

基于能量评估的网格简化算法及其应用*

陶志良 潘志庚 石教英

(浙江大学 CAD&CG 国家重点实验室 杭州 310027)

(浙江大学计算机系 杭州 310027)

摘要 网格简化是曲面重构和多细节层次构造中的一个重要步骤,本文提出了一个基于局部能量评估的网格简化算法,能够大量地删除密集网格的三角形面片。为了加速算法的执行还提出了一种简单有效的采样方法。算法自动化程度高,能有效地支持多细节层次模型的建立。

关键词 网格简化,几何模型,网格匹配,虚拟现实。

中图法分类号 TP391

表面重构技术所生成三角化网格中的三角形数目通常很多。^[1]为了模型操作、绘制、修改和信息压缩的需要,对这种三角形数目庞大的网格进行简化十分必要。网格简化可以用来生成多细节层次描述模型。而多细节层次描述模型是虚拟现实系统中实现限时计算的一项关键技术。^[2,3]在网格简化方面,国内外都相继开展了一些卓有成效的研究开发工作,但总的来说,目前并没有十分理想的通用网格简化算法。对任意拓扑形状的网格都能达到既要网格形状保持不变,又要网格数目少的要求是非常困难的。在 Schroeder 等人^[4]的方法中,他们通过在局部区域中评估顶点到相关平均平面的距离来判断这个顶点是否应该删除;如果删除,则重新进行周围顶点的三角化过程。这个算法能够做到大量删减三角面片的目的,但由于在迭代过程中误差不断积累,不能确保与原始网格的很好匹配。Turk^[5]提出了一个基于粒子互斥模型的多层次网格简化方法,算法将新的数据点分布到原始三角网格表面,利用互斥原理使它们保持平衡并合理地分布,然后删除原有顶点,对新顶点所组成的图形重新三角化。由于算法采用了简单的模型,在某些复杂模型的简化过程中并不能取得理想的结果。在文献^[6]中,提出了一种新的基于特征角准则的多面体模型简化方法,算法适合对网格进行少量的删除。H. Hoppe^[7]等提出了一种整体的网格优化过程,既包括网格匹配过程,又包括了网格简化过程。优化算法需要两组输入数据,建立比较复杂的全局能量优化方程。整个优化过程所需的计算量大,自动化程度不高,不能满足限时计算的要求。

基于上述讨论,我们对 H. Hoppe 等人提出的网格优化算法进行改进,提出了一个基于

* 本文研究得到国家自然科学基金和浙江大学曹光彪科技基金资助。作者陶志良,1974年生,硕士生,主要研究领域为计算机图形,虚拟现实。潘志庚,1965年生,博士,研究员,主要研究领域为虚拟现实,多媒体计算,分布式图形。石教英,1937年生,教授,博士生导师,主要研究领域为科学计算可视化,虚拟现实,多媒体。

本文通讯联系人:潘志庚,杭州 310027,浙江大学 CAD&CG 国家重点实验室

本文 1997-06-19 收到修改稿

边操作的、适用于任意拓扑形状的网络简化算法. 为了有效地调整网络简化和网络形状匹配的关系, 我们引入了评估简化元操作的能量值. 同时也可以根据要求, 建立不同细节层次的网格模型. 与其它算法不同, 本算法可以在大量删减面片的同时, 保持原模型棱角特征, 并能够根据能量值的变化评估网格匹配的程度和误差. 而且自动化程度比较高, 所需要的计算量不大, 可以根据简化要求调整.

本文第 1 节介绍网格描述方法并给出一组基本定义, 以方便算法的进一步描述; 第 2 节详细描述算法; 第 3 节给出网格简化方法在多细节层次模型生成中的应用, 最后是结论.

1 网格描述方法及相关定义

为描述方便, 我们先定义一些基本概念, 并简述算法基本步骤.

1.1 网格描述

三角形网格是由三维空间中的三角形相互通过边和顶点连接而成的分段线性曲面, 其中每条边最多包含在两个三角形之中. 我们可以用网格中顶点的位置和相互关系来表示一个网格. 定义网格 $M=(K, V)$, 其中 K 包含了顶点的连接关系; $V=\{v_1, v_2, \dots, v_m\}$, $v_i \in R^3$ 表示网格 M 的顶点在三维空间中的位置.

如果 K 满足下面的条件, 就称 K 为一个单纯复形: K 包含了顶点集 $\{1, 2, \dots, m\}$ 及其非空子集. 特别地, 把 $\{i\} \in K$, $\{j, k\} \in K$ 和 $\{i, j, k\} \in K$ 分别称作点、边和面, 它们都是 K 中的一个单纯形. 对一个给定的单纯复形 K , 通过定义顶点集 $\{1, 2, \dots, m\}$ 为 R^m 中的标准基向量, 形成 K 在 R^m 中的拓扑表征 (Realization), 记为 $|K|$.

对任意单纯形 $s \in K$, $|s|$ 表示 s 在 R^m 中的凸包. 令 $|K| = \bigcup_{s \in K} |s|$, 定义从 R^m 空间到 R^3 空间的映射: $\varnothing_v: R^m \rightarrow R^3$. $\varnothing_v(|K|)$ 称为 M 的几何表征, 这样, 网格就表示成三维空间中的面.

如果 $\varnothing_v(|K|)$ 是一一映射, 即没有表面相交现象; 如果对任意三维空间中的点 $p \in \varnothing_v(|K|)$, 且仅存在点 $b \in |K|$, 使得 $p = \varnothing_v(b)$, 则我们称 b 为 p 的重心坐标.

b 可以表示成标准基向量 e_i, e_j 和 e_k 的凸组合, 其中 $\{i, j, k\} \in K$ 为 K 中的单纯形. 这里, 每个 b 最多有 3 个非零项. 而当 $b \in |\{j, k\}|$ 时仅有两个非零项, 即 b 点落在边 $\{j, k\}$ 上; 当 $b \in |\{i\}|$ 时与顶点重合, 只有一个非零项.

1.2 算法基本思想

我们希望用一个能量值表示网格匹配程度, 指导简化元操作的进行. 为了记录原始网格 M_0 的形状信息, 需要先从网格表面采样点集 X ; 能量值就表示了点集 X 与简化网格 M 之间的距离, M 匹配原始网格 M_0 越好, 能量值就越小. 由此, 我们就能利用能量最小化原理指导简化元操作的进行.

下面列出算法的基本步骤:

- 步骤 1: 读入原始网格 M_0 , 网格数据结构初始化;
- 步骤 2: 在 M_0 表面采样点集 X , 保存形状信息;
- 步骤 3: 选择符合条件的边 $\{j, k\}$, 建立局部能量方程, 计算原始能量值 E ;
- 步骤 4: 删除被选择边 $\{j, k\}$, 重新计算能量值 E' , 如 $E' > E$ 就放弃此简化操作;
- 步骤 5: 判断条件是否收敛, 如果不收敛转第 2 步;

步骤 6: 输出网格 M_n .

2 算法具体描述

算法通过不断比较,寻找网格匹配程度和网格简化程度之间的平衡点,指导简化元操作的进行.为了有效地记录原始网格 M_0 的形状信息,可以先从网格表面采样点集 X . 能量值表示 X 中的点与简化网格 M 之间的距离.我们可以利用能量值最小化原理,通过解能量方程来指导简化元操作的进行.

2.1 网格表面采样

采样点集要求能够记录足够多的网格细节,在平坦的网格面上仅需少量的点,而在某些细节丰富的部位则需足够多的采样点表征.因此,简单地进行均匀采样是不合适的,会引起细节模糊,形成不规则的网面.但是如果为了避免这种情况而加大采样密度,生成数目庞大的采样点集,则会明显地影响算法的运行速度.这就有必要在两者之间进行调节,寻找一种有效的采样方法.我们给出一种基于特征的采样方法:给定任一初始网格 $M_0=(K,V)$,假定 M_0 无自相交现象且很好地表现模型的各个细节.这样,我们首先想到,由于三角形网格分布平衡,顶点集 V 中的点就成为一部分理想的采样点. V 的数目相对讲并不大,大约平均每两个三角形上采一个点.由此可见,仅仅这部分点来表示整个网格信息是不够的.我们希望在某些特征区域多采集样点,比如棱边和尖角周围等.为了更好地说明这一点,做以下定义

$$\text{Sharp}(\{j,k\}) = \begin{cases} \text{TURE} & \text{Dihedral}(\{j,k\}) < \theta \\ \text{FALSE} & \text{Dihedral}(\{j,k\}) \geq \theta \end{cases} \quad (1)$$

$$E_\theta = \{\{j,k\} \mid \{j,k\} \in K \text{ 且 } \text{Sharp}(\{j,k\})\} \quad (2)$$

其中 $\{j,k\}$ 为 K 中的边, $\text{Dihedral}(\{j,k\})$ 表示关于 $\{j,k\}$ 的二面角, θ 为一个二面角的阈值,当关于边 $\{j,k\}$ 的二面角小于阈值 θ 时,认为此边是棱边,也即特征边,而 E_θ 为关于 θ 的棱边集合.根据上述表示,我们定义 E_θ 上的采样点集 X_θ

$$X_\theta = \{x \mid x = \frac{1}{2}(v_j + v_k), \{j,k\} \in E_\theta\} \quad (3)$$

于是最终得到采样点集 X 为两者之和,共 n 个点

$$X = V \cup X_\theta = \{x_1, x_2, \dots, x_n\} \quad (4)$$

X 具有采样点数目适中、有针对性和关于 θ 值可调节性等特点.通过实践,证明这种方法十分有效.并且根据点的来源,我们可以立即找到与采样点有投影关系的三角形,算出重心坐标 b_i ,为进一步计算打下基础.

2.2 能量方程的定义

简化网格 M 与 M_0 的匹配程度可以用网格 M 与点集 X 之间的距离来表示,定义距离能量 E_{dist} 为 X 中的点到 M 表面的投影平方距离之和

$$E_{dist}(K,V) = \sum_{i=1}^n d^2(x_i, \varnothing_v(|K|)) \quad x_i \in X \quad (5)$$

其中
$$d^2(x_i, \varnothing_v(|K|)) = \min_{b_i \in K_i} \|x_i - \varnothing_v(b_i)\|^2 \quad b_i \in R^m \quad (6)$$

其中 $\varnothing_v(|K|)$ 是 M 的几何表征, $d(x_i, \varnothing_v(|K|))$ 为 x_i 到 M 的最短距离.如图 1 所示,在 $|K|$ 空间中,顶点 v_1, v_2 和 v_3 分别对应基向量 e_1, e_2 和 e_3 .重心坐标 $b_i \in \{|1, 2, 3|\}$, 且 $b_{i1} = r$,

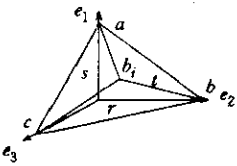


图1 重心坐标示意

$b_{i2}=s, b_{i3}=t$; 对于其他 $j \neq 1, 2, 3, b_{ij}=0$, 其中 r, s, t 满足下列条件:

(1) $r+s+t=1$;

(2) r, s 和 t 分别正比于三角形 bcb_i, acb_i 和 abb_i 的面积值.

假设点 x_i 在面 $\{1, 2, 3\}$ 上的投影点为 p_i , 我们有

$$p_i = \mathcal{O}_v(b_i) = b_{i1}v_1 + b_{i2}v_2 + b_{i3}v_3 = rv_1 + sv_2 + tv_3 \quad (7)$$

根据前面所介绍的采样点集 X , 对 $M_0 = (K, V), E_{dis}(K, V) = 0$. 由能量方程的定义可知, 初始网格 M_0 的匹配程度是最好的. E_{dis} 是一项表征网格匹配的能量值, 通过初始网格的试验, 能够说明定义比较合理. 为了与网格简化相统一, 还要增加表征网格简化的能量项

$$E_{rep} = cm, \quad (8)$$

m 为顶点数目, c 为简化能量系数

结构越简单, 顶点数目越少, 使能量降低. 网格简化过程可以看成通过设定系数 c , 不断调整 E_{dis} 和 E_{rep} 两项能量因子, 得到一个网格平衡状态 M_n , 也就是最终的结果. 这样, 总的能量方程为

$$E(K, V) = E_{dis}(K, V) + E_{rep}(K) \quad (9)$$

2.3 网格简化的元操作

本算法进行网格简化的基本操作是边操作, 即以删除网格边为基本步骤. 如图 2 所示, 我们定义把边 $\{i, j\} \in K$ 合并成新顶点 $\{h\} \in K'$ 的操作为一个网格简化元操作 OP . 元操作在网格中关于 $\{i, j\}$ 的局部区域 $Star\{\{i, j\}, K\}$ 上进行, 结合图 2 可以得到关于局部区域 $Star$ 的定义, 对所有单纯形 $s \in K$

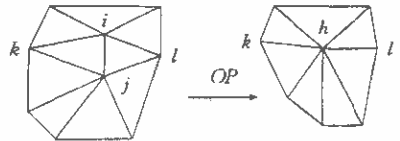


图2 基本边操作OP

$$Star\{s, K\} = \{t | t \cap s \neq \emptyset, \text{且 } t \in K\}; \quad (10)$$

于是可以得到对任何边 $\{i, j\} \in K$ 的网格简化元操作为 $OP: Star\{\{i, j\}, K\} \rightarrow Star\{\{h\}, K'\}$. 但是, 并非所有的边都能进行元操作. 元操作 OP 有定义的条件是:

- ① 对所有同时与点 $\{i\}$ 和 $\{j\}$ 存在连线的点 $\{k\}$, 必须保证 $\{i, j, k\}$ 是 K 中的一个面;
- ② $\{i, j\}$ 为非边界边, 且顶点 $\{i\}, \{j\}$ 不能同时为边界点;
- ③ $E(K', V') < E(K, V)$, 其中 K' 和 V' 分别为经过简化操作后的 K 和 V .

经过一次元操作 OP , 能量的变化 $\Delta E = \Delta E_{dis} + c\Delta m = \Delta E_{dis} + c$, 因此 c 值与 m 的关系并不大. 由于局部区域以外的其他部分网格并无变化, 故第 3 个条件实际上是指区域 $Star\{\{h\}, K'\}$ 上的能量值要小于区域 $Star\{\{i, j\}, K\}$ 上的能量值, 即 $E_{dis}(Star\{\{h\}, K'\}, V') < E_{dis}(Star\{\{i, j\}, K\}, V) + c$. c 值实际上表示了局部简化操作的最大误差. 其中原始区域上的能量值很容易计算出来, 而前者由于新顶点 h 的位置不定, 使得投影距离和顶点位置相互影响, 使得直接根据能量方程计算能量值非常困难. 为了解这个非线性问题, 我们利用了一些近似计算和迭代计算方法.

相应于局部区域 $Star\{s, K\}$, 有 X 中投影到 $Star\{s, K\}$ 上的采样点集 X_i . 我们先给新顶点 h 取一个初始值 v_h , 然后对所有点 $x'_i \in X_i$ 重新在 $Star\{\{h\}, K'\}$ 上进行投影操作, 找到投影面 $\{h, f_i, g_i\}$, 计算重心坐标 b'_i . 再在固定投影的基础上, 找到能使局部能量值最小的 h 点. 这样, 在这个局部区域的能量方程表示成

$$\min_{v_h} \sum_{x_i \in X_i} \|x'_i - \varphi_v(b'_i)\|^2 = \min_{v_h} \sum_{x_i \in X_i} \|x'_i - (b'_{ih}v_h + b'_{if_i}v_{f_i} + b'_{ig_i}v_{g_i})\|^2 \quad (11)$$

由于方程中关于 b'_i 的系数固定,方程可以转化成线性方程组 $Av_h = d$ 的形式,用最小二乘法来解得

$$\sum_{x_i \in X_i} \|x'_i - (b'_{ih}v_h + b'_{if_i}v_{f_i} + b'_{ig_i}v_{g_i})\|^2 = 0 \quad (12)$$

$$\text{令} \quad A = [b'_{1h}, b'_{2h}, \dots, b'_{lh}]^T \quad (13)$$

$$d = [x'_{i_1} - b'_{1f_1}v_{f_1} - b'_{1g_1}v_{g_1}, x'_{i_2} - b'_{2f_2}v_{f_2} - b'_{2g_2}v_{g_2}, \dots, x'_{i_l} - b'_{lf_l}v_{f_l} - b'_{lg_l}v_{g_l}]^T \quad (14)$$

于是根据最小二乘法和上述推导,可得

$$A^T Av_h = A^T d \quad (15)$$

$$v_h = A^T d / (A^T A) \quad (16)$$

当然如此得到的 v_h 值并不是方程的最优解,因为我们把重心坐标 b'_i 的计算和 v_h 的计算分开进行,使原先的非线性计算近似成了线性计算.值得一提的是,为了取得较好的结果,还需要将这个投影和解线性方程的过程多次迭代,不断逼近非线性方程的最优解.一般迭代次数取 5~6 次就能取得比较满意的结果.实际上, h 初始取值 v_h 对于迭代方向和逼近速度十分重要,我们可以取 3 个初值 v_i, v_j 和 $\frac{1}{2}(v_i + v_j)$, 分别计算局部能量变化,选最优结果.由于投影操作是一个很大的瓶颈,我们也可以只取两个初值 $(\frac{2}{3}v_i + \frac{1}{3}v_j)$ 和 $(\frac{1}{3}v_i + \frac{2}{3}v_j)$, 同时相应地增加迭代次数,也能得到比较满意的结果,而且提高运行速度.

2.4 算法的执行结果及相关问题

算法不断地从网格上选取边并对其进行网格简化元操作.通过设定简化能量系数 c , 或者预定删除的三角形数目,我们可以设置算法迭代过程的收敛条件.下面介绍算法的执行结果及相关问题.

(1) 毛刺噪声的检查

在实验过程中,由于某些局部区域 $Star\{(i, j), K\}$ 上相关的采样点很少,甚至没有,使得新顶点 v_h 的取值范围极大,从而产生长刺现象.为了避免这种不规则的三角网格出现,我们定义了一个限值系数 α , 规定元操作迭代中新顶点 v_h 的相对偏移不得超过 α . 参照标准可以是周围顶点之间的距离.一般如果产生长刺, v_h 值畸变幅度会很大,通过内部比较,我们就容易找到系数 α , 因此这种方法比起在能量方程上再增加一项弹性能量约束项^[7]要简单实用得多.

(2) 自相交的检查

在元操作的引入过程中,实际也引入了改变网格拓扑结构的可能性.因此有可能使原来并无自相交现象的网格产生自相交现象,引起网格的不规则变化.如图 3 所示,边 $\{i, j\}$ 经过合并操作后,三角形 $\{i, k, m\}$ 和 $\{j, k, m\}$ 重合在一起,破坏了网格的拓扑结构.

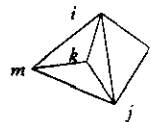


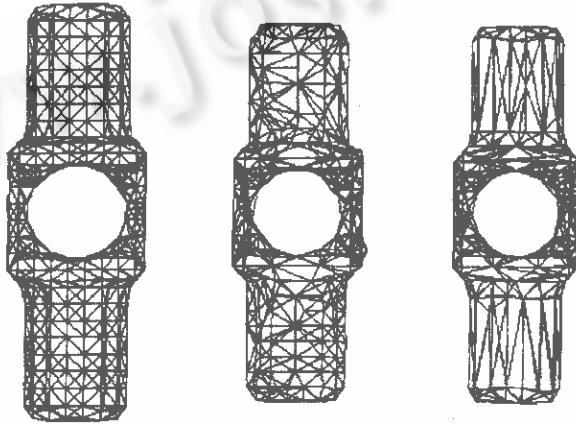
图3 自相交的检查

同样地,为了消除这种现象,我们再定义一个关于二面角的阈值 β , 在每次迭代后对所有边 $\{j, k\} \in Star\{h, K'\}$ 检查其相关二面角是否越过阈值 β 来判断是否需要删除这个元操作.如果生成的二面角小于阈值 β , 则放弃这项操作.在如图 3 所示

的情况中,生成的二面角为零,加了这个检查后,就避免了自相交现象.通过这个方法,一般都能有效地进行网格简化操作而不破坏网格的拓扑形状.

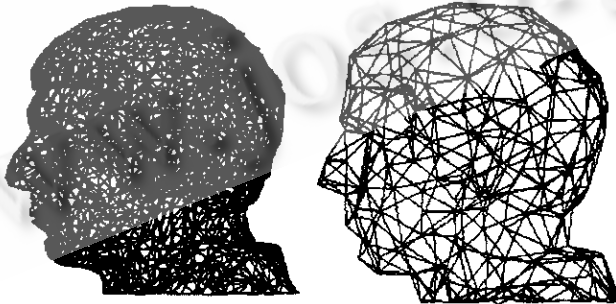
(3)实验结果

我们进行了不同类型的几组网格数据实验(结果如图4~6),其中能量系数 c 的取值都为 10^{-5} ,可以看到简化后的网格能够很好地保持初始网格的形状,特别是棱角等特征细节的保形.事实上,我们对简化后的网格模型用简单的 flat-shading 算法进行显示观察,效果不错,并且某些模型的失真度极小,如图4零件模型.一般这些网格三角面片的删除率可大于80%,较高的可达90%以上.其中图4的数据是通过CAD系统生成的零件模型进行取样后再使用 Marching Cube 算法重构得到;图5的数据取自于Internet网;图6的数据是由我们自己生成采样点集后再用 Marching Cube 算法生成网格.在实际工作中,如果网格数目变化很大,我们还可以动态地调节 c 值的大小.



原始网格共1774个三角形,868个顶点.简化后的两组网格分别有698个三角形,345个顶点和524个三角形,258个顶点,原始网格并非密集形网格,因此简化模型的中间部分变化不多.

图4 零件模型



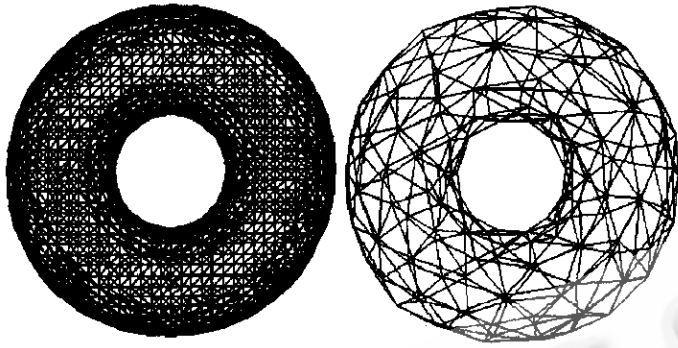
原始网格三角形数目为3525个,顶点数目为1779个.简化后的网格有469个三角形,251个顶点.

图5 人头模型

3 多层次细节模型的建立

基于边的元操作每次删除1个点、1条边和2个面,我们可把元操作序列看成一个迭代过程

$$M_{i+1} = M_i + Star\{\{h\}, K'\} - Star\{\{i, j\}, K\} \tag{17}$$



原始网格共 5 958 个三角形, 2 979 个顶点. 简化后的网格有 334 个三角形, 167 个顶点. 由于原始网络是密集型网格, 简化率已将近 95%.

图 6 圈模型

只要选定了迭代的次数, 或者选定介于 M_0 和 M_n 之间的某个层次 i , 就可以计算出相应的网格 M_i . 即令层次系数 $level \in [0, 1]$, 有 $M(level) = M_i$, 其中 $i = INT(n \times level)$. 实际上, 网格 M_i 的具体形式是与进行元操作的边序列相关的. 同一个层次系数 $level$, 不同的边元操作序列会生成不同的网格. 特别地, 相邻的边操作序列就会产生局部过度简化现象. 为了避免由于相邻边连锁反应产生的不均匀现象, 我们产生一个有优先级的边序列后备集 E_p , 具有下列规则:

① 初始化时, E_p 内放入所有的边 $\{i, j\} \in K$, 且每边具有相同的最高优先级 P_1 ;

② 经过每次元操作后, 生成新顶点 $\{h\}$, 把所有边 $\{h, k\} \in Star\{\{h\}, K'\}$ 放入 E_p , 并置优先级 $P(\{h, k\}) = P(\{i, j\}) + 1$;

③ 每次元操作后, 从 E_p 中取出操作边 $\{i, j\}$.

元操作序列从 E_p 选择操作边按优先级从高到低的原则进行, 相应于 P 值从小到大, 如此一直到 E_p 收敛为止, 则算法结束. 按照这种方法生成的边序列不会在局部区域内产生不良的连锁反应, 而且还能有效地支持多层次细节模型的建立.

4 结 论

基于能量方程的网格简化算法能够找到合适的新顶点, 保持原有的拓扑形状, 并且大量地删除密集网格面上的三角形.^[7] 算法同时建立了一个评估网格简化误差的数学模型, 弥补了其它方法的不足. 然而, 由于算法运行过程中要不断进行投影和最小二乘法计算工作, 影响了算法的速度. 而相关的主要因素是采样点集数目, 为了减少由此带来的速度问题, 我们提出了一种简单有效的采样方法.

本算法自动程度高, 算法中的各种参数基本与具体的网格大小和形状无关, 可以事先根据应用的要求设定. 当然也可以在运行过程中动态的设定. 我们已经在实践中运用了动态参数方法, 取得比较满意的效果.

该算法已作为一个模块, 集成到浙江大学 CAD&CG 国家重点实验室和日本 3D 公司联合开发的“景观设计系统”中. 与原有的 H. Hoppe 等人提出的算法相比, 该算法去除了并不重要的弹性能量 E_{spring} , 采用了一种简单有效的采样方法, 大大提高了算法的速度. 另外,

提供了有效控制简化过程的动态参数设定方法。

致谢 浙江大学应用数学系博士生郑文庭同志参加了部分基础研究工作,日本 3D 公司的 Shirakata 先生对本算法提出了多项有益的改进建议,在此一并表示感谢。

参考文献

- 1 Hoppe H, DeRose T, Duchamp T *et al.* Surface reconstruction from unorganized points. SIGGRAPH'92 Proceedings, 1992. 71~78.
- 2 潘志庚,马小虎,石教英. 虚拟环境中多细节层次模型自动生成算法. 软件学报, 1996. 7(9): 526~532.
- 3 Pan Zhigeng, Zhang Mingmin, Zheng Wenting *et al.* Time-critical computing in virtual environment. Proceedings of CAD/Graphics'95, Wuhan. 1995. 1076~1083.
- 4 Schroeder William, Zarge J, Lorensen W. Decimation of triangle meshes. SIGGRAPH'92, Proceedings, 1992. 217~226.
- 5 Greg Turk. Re-tiling polygonal surfaces. SIGGRAPH'92 Proceedings, July 1992. 55~64.
- 6 周晓云,刘慎权. 基于特征角准则的多面体模型简化方法. 计算机学报, 1996, 增刊: 212~223.
- 7 Hoppe H, DeRose T, Duchamp T *et al.* Mesh optimization. SIGGRAPH'93 Proceedings, 1993. 19~26.

MESH SIMPLIFICATION ALGORITHM BASED ON ENERGY EVALUATION AND ITS APPLICATION

TAO Zhiliang PAN Zhigeng SHI Jiaoying

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

(Department of Computer Science Zhejiang University Hangzhou 310027)

Abstract Mesh simplification is an important stage in the surface reconstruction and the generation of multiple LOD(level of detail) models. In this paper, the authors present a mesh simplification algorithm based on energy evaluation which can effectively reduce the number of triangles in a dense mesh. This algorithm is able to decimate the mesh while well keeping the original topological type of the mesh, and to evaluate the error of mesh fitness according to the energy value. To accelerate the algorithm, a simple effective sampling method is employed. In addition, this algorithm is highly automatic to support the construction of the LOD models.

Key words Mesh simplification, geometric modeling, mesh fitting, virtual reality.

Class number TP391