Mesh Editing with Feature Region Preservation

Jianwei Hu*
School of Mathematics and Statistics, Huangshan University, Tunxi, China
Email: jianweihu@hsu.edu.cn

Gu Song
School of Mathematics and Statistics, Huangshan University, Tunxi, China
Email: lvsonggu@hsu.edu.cn

Juan Cao
School of Mathematical Sciences, Xiamen University, Xiamen, China
Email: juancao@xmu.edu.cn

***Abstract*—In computer graphics, some regions on a 3D model are often required to precisely preserve the features during editing. This paper proposes a new feature preserving mesh editing method. First we use one of the existing mesh editing methods to estimate the final editing result. Then we get the deformation function between this result and the input mesh. Using singular value decomposition on the deformation function of each faces in the feature region which specified by user, we extract the rigid component of the function. Our method keeps the global shape of the feature region, and avoids the scale and distortion that cause the visual unpleasing. We also propose a refined algorithm which makes the resulting mesh more reasonable. The framework produces visually pleasing editing results with simple user interaction.**

***Index Terms*—mesh editing, differential coordinates, singular value decomposition, feature preserving**

I. Introduction

Interactive mesh processing techniques have been intensively studied in computer graphics and computer-aided design. The aim of such research is to develop modeling tools for intuitively editing the shapes of 3D models while preserving their geometric details.

In the early years, free-form deformation (FFD) methods received extensive attention. They modify 3D space where objects are located instead of editing the shapes directly[1-3]. Since FFD does not directly work on mesh models, it is difficult to define geometric constraints on vertices, edges and faces.

Multi-resolution frameworks were first studied by Forsey and Bartels[4] for spline surfaces, and extended to subdivision surfaces and irregular meshes[5-8]. In order to achieve higher efficiency and increased numerical stability, these approaches decompose the original mesh into several levels of details. Mesh editing can be performed to each level. The details are transformed from the local frames of the lower level mesh. However, the shapes of accurate features are not easy to control during editing.

Recently, linear partial differential equations (PDE) based approaches have been published[9-13]. These approaches represent differential properties and vertex positions in a linear system. Users can directly specify the positions of parts of the vertices, and the positions of the rest of the mesh are computed by solving a linear system to preserve the local geometric features. These methods are intuitive and useful, but sometimes users may be disappointed when they want to maintain some feature regions of original model unchanged. Yasuhiro et al.[14] used hard and soft constraints by constructing a typical least squares matrix form to ease the problem. Later, Hiroshi et al.[15] extended this to a discrete framework for preserving the shapes of form-features. For man-made engineering objects, Zheng et al.[16] proposed a component-wise algorithm to automatically preserve the established inter-relations.

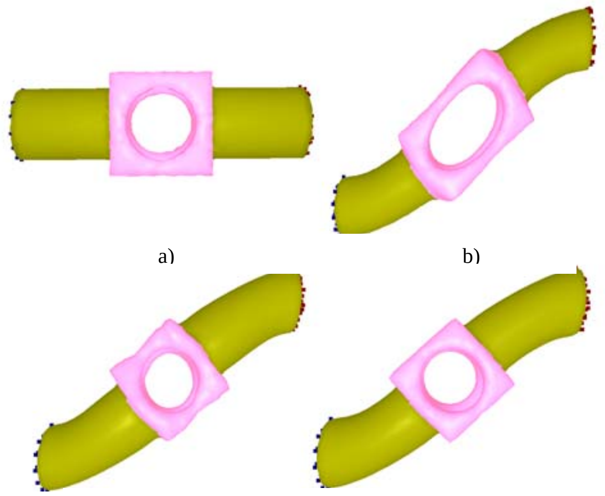
This paper presents an novel shape editing technique to maintain some features unchanged for 3D mesh models. First we use one of the existing mesh editing methods to estimate the final editing result. Then we get the deformation function between this result and the input mesh. Using singular value decomposition on the deformation function of each faces in the feature region which specified by user, we extract the rigid component of the function. Our method keeps the global shape of the feature region, and avoids the scale and distortion that cause the visual unpleasing. We also propose a refined algorithm which makes the resulting mesh more reasonable (Fig. 1d).

II. Feature Preserving Surface Editing System

Our approach is capable of preserving the shape of masked regions of the features while editing the mesh according to the user specifications. We would like to

*Corresponding author
This research received support from the National Natural Science Foundation of China (Grant No. 61100105), the Natural Science Foundation of the Anhui Higher Education Institutions of China (Grant No. KJ2010B212) and the Huangshan University Foundation, China (2011xkjq003).

doi:10.4304/jsw.8.11.2943-2946



a) b) c) d)

Figure 1. Comparison between dual Laplacian editing[17] and our method. a) The input model. b) The result of dual Laplacian editing[17]. c) Our rough result. d) Our final result.

preserve the shape of all the faces contained in the features, meaning that they should undergo solely a rigid transformation.

A. Shape Estimation

Our first step is to estimate the editing results by using one of the existing mesh editing methods. Since this estimating step and our editing algorithm (the next step) are two independent steps, any existing editing method can be used here. In this paper, we use dual Laplacian mesh editing for estimation.

We denote the editing method by T, and the input mesh by $G = (V, E, F)$, where $V = (v_1, v_2, \cdots, v_n)$ is the set of vertex positions, $E = \{(i, j)\}$ is the set of edges and $F = (f_1, f_2, \cdots, f_m)$ is the set of polygonal faces. A polygonal faces with k edges is denoted by $f_i = (v_{i_1}, v_{i_2}, \cdots, v_{i_k})$. The estimating result is denoted by $G' = T(G) = (V', E', F')$, where

$$V' = (v_1', v_2', \cdots, v_n'),\ E' = E = \{(i, j)\},$$

$$F' = (f_1', f_2', \cdots, f_m')$$

are the set of vertex positions, the set of edges and the set of faces after editing respectively. The new polygonal faces with k edges is denoted by $f_i' = (v_{i_1}', v_{i_2}', \cdots, v_{i_k}')$.

B. Extracting Rigid Transformation

We first devise a proper shape preserving transformation T_i for each face $f_i = (v_{i_1}, v_{i_2}, \cdots, v_{i_k})$ such that $f_i' = T_i(f_i)$. We approximate $T_i(f_i)$ with a rotation transformation, by taking the linear component of T_i and extracting the rotation from it by means of the polar decomposition. Specifically, denote the centroid of f_i by $v = \frac{1}{k}\sum_{j=1}^{k} v_{i_j}$; the centered vertices are denoted by $u_{i_j} = v_{i_j} - v$ (and similarly, $u_{i_j}' = v_{i_j}' - v'$ for $T_i(f_i)$). The homogeneous part of T_i on f_i can be linearly approximated by

$$H_{T_i,f_i} = [u_{i_1}' u_{i_2}' \cdots u_{i_k}'] \cdot [u_{i_1} u_{i_2} \cdots u_{i_k}]^* \qquad (1)$$

where A^* denotes the pseudo inverse of matrix A. In fact, H_{T_i,f_i} is an approximation of the Jacobian of T_i on f_i. To extract the rigid component of H_{T_i,f_i} we perform its singular value decomposition: $H_{T_i,f_i} = P\Sigma Q^T$; the rigid component of H_{T_i,f_i} is then

C. Interactive Editing System

1) Laplacian Mesh Editing

Laplacian mesh editing method is proposed by Sorkine et al.[10] Let N_i be the index set of vertices adjacent to v_i. The Laplacian coordinate of a vertex v_i is

$$l_i = \sum_{j \in N_i} w_{ij}(v_j - v_i) \qquad (3)$$

where w_{ij} is the weight of the edge (i, j) corresponding to the vertex v_i. In matrix form, Equation (3) can be written as

$$l = LV \qquad (4)$$

where L is a $n \times n$ matrix.

The basic idea of Laplacian editing is to minimize the sum of the squared differences between the Laplacian coordinate before and after editing. The positions $\tilde{V}$ of the deformed mesh are found by minimizing $\left\| L\tilde{V} - l \right\|^2$, constrained by the positions of some selected vertices as the handles $\tilde{v}_i = c_i$. This is equivalent to solving a sparse linear system

$$A_1\tilde{V} = b_1 \qquad (5)$$

in the least squares sense. The deformed surface can be obtained by solving this linear system.

2) Feature Preserving Mesh Editing

The user provides a feature mask that marks the parts of the mesh whose shape should be preserved. We denote the mask by $M = \{m_1, \cdots, m_n\}$, such that $m_i = 1$ if vertex v_i belongs to a feature F_t and $m_i = 0$ otherwise. We denote the feature regions of the mesh by $F = F_1 \cup F_2 \cup \cdots \cup F_d$, where F_t is a connected component (Fig. 2b).

If face f_i belongs to a feature F (it has at least one node in F), we define the following k equations for f_i:

$$\tilde{v}_{i_{j+1}} - \tilde{v}_{i_j} = R_{T_i,f_i}(v_{i_{j+1}}) - R_{T_i,f_i}(v_{i_j}) \qquad (6)$$

where $\tilde{v}_{i_j}$ are the unknown deformed grid nodes, $j = 1, \cdots, k$ cyclically. Equation (6) can be written as its matrix form:

$$A_2 \tilde{V} = b_2 \quad (7)$$

Finally, we combine Equation (5) and Equation (7) into a new overdetermined linear system:

$$A\tilde{V} = b \quad (8)$$

This least square problem can be solved by solving the following equation:

$$A^T A\tilde{V} = A^T b \quad (9)$$

In order to edit models in real time, the system applies Cholesky factorization in advance.

The result of feature preserving mesh editing is illustrated in Fig. 1c. During our experiments, we found the resulting meshes are not smooth occasionally. To solve this problem, we propose the following optimizing algorithm.

III. Optimizing Algorithm

We found that in feature preserving editing step, the rotation transformation $R_{T_i \cdot f_i}$ of each face is extracted independently. Two neighbor faces take two different rotation transformations during the deformation, thus the unsmooth results are unavoidable.

To avoid this, the rotations of faces in the same feature region are integrated. This integrated transformation is obtained by solving the following system:

$$\min \sum_{i=1}^{m} \left\| R^t - R_{T_i \cdot f_i} \right\| \quad (10)$$

where f_i is a face in feature region F_t, R^t is the rotation matrix of feature region F_t, $t \in \{1, 2, \cdots, d\}$. We rewrite Equation (6) as:

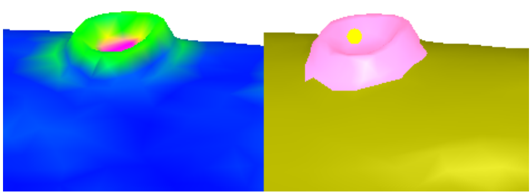
$$\tilde{v}_{i_{j+1}} - \tilde{v}_{i_j} = R^t(v_{i_{j+1}}) - R^t(v_{i_j}) \quad (11)$$

The final resulting mesh can be achieved by replacing Equation (6) with Equation (11) (Fig. 1d).

IV. Implemental Details

This paper provides two feature detecting methods. In the first one, users simply sketch two seeds in and outside the feature region respectively. The feature region is computed automatically[18].

In the second method, there are two substeps: feature detection and feature specification. Feature detection algorithm automatically computes the mesh saliency[18] value for each vertex. In Fig. 2a, the distinguishing feature part is painted in red, yellow and green, the other part is in blue.



a) b)

Figure 2. a) Feature detection. b) Feature specifying.

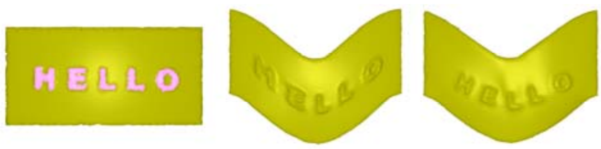
Feature specification is a user interactive step. Users specify one point that lies in the feature region, and the system computes the feature region by propagating from this point. This step increases the feature region iteratively, and stops until the mesh saliency value reach our threshold. In Fig. 2b, the pink region is the feature region, the yellow point is the point specified by users.

V. Experimental Results

We implement our algorithm on a computer with 3.0 GHz Pentium 4 CPU, with 1.0 GB of RAM and Windows OS. The result shows that our method produces visually pleasing editing results with simple user interaction.

Fig. 1 shows comparison between dual Laplacian editing[17] and our method. Fig. 1a is the input model, the feature region is specified by users and painted in pink. We fix the left side of the model and drag the right side upward. Fig. 1b is the result of dual Laplacian editing[17]. The pink region of their result is distorted. Fig. 1d is our final result. The feature region rotates adaptively.

Fig. 3 shows examples of letter wall. Fig. 3b and 3c are the result of Reference [17] and our result respectively. In Fig. 3b, the shapes of letters on the wall are distorted. Our method can preserve letters from distortion. Fig. 4 shows the antenna model of octopus. As illustrated, the results of the suckers with and without feature preserving process are completely different. As shown in Fig. 5, our method (Fig. 5c) can produce better results using feature preserving algorithm to constrain the behavior of the five organs of the head model.



a) b) c)

Figure 3. The letter-wall model.

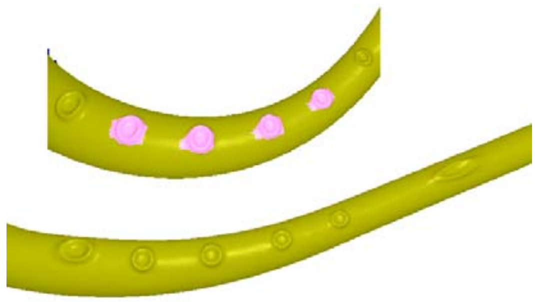


Figure 4. The antenna model.

VI. Experimental Results

In this paper, we show a feature preserving approach for 3D mesh editing. The features specified by users are preserved in resulting mesh, and it is benefited from the extraction of optimized rigid transformations. The result shows that our method produces visually pleasing editing results and is credible.

a) b) c)

Figure 5. The head model. a) The input model. b) The result of Ref. [17]. c) Our result.

Since the optimized rotation transformation is automatically computed, this may not satisfy the user. We'd like to develop an user interactive tool to allow the user specifying the rotation directly in our future work.

ACKNOWLEDGMENT

We would like to express my deepest gratitude to professor Ligang Liu for his helpful suggestions. This research received support from the National Natural Science Foundation of China (Grant No. 61100105), the Natural Science Foundation of the Anhui Higher Education Institutions of China (Grant No. KJ2010B212) and the Huangshan University Foundation, China (2011xkjq003).

REFERENCES

[1] Sederberg T. W. and Parry S. R.. "Free-form deformation of solid geometric models", In Proceedings of the 13th annual conference on Computer graphics and interactive techniques, pp.151-160, 1986.

[2] Coquillart S.. "Extended free-form deformation: A sculpturing tool for 3d geometric modeling", In Proceedings of the 17th annual conference on Computer graphics and interactive techniques, pp.187-196, 1990.

[3] R. MacCracken and K. I. Joy.. "Free-form deformations with lattices of arbitrary topology", In Proceedings of the SIGGRAPH, pp.181-188, 1996.

[4] Forsey D. and Bartels R.. Hierarchical B-Spline refinement. In Proceedings of the SIGGRAPH, pp.205-212, 1988.

[5] Zorin D., Schroder P. and Sweldens W.. "Interactive multiresolution mesh editing", In Proceedings of the SIGGRAPH, pp.259-268, 1996.

[6] Guskov I., Sweldens W. and Schröder P.. "Multiresolution signal processing for meshes", In Proceedings of the SIGGRAPH, pp.325-334, 1999.

[7] Kobbelt L., Campagna S., Vorsatz J. and Seidel H.-P.. "Interactive multi-resolution modeling on arbitrary meshes", In Proceedings of the SIGGRAPH, pp.105-114, 1998.

[8] Botsch M. and Kobbelt L.. "An intuitive framework for real-time freeform modeling", ACM Transactions on Graphics, vol.23, no.3, pp.630-634. 2004.

[9] Lipman Y., Sorkine O., Cohen-Or D., Levin D., Rössl C. and Seidel H.-P.. "Differential coordinates for interactive mesh editing", In Proceedings of the International Conference on Shape Modeling and Applications, pp.181-190, 2004.

[10] Sorkine O., Lipman Y., Cohen-Or D., Alexa M., Rössl C. and Seidel H.-P.. "Laplacian surface editing", In Proceedings of the Symposium on Geometry Processing, pp.179-188, 2004..

[11] Yu Y., Zhou K., Xu D., Shi X., Bao H., Guo B. and Shum H.-Y.. "Mesh editing with poisson-based gradient field manipulation", ACM Transactions on Graphics, vol.23, no.3, pp.644-651, 2004.

[12] Lipman Y., Sorkine O., Levin D. and Cohen-Or D.. "Linear rotation-invariant coordinates for meshes", In Proceedings of the SIGGRAPH, pp.479-487, 2005.

[13] Zhou K., Huang J., Snyder J., Liu X., Bao H., Guo B. and Shum H.-Y.. "Large mesh deformation using the volumetric graph Laplacian", ACM Transactions on Graphics, vol.24, no.3, pp.496-503, 2005.

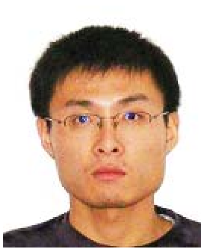
[14] Yoshioka, Y., Masuda, H. and Furukawa, Y.. "A constrained least-squares approach to interactive mesh deformation", In Proceedings of the international conference on Shape Modeling and Applications, pp.153-162, 2006.

[15] Masuda, H., Yoshioka, Y. and Furukawa, Y.. "Preserving form-features in interactive mesh deformation", In Proceedings of the Geometric Modeling and Processing, pp.207-220, 2006.

[16] Zheng Y., Fu H., Cohen-Or D., Au O., and Tai C.. "Component-wise controllers for structure-preserving shape manipulation", Computer Graphics Forum, vol.30, no.2, pp.563-572, 2011.

[17] Au O., Tai C., Liu L. and Fu H.. "Dual Laplacian editing for meshes", IEEE Transactions on Visualization and Computer Graphics, vol.12, no.3, pp.386–395, 2006.

[18] Lee C. H., Varshney A. and Jacobs D. W., "Mesh saliency", ACM Transactions on Graphics, vol.24, no.3, pp.659-666, 2005.



Jianwei Hu is an associate professor at Huangshan University, China. He received a PhD degree in Mathematics from Zhejiang University in 2009. His research interests include digital geometric processing, computer aided design and computer aided geometric design.

Gu Song is a lecturer at Huangshan University, China. His research interests include computer aided design and computer graphics.

Juan Cao is an assistant professor at School of Mathematical Sciences, Xiamen University, China. She received her B.S. degree in Information and Computing Science from Department of Mathematics, Fuzhou University, in June 2004. She received her Ph.D. degree in Applied Mathematics from Zhejiang University, in June 2009. During Oct. 2007 and Oct. 2008, she was a visiting scholar at Department of Computer Science, The State University of New York at Stony Brook (Stony Brook University). Her research interests include computer aided geometric design, digital geometry processing for computer graphics.