

面聚类网格简化新算法

严京旗, 施鹏飞

(上海交通大学图像处理与模式识别研究所, 上海 200030)

摘 要: 三维物体表面重建广泛采用三角网格方法, 密集的数据采样可以重建出精确的三维表面, 但是庞大的数据量不利于多分辨率三维实时显示和三维物体网络传输, 因此三维表面网格简化是迫切需要解决的问题之一. 近年来表面简化问题得到了广泛地研究. 本文提出基于面聚类的网格简化新算法, 通过最小化最大类内距离算法进行面聚类实现区域划分, 然后提取区域特征, 进而根据特征点和边对区域进行受限三角剖分. 实验说明本文提出的面聚类网格简化算法在保持三维表面几何拓扑特征的基础上取得了很好的简化效果.

关键词: 网格简化; 表面区域划分; 聚类; 受限三角剖分

中图分类号: TP391

文献标识码: A

文章编号: 0372-2112 (2002) 01-0038-04

Mesh Simplification with Face Clustering

YAN Jing-qi, SHI Peng-fei

(Inst. of Image Processing & Pattern Recognition, Shanghai Jiaotong Univ., Shanghai 200030, China)

Abstract: The need for an accurate simplification of surface meshes is becoming more and more urgent in many applications. It not only suits multiresolution/level-of-detailed representations and quick visualization, but also allows fast transmission of 3D models in network-based applications. Many different approaches and algorithms for mesh simplification have been proposed in the last few years. In this paper we present a new method, which partitions region by minimizing the maximum intercluster distance clustering algorithm, extracts features in every region and triangulates them by constrained delaunay triangulation. The results of examples illustrate our simplification method is effective.

Key words: mesh simplification; surface segmentation; clustering; constrained triangulation

1 引言

三角形是最基本的绘图元素之一, 重建三维表面时广泛采用三角网格方法, 很多复杂的形体都是由许多三角形面构成. 为了得到精确的三维表面, 数据采集往往较为密集, 因而重建后表面包含成千上万个三角形面. 巨大的数据量不利于多分辨率三维实时显示和三维物体网络传输, 因此三维表面网格简化是迫切需要解决的问题之一. 近年来网格简化问题引起了广泛地研究, 提出了许多不同的简化算法^[1]. 主要有基于点删除的简化算法^[2], 基于边合并的简化算法^[3], 基于能量函数优化的简化算法^[4], 基于局部或全局误差控制的简化算法^[5~7]等.

2 面聚类网格简化算法

本文提出基于面聚类的网格简化新算法, 通过最小化最大类内距离算法进行面聚类实现区域划分; 在每个区域提取区域边缘等特征予以保留, 删除区域内非特征点和边; 最后在每个区域内根据留下的特征点和边对区域进行受限三角剖分.

2.1 基于面聚类的区域划分

进行面聚类需要定义两个面之间的距离, 为此把三角网格 $M(T, V)$ 转化为图 $G(N, C)$, 其中 T, V 分别为网格中三角形和顶点的集合, N, C 分别为图 G 中的节点集合和连接关系. 转化时, 把网格 M 中的一个三角形 T_i 看作图 G 中的节点 N_i ; 若两个三角形面 (T_i, T_j) 有公共顶点 V_i , 则相应的节点 (N_i, N_j) 之间有连接 C , 且:

定义: 连接权值 $C(N_i, N_j)$ 为两个三角形 T_i, T_j 外法矢的夹角; 若两个三角形没有公共顶点, 则相应的节点间没有连接即连接权值为 $+\infty$. 三角网格与图之间的转化如图 1 所示.

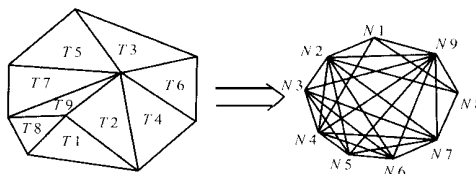


图1 三角网格 $M(T, V)$ 转化为图 $G(N, C)$

在图 $G(N, C)$ 中, 找出最短路树形图 $\text{Tree}(N, C, R)$, 其中 R 是最短路树形图的根节点. 对于不同的根节点会有不同的最短路树形图, 如图 2 所示.

定义: 两个三角形面 T_i, T_j 之间的距离 $d(T_i, T_j)$ 是: 在以节点 N_i 为根节点的最短路树形图 $\text{Tree}(N, C, N_i)$ 中从节点 N_i 到节点 N_j 通路上连接权值总和. 若节点 N_i 到节点 N_j 没有通路时 $d(T_i, T_j) = +\infty$.

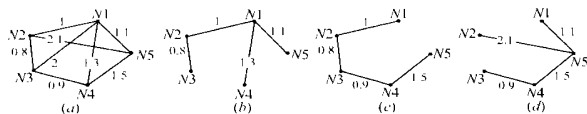


图 2 (a) 图 $G(N, C)$; (b) 以 $N1$ 为根的最短路树形图; (c) 以 $N3$ 为根的最短路树形图; (d) 以 $N5$ 为根的最短路树形图

在图 2(b) ~ 2(d) 中,

$$d(T_1, T_3) = C(N_1, N_2) + C(N_2, N_3) = 1 + 0.8 = 1.8$$

$$d(T_3, T_5) = C(N_3, N_4) + C(N_4, N_5) = 0.9 + 1.5 = 2.4$$

$$d(T_5, T_2) = C(N_5, N_2) = 2.1,$$

按上述方法定义三角形面 T_i, T_j 之间的距离 $d(T_i, T_j)$ 之后, 可以用最小化最大类内距离聚类算法进行面聚类, 从而完成区域划分, 具体步骤如下:

Step 0:

设定算法的结束条件即最大类别数 MaxClassNum . 记 $S = \{ \text{class}_0, \text{class}_1, \dots, \text{class}_n \}$, 其中 class_i 表示 T_i 属于 class_i 类. 并把每个三角形到所属类的类代表三角形之间的距离记为 $D = \{ d_0, d_1, \dots, d_{n-1} \}$. 即如果 T_i 为第 j 类的三角形, h_j 为第 j 类的类代表三角形, 则 $d_i = d(T_i, h_j)$. 初始, 所有的三角形均为 0 类即 $S = \{ 0, 0, \dots, 0 \}$, 类别数 $\text{count} = 0$. 任选一个三角形作为 0 类的类代表三角形 h_0 , 设 $h_0 = T_0$. 构造以类代表三角形 h_0 相应的节点 N_0 为根的最短路树形图 $\text{Tree}(N, C, N_0)$, 计算出从三角形 h_0 到其它三角形的距离 $d(T_i, h_0)$, 相应地更新 D 中的值.

Step 1:

从 D 中找出最大值, 设为 d_i , 此时把 d_i 对应的三角形 T_i 移到新的第 $\text{count} + 1$ 类即 $\text{class}_i = \text{count} + 1$, 并且把三角形 T_i 作为新类的类代表三角形即 $h_{\text{count}+1} = T_i$.

Step 2:

构造以新类代表三角形 $h_{\text{count}+1} (= T_i)$ 相应的节点 N_i 为根的最短路树形图 $\text{Tree}(N, C, N_i)$, 计算出从三角形 $h_{\text{count}+1}$ 到其它三角形 T_j 的距离 $d(T_j, h_{\text{count}+1})$, $j = 0, 1, \dots, n-1$. 如果 $d(T_j, h_{\text{count}+1}) < d_j$ (此处 d_j 是表示三角形 T_j 到原来所属类的类代表三角形之间的距离, 可从 D 中读取), 则把三角形 T_j 移到第 $\text{count} + 1$ 类, 即 $\text{class}_j = \text{count} + 1$, 并相应地更新 D 中的值, 即置 $d_j = d(T_j, h_{\text{count}+1})$.

Step 3:

$\text{count} = \text{count} + 1$. 如果 $\text{count} < \text{MaxClassNum}$, 则转到 Step 1 继续; 否则算法结束.

按照上述面聚类区域划分算法, 如图 2(a) 的网格, 若分成两类则如图 3(a) 所示, 即 $\{ T_1, T_5 \}$ 和 $\{ T_2, T_3, T_4 \}$; 若分成三类则如图 3(b) 所示即 $\{ T_1 \}$, $\{ T_5 \}$ 和 $\{ T_2, T_3, T_4 \}$, 在此例中, 0

类的类代表三角形为与节点 $N1$ 对应的三角形 T_1 .

2.2 特征提取

三维表面经区域划分后可对每个区域进行特征提取, 保留特征点、边, 删除非特征点、边和三角形, 达到网格简化的目的. 在特征提取前, 先对三维表面上的点作划分.

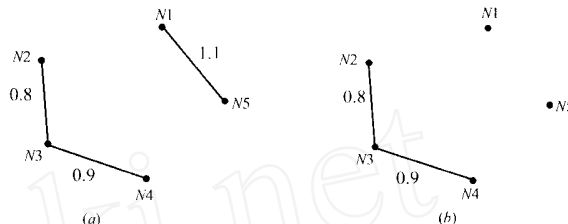


图 3 (a) 分成两类; (b) 分成三类

定义: 表面网格中的一个点被称为正则点 (regular vertex)^[8], 如果包含该点的所有三角形的外轮廓边组成一条连接的折线; 否则该点被称为奇异点 (singular vertex). 正则点又可分为: 内点 (interior vertex, 如果轮廓边是封闭的) 和边界点 (boundary vertex, 如果轮廓边是不封闭的). 分别如图 4 (a) ~ 4(c) 所示.

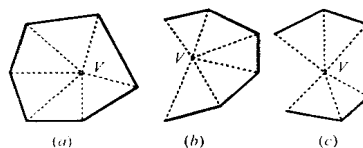


图 4 (a) 外轮廓边是一条封闭折线, v 为内点; (b) 外轮廓边是一条非封闭折线, v 为边界点; (c) 外轮廓边是两条相互不连接的折线, v 为奇异点.

在实验中, 将内点进一步划分为: 区域内点, 两个区域间交界点, 以及三个以上区域间角点. 区域内点是指与该点相连的所有三角形均属于同一区域的正则内点; 两个区域间交界点是指分属两个区域的正则内点; 三个以上区域间角点是指分属三个以上区域的正则内点. 将边界点进一步划分为: 区域内边界点和区域间边界点. 区域内边界点是指与该点相连的所有三角形均属于同一区域的正则边界点; 区域间边界点是指分属两个以上区域的正则边界点.

在提取每个区域的特征时, 删除区域内点, 将其他点作为特征点保留. 为了得到更简的网格, 两个区域间交界点和区域内边界点可以进一步

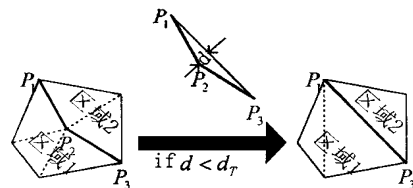


图 5 两个区域间交界点的简化

2.3 受限三角剖分

每个区域经过特征提取后, 留下的是特征点和特征边. 由于每个区域是近似平面的, 因而可以将特征点投影到二维平面上, 在投影平面上根据特征点位置对区域进行受限三角剖分.

分,新产生的三角形必须在区域之内. 设 $V_T = \{v_1, v_2, \dots, v_n\}$ 为区域内特征点的投影点, $E_T = \{e_1, e_2, \dots, e_m\}$ 为区域内的特征边, $e_i = \{v_{i1}, v_{i2}\}$. $T = \{\}$ 来记录产生的三角形,初始为空. 在记录时,三角形三个顶点按逆时针方向排列. $E_{old} = \{olde_1, olde_2, \dots, olde_p\}$, $E_{new} = \{newe_1, newe_2, \dots, newe_q\}$ 为三角化剖分过程中的轮廓边集合,初始均为空. $V_d = \{vd_1, vd_2, \dots, vd_h\}$ 为候选点,初始亦为空. 算法步骤为:

Step 0:

从 E_T 中任选一条未处理过的边, 设为 $e_1 = \{v_{11}, v_{12}\}$, 将其反向加入 E_{old} 中, 即 $E_{old} = \{\{v_{12}, v_{11}\}\}$.

Step 1:

考察 E_{old} 中的当前边 $olde_i = \{v_{i1}, v_{i2}\}$, 依据以下两个判别准则生成候选点集 V_d .

可见性准则: 若点 v_j 在边 $olde_i = \{v_{i1}, v_{i2}\}$ 的右边, 则点 v_j 为边 $olde_i$ 的可见点.

不相交性准则: 若线段 $\{v_j, v_{i1}\}$ 和 $\{v_j, v_{i2}\}$ 与 E_T 中的特征边(线段)重合或者没有交点或者交点是线段端点, 则点 v_j 满足不相交性准则.

若点 $v_j \in V_T$ 且满足可见性准则和不相交性准则, 则将点 v_j 加入 V_d .

Step 2:

从 V_d 中选择点 v_{third} , 使得 $v_{11} v_{third} v_{12} = \max\{v_{11} v_j v_{12} \mid v_j \in V_d\}$. 把由点 $v_{12}, v_{third}, v_{11}$ 组成的三角形加入 T , 即 $T = TU\{\{v_{12}, v_{third}, v_{11}\}\}$. 考察新产生的边 $\{v_{12}, v_{third}\}$ 和 $\{v_{third}, v_{11}\}$, 若 $\{v_{12}, v_{third}\} \notin E_T$ 且 $\{v_{12}, v_{third}\} \notin E_{old}$, 则将其加入 E_{new} 即 $E_{new} = E_{new} \cup \{\{v_{12}, v_{third}\}\}$, 对边 $\{v_{third}, v_{11}\}$ 做同样处理.

Step 3:

使 $V_d = \phi$. 若 E_{old} 中有下一条未处理的边, 则把该边作为当前边, 转到 Step 1 继续执行. 若 E_{old} 中所有边都已处理过, 则 $E_{old} = E_{new}$, $E_{new} = \phi$. 若 E_{old} 仍然为空, 则转到 Step 4; 否则转到 Step 1.

Step 4:

若 E_T 中仍有未处理过的边, 则转到 Step 0 继续执行; 否则算法结束.

如图 6(a) 所示为由特征点和特征边组成的区域, 执行受限三角剖分后如图 6(b) 所示, 所有新产生的三角形均在区域的内部.

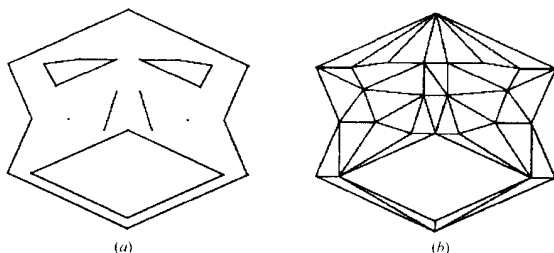


图 6 (a) 由特征点和边组成的区域; (b) 区域受限三角剖分结果

3 算法分析和实验结果

面聚类网格简化算法主要由区域划分、特征提取和受限

剖分三步组成. 其中, 基于聚类的区域划分所需计算时间为: $O(nm + T_s m)$, 其中 n 为点数目, m 为类别数目, T_s 为构造最短路树形图所需时间. 通常最短路径问题求解如 Dijkstra 算法的计算复杂度为 $O(n^2)$. 但在基于面聚类的区域划分过程中, 路径的权值是三角形面外法矢方向的夹角. 任意两节点 N_i 和 N_j (对应于三角形 T_i 和 T_j , 外法矢为 V_i 和 V_j) 之间的最短路径长度为 $l_{\min}(N_i, N_j) = (V_i, V_j) + k\pi$, k 为正整数. 因此, 在最短路树形图构造过程中, 从根节点 N_0 出发, 先扩展最短路径长度在 0 到 π 之间的叶节点, 再扩展最短路径长度在 π 到 2π 之间的叶节点, 依次类推, 构造出最短路树形图, 所需的计算复杂度 T_s 为 $O(n)$. 另外, 特征提取的计算时间为: $O(n)$. 以及, 受限剖分的计算运行时间为: $O(m * (\frac{n_s}{m})^2) = O(n_s^2 / m)$, 其中 n_s 为简化后剩余点个数. 由于通常 n_s 较小, 且当 n_s 增大时, m 也同时增加, 所以受限剖分所需的时间较小. 因此, 面聚类网格简化算法的计算量主要集中在区域划分上, 即总计算复杂度大致为 $O(2nm)$.

文献[1]对简化误差作了定义, 给定点 p 和表面 S , 则点 p 到表面 S 的距离为:

$$e(p, S) = \min d(p, p'), p' \in S;$$

给定表面 S_1 和简化表面 S_2 , 则简化的最大误差为:

$$E(S_1, S_2) = \max_{p \in S_1} (\max_{p' \in S_2} e(p, p')), \max_{p \in S_2} (e(p, S_1));$$

通过积分, 可以定义出空间体积的变化:

$$E_v(S_1, S_2) = \int_{S_1} e(p, S_2) ds;$$

则定义简化的平均误差为:

$$E_m(S_1, S_2) = \frac{1}{|S_1|} \int_{S_1} e(p, S_2) ds$$

实验中用 Mennequin (如图 7(a) ~ 7(d)), Fandisk (如图 8(a) ~ 8(d)) 两个应用来说明了面聚类三维表面网格简化算法的性能, 算法运行时间和简化误差如表 1 所示. 表中的约束箱体是指将物体完全包含的最小长方体, 最大和平均误差都是与该约束箱对角线的比值, 以百分比形式给出; 时间单位为: 秒; 运行环境为: CPU Pentium II 300, RAM 64MB.

表 1 算法性能统计表

对 象	顶点 数目	三角形 数目	最大误 差 (%)	平均误 差 (%)	运行时 间 (秒)
Mennequin (约束箱体为: 177.7 * 240.3 * 288.7, 原始网格顶点数目为 7834, 三角形个数为 15543)	4722 (60 %)	9420	0.2006	0.0188	75
	2358 (30 %)	4703	0.4110	0.0462	36
	1184 (15 %)	2362	0.7392	0.0874	22
	392 (5 %)	780	1.2331	0.1329	10
	160 (2 %)	318	2.0627	0.2813	3
Fandisk (约束箱体为: 4.8 * 5.6 * 2.7, 原始网格顶点数目为 6475, 三角形个数为 12946)	3884 (60 %)	7764	0.0047	0.0004	66
	1964 (30 %)	3924	0.0233	0.0024	28
	982 (15 %)	1960	0.0872	0.0065	17
	330 (5 %)	656	0.2264	0.0222	8
	146 (2 %)	288	0.5297	0.0556	4
	66 (1 %)	128	1.2746	0.2238	3

图 7(a) 是一个人像图, 表面网格包含 15543 个三角形, 数据量极大. 图 7(b) 是图 7(a) 三角形填充后带影调的表面显

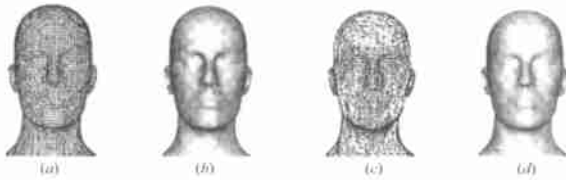


图 7 (a) 原始三角形框架显示 (15543 个三角形); (b) 原始表面显示; (c) 简化了 70 % 后的三角形框架显示 (4703 个三角形); (d) 简化了 70 % 后的表面显示

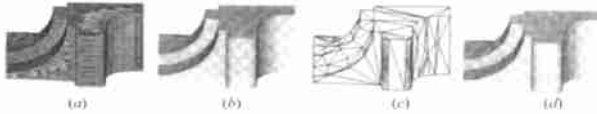


图 8 (a) 原始三角形框架显示 (12946 个三角形); (b) 原始表面显示; (c) 简化了 98 % 后的三角形框架显示 (288 个三角形); (d) 简化了 98 % 后的表面显示

示. 图 7(c) 是图 7(a) 简化了 70 % 后的表面网格显示, 包含 4703 个三角形. 图 7(d) 是图 7(c) 三角形填充后带影调的表面显示. 图 8(a) 是一个机械零件, 初始表面网格包含 12946 个三角形, 图 8(b) 是带影调的表面显示. 图 8(c) 是简化了 98 % 后的网格显示, 包含 288 个三角形, 图 8(d) 是简化后带影调的表面显示. 比较图 7(b) 和图 7(d)、图 8(b) 和图 8(d) 可以得出: 虽然数据量得到了很大的简化 (分别简化了 70 % 和 98 %), 但是简化的最大误差和平均误差很小 (仅约为 0.5 % 和 0.05 %), 简化后的表面基本上近似与原来的表面, 几何拓扑得到了很好地保留, 验证了本文算法的有效性.

4 结论

表面三角网格简化在减少三维物体存储和传输数据量以及实现多分辨率三维动态实时显示等领域具有广泛的应用前景. 为了得到较为精确的三维初始表面, 数据采样密度较高, 重建后的初始三角网格具有很大的冗余 (在一定的误差范围内), 通过表面网格简化可以大大地减少数据量. 本文提出的面聚类网格简化算法, 为网格简化提供了新的思路. 通过面聚类把网格中处在近似平面的三角形聚为一类, 最大限度地保留了三维表面几何拓扑细节, 减少了简化误差. 从实验可以得出: 本文提出的算法取得了很好的简化效果, 大大地降低了描述三维表面所需的三角形数目, 可以提高三维物体显示和传输速度.

参考文献:

- [1] P Cignoni, et al. A comparison of mesh simplification algorithms [J]. Computers & Graphics, 1998, 22(1): 37 - 54.
- [2] WJ Schroeder, et al. Decimation of triangle meshes [A]. In proc. SIGGRAPH 92 [C], 1992: 65 - 70.
- [3] J Cohen, et al. Simplifying polygonal models using successive mappings [A]. In proc. Visualization '97 [C], 1997: 395 - 402.
- [4] H Hoppe, et al. Mesh optimization [A]. In proc. SIGGRAPH 93 [C], 1993: 19 - 26.
- [5] J Cohen, et al. Simplification envelopes [A]. In proc. SIGGRAPH 96 [C], 1996: 119 - 128.
- [6] A Guezic. Surface simplification with variable tolerance [A]. In proc. Second Ann. Int'l Symp [C]. Medical Robotics and Computer Assisted Surgery, 1995: 132 - 139.
- [7] M Garland, P S Heckbert. Surface simplification using quadric error metrics [A]. In proc. SIGGRAPH 97 [C], 1997: 209 - 216.
- [8] A Guezic. Locally toleranced surface simplification [J]. IEEE Trans. Visualization and Computer Graphics, 1999, 5(2): 168 - 189.

作者简介:



严京旗 男, 1975 年 1 月生于浙江省东阳市横店镇. 现为上海交通大学图像处理与模式识别研究所博士生. 1996 年和 1998 年分别在上海交大自动化系和图像所获得学士、硕士学位. 主要研究兴趣: 彩色图像量化、三维表面重建、表面网格简化以及三维物体多分辨率显示等等.



施鹏飞 男, 1939 年生, 上海交通大学电子信息学院教授、博士生导师, 图像处理与模式识别研究所所长. 目前主要从事图像处理和模式识别与智能技术和系统方面的教学科研工作. 曾负责和承担过国家科技攻关、国家自然科学基金、高校博士点基金及国家 863 等项目十余项, 已发表学术论文百余篇, 并获多项科技奖励.