Isotropic Remeshing with Inter-angle Optimization

Hanbing Zheng and Chenlei Lv(✉)

College of Computer Science and Software Engineering, Shenzhen University,
Shenzhen, China
chenleilv@mail.bnu.edu.cn

Abstract. As an important metric for mesh quality evaluation, the isotropy property holds significant value for applications such as texture UV-mapping, physical simulation, and discrete geometric analysis. Classical isotropy remeshing methods adjust vertices and edge lengths, which exhibit certain limitations in terms of input data sensitivity, geometric consistency control, and convergence speed. In this paper, we propose an improved isotropy remeshing solution with inter-angle optimization during mesh editing to enhance shape control capability and accelerate convergence. The advantage of the solution lies in its ability to predict the impact of edge length adjustments on subsequent optimization by monitoring angle transformations. It avoids inefficient editing that may cause performance fluctuations, thereby improving efficiency. Experiments demonstrate that the proposed method effectively improves the overall efficiency of mesh optimization. (The code has been released at Isotropic-Remeshing-InterAngle.

Keywords: Triangular Mesh · Isotropic Remeshing · Inter-Angle Optimization

1 Introduction

As a classical 3D data representation, triangular mesh models have been widely employed in various applications, including facial recognition [22,23], digital infrastructure [11,30], gaming&film production, and geographic information systems. Due to the excellent computability, adaptability to physical properties, and clear data structure, mainstream 3D geometry engines universally adopt triangular meshes as the standard data format for storage and computation. The initial triangular mesh is constructed either through shape reconstruction methods [15,19] or manual editing in CAD software. From the perspective of discrete differential geometry, shapes of triangular facets within the mesh exhibit randomness and often fail to satisfy the Delaunay condition [29], which is unfavorable for subsequent numerical computations. Therefore, isotropic remeshing methods are proposed for raw mesh optimization.

Supported by the National Key R&D Program of China (2024YFB3908500, 2024YFB3908503), Guangdong Basic and Applied Basic Research Foundation (2023A1515110292), Scientific Foundation for Youth Scholars of Shenzhen University.

Z. Lin et al. (Eds.): ICIG 2025, LNCS 16161, pp. 535–546, 2026.
https://doi.org/10.1007/978-981-95-3398-5_43

The concept of isotropic remeshing was proposed as early as the 1990s [3]. The primary objective is to transform an unconstrained triangular mesh (anisotropic) into a new one composed of approximately equilateral triangles (isotropic) while preserving geometric consistency. There are two technical solutions for the target, including Centroidal Voronoi Tessellation (CVT) [16] and four-steps strategy [4]. CVT iteratively adjusts vertex positions to uniform vertex-based Voronoi cells by employing Lloyd's relaxation. Limitations include sensitivity to initial mesh, slow convergence rates, and high computational coupling. Four-steps strategy is to reconnect edges between vertexes with position adjustment. It contains four basic operations (split, collapse, flip, and tangent smoothing) to efficiently adjust vertex-based distances. Intuitively, edge reconnection appears more efficient, but the original scheme still exhibits deficiencies in result stability and convergence efficiency due to insufficient consideration of shape affection.

In this paper, we propose a new isotropic remeshing method based on four-steps strategy. We redesign shape constraints with inter-angle optimization, which ensures better shape controlling in edge reconnection and prevents inefficient editing operations. The inter-angle optimization enhances the efficiency of edge-based operations during each iteration while mitigating cross-interference between different edge operations. To better preserve geometric consistency, we additionally employ point cloud up-sampling with Moving Least Squares (MLS) [1] remapping, ensuring adjusted vertices remain constrained to the original mesh-defined surface. Contributions can be summarized as:

- We present an inter-angle optimization scheme to control basic operations in the isotropic remeshing. It is useful to enhance convergence efficiency while mitigating aggressive shape deformation, thereby maintaining superior geometric consistency.
- We propose a MLS-based solution for geometric consistency keeping. The proposed solution eliminates the need for tangent smoothing, preventing vertex optimization from deviating from the MLS surface. Particularly for sparse meshes, it guarantees the stability for the vertex adjustment.

2 Related Works

As mentioned before, there are two technical solutions for isotropic remeshing, including CVT-based and four-steps remeshing.

CVT-Based Remeshing. Such solutions employ the Voronoi Diagram or Delaunay triangulation to implement remeshing. Representative methods include Restricted Voronoi Diagram (RVD) [27], improved RVD [26], weighted diagram optimization [10], intrinsic Delaunay remeshing [18], geodesic-based CVT [28], parallel CVT [31], and RVD with signed distance field (SDF) [12]. Such methods enhance the original CVT scheme by incorporating intrinsic metrics to suppress geometric artifacts induced by tangent-space displacements, thereby improving the stability of vertex neighborhood structures. However, such improvements inevitably introduce computational overhead, which compromises overall efficiency. Parallel optimization presents a potential solution, yet the inherently high coupling within CVT inevitably compromises isotropic quality.

Four-Steps Remeshing. Edge reconnection-based or four-steps remeshing implement mesh optimization based on four basic operations, including split, collapse, flip, and tangent smoothing. Such operations directly modify edge lengths and vertex degrees, which are more efficient than CVT-based remeshing. Representative methods include principal direction field-based remeshing [2], dynamic surface remeshing [14], adaptively isotropic remeshing [9,21], implicit domain-based remeshing [7], manifold-constrained remeshing [13], and intrinsic&isotropic remeshing [20]. In practice, some researchers have found that introducing shape control mechanisms into the four-steps strategy is essential for performance improvement. Wang *et al.* [24] proposed an isotropic remeshing method by removing large and small angles. Xu *et al.* [25] designed a new edge reconnection strategy to reduce the occurrence of obtuse angles. It does not pursue the strictly isotropic property. Lv *et al.* [21] employs angle-based constraints to judge the implementation of some edge editing operations. Such studies have established preliminary strategies for inter-angle optimization, which inspire our more comprehensive solution.

3 Background

As mentioned in early four-steps scheme [4], basic operations for isotropic remeshing include split, collapse, flip, and tangent smoothing, which uniform edge length. Defining a regular edge length l, the split inserts a new vertex for a long edge ($>4/3l$) and the collapse merges a short edge ($<4/5l$). The flip remove a edge for two vertexes and add a new edge between their common neighbors, which optimizes the vertex-based degree. The tangent smoothing is to adjust vertex positions to achieve accurate isotropic property.

Iteratively processing the four operations can implement the isotropic remeshing for a raw mesh. However, in the original implementation, coupling effects between the four operations are not taken into consideration. For instance, when performing collapse operations, the change in vertex degrees may increase the computational load for flips. Therefore, we aim to achieve more efficient control through inter-angle optimization at each step, thereby reducing mutual interference and enhancing the algorithm's convergence performance.

4 Methodology

We re-design basic operations (split, collapse, and flip) with more reasonable inter-angle optimization for triangular faces. The primary objective is to introduce more precise shape controlling to constrain remeshing operations and prevent "fixing one issue while creating another". In addition, we employ a MLS-based up-sampling to control vertex positions in the tangent smoothing for geometric consistency keeping. It is useful for some regions with few vertexes or sharp curvature changing.

4.1 Inter-angle Optimization

The proposed inter-angle optimization provides shape constraints during the isotropic remeshing. According to different edge editing operations, we develop related inter-angle checking to predict potential shape degradation, thereby preventing the execution of detrimental mesh editing operations. Basically, the optimization includes three parts: split, collapse, and flip schemes.

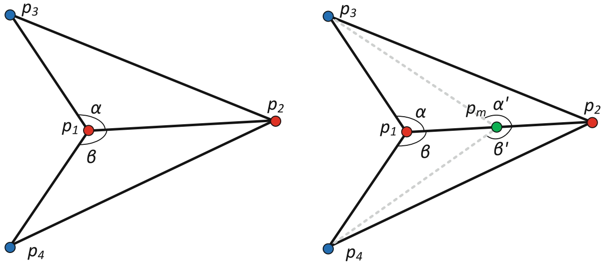

Fig. 1. Inter-angle optimization for split scheme. Angles $\angle\alpha$ and $\angle\beta$ control the split operation for $\overline{p_1p_2}$, which prevent the formation of $\angle\alpha'$ and $\angle\beta'$.

Split Scheme. Controlling split operation is simpler because adding edges to existing faces for splitting does not change the original geometry. Nevertheless, the implementation of angular surveillance remains imperative. The reason is that performing a split operation may introduce lower-quality obtuse triangles, thereby degrading isotropic property. An instance is shown in Fig. 1. If determined solely by edge length criteria, the edge $\overline{p_1p_2}$ should be split. However, since angles $\angle\alpha$ and $\angle\beta$ in the original face are obtuse, performing the split would generate new triangles with even larger obtuse angles $\angle\alpha'$, $\angle\beta'$. These highly distorted triangular elements would adversely affect the convergence of isotropic remeshing. Therefore, it is necessary to evaluate whether any of its four adjacent angles ($\angle\alpha, \angle\beta, \angle p_3p_2p_1$, and $\angle p_4p_2p_3$) are obtuse before splitting.

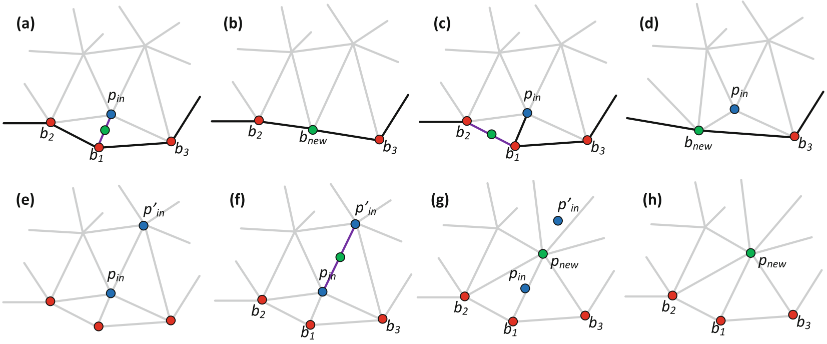

Fig. 2. Inter-angle optimization for collapse scheme. (a)∼(d) show the boundary condition; (e)∼(h) show the degree judgment.

Collapse Scheme. In practice, the collapse operation induces the most significant geometric alterations, not only reducing the vertex count but also directly modifying vertex-based degrees, while causing substantial angular distortions. For collapse scheme, we employ two additional constraints: boundary condition and degree judgment. The boundary condition means that if one point of an edge belong to a boundary edge, then the collapse operation should be blocked, as shown in Fig. 2. Edge points b_1, b_2, and b_3 connect a boundary line labeled by black color. Once we collapse either $\overline{p_{in}b_1}$ (Fig. 2a) or $\overline{b_1b_2}$ (Fig. 2c), the geometry of the boundary line is changed, which breaks the geometric consistency and becomes severe with more iterations.

Another constraint is the degree judgment, which estimates the impact of the collapse operation on vertex degrees to determine whether to proceed. Without vertex-based degree analysis may lead to abnormal degree assignments for the new vertex, consequently increasing the difficulty of subsequent flip operations. From the perspective of inter-angle view, an excessively high vertex degree implies that the average inter-angle associated with that vertex will fall below the ideal threshold (60°). It means the collapse operation may disrupt the isotropic property of the local region. An instance is also shown in Fig. 2. The degree values of p_{in} and p'_{in} are $deg(p_{in}) = 5$ and $deg(p'_{in}) = 6$ (average inter-angle is 65°, see in Fig. 2e). Once the collapse operation is implemented, the degree of new point p_{new} change to $deg(p_{new}) = 7$ (average inter-angle is 50°, see in Fig. 2h). Evidently, performing collapse on $\overline{p_{in}p'_{in}}$ reduces the isotropic quality. Combined mentioned constraints, collapse scheme is completed.

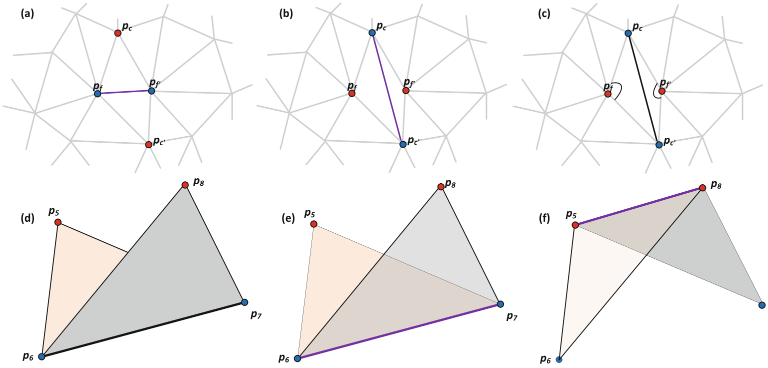

Fig. 3. Inter-angle optimization for flip scheme. (a)∼(c) show the inter-angle constraint; (e)∼(f) show the shape control.

Flip Scheme. Beyond the primary objective of degree optimization, flip operation must account the inter-angle constraint, which is similar to the split scheme. An instance is shown in Fig. 3. When determining whether to flip a given edge $\overline{p_fp'_f}$, we first compute the post-flip degree for each affected vertex. It is clear that the flip can optimize most vertexes, $deg(p_c) = 6, deg(p_f) = 6, deg(p'_f) = 6$, as shown in Fig. 3b. However, generated new obtuse angles ($\angle p_cp_fp'_c$ and $\angle p_cp'_fp'_c$ in

Fig. 3) violate shape control requirements in the isotropic remeshing. Therefore, the flip operation should be blocked for such condition.

An additional consideration is the geometric alteration induced by the flip operation. In fact, unless all four vertices involved in the flip operation are coplanar, it inevitably modifies the local geometry, as shown in Fig. 3e and Fig. 3f. If edge $\overline{p_6p_7}$ lies precisely on a sharp feature boundary, performing the flip operation would introduce geometric artifacts along the boundary, thereby breaking the sharp feature. To mitigate the issue, we implement a face-based dihedral angle checking. Based on the instance in Fig. 3d, we compute the normal vectors of $\triangle p_5p_6p_7$ and $\triangle p_8p_7p_6$, and judge the vector-based intersection angle θ. Once $\theta > \varepsilon(\varepsilon = 20^\circ$ by default), the flip operation is blocked for shape controlling. Combined split, collapse, and flip Schemes, the complete inter-angle optimization is established.

4.2 Geometric Consistency Keeping

The last operation of the four-steps remeshing is the tangent smoothing or vertex relocation, which adjusts vertex positions toward to their 1-ring neighborhood centers (gravity-weighted centroid [4]). The center can be formulated as

$$p_i' = \sum_{p_j \in N(p_i)} w_j p_j, \tag{1}$$

where p_i' is the center for the region of point p_i, p_j is 1-ring neighbor of p_i, w_j is the weight of p_j, which is typically represented by the area or cotangent weights corresponding to p_j to describe its influence on p_i. According to the vector $\overrightarrow{p_ip_i'}$(mapped on the tangent plane of p_i) with a "pulling back" function, p_i can be updated

$$p_i' = p_i + \lambda(I - n_i n_i^T)\overrightarrow{p_ip_i'}, \tag{2}$$

where n_i represent the normal vector of p_i, λ is the control parameter for update step size ($\lambda = 0.5$ by default). The aforementioned process is a common practice for vertex optimization while preserving geometric consistency. However, if the vertices of the original mesh are relatively sparse, the "pulling back" operation may fail to produce correct surface mapping due to ambiguity in neighborhood determination, resulting in local geometric distortion.

To address the issue, we propose a mesh up-sampling step to insert new points into different faces. Firstly, inserting the centroid of a triangular face. Then, based on the centroid, inserting the midpoints of the line segments between each vertex of the triangle and the centroid. Subsequently, inserting new centers between these newly created midpoints. Then, seven new points can be inserted for a single triangle. We adjust these new points' positions by MLS remapping [1]. Finally, we achieve an up-sampled point set that takes more accurate geometric representation. Once we implement the tangent smoothing, the "pulling back" function is based on the neighborhood defined on the up-sampled point set. The local geometric distortion can be significantly suppressed.

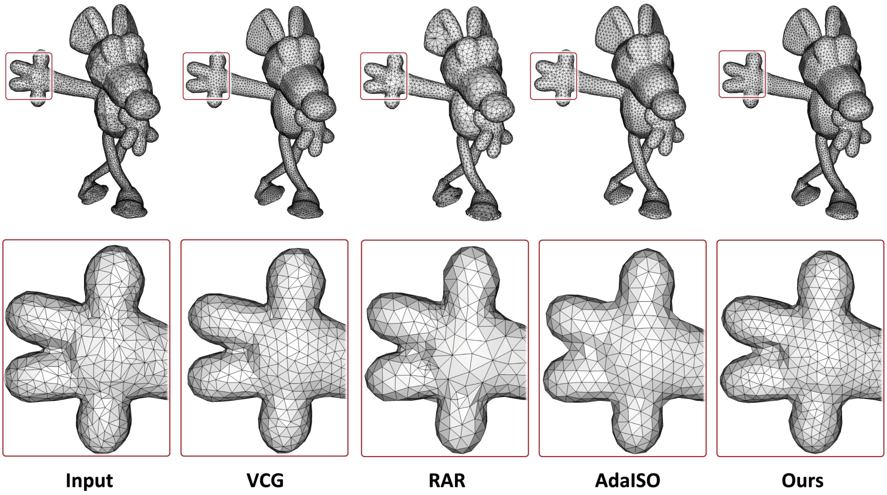

Fig. 4. Some remeshing results by different methods.

Table 1. Quantitative analysis of different remeshing methods. Hd: Hausdorff distance; Md: mean distance; θ_{max}: maximum degree of inter-angel; θ_{avg}: mean degree of inter-angel ($\pi/3 - \sum|\theta_i - \pi/3|$, θ_i is an inter-angel).

Model	VCG [4]				RAR [9]				AdaISO [21]				Ours			
	Hd	Md	θ_{max}	θ_{avg}	Hd	Md	θ_{max}	θ_{avg}	Hd	Md	θ_{max}	θ_{avg}	Hd	Md	θ_{max}	θ_{avg}
T0	0.0159	0.0016	151.5°	50.9°	**0.0130**	0.0016	179.1°	46.5°	0.0157	0.0017	179.9°	52.7°	0.0179	**0.0015**	**118.7°**	**52.8°**
T1008	0.0127	0.0017	167.2°	50.5°	**0.0115**	0.0018	166.8°	48.7°	0.0116	0.0017	179.9°	**53.7°**	0.0129	**0.0016**	111.1°	52.9°
T1022	0.0076	0.0010	164.8°	50.3°	–	–	–	–	**0.0070**	0.0011	179.9°	**52.6°**	0.0072	**0.0009**	**146.2°**	52.5°
T1036	0.0121	0.0016	145.8°	53.2°	0.0117	0.0016	159.1°	46.3°	**0.0111**	0.0017	179.9°	**53.5°**	0.0137	**0.0016**	**110.7°**	53.1°
T1050	0.0097	0.0011	172.5°	51.7°	**0.0089**	0.0013	154.4°	49.3°	0.0090	0.0012	179.9°	**53.4°**	0.0101	**0.0011**	**172.8°**	53.1°
T1064	0.0231	0.0026	177.4°	49.7°	0.0201	0.0028	176.8°	44.1°	**0.0176**	0.0028	180.0°	**53.1°**	0.0202	**0.0026**	**151.6°**	52.7°
T1078	0.0146	0.0020	176.9°	47.7°	0.0156	0.0020	174.8°	48.1°	–	–	–	–	0.0159	**0.0020**	**167.7°**	**52.6°**
T1092	0.0165	0.0017	169.9°	52.9°	**0.0162**	0.0018	177.1°	46.5°	0.0163	0.0019	180.0°	53.2°	0.0180	**0.0017**	**106.0°**	**53.3°**
T1106	0.0105	0.0014	178.2°	50.7°	0.0110	0.0016	175.8°	47.6°	0.0107	0.0015	179.9°	**53.4°**	**0.0096**	**0.0014**	**119.6°**	53.1°
T112	0.0244	**0.0031**	177.1°	46.9°	–	–	–	–	–	–	–	–	**0.0227**	0.0034	**145.1°**	**52.4°**
T1120	0.0155	0.0023	175.7°	49.3°	0.0177	0.0023	174.4°	46.3°	**0.0138**	0.0024	180.0°	53.1°	0.0157	**0.0023**	**137.1°**	**53.2°**
T1134	0.0086	0.0012	175.3°	49.6°	**0.0076**	0.0013	173.7°	48.5°	–	–	–	–	0.0080	**0.0011**	**138.6°**	**53.0°**
T1155	0.0165	0.0020	173.1°	51.7°	**0.0109**	0.0021	176.0°	48.4°	0.0147	0.0021	79.9°	**53.4°**	0.0177	**0.0020**	**114.9°**	53.0°
Avg	0.0120	0.0017	175.6°	50.4°	0.0107	0.0017	169.6°	48.6°	**0.0105**	0.0017	179.7°	52.9°	0.0116	**0.0016**	**113.8°**	**53.0°**

5 Experiments

In this part, We evaluate the performance of our remeshing scheme. The experimental machine equipped with Intel(R) i9-13900K 3.00 GHz, 128G RAM, GeForce RTX4090. The running system is Windows 11 with Visual Studio 2022 (64 bit). Firstly, we introduce the employed dataset and metrics. Then, we compare the performance between different methods. Finally, we show some applications based on our remeshing scheme and provide a discussion.

Dataset and Metrics. The experimental raw meshes are selected from SHREC [5], including non-rigid transformations. For quantitative analysis, we

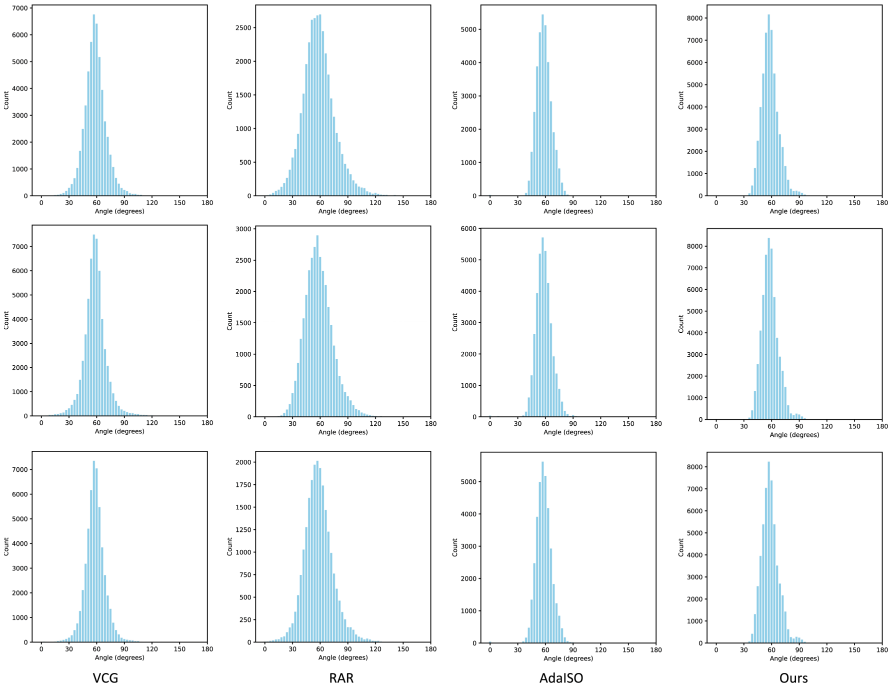

Fig. 5. Inter-angle distribution histograms of re-meshed models (T0, T14, T28).

Table 2. Time cost report for different methods.

	VCG [4]	RAR [9]	FCVT [8]	AdaISO [21]	Ours
T0	19.86 s	9.08 s	400.25 s	31.64 s	**8.93 s**
T14	21.26 s	**2.95 s**	381.85 s	39.72 s	11.26 s
T28	23.56 s	**1.58 s**	387.19 s	43.03 s	13.26 s
Avg	20.63 s	–	391.21 s	38.93 s	**10.89 s**

report Hausdorff distance, mean distance, average and maximum inter-angle values to measure the geometric consistency and isotropic property.

Comparisons. We compare different methods to show the performance of our remeshing scheme, including VCG-based isotropic remeshing (same to MeshLab version [4,6]), RAR remeshing [9], and adaptive isotropic remeshing [21]. Figure 4 shows some remeshing results by different methods. Our solution takes better balance between geometric consistency and isotropic property. Table 1 reports quantitative results based on SHREC models. Our method demonstrates superior shape control for large obtuse triangles (θ_{max} is smaller).

Benefiting from the implementation of geometric consistency keeping, our approach achieves better geometric consistency (Md value is lower). Figure 5 shows some histograms of inter-angle distribution. AdaISO and our solution

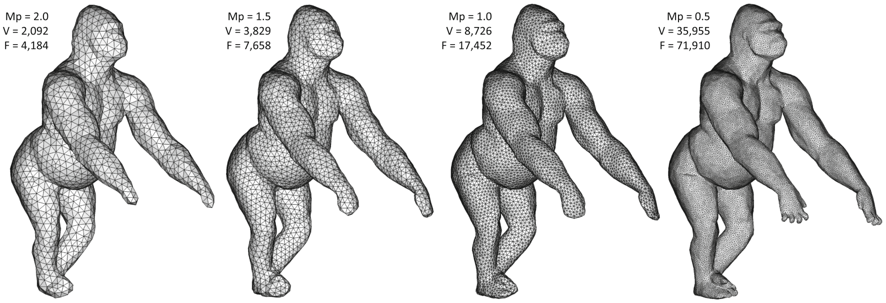

Fig. 6. Remeshing results by our method with different multi-parameters. Mp: multi-parameter value; V: vertex number; F: face number.

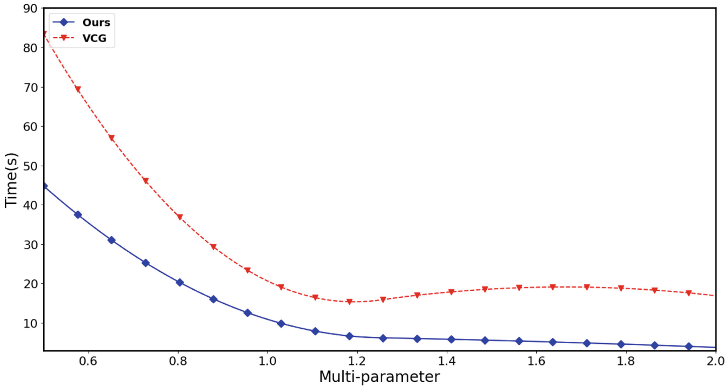

Fig. 7. Time cost curves for different methods with related multi-parameters.

take better distributions (concentrated around 60°). Table 2 reports computational efficiency for different methods. Our solution is faster than VCG-based isotropic remeshing and AdaISO solution, even they share similar four-steps strategy. RAR solution achieves faster runtime speeds for certain models, but fails to converge for some ones based on a larger test dataset. In addition, we employ a CVT-based solution (FCVT) [8] to be a reference in Table 2. According to statistical data, FCVT requires 30 times longer to achieve results comparable to our method. Significantly, our method holds advantages in terms of efficiency and stability.

Applications. A classical application of remeshing is the multi-resolution editing. By modifying the target edge length, we can either simplify or refine the input mesh. We assign different multi-parameters and multiply them by the average edge length to obtain new target edge lengths, thereby generating remeshing results with varying resolutions. Figure 6 shows some remeshing results by our method with different multi-parameters. The mesh resolution can be adjusted while preserving the isotropic property. We also report the time

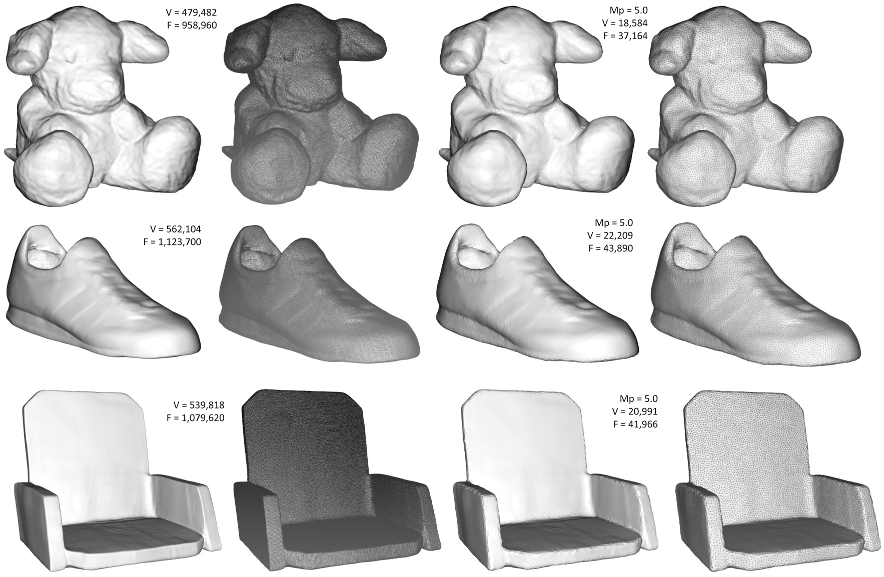

Fig. 8. Remeshing results based on 3D mesh generation framework. Left: generated meshes; right: remeshing results.

cost of remeshing with continuous multi-parameter variations, and add VCG-based remeshing as the reference. Figure 7 shows related time cost curves. Our method demonstrates superior time efficiency in both mesh simplification and refinement.

Another important application is assisting 3D-AIGC tasks. Currently, mainstream 3D mesh generation methods primarily rely on implicit reconstruction, driven by semantic and image features to produce accurate mesh outputs. However, they suffer from poor isotropic quality and excessive mesh volume. Our method can effectively enhance the isotropic quality of meshes while simultaneously simplifying them. Based on the generated results by MeshFormer [17], our remeshing is used to improve the mesh quality. Figure 8 shows some instances. It can be clearly observed that both the mesh volume and isotropic quality are optimized.

Discussion. Compared to the VCG-based remeshing solution, our method utilizes inter-angle control to avoid cross-interference between each basic operation, thereby improving convergence efficiency. Tables 1 and 2 demonstrate that our approach outperforms VCG-based version in both mesh optimization quality and computational efficiency. Compared to the CVT-based approach, our method inherits the advantages of isotropic remeshing while demonstrating significant superiority in convergence speed. The core reason is that our basic operations enable more direct updates to triangle shapes. In contrast, the CVT-based method relies on Voronoi cell computation and centroidal optimization for mesh editing, which inherently suffers from efficiency limitation.

6 Conclusions

In this paper, we propose a new isotropic remeshing method with inter-angle optimization. Angle-based constraints are incorporated into the basic operations (split, collapse, and flip) to suppress the generation of obtuse triangles. Compared to the original four-steps isotropic remeshing, our method effectively reduces repetitive edge editing and improves convergence efficiency. Benefited from a mesh-based up-sampling scheme, the quality of isotropic remeshing can be significantly improved. Experimental reports demonstrate that our solution exhibits comprehensive advantages in isotropic property optimization, geometric consistency keeping, and computational efficiency.

References

1. Alexa, M., Behr, J., Cohen-Or, D., Fleishman, S., Levin, D., Silva, C.T.: Point set surfaces. In: Proceedings of the Visualization, pp. 21–29. IEEE (2001)
2. Alliez, P., Cohen-Steiner, D., Devillers, O., Lévy, B., Desbrun, M.: Anisotropic polygonal remeshing. In: Proceedings of the SIGGRAPH, pp. 485–493 (2003)
3. Bern, M., Eppstein, D., Gilbert, J.: Provably good mesh generation. J. Comput. Syst. Sci. **48**(3), 384–409 (1994)
4. Botsch, M., Kobbelt, L.: A remeshing approach to multiresolution modeling. In: Proceedings of the Eurographics Symposium on Geometry Processing, pp. 185–192 (2004)
5. Bronstein, A., et al.: Shrec 2010: robust large-scale shape retrieval benchmark. Proc. 3DOR **5**(4), 1–8 (2010)
6. Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., Ranzuglia, G.: MeshLab: an open-source mesh processing tool. In: Scarano, V., Chiara, R.D., Erra, U. (eds.) Eurographics Italian Chapter Conference, pp. 129–136. The Eurographics Association (2008)
7. Dapogny, C., Dobrzynski, C., Frey, P.: Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems. J. Comput. Phys. **262**, 358–378 (2014)
8. Du, X., Liu, X., Yan, D.M., Jiang, C., Ye, J., Zhang, H.: Field-aligned isotropic surface remeshing. In: Computer Graphics Forum, vol. 37, pp. 343–357. Wiley Online Library (2018)
9. Dunyach, M., Vanderhaeghe, D., Barthe, L., Botsch, M.: Adaptive remeshing for real-time mesh deformation. In: Proceedings of the Eurographics, pp. 29–32. The Eurographics Association (2013)
10. Goes, F., Memari, P., Mullen, P., Desbrun, M.: Weighted triangulations for geometry processing. ACM Trans. Graph. **33**(3), 28:1–28:13 (2014)
11. He, X., Lv, C., Huang, P., Huang, H.: Windpoly: polygonal mesh reconstruction via winding numbers. In: Proceedings of the European Conference on Computer Vision, pp. 294–311. Springer, Cham (2024)
12. Hou, W., et al.: SDF-RVD: restricted Voronoi diagram on signed distance field. Comput.-Aided Des. **144**, 103166.1–103166.12 (2022)
13. Hu, B.Y., Ye, C., Su, J.P., Liu, L.: Manifold-constrained geometric optimization via local parameterizations. IEEE Trans. Visualization Comput. Graph. **29**(2), 1318–1329 (2021)

14. Jiao, X., Colombi, A., Ni, X., Hart, J.: Anisotropic mesh adaptation for evolving triangulated surfaces. Eng. Comput. **26**(4), 363–376 (2010)
15. Kazhdan, M., Hoppe, H.: Screened Poisson surface reconstruction. ACM Trans. Graph. **32**(3), 29:1–29:13 (2013)
16. Lévy, B., Liu, Y.: Lp centroidal Voronoi tessellation and its applications. ACM Trans. Graph. **29**(4), 119:1–119:11 (2010)
17. Liu, M., et al.: Meshformer: high-quality mesh generation with 3D-guided reconstruction model. arXiv preprint arXiv:2408.10198 (2024)
18. Liu, Y.J., Xu, C.X., Fan, D., He, Y.: Efficient construction and simplification of delaunay meshes. ACM Trans. Graph. **34**(6), 174:1–174:13 (2015)
19. Lv, C., Lin, W., Zhao, B.: Voxel structure-based mesh reconstruction from a 3D point cloud. IEEE Trans. Multimed. **24**, 1815–1829 (2021)
20. Lv, C., Lin, W., Zhao, B.: Intrinsic and isotropic resampling for 3D point clouds. IEEE Trans. Pattern Anal. Mach. Intell. **45**(3), 3274–3291 (2022)
21. Lv, C., Lin, W., Zheng, J.: Adaptively isotropic remeshing based on curvature smoothed field. IEEE Trans. Visualization Comput. Graph. **30**(7), 3196–3209 (2024)
22. Lv, C., Wu, Z., Wang, X., Zhou, M.: 3D facial similarity measurement and its application in facial organization. ACM Trans. Multimed. Comput. Commun. Appl. **16**(3), 82:1–82:20 (2020)
23. Lv, C., Wu, Z., Wang, X., Zhou, M., Toh, K.A.: Nasal similarity measure of 3D faces based on curve shape space. Pattern Recogn. **88**, 458–469 (2019)
24. Wang, Y., et al.: Isotropic surface remeshing without large and small angles. IEEE Trans. Visualization Comput. Graph. **25**(7), 2430–2442 (2018)
25. Xu, Q.C., Yan, D.M., Li, W., Yang, Y.L.: Anisotropic surface remeshing without obtuse angles. In: Computer Graphics Forum, vol. 38, pp. 755–763. Wiley Online Library (2019)
26. Yan, D.M., Guo, J., Jia, X., Zhang, X., Wonka, P.: Blue-noise remeshing with farthest point optimization. In: Computer Graphics Forum, vol. 33, pp. 167–176. Wiley Online Library (2014)
27. Yan, D.M., Lévy, B., Liu, Y., Sun, F., Wang, W.: Isotropic remeshing with fast and exact computation of restricted Voronoi diagram. In: Computer Graphics Forum, vol. 28, pp. 1445–1454. Wiley Online Library (2009)
28. Ye, Z., Yi, R., Yu, M., Liu, Y.J., He, Y.: Geodesic centroidal Voronoi tessellations: theories, algorithms and applications. arXiv preprint arXiv:1907.00523 (2019)
29. Yi, R., Liu, Y.J., He, Y.: Delaunay mesh simplification with differential evolution. ACM Trans. Graph. **37**(6), 263:1–263:12 (2018)
30. Zhang, R., Pan, S., Lv, C., Gong, M., Huang, H.: Architectural co-lod generation. ACM Trans. Graph. (Proc. SIGGRAPH Asia) **43**(6), 193:1–193:16 (2024)
31. Zheng, J., Tan, T.S.: Computing centroidal Voronoi tessellation using the GPU. In: Symposium on Interactive 3D Graphics and Games, pp. 1–9 (2020)