Journal of Zhejiang University SCIENCE A
ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Curvature detail representation of triangular surfaces*

WU Jin-zhong[1,3], LIU Xue-hui[1], WU En-hua[1,2]
([1]State Key Lab of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100080, China)
([2]Department of Computer and Information Science, University of Macau, Macao, China)
([3]Graduate University of Chinese Academy of Sciences, Beijing 100039, China)
E-mail: wjz@ios.ac.cn; lxh@ios.ac.cn; EHWu@umac.mo
Received Apr. 10, 2006; revision accepted Apr. 19, 2006

Abstract: Curvature tells much about details of surfaces and is studied widely by researchers in the computer graphics community. In this paper, we first explain the mean-curvature view of Dirichlet energy of triangular surfaces and introduce a curvature representation of details, and then present surfaces editing applications based on their curvature representation. We apply our method to surfaces with complex boundaries and rich details. Results show the validity and robustness of our method and demonstrate curvature map can be a helpful surfaces detail representation.

Key words: Curvature, Detail representation, Mesh editing, Shape editing, Dirichlet energy
doi:10.1631/jzus.2006.A1210 **Document code:** A **CLC number:** TP391

INTRODUCTION

With the favorites of finer and vivid 3D models in visual reality, games and entertainment community, researchers among academic institutions and industrial organizations construct detail-rich models by 3D scanning tools and geometry modelling. These models usually contain millions of triangles and occupy hundreds and thousands of MB memories which give great difficulties for editing or operating onto them. A good representation of details makes these operations much easier. In this paper, we propose a detail representation of surfaces by its mean curvature information. Not just aiming to simplify mesh operations, our curvature detail representation has clear geometric explanation.

Previous work

First, we will list the technologies about detail representation, which are mainly used in mesh editing, shape deformation and animation. Second, curvature computation method and its applications in computer graphics are discussed.

1. Detail representation

In 3D triangular mesh space, geometry details can be represented in many ways. Ju *et al*.(2005) computed vertices weights by mean value coordinates method in 3D space onto a bounding coarse control mesh. The weights are used to compute deformed vertices position while control mesh deforms. The 3D mean value coordinates method is well-defined on both interior and exterior of the mesh surface and the deformation is real-time. Poisson equation-based method is another useful detail representation scheme and is studied in geometry completion, detail generation, mesh editing and deformation, etc. Poisson equation has its complex physical explanation but can be applied directly, on 2D images (Pérez *et al*., 2003) or on 3D meshes (Yu *et al*., 2004; Nguyen *et al*., 2005). A parameterization of 2D pixel RGB or 3D coordinates onto a planar region is needed to get the gradient map of the surface function *f*. With this boundary condition and guidance gradient map a target image or surface is constructed. Here gradient

* Project supported by the National Basic Research Program (973) of China (No. 2002CB312102), and the Research Grant of University of Macau, China

map works as details information in Poisson equation. Laplacian based method is also widely used as detail representation for mesh editing (Nealen *et al*., 2005; Au *et al*., 2005; Sorkine *et al*., 2004) or smoothing (Desbrun *et al*., 1999). In certain meaning, our mean curvature detail representation can be classified in this catalogue. But we explain the detail not as the displacement to its neighborhood but with a curvature meaning. Laplacian based method usually includes a sparse linear system, which makes it easy to operate and fast to compute. Igarashi *et al*.(2005) uses triangle shape and scales as the details of 2D triangular meshes which should be preserved during deformation. Lipman *et al*.(2005) represents the details as difference between local frames for neighbor vertices and deforms the object while keeping its original frame differences. All these detail representation methods are practically efficient and some of them have their geometrical explanations. The detail representation proposed in this paper relates curvature of surfaces and Dirichlet energy of its triangular approximation.

2. Curvature computation and applications

In differential geometry, curvature is mainly studied on continuous surfaces. To deal with triangular meshes, Meyer *et al*.(2002) defines mean curvature normal operator and Gaussian curvature operator with equations discretized from continuous surfaces. Other schemes base on curvature tensor method (Taubin, 1995) or on the theory of normal cycles (Cohen-Steiner and Morvan, 2003). We use scheme (Meyer *et al*., 2002) in which curvature can be formulated as a linear system of vertex position, simplifying latter operations. Curvature guides many applications in computer graphics, for mesh segmentation of CAD models (Guillaume *et al*., 2004) and improves mesh parameterization of general meshes (Yamauchi *et al*., 2005), for mesh saliency (Lee *et al*., 2005), for mesh fairing while keeping desirable geometric features (Alliez *et al*., 2003; Desbrun *et al*., 1999). But curvature cannot be used in mesh editing easily as it is not as intuitive as vertex position. This paper proposes a mesh editing method guided by its mean curvature representation.

CURVATURE DETAIL REPRESENTATION

Definitions and notations

In continuous 3D surface space, curvature is defined as the derivative of the tangent vector of space lines. Mean curvature is half of the sum of the two principal curvatures k_1 and k_2 of a point x_i on a surface. Its discretization on triangular meshes is

$$\boldsymbol{K}_i = \left\{ \sum_{j\in H_i} (\cot\alpha_{ij} + \cot\beta_{ij})(\boldsymbol{x}_i - \boldsymbol{x}_j) \right\} \Big/ 2\boldsymbol{A}_i ,$$

$\boldsymbol{K}_i$ is the mean curvature normal operator.

For a surface $f:\mathbb{R}^2 \to \mathbb{R}^3$ parameterized on a close region $\Omega \subset \mathbb{R}^2$, usually an energy function is present as global metric. For most energy functions, Dirichlet energy function is among the most frequently used and is defined as $E_\mathrm{D} = \int_\Omega |\nabla f|^2/2$, where ∇f is the gradient operator of f. Its discretization is

$$E_\mathrm{D}(M) = \left\{ \sum_{(i,j)\in E} (\cot\alpha_{ij} + \cot\beta_{ij}) |\boldsymbol{x}_i - \boldsymbol{x}_j|^2 \right\} \Big/ 4.$$

$\partial E_\mathrm{D}(M)/\partial \boldsymbol{x}_i = \left\{ \sum_{j\in H_i} (\cot\alpha_{ij} + \cot\beta_{ij})(\boldsymbol{x}_i - \boldsymbol{x}_j) \right\} \Big/ 2$ is its derivative. The discretization of mean curvature and Dirichlet energy are deduced many ways (Meyer *et al*., 2002; Desbrun *et al*., 1999; 2002; Pinkall and Polthier, 1993), but they use them for different purposes.

Base presentation

From previous discussion, it is obvious that $\partial E_\mathrm{D}(M)/\partial \boldsymbol{x}_i = \boldsymbol{K}_i \boldsymbol{A}_i$. This form views curvature as global Dirichlet energy metric for triangular meshes. It shows that the vertex differential of Dirichlet energy is twice the product of its mean curvature normal and area operators, and that it is an intrinsic property for meshes. Based on this analysis, we propose a curvature detail representation for triangular meshes and assume this detail representation contains the local and global information. Not like other proposed methods just aimed at practicality, our method has firm mathematical basic.

For an unknown mesh M and its guidance mean curvature normal operator k, we aim to construct the mesh surface f with its vertex derivative of Dirichlet energy equaling to K. This is done quadratically by minimizing function $E_\mathrm{Q}(f) = \sum_i \left| \partial E_\mathrm{D}(f)/\partial \boldsymbol{x}_i - \boldsymbol{K}_i \boldsymbol{A}_i \right|^2$.

When f is discretized triangularly with a certain vertex connection, minimizing E_Q is equal to minimizing a linear system $|\boldsymbol{\Phi x} - \boldsymbol{d}|^2$, which can be solved efficiently by optimization methods. When guidance $\boldsymbol{K}$ changes, so does the mesh surface. This observation builds the key base of our scheme.

Validity checks of E_Q is introduced by construction of minimal surfaces for open surfaces with fixed boundaries, with setting $\boldsymbol{KA}$=0. It shows the feasibility of our method.

IMPROVED DETAIL REPRESENTATION

Curvature coefficients

Base scheme needs improvements, as it is not rotation or scaling invariant. This drawback is mainly caused by $\boldsymbol{K}$, a triple-vector attached to each vertex, whose direction and length remain the same while it rotates and scales. It should rotate and scale correspondingly with the deformation. The relative angle and ratio between neighboring $\boldsymbol{K}_i$ and $\boldsymbol{K}_j$ need to be preserved before and after rotation and scaling, so do the details.

Furthering the base representation we define the localization of $\boldsymbol{K}_i\boldsymbol{A}_i$ as a linear equation of its neighbor vertices $\boldsymbol{K}_i\boldsymbol{A}_i = \sum_{j\in H_i} c_{ij}\boldsymbol{K}_j\boldsymbol{A}_j$. Here curvature coefficients $\{c_{ij}\}$ contain the angle and ratio information which do not change while editing, aiming to preserve original mesh details. $\boldsymbol{K}_i$ and $\boldsymbol{K}_j$ are triple-vectors, mathematically speaking, $\boldsymbol{K}_i$ can be uniquely represented by three $\boldsymbol{K}_j$ if they are not in the same plane. When this occurs, other $\boldsymbol{K}_j$ will be useless and the solution of c_{ij} will be singular, which is undesirable. This drawback must be avoided. We put another constraint $\sum_{\min} c_{ij}^2$ on it to get a reasonable solution. Though this additional constraint may consume more time, this part of the work can be precomputed before mesh editing.

Another two candidates are also tested to eliminate rotation and scaling problem, one is to rotate and scale $\boldsymbol{K}$ correspondingly, and the other is to substitute $\boldsymbol{KA}$ with other scalar, for example principal curvature k_1 or k_2. The former one can be comprised in our scheme while the latter one is always unsteady, because of its two-order derivative property. This rotation and scaling problem is common in mesh editing; it can be solved by interpolating (Yu *et al*., 2004) according to the deformation or by putting the rotation and scaling metrics in the optimization stage (Sorkine *et al*., 2004). Not like either of them, we solve it as localized curvature coefficient of its neighbors.

At the stage of editing, curvature normal constraints are imposed on some vertices. The curvature normal of other vertices are derived by the constraints and curvature coefficients $\{c_{ij}\}$. This can be done by solving a linear system or a quadratic minimization function:

$$\boldsymbol{C}_{ij}\begin{pmatrix}\boldsymbol{K}_0\boldsymbol{A}_0\\ \boldsymbol{K}_i\boldsymbol{A}_i\end{pmatrix}=\begin{pmatrix}\boldsymbol{K}_0\boldsymbol{A}_0\\ \boldsymbol{K}_i\boldsymbol{A}_i\end{pmatrix} \text{ and } \sum_{\min}\left|(\boldsymbol{C}_{ij}-\boldsymbol{I})\begin{pmatrix}\boldsymbol{K}_0\boldsymbol{A}_0\\ \boldsymbol{K}_i\boldsymbol{A}_i\end{pmatrix}\right|^2.$$

$\boldsymbol{K}_0\boldsymbol{A}_0$ are known curvature normal constraints and unknown $\boldsymbol{K}_i\boldsymbol{A}_i$ are solved accordingly. In our implementations, the latter is used.

Embedding

Two optimization stages are involved in our scheme, one is curvature optimization stage (Section 3.1) and the other is embedding optimization stage (Section 2.2). After curvature is specified for some vertices, curvature optimization method is used to compute the best approximation curvature normal to the original surface, for other unspecified vertices. The approximation metric is the minimal of the optimization function. When curvature for every vertex is known, embedding optimization is used to get the position for unspecified vertices, with user selected vertices' positions being specified, as the curvature optimization stage does. In this section, we demonstrate the applications of our curvature-detail representation.

Mesh editing

A region of interest for editing is specified in our demonstration. Users can select any inner vertices, modifying their 3D coordinates or curvature normal direction and length. With these constraints and precomputed curvature coefficients, embedding is constructed directly. As Fig.1 shows, just a few vertices are modified, the whole model deformed accordingly.

The more vertices are modified with specified 3D coordinates or curvature normal, less time is consumed at optimization stage. If less unknown vertices remain, more user interaction is needed. In our implementation, boundary position and curvature are fixed for convenience, but actually this is not a necessary condition, they also can be edited and modified.

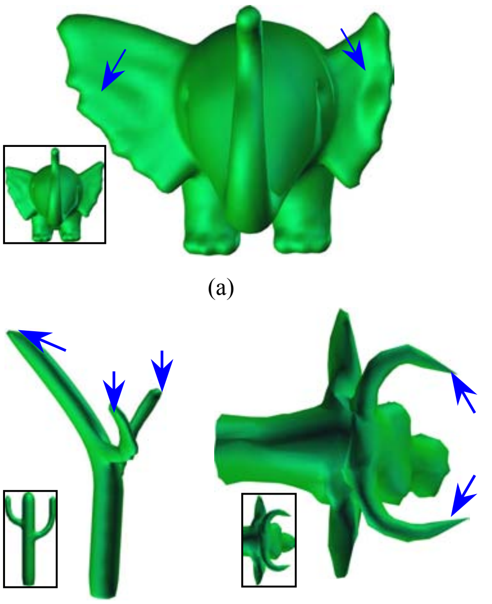
(a)
(b) (c)

Fig.1 Mesh editing using curvature detail representation. (a) Elephant model with left ear lengthened and right ear deformed towards readers; (b) Operation on three features points of cactus model and other vertices are embedded accordingly, with precomputed curvature coefficient; (c) Curvature normal of feature points of the horn is shortened, which means the horn must be lengthened. Whole head is reconstructed for this operation.

Shape editing

Big deformation such as shape editing is also available with our scheme. In this application, the boundary vertices coordinate or curvatures normal are modified. Fig.2 shows the results. The whole head of a cow is selected as region of interest. Curvature normal directions of the feature points on it are modified towards readers, and then the embedding is constructed. For terrain patch, large deformation of the boundary is available. Meaningful results are optimized by our scheme, even big deformation are present.

Remeshing

Remeshing and retriangulariztion are implemented without any special settings, except that a parameterization is present to map curvature information from source surfaces to its retriangularization. Various complex surfaces, with complex boundaries and long and narrow parts are used to test our representation scheme. As shown in Fig.3, our scheme can generate good results for detail rich surfaces.

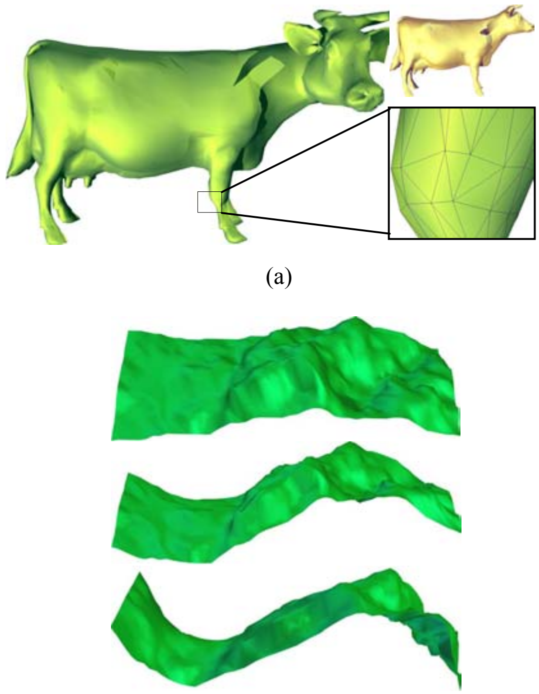
(a)
(b)

Fig.2 Shape editing of cow model (a) and terrain patch (b). (a) The head of cow is moved towards readers and knee of the leg is shown when it is not salient in the original model; (b) The boundary of original terrain patch (top) is modified with sine function in *y* (middle) and *z* (bottom) directions. Curvature normal of boundary is modified accordingly.

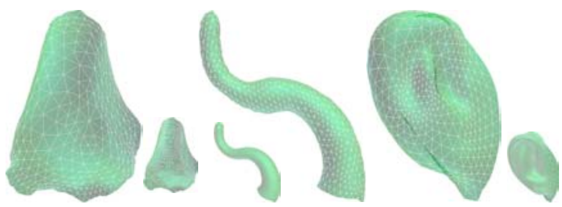
(a) (b) (c)

Fig.3 The remeshing of three different complex surfaces. (a) and (c) are the nose and ear surfaces extracted from the mannequin model; (b) is the nose of an elephant model. These surfaces have complex boundaries or long and narrow parts. Curvature normal information is mapped from original model onto its retriangularization, with the help of mesh parameterization.

TEST AND FUTURE WORK

Our applications are implemented on PC with platform Windows 2000, 2.5 GHz CPU and 512 MB main memory. Time consuming parts are the two optimization stages, which is dominated by the size of $\boldsymbol{\Phi}_{ij}$ and $\boldsymbol{C}_{ij}$. Table 1 illustrates the data collected. The number of vertices and the time spent in optimization procedure are listed.

Table 1 Size of test surfaces and time for optimization procedure

Models	Vertices	Facets	Optimization time (s)
Nose1	601	1135	7.4
Nose2	383	732	2.6
Nose3	874	1717	35.9
Ear	951	1851	113.3
Cat	135	257	0.5

Curvature is an important property of surfaces. But because of its second order differential property, it changes drastically even on smooth surfaces. This drawback makes it difficult to use it directly in mesh editing and mesh operation, preventing it from wider use. This paper proposes a curvature detail representation to solve the mesh editing problem with curvature. It is rotation and isotropic scaling invariant, which makes it easy for editing. The test result shows that our method works well for mesh editing, shape editing and remeshing. Our scheme enriches the tools used for mesh operations.

Further work is needed to widen the applications of curvature detail representation, such as morphing between multi-surfaces or putting desired features of different models together. The seamless connection between different parts is anther concern.

References

Alliez, P., Cohen-Steiner, D., Devillers, O., Levy, B., Desbrun, M., 2003. Anisotropic polygonal remeshing. *ACM Trans. on Graphics*, **22**(3):485-493. [doi:10.1145/882262.882296]

Au, O.K.C., Tai, C.L., Fu, H.B., Liu, L.G., 2005. Mesh Editing with Curvature Flow Laplacian. Proceedings of Symposium on Geometry Processing, poster.

Cohen-Steiner, D., Morvan, J.M., 2003. Restricted Delaunay Triangulations and Normal Cycle. Proc. 19th Annu. ACM Sympos. Comput. Geom., p.237-246.

Desbrun, M., Meyer, M., Schroder, P., Barr, A.H., 1999. Implicit Fairing of Irregular Meshes Using Diffusion and Curvature Flow. Proceedings SIGGRAPH'99, p.317-324.

Desbrun, M., Meyer, M., Alliez, P., 2002. Intrinsic parameterizations of surface meshes. *Computer Graphics Forum*, **21**(3):209-218. [doi:10.1111/1467-8659.00580]

Guillaume, L., Florent, D., Atilla, B., 2004. Curvature Tensor Based Triangle Mesh Segmentation with Boundary Rectification. Proc. Computer Graphics International 2004 (CGI2004), p.10-17. [doi:10.1109/CGI.2004.1309187]

Igarashi, T., Moscovich, T., Hughes, J.F., 2005. As-rigid-as-possible shape manipulation. *ACM Trans. on Graphics*, **24**(3):1134-1141. [doi:10.1145/1073204.1073323]

Ju, T., Schaefer, S., Warren, J., 2005. Mean value coordinates for closed triangular meshes. *ACM Trans. on Graphics*, **24**(3):561-566. [doi:10.1145/1073204.1073229]

Lee, C.H., Varshney, A., Jacobs, D.W., 2005. Mesh saliency. *ACM Trans. on Graphics*, **24**(3):659-666. [doi:10.1145/1073204.1073244]

Lipman, Y., Sorkine, O., Levin, D., Cohen-Or, D., 2005. Linear rotation-invariant coordinates for meshes. *ACM Trans. on Graphics*, **24**(3):479-487. [doi:10.1145/1073204.1073217]

Meyer, M., Desbrun, M., Schröder, P., Barr, A.H., 2002. Discrete Differential-Geometry Operators for Triangulated 2-Manifolds. Visualization and Mathematics III, p.35-57.

Nealen, A., Sorkine, O., Alexa, M., Cohen-Or, D., 2005. A sketch-based interface for detail preserving mesh editing. *ACM Trans. on Graphics (SIGGRAPH'05)*, **24**(3):1142-1147. [doi:10.1145/1073204.1073324]

Nguyen, M.X., Yuan, X.R., Chen, B.Q., 2005. Geometry completion and detail generation by texture synthesis. *The Visual Computer*, **21**(8-10):669-678. [doi:10.1007/s00371-005-0315-1]

Pérez, P., Gangnet, M., Blake, A., 2003. Poisson image editing. *ACM Trans. on Graphics (SIGGRAPH'03)*, **22**(3):313-318. [doi:10.1145/882262.882269]

Pinkall, U., Polthier, K., 1993. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, **2**(1):15-36.

Sorkine, O., Lipman, Y., Cohen-Or, D., Alexa, M., Rössl, C., Seidel, H.P., 2004. Laplacian Surface Editing. Proceedings of Symposium on Geometry Processing, p.179-188.

Taubin, G., 1995. Estimating the Tensor of Curvature of a Surface from a Polyhedral Approximation. Proceedings of the Fifth International Conference on Computer Vision, p.902-907. [doi:10.1109/ICCV.1995.466840]

Yamauchi, H., Gumhold, S., Zayer, R., Seidel, H.P., 2005. Mesh segmentation driven by gaussian curvature. *The Visual Computer*, **21**(8-10):659-668. [doi:10.1007/s00371-005-0319-x]

Yu, Y.Z., Zhou, K., Xu, D., Shi, X.H., Bao, H.J., Guo, B.N., Shum, H.Y., 2004. Mesh editing with poisson-based gradient field manipulation. *ACM Trans. on Graphics*, **23**(3):644-651. [doi:10.1145/1015706.1015774]