

文章编号: 1001-2486 (2000) 02-0082-04

一种适用任意平面多边形的三角剖分算法*

徐春蕾, 李思昆

(国防科学技术大学计算机学院, 湖南 长沙 410073)

摘要: 针对基于凹凸顶点判定的三角剖分算法适用范围有限的缺点, 提出了将凹凸顶点判定与连接多边形内外边界相结合的适用任意平面多边形的三角剖分算法 GTP (General Triangulation of Polygons)。GTP 计算速度快、适用范围广的良好特点已在应用中得到证实。

关键词: 三角剖分; 多边形; 凹凸顶点

中国分类号: TP 391.41 **文献标识码:** A

A Triangulation Algorithm for General Plane Polygon

XU Chun-lei, LI Si-kun

(College of Computer, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: Considering the triangulation algorithm based only on the determination of convex-concave vertices is restricted to simple polygons, this paper presents a triangulation algorithm for the general plane polygon GTP (General Triangulation of Polygons) based on determination of convex-concave vertices and connecting the outer border of a polygon with its inner borders. GTP is rapid and can be applied to any plane polygon. GTP has been implemented and acquired good effect in application.

Key words: triangulation; polygon; convex-concave vertices

平面多边形的三角剖分就是将平面多边形划分成一系列不相重叠的三角形且不产生新的顶点^[1]。在计算机图形学、图像处理、机械仿真等领域, 经常要解决平面多边形的三角剖分问题。一方面三角剖分解决了几何数据存储一致性; 另一方面三角形是最简单的多边形并有许多好的几何特性, 是解决许多问题的基础, 处理上也比较简单。

三角剖分产生的三角形集中任意两个三角形的公共边称为内边。以一条内边为公共边的两个三角形, 如果其中每个三角形的不在该内边上的顶点在另一三角形外接圆上或外部, 则称该内边是局部优化的^[2]。一个三角形内切圆与外接圆的面积比称作它的形态比。所有内边都是局部优化的三角剖分结果, 有最小内角最大和平均形态比最大的性质, 是最优的结果。在不必寻求三角剖分最优解时, 通常采用基于凹凸顶点判定的方法, 这种方法简单有效, 但只适用于简单多边形。将基于凹凸顶点判定的方法与 Delaunay 算法结合可以得到最优解, 但只用于简单多边形。在某些环境(如某些机械仿真系统)中往往存在着带孔几何模型, 它们的上下底面是非简单多边形, 此时以上两种算法将无法适用。

本文提出的 GTP 算法将凹凸顶点判定与连接内外边界相结合, 解决了任意平面多边形的三角剖分问题, 且效率较高; 虽然 GTP 的结果不是最优的, 但它很容易与 Delaunay 算法相结合得到最优解。GTP 的思想是对简单多边形直接采用基于凹凸顶点判定的方法; 对非简单多边形, 先用添加线段的方法把它的各个内边界依次和外边界相连, 得到一个简单多边形, 再采用基于凹凸顶点判定的方法进行三角剖分。将一个内环与外环相连时, 首先从内环顶点循环链表与外环顶点循环链表各取一个顶点连成一条线段。搜索外环的边, 如果与该线段相交, 用交点替换线段上的外顶点; 搜索内环的边, 如果与该线段相交, 用交点替换线段上的内顶点。最后, 将内顶点与外顶点按顺序分别插入内、外环顶点链表; 分别复制内、外顶点结点并插入被复制结点的后面; 从两个内、外顶点结点之间将两个循环链表断开并连接成

* 收稿日期: 1999-10-11

基金项目: 国家“863”计划项目(863-511-42-01)

作者简介: 徐春蕾(1976-), 女, 国防科技大学计算机学院硕士生。

一条新的外环链表。

1 算法描述

1.1 基本概念与计算

定义 1: 设 n^1, n^2, \dots, n^k 是多边形 P 的 k 个沿逆时针方向排列的顶点(如图 1), 对 n_i 来说, 若 $n_{i-1}n_i$ 绕 n_i 顺时针旋转到 $n_i n_{i+1}$ 所在直线上时绕过的角度大于 π 则 n_i 是凹顶点, 否则为凸顶点。

基本计算(1): 设多边形顶点是逆时针方向排列, 多边形的法向量是 n , 顶点 n_i 的前后顶点是 n^{i-1} 与 n^{i+1} , 求 n_i 的凹凸性。

令 $u = \overrightarrow{n^{i-1}n_i} \times \overrightarrow{n_i n^{i+1}}$, 若 $n \cdot u > 0$, 顶点 n_i 是凸顶点, 否则它是凹顶点。

基本计算(2): 有两条线段 ab 和 cd , 求它们是否相交。令

$u = (a - c) \times (d - c), v = (b - c) \times (d - c); p = (c - a) \times (b - a), q = (d - a) \times (b - a)$ 。若 $u \cdot v > 0$ 且 $p \cdot q < 0$, ab 与 cd 相交, 否则不相交。

1.2 算法基本步骤

GTP 算法通过添加线段把非简单多边形的内、外环连接, 生成与该多边形所占区域相同的简单多边形, 然后用判定顶点凹凸性的方法对该简单多边形进行三角剖分。算法 1 描述了本算法的基本步骤。设外环链表是逆时针方向, 内环链表是顺时针方向。

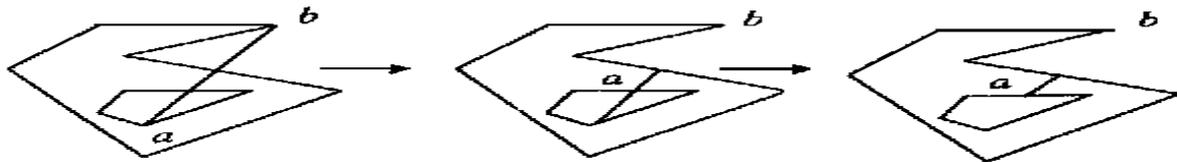


图 2 内外环连接线的计算过程

Fig. 2 the procedure to find the line segment connecting the inner and the outer borders

算法(1):

步骤 1: 如果内环链表不为空, 将第一个内环作为当前内环; 否则转步骤 6。

步骤 2: 从外环顶点链表与当前内环顶点链表中分别取一结点作为当前的内环顶点和外环顶点 a 、 b , 它们组成了当前连线 ab ; 将外环与所有内环上(除以 a 或 b 为端点)的所有边标记为待检测的。

步骤 3: 对外环上每个待检测的边, 计算它与 ab 的交点(见算法 2); 如果交点存在, 把交点作为当前的外环顶点 b 并去掉该边待检测的标记。

步骤 4: 对每个内环上所有待检测的边, 计算它与 ab 的交点(见算法 2); 如果交点存在, 把该内环记为当前内环, 把交点记为当前内环顶点 a 并去掉该边待检测的标记。

步骤 5: 如果 a 不是当前内环上的顶点, 把它按顺序插入当前内环顶点链表; 如果 b 不是外环上的顶点, 把它按顺序插入外环顶点链表。复制 a 得到 a' 并插入 a' 后, 复制 b 得到 b' 并插入 b' 后; 打开 a' 与 a , b' 与 b 的连接, 把 a' 作为 b' 的下一结点, b' 作为 a' 的下一结点, 组成新的外环顶点链表。转步骤 1。

步骤 6: 如果外环顶点链表的结点数大于 3, 把其中任意一结点作为当前结点; 否则把三个结点构成一个三角形, 结束。

步骤 7: 根据基本计算 1 计算当前结点的凹凸性, 如果由当前结点与它前后结点组成的三角形不包含多边形上的其它结点, 记录该三角形, 从链表删除当前结点并把它的下一结点作为当前结点; 否则把下一结点作为当前结点重复步骤 7。

步骤 8: 若链表的结点数大于 3, 转 7; 否则把最后三个结点构成一个三角形, 结束。

图 2 是只有一个内环时线段 ab 的计算过程。

设多边形的单位法向量是 n , cd 是环上待检测的边。

算法(2) (见图 3):

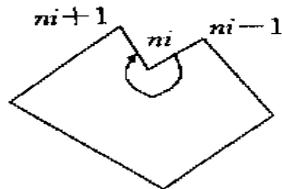


图 1 凹凸顶点示意图

Fig. 1 illustrating the convex and the concave vertices

步骤 1: 根据基本计算 2 计算 ab 、 cd 是否相交, 若不相交, 结束。

步骤 2: 计算在 多边形所在平面且与 ab 垂直的方向向量 $v = ab \times n$, 它的长度是 ab 。

步骤 3: 计算 ab 、 cd 交点 $r = c \cdot ab \cdot v / (ad \cdot v + ac \cdot v) + d \cdot ac \cdot v / (ad \cdot v + ac \cdot v)$ 。

ad 在 v 方向上的投影长度是 $ad \cdot v / ab$, ac 在 v 方向上的投影长度是 $ac \cdot v / ab$,

$$\begin{aligned} r &= c \cdot (ad \cdot v / ab) / (ad \cdot v / ab \\ &\quad + ac \cdot v / ab) + d \cdot (ac \cdot v / ab) / (ad \cdot v / ab + ac \cdot v / ab) \\ &= c \cdot ad \cdot v / (ad \cdot v + ac \cdot v) + d \cdot ac \cdot v / (ad \cdot v + ac \cdot v)。 \end{aligned}$$

2 数据结构与实例

2.1 数据结构

· 顶点

```
struct point{
    float x; // x 轴坐标
    float y; // y 轴坐标
    float z; // z 轴坐标
}
```

· 顶点链表的结点

```
struct node{
    struct point v_ node; // 对应顶点
    struct node * last; // 上一个顶点
    struct node * next; // 下一个顶点
}
```

· 顶点链表

```
struct loop{
    struct node * head; // 链表的头指针
    struct loop * next; // 下一个顶点链表
}
```

· 多边形

```
struct polygon{
    struct loop * out; // 外环链表
    struct loop * in; // 内环链表
}
```

2.2 实例及算法比较

· 实例

GTP 算法可三角剖分任意多边形并得到正确结果, 但结果三角形比较狭长。图 4 是两个非简单多边形的三角剖分结果, 第一个是内部有一圆环的圆, 第二个是内部有两个四边形环的六边形。

· 算法比较

下面对基于凹凸顶点判定的算法, 基于凹凸顶点判定的 Delaunay 算法, GTP 算法进行速度、适用范围、结果优化程度三个方面的比较, 因除 GTP 外的其它两种算法不能用于非简单多边形, 所以速度比较是针对简单多边形的, 如表 1。

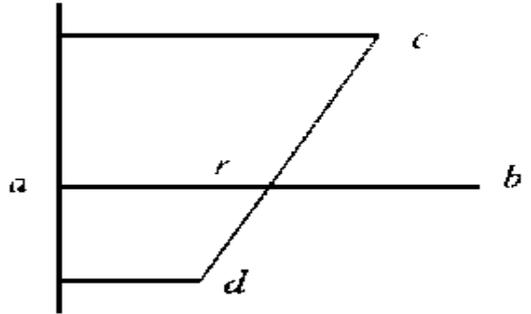


图 3 两线段的相交判定

Fig. 3 judging whether two line segments intersect

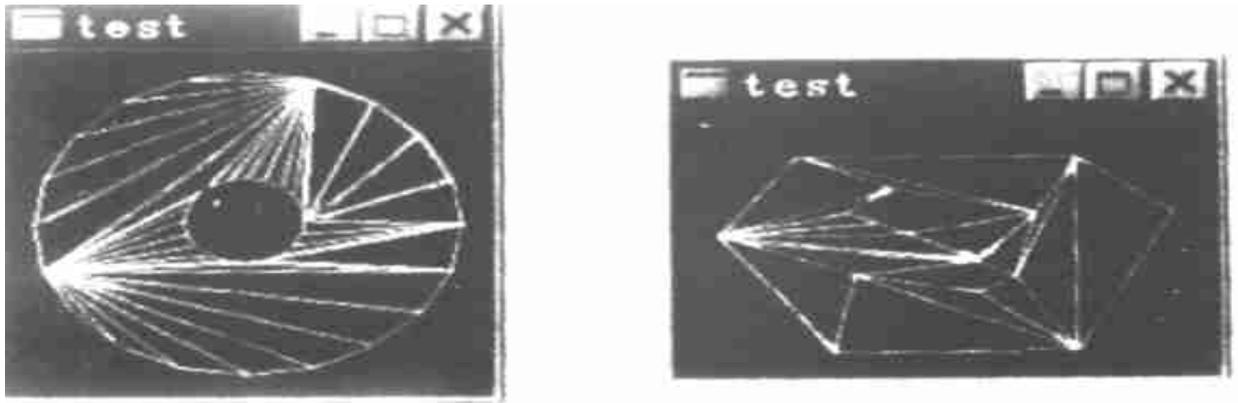


图 4 用 GTP 进行三角剖分的结果示例

Fig. 4 the results of triangulation using GTP

表 1 几种三角剖分算法的比较

Tab.1 comparing a few algorithms for triangulation

	基于凹凸顶点判定	Delaunay	GTP
速度	快	较慢	快
适用范围	简单多边形	简单多边形	任意多边形
结果优化程度	没有优化	最优解	没有优化

由表 1 可以看出, GTP 算法在适用性上是其它两种算法无法比拟的, 三角剖分简单多边形的速度与基于凹凸顶点判定算法相同, 但结果不是最优解。

3 结束语

提出的 GTP 算法将基于凹凸顶点判定的简单多边形三角剖分算法与连接多边形内外边界相结合, 适用于任意形状的平面多边形, 突破了仅依赖凹凸顶点判定的三角剖分算法只适用简单多边形的局限性, 且实现简单, 剖分速度快。GTP 算法已在有作者参加的虚拟原型仿真系统中得到应用, 并取得令人满意的效果。我们打算在以后工作中针对 GTP 的结果三角形比较狭长的缺点, 把 Delaunay 三角剖分思想补充到 GTP 中以求得任意平面多边形三角剖分的最优解。

参考文献:

- [1] 马小虎, 潘志庚, 石教英. 基于凹凸顶点判定的简单多边形 Delaunay 三角剖分[J]. 计算机辅助设计与图形学学报, Jan 1999
- [2] 闵卫东, 唐泽圣. 二维任意域内点集的 Delaunay 三角划分的研究[J]. 计算机学报, 1995, 18(5).
- [3] Turk G. Re-tiling polygonal surface[C]. Computer Graphics, 1992, 26(2).
- [4] Schrosder W J, Zarge J A, et al. Decimation of triangle meshes[C]. Computer Graphics, 1992, 26(2).
- [5] Hamann B. A data reduction scheme for triangulated surfaces[C]. CAGD, 1994, (11).
- [6] 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[J]. 软件学报, 1996, 7(9).
- [7] Kong X S, Everett H, Toussaint G T. The Graham scan triangulates simple polygons[C]. Pattern Recognition Letters, 1990, 11.