Adaptively Isotropic Remeshing based on Curvature Smoothed Field

Chenlei Lv, *Member, IEEE*, Weisi Lin*, *Fellow, IEEE*, Jianmin Zheng

Abstract—With the development of 3D digital geometry technology, 3D triangular meshes are becoming more useful and valuable in industrial manufacturing and digital entertainment. A high quality triangular mesh can be used to represent a real world object with geometric and physical characteristics. While anisotropic meshes have advantages of representing shapes with sharp features (such as trimmed surfaces) more efficiently and accurately, isotropic meshes allow more numerically stable computations. When there is no anisotropic mesh requirement, isotropic triangles are always a good choice. In this paper, we propose a remeshing method to convert an input mesh into an adaptively isotropic one based on a curvature smoothed field (CSF). With the help of the CSF, adaptively isotropic remeshing can retain the curvature sensitivity, which enables more geometric features to be kept, and avoid the occurrence of obtuse triangles in the remeshed model as much as possible. The remeshed triangles with locally isotropic property benefit various geometric processes such as neighbor-based feature extraction and analysis. The experimental results show that our method achieves better balance between geometric feature preservation and mesh quality improvement compared to peers. We provide the implementation codes of our resampling method at github.com/vvvwo/Adaptively-Isotropic-Remeshing.

Index Terms—Adaptively isotropic, remeshing, curvature smoothed field.

1 Introduction

As one kind of 3D data representation, triangular meshes have been successfully used in many commercial applications such as film industry, intelligent city management, advertising, automated manufacturing, and medical analysis. Compared to point clouds and depth maps, triangular meshes retain complete topological structure with better 2-manifold property. They support various geometric analyses such as heat flow computation [1], Ricci flow [2], optimal transportation [3], functional map [4], and geodesic curve extraction [5]. To benefit these applications, high quality meshes are often required. While anisotropic meshes have advantages especially in terms of computational efficiency and representation accuracy [6], they break the uniformity of neighboring structures, which may cause some numerical issues. For instance, some computer aided design (CAD) models often have very thin and long (i.e., distorted) triangle faces, which break the isotropic property of the mesh and have significant differences in local structures. The low quality of meshes will reduce the reliability of geometric feature extraction and analysis [1]–[5]. When there is no anisotropic mesh adaption, isotropic meshes are always a good choice, which provide good numerical stability of computations.

Remeshing helps to convert a low quality mesh into an isotropic mesh. Following the prior knowledge [7] [8], two typical frameworks are designed for remeshing: edge reconnection and neighborhood optimization. The edge reconnection framework attempts to reconnect vertices of the input mesh to achieve related remeshing result. It can achieve the suitable isotropic property without performing complex optimization. However, the methods in this framework are sensitive to original vertex distribution. For some meshes with extremely distorted triangular faces, the methods do not work well [9]. The neighborhood optimization framework optimizes the local structure of vertices directly. Examples are centroidal Voronoi tessellation (CVT) based and particle-based approaches that are robust to original vertex distribution. The drawback is the huge computation cost for local structure optimization [10]. Moreover, these approaches are sensitive to the curvature variation of the input mesh. Their performance is not stable in geometric feature preservation.

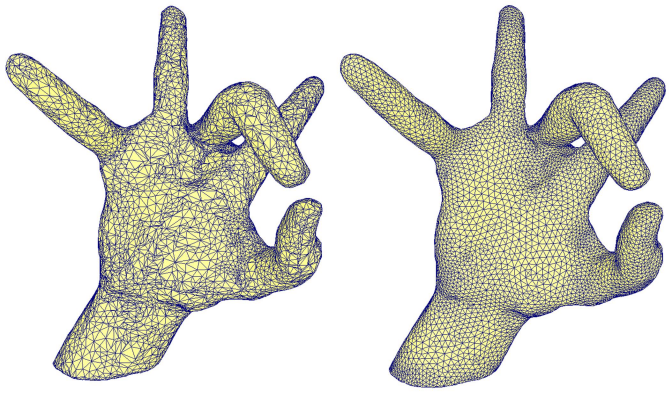


Fig. 1. Adaptively isotropic remeshing. Left: original mesh; right: the result of adaptively isotropic remeshing with curvature sensitivity.

This paper considers remeshing that balances geometric feature preservation and mesh quality improvement. For this purpose, we propose an adaptively isotropic remeshing method, which can be classified into the edge reconnection

• *Chenlei Lv, Weisi Lin, and Jianmin Zheng are with the School of Computer Science and Engineering, Nanyang Technological University. The correspoding author is Weisi Lin, e-mail:(wslin@ntu.edu.sg).*

Manuscript submitted XXX XX, 2022

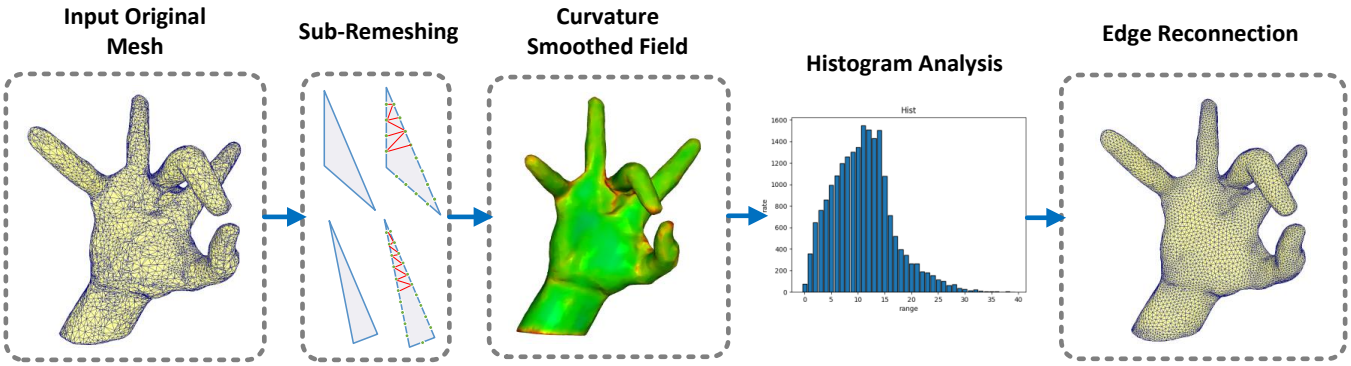


Fig. 2. The pipeline of our adaptively isotropic remeshing method. Firstly, the original mesh is inputted; secondly, distorted triangular faces are checked and refined by sub-remeshing; thirdly, the curvature smoothed field (CSF) is computed to smooth curvature values of the mesh; fourthly, the histogram analysis is performed on the smoothed curvature values at vertices to provide some statistics; finally, the edge reconnection implements adaptively isotropic remeshing based on CSF and the histogram analysis result.

category. To improve the robustness for various vertex distributions, we introduce a curvature smoothed field (CSF) and use it to control the edge reconnection. According to the CSF, we design a histogram-based approach for edge reconnection to improve the distribution of edge lengths of remeshed models. The size of triangles cross the border of regions with different curvatures can be controlled to avoid unstable face distortion and maintain curvature variation. Figure 1 shows one example where the point density is proportional to the curvature of the surface. We also design a sub-remeshing step as a pre-process to handle meshes with extremely distorted triangular faces. This step is a local refinement of the distorted faces. The pipeline of our adaptively isotropic remeshing method is shown in Figure 2, which consists of initial sub-remeshing, curvature smoothed field generation, histogram analysis, and edge reconnection. The method can handle various low-quality, artificially designed meshes. The main contributions of the paper are as follows.

- We introduce a curvature smoothed field for adaptively isotropic remeshing. The field smooths the curvature of the original mesh. It establishes a continuous curvature distribution which guides to reduce the distortion of edge lengths during the remeshing process.
- We design a histogram-based edge reconnection scheme based on the CSF to automatically improve the edge length distribution of the remeshed model, which achieves the consistency between vertex density and curvature variation.
- We present a local refinement based pre-process step (sub-remeshing) to handle meshes with extremely distorted triangular faces. This can improve the robustness and efficiency of the remeshing algorithm.

The rest of the paper is organized as follows. Sec. 2 reviews some classical remeshing methods. Sec. 3 introduces the curvature smoothed field, followed by the histogram-based edge reconnection method in Sec. 4. We report our experimental results in Sec. 5 to show the effectiveness and efficiency of the proposed method. Sec. 6 concludes the paper.

2 Related Works

The problem of remeshing has been studied extensively and many methods have been developed. In this section, we briefly review three categories of remeshing: edge reconnection-based, CVT-based, and particle-based methods.

Edge reconnection-based methods rebuild the connectivity of vertices to realize remeshing. Basic operations used to perform the reconnection process are edge split, edge collapse, edge flip, and vertex relocation. Alliez *et al.* [11] proposed a remeshing method to reconnect the vertices of a mesh based on a principal direction field. Jiao *et al.* [12] proposed a remeshing framework with anisotropic mesh adaptation for dynamic surface meshes. Dunyach *et al.* [13] presented an adaptively isotropic remeshing approach to balance process efficiency, mesh quality, and curvature sensitivity. It considers the influence of curvature changes in the mesh such that the edge lengths of remeshed triangles respect the curvature values. Dapogny *et al.* [14] implemented adaptive remeshing in an implicitly-defined domain. Narain *et al.* [15] designed an adaptive remeshing method for cloth simulation. Dassi *et al.* [16] proposed an optimization-based remeshing method for CAD surfaces, which outputs curvature adapted anisotropic surface meshes with a low number of faces. Wang *et al.* [17] proposed an isotropic remeshing method by removing large and small angles. Xu *et al.* [10] designed a new edge reconnection strategy to reduce the occurrence of obtuse angles. It does not pursue the strictly isotropic property. Instead, it just removes obtuse triangles from the remeshing result and generally cannot assure global or local isotropic results. Verhoeven *et al.* [18] introduced a remeshing method to compute a quad dominant representation for the developable mesh. It utilizes the principal curvature field to guide the edge reconnection. The remeshing effectively simplifies the representation of the original mesh while keeping geometric features. Liu *et al.* [19] presented a robust remeshing method for mesh

simplification. It can generate high-quality meshes with lower approximation errors. Hu *et al.* [20] provided a novel divide-and-conquer methodology for manifold-constrained geometric optimization tasks, including compatible remeshing, quadrilateral mesh optimization, and minimum angle improvement. In general, these methods do not require complex optimization; their computation cost is relatively low and convenient to implement. However, the adaptive process of these methods is sensitive to the point distribution of the input mesh, especially in regions with significant curvature variations.

CVT-based methods use the Voronoi Diagram or its dual processing, Delaunay triangulation, to achieve remeshing. Such methods optimize the mesh structure by Voronoi cell programming [21]. Alliez *et al.* [22] proposed an isotropic remeshing method by Delaunay triangulation. Yan *et al.* [23] introduced Restricted Voronoi Diagram (RVD) computation to optimize Voronoi cells. The method was further improved by the farthest point optimization [24]. Goes *et al.* [25] proposed a weighted triangulation method with diagram optimization. Liu *et al.* [26] designed a Delaunay remeshing method with intrinsic control. Ye *et al.* [27] extended CVT to geodesic-based CVT for remeshing. Zheng and Tan [28] proposed a parallel implementation of CVT on GPU. Usually, the CVT-based methods can produce high quality output meshes with Voronoi Diagram optimization, but they are sensitive to the regions with sharp curvature changes. Using the intrinsic metric control can reduce such influences [26] [27], but the process becomes more time-consuming. Levy *et al.* [29] proposed an anisotropic remeshing method by transforming the 3D anisotropic space into a higher dimensional isotropic space. The curvature-adapted mesh is obtained by centroidal Voronoi tessellation (CVT) in this high dimensional space. Based on the similar idea, Dassi *et al.* [30] presented an intrinsic anisotropic remeshing in R^6 space. It reduces the production of flipped triangles. Su *et al.* [31] proposed to utilize conformal and normal cycle parameterization to implement curvature adaptive surface remeshing. The quality of the result depends on the compactness of the original mesh. Yi *et al.* [32]proposed a practical Delaunay mesh simplification. It can transfer arbitrary manifold triangle mesh to fit user-specified resolution while meeting the local Delaunay condition. Zhang *et al.* [33] proposed a remeshing method with several hard constraints, including bounding approximation errors and ensuring Delaunay conditions. Lv *et al.* [34] introduced isotropic resampling for mesh reconstruction. The obtuse triangles are effectively reduced to fit the Delaunay conditions. Hou *et al.* [35] proposed a method to compute the restricted Voronoi Diagram on the signed distance field. It can naturally work with CVT to optimize raw meshes while improving efficiency.

Particle-based methods are more direct approaches for remeshing, which perform vertex and structure optimization in local regions. Compared to CVT-based methods, particle-based methods do not compute the Voronoi Diagram, which significantly improves computational efficiency. Premvzoe *et al.* [36] proposed a particle-based method for fluid simulation. Hieber *et al.* [37] presented a particle-based simulation for human organ material models. Meyer *et al.* [38] constructed a dynamic particle system to implement the multi-material volumes. Sullivan [39] presented several applications of particle-based discrete element modeling. Takamatsu *et al.* [40] presented a practical solution to fast animation of viscoelastic fluids by particle-based simulation. Zhong *et al.* [41] proposed a particle-based anisotropic remeshing method in a high dimensional space. Cheng *et al.* [42] proposed an adaptive refinement for particle-based remeshing. Zhong *et al.* [43] introduced a unified particle-based formulation for mesh reconstruction. Basically, particle-based methods are widely used for dynamic remeshing tasks due to their computational efficiency. However, the methods are sensitive to sharp curvature variations and easily trap into a local minimum.

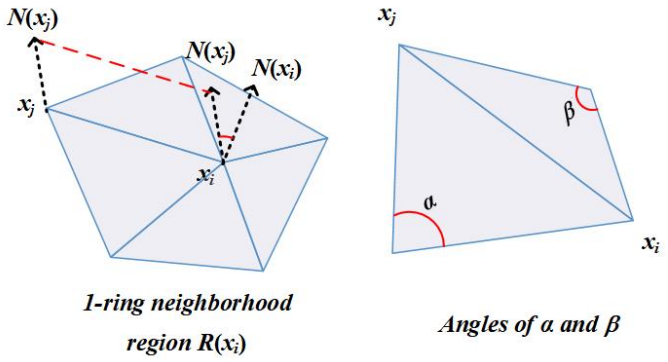


Fig. 3. Left: 1-ring neighborhood region of vertex x_i; right: two angles for the computation of the cotangent weight.

Our adaptively isotropic remeshing method basically belongs to the first category while utilizing the CSF to control edge lengths during remeshing. Compared to the classical adaptive remeshing methods, the CSF smooths the curvature values and thus improves the robustness to the quality of the original mesh. The proposed method keeps the advantages of edge reconnection-based methods and is robust in handling meshes with different vertex distributions. With the histogram-based scheme, it achieves a good balance between geometric feature preservation and mesh quality improvement.

3 Curvature Smoothed Field

Our goal is to remesh an input mesh such that the mesh quality is improved and meanwhile the geometric feature is well preserved. To realize geometric feature preservation, the vertices should be appropriately distributed in space to respect the curvature variations. More vertices are allocated to the regions with high curvature. Consequently, the area of triangular faces should be proportional to the curvature values. Such a property is called curvature sensitivity. Hence the edge reconnection should take the curvature values of the input mesh into account. To improve the mesh quality, the isotropic property is sought. It means that all triangular faces should be approximately equilateral triangles with a similar area locally. Obviously, it is contradictory to keep geometric features and improve the mesh quality simultaneously. Thus the adaptively isotropic remeshing is proposed to balance the two requirements.

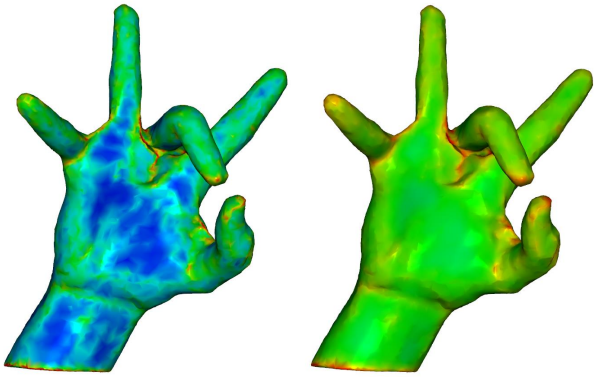


Fig. 4. Comparison of curvature fields by color maps. Left: the initial curvature field; right: the CSF.

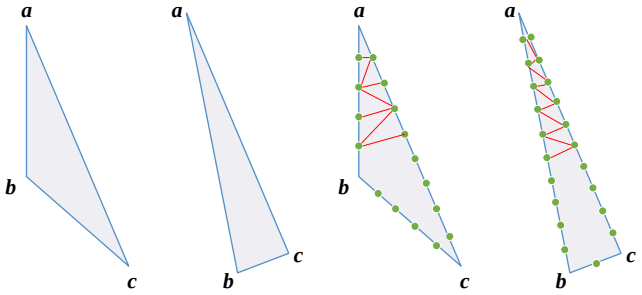


Fig. 5. Two thin and long triangles on the left, and the results after sub-remeshing on the right.

Our basic idea of adaptively isotropic remeshing is to control the area changes of triangles in the regions with different curvature values. A common approach is to define a new metric in a high dimensional space where the influence of geometric features is considered. Then the isotropic result in the high dimensional space gives an adaptively isotropic result in the original Euclidean space. Note that the related values such as curvature for specific geometric feature representation does not necessarily change continuously between adjacent regions, which may produce distorted faces. We thus propose to construct a curvature smoothed field to solve the problem.

We begin with computing a curvature value at each vertex, which forms the initial discrete curvature field S. Specifically, for vertex x_i, we define a normal-based curvature value to be

$$s(x_i) = \max_{x_j \in R(x_i)} \{\arccos \langle N(x_i) \cdot N(x_j) \rangle\}, \tag{1}$$

where $s(x_i)$ represents the curvature value at vertex x_i, which is computed from normal vector $N(x_i)$ of the mesh at vertex x_i and normal vector $N(x_j)$ of the mesh at vertex x_j where x_j is in the 1-ring neighborhood $R(x_i)$ of x_i [44], as shown in Figure 3 (left). This curvature intuitively represents how much the surface bends in the local neighborhood of a vertex.

Next, we smooth the initial curvature field S to generate a smooth scalar field for the mesh, which defines the relationship between each vertex to its neighbors. This is done by introducing the Laplacian with the cotangent weights [45] [46]:

$$\triangle s_i = \sum_{j \in R_i} (\frac{w_{ij}}{w_i} \cdot s_j) - s_i, \; w_i = \sum_{j \in R_i} w_{ij}, \tag{2}$$

where s_i and s_j are the curvature values at vertices x_i and x_j, R_i is the set of indices of vertices in the 1-ring neighborhood region $R(x_i)$, and w_{ij} is the cotangent weight computed by $w_{ij} = (\cot \alpha + \cot \beta)/2$ with two angles α and β opposite to edge $x_i x_j$ as shown in Figure 3 (right). Note that this cotangent weight could be negative. To overcome this issue, we modify the weight as follows:

$$w_{ij} = \begin{cases} \frac{\cot \alpha + \cot \beta}{2}, & \text{if } \cot \alpha + \cot \beta > 0 \\ \eta, & \text{otherwise} \end{cases} \tag{3}$$

where η is a small positive number. In our implementation, we set $\eta = 0.0001$.

The smooth processing is to iteratively update each s_i in S:

$$s_i \leftarrow s_i + (1 - \mu) \cdot \triangle s_i, \tag{4}$$

where $\mu \in (0, 1)$ is a weight controlling the update rate. The default value of μ is set to 0.5. To avoid the smooth processing converging to a trivial case that all s_i become the same, we choose a vertex v_{max} that has the biggest value of s_i and then another vertex v_{min} among all vertices not in the 1-ring neighborhood of v_{max}, which has the smallest value of s_i, and we do not apply the smooth processing to v_{max} and v_{min}. After the smooth processing, the Laplacian $\triangle s_i$ approaches zero, implying that the curvature field reaches an approximately harmonic state, and then the CSF is achieved. Figure 4 shows the comparison of an initial curvature field and the CSF. It can be seen that after the smooth processing, the curvature field becomes smoother.

Remark. The above smooth processing is guaranteed to converge. The mathematical proof is given in the appendix.

4 Adaptively Isotropic Remeshing

We are ready to present our adaptively isotropic remeshing method, which consists of two processes: sub-remeshing and histogram-based edge reconnection. The sub-remeshing is pre-processing step that splits long edges to give a coarse refinement of thin and long triangles. It is introduced to handle extremely distorted triangles, which helps improve the robustness and efficiency of the whole algorithm. The histogram-based edge reconnection is a fine process guided by the CSF, which reconnects vertices of the mesh to achieve the adaptively isotropic property and keep curvature sensitivity. Below we elaborate on the details of these two processes.

4.1 Sub-Remeshing

In some artificially designed meshes, many distorted triangles have uneven edge length distribution. Such triangles affect the performance of adaptively isotropic remeshing and even lead to incorrect remeshing results. See Figure 5 for two distorted triangular faces where edge ac has a much larger length than one of its adjacent edges.

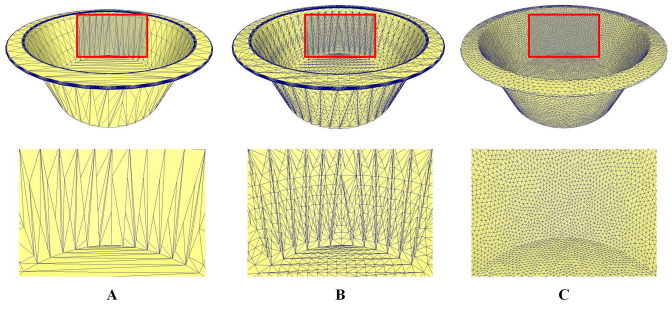


Fig. 6. Sub-remeshing and adaptively isotropic remeshing. A: input mesh; B: sub-remeshing result; C: adaptively isotropic remeshing result.

Algorithm 1 Implementation of Sub-Remeshing

1) Input mesh M with $\{V, E, F\}$ (V: vertices, E: edges, and F: faces).
2) Compute the average edge length E_{ave} from the edge set $\{E\}$.
3) Detect edges with length $(> 2E_{ave})$ and insert them to $\{E_d\}$.
4) Collect distorted faces to form set F_d.
5) **for** e **in** $\{E_d\}$
6) Compute current edge length E_e.
7) Compute # of inserting points $n = \lfloor E_e/E_{ave} \rfloor$.
8) Insert n points in the current edge & update $\{V\}$.
9) Add the faces (sharing the current edge) into F_d.
10) **end for**
11) **for** f **in** $\{F_d\}$
12) Find the edge with longest length in face f.
13) Connect the inserting points in the edge to the points in the other two edges of f according to the length ratio.
14) If the new edge length $> 2E_{ave}$, insert new points to split the edge.
15) Update $\{V\}$,$\{E\}$, and $\{F\}$ according to new edges.
16) **end for**
17) Output $M^{'}$ with new $\{V^{'}, E^{'}, F^{'}\}$.

Sub-remeshing divides the edges with long length by inserting new vertices and reconnects the vertices to form new edges. The process includes three steps: 1). compute the average E_{ave} of all edge lengths; 2). for those edges of the input mesh whose length is larger than $2E_{ave}$, divide them by inserting points; 3). reconnect the vertices in each of the faces sharing the edges. The reconnection strategy includes two cases: the triangle includes just one larger edge and two more edges. For the first case (refer to the triangle in the first column of Figure 5), we insert new points to isometrically cut the larger edge with length E_L. The number of points is equal to $\lceil \mathrm{E_L}/\mathrm{E_{ave}} \rceil$, which makes each new edge shorter and closer than E_{ave}. We insert points with the same number into the other two edges. According to the point order, there is a one-to-one correspondence between points on the larger edge and other edges. Then, we connect points in the same order to generate a set of adjacent quadrilaterals. Finally, we insert a new edge into each quadrilateral to produce two new triangles. The new edge should avoid the production of obtuse angles as much as possible. For the second case (refer to the triangle in the second column of Figure 5), we also cut the largest edge of the length E_L with the same number of points. The difference is that we just insert the same number of points into the second larger edge to establish the correspondence. Then we generate new triangles with the same method as case one. We iteratively check the newly generated triangles and use the reconnection strategy to keep all edges having length smaller than $2E_{ave}$. By inserting new points into edges with large length, the long edges are split so that their lengths are in an acceptable range. The implementation of sub-remeshing is given in Algorithm 1. Figure 6(B) shows examples of sub-remeshing, where the long edges are split and the triangular faces are divided into small ones.

4.2 Histogram-based Edge Reconnection

The histogram-based edge reconnection is similar to isotropic remeshing [13] and mesh optimization [9]. The difference is that it controls edge lengths guided by CSF, which gives a smoothed vertex distribution. Precisely, with a histogram analysis, the reconnection process automatically controls edge lengths taking the curvature values into consideration. This helps achieve the balance between geometric feature preservation and mesh quality improvement. In [13], a similar approach is proposed, which is guided by a curvature field. Without smoothing of the field, the area changes of adjacent triangles with different curvature values may become unstable, and the remeshing result is sensitive to the positions of vertices.

Our edge reconnection proceeds in the following steps:

1) Compute the curvature values for each vertex and create CSF by smoothing the values (using Equations 1-4).
2) Compute the average edge length l of the mesh.
3) Assign a multiplication factor m for each vertex according to its curvature value obtained from CSF.
4) Split an edge if its length is larger than $\frac{5}{3}l \times \min\{m_a, m_b\}$ where m_a and m_b represent the multiplication factors assigned to the two endpoints of the edge.
5) Collapse an edge into its middle point if its length is smaller than $\frac{4}{5}l \times \max\{m_a, m_b\}$ where m_a and m_b represent the multiplication factors assigned to the two endpoints of the edge.
6) Flip the edge to implement valence improvement. If the flip generates obtuse angles, the flip will not be processed.
7) If a vertex is not on the boundary of the mesh, relocate it to the center of its 1-ring neighborhood on the tangent plane.

Steps 1 and 2 are for initialization. Steps 3 to 7 are for actual edge reconnection performed iteratively. The basic processes include split, collapse, flip, and vertex relocation [44]. Figure 7 shows examples of these processes for remeshing.

One important strategy for achieving adaptively isotropic remeshing is to properly define the multiplication factor m for each vertex (see Step 3 in the above procedure).

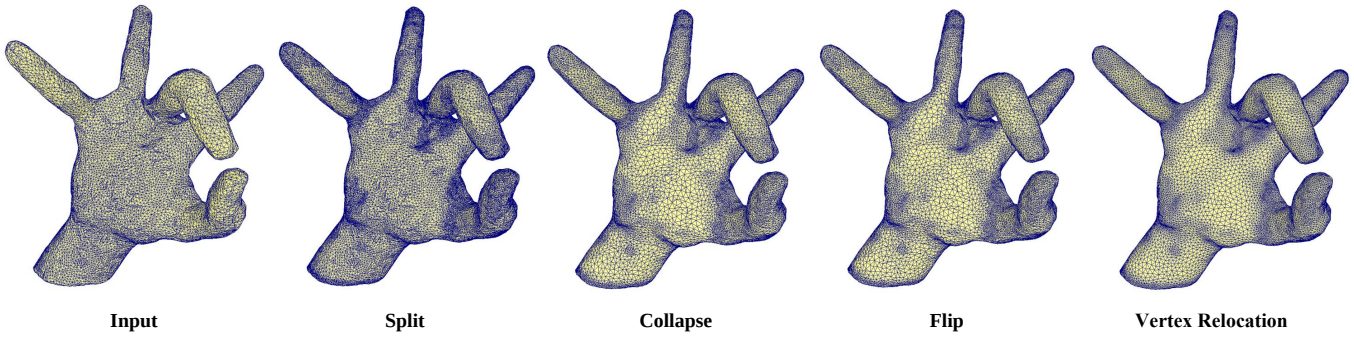


Fig. 7. An instance of adaptively isotropic remeshing.

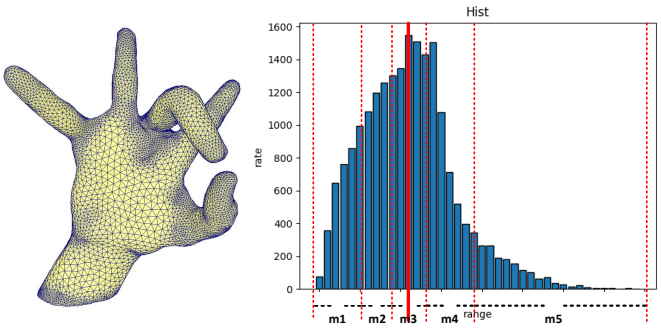


Fig. 8. Histogram-based edge reconnection. $\{m_1, ..., m_5\} = \{1.8, 1.4, 1.0, 0.8, 0.6\}$, X axis: curvature; Y axis: number of vertices. The bins corresponding to m_1, m_2 and the left half of the bin corresponding to m_3 contain the same number of vertices. Similarly, the bins corresponding to m_4, m_5 and the right half of the bin corresponding to m_3 contain the same number of vertices.

The value of m influences how many new points should be inserted. When all the m values are equal, adaptively isotropic remeshing degenerates to isotropic one [44]. To reflect curvature variation, the value of m is chosen according to the curvature distribution. In particular, we group the curvature distribution into k intervals: $[s_{g-1}, s_g], g = 1, 2, \cdots, k$, where k is an odd integer preset by the user. We then set k values for $m : m_1, m_2, \cdots, m_k$, respectively. Then for each vertex p_i we establish the following correspondence:

$$m(p_i) = m_g \quad \text{if} \quad s(p_i) \in [s_{g-1}, s_g]\,. \tag{5}$$

To group the curvature distribution, we first compute the histogram of curvature for all vertices. Then we select the curvature with the maximal count in the histogram as the base value (see the red line in Figure 8)). If there have different maximal values exist at the same time, we select the one with the middle position. It balances the global center and gravity center of curvature statistics as much as possible. Next we use the base value to form k groups. Specifically, denote by T_r the total number of vertices whose curvature is larger than the base value, and by T_l the total number of vertices whose curvature is not larger than the base value. We find the partition of the curvature distribution:

$$[s_0, s_k] = \bigcup_{j=1}^{k} [s_{j-1}, s_j] \tag{6}$$

where s_0 and s_k are the minimal and maximal curvature values among all vertices of the mesh, such that there are $2T_l/k$ vertices whose curvature is in $[s_{l_1}, s_l)$ for $l \leq k/2$, and there are $2T_r/k$ vertices whose curvature is in $(s_r, s_{r+1}]$ for $r \geq k/2$.

To set values for $m_g, g = 1, 2, \cdots, k$, we let $m_{\frac{k+1}{2}} = 1$, which corresponds to the base value of curvature. The others are distributed at the two sides of 1, corresponding to the statistical distributions of curvature. To avoid generating thin and long triangles, the ratio of a triangle's longest and shortest edge lengths should be bounded (for example, by $\sqrt{2} : 1$). Thus we set values in $\{m\}$ to be $\{1.8, 1.4, 1, 0.8, 0.6\}$ for $k = 5$, which is our default value. This can achieve a good balance in practice. An example is shown in Figure 8.

5 Experiments

We evaluate the performance of our adaptively isotropic remeshing in this section. The experimental point cloud models were selected from SHREC [47] and ModelNet [48]. We conducted the experiments on a machine equipped with Intel Xeon W 2133 3.6G Hz, 32 GB RAM, Quadro P620, and with Windows 10 as its running system and Visual Studio 2019 (64 bit) as the development platform. The evaluation is performed using various metrics, including geometric consistency, mesh quality, and curvature sensitivity. Quantitative comparisons are also provided in terms of convergence and runtime.

5.1 Geometric Consistency

Geometric consistency is one of the most important metrics for remeshing. If the geometric consistency between the original mesh and the remeshing result can not be kept, it means that shape fidelity is broken. According to the benchmark [8], Hausdorff distance is often utilized to measure the geometric consistency. The formula is

$$H(M_O, M_R) = \max\{\max_{p_i \in M_O}\{d(p_i, M_R)\}, \max_{q_j \in M_R}\{d(q_j, M_O)\}\}, \tag{7}$$

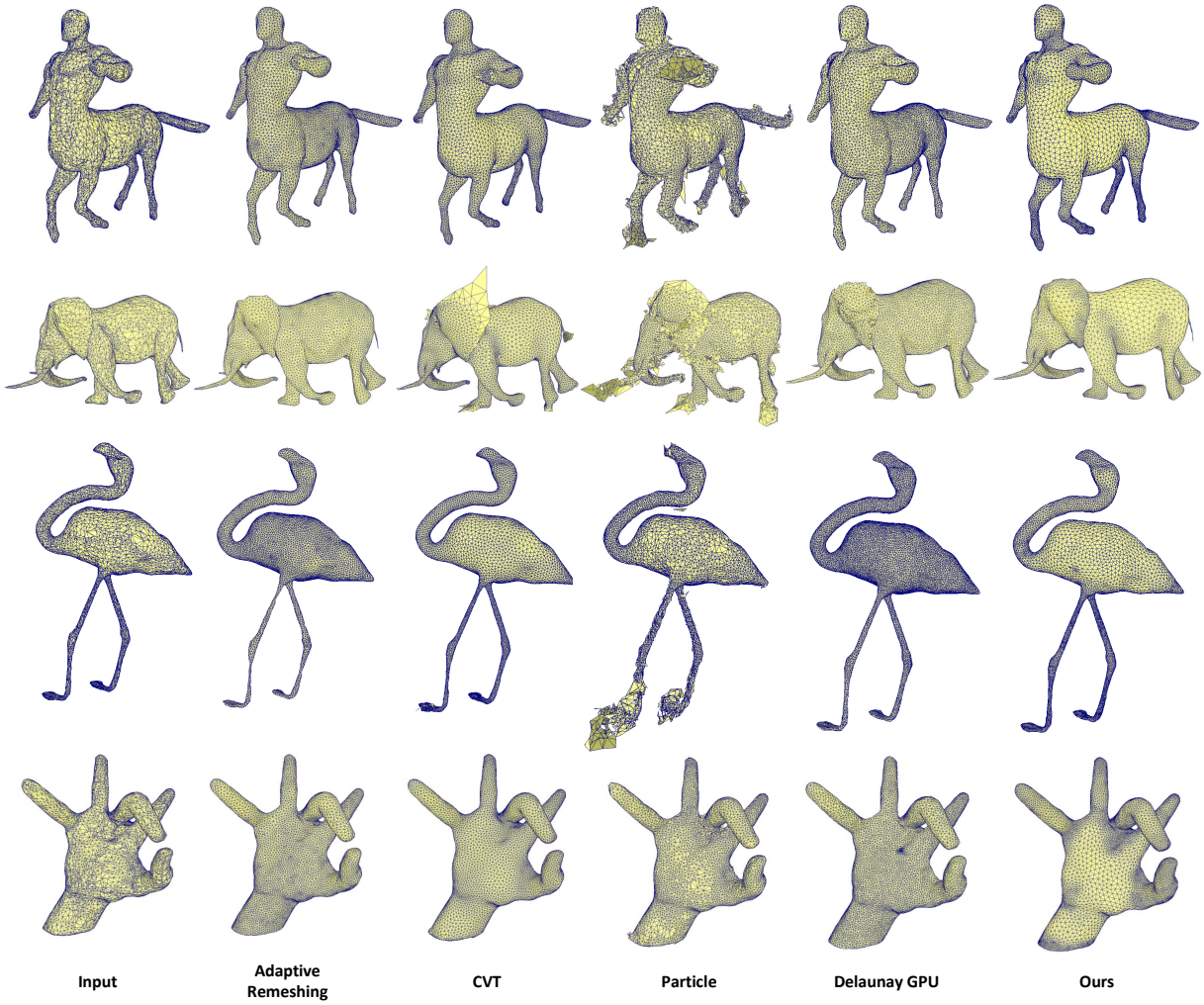


Fig. 9. Comparisons of different remeshing methods for SHREC models (9,000∼13,000 vertices).

where M_O and M_R represent the original mesh and the remeshing result, and the Hausdorff distance H between M_O and M_R is computed from the maximum distance d between vertices and meshes. To achieve more accurate H, the distance d is computed by the MLS surface mapping [50]. The Hausdorff distance provides the maximum error measurement between two meshes. We also add the mean distance measurement to evaluate the average error to be a supplement:

$$M(M_O, M_R) = \frac{1}{n}\sum_{j=1}^{n} d(q_j, M_O), q_j \in M_R, \tag{8}$$

where M is the mean distance between M_O and M_R, n is the number of points of M_R. The Hausdorff and mean distances are used to quantitatively describe the geometrical consistency between the meshes. We perform a quantitative analysis of geometric consistency for different remeshing methods, including adaptive remeshing [13], centroidal Voronoi tessellation(CVT) [49], particle-based resampling [43], and Delaunay-GPU [28]. The adaptive remeshing balances the isotropic property and curvature sensitivity in reconnecting the original mesh. It is an extension of isotropic remeshing [23]. The CVT is a widely used solution for mesh reconstruction and remeshing. It optimizes Voronoi cells of a mesh to obtain the isotropic property. By contrast, the particle-based resampling achieves a similar function by adjusting points' positions directly.

To perform a fair comparison, we let different remeshing methods output a similar number of vertices (i.e., in a range of 9,000∼13,000 points). In Figures 9 and 10, we display remeshing results generated by different methods. The Hausdorff and mean distances are reported in Tables 1 and 3. The label "Nan" represents the illegal value that is produced by a program crash or a significant error result. Specifically, our method obtains smaller Hausdorff and mean distances for 70% test meshes. The reason is that our method keeps more points in their original positions and maps the new points into the mesh with lower MLS errors. It avoids extreme displacements for original points. Therefore, our method achieves better geometric consistency.

5.2 Mesh Quality

The quality of the remeshing result can be evaluated by a triangle quality measurement function $Q(t) \in [0, 1]$ for triangle t:

$$Q(t) = \frac{6}{\sqrt{3}} \cdot \frac{S_t}{p_t h_t} \tag{9}$$

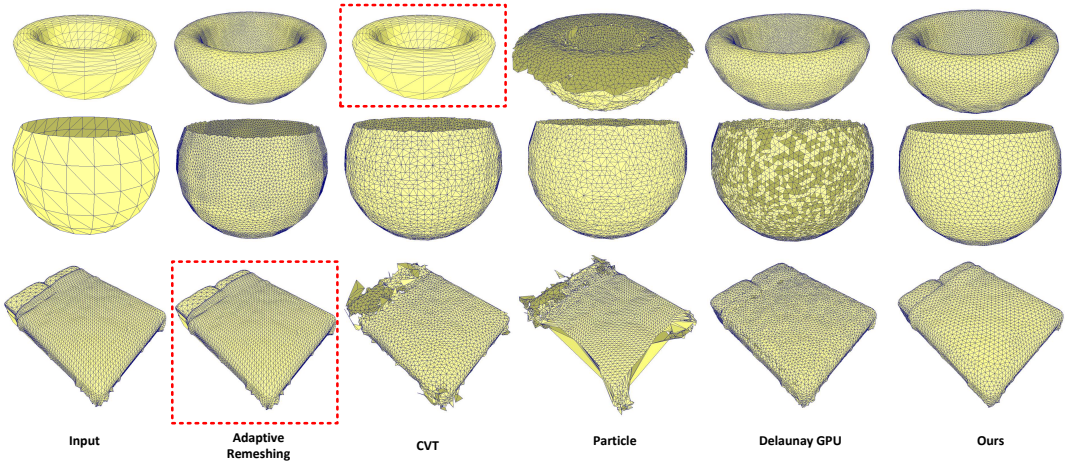


Fig. 10. Comparisons of different remeshing methods for ModelNet models (9,000∼13,000 vertices). Red labels mean that the related method crashed during execution.

TABLE 1
Geometric consistency measurements (Hd and Md) of different methods for SHREC models.

Methods / Models\Measurement	Adaptive Remeshing		CVT		Particle		Delaunay-GPU		Our	
	Hd	Md	Hd	Md	Hd	Md	Hd	Md	Hd	Md
centaur	0.0101	0.00056	0.2594	0.00596	0.2519	0.16774	0.01832	**0.00025**	**0.0078**	0.00042
crane	0.0139	0.00047	0.2296	0.00389	0.2263	0.06133	0.01755	0.00133	**0.0046**	**0.00025**
elephant	0.0197	0.00094	0.1259	0.04261	0.1265	0.04334	0.06601	**0.00055**	**0.0165**	0.00094
hand	0.0105	0.00047	0.1911	0.01094	0.1916	0.01372	0.02035	**0.00025**	**0.0088**	0.00037
tyrannosaurus	0.0082	0.00035	0.3166	0.00328	0.2872	0.02012	0.01301	0.00094	**0.0061**	**0.00028**
ant	0.0122	0.00051	0.1656	0.00738	0.1738	0.01554	Nan	0.06352	**0.0116**	**0.00049**
bird	0.0104	0.00042	0.2426	0.00503	0.2415	0.01699	Nan	0.06981	**0.0104**	**0.00034**
crocodile	0.0235	0.00073	0.2245	0.07449	0.2081	0.00895	Nan	0.05484	**0.0119**	**0.00071**
light	0.0136	0.00144	0.2701	0.11083	0.2701	0.02982	Nan	0.05495	**0.0108**	**0.00065**
spider	0.0097	0.00055	0.1871	0.00745	0.1797	0.01429	Nan	0.05447	**0.0083**	**0.00045**

TABLE 2
Mesh quality measurements (Q_{avg} and Q_{min}) of different methods for SHREC models.

Methods / Models\Measurement	Adaptive Remeshing		CVT		Particle		Delaunay-GPU		Our	
	Q_{avg}	Q_{min}	Q_{avg}	Q_{min}	Q_{avg}	Q_{min}	Q_{avg}	Q_{min}	Q_{avg}	Q_{min}
centaur	0.85	Nan	0.84	0.33	0.73	Nan	0.82	0.47	**0.85**	0.51
crane	**0.87**	Nan	0.84	0.15	0.71	Nan	0.82	0.48	0.84	0.49
elephant	0.82	Nan	0.82	0.17	0.74	Nan	0.82	0.47	**0.83**	0.46
hand	0.84	Nan	0.85	0.28	0.73	Nan	0.82	0.47	**0.86**	0.51
tyrannosaurus	**0.86**	Nan	0.85	0.33	0.75	Nan	0.82	0.48	0.85	0.45
ant	0.84	Nan	**0.85**	0.31	0.68	Nan	0.82	0.47	0.84	0.52
bird	0.81	Nan	**0.85**	0.29	0.7	0.14	0.82	0.47	0.82	0.49
crocodile	0.82	Nan	0.84	0.16	0.67	0.11	0.82	0.48	**0.85**	0.47
light	**0.87**	Nan	0.82	0.14	0.71	Nan	0.82	0.47	0.84	0.51
spider	0.84	Nan	0.85	0.33	0.71	0.11	0.82	0.49	**0.86**	0.52

where S_t is the area of triangle t, p_t is half of the perimeter of t, and h_t is the length of the longest edge length [7]. It provides a quantitative analysis of the isotropic property. Hence $Q(t) = 1$ means that the triangle is equilateral. We compute the average value Q_{ave} and minimum value Q_{min} over all triangles for different remeshing methods, which are reported in Tables 2 and 4. It is observed that the adaptive method could produce incorrect edges in regions with sharp curvature changes. For some models, the remeshing process of the adaptive remeshing method may even crash. For CVT and Delaunay-GPU remeshing methods, Voronoi cells are optimized and the mesh quality is improved theoretically. However, they do not perform stably for mesh models with sharp curvature changes and extreme obtuse triangles, in which incorrect cell optimization may occur. Our method provides a feasible solution consistently. It achieves a balance between isotropic property and geometric feature preservation. Our method obtains the best $Q(t)$ values for 53% test meshes, while the Delaunay-GPU [28] (the state-of-the-art) achieves the best $Q(t)$ values for only 22% test meshes. Figure 11 uses color maps to display the minimum inner angle for the results generated by the state-of-the-art and our method. The color changes from red to blue in our remeshing results are smoother, which means that

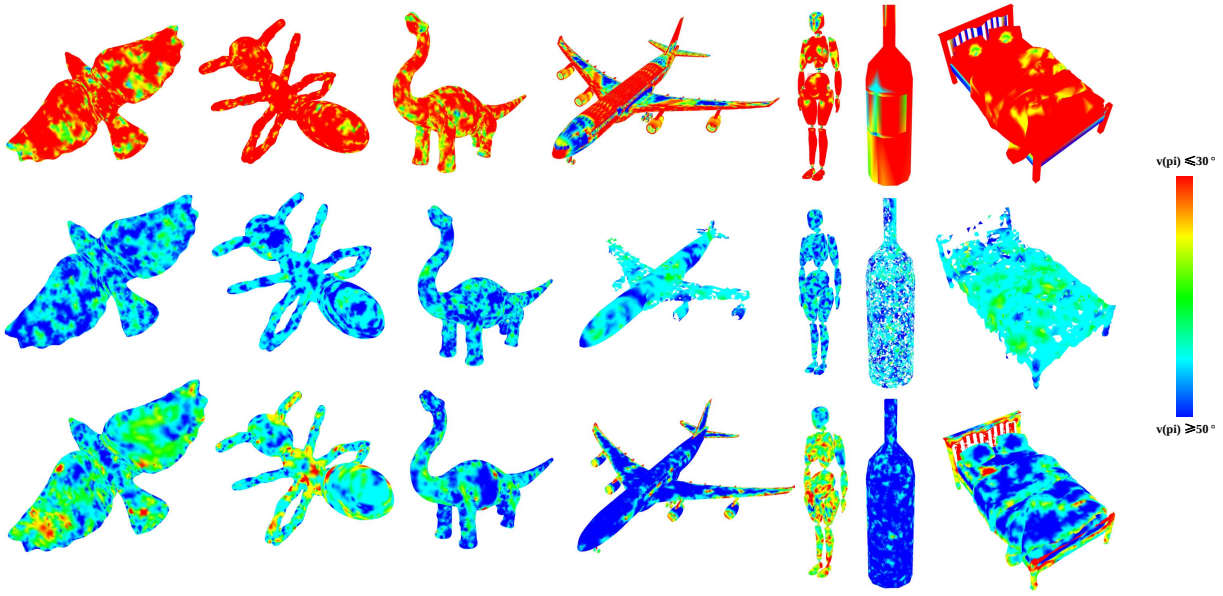


Fig. 11. Color maps of remeshing results for mesh quality visualization. First row: original meshes; second row: Delaunay-GPU remeshing results; third row: our remeshing results. More geometric details are kept and isotropic property is improved by our method.

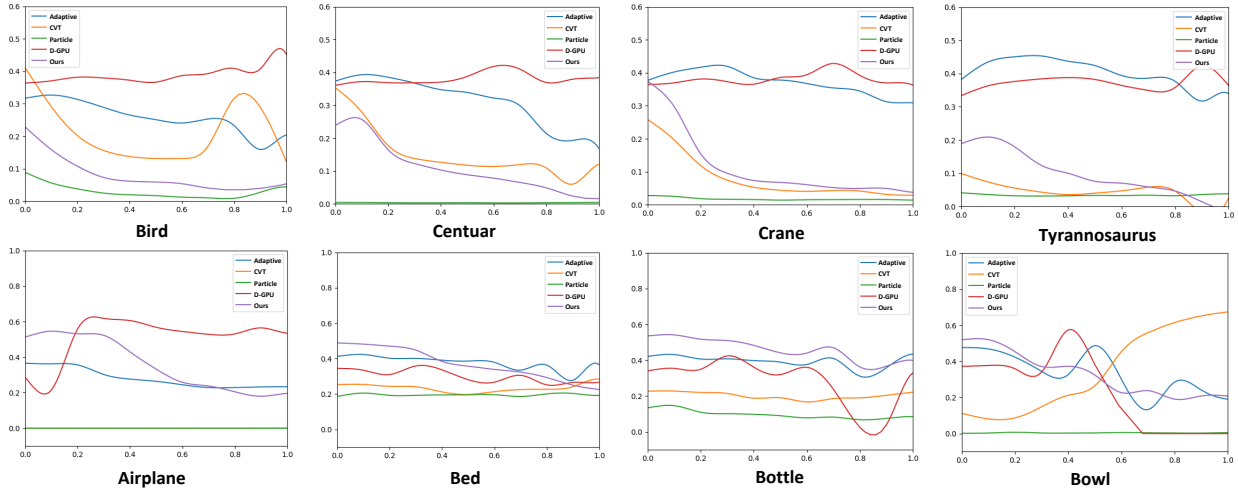


Fig. 12. Some curve-based charts of remeshing results. X axis: curvature values computed by Equation 1; Y axis: triangle area. Adaptive: adaptive remeshing [13]; CVT: centroidal Voronoi tessellation [49]; Particle: particle-based resampling [43]; D-GPU: Delaunay-GPU [28].

our method achieves a better balance between the isotropic property and geometric feature keeping. Conversely, the Delaunay-GPU method [28] loses many geometric details and fails to obtain curvature adaptation in the remeshing results.

5.3 Curvature Sensitivity

To analyze the performance of adaptive remeshing with respect to curvature, we compute a curve-based chart to show the relationship between triangle areas and curvature values of the remeshing results. Ideal curvature-based remeshing is expected to have the property that the area of triangles should decrease monotonically with the increasing curvature. That is, more vertices are distributed in regions with high curvature values. To compare the monotonicity for different methods in the chart, we normalize triangle areas from $[A_{min}, A_{max}]$ into $[0, 1]$, where A_{min} and A_{max} represent the minimal and maximal triangle areas of the mesh, respectively. The curves displayed in Figure 12 visualize such monotonicity behavior of different methods applied to SHREC and ModelNet models. If a curve does not exhibit a monotonic shape, it implies that the corresponding remeshing method is not curvature sensitive. From the figure, we can see that the curves generated by our method have better monotonicity, which means that our method achieves better curvature sensitivity.

To estimate the performance of geometric feature preservation, we compare our method and adaptive remeshing [13]. Some examples are shown in Figure 13. It is clear that our method achieves a better balance between the isotropic behavior and geometric feature preservation with a similar target number of vertices, even though the two

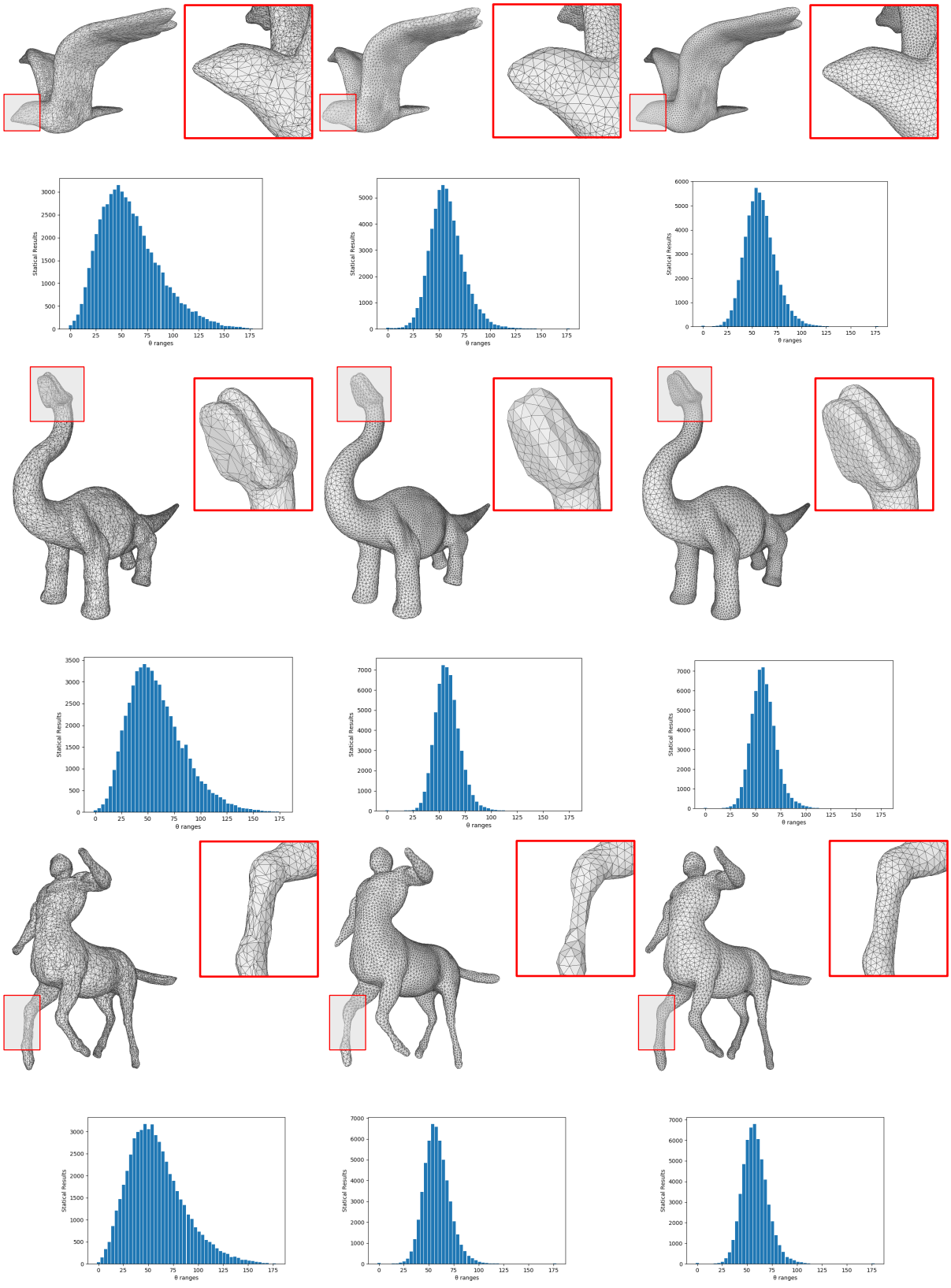


Fig. 13. Comparisons of adaptive remeshing [13] and our method. The histograms represent the interior angle distributions of meshes. With similar target numbers of vertices and faces, our remeshing results keep geometric features better.

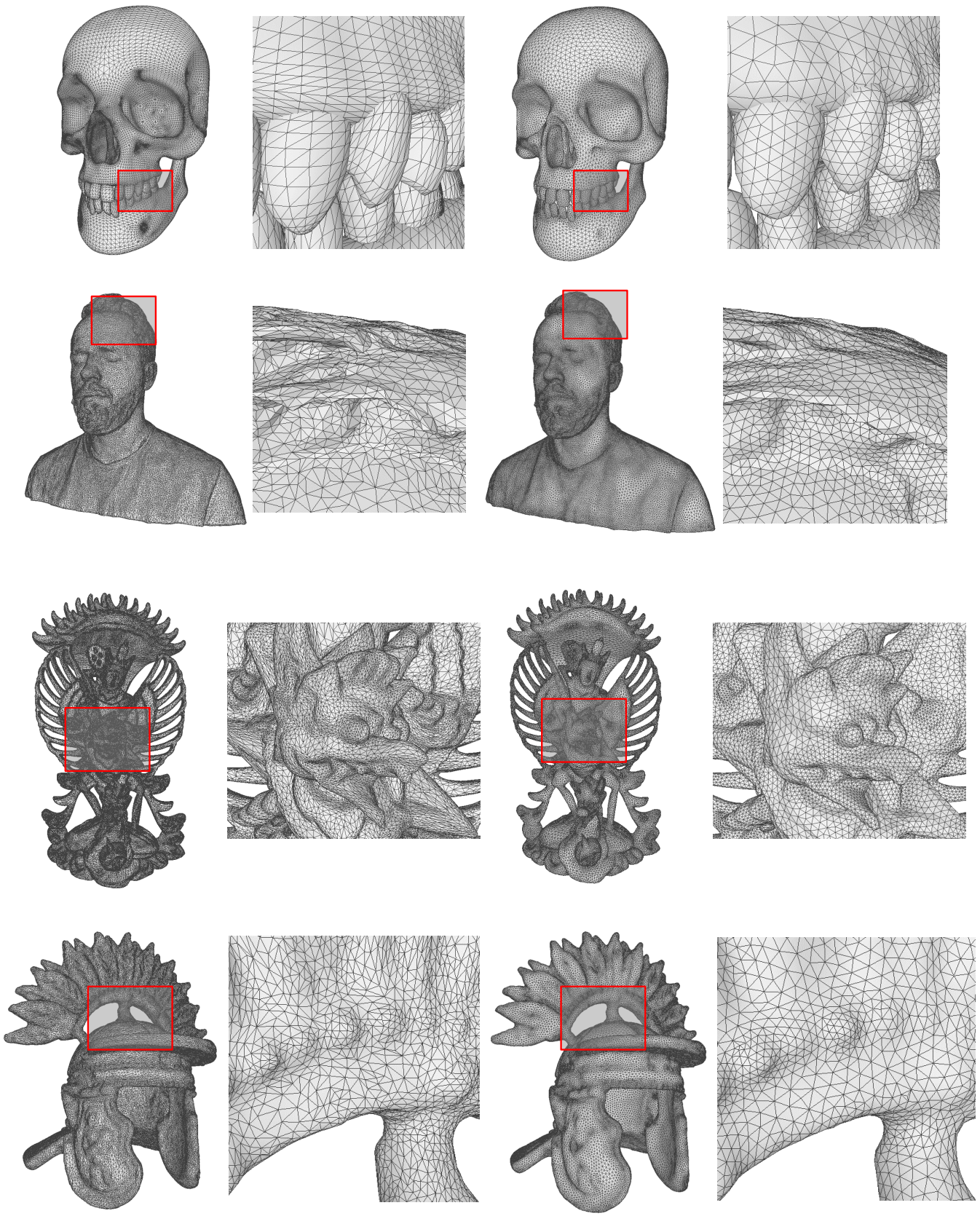


Fig. 14. More examples. Left: original meshes; right: results by our remeshing methods.

TABLE 3
Geometric consistency measurements (Hd and Md) of different methods for ModelNet models.

Methods / Models\Measurements	Adaptive Remeshing		CVT		Particle		Delaunay-GPU		Our	
	Hd	Md	Hd	Md	Hd	Md	Hd	Md	Hd	Md
bed	0.52825	0.01261	0.57317	0.07459	0.52917	**0.00554**	**0.30601**	0.04294	0.3133	0.00841
bottle	0.28981	0.00053	0.01666	0.00049	0.03771	0.00067	0.01427	**0.00024**	**0.0128**	0.00041
bathtub	**0.26766**	0.00249	0.42598	0.08404	0.32563	**0.00233**	Nan	Nan	0.2688	0.00285
bowl	0.25844	0.00394	0.26026	0.00961	0.25958	0.00539	0.25666	0.00672	**0.1038**	**0.00461**
car	0.22766	0.00351	0.50472	0.03316	0.34477	0.00817	Nan	Nan	**0.2271**	**0.00334**
airplane	0.04249	0.00096	0.39792	0.11671	0.06676	**0.00043**	0.30751	0.05903	**0.0147**	0.00073
person	0.02533	0.00165	0.05651	0.00185	0.43426	0.00444	0.02362	**0.00042**	**0.0211**	0.00138
sofa	**0.03021**	0.00141	0.27405	0.06759	0.21571	0.00763	Nan	Nan	0.0975	**0.00119**

TABLE 4
Mesh quality measurements (Q_{avg} and Q_{min}) of the remeshing results for ModelNet models.

Methods / Models\Measurement	Adaptive Remeshing		CVT		Particle		Delaunay-GPU		Our	
	Q_{avg}	Q_{min}	Q_{avg}	Q_{min}	Q_{avg}	Q_{min}	Q_{avg}	Q_{min}	Q_{avg}	Q_{min}
bed	Nan	Nan	0.76	0.31	0.72	Nan	0.82	0.48	**0.83**	0.23
bottle	0.89	0.25	0.85	0.26	0.74	Nan	0.82	0.47	**0.91**	0.11
bathtub	Nan	Nan	**0.82**	0.15	0.77	Nan	Nan	Nan	0.81	0.26
bowl	0.88	0.21	Nan	0.24	0.74	Nan	0.82	0.48	**0.91**	0.59
car	Nan	Nan	**0.84**	Nan	0.76	Nan	Nan	Nan	0.83	0.32
airplane	Nan	Nan	0.82	Nan	Nan	Nan	**0.82**	0.48	0.71	0.29
person	**0.84**	Nan	0.81	Nan	0.76	Nan	0.82	0.48	0.76	0.32
sofa	0.83	Nan	0.83	Nan	0.76	Nan	Nan	Nan	**0.84**	0.16

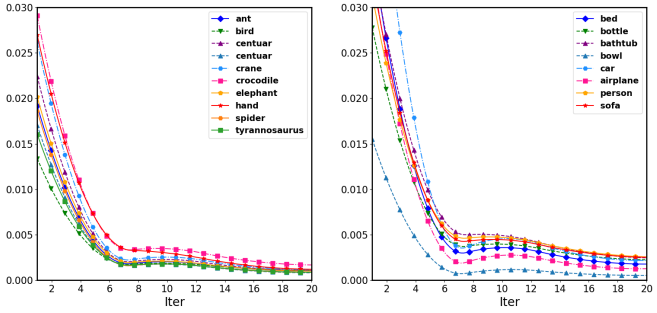


Fig. 15. Convergence curves of the smooth processing for generating CSF for different models. Left picture: results in SHREC models; right picture: results in ModelNet models.

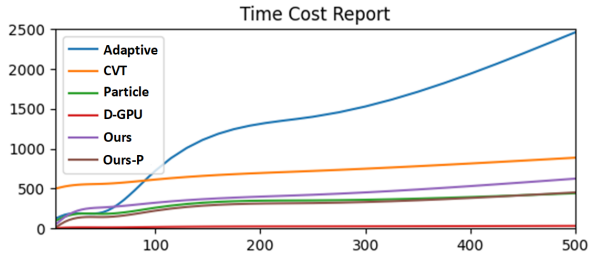


Fig. 16. Run time chart for different methods. Y axis represents time (seconds); X axis is the number of input points (1k). Adaptive: adaptive remeshing [13]; CVT: centroidal Voronoi tessellation [49]; Particle: particle-based resampling [43]; D-GPU: Delaunay-GPU [28]; Ours: our adaptively isotropic remeshing without parallel acceleration; Ours-P: our adaptively isotropic remeshing with parallel acceleration.

methods share a similar edge reconnection strategy. This is benefited of the CSF.

5.4 Comprehensive Analysis

We have used different metrics to show the performance of our method, including Hausdorff and mean distances for geometric consistency, triangle quality measurement for mesh quality, and curve monotonicity for curvature sensitivity. All results show that our remeshing method generally achieves better performance. More complicated instances are depicted in Figure 14, which further demonstrate the performance of our method.

Figure 15 shows the convergence curves of the smooth processing for generating the CSF for SHREC and ModelNet models. To analyze the convergence, we compute the average value $\overline{\triangle s_i}$ of $\triangle s_i$ of Equation 4. It can be seen that most test models can achieve $\overline{\triangle s_i} < 0.005$ after 8 iterations.

In Figure 16, we show a time cost chart for different methods, in which the number of vertices in the remeshing results is controlled in 9000∼13000). The Delaunay-GPU achieves the fastest speed for remeshing tasks. The reason is that it utilizes GPU-based parallel computing to improve the Delaunay triangulation. However, the isotropic property and geometric consistency decrease to a certain extent. In contrast, our method can achieve a better balance.

Although our method has advantages in adaptively isotropic remeshing, some limitations exist. First, for some manually edited meshes, incorrect connections break the manifold property cannot be completely avoided, though the sub-remeshing with default grid repair functions reduces the probability of its occurrence. In Figure 17, we show a few failure examples of our method. Second, the proposed remeshing method cannot provide real time performance. According to the statistics of the runtime report,

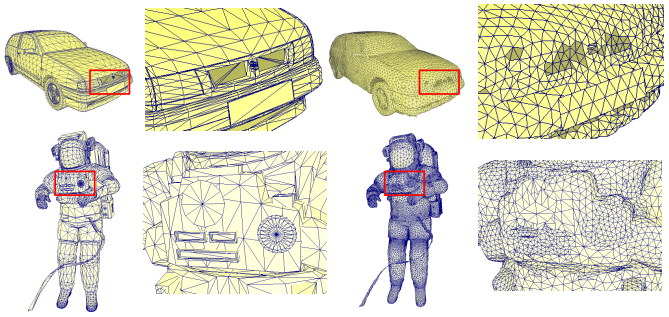


Fig. 17. Some failure examples of our remeshing method. The manifold property is broken in some regions where incorrect connections occur. Left: original meshes; right: remeshing results.

TABLE 5
Runtime statistics of our method with different numbers of input and output points. The top row represents the number of input points (5k, 10k, 50k, 100k). The left column represents the number of output points (1k, 5k, 10k, 50k).

	5k	10k	50k	100k
1k	26.2s	34.06s	100.96s	150.36s
5k	32.5s	52.256s	161.68s	189.63s
10k	–	61.88s	200.68s	249.36s
50k	–	–	>5min	>5min

more time costs are required for edge reconnection and vertex list updating. Table 5 shows our method's runtime statistics with different input and output points.

6 Conclusion

We propose an adaptively isotropic remeshing method to improve the mesh quality while keeping important geometric features. To keep geometric features, we construct a curvature smoothed field (CSF) to represent the curvature variation while making the curvatures change smoothly across the whole mesh. To improve mesh quality, we propose a histogram-based edge reconnection with sub-remeshing. The isotropic property can be reconstructed even for a mesh with extremely distorted triangular faces. Based on the CSF, our framework is able to obtain adaptively isotropic remeshing results efficiently. Experiments show that our method achieves better performance in several evaluation metrics, including geometric consistency, mesh quality measurement, and curvature sensitivity estimation. Compared to the traditional edge reconnection strategies, our method achieves a better balance between mesh quality improvement and geometric feature preservation.

Acknowledgments

This research is supported by the Ministry of Education, Singapore, under its Tier-2 Fund MOE2016-T2-2-057(S).

Appendix

Here we analyze the convergence of the smooth processing for the CSF generation.

Let I_{n-2} be the identity matrix of size $(n-2)\times(n-2)$ and

$$A_n = \begin{bmatrix} 0 & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & 0 & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & 0 \end{bmatrix} \tag{10}$$

where $a_{ij} = \dfrac{w_{ij}}{w_i}$ for $j \in R_i$; otherwise, $a_{ij} = 0$; and w_i, w_{ij}, R_i are defined in Equations 2 and 3.

Without loss of generality, we assume that v_{max} and v_{min} mentioned in Section 3 are in the last two entries in the vertex list of the mesh. Let A_{n-2} be the matrix of size $(n-2)\times(n-2)$ obtained from A_n by removing the last two columns and the last two rows. Then the iteration of the smooth processing defined in Equation 4 can be written

$$\begin{bmatrix} s_1^{(k+1)} \\ \vdots \\ s_{n-2}^{(k+1)} \end{bmatrix} = (I_{n-2} + (1-\mu)(A_{n-2} - I_{n-2})) \begin{bmatrix} s_1^{(k)} \\ \vdots \\ s_{n-2}^{(k)} \end{bmatrix}$$

$$+B = (\mu I_{n-2} + (1-\mu)A_{n-2}) \begin{bmatrix} s_1^{(k)} \\ \vdots \\ s_{n-2}^{(k)} \end{bmatrix} + B$$

where the superscript 'k' represents the k-th iteration, and B is a vector:

$$B = (1-\mu) \begin{bmatrix} a_{1,(n-1)} & a_{1n} \\ a_{2,(n-1)} & a_{2n} \\ \vdots & \vdots \\ a_{(n-2),(n-1)} & a_{(n-2),n} \end{bmatrix} \begin{bmatrix} s_{n-1} \\ s_n \end{bmatrix}. \tag{11}$$

Let $G = [g_{ij}] = \mu I_{n-2} + (1-\mu)A_{n-2}$. Then G satisfies the following properties:

(1) $\sum_{j=1}^{n-2} |g_{ij}| = \mu + (1-\mu)\sum_{j=1}^{n-2} a_{ij} \le 1$ for any $i \in \{1, 2, \cdots, n-2\}$.
(2) G is irreducible, which means that for any $i, j \in \{1, \cdots, n-2\}$ there exists a chain of indices $i_1, i_2, \cdots, i_k$ such that $A_{i,i_1} \neq 0, \cdots, A_{i_{k-1},i_k} \neq 0$ and $A_{i_k,j} \neq 0$. This is because the model we are considering is a mesh with only a single component.
(3) There exists at least one $h \in \{1, \cdots, n-2\}$ such that $\sum_{j=1}^{n-2} |g_{hj}| = \mu + (1-\mu)\sum_{j=1}^{n-2} a_{hj} < \mu + (1-\mu) = 1$. This is due to the fact that v_{max} and v_{min} have at least two adjacent vertices, implying that there are some non-zero $a_{h,n-1}$ or $a_{h,n}$.

Now consider matrix $\lambda I_{n-2} - G = (\lambda-\mu)I_{n-2} - (1-\mu)A_{n-2}$. It is noted that for $\lambda \neq \mu$, matrices G and $\lambda I_{n-2} - G$ have zero elements in exactly the same locations, which tells that $\lambda I_{n-2} - G$ is also irreducible.

Moreover, for row i of matrix $(\lambda-\mu)I_{n-2} - (1-\mu)A_{n-2}$, the diagonal entry has

$$|(\lambda-\mu)| = |(\lambda-1) + (1-\mu)| \ge 1-\mu$$

for $|\lambda| \ge 1$, and the other entries have

$$|(1-\mu)\sum_{j\neq i} a_{ij}| = (1-\mu)\sum_{j\neq i} a_{ij} \le 1-\mu.$$

There exists an h such that

$$|(1-\mu)\sum_{j\neq h} a_{hj}| = (1-\mu)\sum_{j\neq h} a_{hj} < 1-\mu.$$

Therefore for $|\lambda| \geq 1$, matrix $\lambda I_{n-2} - G$ is diagonally dominant (by rows).

It thus follows from [51] that for $|\lambda| \geq 1$, matrix $\lambda I_{n-2} - G$ is nonsingular since it is irreducible and diagonally dominant.

Note that the eigenvalues λ of matrix G are all and only the roots of $\det(\lambda I_{n-2} - G) = 0$. Since $\lambda I_{n-2} - G$ is nonsingular for $|\lambda| \geq 1$, all of the eigenvalues of G are in the interior of the unit circle. Hence the spectral radius of G is less than 1, which concludes that the iteration of the smooth processing defined in Equation 4 converges.

REFERENCES

[1] K. Crane, C. Weischedel, and M. Wardetzky, "Geodesics in heat: A new approach to computing distance based on heat flow," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 5, pp. 1–11, 2013.

[2] M. Jin, J. Kim, F. Luo, and X. Gu, "Discrete surface ricci flow," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 5, pp. 1030–1043, 2008.

[3] N. Lei, K. Su, L. Cui, S.-T. Yau, and X. D. Gu, "A geometric view of optimal transportation and generative model," *Computer Aided Geometric Design*, vol. 68, pp. 1–21, 2019.

[4] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas, "Functional maps: a flexible representation of maps between shapes," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, pp. 1–11, 2012.

[5] C. Lv, Z. Wu, X. Wang, M. Zhou, and K.-A. Toh, "Nasal similarity measure of 3d faces based on curve shape space," *Pattern Recognition*, vol. 88, pp. 458–469, 2019.

[6] K. Shimada, A. Yamada, and T. Itoh, "Anisotropic triangulation of parametric surfaces via close packing of ellipsoids," *Int. J. Comput. Geom. Appl.*, vol. 10, no. 4, pp. 417–440, 2000. [Online]. Available: https://doi.org/10.1142/S0218195900000243

[7] P. J. Frey and H. Borouchaki, "Surface mesh quality evaluation," *International journal for numerical methods in engineering*, vol. 45, no. 1, pp. 101–118, 1999.

[8] M. Berger, J. A. Levine, L. G. Nonato, G. Taubin, and C. T. Silva, "A benchmark for surface reconstruction," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 2, pp. 1–17, 2013.

[9] C. Lv, W. Lin, and B. Zhao, "Voxel structure-based mesh reconstruction from a 3d point cloud," *IEEE Transactions on Multimedia*, vol. 24, pp. 1815–1829, 2021.

[10] Q.-C. Xu, D.-M. Yan, W. Li, and Y.-L. Yang, "Anisotropic surface remeshing without obtuse angles," in *Computer Graphics Forum*, vol. 38, no. 7. Wiley Online Library, 2019, pp. 755–763.

[11] P. Alliez, D. Cohen-Steiner, O. Devillers, B. Lévy, and M. Desbrun, "Anisotropic polygonal remeshing," in *ACM SIGGRAPH 2003 Papers*, 2003, pp. 485–493.

[12] X. Jiao, A. Colombi, X. Ni, and J. Hart, "Anisotropic mesh adaptation for evolving triangulated surfaces," *Engineering with Computers*, vol. 26, no. 4, pp. 363–376, 2010.

[13] M. Dunyach, D. Vanderhaeghe, L. Barthe, and M. Botsch, "Adaptive remeshing for real-time mesh deformation," in *Eurographics 2013*. The Eurographics Association, 2013.

[14] C. Dapogny, C. Dobrzynski, and P. Frey, "Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems," *Journal of computational physics*, vol. 262, pp. 358–378, 2014.

[15] R. Narain, A. Samii, and J. F. O'brien, "Adaptive anisotropic remeshing for cloth simulation," *ACM transactions on graphics (TOG)*, vol. 31, no. 6, pp. 1–10, 2012.

[16] F. Dassi, A. Mola, and H. Si, "Curvature-adapted remeshing of cad surfaces," *Procedia Engineering*, vol. 82, pp. 253–265, 2014.

[17] Y. Wang, D.-M. Yan, X. Liu, C. Tang, J. Guo, X. Zhang, and P. Wonka, "Isotropic surface remeshing without large and small angles," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 7, pp. 2430–2442, 2018.

[18] F. Verhoeven, A. Vaxman, T. Hoffmann, and O. Sorkine-Hornung, "Dev2pq: Planar quadrilateral strip remeshing of developable surfaces," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 3, pp. 1–18, 2022.

[19] Z.-Y. Liu, J.-P. Su, H. Liu, C. Ye, L. Liu, and X.-M. Fu, "Error-bounded edge-based remeshing of high-order tetrahedral meshes," *Computer-Aided Design*, vol. 139, p. 103080, 2021.

[20] B. Hu, C. Ye, J.-P. Su, and L. Liu, "Manifold-constrained geometric optimization via local parameterizations," *IEEE Transactions on Visualization & Computer Graphics*, no. 01, pp. 1–1, 2021.

[21] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Lévy, *Polygon mesh processing*. CRC press, 2010.

[22] P. Alliez, E. C. De Verdire, O. Devillers, and M. Isenburg, "Isotropic surface remeshing," in *2003 Shape Modeling International.* IEEE, 2003, pp. 49–58.

[23] D.-M. Yan, B. Lévy, Y. Liu, F. Sun, and W. Wang, "Isotropic remeshing with fast and exact computation of restricted voronoi diagram," in *Computer graphics forum*, vol. 28, no. 5. Wiley Online Library, 2009, pp. 1445–1454.

[24] D.-M. Yan, J. Guo, X. Jia, X. Zhang, and P. Wonka, "Blue-noise remeshing with farthest point optimization," in *Computer Graphics Forum*, vol. 33, no. 5. Wiley Online Library, 2014, pp. 167–176.

[25] F. d. Goes, P. Memari, P. Mullen, and M. Desbrun, "Weighted triangulations for geometry processing," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 3, pp. 1–13, 2014.

[26] Y.-J. Liu, C.-X. Xu, D. Fan, and Y. He, "Efficient construction and simplification of delaunay meshes," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, pp. 1–13, 2015.

[27] Z. Ye, R. Yi, M. Yu, Y.-J. Liu, and Y. He, "Geodesic centroidal voronoi tessellations: Theories, algorithms and applications," *arXiv preprint arXiv:1907.00523*, 2019.

[28] J. Zheng and T.-S. Tan, "Computing centroidal voronoi tessellation using the gpu," in *Symposium on Interactive 3D Graphics and Games*, 2020, pp. 1–9.

[29] B. Lévy and N. Bonneel, "Variational anisotropic surface meshing with voronoi parallel linear enumeration," in *Proceedings of the 21st international meshing roundtable*. Springer, 2013, pp. 349–366.

[30] F. Dassi and H. Si, "A curvature-adapted anisotropic surface re-meshing method," in *New Challenges in Grid Generation and Adaptivity for Scientific Computing*. Springer, 2015, pp. 19–41.

[31] K. Su, N. Lei, W. Chen, L. Cui, H. Si, S. Chen, and X. Gu, "Curvature adaptive surface remeshing by sampling normal cycle," *Computer-Aided Design*, vol. 111, pp. 1–12, 2019.

[32] R. Yi, Y.-J. Liu, and Y. He, "Delaunay mesh simplification with differential evolution," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 6, pp. 1–12, 2018.

[33] W.-X. Zhang, Q. Wang, J.-P. Guo, S. Chai, L. Liu, and X.-M. Fu, "Constrained remeshing using evolutionary vertex optimization," in *Computer Graphics Forum*, vol. 41, no. 2. Wiley Online Library, 2022, pp. 237–247.

[34] C. Lv, W. Lin, and B. Zhao, "Intrinsic and isotropic resampling for 3d point clouds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–18, 2022.

[35] W. Hou, C. Zong, P. Wang, S. Xin, S. Chen, G. Liu, C. Tu, and W. Wang, "Sdf-rvd: Restricted voronoi diagram on signed distance field," *Computer-Aided Design*, vol. 144, pp. 103 166.1–103 166.12, 2022.

[36] S. Premžoe, T. Tasdizen, J. Bigler, A. Lefohn, and R. T. Whitaker, "Particle-based simulation of fluids," in *Computer Graphics Forum*, vol. 22, no. 3. Wiley Online Library, 2003, pp. 401–410.

[37] S. E. Hieber, J. H. Walther, and P. Koumoutsakos, "Remeshed smoothed particle hydrodynamics simulation of the mechanical behavior of human organs," *Technology and Health Care*, vol. 12, no. 4, pp. 305–314, 2004.

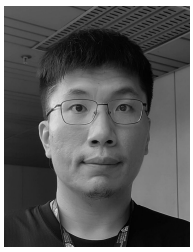
[38] M. Meyer, R. Whitaker, R. M. Kirby, C. Ledergerber, and H. Pfister, "Particle-based sampling and meshing of surfaces in multimaterial volumes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1539–1546, 2008.

[39] C. Oâ€™Sullivan, "Particle-based discrete element modeling: geomechanics perspective," *International Journal of Geomechanics*, vol. 11, no. 6, pp. 449–464, 2011.

[40] K. Takamatsu and T. Kanai, "A fast and practical method for animating particle-based viscoelastic fluids," *International Journal of Virtual Reality*, vol. 10, no. 1, pp. 29–35, 2011.

[41] Z. Zhong, X. Guo, W. Wang, B. Lévy, F. Sun, Y. Liu, W. Mao *et al.*, "Particle-based anisotropic surface meshing." *ACM Trans. Graph.*, vol. 32, no. 4, pp. 99–1, 2013.

[42] X.-X. Cheng, X.-M. Fu, C. Zhang, and S. Chai, "Practical error-bounded remeshing by adaptive refinement," *Computers & Graphics*, vol. 82, pp. 163–173, 2019.
[43] S. Zhong, Z. Zhong, and J. Hua, "Surface reconstruction by parallel and unified particle-based resampling from point clouds," *Computer Aided Geometric Design*, vol. 71, pp. 43–62, 2019.
[44] M. Botsch and L. Kobbelt, "A remeshing approach to multiresolution modeling," in *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, 2004, pp. 185–192.
[45] Y. Guo, J. Wang, H. Sun, X. Cui, and Q. Peng, "A novel constrained texture mapping method based on harmonic map," *Computers & Graphics*, vol. 29, no. 6, pp. 972–979, 2005.
[46] C. Lv, Z. Wu, X. Wang, and M. Zhou, "Constructing 3d facial hierarchical structure based on surface measurements," *Multimedia Tools and Applications*, vol. 78, no. 11, pp. 14 753–14 776, 2019.
[47] A. Bronstein, M. Bronstein, U. Castellani, B. Falcidieno, A. Fusiello, A. Godil, L. Guibas, I. Kokkinos, Z. Lian, M. Ovsjanikov *et al.*, "Shrec 2010: robust large-scale shape retrieval benchmark," *Proc. 3DOR*, vol. 5, no. 4, 2010.
[48] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.
[49] Z. Chen, T. Zhang, J. Cao, Y. J. Zhang, and C. Wang, "Point cloud resampling using centroidal voronoi tessellation methods," *Computer-Aided Design*, vol. 102, pp. 12–21, 2018.
[50] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva, "Point set surfaces," in *Proceedings Visualization, 2001. VIS'01.* IEEE, 2001, pp. 21–29.
[51] R. Bagnara, "A unified proof for the convergence of jacobi and gauss-seidel methods," *SIAM Rev.*, vol. 37, no. 1, pp. 93–97, 1995. [Online]. Available: https://doi.org/10.1137/1037008



Chenlei Lv received PhD degree in College of information science and technology, Beijing Normal University (BNU). He is currently a Postdoctor in School of Computer Science and Engineering, Nanyang Technological University (NTU). His research interests include Computer Vision, 3D Biometrics, Computer Graphics, etc. He has published several papers in IEEE TPAMI, IEEE TIP, IEEE TMM, PR, ACM TOMM, etc. He has also served as reviewer for IEEE TMM, PR, IEEE TCSVT, NEUROCOMPUTING, etc. The personal page of the link is: https://aliexken.github.io/.

Weisi Lin (M'92–SM'98–F'16) received the Ph.D. degree from King's College London, U.K. He is currently a Professor with the School of Computer Science and Engineering, Nanyang Technological University. His research interests include image processing, perceptual signal modeling, video compression, and multimedia communication, in which he has published over 200 journal papers, over 230 conference papers, filed seven patents, and authored two books. He is a fellow of the IET and an Honorary Fellow of the Singapore Institute of Engineering Technologists. He was the Technical Program Chair of the IEEE ICME 2013, PCM 2012, QoMEX 2014, and the IEEE VCIP 2017. He has been an Invited/Panelist/Keynote/Tutorial Speaker at over 20 international conferences and was a Distinguished Lecturer of the IEEE Circuits and Systems Society from 2016 to 2017 and the Asia–Pacific Signal and Information Processing Association (APSIPA) from 2012 to 2013. He has been an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON MULTIMEDIA, and the IEEE SIGNAL PROCESSING LETTERS.

Jianmin Zheng received the BS and PhD degrees from Zhejiang University, China. He is a professor with the School of Computer Science and Engineering at Nanyang Technological University, Singapore. His current research interests include T-spline technologies, digital geometric processing, reality computing, AR/VR, and AI assisted part design for 3D printing. He is currently the programme director for the research pillar of ML/AI under the HP-NTU Digital Manufacturing Corporate Lab. He is also a member of executive committee of the AsiaGraphics Association.