

多边形网格的保形除噪声算法

刘新国 鲍虎军 彭群生

浙江大学 CAD&CG 国家重点实验室

摘要 本文给出了一个高效的无收缩磨光算法，用于消除多边形网格表示的几何模型表面上的粗糙的噪声，生成光滑的几何模型。我们首先将磨光问题转化为一个能量泛函的极小化问题。但是直接迭代求解该极值问题将产生原多边形网格的极度收缩，从而导致磨光的失败。因此，我们又引入面片重心不变的约束条件，将问题转化为条件极值问题，并给出了一个新的高效迭代求解方法。实验结果表明，本算法不仅能快速地除掉表面上粗糙的噪声，同时保持其形状特征，而且具有运算量少，计算稳定，收敛快等诸多好性质。

关键词： 多边形网格，能量泛函，Laplacian 算子，曲面光顺

一、引言

多边形网格表示几何模型在计算机图形学各个领域得到越来越广泛的应用，这不仅仅因为它简单、实用，能够表示任意拓扑、形状的几何，适合图形硬件处理，还因为人们能够通过很多各种各样的手段获得它。过去大多使用造型软件进行人工搭建，现在人们更喜欢通过对实物模型测量的方法，采用多边形网格（尤其是三角片网格），重建它的几何[4, 7, 11]。因为实物测量重建的方法不仅简单、快捷，而且能够生成非常逼真、精细的模型。然而，实物测量技术不可避免地带来测量噪声和误差，所以重建后的多边形网格模型的表面总是有凹凸不平的细小噪声，这需要我们有一套平滑算法，对重建几何的表面进行磨光，去除那些噪声。

在许多的曲面光顺的文献中[12, 13]，最常用的工具是带约束的能量最小化方法。一般地，对参数化的曲面进行磨光的方法都是考虑一种定义表面上的能量，然后通过求解一组能量最小化方程组以达到磨光的目的。设有一被磨光曲面 $S = X(u, v)$ ，最基本的一种能量泛函为曲率能量，即两个主曲率的平方和在表面上的积分：

$$E(S) = \int_S (k_1^2 + k_2^2) dS$$

由微分几何知识我们知道，一般形式曲面主曲率的求取和估计是非常困难的，因此在实际的曲面光顺算法中，人们通常采用薄膜(membrane)能量：

$$E_{mem}(S) = \frac{1}{2} \int_S (X_u^2 + X_v^2) dudv$$

和薄板(thin-plate)能量：

$$E_{thin}(S) = \frac{1}{2} \int_S (X_{uu}^2 + 2X_{uv}^2 + X_{vv}^2) dudv$$

然而，即使作上述简化，求解能量最小化问题还是一个非常复杂、费时的计算过程，而且结果和所使用的参数有很大关系。

为了能够磨光大型的任意多边形网格数据，1995 年，Taubin[8]把经典的傅立叶分析推广到离散的二维曲面上去，将二维曲面解释为定义在多边形网格上的函数（或信号），通过一种对 Laplacian 算子的线性逼近表示，从而将离散曲面的磨光整形转化为曲面信号的低通

滤波问题。其实早在 1994 年, Willian 等人[10]就使用了一种离散的 Laplacian 算子进行基于三角形网格的光顺自由曲面造型。另一个与此相关的工作是 Kobelt 方法[6], 他在一种特殊的局部参数化条件下推导出与 Taubin 一样的离散 Laplacian 算子, 用于光顺细分插值曲面造型。

各种形式的离散 Laplacian 算子的成功之处在于它们能够光顺大型的多边形网格, 因为它们不要求解复杂的优化方程组, 而只是对网格不断地施以 Laplacian 算子进行磨光迭代, 在时间和空间上仅具有线性复杂度。但是, 正是这种对 Laplacian 算子简单的逼近, 使得 Laplacian 算子在处理一些不规则的网格的磨光过程中, 会引起几何收缩和变形等不好的结果。对于噪声处理来说, 这些收缩和变形是不可接受的。

为了防止磨光过程的这种收缩和变形现象, Taubin[8]利用一阶、二阶 Laplacian 算子的线性组合方法来减少收缩和变形的幅度。然而如何选择组合系数并没有统一的准则, 非常依赖于具体被磨光模型本身的性质。Mathieu 等人[1]通过保持模型的体积不变的方法来防止收缩现象, 即在每一步磨光迭代之后, 对所有的顶点位置作一个统一的伸缩变换, 使得变换后的模型与原始模型具有相同的体积。Vollmer 等人[2]则在磨光过程中, 考虑顶点的原始位置, 将每一次磨光迭代得到的新点位置朝着原始位置方向拉回一些, 以减少收缩和变形。这些算法均是经验的, 实际上它们只减少了原始网格的收缩程度。

本文将从另一个完全不同的角度来处理磨光过程中的收缩和变形问题, 提出了一个无收缩的去噪声算法。该算法的基本思想是在磨光的过程中尽量保持面片的重心不动。这种松弛的约束既保证了模型不会变形收缩, 又允许顶点位置有活动的自由度, 参与磨光过程。

下面我们首先介绍一些基本概念、有关记号和传统的 Laplacian 磨光算法, 然后从能量泛函和约束条件入手推导出我们的磨光算法。最后我们给出算法的一些执行结果和结论。

二、基本概念和记号

三角形网格是最为常见的一种多边形网格模型, 为了方便, 在这篇文章中我们将仅仅讨论三角形网格, 当然本文中的算法并不局限于此, 它适用于一般的多边形网格。

一个三角形网格 M 可以表示为一个三元组 $\langle V, T, X \rangle$ 其中 $V = \{1, 2, 3, \dots, N\}$ 为网格 M 的顶点集; $T \subseteq 2^V$ 是网格 M 上的面片, 由于这里 M 是三角形网格, 所以对任何 $t \in T$, 它是一个三角形, 可表示为顶点元素的有序三元组 $t = \langle i, j, k \rangle$; $X: V \rightarrow R^3$ 是一个映射, 将每一个顶点映射到它在空间中所处的位置上。

为了叙述方便, 再引入一个记号 $Adj(i)$, $Adj(i) \subseteq V$ 表示与顶点 i 相邻的顶点的集合, 相应地, $|Adj(i)|$ 表示顶点 i 相邻的顶点的个数。在不引起混淆的情况下, 我们用 x_i 代表 $X(i)$, 即顶点 i 的空间位置; 还用 X 表示顶点坐标组成的向量, 即 $X = [x_1 \ x_2 \ \dots \ x_N]^T$ 。

三、Laplacian 磨光算法

前面提到的薄膜能量和薄板能量的变分形式分别对应于一阶和二阶 Laplacian 算子:

$$L(X) = X_{uu} + X_{vv}$$

$$L^2(X) = L \circ L(X) = X_{uuuu} + 2X_{uuvv} + X_{vvvv}$$

对于离散的多边形网格曲面, Laplacian 算子可以离散地表示为:

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

容易看到离散的 Laplacian 算子是相邻顶点坐标的线性组合, 其中 ω_{ij} 是组合系数, 它们满

足:

$$\omega_{ij} \geq 0 \text{ 并且 } \sum_{j \in Adj(i)} \omega_{ij} = 1$$

写成矩阵形式, 有:

$$L(X) = KX \tag{2}$$

其中 $K = (\omega_{ij})_{N \times N}$ 。注意 $w_{ii} = 1$, 且当 $j \notin Adj(i)$ 时, $\omega_{ij} = 0$, 因此 K 是一个稀疏矩阵。

在一个网格上进行磨光过程可以看作成一个扩散过程[1]:

$$\frac{\partial X}{\partial t} = \lambda L(X)$$

通过在时间轴上的积分, 曲面上细小的起伏、噪声很快地扩散到它的邻域中, 使得整个曲面变得光滑起来。如果采用显式的 Euler 积分方法, 我们有:

$$X^{n+1} = (1 + \lambda dt \cdot L)X^n$$

于是可以得到一系列越来越光滑的网格 $M^n = \langle V, T, X^n \rangle$ 。

图 1 是 Laplacian 算子磨光的一个例子, 从左到右分别是原始网格和经过 1, 3, 10 次迭代的结果。从中我们可以看到网格非常明显收缩现象。其实, 如果迭代过程一直进行下去, 整个网格将会退化成为一个点。

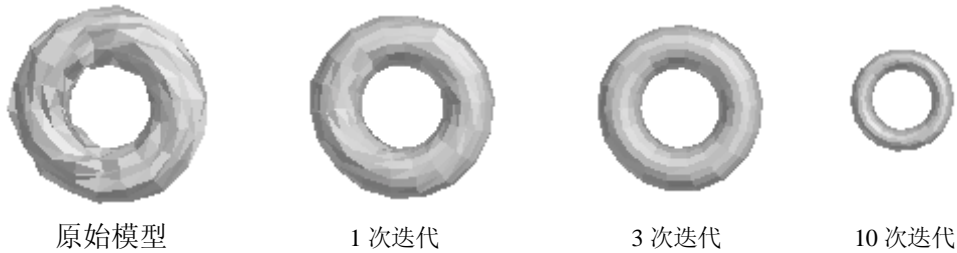


图 1 Laplacian 磨光的网格收缩现象

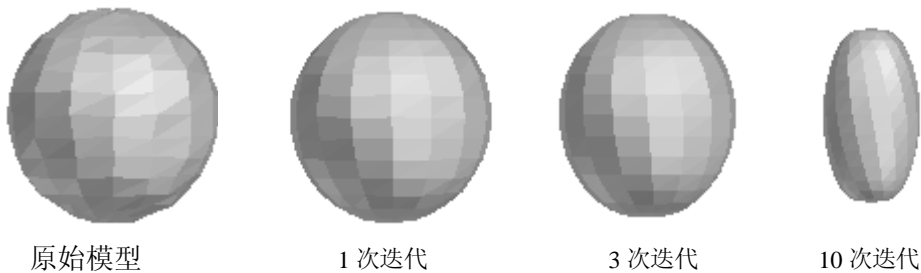


图 2 Laplacian 磨光的网格非均匀收缩现象

图 2 是 Laplacian 算子磨光的另一个例子, 从左到右分别是原始网格和经过 1, 3, 10 次迭代的结果。从中我们可以看到网格非常明显变形现象。这种变形是因为网格中面片大小不一造成的, 因为面片尺寸上的差别使得网格上不同部分的收缩速度有快有慢, 尺寸越大收缩越快, 从而导致网格的变形。

四、重心约束的磨光算法

下面我们从能量泛函极小化问题出发，通过引进约束条件，使得网格在磨光过程中不会收缩、退化，从而推导出一个新的磨光算法，称之为重心约束的磨光算法。多边形网格上的能量可以用下式离散地近似：

$$E(M) = \sum_{i \in V} |L(x_i)|^2 = \|L(X)\|^2 \quad (3)$$

将(1)式代入上式得：

$$E(M) = X^T (K^T K) X \quad (4)$$

为了防止磨光过程中收缩变形现象的发生，必须在一些约束条件下求解上面的离散能量最小值问题。对于去噪声算法，我们加入的约束是三角形的重心，即在三角形重心不动的条件，使得能量达到最小。

用 X^0 表示三角形网格 M 中顶点的初始位置。对 M 中一个三角形 $t = \langle i, j, k \rangle \in T$ ，重心约束可以写成：

$$\begin{aligned} x_i + x_j + x_k &= x_i^0 + x_j^0 + x_k^0 \\ (x_i + x_j + x_k) - (x_i^0 + x_j^0 + x_k^0) &= 0 \end{aligned}$$

则加在所有三角形上的约束构成一个线性方程组：

$$HX - HX^0 = 0$$

且每一个三角形 $t = \langle i, j, k \rangle \in T$ 对应于矩阵 H 的一个行向量 H_t ：

$$H_{ts} = \begin{cases} 1 & s = i, j, k, \\ 0 & \text{其他。} \end{cases}$$

因此，重心约束的磨光算法就是求解下面的条件极值问题：

$$\begin{cases} \min & X^T (K^T K) X \\ & HX - HX^0 = 0 \end{cases} \quad (5)$$

求解上述条件极值的基本方法有两种[3]：一种是大家熟悉的 Lagrange 乘子法，这种方法能够精确地满足约束条件，但是需要引入额外的未知变量，将上述条件极值问题转化下面的无条件极值问题；另一种是 Penalty 方法，这种方法虽然不能够精确地满足约束条件，但是它能在无需引入额外的未知变量的情况下，将上述条件极值问题（5）转化为下面的无条件极值问题：

$$\min \quad X^T (K^T K) X + \mu \cdot (X - X^0)^T (H^T H) (X - X^0)$$

其中 μ 是一个正的常数，通过 μ 的选择，在极小化和约束条件之间达成一种妥协。

然而，上面提到的两种方法都要求解一个大型的线性方程组，对于大数据量的多边形网格，无论是时间上的开销还是空间上的开销都是非常大的。为此我们设计了局部求解的算法。

4.1 局部求解

考察如图 3 所示的带阴影的内部三角形 t ，其三个顶点分别是 p_0, p_1, p_2 。所谓的局部求解方法就是在保证三角形 t 的重心不变的条件，将三角形 t 局部邻域内的能量最小化。

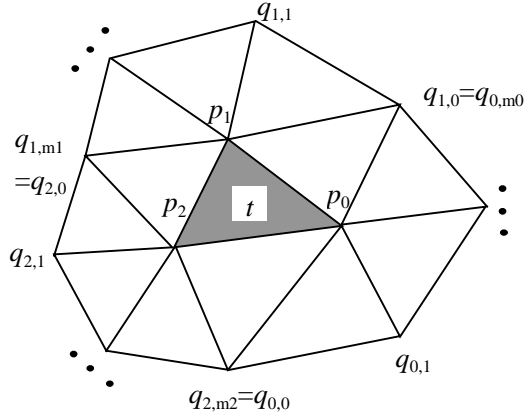


图3 三角形 t 及其邻域

取三角形 t 的局部邻域为 t 以及与 t 相邻的一圈三角形（如图3所示），则其局部能量为三角形 t 三个顶点 p_0 , p_1 , p_2 处的能量之和：

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

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

重心约束条件为：

$$BP = C$$

其中 $B = [1 \ 1 \ 1]$, $C = BP^0$, $P^0 = [p_0^0 \ p_1^0 \ p_2^0]^T$ 。

故局部条件极值问题就是：

$$\begin{cases} \min (P^T A^T - Q^T) \cdot (AP - Q) \\ BP = C \end{cases} \quad (6)$$

用 Lagrange 乘子法解得：

$$P = A^{-1}Q + \lambda \cdot A^{-1}(A^{-1})^T B^T \quad (7)$$

其中

$$\lambda = \frac{C - BA^{-1}Q}{BA^{-1}(A^{-1})^T B^T} = \frac{1}{\|BA^{-1}\|^2} (C - BA^{-1}Q) \quad (8)$$

式(1)中 Laplacian 算子的权因子 ω_{ij} 的选择方法有很多，大致可分为两种：一种方法是根据顶点周围的局部几何性质，如相邻边的长度等，选取权因子 ω_{ij} ，这就是非均匀的 Laplacian 算子。另一种方法是根据顶点的局部连接关系选取权因子 ω_{ij} ，即：

$$\omega_{ij} = \begin{cases} \frac{-1}{|Adj(i)|} & j \in Adj(i) \\ 0 & \text{否则} \end{cases} \quad (9)$$

这种方法对应于均匀 Laplacian 算子。本文采用第二种权因子，除了因为它简单、计算稳定外，另一主要原因是选择这样的权因子后，还可以进一步省去很多的计算，加快算法的速度。考察式(7)、(8)中的 A^{-1} 、 BA^{-1} 、 $BA^{-1}(A^{-1})^T B^T$ ，它们均是 ω_{ij} 的函数，而 ω_{ij} 是顶点入度的倒数的负数，所以它们是三角形三个顶点的入度的函数。因为在三角形网格中，绝大多数的顶点入度为 5、6 或 7 等有限几个值。因此，我们可以对 A^{-1} 、 BA^{-1} 、 $BA^{-1}(A^{-1})^T B^T$ 建立一个查找表，这样在局部求解的时候，在绝大多数情况下，都可以根据三角形的三个顶点的入度从表中查找出这些值，而不用直接实时计算。

图 4 是一个局部求解的例子，原本高高耸起的尖峰得到很好的平滑，而且突起的形状也没有塌陷下来。

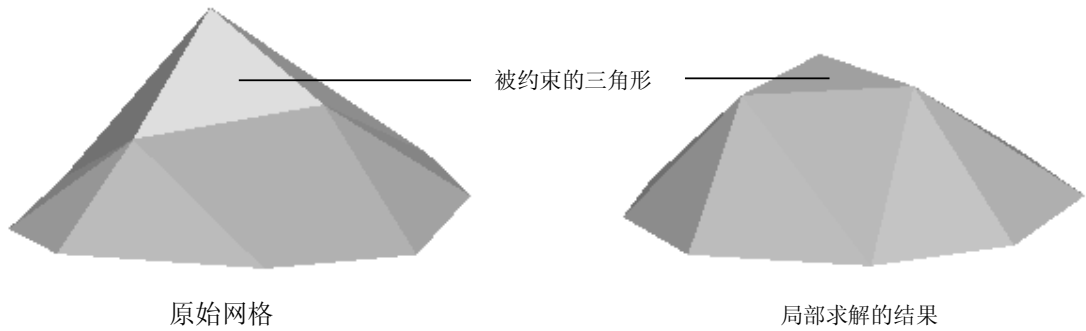


图 4 保持重心的局部能量最小化求解示例



图 5 对贝多芬雕塑模型的磨光迭代。注意原始雕塑上非常粗糙、带棱角头发，以及沧桑的面部都得到很好的光滑，并且保持了发型和面部表情。

经过上面的求解过程之后，我们同时得到三角形三个顶点的新位置。对于复杂的、一般的网格模型，一个顶点 p 有多个三角形 t_1, t_2, \dots, t_m 与其相连接，这意味着有多个新位置 $p_{new}^1, p_{new}^2, \dots, p_{new}^m$ 与其相对应。因此我们还要解决一个小小的问题，即如何根据这些新点位置决定该顶点的最终新位置 p_{new} 。一种简单的方法是取平均值，即：

$$p_{new} = \frac{1}{m} \sum_{i=1}^m p_{new}^i$$

根据以上推导，我们的磨光迭代过程可描述如下。

Pold 和 **Pnew** : 存放顶点迭代前旧位置和迭代后新位置的两个数组;

Cnew: 存放顶点新位置个数的数组;

将数组 **Pnew** 的所有元素清零;

将数组 **Cnew** 的所有元素清零;

对每一个内部三角形

用保持重心的局部能量最小化算法计算三个顶点 **I, j, k** 的

新位置坐标 **Newi, Newj** 和 **Newk**;

Pnew[i] = Pnew[i]+Newi;

Cnew[i] = Cnew[i]+1;

Pnew[j] = Pnew[j]+Newj;

Cnew[j] = Cnew[j]+1;

Pnew[k] = Pnew[k]+Newk;

Cnew[k] = Cnew[k]+1;

对所有的顶点 **v**

如果 **Cnew[v]>0** 那么

Pold[v] = Pnew[v]/Cnew[v];

返回。

我们用上述方法对图 5 中的贝多芬雕塑模型进行磨光迭代。从中我们可以看到原来雕塑模型上非常粗糙、带棱角的头发，以及沧桑的面部都得到很好的光顺，而且迭代后的雕塑模型很好地保持了原来的形状特征，没有收缩变形现象发生。在图 5 中我们只进行了 2 次磨光迭代。实验表明经 1、2 次磨光迭代后，模型就收敛到极限曲面了，对其他的实验模型也是一样。这说明我们对条件极值的局部求解方法收敛很快、十分有效。

4.2 边界处理

有的网格模型不是封闭的，因此我们还需要处理边界曲线的磨光问题。处理方法大致可分为两种[1]：一种方法是在边界曲线的外层构造一圈虚拟的三角形，将边界问题转化为内部点进行统一处理；另一种方法是将边界曲线单独拎出来，用曲线磨光的算法进行处理。本文采用第二种方法。

其实，对离散曲线的磨光也可以用上面的条件极值方法。对于一条曲线，如图 6 所示，顶点 i 处的 Laplacian 算子定义为：

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

其中 $\omega_{i,-1}, \omega_{i,1} > 0$ ，且 $\omega_{i,-1} + \omega_{i,1} = 1$ 。在本文中，我们取 $\omega_{i,-1} = \omega_{i,1} = \frac{1}{2}$ 。

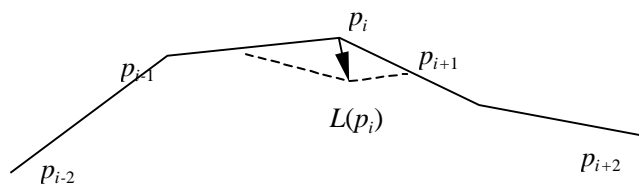


图6 一维 Laplacian 算子的构造

类似于曲面三角形重心条件，在曲线磨光时，我们亦可加上边的中心保持不变的条件。然后采用上一节的方法，得到曲线磨光的局部求解公式：

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

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

图7中维纳斯雕塑模型是一个带边界的三角形网格。注意这个模型中的边界部分的变化，我们可以看到，上面的边界处理办法能够得到满意的结果。

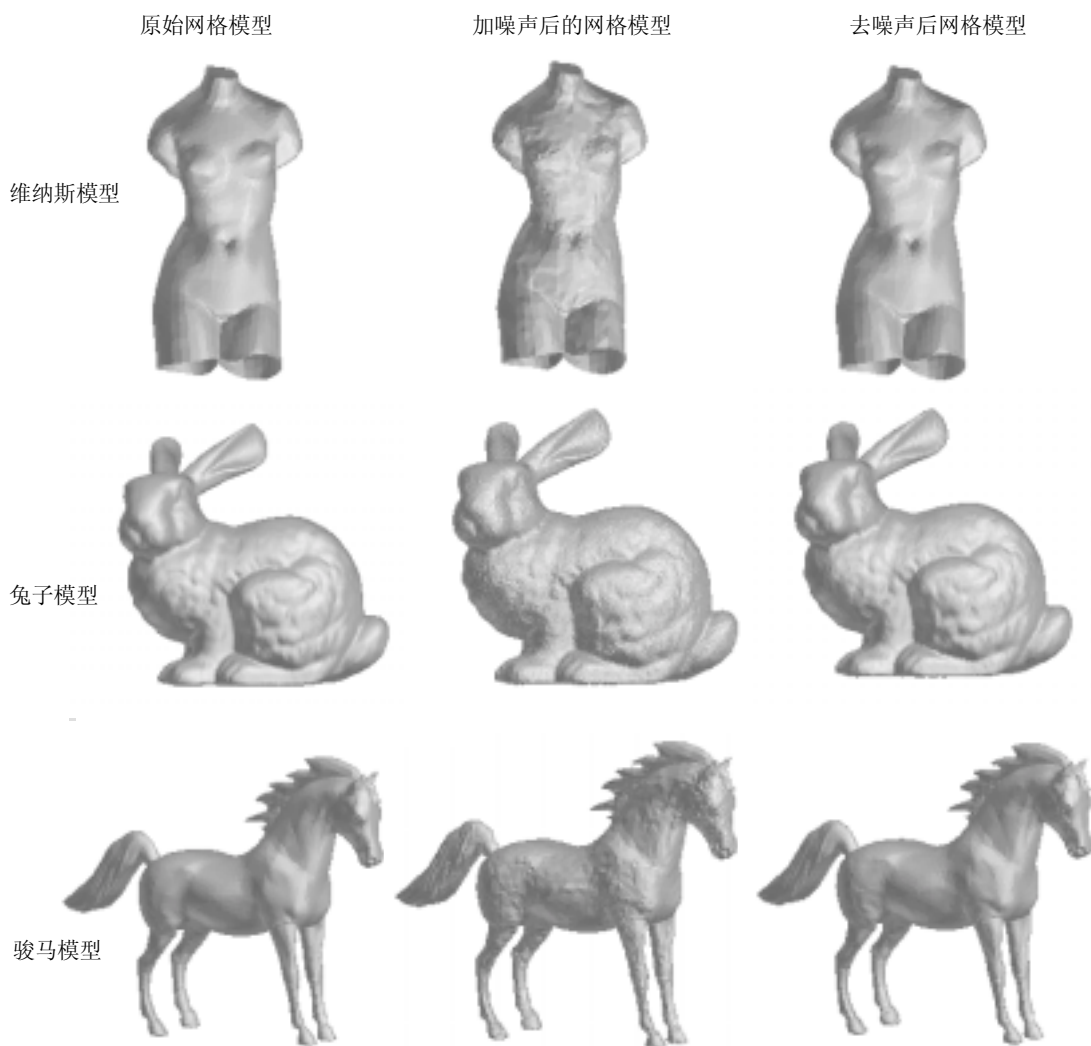


图7 对几个不同网格模型进行去噪声磨光处理的效果

五、结果与结论

我们用 C++ 在 Compaq NT 工作站上实现了本算法，用于多边形网格上的去噪声处理，参考图 7 中的实例。为清晰起见，所有的图例均采用 Flat 绘制模式进行显示，没有作任何光亮度光滑过度处理。对图 7 中的三个原始的网格模型，我们首先加入随机的白噪声，然后再运用我们的磨光算法进行去噪声处理。在去噪声时都只需要进行一次迭代。容易发现，去噪声后所恢复的结果和原模型几乎一致。这说明本算法具有非常良好的去噪声、磨光作用。

由于引入了重心约束条件，通过求解条件约束的极值问题，本算法有效保证了磨光过程的无收缩性。而局部求解迭代技术的运用，极大简化了复杂的优化求解过程，往往只需一至二次迭代后网格模型就收敛到极限，提高了算法的效率。今后的工作是将该算法同多分辨率模型有机地结合起来，设计对任意多边形网格的滤波算法以及基于多边形网格的光顺曲面造型工具。

参考文献

- [1] Mathieu Desbrun, Mark Meyer, Peter Schrör and Alan H. Barr. Implicit Fairing of Irregular Meshes using Diffusion and Curvature Flow. SIGGRAPH '99 Conference Proceedings, pp317-324.
- [2] J. Vollmer, R. Mencl, and H. Müller. Improved Laplacian Smoothing of Noisy Surface Meshes. EUROGRAPHICS '99 Conference Proceedings.
- [3] D. Baraff, A. Witkin. Large Steps in Cloth Simulation. SIGGRAPH 98 Conference Proceedings, pp.43-54.
- [4] N. Amenta, M. Bern, M. Kamvysselis. A New Voronoi-Based Surface Reconstruction Algorithm. In SIGGRAPH '98 Proceedings. pp.415-421.
- [5] Kobbelt, L., Campagna, S., Vorsatz, J., and Seidel, H. P. Interactive Multiresolution Modeling on Arbitrary Meshes. In SIGGRAPH '98 Conference Proceedings. pp.105-114.
- [6] Kobbelt, L. Discrete Fairing. In Proceedings of the Seventh IMA Conference on the Mathematics of Surfaces, pp.101-131, 1997.
- [7] Brian Curless and Marc Levoy. A Volumetric Method for Building Complex Models from Range Images. In SIGGRAPH 96 Conference Proceedings, pp.303-312.
- [8] Gabriel Taubin. A Signal Processing Approach to Fair Surface Design. SIGGRAPH 95 Conference Proceedings, pp351-358.
- [9] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle. Piecewise Smooth Surface Reconstruction. In SIGGRAPH 94 Conference Proceedings, pp.295-302.
- [10] W. Welch and A. Witkin. Free-form shape design using triangulated surfaces. In SIGGRAPH 94 Conference Proceedings, pp.247-256, July 1994.
- [11] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface Reconstruction from Unorganized Points. In SIGGRAPH '92 Proceedings, pp.71-78, July 1992.
- [12] H. P. Morton and C. H. Sequin. Functional Optimization for Fair Surface Design. In SIGGRAPH 92 Conference Proceedings, pp.167-176, July 1992.
- [13] W. Welch and A. Witkin. Variational Surface Modeling. In SIGGRAPH 92 Conference Proceedings, pp.157-166, July 1992.

Non-shrinking Denoise Algorithm for Polygonal Meshes

Liu Xinguo Bao Hujun Peng Qunsheng

State Key Lab. of CAD&CG, Zhejiang University

Abstract In this paper we propose an efficient non-shrinking denoise algorithm used to produce smooth polygonal meshes by removing geometric noise. The smoothing problem is first converted into a minimization problem of energy functional. Since directly solving this minimization problem may cause extreme geometric shrinkage and distortion, and finally lead to the failure of smoothing, we present a conditional minimization equation with extra constraints on the facets' center of gravity. An effective iterative scheme is designed to solve the large equation. The experimental results demonstrate that our algorithm can remove the coarse noise of the models while preserve their shape features very well, and has many computational advantages such as low cost, high stability, fast convergence etc.

Keywords: Polygonal Mesh, Energy Functional, Laplacian Operator, Surface Fairing.