Salience Feature Detection for Polygonal Meshes Based on Maximum Normal Variation

Hung-Kuang Chen
Dept. of Electronic Engineering,
National Chin-Yi University of Technology,
Taichung, Taiwan 40109
Email: hankchentw@gmail.com

Ying-Chuan Chen
Dept. of Electronic Engineering,
National Chin-Yi University of Technology,
Taichung, Taiwan 40109
Email: xlllxd@gmail.com

***Abstract*—Feature detection or extraction for polygon mesh is one of the most fundamental and extensively used techniques. Applications such as mesh simplification, smoothing, parameterization, segmentation, morphing, and shape matching, etc. requires feature detection. A common technique is to identify the edge features through the dihedral angles of the two neighbouring faces adjacent to the edge. In this paper, we propose a variation of such technique that identifies feature vertices by considering the maximum normal variation of the neighbouring faces in the 1-ring neighbourhood of a vertex. By clustering and labelling the feature vertices, we may successfully identify the features of a given object as disjoint cluster of feature vertices. By means of Shortest Path or Spanning Tree with proper weighting scheme, we may further process each cluster as a feature curve that potentially enables a cut to the mesh.**

***Keywords*-feature detection; feature extraction; dihedral angle; mesh cut;**

I. Introduction

As the graphics hardware evolving from Hi-End expensive equipments into commonly available inexpensive commodity, 3D graphics applications are rapidly popularized. Feature detection, as one of the fundamental graphics processing techniques is vital to the success of various applications such as shape recognition, shape matching, mesh simplification, smoothing or fairing, morphing, 3D archiving, etc. Polygonal mesh as a common 3D object representation usually provides only a raw description to the object surface geometry and topology. Higher-order descriptions such as the sharp features, concave creases, sharp corners, etc., are not directly available to the user, which often requires a further processing to the raw mesh data. Salient features can be easily identified by human beings. However, it requires a definite or mathematical procedure to describe such for computer to identify. Previous techniques on feature detection are plenty over last decade. Challenge remains on noise proofing, efficient representation, indexation and matching, etc. For a more extensive survey, readers are encouraged to refer the relevant study [1].

II. Related Works

According to the theory of differential geometry, one may define the feature of a smooth orientable surface as the extrema of the first and second order of derivatives such as the principle curvatures and the principle direction. Previous works on feature detection may be classified into global techniques [2] or local techniques [3]–[13]. Ohtake et al. find perceptually salient features by global fitting, which is appropriate for high order feature search and is more accurate than local methods at the cost of expensive computation [2]. Yoshizawa et al. proposed a local technique using polynomial fitting [5]. Hildebrandt et al. proposed another local technique based on discrete curvature computations [6]. On the basis of Hildebrandt et al.'s work , Yoshizawa et al. proposed another method for the detection of crest lines [7]. Considering features as the grid points that has extrema in curvature value along principle directions, Stylianou and Farin find feature curves by applying region growing from the feature points [3]. Mao et al adopted similar technique in detecting perceptually salient features [4].

Since the approximation of curvature usually requires expensive computation, an alternative approach is based on the used of normal vectors. Hubeli and Gross proposed using the second order difference (SOD) [8]. Wang et al. suggest filtering through dihedral angles of adjacent faces [9], [10]. Similarly, Angelo et al. continuity detection. To against with noises, a number of works suggest normal tensor voting [12]–[14].

In addition, a number of techniques such as multi-scale, integral invariant, and morse theory etc., has been applied to feature detection or related works [15]–[25].

III. Overview

Given an orientable 3D mesh denoted as $M = (V, F)$. The set V comprises the vertices of the mesh defined in a three dimensional Cartesian space R^3, which is represented by a 3-tuple of real, say $v_i = (x_i, y_i, z_i) \in R^3$. The set F contains the faces of the mesh defined by a subset of different vertices of V; in particular, a triangular faces represented by a 3-tuple of index to V, say $f = (I_a, I_b, I_c) \in I^3$. Furthermore, we

978-1-5090-3071-2/16 $31.00 © 2016 IEEE
DOI 10.1109/IS3C.2016.219

confine the computation to the local geometry described by the 1-ring neighbourhood of a vertex v, denoted as $R(v)$, which comprises a set of faces adjacent to v. Moreover, we denote the number of vertices as $|V|$, the number of faces as $|F|$, and the number of faces in the 1-ring neighbourhood of a vertex v as $|R(v)|$.

A. The dihedral angle and the characterization of surface type

Consider two adjacent faces, say f_a and f_b, of an edge $e_{i,j} = (v_i, v_j)$ as shown in Fig. 1.

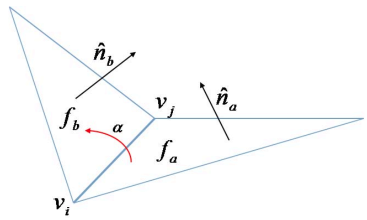

Figure 1. The dihedral angle α, of two faces, f_a and f_b, adjacent to an edge $e_{i,j} = (v_i, v_j)$.

Let $\vec{n}_a$ and $\vec{n}_b$, respectively, be the face normals of face f_i and f_j. The dihedral angle α can be given by

$$\alpha = \cos^{-1}\left(\frac{\vec{n}_a \cdot \vec{n}_b}{|\vec{n}_a||\vec{n}_b|}\right). \tag{1}$$

Thereby, we may characterize the surface into three major types according to the value of dihedral angle α by introducing a threshold δ.

$$\begin{array}{lll} \alpha = [0, \pi - \delta), & e_{i,j} & \text{is a concave crease;} \\ \alpha = [\pi - \delta, \pi + \delta] & e_{i,j} & \text{is an interior edge of a flatten} \\ & & \text{surface;} \\ \alpha = [\pi + \delta, 2\pi), & e_{i,j} & \text{is a convex crease.} \end{array} \tag{2}$$

B. Feature detection by maximum variation angle

Previous literatures mostly identify features in points, lines, or curves. In our work, the detection of feature points are performed on the basis of the salience value of a vertex v defined by an extension of the aforementioned dihedral angle, called the maximum variation angle defined as follows.

Definition 1: **Maximum Variation Angle—** Given a vertex v of V and its 1-ring neighbourhood $R(v)$, we define the maximum variation angle α_{max} of v with respect to the faces in $R(v)$ by

$$\alpha_{max} = \mathbf{max}\{\cos^{-1}\left(\frac{\vec{n}_a \cdot \vec{n}_b}{|\vec{n}_a||\vec{n}_b|}\right)\}, \tag{3}$$

where $\hat{n}_a$ and $\hat{n}_b$, respectively are the unit normals of any two distinct faces f_a and f_b of $R(v)$.

In terms of the maximum variation angle α_{max}, we may define the salience of a vertex v, denoted as $S(v)$ with respect to its 1-ring neighbourhood $R(v)$ by

$$S(v) = \frac{1 + \cos\alpha_{max}}{2} \tag{4}$$

Note that $S(v)$ is approximately 0 for flatten surfaces but close to 1.0 for both concave and convex regions that we consider, in both case, salient features.

As an alternative, we may also normalize the salience value by

$$S(v) = \frac{|\alpha_{max} - \pi|}{2} \tag{5}$$

Following to the computation of the salience values of the vertices, the feature points are then filtered by applying a threshold value δ calculated by the sum of an increment or decrement of integer multiple τ of variance σ^2, the mean μ, and the standard deviation σ:

$$\delta = \mu + \sigma - \tau \cdot \sigma^2 \tag{6}$$

where

$$\mu = E[S(v)] = \frac{1}{n}\sum_{i=1}^{n} S(v_i), \tag{7}$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(S(v_i) - \mu)^2}. \tag{8}$$

Consequently, the set of filtered feature points P can be given by

$$P = \{v | S(v) > \delta, \forall v \in V\} \tag{9}$$

C. Feature clustering

After the set of feature points P is computed, we use region growing to cluster features that have geometric proximity. The clustering procedure is as follows.

Algorithm: *Feature Clustering by Region Growing*

1: Given initial untagged features P and set cluster count $i = 0$.
2: Pick up an untagged point p from P.
3: Initialize cluster C_i by letting $C_i = \{p\}$ and set $P = P \setminus \{p\}$.
4: Initialize the set of neighbouring features N by $N = P \cap R(p)$.
5: Choose a feature point p in N.
6: Insert p to C_i and exclude p from P by $P = P \setminus \{p\}$.
7: Introduce more neighbouring feature points by $N = N \cup P \cap R(p)$.
8: Repeat Steps 5 to 7 until N is empty.
9: Repeat Steps 2 to 8 until P is empty.

D. Feature representation

The problem we have left so far is the abstraction or representation of higher order features from the clustered feature sets. A common practice is to use the shortest path algorithm over each cluster to find one or more feature curves or crest lines and kept these features as a collection of lines or curves. In our practice, we have also adopted the Prims algorithm to find a spanning tree structure of the cluster. For these two algorithms are commonly available, we do not exert much effort in describing our implementation details.

IV. Results and Discussions

To verify our approach, we have chosen two types of meshes, the Bunny and Octopus meshes with sophisticated surfaces from Stanford 3D scanning repository as well as the Fandisk and Octa-flower meshes from Aim@Shape for CAD domain. The size of the test meshes as well as their corresponding processing times are listed in Table I.

Table I
THE TEST MESHES AND THE EXECUTION TIMES FOR FEATURE POINT EVALUATION

Test mesh	**Fandisk**	**Octa-flower**	**Octopus**	**Bunny**
# of vertices	6475	7919	16944	34817
# of faces	12946	15834	33872	69630
Processing times	$<$ 1ms	$<$ 1ms	15 ms	31 ms

The test codes are written in C++ and OpenGL using CodeBlock ver. 13.12 with MingGW32 4.8.1 running on Microsoft Windows 8.1 Professional 64Bit platform equipped with Intel Core i7-2600K, 4GBytes RAM, and NVIDIA Geforce GTX 560Ti. The running times of our algorithm for feature point evaluation is $O(n)$. According to our experimental results, the execution times for the four meshes are all less than 0.1 second. Hence, it is very computational efficient.

In the finale stage, we have implemented the Shortest Path Algorithm and the Prims Minimum Spanning Tree Algorithm for feature curves extraction from the clusters of feature. The running times for both case are summarized in Table II.

Table II
THE EXECUTION TIMES FOR FEATURE LINES FEATURE EXTRACTION

Threshold $\delta = \mu + \sigma - \tau \cdot \sigma^2$	Shortest Path	Prims MST
$\tau = 0$	1.703s	0.703s
$\tau = 6$	3.173s	1.219s
$\tau = 10$	9.516s	3.454s

To verify the detect features, we rendered each feature cluster with different color. The results on Bunny are presented in Figure 2(a)-(c) using different threshold, i.e., $\tau = 0, 6, 10$, respectively.

According to the results presented in Figure 2, we may found that, larger σ introduces more feature clusters and

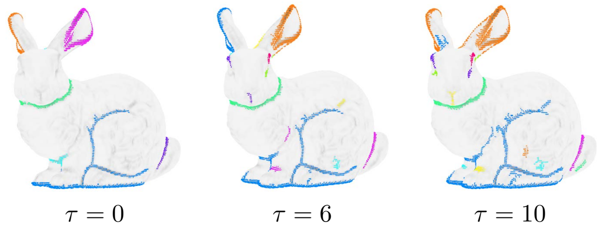

Figure 2. The computed feature clusters using three different thresholds.

larger cluster size. Hence, a proper choice of σ may depends on the type of application or mesh in use and is better left for the user to decide.

To conclude our experiment, a demonstration of the extracted features and the final crest lines found by applying either Shortest Path or Prims MST are given in Figure 3.

According to the results presented so far, we may conclude that our new approach is successful at extracting feature points, organizing features in clusters, and potentially permit a further abstraction of each cluster by a further processing procedure such as the Shortest Path and Prims MST.

V. Conclusion

In this paper, we have proposed a novel normal-based salience feature detection technique for feature points identification called the maximum variation angle, a threshold tunning mechanism, a feature clustering technique using region growing, and two possible abstraction implementations. From the experimental results presented in this paper, we have conclude that our new approach is successful at identifying, clustering and abstracting features from at least two types of meshes, i.e., the meshes acquired from 3D scanning and the meshes produced from CAD software.

References

[1] E. Boyer, A. M. Bronstein, M. M. Bronstein, B. Bustos, T. Darom, R. Horaud, I. Hotz, Y. Keller, J. Keustermans, A. Kovnatsky, *et al.*, "Shrec 2011: robust feature detection and description benchmark," *arXiv preprint arXiv:1102.4258*, 2011.

[2] Y. Ohtake, A. Belyaev, and H.-P. Seidel, "Ridge-valley lines on meshes via implicit surface fitting," *ACM Trans. Graph.*, vol. 23, pp. 609–612, Aug. 2004.

[3] G. Stylianou and G. Farin, "Crest lines for surface segmentation and flattening," *IEEE Transactions on Visualization and Computer Graphics*, no. 5, pp. 536–544, 2004.

[4] Z. Mao, G. Cao, and M. Zhao, "Robust detection of perceptually salient features on 3d meshes," *The Visual Computer*, vol. 25, no. 3, pp. 289–295, 2009.

[5] S. Yoshizawa, A. Belyaev, and H.-P. Seidel, "Fast and robust detection of crest lines on meshes," in *Proceedings of the 2005 ACM Symposium on Solid and Physical Modeling*, SPM '05, (New York, NY, USA), pp. 227–232, ACM, 2005.

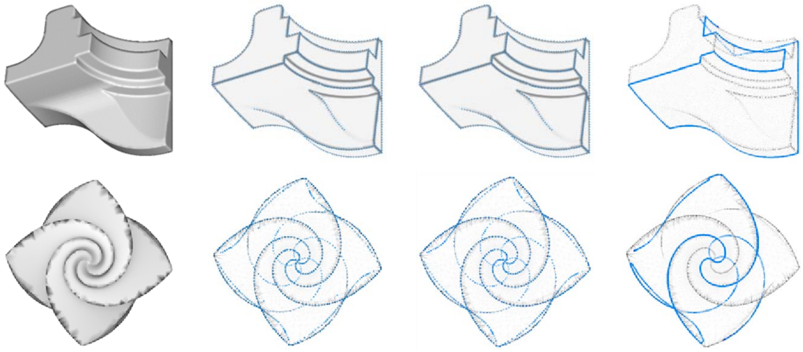

Figure 3. A summary of our experimental results.

[6] K. Hildebrandt, K. Polthier, and M. Wardetzky, “Smooth feature lines on surface meshes,” in *Proceedings of the Third Eurographics Symposium on Geometry Processing*, SGP ’05, (Aire-la-Ville, Switzerland, Switzerland), Eurographics Association, 2005.

[7] S. Yoshizawa, A. Belyaev, H. Yokota, and H.-P. Seidel, “Fast, robust, and faithful methods for detecting crest lines on meshes,” *Computer Aided Geometric Design*, vol. 25, no. 8, pp. 545–560, 2008.

[8] A. Hubeli and M. Gross, “Multiresolution feature extraction for unstructured meshes,” in *Visualization, 2001. VIS ’01. Proceedings*, pp. 287–294, 2001.

[9] C. C. Wang, “Bilateral recovering of sharp edges on feature-insensitive sampled meshes,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 12, no. 4, pp. 629–639, 2006.

[10] C. C. Wang, “Incremental reconstruction of sharp edges on mesh surfaces,” *Computer-Aided Design*, vol. 38, no. 6, pp. 689–702, 2006.

[11] L. Di Angelo and P. Di Stefano, “continuities detection in triangular meshes,” *Computer-Aided Design*, vol. 42, no. 9, pp. 828–839, 2010.

[12] G. Medioni, C.-K. Tang, and M.-S. Lee, “Tensor voting: Theory and applications,” *Proceedings of RFIA, Paris, France*, vol. 3, 2000.

[13] D. L. Page, Y. Sun, A. Koschan, J. Paik, and M. A. Abidi, “Normal vector voting: crease detection and curvature estimation on large, noisy meshes,” *Graphical models*, vol. 64, no. 3, pp. 199–229, 2002.

[14] H. S. Kim, H. K. Choi, and K. H. Lee, “Feature detection of triangular meshes based on tensor voting theory,” *Computer-Aided Design*, vol. 41, no. 1, pp. 47–58, 2009.

[15] M. Pauly, R. Keiser, and M. Gross, “Multi-scale feature extraction on point-sampled surfaces,” *Computer Graphics Forum*, vol. 22, no. 3, pp. 281–289, 2003.

[16] G. Lavoué, F. Dupont, and A. Baskurt, “A new cad mesh segmentation method, based on curvature tensor analysis,” *Computer-Aided Design*, vol. 37, no. 10, pp. 975–987, 2005.

[17] T. Shimizu, S. Kanai, T. Kishinami, *et al.*, “A new bilateral mesh smoothing method by recognizing features,” in *Computer Aided Design and Computer Graphics, 2005. Ninth International Conference on*, pp. 6–pp, IEEE, 2005.

[18] C. H. Lee, A. Varshney, and D. W. Jacobs, “Mesh saliency,” in *ACM SIGGRAPH 2005 Papers*, SIGGRAPH ’05, (New York, NY, USA), pp. 659–666, ACM, 2005.

[19] Y.-L. Yang, Y.-K. Lai, S.-M. Hu, and H. Pottmann, “Robust principal curvatures on multiple scales,” in *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, SGP ’06, (Aire-la-Ville, Switzerland, Switzerland), pp. 223–226, Eurographics Association, 2006.

[20] Y.-K. Lai, Q.-Y. Zhou, S.-M. Hu, J. Wallner, and H. Pottmann, “Robust feature classification and editing,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 13, no. 1, pp. 34–45, 2007.

[21] J. Sahner, B. Weber, S. Prohaska, and H. Lamecker, “Extraction of feature lines on surface meshes based on discrete morse theory,” *Computer Graphics Forum*, vol. 27, no. 3, pp. 735–742, 2008.

[22] T. Weinkauf and D. Gnther, “Separatrix persistence: Extraction of salient edges on surfaces using topological methods,” *Computer Graphics Forum*, vol. 28, no. 5, pp. 1519–1528, 2009.

[23] S. Wang, T. Hou, Z. Su, and H. Qin, “Diffusion tensor weighted harmonic fields for feature classification,” in *Pacific Graphics Short Papers* (B.-Y. Chen, J. Kautz, T.-Y. Lee, and M. C. Lin, eds.), The Eurographics Association, 2011.

[24] X.-c. Wang, J.-j. Cao, X.-p. Liu, B.-j. Li, X.-q. Shi, and Y.-z. Sun, “Feature detection of triangular meshes via neighbor supporting,” *Journal of Zhejiang University Science C*, vol. 13, no. 6, pp. 440–451, 2012.

[25] J. Wu, X. Shen, W. Zhu, and L. Liu, “Mesh saliency with global rarity,” *Graphical Models*, vol. 75, no. 5, pp. 255–264, 2013.