ESTIMATING THE CREST LINES ON POLYGONAL MESH MODELS BY AN AUTOMATIC THRESHOLD

Zhihong Mao, Ruichao Wang, Fan Liu and Yulin Zhou

Division of Intelligent Manufacturing, Wuyi University, Jiangmen529020, China

ABSTRACT

Crest lines convey the inherent features of the shape. Mathematically Crest lines are described via extremes of the surface principal curvatures along their corresponding lines of curvature. In this study we used an automatic threshold estimation technique to estimate crest lines. We firstly computed the principal curvature and corresponding direction for each vertex in the mesh; then we computed the saliency value by a linear combination of the maximal absolute curvature and the absolute curvature difference; finally, we automatically determine the threshold to detect the crest lines according to the saliency value. For illustrative purpose, we demonstrated our method with several examples.

KEY WORDS

Crest Lines, Polygonal Mesh Models, Curvature

1. INTRODUCTION

Digital geometry processing (DGP) is a new branch of computer graphics, which analyses and processes the polygonal mesh models. Estimating the crest lines is an important area of DGP. Many applications in mesh processing and analysis require the detection of crest lines, such as simplification of 3D models, shape matching and shape editing of 3D models, 3D animations, Medical imaging. Crest lines convey the inherent features of the shape. Mathematically Crest lines are described via extremes of the surface principal curvatures along their corresponding lines of curvature, which are traced by using high-order curvature derivatives.

Recent research in 3D computer graphics has focused on estimating the crest lines over triangle mesh models [1-6]. The basic idea of these papers is first to robustly estimate surface principal curvature and then to identify the crest lines as lines of curvature extremes. However, most 3D models are collections of discrete triangular facets, which are not continuous parameterized. Moreover, natural objects have various morphologies. So it is very difficult to find efficient functions to describe these shapes, which makes it difficult to detect the crest lines on discrete triangular mesh.

We present an automatic threshold estimation technique for robustly extracting crest lines on polygonal mesh models based on the observation that polygonal surface models with features usually contain many flat regions and a little sharp edges. Firstly, our method computed the principal curvature and corresponding direction for each vertex in triangular mesh models; then the method computed the saliency value [7, 8] by a linear combination of the maximal absolute

DOI: 10.5121/ijcsit.2019.11504

curvature and the absolute curvature difference, which can identify regions that are different from their surrounding

context and reduce ambiguity in noisy circumstances; Finally, our method automatically determines the threshold to detect the crest lines according to the saliency value.

2. RELATED WORK

Crest lines, sometimes called ridges and valleys or Feature lines, are invariant of transformation, rotation and scale. They are the basis of mesh shape processing such as shape recognition, matching and segmentation. With the continuous improvement of 3D scanning technology the model becomes more and more detailed, making crest lines extraction techniques increasingly popular in computer graphics [16, 17, 23]. Lee presented an image-based method to extract crest lines on a rendered projection image by edge detection algorithm in image processing [16].

Although the algorithm is visually more pleasant and less computational complexity, the accuracy of pixel-based representation method is lower. Model-based method extracts lines directly on polygonal mesh models based on curvature, the differential geometric property of the surface. Briefly, Curvature estimation methods can be categorized as follows: 1) Normal curvature approximation method where the Weingarten matrix is used to estimate the principal curvature and direction [9, 10]. 2) Surface (curve) fitting method where a polynomial surface (curve) is fitted to points in a local region [11, 12]. 3) Discrete differential geometry method [13, 14] where discrete versions of differential geometry theorems, such as the Laplace-Beltrami operator and Gauss-Bonnet theorem, are developed and applied to the neighbourhood of each vertex. Now model-based method dominates the field of 3D computer graphics. There are a number of lines that serve to extract the geometric linear features of the surface. The first class is view-dependent curves. The silhouette (contour) is the curves that are only defined with respect to a viewing screen. Suggestive contours are contours that would first appear with a minimal change in viewpoint [18]. Apparent ridges are defined as the ridges of view-dependent curvature, the variation of surface normal with respect to a viewing screen plane [6]. View-dependent curves depend on the viewing direction and produce visually pleasing line drawings. So they are usually used for non-photorealistic rendering methods. The second class is view-independent curves that depend only on the differential geometric properties of the surface. Many researchers have tried to depict the shape of the 3D model with a high-quality crest lines. But the features are traced using high-order curvature derivatives and are not easy to be detected from discrete surface. The most common curves are ridges and valleys which occur at points of extremal principal curvatures. This paper focuses on the problem of accurately detecting crest lines on surfaces. Ohtake et al. [2] proposed a method for detecting ridge-valley lines by combining multi-level implicit surface fitting and finite difference approximations. Because the method involves computing curvature tensors and their derivatives at each vertex, it is time-consuming. Yoshizawa [3] detected the crest lines based on a modification of Ohtake's method. The method reduced the computation times since Yoshizawa's method estimated the surface derivatives via local polynomial fitting. Soo-Kyun Kim [4] found ridges and valleys in a discrete surface using a modified MLS approximation. The algorithm was quick because the modified MLS approximation exploited locality. Stylianou and Farin [5] presented a reliable, automatic method for extracting crest lines from triangulated meshes. The algorithm identified every crest point, and then joined them using region growing and skeletonization.

However, a purely curvature-based metric may not necessarily be a good metric of crest lines extraction. Non-uniform sampling, noise and irregularities of triangulation make the curvature-based

method get false crest lines or incomplete crest lines. Saliency map [15] that assigns a saliency value to each image pixel has done excellent work in image processing. By the saliency value, salient image edges will be distinct from their surroundings. Encouraged by the success of the method on 2D problem, Lee [7] introduces the idea of mesh saliency as a measure of regional importance for graphics meshes. Lee discusses how to apply the saliency value to graphics applications such as mesh simplification and viewpoint selection. Ran Gal [8] defined salient geometric features for partial shape matching and similarity by the salient parts of an object. By "salience of a part", he captured the object's shape with a small number of parts. He proposed that the salience of a part depends on (at least) two factors: its size relative to the whole object, the number of curvature changes and strength. Similar to the two methods, we will compute the saliency value of each mesh point for our automatic algorithm.

Other types of curves aren't defined as the maximum of curvature, but defined as the zero crossings of some function of curvature. Demarcating curves [19] are the loci of points for which there is a zero crossing of the curvature in the curvature gradient direction. Demarcating curves can be viewed as the curves that typically separate ridges and valleys on 3D surface. Relief edges [20] are defined as the zero crossing of the normal curvature in the direction perpendicular to the edge. Compared to view-dependent curves, view-independent curves are locked to the object surface, and do not slide when the viewpoint changes. Moreover, they are pure geometrical and convey prominent and meaningful information about the shape. Our approach is closely related to traditional ridges and valleys.

3. PRELIMINARIES

A good introduction to differential geometry can be found in [21]. We just recall some notions of differential geometry, useful for the introduction of the extraction of crest lines on 3D meshes.

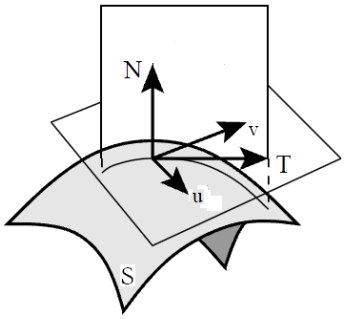

Figure 1. Normal section of surface S.

Differential geometry of a 2D manifold embedded in 3D is the study of the intrinsic properties of the surface. Here, we will recall basic notions of differential geometry required in this paper. The unit normal N of a surface S at p is the vector perpendicular to S, i.e., the tangent plane of S at p. A normal section curve at p is constructed by intersecting S with a plane normal to it, i.e., a plane that contains N and a tangent direction T. The curvature of this curve is the normal curvature of S in the direction T (See Fig.1).

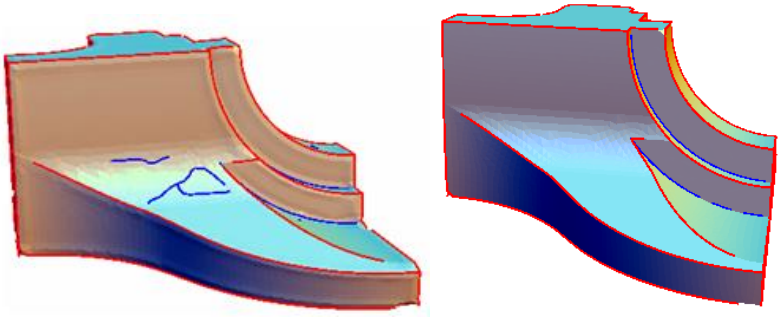

Figure 2. Comparison of the detection algorithm by saliency values with the algorithm only by principal curvatures. Left: Detect the feature lines only by principal curvatures; Right: Detect the feature lines by saliency values, a measure of regional importance.

For a smooth surface, the normal curvature in direction $\mathbf{v}$ is $k(\mathbf{v}) = \mathbf{v}^T II \mathbf{v}$, where the symmetric matrix II is the second fundamental form. The eigenvalues of II are the principal curvature values (k_{max}, k_{min}). The eigenvectors of II are the coordinates of the principal curvature directions $(\mathbf{t}_{max}, \mathbf{t}_{min})$. Let e_{max} and e_{min} be the derivatives of the principal curvatures k_{max}, k_{min} along their corresponding curvatures directions $\mathbf{t}_{max}, \mathbf{t}_{min}$. Mathematically crest lines are described via extrema of the surface principal curvatures along their corresponding lines of curvature:

$$e_{max} = 0, \quad \nabla e_{max} \cdot \mathbf{t}_{max} < 0, \quad k_{max} > |k_{min}| \quad \text{(Ridges)} \tag{1}$$

$$e_{min} = 0, \quad \nabla e_{min} \cdot \mathbf{t}_{min} < 0, \quad k_{min} < -|k_{max}| \quad \text{(Valleys)} \tag{2}$$

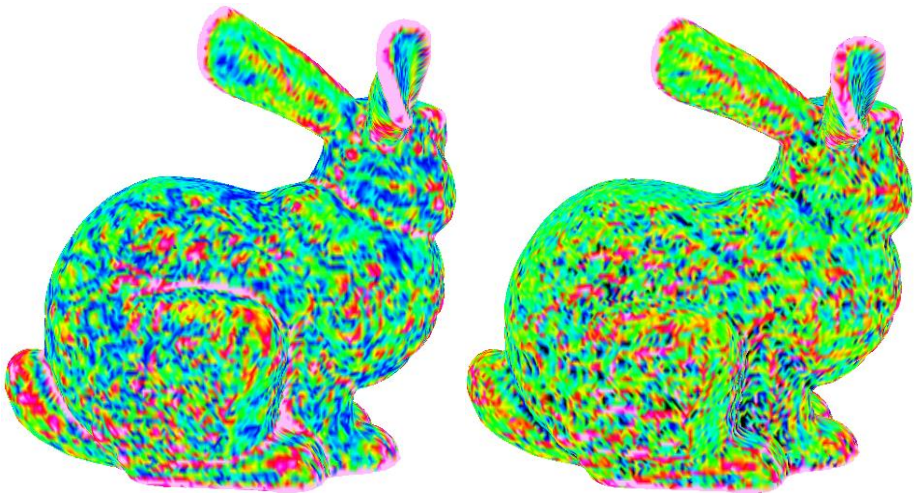

Figure 3. The saliency values can capture the interesting features at all perceptually meaningful scales and reveals the difference between the vertex and its surrounding context. Left: Visualize the saliency values of each mesh points with colour. Right: Visualize the principal curvature (max) of each mesh points with colour.

4. AUTOMATIC METHOD FOR FEATURE DETECTION

4.1 Detect the Crest Points Via Local Polynomial Fitting

In order to achieve an accurate estimation of the crest lines, we used Stylianou[5] crest point classification method to determine whether the main curvature kmax(kmin) of mesh points reaches the maximum (minimum) value along its curvature line.

1. Calculate the intersection of the normal cross section, containing $\boldsymbol{t}_{max}$ and $\boldsymbol{N}$, and the neighborhood polygon of the target point $\boldsymbol{p}$. Assuming that they intersect at the points $\boldsymbol{a}$ and $\boldsymbol{b}$, find the two ends of the edge where the points $\boldsymbol{a}$ and $\boldsymbol{b}$ are. For example, the two ends, $\boldsymbol{p}_i$ and $\boldsymbol{p}_{i+1}$, of the edge where the point $\boldsymbol{a}$ is.
2. The principal curvatures of the points of intersection $\boldsymbol{a}$ and $\boldsymbol{b}$ are calculated by linear interpolation. For example, the principal curvature $k_{\max}^{a}$ of the point $\boldsymbol{a}$ is obtained by $k_{\max}^{p_i}$ and $k_{\max}^{p_{i+1}}$ of the two ends $\boldsymbol{p}_i$ and $\boldsymbol{p}_{i+1}$.
3. if $(k_{\max}^{p_i})^2 - (k_{\max}^{a})^2 > 0$, $(k_{\max}^{p_i})^2 - (k_{\max}^{b})^2 > 0$ and $k_{\max}^{p} > T_{\max}$, then $\boldsymbol{p}$ is a crest point. Here, $T_{\max} > 0$ is the threshold.

Computing of the crest points involves estimation of high-order surface derivatives, so these surface features are very sensitive to noise and irregularities of the triangulation (see Fig.2).

4.2 Modify the Detection Algorithm by Saliency Values

Mesh saliency, a measure of regional importance, can identify regions that are different from their surrounding context and reduce ambiguity in noisy circumstances. In this paper, the computation of the salience value is built on the methods of the salience of visual parts proposed by Ran Gal [8] and mesh saliency proposed by Lee [7]. Differing from the salient parts, we compute a saliency value of each mesh point to identify mesh points that are different from their surrounding context. We have modified their algorithms slightly to define a saliency value S as a linear combination of two terms:

$$S = W_1 Curv(\mathrm{p}) + W_2 Var(\mathrm{p}) \qquad (5)$$

Where $Curv(\mathrm{p})$ is the maximal absolute curvature of a point p and $Var(\mathrm{p})$ is the absolute curvature difference. The first term of equation (5) expresses the saliency of the mesh point. The second term expresses the degree of the difference between the point and its surrounding context. We use 0.4 for W_1 and 0.6 for W_2. Let $k(\mathbf{p})$ denote the maximal absolute value of the principal curvatures and $G(k(\mathrm{p}))$ denote the Gaussian-weighted average of $k(\mathbf{p})$,

$$G(k(\mathrm{p})) = \frac{\sum_{x \in neighbor(p)} k(\mathrm{x}) \exp[-|\mathrm{x}-\mathrm{p}|^2 / (2\sigma^2)]}{\sum_{x \in neighbor(p)} \exp[-|\mathrm{x}-\mathrm{p}|^2 / (2\sigma^2)]} \qquad (6)$$

σ is a scale factor that is estimated by $\sigma = \lambda \bar{e}$, where $\bar{e}$ is the average length of edges of the mesh. We compute the saliency value $\varphi_i(\mathrm{p})$ of a vertex p as the absolute difference between the Gaussian-weighted averages $G(k(\mathrm{p}))$ computed at the two neighboring boundaries:

$$\varphi_K(p) = | G(k(p), K+1) - G(k(p), K) | \tag{7}$$

$Var(p)$ is computed by an average value at multiple scales:

$$Var(p) = \sum_{K=1}^{n} \varphi_K(\mathrm{p}) \tag{8}$$

Where n is set 3 or 4. We define a salient geometric feature point as a measure of regional importance which is salient and interesting compared to its neighborhood. The top graded points define the salient geometric feature of a given shape. So it is better than the method extracting the crest lines only by the curvature (see Fig.2 and Fig.3).

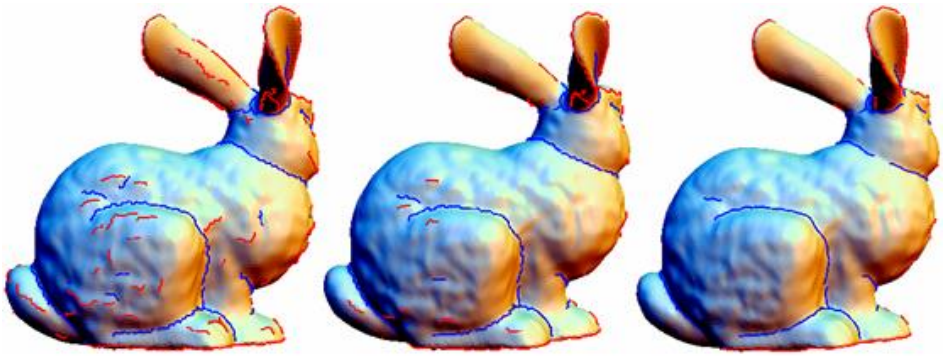

Figure 4. The comparison of the saliency algorithm with different thresholds. Left: Top 30% for ridge points, bottom 30% for valley points; Middle: Top 20% for ridge points, bottom 25% for valley points; Right: Top 8%for ridge points, bottom 15% for valley points.

4.3 An Automatic Crest Points Detection Method

Suppose that the surface points are composed of feature points and flat points. One obvious way to extract the feature points on a mesh is to select a threshold T that separates these mesh points.

Because of its intuitive properties and simplicity of implementation, threshold method enjoys a central position in applications of the extraction of the surface feature points. Unfortunately it is difficult for a user to set the threshold if no any apriori knowledge (see Fig.4 and Fig.6)

Our automatic algorithm is based on the observation that polygonal surface models with features usually contain many flat regions and a little sharp edges. Sorting the mesh saliency value of each point by the ascending order, we can find: The saliency values begin to increase slowly and the plot almost corresponds to a planar line, after arriving to a value, the plot rises steeply. Obviously, flat regions correspond to the planar line part of the plot and sharp edges correspond to the steep

line part of the plot. So this value is the optimal threshold to extract the feature points (see Fig.5).

Finally, we summarize the automatic crest lines extraction algorithm as follows:

1. Estimate necessary surface curvature and their derivatives via local polynomial fitting.
2. Compute the saliency values as a measure of regional importance for graphics meshes.
3. Seek the optimal threshold by the analysis mentioned above.
4. Extract the feature points by the threshold.

For step 3, we give the pseudo-ode as follow:

Procedure SeekThreshold(saliency)

1. Quicksort (saliency) /* Sort the mesh saliency value of each point.*/
2. for i = vertex_num / 2 to vertex_num step k
   /* vertex_num represents the number of mesh vertices. Firstly, we search the value at a coarse scale, usually set k = vertex_num /200. */

2.1. m_i = saliency[i] – saliency[i-1]; m_{i-1} = saliency[i-1] – saliency[i-2]

2.2 if $m_i > 1.3 * m_{i-1}$, then shrink the search range in size and repeat step 2 at a finer scale till finding the threshold. /*From Fig.5 we can find the plot rises steeply, we used 1.3 for the coefficient.*/

4.4 Generation of Crest Lines

Once the crests have been detected, we need to connect them together. We follow the procedure proposed in [4] with an addition which can reduce the fragmentation of the crest lines:

1) Get the optimal threshold in section 4.3. Flag the vertex which saliency value is greater than the threshold T as feature points set M_1. Flag the vertex which saliency value is less than T and greater than $0.8*T$ as weak feature points set M_2. Define k-neighbor of a point p as $N(\mathrm{p},k)$.

2) For a feature point p, examine the point q in $N(\mathrm{p},1)$. If only one point $\mathrm{q} \in \mathrm{M}_1$, then connect it to p.

3) If two or more points $\mathrm{q}_i \in \mathrm{M}_1, i = 1,...,n \text{ and } n \geq 2$, then we connected p to one of them by following the vertex q_i corresponding to the smaller dihedral angle with the orientation of the principal curvature.

4) No any point in M_1, but at least one point $\mathrm{q} \in \mathrm{M}_2$. If having $\mathrm{k} \in N(\mathrm{q},1) \textit{ and } \mathrm{k} \in M_1 \text{ and } \mathrm{k} \notin N(\mathrm{p},1)$, then connect p to q similarly following the rule in step 2 and step 3.

Repeat step 2 to step 4.

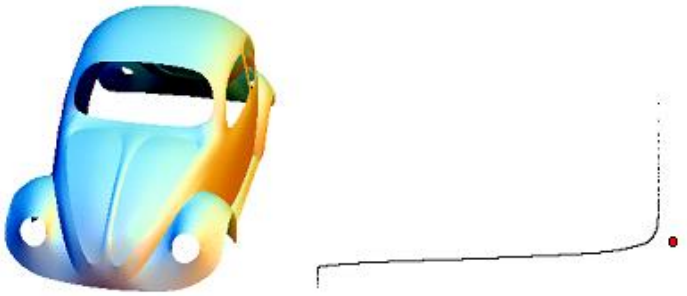

Figure 5 Sort the mesh saliency value of each point by the ascending order, we can get a plot: Firstly, the plot almost corresponds to a planar line, after arriving to a value, the plot rises steeply. Left: 3D model; Right: The plot corresponding to the mesh saliency values by the ascending order.

5. RESULTS

For evaluating the effectiveness of our automatic mesh saliency method, this section shows results of our algorithm and compares it to the user-specified threshold algorithm. All of our tests were run on a PC with Intel® Core™ 2 1.73.GHz processor and 1.0 GB of main memory.

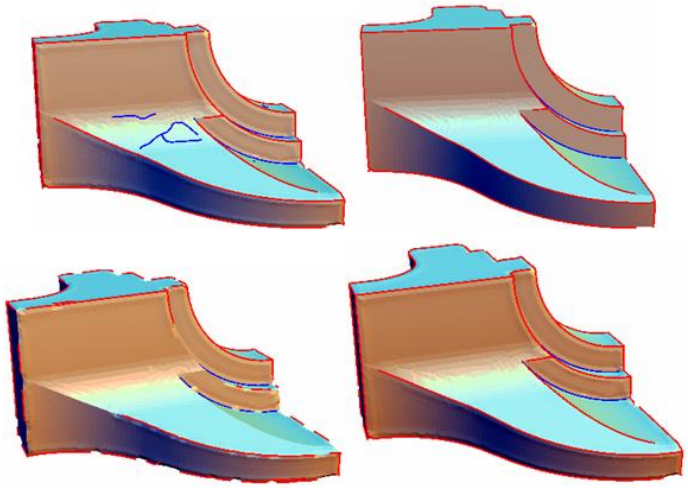

Figure 6 The comparison of the saliency algorithms with different thresholds and our automatic detection algorithm. Upper Left: Top 25% for ridge points, bottom 30% for valley points； Upper Right: Top 15% for ridge points, bottom 20% for valley points； Bottom Left: Top 5% for ridge points, bottom 5% for valley points; Bottom Right: The automatic detection algorithm.

Fig.2 and 3 show some results. Fig.2 shows the comparison of the detection algorithm by saliency

values and the algorithm only by principal curvatures. Owing to its locality, the method only by principal curvature is sensitive to noise and irregularities of the triangulation and usually produces spurious crest lines. The result shows on the left of Fig.2. Mesh saliency that measure the region importance at multiple scales in the neighborhood can reveal the difference between the vertex and its surrounding context. So it can extract the most salient features points robustly. The result on the right shows that the lines are clearly detected by saliency values. In Fig. 3, we

visualize the magnitude of principal curvature value and the saliency value of each point with color on a bunny model. Warm color corresponds to the sharp features and cool color corresponds to the flat regions. We can find that saliency values differentiate the neck and the leg from their circumferences.

Fig.4 and Fig.6 show the comparison between the different thresholds. The surface points are composed of feature points and flat points. One obvious way to extract the feature points on the surface is to select a threshold T , for example, TOP 20% means that the feature points are the top 20% points with high saliency values. But improper threshold produces many spurious crest lines. Polygonal surface models can be roughly divided into two classes: CAD-like models showed in Fig.6, which usually have large flat regions; Non-CAD-like model showed in Fig.4, which have many fine details. We can't find a unified threshold for the detection method. So how to decide the threshold is a problem for user-specified threshold methods.

Fig.5 gives an analysis for the optimal threshold. On the left is the 3D model; on the right is the corresponding plot of the saliency values in ascending order. The plot begins to rise slowly and almost corresponds to a planar line, after arriving to a value (Flagged in red circle dot), the plot rises steeply. The value at the red circle dot is the threshold we need. Fig.6 shows results from the fandisk CAD model, in which our automatic method works well. Moreover, our method doesn't need a user to select the threshold. It is not easy for a user to modify the threshold if no any apriori knowledge. Ohtake's method [2, 22] is a reliable detection of ridge-valley structures on surfaces approximated by dense triangle meshes and has become a representative method of crest lines detection algorithm. Fig.7 shows that our automatic method almost has the same precise as Ohtake's method.

6. CONCLUSION

This paper has presented an automatic algorithm for the detection of crest lines on triangular meshes basesssssd on the concept of salient geometric features. The utility of saliency values for robustly detecting the crest lines on surface has been demonstrated. The results show saliency values effectively capture important shape features.

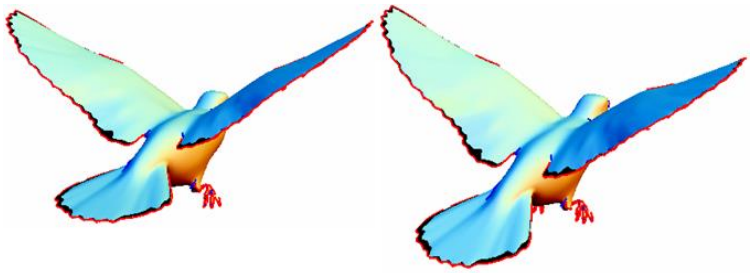

Figure 7 Our automatic algorithm VS. Ohtake's method, left: Our automatic algorithm; right: Ohtake's method.

Our automatic algorithm is a fully automatic method. It can advantageously select a "good" threshold without any user intervention. The results show that our automatic algorithm can find an optimal threshold to detect the crest lines on 3D meshes. In the future we intend to develop data clustering method into the field of the crest lines detection.

ACKNOWLEDGEMENT

This study was supported by the Innovation projects of Department of Education of Guangdong Province, China (NO.2017KTSCX183) and the Funding for Introduced Innovative R&D Team Program of Jiangmen (Grant No.20186301000900198 44).

REFERENCES

[1] Forrester Cole & Kevin Sanik, (2009) "How Well Do Line Drawings Depict Shape?", *ACM Transaction on Graphics*, Vol. 28, No.3, pp43-51.

[2] Ohtake Y. & Belyaev A., (2004) "Ridge-valley Lines on Meshes via Implicit Surface Fitting", *ACM Transactions on Graphics*, Vol. 23, No. 3, pp609-612.

[3] Shin Yoshizawa & Alexander Belyaev, (2005) "Fast and Robust Detection of Crest Lines on Meshes", *Symposium on Solid and Physical Modeling'05*, pp227-232.

[4] Soo-Kyun Kim & Chang-Hun Kim, (2006) "Finding Ridges and Valleys in A Discrete Surface Using A Modified MLS Approximation", *Computer-Aided Design*, Vol. 38, No.2, pp173-180.

[5] Georgios Stylianou & Gerald Farin, (2004) "Crest Lines for Surface Segmentation and Flattening", *IEEE Transaction on Visualization and Computer Graphics*, Vol. 10, No. 5, pp536-543.

[6] Tilke Judd & Fredo Durand, (2007) "Apparent ridges for line drawing", *ACM Transactions on Graphics*, Vol. 26, No. 3, pp19-26.

[7] Chang Ha Lee & Amitabh Varshney, (2005) "Mesh Saliency". *Proceedings of ACM Siggraph'05*, pp659-666.

[8] Ran Gal & Daniel Cohen-Or, (2006) "Salient Geometric Features for Partial Shape Matching and Similarity", *ACM Transactions on Graphics*, Vol. 25, No. 1, pp130-150.

[9] Taubin G, (1995) "Estimating the Tensor of Curvature of a Surface from a Polyhedral Approximation", *In Proceedings of Fifth International Conference on Computer Vision'95*, pp902-907.

[10] Sachin Nigam & Vandana Agrawal, (2013) "A Review: Curvature approximation on triangular meshes", *Int. J. of Engineering science and Innovative Technology*, Vol. 2, No. 3, pp330-339

[11] Xunnian Yang & Jiamin Zheng, (2013) "Curvature tensor computation by piecewise surface interpolation", *Computer Aided Design*, Vol. 45, No. 12, pp1639-1650.

[12] Gady Agam & Xiaojing Tang, (2005) "A Sampling Framework for Accurate Curvature Estimation in Discrete Surfaces", *IEEE Transaction on Visu alization and Computer Graphics*, Vol. 11, No. 5, pp573-582.

[13] Meyer M. & Desbrun M., (2003) "Discrete Differential-geometry Operators for Triangulated 2-manifolds", *In Visualization and Mathematics III' 03*, pp35-57.

[14] Stupariu & Mihai-Sorin, (2016) "An application of triangle mesh models in detecting patterns of vegetation", *WSCG' 2016*, pp87-90.

[15] Chen L. & Xie X., (2003) "A visual attention model for adapting images on small displays", *ACM Multimedia Systems Journal*, Vol. 9, No. 4, pp353-364.

[16] Lee, Y. & Markosian, L., (2007) "Line drawings via abstracted shading", *ACM Transactions on Graphics*, Vol. 26, No. 3, pp1-9.

[17] Chen, J. S.-S. and H.-Y. Feng, (2017) "Idealization of scanning-derived triangle mesh models of prismatic engineering parts", *International Journal on Interactive Design and Manufacturing*, Vol. 11, No. 2, pp205-221.

[18] Decarlo D. & Finkelstein A., (2003) "Suggestive Contours for Conveying Shape", *ACM Transactions on Graphics*, Vol.22, No. 3, pp848-855.

[19] M. Kolomenkin & I. Shimshoni, (2008) "Demarcating curves for shape illustration", *ACM Transactions on Graphics*, Vol.27, No.5, pp157-166.

[20] M. Kolomenkin & I. Shimshoni, (2009) "On Edge Detection on Surface", *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp2767-2774.

[21] M. P. Do Carmo (2004) *Differential geometry of curves and surfaces*, Book, China Machine Press.

[22] A. Belyaev & P.-A. Fayolle, (2013) "Signed Lp-distance fields", *CAD*, Vol.45, No. 2, pp523-528.

[23] Y Zhang & G Geng, (2016) "A statistical approach for extraction of feature lines from point clouds", *Computers & Graphics*, Vol. 56, No. 3, pp31-45.

AUTHORS:

Zhihong Mao received his PhD degree at Department of Computer Science and Engineering, Shanghai Jiao Tong University. He is now a teacher at Division of Intelligent Manufacturing, Wuyi University, China. His research interests include computer aided design and Digital Geometry Processing.

Ruichao Wang received his PhD degree at Department of Mechanical and Automotive Engineering, South China University of Technology. He is now a teacher at Division of Intelligent Manufacturing, Wuyi University, China. His research interests include Intelligent manufacturing and industrial robot technology, precision numerical control technology, digital power inverter technology

Fan Liu was born in 1996. He is now a Master Candidate at Division of Intelligent Manufacturing, Wuyi University, China. His research interests include Reverse Engineering.

Yulin Zhou was born in 1995. She is now a Master Candidate at Division of Intelligent Manufacturing, Wuyi University, China. Her research interests include computer aided design.