

A Ridge-Lines-Based Interface for Triangle Mesh Deforming

Yuhui Hu, Juan Lin, Baoquan Zhao, Shujin Lin, and Xiaonan Luo

Abstract—Most surface in computer graphics are represented as triangle meshes. Techniques for interactive deformation of triangle meshes are a fundamental important part in a host of applications. Most traditional approaches to the deformation have emphasized precise control over the models by a man made selection of a set of control point. However, they are often cumbersome and non-intuitive for the non-expert users. In this paper, we present a system for deforming triangle meshes which is easy to use and need less interaction. Our method computes the high order curvature derivatives describing the salient features of meshes, such as the ridge lines. By interacting with the ridge lines as the handle to control the deformation, the user can implicitly and intuitively control the deformation of all meshes. We demonstrate that our system can make the surface manipulated and modified by preserving the geometric details.

Index Terms—Ridge line, laplacian deformations, differential representations.

I. INTRODUCTION

A few lines suffice to sketch the main feature of a shape. This is why many designers still prefer using the pen and paper to communicate, and explains the great success of sketch-based model editing, such as Teddy [1] and FiberMesh [2]. Laying out the coarsest level silhouette and then modify control points to generate details is often a confusing task for non-expert users, so we propose an intuitive interface for mesh deformation by automatically finding ridge lines which describe the salient features of meshes. Combined, these lines can act as the control points during the deformation.

In our system, a 3D model is automatically generated by a Boolean operation among the silhouettes from front, side, and top views. After the mesh subdivision, we usually get a model with some noises. It's a tedious work for non-expert users to select the precise control points to make sure the deformation has got a desired result. Our system provides a

great deal of flexibility to the user. We successfully avoid the tedious work and the unintuitive work by automatically finding the ridges. Thus, much interaction between users and computers can be omitted, and the ridges successfully sketch a reference curve for the users.

II. RELATED WORK

A great deal of work has been done in the past on developing techniques for the modeling and deforming in Computer Graphics. Here we just survey some most relevant work with an emphasis technique for interactive deformation.

Lattice-based deformation as Free-From Deformations (or rename as lattice-deformation) are one of the most important techniques for deformation [3]-[5]. While their complex control lattice provide precise control over the resulting deformation, editing such lattices can be an unintuitive process, and can not guarantee a well detail- preserved work.

Curve-based deformation approaches such as sketch mesh deformation [6]. Their system uses a curve-based approach with an emphasis on specifying a free-from control curves to capture the structure of surfaces. Another approach for 3D modeling and deformation is the use of freeform strokes for 2D applications [1]. Their system uses some single 2D strokes as the tools to create the 3D shape and the tools to control the 3D deformation. There is also some interaction between user and system while editing those curves or skeleton for some complex models. In order to reduce the interactive work, our system brings the ridge lines as the free-form control curves during the deformation.

III. LAPLACIAN COORDINATE AND RIDGES EXTRACTION ON MESHES

A. Laplacian Coordinate

For a mesh model, we usually let $M=(V, E, F)$ to present its' structure. V denotes the set of vertexes, E denotes the set of edges, and F denotes the set of faces. Each vertex $V_i \in V$ is represented by Cartesian coordinates, denotes by $V_i = (x_i, y_i, z_i)$. Compute the center of mass of the neighbors of V_i , then the Laplacian coordinate of V_i will be easily got:

$$\delta_i = (\delta_i^x, \delta_i^y, \delta_i^z) = v_i - \frac{1}{d_i} \sum_{j \in N(i)} v_j$$

where $N(i)=\{j|(I, j) E\}$ and means the number of neighbors of vertex (the degree of). The detailed definitions is also

Manuscript received October 21, 2013; revised January 9, 2014.

Yuhui Hu and Baoquan Zhao are with the National Engineering Research Center of Digital Life, State-Province Joint Laboratory of Digital Home Interactive Applications, School of Information Science & Technology, Sun Yat-sen University, Guangzhou 510006, China (e-mail: huylh5@mail2.sysu.edu.cn, zbqsys@gmail.com).

Juan Lin is with the National Engineering Research Center of Digital Life, State-Province Joint Laboratory of Digital Home Interactive Applications, School of Software, Sun Yat-sen University, Guangzhou 510006, China (e-mail: linjuan515@gmail.com).

Shujin Lin is with the School of Communication and Design, Sun Yat-sen University, Guangzhou, 510006, China (e-mail: linshjin@mail.sysu.edu.cn).

Xiaonan Luo is with School of Information Science & Technology, Sun Yat-sen University, Guangzhou, 510006, China, and Research Institute of SunYat-sen University in Shenzhen, Shenzhen ,518057,China (e-mail: lnslxn@mail.sysu.edu.cn).

introduced in [7].

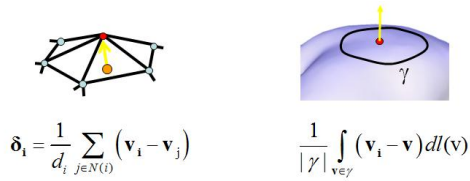


Fig. 1. The vector of the differential coordinates at the red vertex approximates the local shape characteristics of the surface: the normal direction and the mean curvature.

From a differential geometry perspective, the Laplacian coordinates can also write as follows if we assume that our mesh M is a piecewise-linear approximation of a mesh surface.

$$\delta_i = \frac{1}{d_i} \sum_{j \in N(i)} (v_i - v_j) = \frac{1}{|r|} \int_{v \in r} (v - v_i) dl(v)$$

where r is a closed simple surface curve around, and $|r|$ means the length of r . It is known from differential geometry that

$$\lim_{|r| \rightarrow 0} \frac{1}{|r|} \int_{v \in r} (v - v_i) dl(v) = -H(v_i) n_i$$

where $H(v_i)$ is the mean curvature at V_i , and n_i is the surface normal. Therefore, the Laplacian coordinate encapsulate the local surface shape to some extent.

B. Ridge Extraction on Meshes

Given a smooth surface, a ridge is a curve along which one of the principal curvatures has an extremum along its curvature line [8]. Ridges are curves of extremal curvature, so they describe the salient features of meshes to some extent.

For a detailed introduction to ridges and related topics, the reader can consult [9], as well as the following article [10]. Consider a smooth surface, we denote $K1$ and $K2$ the principal curvatures, with $K1 \geq K2$. And we denote the corresponding principal directions $d1$ and $d2$. In local coordinates, we denote \langle, \rangle the inner product induced in Euclidean space, and $dK1, dK2$ the gradients of the principal curvatures. Then a ridge point is defined by:

- A ridge point if the extremality coefficient $b_0 = \langle dK1, d1 \rangle$ ($b_3 = \langle dK2, d2 \rangle$) vanished, i.e. $b_0 = 0$ ($b_3 = 0$)

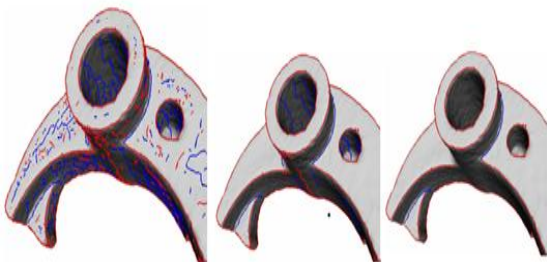


Fig. 2. Mechanical part (37k pts): all lines(left),lines filtered with the strength(middle)and lines filtered with the sharpness(right). Notice that any point on a flat or cylindrical part lies on two ridges, so that the noise observed on the left two Figs. is unavoidable. It is however easily filtered out with the sharpness on the right figure.

For a real word applications dealing with coarse meshes, or meshes conveying some noise, or meshes featuring sharp features, the ridge lines we had found cannot be met the real feature of models. In that case, we must found a threshold to filter the ridges. Such algorithm for filtering can also be founded in [10], [11].

IV. DEFORMATION TECHNIQUES

The primary deformation we use is operated on the Laplacian coordinate. The main idea of Laplacian deforming algorithm is to keep the vertexes' Laplacian coordinate as possible as we can, thus can make sure that the feature of shape has been preserved.

In the following, we will introduce a matrix L to present the transformation of the vector of absolute Cartesian coordinate to the vector of Laplacian coordinate.

$$LV=W$$

where $V = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \dots \\ x_n & y_n & z_n \end{bmatrix}$ means the vector of Cartesian

coordinate, $W = \begin{bmatrix} \delta_1^x & \delta_1^y & \delta_1^z \\ \delta_2^x & \delta_2^y & \delta_2^z \\ \dots \\ \delta_n^x & \delta_n^y & \delta_n^z \end{bmatrix}$ means the vector of

Laplacian coordinate.

Before we construct the matrix L , we bring two matrix A and D . Matrix A is the adjacency matrix of the mesh

$$A_{ij} = \begin{cases} 1 & (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

And matrix D is a diagonal matrix where $D_{ii} = d_i$. Then Laplacian matrix L can computes as follows:

$$L = I - D^{-1}A$$

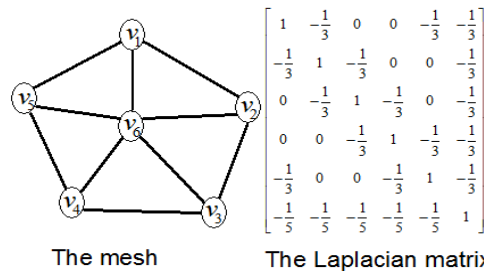


Fig. 3. A small example of a triangular mesh and its associated Laplacian matrix.

As we can see, the sum of every row of L is zero, which implies that we can not use this following formula to compute the new Cartesian coordinate of vertexes after the deformation.

$$V' = L^{-1}W'$$

In order to make the Laplacian coordinates can uniquely restore the global Cartesian coordinates, we usually will add some control point into the equation $LV=W$. Therefore the equation will be changed as:

$$\begin{pmatrix} L \\ I \end{pmatrix} V = \begin{pmatrix} W \\ C \end{pmatrix}$$

The same as $\tilde{L}V = \tilde{W}$, where C means the vector of control point's Cartesian coordinate, and I means such control points' unit vector (the j -th element is 1, the others is 0) (ex. $I_3 = (0,0,1,0,\dots,0)$).

The additional control point make the above linear system over-determined (more equation than unknowns) and always have no exact solution existed. However, the equation is full-rank and thus has a unique solution in the least-squares sense:

$$V' = \arg_v \min \left(\|LV - W\|^2 + \sum_{v_j \in C} |v_j - c_j|^2 \right)$$

The least-squares solution can also expressed in terms of matrix form:

$$V' = \begin{pmatrix} \tilde{L}^T & \tilde{L} \end{pmatrix}^{-1} \tilde{L}^T \tilde{W}$$

V. RESULT AND APPLICATIONS

In this section, we consider several examples of using our system to edit the triangle meshes. All the results were on a standard consumer-level Windows PC, and the algorithm is implemented in the environment of VC.

An example-step in our system is shown as follows:

Step1: import a 3D model and display it on the user interface;

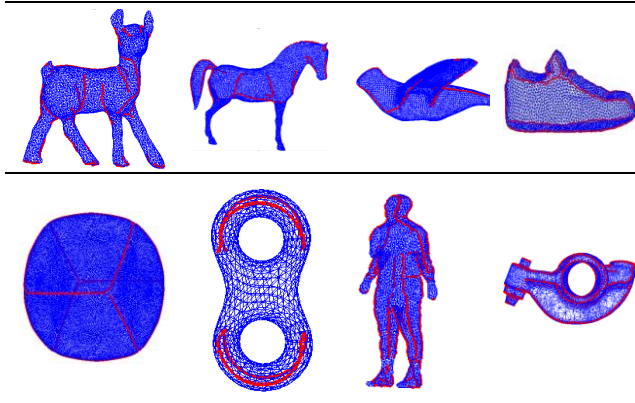
Step2: rotate or scale the model to an appropriate position and a suitable size.

Step3: click the ridge-found button to find some ridge lines automatically for the user's interaction reference.

Step4: by editing the reference ridge lines to control the model procession operations.

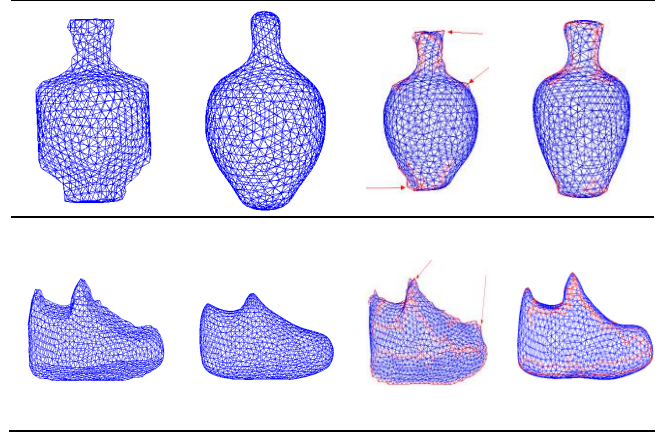
In the experiment, we use some models in our system as examples. The results of finding ridges are shown in Table I.

TABLE I: THE RIDGES ON DIFFERENT MODELS



For the rough model, especially which have some distinct edges and corners, the smooth operation becomes necessary. Usually we use the shrinkage by normal or the Laplacian smoothing. During the experiment, we found that shrinkage by normal cannot keep the model's sketch well, the same as we do it by adding some control. So during the experiment, we add the ridge lines as the con-trolling point in Laplacian Smoothing. By giving different weight factor for ridge points and the others to control the result of smooth, we have got the satisfactory result. That means we can use the following formula to do the Laplacian smoothing. An example of the result for experiments has shown in Table II.

TABLE II: AN EXAMPLE OF THE EXPERIMENT



$$\begin{pmatrix} L \\ \omega I \end{pmatrix} V = \begin{pmatrix} \lambda W \\ \omega C \end{pmatrix}$$

In the experiment, we set the weight factor $\omega=0.9$ to regulate the strength of restriction, that means the ridge line vertices will smooth fluently. And we set the weight factor $\lambda=0$, that means the other vertices will be close to the centroid of their adjacent vertices as much as possible.

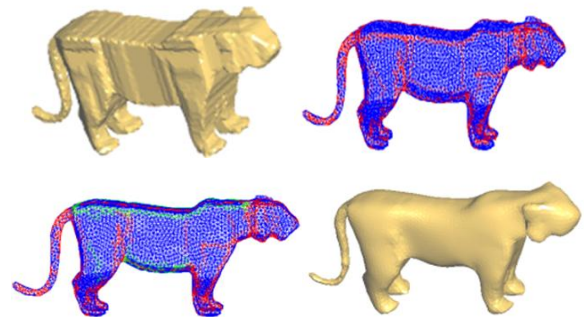


Fig. 4. By using different weight factor λ to control the result of smoothing, we can get a satisfactory model.

As shown in Table II, the first column was the original model; If we shrink all the vertexes by their normal vector, then we can not hold the profile well, the result can be seen in the second column; If we keep some feature vertexes immobile, and shrink the others by the normal vector, the feature vertexes will stick out and become the noise point, as shown in the third column; The fourth column show the model after smoothing by using our method. Not only do it become smooth, but also do it keep the profile well.

In the experiment, we can also give different weight factor λ to different ridges so as to control the result of smoothing.

Fig. 4 demonstrates a smoothing deformation by using different weight factor λ . We can clearly see that the original tiger's back is too rough and flat. After finding the ridge lines, we can add some additional lines on the tiger's back to control the smoothing. Also, the same operate can be easily done to pick out some ridges lines, such as the green lines in the third picture, to make it to be shrunk more. Thus we can get a more satisfactory model.

VI. CONCLUSION

We have proposed a new and intuitive approach to interactive deformation of triangle meshes. Users of our system can make significant edits to 3D models by simply controlling the ridge-lines on models. The reference ridges found by system automatically, serve as a handle for the model procession operations, such as the Laplacian deformation and the normal shrink.

The ridge-lines-based interface is a very effective tool, but there are also numerous ways in which it could be improved. Considering the ridges of a smooth surface are point with prescribed differential properties, our system must guarantee that differential quantities are available for each vertex of the mesh. So we can only deal with the mesh which is dense enough and meet the conditions of the acute rule [10]. Although our system is suitable for interactive editing of fairly large models, there is some cases that our system would be very desirable, such as the case on a plane or a cylinder, because all points are ridge point in such case, and we have filter these point by setting the threshold. While we are satisfied with the simplicity of current ridge-based interface, the extended model operations are need more accurate controlling handle, and careful interface design will be required.

ACKNOWLEDGMENT

This research is supported by the National Key Basic Research and Development Program of China (973) (No. 2013CB329505), the National Natural Science Foundation of China (61320106008, 61232011, 61103162), the National Key Technology R&D Program (No. 2011BAH27B01, 2011BHA16B08).

REFERENCES

- [1] I. Takeo, S. Matsuoka, and H. Tanaka, "Teddy: A sketching interface for 3D freeform design," ACM SIGGRAPH 2007 courses, ACM, 2007.
- [2] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa, "FiberMesh: Designing freeform surfaces with 3D curves," ACM SIGGRAPH 2007 papers, San Diego, California, August 05-09, 2007.
- [3] S. Coquillart, "Extended free-form deformation: A sculpturing tool for 3D geometric modeling," *ACM SIGGRAPH Computer Graphics*, vol. 24, no. 4, pp.187-196, Aug. 1990.
- [4] T. W. Sederberg and S. R. Parry, "Free-Form deformation of solid geometric models," *ACM SIGGRAPH Computer Graphics*, vol. 20, no. 4, pp. 151-160, Aug. 1986.
- [5] R. MacCracken and K. I. Joy, "Free-Form deformations with lattices of arbitrary topology," in *Proc. the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, August 1996, pp. 181-188.
- [6] Y. Kho and M. Garland, "Sketching mesh deformations," in *Proc. the 2005 Symposium on Interactive 3D Graphics and Games*, Washington, District of Columbia, April 03-06, 2005.
- [7] O. Sorkine, "Differential representations for mesh processing," in *Proc. Computer Graphics Forum 25*, 2006, vol. 4, pp. 789-807.
- [8] F. Cazals and M. Pouget, "Topology driven algorithms for ridge extraction on meshes," Research Report RR-5526, INRIA, 2005.
- [9] P. W. Hallinan, G. G. Gordon, A. L. Yuille, P. Giblin, and D. Mumford, "Two- and three-dimensional patterns of the face," A. K. Peters, Ltd., Natick, MA, 1999.
- [10] F. Cazals and M. Pouget, "Smooth surfaces, umbilics, lines of curvatures, foliations, ridges and the medial axis: Selected topics," *Int. J. of Computational Geometry and Applications*, vol. 15, no. 5, pp. 511-536, 2005.
- [11] S. Kim and C. Kim, "Finding ridges and valleys in a discrete surface using a modified MLS approximation," *Computer-Aided Design*, vol. 37, no. 14, pp. 1533-1542, 2005.



Yuhui Hu is currently a master degree candidate in the Department of Information Science and Technology, Sun Yat-Sen University. He received the B.S. degree in the Department of Computer Science from the AnHui University of Technology in 2012, his current research interests include computer graphics, geometric modeling and processing.



Juan Lin obtained her M.S. degree in the School of Software from the Sun Yat-sen University of China in the summer of 2013. She received the B.S. degree in the School of Information Science and Technology from the Zhongkai University of Agriculture and Engineering in 2011, her research interests include computer graphics.



Baoquan Zhao is currently a master degree candidate in the Department of Information Science and Technology, Sun Yat-Sen University. He received the B.S. degree in the Department of Computer Science from the China University of Geosciences, Wuhan in 2012, his current research interests include computer graphics and information visualization.



Shujin Lin obtained her Ph.D. degree in computer sciences from the Sun Yat-sen University of China in 2008. Now she is an associate professor at Sun Yat-sen University. Her search interests include computer graphics and information visualization.



Xiaonan Luo is currently a professor of School of Information and Science Technology at Sun Yat-Sen University. He received the Ph.D. degree in computational mathematics from Dalian University of Technology in 1991. He was granted the government special allowance by the State Council of China and the National Science Fund for Distinguished Young Scholars by the National Natural Science Foundation of China. His research interests include computer graphics, image processing, and mobile computing.