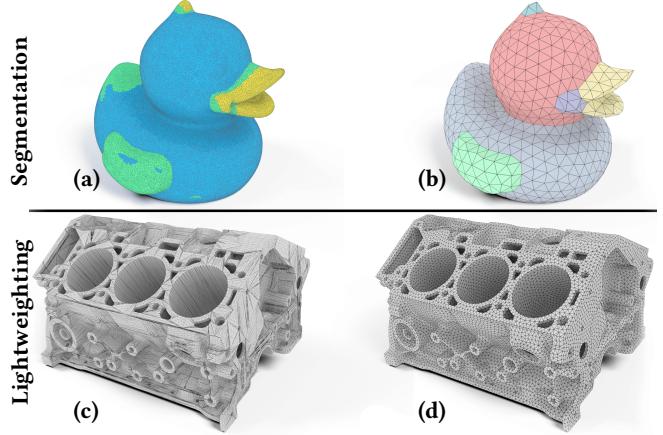


Fig. 21. Two typical applications on mesh segmentation and CAD model lightweight representation.

mesh segmentation but also produces visually straight segmentation boundaries.

Lightweighting. In the realm of Computer-Aided Design (CAD), mesh simplification is a vital tool for achieving model lightweighting—a process that reduces the complexity of CAD models without compromising their functional integrity or design intent. Simplification is crucial for managing intricate CAD models that can be computationally demanding due to their high polygon counts. As seen in Fig. 21 (c), the original CAD model is displayed with all its detailed features, which, while accurate, results in a heavy model that requires significant computational resources. In contrast, Fig. 21 (d) showcases the CAD model post-simplification, where the mesh has been effectively reduced in complexity. This streamlined version preserves the model’s fundamental geometry and essential characteristics, yet is lighter in terms of data size and easier to manipulate and render. The lightweight model offers numerous benefits, including faster loading times, enhanced performance in real-time visualization, and more efficient storage and transfer. Additionally, it supports more effective collaboration and integration with other systems (e.g., VR devices) that have lower hardware specifications.

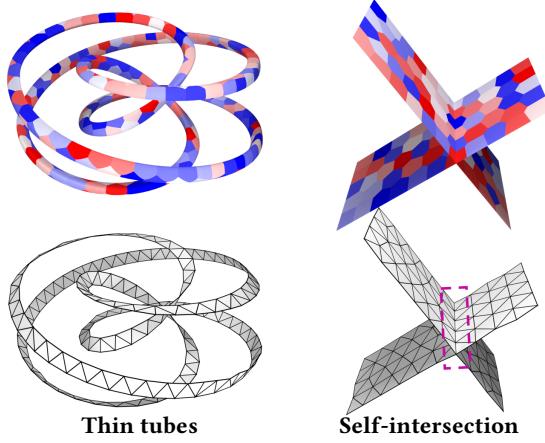


Fig. 22. Our algorithm, in its current form, cannot handle cases where the target number of points is too few. Additionally, it struggles with input models that have self-intersections.

5.9 Limitations

Our methodology has some limitations. The first arises from the lack of a robust solver capable of computing anisotropic Restricted Voronoi Diagrams (RVDs). Consequently, our approach does not fully exploit the potential to produce anisotropic meshes that align with features of interest. The second limitation relates to the suboptimal efficiency of our implementation, which could benefit from GPU-based acceleration. The third limitation, but certainly not the least, is that the RVD strategy proposed in this paper, while simple and effective, may fail when the target number of points is too few. This is because the dual of the RVD cannot define a manifold triangle mesh, as demonstrated in the first example in Fig. 22. However, it is also noted that our RVD strategy performs well as long as the target number of points is sufficient, based on our numerous tests. We also present a scenario where two distinct faces intersect; our technique will lead to the occurrence of a non-manifold artifact. This implies that when the input has self-intersections, our algorithm may produce non-manifold vertices/edges.

6 CONCLUSIONS

In this paper, we propose an objective function that concurrently integrates the requirements of accuracy, triangle quality, and feature alignment. Our function includes the normal anisotropy term and the CVT energy term, balanced with a decaying weight. We conducted extensive experiments to compare our approach with existing state-of-the-art (SOTA) methods, validating its efficacy. Notably, our approach not only meets multiple requirements but also consolidates weak features, setting it apart from other SOTA methods. Additionally, we introduce a simple yet effective technique for computing RVDs on thin-plate models.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. This work is supported by National Key R&D Program of China (2022YFB3303200), National Natural Science Foundation of China (62272277, U23A20312,

62072284) and NSF of Shandong Province (ZR2020MF036). Ningna Wang and Xiaohu Guo were partially supported by National Science Foundation (OAC-2007661). Zichun Zhong was partially supported by National Science Foundation (OAC-1845962, OAC-1910469, and OAC-2311245).

REFERENCES

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

Kanchan Bahirat, Chengyuan Lai, Ryan P. McMahan, and Balakrishnan Prabhakaran. 2018. Designing and evaluating a mesh simplification algorithm for virtual reality. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 14, 3s (2018), 1–26.

Daniela Cabiddu and Marco Attene. 2015. Large mesh simplification for distributed environments. *Computers & Graphics* 51 (2015), 81–89.

Yiqi Cai, Xiaohu Guo, Yang Liu, Wenping Wang, Weihua Mao, and Zichun Zhong. 2016. Surface approximation via asymptotic optimal geometric partition. *IEEE Transactions on Visualization and Computer Graphics* 23, 12 (2016), 2613–2626.

Zhen Chen, Zherong Pan, Kui Wu, Etienne Vouga, and Xifeng Gao. 2023. Robust Low-Poly Meshing for General 3D Models. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–20.

Zhiqin Chen and Hao Zhang. 2021. Neural Marching Cubes. *ACM Transactions on Graphics (Special Issue of SIGGRAPH Asia)* 40, 6 (2021).

Chien-Hsing Chiang, Bin-Shyan Jong, and Tsong-Wuu Lin. 2011. A robust feature-preserving semi-regular remeshing method for triangular meshes. *The Visual Computer* 27 (2011), 811–825.

Fernando De Goes, Katherine Breeden, Victor Ostromoukhov, and Mathieu Desbrun. 2012. Blue noise through optimal transport. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 1–11.

Zhiyang Dou, Cheng Lin, Rui Xu, Lei Yang, Shiqing Xin, Taku Komura, and Wenping Wang. 2022. Coverage axis: Inner point selection for 3d shape skeletonization. In *Computer Graphics Forum*, Vol. 41. Wiley Online Library, 419–432.

Zhiyang Dou, Shiqing Xin, Rui Xu, Jian Xu, Yuanfeng Zhou, Shuangmin Chen, Wenping Wang, Xiuyang Zhao, and Changhe Tu. 2020. Top-down shape abstraction based on greedy pole selection. *IEEE Transactions on Visualization and Computer Graphics* 27, 10 (2020), 3982–3993.

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

Qiang Du and Desheng Wang. 2005. Anisotropic centroidal Voronoi tessellations and their applications. *SIAM Journal on Scientific Computing* 26, 3 (2005), 737–761.

Xingyi Du, Xiaohan Liu, Dong-Ming Yan, Caigui Jiang, Juntao Ye, and Hui Zhang. 2018. Field-Aligned Isotropic Surface Remeshing. In *Computer Graphics Forum*, Vol. 37. 343–357.

John Edwards, Wenping Wang, and Chandrajit Bajaj. 2013. Surface segmentation for improved remeshing. In *Proceedings of the 21st International Meshing Roundtable*. 403–418.

Pascal J. Frey and Houman Borouchaki. 1999. Surface mesh quality evaluation. *International journal for numerical methods in engineering* 45, 1 (1999), 101–118.

Zhiying Fu, Rui Xu, Shiqing Xin, Shuangmin Chen, Changhe Tu, Chenglei Yang, and Lin Lu. 2022. Easymodeling: Easily create 3d models by an immersive vr system. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 5, 1 (2022), 1–14.

Zhanheng Gao, Zeyun Yu, and Michael Holst. 2013. Feature-preserving surface mesh smoothing via suboptimal Delaunay triangulation. *Graphical Models* 75, 1 (2013), 23–38.

Michael Garland and Paul S. Heckbert. 1997. Surface simplification using quadric error metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. 209–216.

Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. 1993. Mesh optimization. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*. 19–26.

Kaimo Hu, Dong-Ming Yan, David Bommes, Pierre Alliez, and Bedrich Benes. 2016. Error-bounded and feature preserving surface remeshing with minimal angle improvement. *IEEE transactions on visualization and computer graphics* 23, 12 (2016), 2560–2573.

Wenzel Jakob, Marco Tarini, Daniele Panozzo, and Olga Sorkine-Hornung. 2015. Instant field-aligned meshes. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 1–15.

Myeong-Cheol Ko and Yoon-Chul Choy. 2002. 3D Mesh simplification for effective network transmission. In *5th IEEE International Conference on High Speed Networks and Multimedia Communication*. 284–288.

Leif Kobbelt, Swen Campagna, and Hans-Peter Seidel. 1998. A general framework for mesh decimation. In *Graphics interface*, Vol. 98. 43–50.

Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. 2019. Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9601–9611.

Thibault Lescoat, Hsueh-Ti Derek Liu, Jean-Marc Thiery, Alec Jacobson, Tamy Boubekeur, and Maks Ovsjanikov. 2020. Spectral mesh simplification. In *Computer Graphics Forum*, Vol. 39. 315–324.

Bruno Lévy and Yang Liu. 2010. Lp centroidal voronoi tessellation and its applications. *ACM Transactions on Graphics (TOG)* 29, 4 (2010), 1–11.

Minglei Li and Liangliang Nan. 2021. Feature-preserving 3D mesh simplification for urban buildings. *ISPRS Journal of Photogrammetry and Remote Sensing* 173 (2021), 135–150.

Hsueh-Ti Derek Liu, Mark Gillespie, Benjamin Chislett, Nicholas Sharp, Alec Jacobson, and Keenan Crane. 2023. Surface simplification using intrinsic error metrics. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–17.

Yang Liu, Wenping Wang, Bruno Lévy, Feng Sun, Dong-Ming Yan, Lin Lu, and Chen-gelei Yang. 2009. On centroidal Voronoi tessellation—energy smoothness and fast computation. *ACM Transactions on Graphics (TOG)* 28, 4 (2009), 1–17.

Chenlei Lv, Weisi Lin, and Jianmin Zheng. 2022. Adaptively Isotropic Remeshing based on Curvature Smoothed Field. *IEEE Transactions on Visualization and Computer Graphics* (2022).

Xuegang Ma, Jinjin Zheng, Yong Shui, Hongjun Zhou, and Lianguan Shen. 2012. A novel method of mesh simplification using hausdorff distance. In *2012 Third World Congress on Software Engineering*. IEEE, 136–139.

Hiromu Ozaki, Fumihiro Kyoto, and Takashi Kanai. 2015. Out-of-Core Framework for QEM-based Mesh Simplification.. In *EuroVis*. 87–96.

Dakshita Panchal and Deepak Jayawal. 2022. Feature sensitive geometrically faithful highly regular direct triangular isotropic surface remeshing. *Sādhanā* 47, 2 (2022), 94.

Lior Shapira, Ariel Shamir, and Daniel Cohen-Or. 2008. Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer* 24 (2008), 249–259.

Kehua Su, Na Lei, Wei Chen, Li Cui, Hang Si, Shikui Chen, and Xianfeng Gu. 2019. Curvature adaptive surface remeshing by sampling normal cycle. *Computer-Aided Design* 111 (2019), 1–12.

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

Philip Trettner and Leif Kobbelt. 2020. Fast and robust qef minimization using probabilistic quadrics. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 325–334.

Sébastien Valette and Jean-Marc Chassery. 2004. Approximated centroidal voronoi diagrams for uniform polygonal mesh coarsening. In *Computer Graphics Forum*, Vol. 23. Wiley Online Library, 381–389.

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

Ningna Wang, Hui Huang, Shibo Song, Bin Wang, Wenping Wang, and Xiaohu Guo. 2024b. MATTOP: Topology-preserving Medial Axis Transform with Restricted Power Diagram. *arXiv preprint arXiv:2403.18761* (2024).

Ningna Wang, Bin Wang, Wenping Wang, and Xiaohu Guo. 2022a. Computing Medial Axis Transform with Feature Preservation via Restricted Power Diagram. *ACM Transactions on Graphics (TOG)* 41, 6 (2022), 1–18.

Pengfei Wang, Zixiong Wang, Shiqing Xin, Xifeng Gao, Wenping Wang, and Changhe Tu. 2022b. Restricted Delaunay Triangulation for Explicit Surface Reconstruction. *ACM Trans. on Graphics* (2022).

Yiqun Wang, Dong-Ming Yan, Xiaohan Liu, Chengcheng Tang, Jianwei Guo, Xiaopeng Zhang, and Peter Wonka. 2018. Isotropic surface remeshing without large and small angles. *IEEE Transactions on Visualization and Computer Graphics* 25, 7 (2018), 2430–2442.

Zimeng Wang, Zhiyang Dou, Rui Xu, Cheng Lin, Yuan Liu, Xiaoxiao Long, Shiqing Xin, Lingjie Liu, Taku Komura, Xiaoming Yuan, et al. 2024a. Coverage Axis++: Efficient Inner Point Selection for 3D Shape Skeletonization. *arXiv preprint arXiv:2401.12946* (2024).

Jin Wei and Yu Lou. 2010. Feature preserving mesh simplification using feature sensitive metric. *Journal of Computer Science and Technology* 25, 3 (2010), 595–605.

Lijun Xie, Ligang Chen, Jianjun Chen, and Yao Zheng. 2012. Surface Remeshing with Feature Preservation. *Journal of Information & Computational Science* 9, 9 (2012), 2393–2400.

Qun-Ce Xu, Dong-Ming Yan, Wenbin Li, and Yong-Liang Yang. 2019. Anisotropic surface remeshing without obtuse angles. In *Computer Graphics Forum*, Vol. 38. 755–763.

Rui Xu, Zixiong Wang, Zhiyang Dou, Chen Zong, Shiqing Xin, Mingyan Jiang, Tao Ju, and Changhe Tu. 2022. RFEPS: Reconstructing feature-line equipped polygonal surface. *ACM Transactions on Graphics (TOG)* 41, 6 (2022), 1–15.

Dong-Ming Yan, Jianwei Guo, Xiaohong Jia, Xiaopeng Zhang, and Peter Wonka. 2014. Blue-Noise Remeshing with Farthest Point Optimization. In *Computer Graphics Forum*, Vol. 33. 167–176.

Dong-Ming Yan, Bruno Lévy, Yang Liu, Feng Sun, and Wenping Wang. 2009. Isotropic remeshing with fast and exact computation of restricted Voronoi diagram. In *Computer graphics forum*, Vol. 28. Wiley Online Library, 1445–1454.

Dong-Ming Yan and Peter Wonka. 2013. Gap processing for adaptive maximal Poisson-disk sampling. *ACM Transactions on Graphics (TOG)* 32, 5 (2013), 1–15.

Dong-Ming Yan and Peter Wonka. 2015. Non-obtuse remeshing with centroidal Voronoi tessellation. *IEEE Transactions on Visualization and Computer Graphics* 22, 9 (2015), 2136–2144.

Yunxiao Zhang, Zixiong Wang, Zihan Zhao, Rui Xu, Shuangmin Chen, Shiqing Xin, Wenping Wang, and Changhe Tu. 2023. A Hessian-Based Field Deformer for Real-Time Topology-Aware Shape Editing. In *SIGGRAPH Asia 2023 Conference Papers*. 1–11.

Tong Zhao, Laurent Busé, David Cohen-Steiner, Tamy Boubekeur, Jean-Marc Thiery, and Pierre Alliez. 2023. Variational shape reconstruction via quadric error metrics. In *ACM SIGGRAPH 2023 Conference Proceedings*. 1–10.

Zichun Zhong, Xiaohu Guo, Wenping Wang, Bruno Lévy, Feng Sun, Yang Liu, Weihua Mao, et al. 2013. Particle-based anisotropic surface meshing. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 99–1.