Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Feature detection of triangular meshes via neighbor supporting*

Xiao-chao WANG$^{\dagger 1}$, Jun-jie CAO^{1,2}, Xiu-ping LIU$^{\dagger\ddagger 1}$, Bao-jun LI^{3}, Xi-quan SHI^{4}, Yi-zhen SUN^{2}

(*[1]School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China*)

(*[2]State Key Laboratory of Structural Analysis for Industrial Equipment, Department of Engineering Mechanics,
Dalian University of Technology, Dalian 116024, China*)

(*[3]State Key Laboratory of Structural Analysis for Industrial Equipment, School of Automotive Engineering,
Faculty of Vehicle Engineering and Mechanics, Dalian University of Technology, Dalian 116024, China*)

(*[4]Department of Mathematical Sciences, Delaware State University, Dover, DE 19901, USA*)

†E-mail: xpliu@dlut.edu.cn; wangxiaochao18@gmail.com

Received Nov. 1, 2011; Revision accepted Dec. 14, 2011

Abstract: In this paper, we propose a robust method for detecting features on triangular meshes by combining normal tensor voting with Neighbor Supporting. Our method contains two stages: feature detection and feature refinement. First, the normal tensor voting method is modified to detect the initial features, which may include some pseudo features. Then, at the feature refinement stage, a novel salient measure deriving from the idea of Neighbor Supporting is developed. Benefitting from the integrated reliable salient measure feature, pseudo features can be effectively discriminated from the initially detected features and removed. Compared to previous methods based on differential geometry property, the main advantage of our method is that it detects not only sharp features, but weak features as well. Numerical experiments show that our algorithm is robust, effective, and can produce more accurate results. We also discuss how detected features are incorporated into applications, such as feature-preserving mesh denoising and hole-filling, and present visually appealing results by integrating feature information.

Key words: Feature detection, Neighbor Supporting, Normal tensor voting, Salient measure
doi:10.1631/jzus.C1100324 **Document code:** A **CLC number:**

1 Introduction

In recent years, triangular meshes have been extensively used to represent objects in computer-aided design and computer graphics, not only due to their simplicity and efficiency, but also the rapid development of 3D acquisition techniques. In the process of model analysis, understanding and editing, feature detection usually plays an important and preliminary role in a variety of applications, such as feature-preserving mesh denoising (Takafumi *et al.*, 2005; Fan *et al.*, 2010; Bian and Tong, 2011), simplification (Kim *et al.*, 2006), segmentation (Stylianou and Farin, 2004), hole-filling (Li *et al.*, 2010). The features involved in this paper refer to vertices lying at the intersection of multiple smooth surfaces, which are also called discontinuous points (Angelo and Stefano, 2010). Feature lines can be obtained by connecting feature vertices.

For most methods, (Watanabe and Belyaev, 2001; Stylianou and Farin, 2004; Ohtake *et al.*, 2004; Yoshizawa *et al.*, 2008; Mao *et al.*, 2009), high quality estimating of the differential geometric properties is critical to feature detection. However, the acquired data are inevitably contaminated with noise

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. U0935400, 60873181, and 61173102) and the Fundamental Research Funds for the Central Universities, China (No. DUT11SX08)

as higher order derivatives of the surface are noise sensitive. These unreliable differential geometric property based methods lead to poor results. Another challenge for feature detection is to precisely estimate the differential geometric properties in discontinuity regions. For instance, a corner has no preferred orientation and the curvature is also meaningless (Ohtake *et al.*, 2004). Therefore, the noise and discontinuities should be specially taken care of for piecewise-smooth surfaces in feature detection.

Optional preprocessing can be adopted to deal with noise, such as the smoothing used in (Hildebrandt *et al.*, 2005). Although smoothing can minimize the effects of the noise, directly smoothing the original surface will change or destroy the original surface. Furthermore, some salient features might be diffused and weak features will be filtered out.

To detect features on triangular meshes, a two-stage method is proposed in this paper. At the first stage, the modified normal tensor voting method is adopted to detect the initial features, which include all potential features, such as sharp and weak features and possibly with noise. At the second stage, a refinement of feature selecting is conducted to extract the real features from the initial detected features. To this end, we introduce a novel salient measure via Neighbor Supporting. From this measure, we develop an efficient and robust feature detection algorithm, which not only extracts sharp features, but weak features as well. The contributions of our work can be summarized as follows:

- Based on the idea of Neighbor Supporting, an anisotropic vertex salient measure is defined, which can effectively characterize the geometric features of the surface.
- Compared to the methods based on purely differential geometric properties, the newly defined salient measure allows the simultaneous detection of both sharp and weak features.
- A unified framework for feature detection on triangular meshes is proposed, which is insensitive to noise and has a strong ability to discriminate actual features from noise.

The rest of the paper is organized as follows. Related work is briefly reviewed in Section 2. Method overview and the idea of Neighbor Supporting are presented in Section 3, while Section 4 provides the details of the algorithm. Experimental results and applications are presented in Section 5. Section 6 concludes the article and provides some future research directions.

2 Related works

Recently, numerous research techniques have been developed for feature detection on triangular meshes.

According to differential geometry preliminaries, for a smooth oriented surface, feature lines can be defined via first and second order curvature derivatives, i.e., the extreme of principal curvatures along corresponding principal directions. To detect features, a natural idea is following the mathematical definition, such as the method proposed by Ohtake *et al.* (2004). Stylianou and Farin (2004) first identified the feature vertex by testing whether its largest (smallest) curvature was locally maximum (minimum) in its corresponding direction. Then, the region growing and skeleton techniques were employed to get the final feature lines. This method coupled with the similar measure was further used in (Mao *et al.*, 2009) to detect perceptually salient features on 3D meshes. Yoshizawa *et al.* (2005) extracted the feature lines by estimating the curvature tensor and curvature derivatives via local polynomial fitting. Kim et al. (Kim and Kim, 2006) adopted the moving-least-squares approximation method to estimate the local differential information and extracted the feature vertices as the zero-crossing of the curvature derivative.

In an alternative method, Watanabe and Belyaev (2001) extracted features on a polygonal surface by analyzing the focal surface instead of the original mesh. They contend that the focal ribs correspond to the lines on the surface where the principal curvatures have extremes along their associated principal directions and the points where the principal curvatures are equal. Inspired by this observation, Yoshizawa *et al.* (2008) proposed a method for detecting feature lines on meshes.

Another important category is normal vector based methods (Sunil and Pande, 2008). These methods usually identify the features by analyzing the dihedral angle of two triangles sharing an edge (Hubeli and Gross, 2001) or the diversity of the normal in a local region around the current vertex

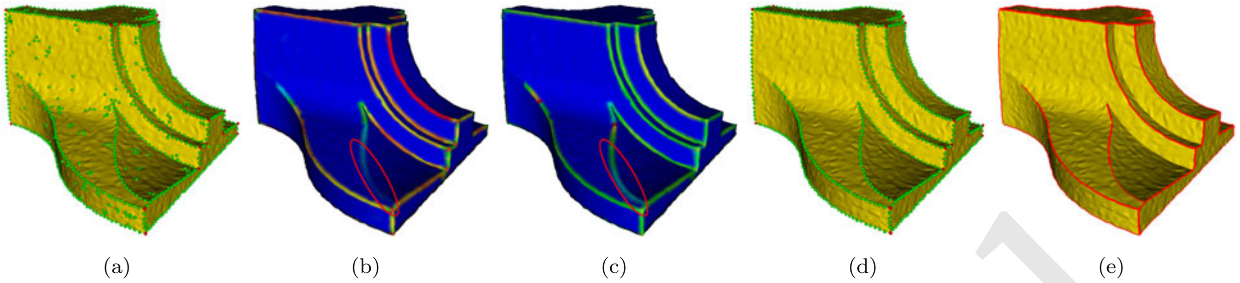


Fig. 1 Method overview. (a) Initial feature vertex detection on a noisy Fandisk model. Sharp edge and corner type vertices are shown in green and red respectively. (b) and (c) The color maps of salient measure before and after weak feature enhancing. The weak features at the top of the fan are marked in red ellipse region. (d) and (e) The final detected feature vertices and feature lines.

(Wang, 2006a,b; Angelo and Stefano, 2010). Page *et al.* (2002) proposed a normal vector voting method for feature detection and curvature estimation on noisy meshes. This method is further used in surface segmentation (Takafumi *et al.*, 2005) and feature detection (Kim *et al.*, 2009; Wang *et al.*, 2011b).

As pointed out in (Page *et al.*, 2002; Kim *et al.*, 2009), the normal tensor voting method can handle sharp features and show robustness to noisy data. It does not involve higher-order derivatives, only first-order differential geometric properties, such as normal, are used. For piecewise-smooth surfaces, the sharp edge and corner vertices can be easily identified. In light of these advantages, the normal tensor voting method is also considered in this paper. There are also some other detection methods, such as (Sahner *et al.*, 2008; Weinkauf and Günther, 2009) based on Morse theory and (Yang *et al.*, 2006; Lai *et al.*, 2007) based on integral invariants.

3 Overview

3.1 Method overview

Given a triangular mesh $\boldsymbol{M} = (V, E, F)$, where $V = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n\}$, denotes the set of vertices, E denotes the set of edges and $F = \{f_1, f_2, \cdots, f_m\}$ denotes the set of faces. Each vertex $\boldsymbol{v}_i \in V$ is represented using Cartesian Coordinates, denoted by $\boldsymbol{v}_i = (v_{ix}, v_{iy}, v_{iz})$. Let $N_f(\boldsymbol{v}_i)$ be the face indices of 1-ring neighbors of $\boldsymbol{v}_i$. Our method involves four main steps:

- Initial feature vertex detection.

  The initial feature vertices are first extracted and classified into different types based on the modified normal tensor voting, see Fig.1a.

- Salient measure computation.

  For each sharp edged type vertex, a novel salient measure is defined according to the Neighbor Supporting. One salient color map is shown in Fig.1b.

- Weak feature enhancing.

  For detecting weak features, a weak feature enhancing technique is implemented. An enhanced salient map and the final detected features are shown in Figs.1c-1d.

- Post-processing.

  The filtered feature vertices can be connected to generate feature lines (see Fig.1e). If there are tough noisy vertices, which may result in tiny feature lines, an optional pruning operation will be conducted.

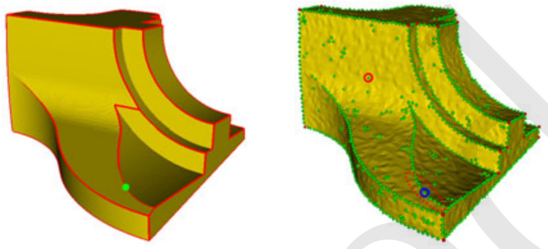
In the first step, to avoid missing any interesting feature, we generate a large initial feature set. This feature set is typically noisy. The second stage includes the remaining three steps, which refine the initial features by employing the novelty defined salient measure, weak feature enhancing and the optional pruning operation.

3.2 Neighbor Supporting

To further enhance the robustness of the normal tensor voting to detect features on noisy meshes, we propose a novel salient measure benefiting from the Neighbor Supporting, which is inspired by the following observation.

A crest point that has maximum curvature in its corresponding direction and a crest line naturally follows the direction of the minimum curvature of its composing crest point (Stylianou and Farin, 2004). Consequently, the feature vertices lie on the principal curvature line. This vertex marked in green lying on a feature line is shown in Fig.2a. In fact, if $\boldsymbol{v}$ is a feature vertex, there will be more feature vertices that can be located in the principal direction or the opposite principal direction corresponding to its smallest principal curvature.

Tracing the located feature vertex's principal direction, we may find more feature vertices lying on a potential feature line. In other words, if $\boldsymbol{v}$ is a feature vertex, there will be a certain number of feature vertices along the principal curvature line to support it. On the contrary, if $\boldsymbol{v}$ is a noisy vertex, there are very few or no feature vertices found by tracing its principal direction. That is, there is weak support or no support from its neighbors, which can be easily seen from Fig.2b. In this paper, this observation is called Neighbor Supporting.



(a) (b)

Fig. 2 (a) A green vertex lying on a feature line. (b) A noisy vertex located in a red circle has no support from neighbors. A real feature vertex located in a blue circle has a strong support from the line neighbors marked in a red dotted oval.

Based on the Neighbor Supporting, we can define a new salient measure for each sharp edge type vertex, $\boldsymbol{v}$ by the integral of Ω with the weight function W along the principal curvature line in a local region:

$$S(\boldsymbol{v}) = \int_l W\,\Omega\; ds, \tag{1}$$

where l is a part of the principal curvature line centered around $\boldsymbol{v}$. Ω is a scalar function, which measures the intrinsic feature intensity of the vertex, and will be defined in Section 4.2. This definition can promote the salient measure of the real features and suppress the noisy data relatively, which can be tested and verified by the experiments subsequently.

4 The proposed method

4.1 Initial feature vertex detection

The initial feature vertices are first detected by the normal tensor voting method with the modified voting weight, which can make the algorithm more robust to irregular tessellated meshes.

4.1.1 Normal tensor voting

The normal voting tensor of a vertex on a triangular mesh can be defined by the unit normal vector of its neighbor triangles (Page *et al.*, 2002). First, the covariance matrix $\boldsymbol{V}_{\boldsymbol{v}}^{f_i}$ of the triangle f_i is written as:

$$\boldsymbol{V}_{\boldsymbol{v}}^{f_i} = \boldsymbol{n}_{f_i}\boldsymbol{n}_{f_i}^T = \begin{pmatrix} a^2 & ab & ac \\ ab & b^2 & bc \\ ac & bc & c^2 \end{pmatrix}, \tag{2}$$

where $\boldsymbol{n}_{f_i} = (a, b, c)^T$ is the unit normal of f_i.

The normal voting tensor of vertex $\boldsymbol{v}$ is defined by:

$$\boldsymbol{T}_{\boldsymbol{v}} = \sum_{f_i \in N_f(\boldsymbol{v})} \mu_{f_i}\boldsymbol{n}_{f_i}\boldsymbol{n}_{f_i}^T, \tag{3}$$

where μ_{f_i} is a weight given by (Kim *et al.*, 2009):

$$\mu_{f_i} = \frac{A(f_i)}{A_{max}} \cdot exp(-\frac{\|\mathbf{c}_{f_i} - \boldsymbol{v}\|}{\sigma/3}), \tag{4}$$

and $A(f_i)$ is the area of triangle f_i, A_{max} is the maximum area among $N_f(\boldsymbol{v})$, $\mathbf{c}_{f_i}$ is the barycenter of triangle f_i, σ is the edge length of a cube that defines the neighboring space of each vertex.

$\boldsymbol{T}_{\boldsymbol{v}}$ is symmetric positive semi-definite and can be represented as:

$$\boldsymbol{T}_{\boldsymbol{v}} = \lambda_1\boldsymbol{e}_1\boldsymbol{e}_1^T + \lambda_2\boldsymbol{e}_2\boldsymbol{e}_2^T + \lambda_3\boldsymbol{e}_3\boldsymbol{e}_3^T, \tag{5}$$

where $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ are its eigen-values and $\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3$ are the corresponding unit eigen-vectors.

4.1.2 Vertex classification

According to the eigen-values (Takafumi *et al.*, 2005; Kim *et al.*, 2009), vertices can be classified into face type, sharp edge type and corner type by the following rules: both sharp edge type and corner type vertices are called feature vertices.

- Face: λ_1 is dominant, and λ_2, λ_3 are close to 0.
- Sharp edge: λ_1, λ_2 are dominant, and λ_3 is close to 0.

- Corner: λ_1, λ_2 and λ_3 are approximately equal.

For some irregular tessellated meshes, the weight function, μ_{f_i} used in Eq.4 does not work as well, such as the detection result shown in left of Fig.3. The reason is that there are triangles with smaller areas that play an important role in model representation. To overcome this shortcoming, the maximum distance between the barycenters of the neighbor triangles and the current vertex is used to control the rate of exponential decay, which increases the weight of the triangle with a closer distance between the barycenter and the current vertex. That is:

$$\mu_{f_i} = \frac{A(f_i)}{A_{max}} \cdot exp(-\frac{\|\mathbf{c}_{f_i} - \boldsymbol{v}\|}{max(\|\mathbf{c}_f - \boldsymbol{v}\|)}), \qquad (6)$$

where $f \in N_f(\boldsymbol{v})$. Adopting the modified weight, more reasonable results can be obtained, such as the result shown in the right of Fig.3.

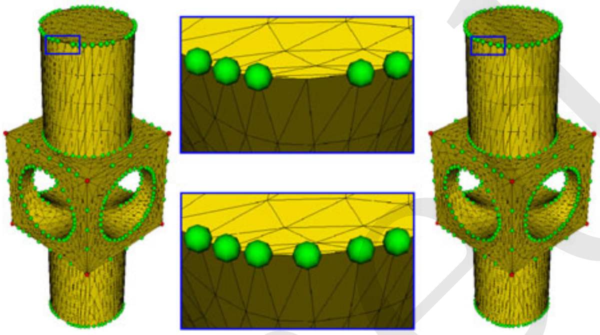


Fig. 3 Left and right pictures are the detection result of the Mechanical Part using the weight of Eq.4 and Eq.6 respectively. The close-up of the detection results are shown in the middle pictures.

As mentioned above, corners have no preferred orientation. Fortunately, they can be identified by the normal tensor voting method in advance and handled with special care. In the following, we turn our attention to sharp edge type vertices.

4.2 Salient measure computation

Many feature detecting approaches measure the salient of a vertex by computing the difference of some properties with its neighbors, such as normal or principal curvature. Usually, the center-surrounding neighbors are used (Lee *et al.*, 2005; Liu *et al.*, 2007; Mao *et al.*, 2009). Differently, in our work the salient of a vertex is measured by collecting the intrinsic feature intensity from its supporting neighbors. Certainly, it is not the center-surrounding neighbors, but the anisotropic line style neighbors, which can reflect the line features more effectively.

To compute the salient measure for each sharp edge type vertex according to the Eq.1, two essential ingredients should be determined. One is the initial measure, Ω the other is the integral direction, $\boldsymbol{t}$.

For each vertex, there are three eigen-values λ_1, λ_2 and λ_3 and eigen-vectors $\boldsymbol{e}_1$, $\boldsymbol{e}_2$ and $\boldsymbol{e}_3$. Before the Ω is defined, λ_1, λ_2 and λ_3 are put into a vector and normalized first. The vertices can be classified into different types according to their eigen-values, Ω can be constructed as:

$$\Omega = \frac{\lambda_1 + \lambda_2 + \lambda_3}{2} - \frac{1}{2}, \qquad (7)$$

which measures the intrinsic feature intensity of each vertex. In fact, the magnitude of this measure is large for sharp edge and corner type vertices. On the contrary, it is small for face type vertices. For instance, the Ω of a cube model is shown in Table.1. After the Ω of all sharp type feature vertices are computed, we normalize them to [0 1], which allows us to set coarse thresholds valid for most models.

Table 1 Eigen-values and Ω of a cube model

Type	λ_1	λ_2	λ_3	Ω
Face	1	0	0	0
Edge	0.7071	0.7071	0	0.2071
Corner	0.5774	0.5774	0.5774	0.3661

For the integral direction $\boldsymbol{t}$, following the statement of (Moreno *et al.*, 2011), ideally, if a point belongs to a curve, the third eigen-vector of its tensor must be aligned with the tangent to the curve at that point, and λ_3 must be zero. Thus, $\boldsymbol{t}$ can be naturally initialized by $\boldsymbol{e}_3$, which is called feature direction in this paper.

At this point, we have the initial measure Ω and the feature direction $\boldsymbol{t}$. According to Eq.1, for each sharp edge type vertex $\boldsymbol{v}$, the salient measure is computed by:

$$S(\boldsymbol{v}) = \sum_{\boldsymbol{v}_i \in N(\boldsymbol{v})} W(\boldsymbol{v}_i)\, \Omega(\boldsymbol{v}_i), \qquad (8)$$

where $N(\boldsymbol{v})$ represents the supporting neighbors of $\boldsymbol{v}$

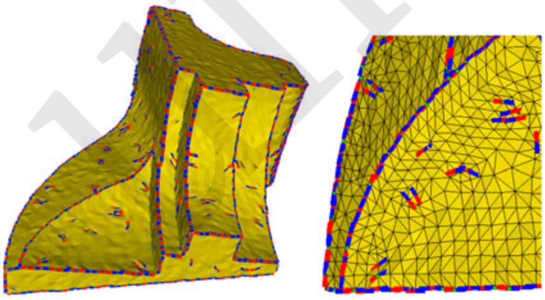
and the weight is:

$$W(\boldsymbol{v}_i) = exp(-\frac{\|\boldsymbol{v} - \boldsymbol{v}_i\|}{2\delta^2}), \tag{9}$$

where δ is 1.5 times of the mean edge length of the mesh. Generally speaking, the new salient measure of a feature vertex is cast by the initial salient measure of the supporting neighbors, which is more robust than the initial salient measure.

The supporting neighbors $N(\boldsymbol{v})$ can be constructed by the following strategy. First, $\boldsymbol{v}$ is put into $N(\boldsymbol{v})$ as a front vertex. Then, following its feature direction $\boldsymbol{t}$, we may find one or no feature vertex in its one-ring vertex. The feature vertex can be selected as a new front, if the intersecting angle between the vertex's feature direction and $\boldsymbol{t}$ is smaller than a specified threshold, such as 15^o. At the same time, the vertex should have a smaller angle than other candidates. If there are two feature vertices with the same smallest angle, the one with the smaller distance to $\boldsymbol{v}$ is selected. If there are still two feature vertices that satisfy the above conditions, each of them can be selected as the new front. In practice, this situation rarely occurs in our experiments.

In the same way, one or no feature vertex can be found in opposite direction $-\boldsymbol{t}$. This process can be easily observed in Fig.4, where the feature directions of the initial detected features are shown. The obtained sharp edge type vertex is marked as a new front and the procedure is continued until the maximum number, K of the supporting neighbor $N(\boldsymbol{v})$ is reached or no feature vertex can be found. K is the smallest length of the feature segment to be detected. In our experiments, K was always set to 5.



(a) (b)

Fig. 4 (a) Each initial feature vertex is shown in green. Feature direction and its opposite direction are shown in red and blue, respectively. (b) The zoom-in view of the part of the model.

For corner type vertices, the salient measure can be set by the average value of the top 20% sharp edge type vertices' measure. The measure of the face type vertices are naturally set to zeros. The salient measure of the real feature vertices is promoted owing to the Neighbor Supporting while the salient measure of noisy vertices are suppressed relatively. It is helpful to filter the noisy data (with smaller salient measure) in an effective way by setting a proper threshold φ, which will be discussed in Section 5.1.

4.3 Weak feature enhancing

Usually, sharp features are sufficient for some applications, but in some cases weak features are required. However, it is hard to distinguish them from noise signals. Although the Neighbor Supporting can promote the salient measure of the weak features to some extent, some weaker features may still be filtered out.

To solve this issue, before the filtering process, the weak feature enhancing is preformed. A feature vertex $\boldsymbol{v}$ can be identified as a weak feature if the number of the elements in $N(\boldsymbol{v})$ is a larger value, such as larger than 3 when K is set to 5. While the $\Omega(\boldsymbol{v})$ and the average Ω of the supporting neighbors are small, such as smaller than 0.45. To avoid the weak features from being filtered out with the noise, the salient measure $S(\boldsymbol{v})$ of the identified weak features are promoted to a higher value by multiplying the adaptive weight:

$$S(\boldsymbol{v}) \leftarrow K \cdot exp(-\Omega(\boldsymbol{v})) \cdot S(\boldsymbol{v}). \tag{10}$$

The salient measure of the weak feature is promoted by the weak feature enhancing, which further enlarges the gap of the salient measure between the real features and pseudo features. The effect of the weak feature enhancing is evident in Fig.1c, in which the salient map of the weak features (marked in red ellipse) becomes much clearer after the weak feature enhancing.

4.4 Post-processing

In this section, the filtered sharp edge type feature vertices and corners will be connected to generate feature lines. For some models, due to the larger scale noise, there might be a few tough noisy vertices left, which may result in some small branches or tiny lines. It is desirable to prune them out by an optional branch pruning process.

4.4.1 Connecting feature vertices

To generate feature lines, the method used in (Ohtake *et al.*, 2004) is modified and employed here. Firstly, if two feature vertices are detected on a triangle, they are connected by a straight segment. Secondly, if a triangle contains three feature vertices, according to (Ohtake *et al.*, 2004), the vertices are connected with the centroid of the triangle formed by the vertices. In this paper, if one of them is a corner, the priority is accorded to it, other sharp edge type vertices are just connected to the corner, and no straight segments are drawn between sharp edge type vertices again. Thirdly, if a feature vertex has no other feature vertices connected to it, it is treated as a noisy vertex and deleted. By adopting this procedure, some noisy corners or sharp edge type vertices may be further filtered out.

4.4.2 Branches pruning

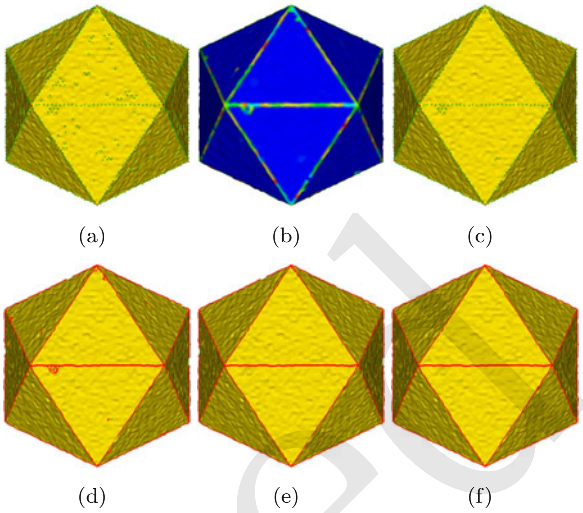
Although an effective feature filtering has been implemented, there still may be few tough noise signals, which may result in some tiny branches, as shown in Fig.5d. The reason is obvious from Figs.5a-5c, where the noisy vertices are not only close to the real feature lines, but also cluster in small groups and support each other. To obtain satisfactory results, the pruning algorithm mentioned in (Demarsin *et al.*, 2007) is used. However, only the length of feature line and not the intensity measure was in (Demarsin *et al.*, 2007). Before the pruning algorithm is executed, an edge filtering is carried out by setting a suitable threshold ψ according to the edge intensity measure ST for each feature edge $[\boldsymbol{v}_j \;\; \boldsymbol{v}_k]$,

$$ST(jk) = S(\boldsymbol{v}_j) \cdot S(\boldsymbol{v}_k) \cdot \#N(\boldsymbol{v}_j) \cdot \#N(\boldsymbol{v}_k), \quad (11)$$

where $\boldsymbol{v}_j$ and $\boldsymbol{v}_k$ are the connected feature vertices of a feature line. $S(\boldsymbol{v}_j)$ is the salient measure of $\boldsymbol{v}_j$ and $\#N(\boldsymbol{v}_j)$ is the number of the supporting neighbors of $\boldsymbol{v}_j$. After the pruning process, the tiny branched edges are filtered out, as shown in Fig.5f.

5 Implementations and results

In this section, we test the proposed method on various models. Some parameters are discussed first, and then the detected results as well as applications are shown subsequently.


(a) (b) (c)

(d) (e) (f)

Fig. 5 Branches pruning. (a) Initial detected feature vertices. (b) Color map of the salient measure. (c) Feature vertices after filtering. (d) Initial feature lines. (e) Feature lines after the branches pruning. (f) Final feature lines.

5.1 Parameters

In the initial feature detection, λ_2 and λ_3 play a important roles in vertex classification. The strategy shown in **Algorithm** 1 is adopted.

Algorithm 1 Feature vertex classification

```
// $\#(V)$ is the number of vertex
// $FaceV$, $SharpV$ and $CornerV$ are the index set of
    face, sharp edge and corner type vertices, respectively.
for $i \leftarrow 1$ to $\#(V)$ do
    $\lambda_1, \lambda_2, \lambda_3$ are initialized
    if $\lambda_3 \le \alpha$ then
        if $\lambda_2 \le \beta$ then
            $FaceV \leftarrow [FaceV \;\; i]$;
        else
            $SharpV \leftarrow [SharpV \;\; i]$;
        end if
    else
        $CornerV \leftarrow [CornerV \;\; i]$;
    end if
end for
```

For noise-free models with salient features, such as a cube, the detection result is insensitive to thresholds selecting. However, for detecting some weak features, β is usually set to a smaller number, such as in Fig.9g, $\alpha = 0.055$, $\beta = 0.025$ were used. But for noisy models, α should be larger to avoid extracting many false corners, which are undesired. β is a fine-tuning parameter around a value, e.g., 0.05, for finding a tradeoff between detecting weak features and the extra number of noisy vertices due to setting a smaller β. In Fig.10, for a larger scale noisy model, $\alpha = 0.2$, $\beta = 0.045$ produce reasonable result.

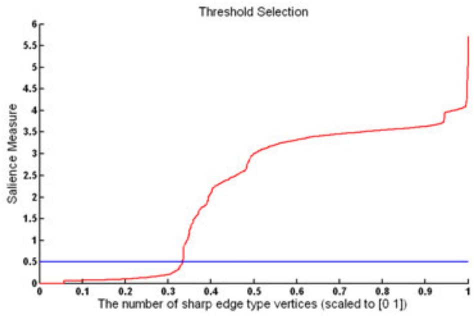


Fig. 6 The salient measure of the detected features are sorted and shown in red curve and a settled threshold 0.5 **is shown in straight blue line.**

Next, in noisy data filtering and the branch pruning, two proper thresholds φ and ψ should be set. In the literature, many algorithms select thresholds in a trial-and-error way; although, it is tedious and time-consuming. In our experiments, immediate visual feedback is allows the selection of a proper threshold. In the test of Fig.1, a fast visual feedback of the salient measure and a settled threshold are shown in Fig.6.

5.2 Experiments results

After the discussion of parameters setting, we present experimental results on various noise-free and noisy models to demonstrate the performance of our method, and then apply the detected features in feature-preserving mesh denoising and hole-filling.

5.2.1 Noise-free models

Fig.7 shows the feature detection result of the Mechanical bin. For the noise-free case, the feature vertices can be extracted successfully and the feature lines are clearly shown. The optional branches pruning does not need to be conducted. In the same way, for a noise-free Octa-flower, near perfect detected results are shown in Fig.8.

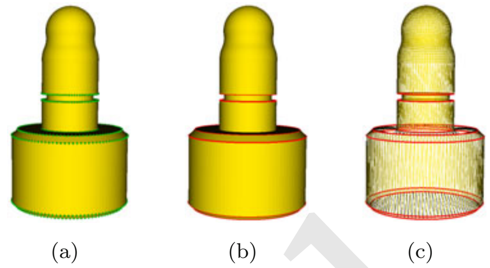


Fig. 7 Mechanical bin. (a) Detected feature vertices. (b) and (c) Feature lines.

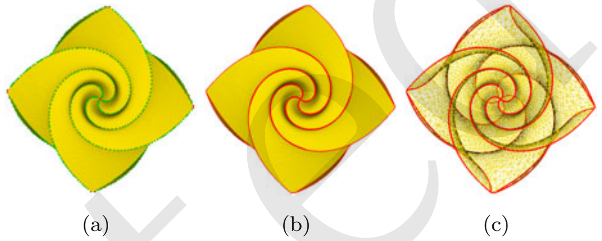


Fig. 8 Octa-flower. (a) Detected feature vertices. (b) and (c) Feature lines.

5.2.2 Noisy models

To demonstrate the ability of our method in detecting features on noisy models, in Fig.9, we compare our method with Mao *et al.* (2009); Wang (2006b); Angelo and Stefano (2010) via Fandisk model contained with the Gaussian noise (standard derivation is 5% of the mean edge length of the mesh). From the results, we can see that Mao *et al.* (2009) gets a blurred color map over the feature lines in Fig.9b for the isotropic neighbors adopted. Although clearer color maps of the feature salient can be obtained by Wang (2006b) and Angelo and Stefano (2010) in Fig.9c and Fig.9d, the maps of the weak features are missing. Comparing to Figs.9b-9d, our method generates a clearer map of salient measure in Fig.9d, especially for the weak feature at the top of the fan. It is evident from Figs.9g-9j, our method gives rise to satisfactory results, while other methods result in missing weak features.

Table 2 Parameters

Model	Noise	α	β	φ	ψ
Fig.10a	8%	0.2	0.045	0.4	0.8
Fig.10b	8%	0.2	0.045	0.4	3.9
Fig.10c	8%	0.055	0.05	0.4	3.9
Fig.10d	5%	0.05	0.06	0.3	3.9
Fig.10e	5%	0.3	0.4	1.2	18

More satisfactory detection results with larger scale noise are shown in Fig.10, such as the detection result of another Fandisk model is shown in Fig.10 (standard derivation is 8% of the mean edge length of the mesh). Although so many pseudo features together with real features may be detected for the

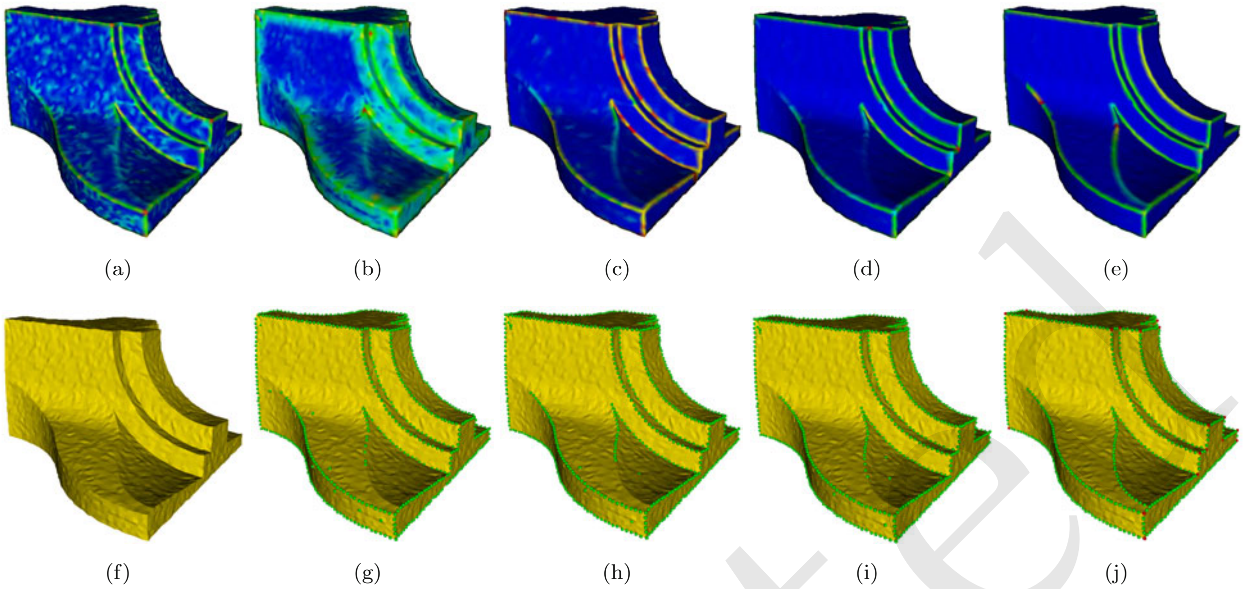


Fig. 9 Comparisons. (a) Color map of Gaussian curvature of (f), computed by Taubin (1995). (b-e) Color maps of the salient measure corresponding to Mao *et al.* (2009); Wang (2006b); Angelo and Stefano (2010) and our method, respectively. (f) Noisy Fandisk model with standard derivation is 5% of the mean edge length of the mesh. (g) Result of Mao *et al.* (2009) with $T_{max} = 98\%$, $T_{min} = 90\%$, $S_{max} = 86\%$ and $S_{min} = 65\%$. (h) Result of Wang (2006b) with $\kappa = 6$ and one-ring neighbor is used. (i) Result of Angelo and Stefano (2010) with the best selected parameters. (j) Our result with $\alpha = 0.055$, $\beta = 0.025$, $\varphi = 0.5$ and $\psi = 0.7$.

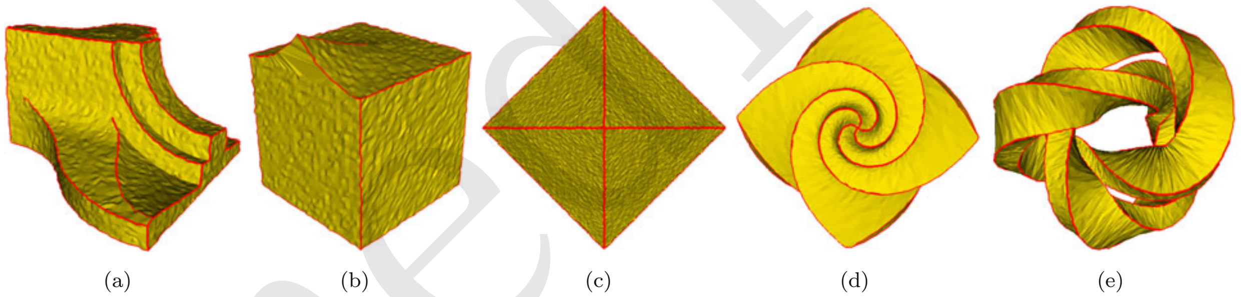


Fig. 10 More results. (a) Fandisk. (b) Smooth-feature. (c) Octahedron. (d) Octa-flower. (e) Twist model.

large scale noise, most of them can be effectively filtered out and satisfactory results can be obtained. The details of the standard derivations of the Gaussian noise and the used parameters are listed in Table.2.

5.2.3 Feature-preserving mesh denoising

As mentioned at the beginning of the article, various applications will benefit from the robustness feature detection, such as mesh denoising. A mesh denosing method based on differential coordinates is proposed in (Su *et al.*, 2009). It first smooths the Laplacian coordinates by the classical mean filter, and then reconstructs the new Cartesian coordinates to fit the smoothed Laplacian coordinates with face barycenter constraints. Thus, the three linear systems below are solved.

$$\boldsymbol{A}\boldsymbol{V}_d^{'} = \begin{pmatrix} \boldsymbol{L} \\ \boldsymbol{F} \end{pmatrix} \boldsymbol{V}_d^{'} = \begin{pmatrix} \boldsymbol{\delta}_d^{'} \\ \boldsymbol{b}_d \end{pmatrix} = \boldsymbol{B}_d, d \in \{x, y, z\}, \tag{12}$$

where $\boldsymbol{L}$ is the Laplacian matrix of the mesh, and $\boldsymbol{V}_d^{'}$ is the Cartesian coordinates of the reconstructed mesh, $\boldsymbol{F}$ is an $m \times n$ matrix in which the k-th row contains only three non-zero elements to constrain the position of barycenter of the corresponding face $f_k = (r, s, t)$ with elements:

$$\boldsymbol{F}_{kj} = \begin{cases} \frac{1}{3}, & j \in \{r, s, t\} \\ 0, & otherwise \end{cases} \tag{13}$$

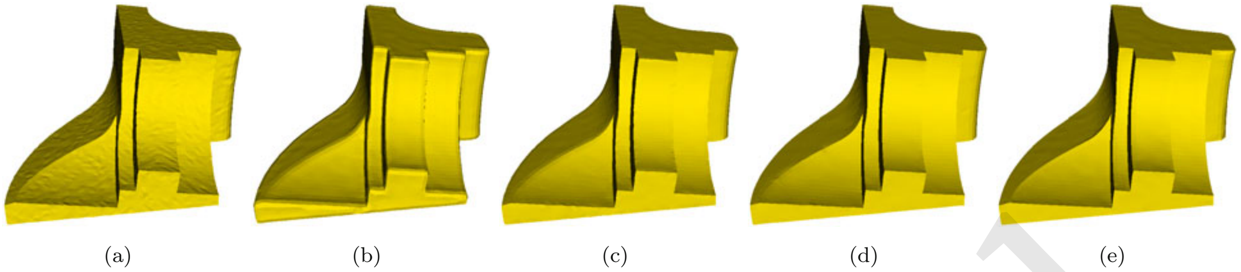


Fig. 11 Feature-preserving mesh denoising. (a) Fandisk model with Gaussian noise (the standard variance is 3% **of the mean edge length of the mesh). (b) Smoothed result of (Su *et al.*, 2009) in** 3 **iterations by using cotangent weights. (c) Smoothed result after the bilateral filter in (Fleishman *et al.*, 2003) in 3 iterations with** $\rho = 2.5\eta$, $\delta_c = 1.2\eta$ **and** $\delta_s = 0.15\eta$, **where** η **is the mean edge length of the mesh. (d) Smoothed result after performing bilateral filter to Laplacian coordinates 3 iteration with** $a = 2.0$ **and** $b = 1.0$ **in (Wang *et al.*, 2011a). (e) Smoothed result with feature vertex constraints based on the scheme of (Su *et al.*, 2009) with cotangent weights in** 3 **iterations and** $\mu = 0.1$**.**

with $1 \le k \le m, 1 \le j \le n$, $\boldsymbol{b}_d$ is an m-vector with elements

$$\boldsymbol{b}_{dk} = \frac{1}{3}(v_{rd} + v_{sd} + v_{td}), f_k = (r, s, t), d \in \{x, y, z\}, \tag{14}$$

with $1 \le k \le m$.

$\boldsymbol{\delta}'_d = (\delta'_{1d}, \delta'_{2d}, \cdots, \delta'_{nd})^T, d \in \{x, y, z\}$ is the n-vector containing the x, y or z smoothed Laplacian coordinates of the n vertices and the cotangent weight is used in our experiments.

However, the mean filter is an isotropic method. It cannot effectively preserve the sharp features, this is seen in Fig.11b. In order to preserve the sharp features and overcome the shortcoming of the isotropic method, the detected features are used as constraints in the mesh reconstruction. The following linear systems will be solved:

$$\boldsymbol{A}\boldsymbol{V}'_d = \begin{pmatrix} \boldsymbol{L} \\ \boldsymbol{F} \\ \boldsymbol{C} \end{pmatrix} \boldsymbol{V}'_d = \begin{pmatrix} \boldsymbol{\delta}''_d \\ \boldsymbol{b}_d \\ \boldsymbol{c}_d \end{pmatrix} = \boldsymbol{B}_d, d \in \{x, y, z\}, \tag{15}$$

where $\boldsymbol{C}$ is an $w \times n$ matrix in which each row contains only one non-zero element used to constrain the position of the detected feature vertex with the element:

$$\boldsymbol{C}_{kj} = \begin{cases} \mu, & j \in FT \\ 0, & otherwise \end{cases} \tag{16}$$

with $1 \le k \le w$, and $FT = \{i_1, i_2, \cdots, i_w\}$ is the index set of the detected feature vertices. Usually, $\mu = 0.1$ is set.

$\boldsymbol{c}_d$ is a $w \times 1$ column vector of the product of feature vertices and μ: $\boldsymbol{c}_{dk} = \mu \boldsymbol{v}_{di_k}, 1 \le k \le w, d \in \{x, y, z\}$. And $\boldsymbol{\delta}''_d$ is constructed as:

$$\boldsymbol{\delta}''_{jd} = \begin{cases} (4\boldsymbol{\delta}'_{jd} + \boldsymbol{\delta}_{jd})/5, & if\ j \in FT \\ \boldsymbol{\delta}'_{jd}, & otherwise \end{cases} \tag{17}$$

where $\boldsymbol{\delta}'_{jd}$ and $\boldsymbol{\delta}_{jd}$ are the smoothed and original Laplacian coordinates of the vertices at each iteration.

The Cartesian coordinates of the smoothed mesh can be found by solving the least-square problems in Eq.15 as:

$$\boldsymbol{V}'_d = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{B}_d, d \in \{x, y, z\}. \tag{18}$$

Fig.11a is a noisy Fandisk model. Fig.11b is the result of Su *et al.* (2009), where the sharp features are obviously blurred due to the isotropic mean filter. The improved results can be obtained by the bilateral filter (Fleishman *et al.*, 2003) and bilateral filter applied on the Laplacian coordinates (Wang *et al.*, 2011a), which are shown in Fig.11c and Fig.11d, respectively. Although the sharp features are well preserved in Fig.11c and Fig.11d, the weak features at the top fan of the model is blurred to some extent. Thanks to the effective weak feature detection of our method, the noise of the Fandisk is effectively filtered out, while the geometry features are well preserved, especially the weak features at the top of the fan, as shown in Fig.11e.

5.2.4 Feature-preserving hole-filling

Hole-filling is a preliminary work and has received much attention in recent years. Most of them work well for smaller holes located on smooth regions. However, it is still a challenge to fill large and complex holes with some missing sharp features. In Chen and Cheng (2008), an iterative sharpness dependent filtering is adopted to recover the missing sharp features by adjusting the normal and the positions of the initially filled mesh. For the position of the features not predicted, this implicit feature recovery method may not work well in some cases, such as the result shown in Fig.12c.

Feature detection plays an important role in feature-preserving hole-filling. If features around the hole are explicitly extracted, the positions of the missing features can be inferred more accurately. The detected features are first matched into different feature sets to construct the missing features curves, which divide the original hole into some simple sub-holes. Then each sub-hole is filled separately, and the reconstructed feature curves are explicitly preserved. For more details the reader is referred to Wang *et al.* (2011c). In Fig.12d, we can see that the missing sharp features are successfully reconstructed with the help of detected features.

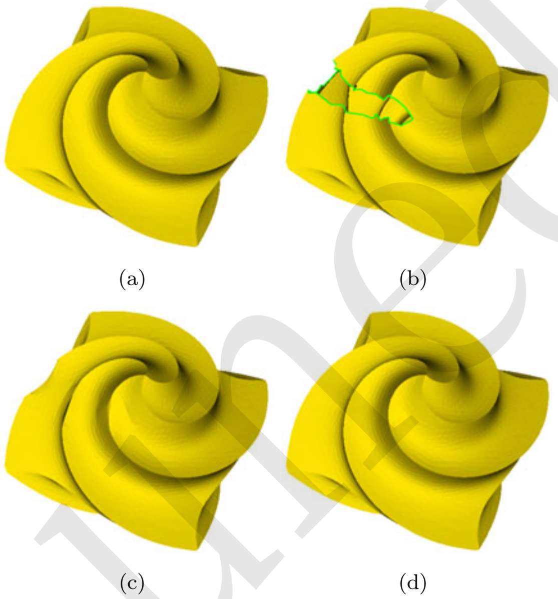

(a) (b) (c) (d)

Fig. 12 Feature-preserving hole-filling. (a) Octa-flower model. (b) A hole with sharp features missing. (c) Result of Chen and Cheng (2008). (d) Result of Wang *et al.* (2011c) using the detected features.

6 Conclusions and future work

In this paper, the problem of feature detection on triangular meshes with a focus on the feature detection of noisy models has been discussed. We present a simple and robust method. In comparison with previous works, our method can effectively preserve the weak features while filtering out the noisy data owing to the novel salient measure and the weak feature enhancing. More feature-based algorithms can be implemented in computer-aided design and computer graphics, such as the feature-preserving mesh denoising and hole-filling presented in this paper.

When the initial detected feature vertices are grouped into clusters, the presented approach may provide unsatisfactory performance for the noisy vertices supporting each other. Solutions to overcome this limitation, as well as plans to extend the idea of Neighbor Supporting to detect features on point clouds, are the next subjects of research in our group.

Acknowledgements

We would like to thank the anonymous reviewers for their help in improving this work. The models in this paper are provided by the courtesy of AIM@SHAPE Repository and the collections of Hugues Hoppe, and others.

References

Angelo, L.D., Stefano, P.D., 2010. c^1 continuities detection in triangular meshes. *Comput. Aided Des.*, **42**(9):828-839. [doi:10.1016/j.cad.2010.05.005]

Bian, Z., Tong, R.F., 2011. feature-preserving mesh denoising based on vertices classification. *Comput. Aided Des.*, **28**(1):50-64. [doi:10.1016/j.cagd.2010.10.001]

Chen, C.Y., Cheng, K.Y., 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.

Demarsin, K., Vanderstraeten, D., Volodine, T., Roose, D., 2007. detection of closed sharp edges in point clouds using normal estimation and graph theory. *Comput. Aided Des.*, **39**(4):276-283. [doi:10.1016/j.cad.2006.12.005]

Fan, H.Q., Yu, Y.Z., Peng, Q.S., 2010. robust feature-preserving mesh denoising based on consistent subneighborhoods. *IEEE Transactions on Visualization and Computer Graphics*, **16**(2):312-324. [doi:10.1109/TVCG.2009.70]

Fleishman, S., Drori, I., Cohen-Or, D., 2003. Bilateral mesh denoising. SIGGRAPH, p.950-953. [doi:10.1145/882262.882368]

Hildebrandt, K., Polthier, K., Wardetzky, M., 2005. Smooth feature lines on surface meshes. SGP, p.85-90.

Hubeli, A., Gross, M., 2001. Multiresolution feature extraction for unstructured meshes. Proceedings of the Conference on Visualization, p.287-294.

Kim, H.S., Choi, H.K., Lee, K.H., 2009. feature detection of triangular meshes based on tensor voting theory. *Comput. Aided Des.*, **41**(1):47-58. [doi:10.1016/j.cad.2008.12.003]

Kim, S.K., Kim, C.H., 2006. finding ridges and valleys in a discrete surface using a modified mls approximation. *Comput. Aided Des.*, **38**(2):173-180. [doi:10.1016/j.cad.2005.05.002]

Kim, S.K., Kim, S.J., Kim, C.H., 2006. Extraction of ridges-valleys for feature-preserving simplification of polygonal models. International Conference on Computational Science (2), p.279-286. [doi:10.1007/11758525_37]

Lai, Y.K., Zhou, Q.Y., Hu, S.M., Wallner, J., Pottmann, H., 2007. robust feature classification and editing. *IEEE Transactions on Visualization and Computer Graphics*, **13**(1):34-45. [doi:10.1109/TVCG.2007.19]

Lee, C.H., Varshney, A., Jacobs, D.W., 2005. Mesh saliency. SIGGRAPH, p.659-666. [doi:10.1145/1073204.1073244]

Li, Z., Meek, D. S., Walton, D. J., 2010. polynomial blending in a mesh hole-filling application. *Comput. Aided Des.*, **42**(4):340-349. [doi:10.1016/j.cad.2009.12.006]

Liu, Y.S., Liu, M., Kihara, D., Ramani, K., 2007. Salient critical points for meshes. SPM, p.277-282. [doi:10.1145/1236246.1236285]

Mao, Z.H., Guo, G., Zhao, M.X., 2009. robust detection of perceptually salient features on 3d meshes. *Vis. Comput.*, **25**(3):289-295. [doi:10.1007/s00371-008-0268-2]

Moreno, R., Garcia, M.A., Puig, D., Pizarro, L., Burgeth, B., Weickert, J., 2011. On improving the efficiency of tensor voting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Ohtake, Y., Belyaev, A., Seidel, H.P., 2004. ridge-valley lines on meshes via implicit surface fitting. *ACM Trans. Graph.*, **23**(3):609-612. [doi:10.1145/1015706.1015768]

Page, D. L., Sun, Y., Koschan, A. F., Paik, J., Abidi, M. A., 2002. normal vector voting: crease detection and curvature estimation on large, noisy meshes. *Graphical Models*, **64**(3-4):199-229. [doi:10.1006/gmod.2002.0574]

Sahner, J., Weber, B., Prohaska, S., Lamecker, H., 2008. extraction of feature lines on surface meshes based on discrete morse theory. *Comput. Graph. Forum*, **27**(3):735-742. [doi:10.1111/j.1467-8659.2008.01202.x]

Stylianou, G., Farin, G., 2004. crest lines for surface segmentation and flattening. *IEEE Transactions on Visualization and Computer Graphics*, **10**(5):536-544. [doi:10.1109/CAD-CG.2005.10]

Su, Z.X., Wang, H., Cao, J.J., 2009. Mesh denoising based on differential coordinates. SMI, p.1-6. [doi:10.1109/SMI.2009.5170156]

Sunil, V.B., Pande, S.S., 2008. automatic recognition of features from freeform surface CAD models. *Comput. Aided Des.*, **40**(4):502-517. [doi:10.1016/j.cad.2008.01.006]

Takafumi, S., Hiroaki, D., Satoshi, K., Takeshi, K., 2005. A new bilateral mesh smoothing method by recognizing features. CAD/CG, p.281-286. [doi:10.1109/CAD-CG.2005.10]

Taubin, G., 1995. Estimating the tensor of curvature of a surface from a polyhedral approximation. ICCV, p.902-907.

Wang, C.C.L., 2006a. bilateral recovering of sharp edges on feature-insensitive sampled meshes. *IEEE Transactions on Visualization and Computer Graphics*, **12**(4):629-639. [doi:10.1109/TVCG.2006.60]

Wang, C.C.L., 2006b. incremental reconstruction of sharp edges on mesh surfaces. *Comput. Aided Des.*, **38**(6):689-702. [doi:10.1016/j.cad.2006.02.009]

Wang, H., Chen, H.Y., Su, Z.X., Cao, J.J., Liu, F.S., Shi, X.Q., 2011a. versatile surface detail editing via laplacian coordinates. *Vis. Comput.*, **27**(5):401-411. [doi:10.1007/s00371-011-0558-y]

Wang, S.F., Hou, T.B., Su, Z.X., Qin, H., 2011b. Diffusion tensor weighted harmonic fields for feature classification. PG, p.93-98. [doi:10.2312/PE/PG/PG2011short/093-098]

Wang, X.C., Liu, X.P., Lu, L.F., Li, B.J., Cao, J.J., Yin, B.C., Shi, X.Q., 2011c. Automatic hole-filling of CAD model with feature-preserving. *Comput. Graph.* (in press). [doi:10.1016/j.cag.2011.12.007]

Watanabe, K., Belyaev, A.G., 2001. detection of salient curvature features on polygonal surfaces. *Comput. Graph. Forum*, **20**(3):385-392. [doi:10.1111/1467-8659.00531]

Weinkauf, T., Günther, D., 2009. separatrix persistence: extraction of salient edges on surfaces using topological methods. *Comput. Graph. Forum*, **28**(5):1519-1528. [doi:10.1111/j.1467-8659.2009.01528.x]

Yang, Y.L., Lai, Y.K., Hu, S.M., Pottmann, H., 2006. Robust principal curvatures on multiple scales. Proceedings of the fourth Eurographics symposium on Geometry processing, p.223-226.

Yoshizawa, S., Belyaev, A., Seidel, H.P., 2005. Fast and robust detection of crest lines on meshes. SPM'05, p.227-232. [doi:10.1145/1060244.1060270]

Yoshizawa, S., Belyaev, A., Yokota, H., Seidel, H.P., 2008. fast, robust, and faithful methods for detecting crest lines on meshes. *Comput. Aided Geom. Des.*, **25**(8). [doi:10.1016/j.cagd.2008.06.008]