

三维图形简化新算法

陈礼民

秦爱红

(山西大学, 计算中心, 太原 030006) (山西大学, 计算机科学系, 太原 030006)

摘要 本文的图形简化新算法, 具有保细节、保真和快速等优点。快速是采用传播法快速形成近平面块的结果。保真和保细节都是可以控制的。算法规整, 效果良好, 适应面宽, 很有实用价值。

关键词 图形简化, 计算机图形, 多边形简化, 可视化

1 引言

在图形的计算机处理中, 为了逼真地表示一个3-D物体, 往往用成千上万个多边形图元来表示物体表面, 这对计算机的容量和处理速度都提出了很高的要求, 成为发展动画和多媒体等技术的瓶颈。因此, 图形能否简化就成为计算机图形学中令人感兴趣的重要课题。图形简化就是用最少的图元来逼真地表示一个物体的三维表面。由于图形简化的实际意义, 许多学者进行研究, 发表了不少论文, 如Schroeder^[1], Hinker^[2], Turk^[3], 和Hoppe^[4]等。

Schroeder的算法在保细节方面效果很好。Hinker避免了逐点处理, 提出了快速形成近平面块的算法。该算法的优点在于能快速处理大块的近平面块。但该算法在保真性方面没有专门措施, 算法的时间复杂度也有待改进。本文提出一个基于快速形成近平面块的新算法。该算法不仅有广泛的适应面, 而且也具有保细节、保真和快速等优点。

2 相关工作

图形表面简化的根本原理是用不同分辨率来表示图形不同的表面部分, 几何和拓扑性复杂部分采用较高的分辨率, 反之则采用较小的分辨率来表示。

Clark^[5]提出了分层的多种表示模式, 该算法靠减小网格的分辨率来产生简化的表面。

Schroeder提出了“Decimation of triangle Meshes”^[1]算法, 算法先估值网格上格点的局部的

拓朴和几何性质, 把格点分为简单、尖锐和复杂等类型。简单型又分为内部和边界两种类型。其次, 格点删除判据是 $d < \epsilon$ 。对内部点, d 是候选点到平均平面的距离。 ϵ 是给定的允许偏离值。对边界点, d 是候选点到其二邻居点连线的距离。最后对删除了的格点的邻域, 重新进行三角形划分。划分采用最佳劈分平面的方法(见第5节)。时间复杂度是线性的。算法优缺点:(1)有很好的保细节特性;(2)逐点处理, 导致删除和再划分十分频繁;(3)表面中平坦部分也是逐点处理, 没有较好办法”;(4)若候选点的邻域有丰富的拓朴和几何性质的对称图形时, 可能出现候选点到平均平面的距离为零, 满足删除条件, 导致细节丢失。

Hinker提出了“Geometric Optimization”^[2]算法。该算法能快速形成近平面块。算法可归纳为5步:(a)把法向量相近的三角形分在同一组;(b)移走重复边, 得到近平面块的边表;(c)边界跟踪, 形成一个或多个近平面块;(d)移走共线点;(e)对每个近平面块重新进行三角形划分。算法时间复杂度为 $O(n \log n)$ 。

分析上述可知, 若步骤(a)中诸近平面块可以分别形成, 则时间复杂度将减小; 另外, 形成近平面块有用的只是边, 所以步骤(b)不是必需的; 再有, 没有保细节措施和时间复杂度比线性的大等不足。为此本文提出一个基于快速形成近平面块的新算法。

3 图形简化算法简介

本算法分作两部分, 首先是迅速形成近平面块

的边界,删除块内所有图元,接着对每个近平面块进行三角形划分。具体可分下列几步:(1)估值三角形的边,并对其分类;(2)用传播法获得近平面块的边集;(3)边缘跟踪;(4)删除共线点,形成近平面块边界;(5)对近平面块重新进行三角形划分。

本算法用三角形作为描述物体表面的图元。把相邻三角形之间的拓扑特性赋予它们的相交边。相交边可以分为 3 类,复杂边:被 3 个或 3 个以上三角形所共用的边。简单边:被 2 个三角形共用的边。边界边:只属于 1 个三角形的边。

4 形成近平面块

3-D 物体表面简化,本质上就是用平面来代替曲面。所以算法的主要工作是形成一个近平面的曲面,称为近平面块。控制近似误差就可以控制算法的保真性。

一个候选三角形,能否加入近平面块中,应该满足 3 个条件:与近平面块是相邻的;满足保真性和保细节判据。满足了第一个条件可以避免属于不相邻的平面块的三角形挤在同一组内。保真性判据用于控制平面代替曲面的近似程度。保真不一定能保细节。所以引入保细节判据是十分必要的。

4.1 保真性判据 $a_f < \epsilon_f$

其中 ϵ_f 是允许误差, $a_f = 1 - \text{dot}(N, n) / \text{length}(n) \cdot \text{length}(N)$ 是保真性角特性值,表徵候选三角形和近平面块的关系, n 是候选三角形的法向量, N 是近平面块平均法向量, dot 是向量点积的函数, length 是向量长度的函数。 N 采用动态平均的方法,即随着近平面块中三角形的增加,不断修正 N 的值。但为了避免起始点影响,本算法采用同时考察近平面块所有邻居的办法。若有 m_j 个候选三角形满足保真性判据,用 n_k 分别表示三角形的法向量, m_j 个三角形法向量的和用 N_j 表示:

$$N_j = \sum_k^{m_j} n_k$$

m_j 个三角形加入近平面块后,新的近平面块的平均法向量为:

$$N = \sum_j^l N_j / \sum_j^l m_j$$

4.2 保细节判据 $a_d < \epsilon_d$

其中, $a_d = 1 - \text{dot}(n_i, n_j) / \text{length}(n_i) * \text{length}$

(n_j) 是保细节角特性, ϵ_d 为保细节允许角误差, n_i, n_j 为相邻三角形的法向量。也就是说候选三角形要加入近平面块,则它和近平面块中相邻的三角形的法向量之间夹角 a_d 必须小于 ϵ_d 。引入保细节是必要的,否则角特性值小于 2ϵ 的尖锐边将会被吞没。

4.3 近平面块形成算法——传播法

基本思想是任选一个三角形作为传播源,然后通过它的边,考查它的邻居三角形。如果满足扩展条件,就作为新源向外扩展。不能扩展的边即为近平面块的边。

近平面块形成算法:设 temp1 存放和近平面块相邻的正在处理的一层三角形, temp2 存放下一层三角形, pbset 存放近平面块的边。从③执行起:

①若 temp1 不空,从中取出一个未测试过的三角形,作为传播源, goto④;

②若 temp2 不空,内容转存 temp1,修改近平面块平均法向量 N , goto①,否则若 pbset 不空,一条近平面块的边界已得到,转入边界跟踪,删除共线点,最后形成近平面块边界,保存并清理有关数据;

③若三角形表不空,从中取一个未测试过的三角形作为传播源,否则 stop;

④若三角形的三条边都已测试过 goto①,否则取一条边作为传播边;

⑤若传播边为复杂边,边界边,记入 pbset 表中, goto④;

⑥若不满足保真性或保细节判据,传播边记入 pbset 表中, goto①;

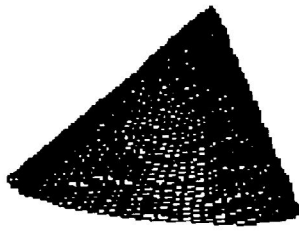
⑦把传播边有关的三角形记到 temp2, goto④。

算法中边界跟踪的目的是把近平面块边表以实际次序排列起来。通过上述算法很容易形成单个的封闭的近平面块边界,并且不包含蜕化的三角形。算法时间复杂度 $O(2n)$,其中 n 是三角形的个数。

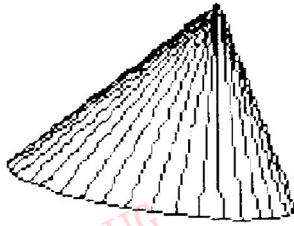
5 近平面块三角形的重新划分

如何保持重要的几何和拓扑特性已经在近平面块形成过程中考虑过了,所以本文仅以 aspect ratio (记作 AR) 作为三角形重新划分的标准。Hinker^[2] 介绍了 3 种三角形重划分算法,它选择了周游点算法,而 Schroeder^[1] 提出了 split 算法,下面简介这两种算法:

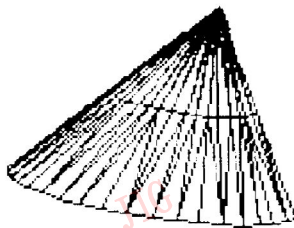
(1) Split 算法定义 AR 为 loop 的格点(劈分线的端点除外)到劈分平面的距离中最小的距离除以



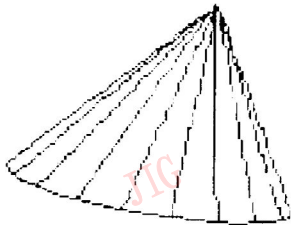
a. 原图
a. original.



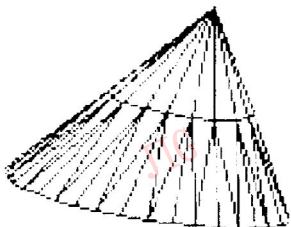
b. e 的平面块
b. planar set of e.



c. 简化了 91%
c. reduct 91%.



d. e 的平面块
d. planar set of e.

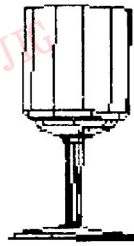


e. 简化了 95%
e. reduct 95%.

图 1 圆锥
Fig. 1 corn.



a. 原图
a. original.



b. e 的平面块
b. planar set of e.



c. 简化了 62.5%
c. reduct 62.5%.



d. 图 2e 的平面块
d. planar set of e.

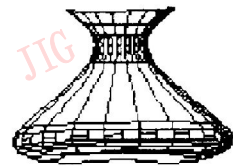


e. 简化了 70%
e. reduct 70%.

图 2 杯子
Fig. 2 cup.



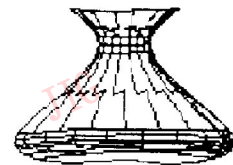
a. 原图
a. original.



b. e 的平面块
b. planar set of e.



e. 简化了 55%
e. reduct 55%.



d. e 的平面块
d. planar set of e.



e. 简化了 61%
e. reduct 61%.

图 3 瓶子
Fig. 3 bottle.

劈分线的长度。选择 AR 最大的劈分平面为最佳劈分平面。这个算法计算较繁,区域重新三角形划分效果不错,但是如果区域的形状比较复杂时,往往会出现劈分错误。

(2)周游点算法中构造三角形的起点不是固定的。算法产生的三角形形状虽然不一定很理想,但算法简单,速度快,适应面较宽。所以本文打算在此算法基础上,作一些改进。

(3)算法简介。本文的 AR 定义是:三角形高和底的比,本算法考查候选三角形和其邻居三角形,按 AR 最大的三角形进行划分,所以效果比周游点算法好。算法步骤:①在多边形上任取一点 i 作为起点 s ; ②若 s 点是凹点, $s=s+1$; ③检查三角形 1 ($s-2, s-1, s$), 三角形 2 ($s-1, s, s+1$) 及三角形 3 ($s, s+1, s+2$) 的 AR 值, 令带凹点的三角形的 AR 值为 0。④若三角形 2 的 AR 是最大, 记录三角形 2, $s=s+2$, 删除 s 点, goto ②, 否则 $s=s+1$ goto ②。

6 实验

本算法选择了 3 个图形:分别为圆锥 corn、杯子 cup 和瓶子 bottle, 目的要观察近平面块形成的情况图形简化的效果。图 1, 2, 3 的 a 图是它们的原始图, 分别包含 1441 个、1800 个以及 2772 个点; b 图和 d

图是原始图简化时,近平面块形成的情况图;以及 c 图和 e 图是相应的近平面块三角形重划分后的简化图。从结果看,本文算法形成的近平面块比较规整,简化结果能较好的模拟原图。

7 结论

采用快速形成近平面块方法的 3-D 图形简化算法,证明具有较好的效果,算法规整、速度快、保真和保细节效果均较好(可控)、适应面宽、有实用价值。

参考文献

- 1 William J S, Jonathan A Z, William E L. Decimation of triangle meshes, Computer Graphics, 1992, 26(2): 65-69.
- 2 Paul H Charles H. Geometric optimization. Computer Graphics, SIGGRAPH 1994.
- 3 Greg T. Re-tiling polygonal surface, Computer Graphics, 1992, 26(2), 55-64.
- 4 Hoppe H, Deroose T, Duchamp T, et al. Mesh optimization, computer graphics, SIGGRAPH 1994.
- 5 Clark J H. Hierarchical geometric models for visible surface algorithms, Comm ACM, 1976, 19(10): 547-554.
- 6 Chen Limin, Junhson O, Huang S. A surface reduction algorithm which combines speed and fidelity, IEEE JamCon95, 1995, 26: 452-456.



陈礼民: 山西大学教授, 计算机应用硕士生导师。1961 年毕业于南开大学。目前从事三维图形与图象处理教学和研究工作。

1983. 10—1985. 10 在美国南卡大学计科系从事图象处理的学习和研究工作。1994. 6—1995. 8 在美国休斯顿大学高性能计算中心, 进行三维图形学方面的研究工作, 在国内外发表论文多篇。

3-D Surface Reduction New Algorithm

Chen Limin

(Computer Center of Shanxi University Computer,
Taiyuan 030006)

Qin Aihong

(Science Dept. of Shanxi University,
Taiyuan 030006)

Abstract The 3-d surface reduction new Algorithm in this paper has advantage of keeping detail, fidelity and fast speed. For speed, the Source Propagation Method is suggested to form near planar set quickly. The characteristics of keeping detail and fidelity can be controlled.

This Method is normal and unified, it can get wide applications and good result. So it is of great value to use.

Keywords Graphical simplification, Computer graphics, Polygon reduction, Visualization