UDC 621.397

E.V. OSHAROVSKAYA
Odessa National O.S. Popov's Academy of Telecommunications , Ukraine

DETECTING FEATURES ON TRIANGULAR MESHES BY TENSOR VOTING

Abstract – In this paper, triangular meshes have been used to represent objects in computer-aided design, 3D TV and computer graphics, not only due to their simplicity and efficiency, but also the rapid development of 3D acquisition techniques. Tensor voting (TV) is a method for inferring geometric structures from sparse, irregular and possibly noisy input. 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 selection is conducted to extract the real features from the initially detected features.

Keywords: surface modeling, 3D TV, triangular meshes, tensor voting

Е. ОШАРОВСКАЯ
Одесская национальная академия связи им. А.С. Попова

ОПРЕДЕЛЕНИЕ ПРИЗНАКОВ НА ТРЕУГОЛЬНЫХ СЕТКАХ ТЕНЗОРНЫМ ГОЛОСОВАНИЕМ

Аннотация – В статье рассматриваются полигональные сетки для представления объектов в системах автоматизированного проектирования, 3D-телевидения и компьютерной графики, не только в связи с их простотой и эффективностью, но и быстрым развитием методов 3D об работки. Тензор голосования (ТВ) является методом для выделения в геометрических структурах разреженностей, нерегулярностей и, возможно, шумом источника. В данной статье предлагается метод двух этапов для обнаружения особенностей на треугольных сетках, На первом этапе, методом ТИ обнаруживаются начальные признаки, включающие в себя такие как острые края и слабые особенности и, возможно, шумы. На втором этапе, происходит уточнение отобранных признаков для извлечения реальных возможностей обнаружения особенностей.

Ключевые слова: Полигональное моделирование, 3D TV, треугольные сетки, тензор голосования.

1. Introduction

Polygonal meshes [1] are the most popular 3D scene representation in many industries such as architecture and entertainment. Due to realism requirements in computer graphics and the development of 3D scanning technologies, polygonal meshes representing 3D surfaces contain millions of polygons. On one hand they can represent satisfactorily almost any geometric detail of the surface. On the other hand these meshes are complex and computationally expensive to be stored, transmitted and rendered. To overcome these limitations, many techniques to compress and simplify complex meshes have been developed leading to progressive approaches [2], even for time-varying meshes [3].

Researchers in [4, 5] present a deformable three-dimensional mesh model which allows the recovery of the 3D shape and 3D motion. The shape is represented by the triangular mesh, while the movement by vertices translations. Deformations occur inter- and intra-frames, with photometric and smoothness constraints.

As the use of implicit 3D representations gains popularity, the need for modeling software that provides implicit model editing capabilities increases. New surface editing approaches are continually being explored and much research is being conducted to find ways to interactively modify implicit models. Towards these ends we investigated tensor voting as a possible technology that could provide a new and interesting approach for editing implicit models. Tensor voting is a method for grouping geometric features [6]. It can be used to generate surfaces from a sparse set of possibly noisy and irregular input data, and therefore may provide novel editing capabilities within a 3-D modeling context. The goal of our work was to investigate tensor voting as a technique for interactive surface modeling. We were interested in determining if TV can be used to model simple objects, edit them interactively and control their shape via "input tokens." To achieve this, we developed a TV modeling system (TVMS) based on an already existing TV framework and conducted .experiments to evaluate TV as a 3-D modeling tool. Spheres were used to examine the parameters of TVMS. [7]

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). [8] They 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 obtain the final feature lines. This method coupled with the similar measure was further used in [9] to detect perceptually salient features on 3D meshes. Yoshizawa *et al.* (2005) [10] extracted the feature lines by estimating the curvature tensor and curvature derivatives via local polynomial fitting. Kim and Kim (2006) [11] 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) [12] extracted features on a polygonal surface by analyzing the focal surface instead of the original mesh. They contended 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 proposed a method for detecting feature lines on meshes. [10]

Another important category is normal vector based methods. These methods usually identify the features by analyzing the dihedral angle of two triangles sharing an edge or the diversity of the normal in a local region around the current vertex (di Angelo and di Stefano, 2010).[13] proposed a normal vector voting method for feature detection and curvature estimation on noisy meshes. This method is further used in surface segmentation and feature detection. The normal tensor voting method can handle sharp features and show robustness to noisy. [8]

Tensor voting (TV) is a method for inferring geometric structures from sparse, irregular and possibly noisy input. It was initially proposed by Guy and Medioni [14] and has been applied to several computer vision applications. TV generates a dense output field in a domain by dispersing information associated with sparse input tokens. In 3-D this implies that a surface can be generated from a set of input data, giving tensor voting a potential application in surface modeling. As higher-order derivatives of the surface are noise sensitive. These unreliable differential geometric properties 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.[15] Therefore, the noise and discontinuities should be specially taken care of for piecewise-smooth surfaces in feature detection. [9]

3 Tensor voting

This section provides an introduction to the Tensor Voting methodology. TV has its background in early computer vision problems where the available data often is sparse and noisy, making it difficult to extract relevant information and structures. TV identifies local feature descriptions by spreading the information associated with shape-related input within a neighborhood while enforcing a smoothness constraint. This process refines the information and accentuates local features. By doing so, coherent, locally smooth, geometric features are defined and noise is discarded.

Each data point communicates its information in a neighborhood through a voting process. [9] The more information that is received at each data point, the stronger is the likelihood of a geometric feature being present at a certain location. This likelihood is expressed through a confidence measure, saliency, which is used in the feature extraction process. TV is based on two elements; a data representation, which is obtained by means of tensor calculus, and communication of data through linear voting, a process similar to linear convolution. The input elements, referred to as input tokens, are encoded into tensor form and communicate their information to their neighboring tokens via pre-calculated tensor voting fields. After this initial voting step, each token has its confidence and surface orientation encoded into a generic second order symmetric tensor. The tokens vote a second time to propagate their information throughout a neighborhood. The result is a dense tensor field which assigns a measure of confidence and saliency to each point in the domain. This dense map is decomposed into three dense maps, each representing a geometric feature (junctions, curves or surfaces), which are analyzed during feature extraction. The 3-D case, specifically surface voting, is sufficient for our needs and will therefore be the focus of this paper.

Diagonalizing a second order symmetric tensor, which can be represented by a 3×3 matrix, produces the associated characteristic equation. Solving this equation leads to a representation based on the eigenvalues λ_1; λ_2; λ_3 (in decreasing order) and the associated eigenvectors e_1;e_2;e_3 of the tensor. A second order symmetric tensor may be graphically represented as an ellipse in 2-D or an ellipsoid in 3-D. The eigenvalues describe the general size and shape of the ellipsoid and the eigenvectors describe its principal directions. Because of the properties of second order symmetric tensors, the eigenvalues are real and positive, or zero, and the eigenvectors form an orthonormal basis. A tensor can be decomposed into three components.

The first term corresponds to a 3-D stick tensor, which implies a surface patch with normal e_1. The second term corresponds to a 3-D plate tensor and implies a curve or surface intersection with tangent e_3 that is perpendicular to the plane defined by e_1 and e_2. The last term corresponds to a 3-D ball tensor and implies a structure with no orientation preference. [8]

The voting process is similar to convolution with the difference that convolution produces scalar values and tensor voting produces tensors. Voting kernels which encode certain constraints such as smoothness and proximity are used. These voting kernels are continuous tensor fields that assign a value to every point within the domain. Any voting kernel, regardless of dimension, can be derived from the 2-D stick tensor, which is therefore referred to as the fundamental 2-D stick voting field (VF).

In its most general formulation, it takes as input points belonging to some N-dimensional space and encodes them as N-dimensional tensors. Tensor voting propagates the information of each tensor on its local neighborhood by way of tensor voting, ultimately creating a dense tensor field from the originally sparse input. Each tensor can be decomposed into features that have some geometric meaning, and each feature has a corresponding saliency component. In the 3-dimensional case, a tensor can be decomposed into three geometrically meaningful terms, the stick, plate and ball features.

A token's stick feature represents the surface normal at it's location while the plate feature represents a curve tangent vector. The saliency of the ball feature represents the degree to which the point appeases to have no orientation, describing neither surface nor curve, If the saliency of a token's stick feature is sufficiently high, the point is likely to lie on surface. Surfaces and curves can be extracted from the tensor saliency field using non-maximal suppression.

Each point in the range image has color data sampled from a color camera.

At a higher level of organization, a group of geometric primitives can be processed without any proximity assumption (unstructured) or with some smoothness constrains (structured). Examples of representation for a 1D manifold is a spline, while for a 2D manifold simple mesh or Non-uniform rational B-spline (NURBS) can be used.[16] When it comes to a noisy group of points, tensor voting can be used to interpolate shape on a dense 3D grid. The voting results can then be later process to extract 1D and 2D manifolds out of the dense volume.

In Table 1 you can see characteristics of primitives used for shape approximations. Number in parenthesis of the column Nb. Param. corresponds to the minimum number of parameters that can be used to express the same primitive.

Table 1

Characteristics of primitives used for shape approximations

Primitives	Parameters	Derivative	Manifold	Bound	Nb Param.
Point	*P*	0	0	-	*3 (3)*
Line	*l={p,t}*	1	1	*point*	*6 (5)*
Plane	*w={p,n}*	1	2	*line, curve*	*6 (5)*
Curve	*c={p,t.n,k}*	2	1	*point*	*10 (7)*
Quadric	*s={p.n.γ.K}*	2	2	*line, curve*	*14 (8)*

The shape representation is affected differently by transformation functions.

In application using point clouds, features arrive already sparse but not uniformly distributed. Nevertheless, the fact that sensors can provide a huge number of readings on a short period of time creates a bottleneck in term of computation power for the match function. Several techniques are used to reduce the number of features: random sampling, uniform grid, grid projection, octree.

All these techniques reduce the number of features without considering their distinctiveness. It does not involve higher-order derivatives. Only the first-order differential geometric property, i.e., normal, is 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.

4 Method overview

Give a triangular mesh $M = (V, \mathrm{E}, \mathrm{F})$, where $V = \{v_1, v_2, ..., v_n\}$ denotes the set of vertices, E denotes the set of edges and $F = \{f_1, f_2, ..., fm\}$ denotes the set of faces. Each vertex $v_i \in V$ is represented using Cartesian coordinates, denoted by $v_i = (v_{ix}, v_{iy}, v_{iz})$. Let $N_f(v_i)$ be the face indices of 1-ring neighbors of v_i. Method involves four main steps:

1. Initial feature vertex detection. The initial feature vertices are first extracted and classified into different types based on the modified normal tensor voting.

2. Salient measure computation. For each sharp edged type vertex, a novel salient measure is defined according to neighbor supporting. .

3. Weak feature enhancing. For detecting weak features, a weak feature enhancing technique is implemented.

4. Post-processing. The filtered feature vertices can be connected to generate feature lines. 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.

To further enhance the robustness of normal tensor voting to detect features on noisy meshes, we propose a novel salient measure benefiting from neighbor supporting, which is inspired by the following observation. A crest point has maximum curvature in its corresponding direction and a crest line naturally follows the direction of the minimum curvature of its composing crest point.

That is, the feature vertices lie on the principal curvature line. The vertex lying on a feature line is a feature vertex. In fact, if *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 *v* is a feature vertex, there will be a certain number of feature vertices along the principal curvature line to support.

The normal voting tensor of a vertex on a triangular mesh can be defined by the unit normal vectors of its neighbor triangles. First, the covariance matrix $V_v^{f_i}$ of the triangle f_i is written as

$$V_v^{f_i} = n_{f_i} n_{f_i}^T = \begin{pmatrix} a^2 & ab & ac \\ ab & b^2 & bc \\ ac & bc & c^2 \end{pmatrix} \tag{1}$$

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

The normal voting tensor of vertex ***v*** is defined by

$$T_v = \sum_{f_i \in N_T(v)} \mu_{f_i} n_{f_i} n_{f_i}^T \tag{2}$$

where μ_f is a weight given by (Kim *et al.*, 2009) [11]

$$\mu_{f_i} = \frac{A(f_i)}{A_{\max}} \cdot \exp\left(-\frac{\left\| c_{f_i} - v \right\|}{\sigma/3}\right), \tag{3}$$

and $A(fi)$ is the area of triangle fi, Amax is the maximum area among $Nf(\boldsymbol{v})$, cfi is the barycenter of triangle fi, and σ is the edge length of a cube that defines the neighboring space of each vertex. [12]

Because of the properties of second order symmetric tensors, the eigenvalues are real and positive or zero, and the eigenvectors form an orthonormal basis. A tensor can be decomposed into three components defined by

$$T_v = \lambda_1 e_1 e_1^T + \lambda_2 e_2 e_2^T + \lambda_3 e_3 e_3^T \tag{4}$$

where $\lambda_1 \geq \lambda 2 \geq \lambda_3 \geq 0$ are its eigenvalues and $\boldsymbol{e}_1$, $\boldsymbol{e}_2$, $\boldsymbol{e}_3$ are the corresponding unit eigenvectors.

To compute the salient measure for each sharp edge type vertex according to Eq. (1), two essential ingredients should be determined. One is the initial measure, Ω, and the other is the integral direction, ***t***

$$\Omega = \frac{\lambda_1 + \lambda_2 + \lambda_3}{2} - \frac{1}{2} \tag{6}$$

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. In the initial feature detection, λ_2 and λ_3 play important roles in vertex classification.

After the Ω's 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.

For the integral direction *t*, ideally, if a point belongs to a curve, the third eigenvector of its tensor must be aligned with the tangent to the curve at that point, and λ_3 must be zero. Thus, *t* can be naturally initialized by $\boldsymbol{e}_3$, which is called 'feature direction' At this point, we have the initial measure Ω and the feature direction *t*.

Summary

We study the tensor voting methodology in a modeling context by implementing a simple 3-D modeling tool. The user creates a surface from a set of points and normalizes. The user may interact with these tokens in order to modify the surface. We describe the results of our investigation.

Voting and surface extraction is slow for large models and large scales of analysis. The curve and junction capabilities of tensor voting should be explored for modeling sharp features and fine details. Each of these limitations should be addressed in future versions of a TV-based modeling system.

The contributions of our work can be summarized as follows:

1. Based on the idea of neighbor supporting, an anisotropic vertex salient measure is defined, which can effectively characterize the geometric features of the surface.

2. 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.

3. 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.

References

1. Coding Algorithms for 3DTV—A Survey / A. Smolic, K. Mueller, , N. Stefanoski, J. Ostermann, A. Gotchev, G. B. Akar, G. Triantafyllidis, A. Koz // IEEE Transactions on circuits and systems for video technology— vol.17, №. 11, –2007 – pp.1606-1621

2. *Computer Graphics: Principles and Practice /* Foley, J. D., van Dam, A., Feiner, S. K. & Hughes, J. F. // *C*, 2 edn, Addison-Wesley Publishing Company – 1995 -

3. Hoppe, H. Progressive meshes // *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* –- ACM, – New York, NY – USA - 96 – pp. 99–108.

4. Kircher, S.,Garland, M. Progressive multiresolution meshes for deforming surfaces // *Proceedings of the 2005 – ACM SIGGRAPH – Eurographics symposium on Computer animation* –ACM – New York, NY –USA, *2005* — pp. 191–200.

5. Matsuyama, T. Exploitation of 3d video technologies // *Proceedings of the International Conference on Informatics Research for Development of Knowledge Society Infrastructure* –ICKS '04 – IEEE Computer Society – Washington, DC, –USA, –2004 – pp. 7–14.

6. Real-time 3d shape reconstruction, dynamic 3d mesh deformation, and high fidelity visualization for 3d video / Matsuyama, T., Wu, X., Takai, T. & Nobuhara, S. // *Computer Vision and Image Understanding –№* 96(3): –2004 – *pp.*393–434.

7. Osharovskaya E. Evaluation spectral properties of mesh television objects /E. Osharovskaya, V. Solodka // Measuring and computing devices in technological processes – Khmelnitsky–2013 – №4 – pp.40-42.

8. Beltowska J. Investigations of Tensor Voting Modeling / J. Beltowska K. Museth D Breen // Communication papers – University Norrköping, – Sweden –2008 –№16-9 – pp 55-62.

9. Wang et al. Feature detection of triangular meshes via neighbor supporting */* Xiao-chao WANG, Jun-jie CAO2, Xiu-ping LIU, Bao-jun, Xi-quan, Yi-zhen SUN // Journal of Zhejiang University – SCIENCE C (Computers & Electronics) *– J Zhejiang Univ-Sci C (Comput & Electron) – 2012 13(6)* – pp.*440-451*

10. Yoshizawa, S. Fast, robust and faithful methods for detecting crest lines on meshes./ Yoshizawa, S., Belyaev, A., Yokota, H., Seidel, H.P.// *Comput. Aided Geom.* – 2008 *Des.*, 25(8): – pp. 545-560.

11. Kim, S.K Extraction of ridgesvalleys for feature-preserving simplification of polygonal models. / Kim, S.K., Kim, S.J., Kim, C.H.,// *LNCS* : – 2006. – 3992: –pp.279-286

12. Watanabe K. Detection of salient curvature features on polygonal surfaces. / K.Watanabe, A.G. Belyaev // *Comput. Graph. Forum* – 2001– №20(3):– pp.385-392.

13. Angelo L. Continuities detection in triangular meshes ./ di Angelo L., di Stefano P. // *Comput.-Aided Des* – 2010 –42(9):– pp.828- 839.

14. Guy, G. and Medioni, G. Inference of surfaces, 3D curves, and junctions from sparse, noisy 3D data./ G. Guy,. G. Medioni // IEEE Transactions on Pattern Analysis and Machine Intelligence –1997 – 19, No. 11 – pp. 1265–1277,

15. Ohtake Y Ridge-valley lines on meshes via implicit surface fitting / Ohtake Y., Belyaev, A., Seidel, H.P. // *ACM Trans. Graph* –, 2004 – 23(3):– *pp.* 609-612.

16. Lujun Wang Trivariate polynomial splines on 3D T-meshes // Dissertation for the degree of doctor of philosophy in Mathematics – Nashville, – Tennessee – 2012. – 66 P.

Рецензія/Peer review : 23.10.2014 р. Надрукована/Printed :23.10.2014 р.