Vol. 10 No. 10 PROGRESS IN NATURAL SCIENCE October 2000

Noise removal algorithm for polygonal meshes *

LIU Xinguo (刘新国), BAO Hujun (鲍虎军) and PENG Qunsheng (彭群生)

(State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou 310027, China)

Received January 31, 2000; revised February 25, 2000

Abstract An effective noise removal algorithm which does not cause the shrinkage of surface is presented. By introducing the constraints of keeping all triangle barycenters invariant at each smoothing step, the algorithm turns the problem of mesh fairing into a constrained minimization of the energy functional, which is then solved efficiently by our iterative method. Experimental results demonstrate that our algorithm not only can preserve the good shape of the original surface while quickly removing the noise, but also has the advantages of fast convergence, low computational cost and stable performance.

Keywords: polygonal mesh, energy functional, Laplacian operator, shrinkage and distortion.

Polygonal meshes have been widely employed in computer aided design and computer graphics due to their simplicity and powerful ability to model complex objects. We can sample an object surface densely to capture its shape detail by 3-dimension (3D) scanning technology, and construct the triangle mesh model of object surface using these sample points[1, 2]. However, the reconstructed models contain frequently a lot of noises, and the pre-smoothing before being adopted by other applications is necessary.

The most popular way of fairing a surface is to minimize an energy functional[3, 4] defined over the surface. The extreme value of the following curvature energy is generally used for fairing a parametric surface $S: X = X(u, v)$:

$$E(S) = \int_S (k_1^2 + k_2^2) \mathrm{d}S,$$

where k_1 and k_2 are the principlal curvatures of surface S. Since evaluating the principle curvatures of the surface based on its sample points is very difficult, in practice one usually adopts the membrane energy and thin-plate energy[5, 6]:

$$E_{\mathrm{memb}}(S) = \frac{1}{2}\int_S (X_u^2 + X_v^2) \mathrm{d}u\mathrm{d}v,$$

$$E_{\mathrm{thin}}(S) = \frac{1}{2}\int_S (X_{uu}^2 + 2X_{uv}^2 + X_{vv}^2) \mathrm{d}u\mathrm{d}v,$$

* Project supported by the National Science Fund for Distinguished Young Scholars (Grant No. 69925204) and the National Natural Science Foundation of China (Grant No. 69673027).

where X_u, X_v and X_{uu}, X_{vv}, X_{uv} are the first and the second derivatives of the surface respectively. Nevertheless, to minimize the above simplified energy functional is still complicated and time-consuming, and the results heavily depend on the parameterization of the surface.

In order to smooth a mesh with a large data set, Taubin[5] generalized the classical Fourier analysis to discrete 2-manifold meshes which can be regarded as a low-pass filtering of surface signal by linearly approximating with Laplacian operator. Under a special parameterization, Kobbelt et al.[6] derived a similar Laplacian operator for subdividing model. Rather than solving the large sparse system equation directly, both the algorithms employed a local iterative technique to fair a mesh with a large data set. The approximation realized by the Laplacian operator probably results in a serious shrinkage and distortion of meshes. T these two approaches are unacceptable for modeling.

To reduce the shrinkage and distortion, Taubin[5] suggested a linear combination of the first and the second Laplacian operators to fair polygonal meshes, but it heavity depends on the choice of the two mesh-related constants. Recently, Desbrun et al.[7] and Vollmer et al.[8] presented two experimental methods to reduce the shrinkage and distortion. In this paper, we will introduce a novel noise removal algorithm to solve the above problems in a different way. The key of this algorithm is to keep all triangle barycenters invariant at each smoothing step by introducing relaxation constraints in to the minimized energy functional. These relaxation constraints can not only prevent the generated mesh from shrinkage, but also provide the vertices of mesh with enough degree of freedom.

1 Laplacian operator and energy functional

Without losing generality, we will discuss the triangular meshes here, which can easily be generalized to arbitrary polygonal meshes. A triangular mesh M is denoted by a triple $\langle V, T, X\rangle$, where $V=\{1,2,3,\cdots,N\}$ is its vertex set, and $T\subseteq 2^V$ its triangle set composed of all elementary triangles. An arbitrary triangle $t\in T$ is denoted by an ordered vertex triple $t=\langle i, j, k\rangle$. The geometric realization of the mesh $X: V\to R^3$ is a mapping from the vertices to their location in 3D space. Let Adj $(i)\subseteq V$ be the set of the 1-ring vertices neighboring vertex i, and $|\mathrm{Adj}(i)|$ be the valence of vertex i (i.e. the number of its 1-ring neighbors). For short hand, we use x_i to denote the position of vertex i, so the geometry of M can be expressed by a vector $\boldsymbol{X}$ that consists of all its vertices, namely, $\boldsymbol{X}=[x_1, x_2\cdots x_N]^{\mathrm{T}}$.

The variational derivatives of the above-mentioned energy functionals with respect to the Laplacian and the second Laplacian operators respectively are

$$L(\boldsymbol{X}) = \boldsymbol{X}_{uu} + \boldsymbol{X}_{vv}, \quad L^2(\boldsymbol{X}) = L\circ L(\boldsymbol{X}) = \boldsymbol{X}_{uuuu} + 2\boldsymbol{X}_{uuvv} + \boldsymbol{X}_{vvvv}.$$

The Laplacian operator of the curve surface composed by polygonal meshes can be expressed as the following discrete form:

$$L(x_i) = \sum_{j\in \mathrm{Adj}(i)} \omega_{ij}(x_j - x_i). \tag{1}$$

Obviously, the discrete Laplacian operator is a linear combination of the neighboring vertices of a vertex. Writing eq.(1) in a matrix form, we have $L(\boldsymbol{X}) = \boldsymbol{K}\boldsymbol{X}$, where $\boldsymbol{K}=(\omega_{ij})_{N\times N}$. In eq. (1),

ω_{ij} are the non-negative weights satisfying $\sum_{j \in \mathrm{Adj}(i)} \omega_{ij} = 1$, and $\omega_{ij} = 0$ when $j \notin \mathrm{Adj}(i)$. Therefore, $\boldsymbol{K}$ is a sparse matrix.

Under a special parameterization, the energy functional of a mesh M can be approximated by

$$E(M) = \| L(\boldsymbol{X}) \|^2 = \sum_{i \in V} | L(x_i) |^2 = \boldsymbol{X}^{\mathrm{T}}(\boldsymbol{K}^{\mathrm{T}}\boldsymbol{K})\boldsymbol{X}. \tag{2}$$

2 Barycenter constrained fairing algorithm

As we know, to minimize directly the discrete energy functional of eq. (2) may result in serious shrinkage and distortion. To avoid this, the barycenter constraints are introduced into the discrete energy functional.

Let $\boldsymbol{X}^0$ be the initial vertex positions of triangle mesh M, the barycenter constraint of triangle $t = \langle i, j, k \rangle \in T$ can then be expressed as

$$(x_i + x_j + x_k) - (x_i^0 + x_j^0 + x_k^0) = 0.$$

Thus, the constraints on the whole mesh can be written in the following matrix form:

$$\boldsymbol{H}(\boldsymbol{X} - \boldsymbol{X}^0) = 0,$$

in which each triangle $t = \langle i, j, k \rangle \in T$ corresponds to a row in the $N \times N$ matrix $\boldsymbol{H}$ with elements $h_{ts} = \begin{cases} 1 & s = i, j \text{ or } k \\ 0 & \text{otherwise} \end{cases}$. The fairing for mesh M then can be obtained by solving extreme value

$$\begin{cases} \min & \boldsymbol{X}^{\mathrm{T}}(\boldsymbol{K}^{\mathrm{T}}\boldsymbol{K})\boldsymbol{X}, \\ \text{subject to} & \boldsymbol{H}(\boldsymbol{X} - \boldsymbol{X}^0) = 0. \end{cases} \tag{3}$$

This is a typical constrained optimization problem, which can be transferred to an unconstrained one using the well-known penalty approach:

$$\min \quad \boldsymbol{X}^{\mathrm{T}}(\boldsymbol{K}^{\mathrm{T}}\boldsymbol{K})\boldsymbol{X} + \mu \cdot (\boldsymbol{X} - \boldsymbol{X}^0)^{\mathrm{T}}(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})(\boldsymbol{X} - \boldsymbol{X}^0),$$

where μ is a positive constant to reach a compromise between the minimization and the constraints. Nevertheless, solving a large sparse system using this method is usually time-and memory-consuming while the local iterative solution proposed by us can effectively solve this problem.

2.1 Local iterative solution

The local iterative solution is to minimize the local energy in the neighborhood of triangle t under the barycenter invariant constraint. Consider a triangle t and its neighboring triangles shown in fig. 1, and let p_0, p_1, and p_2 be the three vertices of t. Then the discrete energy in the neighborhood of t can be defined as

$$E(t) = \sum_{i=0}^{2} L(p_i) = \| AP - Q \|^2 = (P^T A^T - Q^T) \cdot (AP - Q),$$

where $A = \begin{bmatrix} 1 & -\omega_{01} & -\omega_{02} \\ -\omega_{10} & 1 & -\omega_{12} \\ -\omega_{20} & -\omega_{21} & 1 \end{bmatrix}$, $P = \begin{bmatrix} p_0 \\ p_1 \\ p_2 \end{bmatrix}$, $Q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \end{bmatrix}$, $q_i = \sum_{j=0}^{m_i} \omega_{ij} q_{i,j}, (i = 0,1,2)$.

The barycenter constraint of a triangle can be rewritten as

$$BP = C,$$

where $B = [1 \quad 1 \quad 1]$, $C = BP^0$ in which $P^0 = [p_0^0 \quad p_1^0 \quad p_2^0]^T$ represents the original position of the triangle. Consequently, the local relaxation in the neighborhood of the triangle t can be defined as the minimum of $E(t)$ under the barycenter constraint:

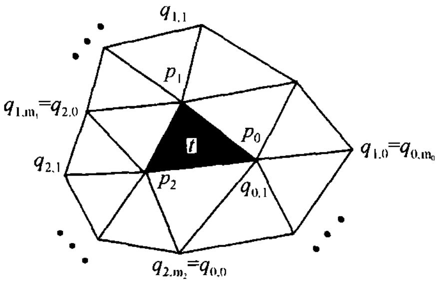

Fig. 1. Triangle t and its neighboring triangles.

$$\begin{cases} \min & (P^T A^T - T^Q) \cdot (AP - Q), \\ \text{subject to} & BP = C. \end{cases} \tag{4}$$

Since the functional is quadratic, it can be solved using the Lagrangian multiplier method of

$$P = A^{-1}Q + \frac{1}{\| BA^{-1} \|^2}(C - BA^{-1}Q) \cdot A^{-1}(A^{-1})^T B^T. \tag{5}$$

There are two popular ways to choose weights ω_{ij}. One only depends on the local connectivity of meshes, the other considers both the geometry and connectivity of meshes. Here, we use the first method to define the weights:

$$\omega_{ij} = \begin{cases} \frac{1}{n_i}, & j \in \mathrm{Adj}(i), \\ 0, & \text{otherwise}. \end{cases}$$

Therefore, A^{-1}, BA^{-1}, and $BA^{-1}(A^{-1})^T B^T$ in eq.(5) depend on only the valence of vertex i. In most cases, the valence of a vertex is around 5, 6 or 7. Instead of computing these matrices in iterative solution, we build look-up tables of the matrices for each valence number in a preprocessing step. Thus, we can then efficiently calculate the new locations of three vertices of triangle t by looking up the tables.

The above mentioned method simultaneously relaxes three vertices of a triangle to minimize the local energy each time. It is notable that there are several triangles surrounding a vertex p, which implies that one may obtain several new locations p_{new}^1, p_{new}^2, $\cdots$, p_{new}^m, after relaxing vertex p (m is its valence). The final new location of vertex p can be determined by averaging these locations:

$$\bar{p}_{\text{new}} = \frac{1}{m}\sum_{i=1}^{m} p_{\text{new}}^{i}$$

2.2 Boundary fairing

For an open mesh, we need also smoothing its boundary curves. Here, we extract the boundary curves of the mesh and smooth them by a separate process. Similarly, the Laplacian operator at vertex p_i for a discrete curve with m vertices $p_0, p_1, \cdots, p_m$ is defined as

$$L(p_i) = \omega_{i,-1}(p_{i-1} - p_i) + \omega_{i,1}(p_{i+1} - p_i).$$

where $\omega_{i,-1}$ and $\omega_{i,1} > 0$, and $\omega_{i,-1} + \omega_{i,1} = 1$. For simplicity, we take $\omega_{i,-1} = \omega_{i,1} = \frac{1}{2}$. Adhering to the idea mentioned in previous sections, the barycenter/midpoint constraints may also be introduced to prevent the curves from shrinkage and distortion. It is easy to deduce the local iterative equation

$$\begin{bmatrix} p_{i-1} \\ p_i \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{2}{3} \end{bmatrix} \begin{bmatrix} p_{i-2} \\ p_{i+1} \end{bmatrix} + \begin{bmatrix} d \\ d \end{bmatrix},$$

where $d = [(p_{i-1}^0 + p_i^0) - (p_{i-2} + p_{i+1})]/2$.

2.3 Test results

Our algorithm was tested on many models, and one of the results is shown in fig. 2. In the test, some stochastic white noises were added to the original horse model, then our smooth method was used to remove these noises. Only one iterative step was implemented to this example. Comparing the resultant model with the original one, it can be found that they are almost the same! Our experiments show that the algorithm not only can effectively remove surface noises while preserving the shape of the original model, but also has nice features of fast convergence, low computational cost and stable performance.

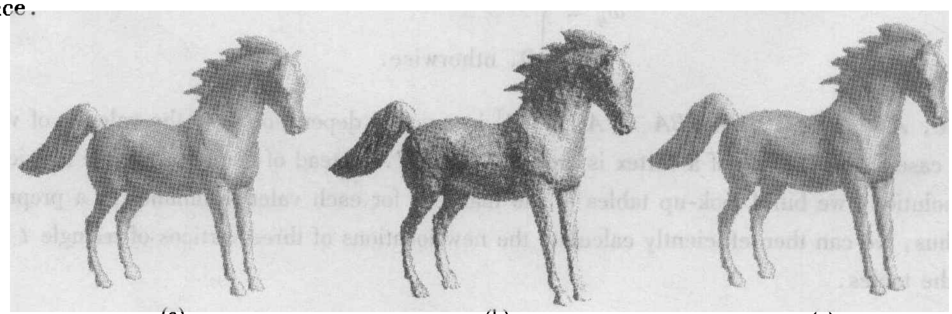
(a) (b) (c)

Fig. 2. Test result of the algorithm. (a) Original model; (b) model with white noises; (c) model after being faired using our algorithm.

References

1 Hoppe, H., DeRose, T., Duchamp T. et al., Surface reconstruction from unorganized points, in *Proceedings of SIGGRAPH' 92 Conference*, USA, Danvers: Addison-Wesley, 1992, 71.

2 Curless, B., Levoy, M., A volumetric method for building complex models from range images, in *Proceedings of SIGGRAPH' 96 Conference*, USA, Danvers: Addison-Wesley, 1996, 303.

3 Morton, H. P., Sequin, C. H., Functional optimization for fair surface design, in *Proceedings of SIGGRAPH' 92 Conference*, USA, Danvers: Addison-Wesley, 1992, 167.

4 Welch, W., Witkin, A., Varational surface modeling, in *Proceedings of SIGGRAPH' 92 Conference*, USA, Danvers: Addison-Wesley, 1992, 157.

5 Taubin, G., A signal processing approach to fair surface design, in *Proceedings of SIGGRAPH' 95 Conference*, USA, Danvers: Addison-Wesley, 1995, 351.

6 Kobbelt, L., Campagna, C., Vorsatz, J. et al., Interactive multi-resolution modeling on arbitrary meshes, in *Proceedings of SIGGRAPH' 98 Conference*, USA, Danvers: Addison-Wesley, 1998, 105.

7 Desbrun, M, Meyer, M, Schroder, P. et al., Implicit fairing of irregular meshes using diffusion and curvature flow, in *Proceedings of SIGGRAPH' 99 Conference*, USA, Danvers: Addison-Wesley, 1999, 317.

8 Vollmer, J., Mencl, R., Müller, H., Improved Laplace smoothing of noisy surface meshes. in *Proceedings of EUROGRAPHICS '99 Conference*, UK, London: Blackwell Publishers, 1999, 211.