Acta Technica *62* (2017), No. 6B, 985–994

An Effective Approach on Recovering Sharp Features of Triangular Meshes

Yuhui Peng[1,*], Yan Zhao[2,*]

Abstract. The sharp features, including edges, corners and boundaries, usually express the most important geometric information for triangular meshes. The purpose of this investigation is to reconstruct sharp edges from blended or chamfered features for mesh processing with high accuracy. The proposed approach involves two processes: sharpness-based region-recognizing, which identifies the feature vertices based on the vertex sharpness, and sharp-features reconstructing. In the process of region-recognizing, the angle variations of normal vectors are introduced to indicate the vertex sharpness. Then, the feature regions comprised of "sharp" vertices and facets could be identified. Furthermore, during the process of sharp feature reconstructing, the coordinates of concerned vertices are adjusted gradually using an iterative filtering algorithm depending on sharpness, which updates the feature regions from the inside-outside. Finally, the experimental results validate the effectiveness and robustness of the proposed method in sharpening meshes.

Key words. Triangular meshes, Sharp feature, Vertex sharpness, Mesh sharpening, Normal filtering.

1. Introduction

Triangular mesh in STL format is gaining wider acceptance in various applications such as industrial product design, computer graphics, reverse engineering, CAE applications, Rapid Prototyping and CNC machining [1-3] etc. Sharp features containing corners, edges, and boundaries, ordinarily demonstrate the essential geometric information of the whole shape. During the process of simplification, denoising and smoothing, sharp features are often blurred, deformed or even lost. Therefore, it is necessary to reconstruct the sharp features for mesh processing because the feasibility of the model reconstructing relies on the accuracy of feature recovering.

In order to reconstruct the sharp features of triangular meshes, many researchers focused on developing various algorithms for mesh sharpening. Generally, the di-

[1]School of Mechanical Engineering and Automation, Fuzhou University, Fuzhou, 350116, China
[2]The Fist Affiliated Hospital of Fujian Medical University, Fuzhou 350009, China
* These authors contribute equally to the article and should be viewed as co-first authors

hedral angle (angle between two neighboring triangular facets), is applied to detect the sharp meshes. Then, facets located on the chamfer are subdivided to recover sharp features on the basis of the six steps color-marking method [4]. However, this solution seems to be invalid, in the case of dense distribution of facets on the chamfer. It means that the associated second-order nearest-neighbor facets still stay on the chamfer. It results in the concerned meshes cannot be recognized according to this approach. Moreover, Bayes classifier is applied to detect the sharp region in some researches, through different Normal distribution between sharp and unsharp meshes. As a result, a sharpness-dependent filtering algorithm is proposed to effectively recover sharp features in repairing boundary of holes and caves [5]. Meanwhile, due to no control of vertices sharpness, the meshes on the fillet are liable to be mistakenly sharpened in the reconstructing procedure. Thus, the USSOD (uniformly supported second-order difference) method is presented to calculate the max-error for each vertex covered by the range parameter λ. After that, the value of max-error is applied to evaluate the sharpness of vertex, classify them into different groups, and then narrow the feature region supposed to contain sharp edges. Subsequently, the sharp features on fillets and chamfers are reconstructed in the process of vertices shrinking based on the geometric prediction method [4, 6]. It has been realized that this approach is likely to re-sharpen the facets in the sharp regions for the misjudgment of vertices in high curvature area, since this feature recognition method is highly related to the dihedral edge angle. Furthermore, in the previous approach [7], a feature enhancing and smoothing mesh algorithm is put forward to realize repair and reconstruction of the target features. It bases on the principle that iteratively use of diffusion of normals and vertices by applying the Perona-Malik gray-scale image nonlinear diffusion method. In addition, some other approaches have also been proposed [8, 9] for the similar purpose of sharp region reconstructing.

Motived by the impressive results of sharpen range control[10] and sharpness-dependent[5] for meshes sharpening, a new method is proposed in this paper to eliminate the influence of feature deterioration caused by smoothing or denoising and sharp features on the fillets and chamfers, and then the sharp region in meshes can be effectively reconstructed. In the proposed method, it relies on the following basic processing procedure: (1) Feature vertices in the sharpening region are detected according to the user-specified parameter λ and the value of their sharpness; (2) Classify the facets containing feature vertex into the static and dynamic type; (3)Adjust the normal vector of dynamic facets progressively from inside-outside; (4) Update the vertex coordinates iteratively to meet with the adjusted facet normal.

2. Sharp region detection

The sharp region defined herein is that the special area consists of meshes degraded or deformed from the original sharp area, when processing meshes with the methods of smoothing, simplification and so on. Therefore, in order to recover the sharp features, sharp region embedded in the whole model should be effectively identified at first.

2.1. Vertex sharpness

As shown in figure 1, Vi is defined as the neighborhood set of vertex vi , Vi={vj, $d(vi, vj) \leq r$}. Where $d(vi, vj)$ is the Euclidean distance between vi and vj , and r is an user-specified parameter to control the range within sharpening region. It also help to avoid the existed sharp region being sharpened again. The angle θij is expressed as the angle between normal vector ni of vertex vi and normal vector nj of vertex vj . If vi is located in a smooth region, the variation of θij is low, especially in case of plane area, the value of θij equals zero at this point. Therefore, the variation of θij is defined as vertex sharpness in the following Eq. (1).

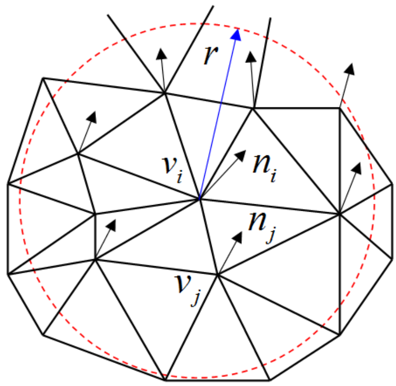


Fig. 1. Feature points in sharpening region

$$s_i = \frac{1}{\|\mathrm{V}_i\|} \sum_{v_j \in \mathrm{V}_i} \left(\theta_{ij} - \overline{\theta_i}\right)^2 . \tag{1}$$

Where $||\mathrm{V}i||$ represents the number of vertices in Vi, $\overline{\theta_i}$ is arithmetic average value of θij , and it can be written by Eq. (2).

$$\overline{\theta_i} = \frac{1}{\|\mathrm{V}_i\|} \sum_{v_j \in \mathrm{V}_i} \theta_{ij} . \tag{2}$$

2.2. Methodology

Assuming a two manifold triangular meshes M with no noise or aliasing errors on the non-feature region, thus the steps taken to detect sharpening region are expressed as follows:

(1) Identification of sharp feature vertices

If one dihedral angle involved in 1-ring facets of vertex vi (adjacent facets who share vertex vi), is greater than 90°, the sharpness value of vertex vi is set to zero and the vertex vi is treated as static vertex, whose position will not be moved. By doing so, it is able to prevent the sharp feature vertex from being re-sharpened as the previous approaches.

(2) Calculation of vertex sharpness

The vertex sharpness si can be computed according to the Eq. (1) for each vertex in the detected feature region.

(3)Dynamic facets

Assuming the maximum value of vertex sharpness $smax$ equals 1.0, then all vertex sharpness are distributed in the range of [0, 1]. Each vertex vi, whose sharpness value si is greater than a pre-set threshold η, is defined as a dynamic vertex. The corresponding facet containing dynamic vertex is called dynamic triangle.

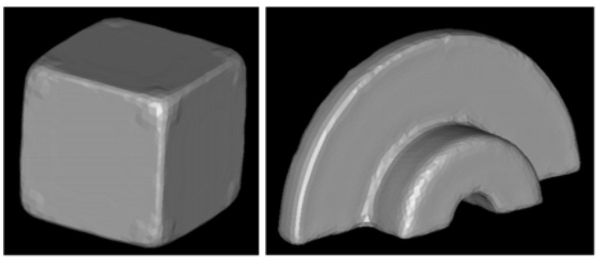
(4) Determination of r and η

The value of parameter r is determined by the following criterion. The greater value of r is, means that the larger scale of involved meshes will be. In the practice, the value of r expands from a small to a large value progressively. The minimum r is set as the average edge length $\overline{e}$ of the whole model, and each increasing step is 0.2 $\overline{e}$ till all vertices concerned can be covered under the r value.

The sharpness threshold η is chosen in another way. It should be noticed that the value η determines how many vertices will be identified as feature vertices. By means of manual sampling in the sharpening region, an initial η is set as the minimum value calculated from sampled vertices.

2.3. Example

Finally, two models, Cube and Half-disk (shown in figure 2.), are taken as examples to test the effectiveness of the proposed method. Then, the test results, from the USSOD method in [10] and the method proposed in this paper, will be compared during the process of sharp region detection.

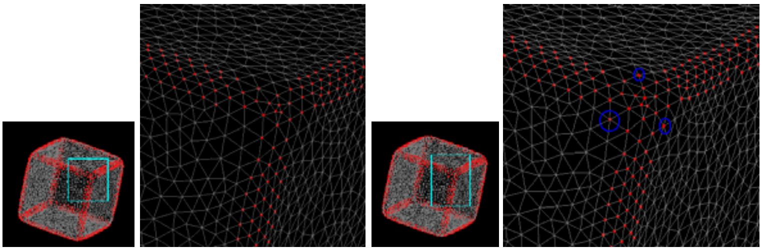


(a) Cube (b) Half-disk

Fig. 2. original models

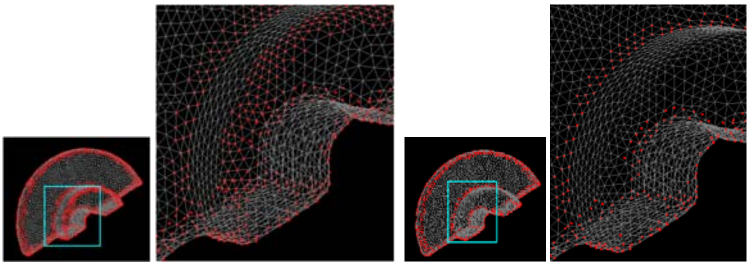
As shown in figure 3, both methods have a good performance in recognizing sharpening regions degraded from the boundary of Cube. It should be to be noticed that the vertices in blue highlight, are missed by USSOD method due to small dihedral angles, in contrast, they can be detected by the vertex sharpness value proposed in this paper.

The test results also show that the USSOD method often judge vertices on the fillet as sharpening feature vertices depending on the criterion of dihedral angle. It inevitably results in mis-sharpening shown in figure 4(a). Obviously, the proposed



(a) USSOD (b) This paper

Fig. 3. Comparison of sharpening region detection for cube



(a) USSOD (b) This paper

Fig. 4. Comparison of sharpening region detection for the half disk

method relying on the variation of the vertex normal is a prefer method to distinguish the vertices of sharpening region from fillets with higher accuracy.

3. Sharp feature reconstruction

The core of sharp feature recovering is how to determine the new position for dynamic vertices. In this paper, the method of facet normal filtering is developed to adjust facet normal with the least error rule. Actually, the new facet normal determines the moving direction of dynamic vertices.

3.1. Facet normal filtering

The process of facet normal filtering is mainly used in smoothing or denoising. It usually moves the feature vertices along a certain direction and then blurs or degrades the sharp features. In other words, the procedure of sharpening is a reversing procedure of smoothing.

Moreover, mean filtering is one of most popular ways to adjust normal vector. The new normal n'_{f_i} [11] of triangular facets *fi* is calculated by Eq. (3), based on the average weighted normals of the neighboring triangles *fi* , where $\left\|N_i^f\right\|$ represents the number of neighboring triangles and weight wij is expressed by $w_{ij} = \frac{1}{\|N_i^f\|}$.

$$n'_{f_i} = \frac{\sum\limits_{f_j \in N_i^f} w_{ij} n_{f_j}}{\left\| \sum\limits_{f_j \in N_i^f} w_{ij} n_{f_j} \right\|}. \tag{3}$$

In the view of the influence of facet area size, Eq. (3) can be translated into Eq. (4) through weighted size area, where Aj expresses the area of facet *fi*.

$$n'_{f_i} = \frac{\frac{1}{\sum\limits_{f_j \in N_i^f} A_j} \sum\limits_{f_j \in N_i^f} A_j n_{f_j}}{\left\| \frac{1}{\sum\limits_{f_j \in N_i^f} A_j} \sum\limits_{f_j \in N_i^f} A_j n_{f_j} \right\|}. \tag{4}$$

The principle of mean filtering reveals that the new normal of facet *fi* is determined by the normals of neighboring triangles. It possess the average contribution or different contribution according to the area size respectively. Considering the influence of different angle between facet *fi* normal and other neighboring facet, another normal filtering method is presented as the following Eq. (5) in [13], where the weighting function is defined in Eq. (6).

$$n'_{f_i} = \frac{\sum\limits_{f_j \in N_i^f} w_j n_{f_j}}{\left\| \sum\limits_{f_j \in N_i^f} w_j n_{f_j} \right\|}. \tag{5}$$

$$w_j = \begin{cases} f(n_{f_i} \cdot n_{f_j} - T) & if \;\; n_{f_i} \cdot n_{f_i} > T \\ 0 & if \;\; n_{f_i} \cdot n_{f_i} \le T \end{cases} \tag{6}$$

Where T is a user-defined threshold and n_{f_j} is the normal of facet *fi* and *fj*[12] respectively, when assuming $f(x) = x2$. In Eq. (5), it clearly explains that, the closer n_{f_j} and n_{f_i} are, the greater contribution facet *fj* will makes on the new normal of facet *fi*.

During the procedure of meshes sharpening in this paper, the normals of dynamic facets are adjusted progressively in accordance with the normals of neighboring static facets. Then, Eq. (6) can be translated into Eq. (7) as follows, where $s_j^1 s_j^2 s_j^3$ is denoted as vertex sharpness for three vertices on facet *fj* respectively, and facet sharpness $s_{f_j} = \frac{1}{3}(s_j^1 + s_j^2 + s_j^3)$, when assuming $f(x) = x2$.

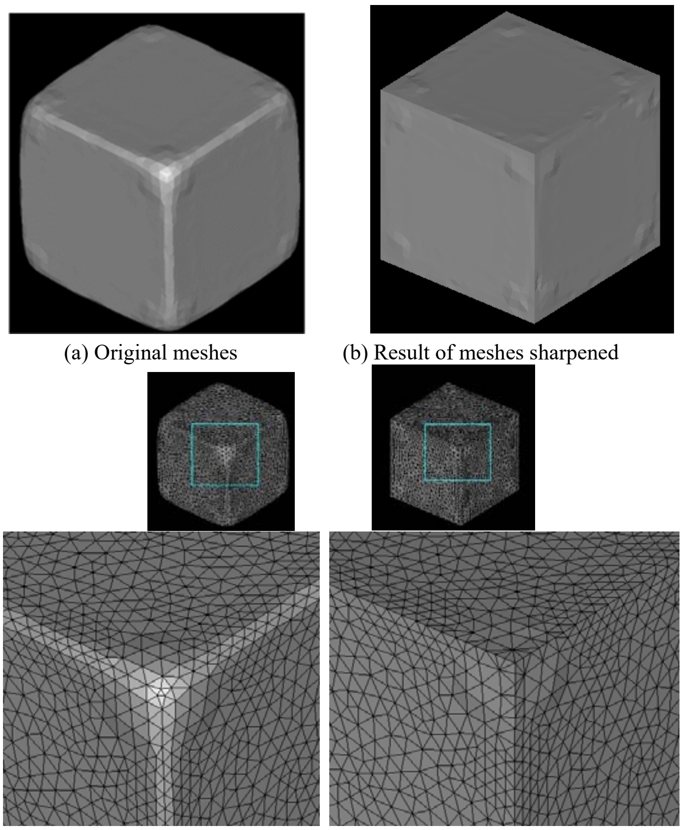
$$w_j = f(1 - s_{f_j}). \tag{7}$$

Furthermore, the neighboring triangles set of triangle *fi* is written by N_i^f, which is the union of adjacent triangles with triangle *fi*. Then, the neighboring triangles set of vertex *vi* can be expressed as $Fi=\{fj,\ vk\in fj \text{ and } vk\in \mathrm{V}i\}$. In Eq. (7), it describes that, if the triangle *fj* is more flat, the value of *wj* is larger. Consequently, it will exert greater contribution on the new normal vectorn'_{f_i}.

In addition, it is noteworthy that the proposed normal filtering is implemented with facets in the order from low to high facet sharpness. In other words, the facet normal adjustment is from inside flat region to the outside.

3.2. Position updating

The position updating for each feature vertex follows the Eq. (8) as previous approach [13], where $||Fi||$ is the number of triangles in *Fi*, *cj* is the centroid of triangle *fj*, and *xi*, x'_i represents the original coordinates and the new coordinates of vertex *vi* respectively.

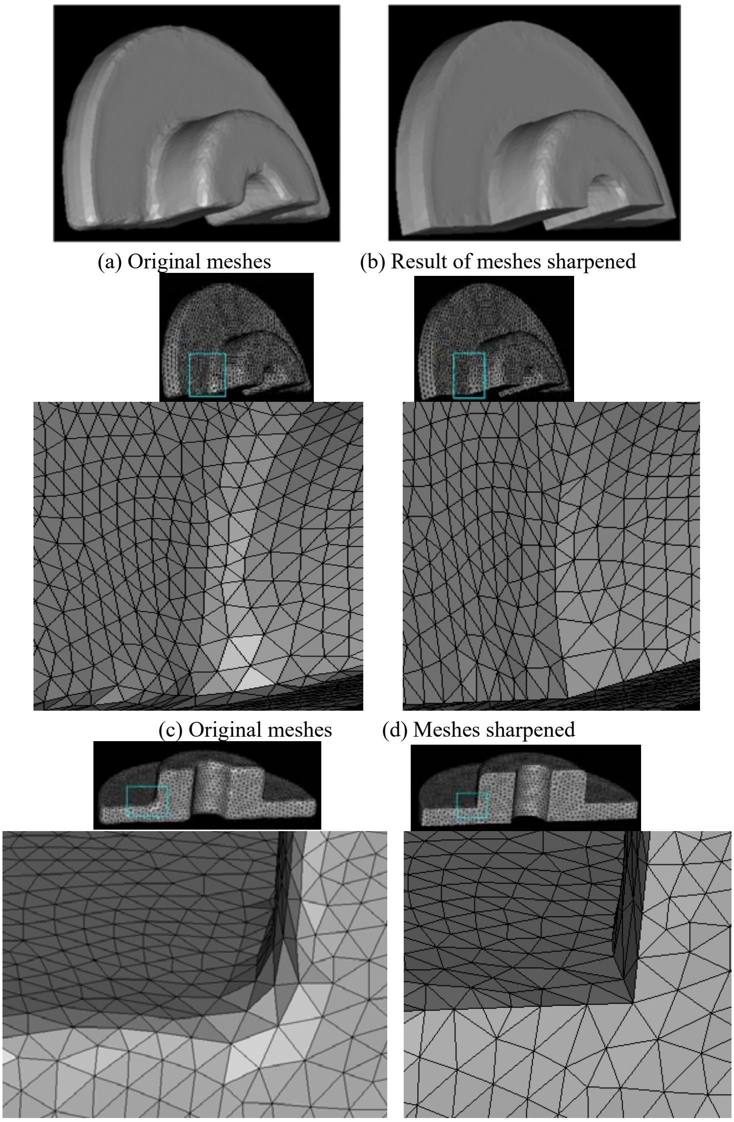


(a) Original meshes (b) Result of meshes sharpened

(c) Original meshes (d) Meshes sharpened

Fig. 5. Result of Cube sharpened

$$x_i' = x_i + \frac{1}{\|F_i\|} \sum_{f_j \in F_i} n_j [n_j \cdot (c_j - x_i)] \,. \tag{8}$$



(a) Original meshes (b) Result of meshes sharpened

(c) Original meshes (d) Meshes sharpened

(e) Original meshes (f) Meshes sharpened

Fig. 6. Results of Half-disk sharpened

Then, the new coordinates of vertex vi will be iteratively calculated according to the Eq. (8) till the facet normal n_{f_i} for dynamic facets is as close as to the ideal normal n'_{f_i}, when assuming that $1 - n_{f_i} \cdot n'_{f_i} < \xi$, where n_{f_i} is in accordance with

n'_{f_i}, $n_{f_i} \cdot n'_{f_i} = 1$. In the practical application, the value of ξ is selected in the range of 0 to 0.01 normally.

3.3. Results

The algorithm described in this paper is implemented with MFC and OpenGL running on the following hardware: Desktop PC, CPU PIV 2.8Ghz, Memory 1G, NVIDIA GeForce4 MX440 128MB.

The examples in figure 5 and figure 6 are conducted to indicate the ability of proposed method on sharp feature reconstruction. According to the results shown in figure (6-b), it clearly demonstrates that this approach perfectly preserves the original fillet feature from being sharpened.

The concerned initial values and computing time are listed in table 1. Evidently, most of the computing time is expended in sharpening region recognition. The value of $\boldsymbol{r}/\bar{e}$ is becoming larger when there exist a larger difference in the facets' size, which leads to lower efficiency of the algorithm.

Table 1. initial value and time consumed

Model	Number of triangular facets	$r/\bar{e}$	η	Time for sharpening region detection (s)	Time for normals filtering (s)	Time for position updating (s)
Cube	6, 276	1.4	0.05	5.4	1.7	0.6
Half-disk	7, 406	1.6	0.1	8.1	2.2	0.7

4. Conclusion

Sharp features detail, such as corners, edges and the boundary, is the most important geometrical information for a model. Hence, this paper aims to develop an effective mesh sharpening method, in order to support sharp features reconstruction effectively, including sharp edges and corners. The proposed approach primarily consists of two processes, sharpening region detection and facet normal filtering. Experimental results reveal that the developed approach works better on triangular meshes with fillets than the previous approaches.

Acknowledgement

The author would like to thank the generous support of the Scientific Research Foundation of National Health and Family Planning Commission-Joint Research Projects of Fujian Provincial Health and Education (grant no. WKJ2016-2-24.) and Fujian Provincial High-Tech research program (grant no. 2016Y0036).

References

[1] Bianconi F: (2002) *Bridging the gap between CAD and CAE using STL files.* International Journal of CAD/CAM, 2(1):55–67.

[2] Lefebvre PP, Lauwers B: (2004) *STL model segmentation for multi-axis machining operations planning.* Computer-Aided Design & Applications, 1(1–4):277–84.

[3] Bahattin Koc, Yawei Ma, Yuan-Shin Lee: (2000) *Smoothing STL files by Max-Fit biarc curves for rapid prototyping*, Rapid Prototyping Journal, Volume 6 . Number 3. 2000. pp.186-203.

[4] M.Attene, B.Falcidino, M.Spagnuolo, et al.: (2005) *Sharpen&Bend: Recovering curved edges in triangle meshes produced by feature insensitive sampling.* IEEE Trans on Visualization and Computer Graphics, 11(2):181-192.

[5] C.Y. Chen, K.Y.Cheng: (2008) *A Sharpness-Dependent Filter for Recovering Sharp Features in Repaired 3D Mesh Models.* IEEE Transactions on Visualization and Computer Graphics, 14(1):200-212.

[6] C.L.Wang: (2006) *Bilateral Recovering of Sharp Edges on Feature-insensitive Sampled Meshes.* IEEE Transactio ns on Visualization and Computer Graphics, 12(4):629-639.

[7] Nie Junhong, Zhou Laishui, Liu Shenglan: (2004) *A Mesh Smoothing Algorithm for Feature Enhancing.* Mechanical science and technology, 23(1):110-112.

[8] J.G Shen, S.Y Zhang, Z.Y Chen, et al.: (2009) *Mesh sharpening via normal filtering.* Journal of Zhejiang University Science A, 10(4):546-553.

[9] Zhao Liu, Maodong Pan, Zhouwang Yang, et al.: (2015) *Recovery of Sharp Features in Mesh Models.* Commun. Math. Stat, (3):263–283.

[10] C.C.L.Wang: (2006) *Incremental reconstruction of sharp edges on mesh surfaces.* Computer-Aided Design, 38(6):689-702.

[11] Y.Ohtake, A.G.Belyaev, I.A.Bogaevski: (2001) *Mesh regularization and adaptive smoothing.* Computer-Aided Design, 33(4):789–800.

[12] H.Yagou, Y.Ohtake, A.Belyaev: (2002) *Mesh smoothing via mean and median filtering applied to face normals.* In:GeometricModeling and Processing Proceedings, 124–131.

[13] X.Sun, P.L.Rosin, R.R.Martin, et al.: (2007) *Fast and Effective Feature-preserving Mesh Denoising.* IEEE Transactions on Visualizatio n and Computer Graphics, 13(5):925-938.

Received May 7, 2017