第8卷(A版) 第4期
2003年4月
中国图象图形学报
Journal of Image and Graphics
Vol. 8(A),No. 4
Apr. 2003

基于环链的多面体剖分快速算法研究

马　泳　刘文予
(华中科技大学电子与信息工程系,武汉 430074)

摘　要　利用环链提出了一种对任意多面体不添加顶点的凸剖分快速方法,它对多面体的剖分个数接近最少.该方法首先从多面体的棱和对角棱所构成的所有环中,以最小周长选取一个最好的环,然后利用这个环的各个边所形成的一系列面,对多面体进行一次剖分.实验证明,这种方法可找到对多面体不添加顶点剖分的最好剖分面,使剖分的次数接近最少,具有较好的实用价值和广泛的应用前景.

关键词　计算机图形学(520·6030)　快速算法　多面体　凸剖分　环链

中图法分类号:TP391.41　　**文献标识码**:A　　**文章编号**:1006-8961(2003)04-0459-05

A Fast Algorithm of Decomposing Polyhedrons by Using Loop-chain

MA Yong,LIU Wen-yu
(*Dept. of Electronics and information*, *HuaZhong University of Science and Technology*, *Wuhan* 430074)

Abstract　Based on loop-chain, a fast algorithm is presented for decomposing arbitrary polyhedron into convex polyhedrons without adding new vertexes, in which the final number of convex polyhedrons is close to the least. A loop is consist of several vertices that start from one vertex go to next adjacent vertex can come back this vertex, all vertices in a loop is consist of a partition-plane. The least perimeter loop-chain is selected from all the loops consisted of edges and diagonal-edges of the polyhedron, and decomposes the polyhedron with a series of planes consisted of all edges of the loop. Experiments show that this method can decomposing arbitrary polyhedron into the least number convex polyhedrons without adding new vertexes, can process most kinds of polyhedrons (e.g. a polyhedron with a inner hole), and has low cost in calculation. The algorithm of convex decomposing 3D objects has been a research direction of computer geometry for a period and is widely used in fields of pattern identifying, animation and CAD.

Keywords　Computer graphics, Fast algorithm, Polyhedron, Convex decomposition, Loop-chain

0　引　言

三维物体的凸剖分广泛应用于模式识别、图形动画、计算机辅助设计等领域.多边形三角剖分是剖分的一个特例,国际上三角剖分的研究在20世纪70年代后期有较大的发展,其间令人瞩目的成果是两个三角剖分算法——贪心算法GT和Delaunay算法DT的出现[1,2].另外,还有一个时间复杂度为$O(n\log n)$的剖分算法[3];简单多边形三角剖分的线性时间算法[4];Keil利用动态编程技巧,提出了一种将简单多边形分解成凸多边形、螺旋型多边形、星状多边形以及单调多边形的渐进时间算法,同时证明了带有内孔的任意多边形的分解是一个NP-hard问题[5];陈向平等通过"桥边"将带有内孔的任意多边形统一转化为非自交多边形,然后再进行非自交多边形的三角剖分[6].然而,关于三维多面体的凸剖分,一直没看到非常有效的算法,但是三维凸剖分是研究许多问题的前提,因为一个凸的图形较其他图形,在计算机表示、分析及处理方面方便得多.

本文提出了一种对三维多面体凸剖分的快速算法,由于多面体凸剖分是一个NP-hard问题,因此寻找快速算法是唯一途径.广义形态内插的方法[7],其广义形态变换中要求任意多面体剖分后的凸体个数最少.而本算法能对任意多面体不添加顶点进行凸剖分,且它对多面体剖分后的凸多面体个数接近

收稿日期:2002-08-13;改回日期:2002-12-06

最少,同时该算法对各种多面体都能剖分,是一种有效的方法.

1 基本概念与定义

定义 1 设 P_1P_2 是空间任意多面体中由平面 α 和 β 相交形成的一条棱,若平面 α 和 β 在多面体内部所夹的角小于或等于 π,则称棱 P_1P_2 为凸棱,否则称为凹棱.

定义 2 设多面体 A 的某个顶点 $\boldsymbol{p}$ 与 k 个面相邻,每个面的法线矢量为 $\boldsymbol{N}_1,\boldsymbol{N}_2,\cdots,\boldsymbol{N}_k$,则顶点 $\boldsymbol{p}$ 的局部方向矢量定义为 $\boldsymbol{N}_p=\frac{1}{k}\sum_{i=1}^{k}\boldsymbol{N}_i$,当且仅当 $\boldsymbol{p}$ 点存在一个局部半球与半空间 $\{\boldsymbol{x}:(\boldsymbol{x}-\boldsymbol{p})\cdot\boldsymbol{N}>0\}$ 的交在 A 的外部,则称此点为凸点,否则为凹点.

显然,凹点一定在凹棱上,凹棱上至少有一个顶点为凹点,凹棱上的另一个顶点可能为凸点.如果一个多面体每一条棱都是凸棱,那么称此多面体为凸多面体,否则,称之为凹多面体.

定义 3 多面体的顶点中,凡有边相连的两个顶点互称为相接点.对多面体的每一个表面,它上面的每一个顶点互称为共表面点.

定义 4 多面体棱 l 的两个顶点分别为 A、B,那么 A 所有共表面点称为 A 的可链点.A 的可链点与 A 相连所得的线如果不是多面体的棱,则称其为多面体的对角棱,如图 1 所示,连接 A 与它的可链点 D,AD 就是多面体的一条对角棱.

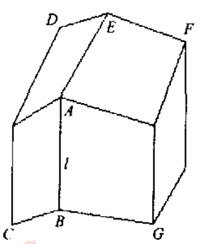


图 1 AD 为一条对角棱,D 为 A 的可链点

定义 5 从多面体的一顶点 A_1 出发,寻找 A_1 的可链点 A_2,再寻找 A_2 的可链点 A_3,依次寻找下一个的可链点,最后可回到 A_1,这一系列的边称为多面体的一条环链,这条环链是由多面体的棱和对角棱组成.

在这样所有的环链中,可能存在这样的环链,这些点所围成的面是多面体的表面,这样的环链被认为是不成功的环链.如果环链上的点围成的面不全部在多面体的表面,则称这样的环链为成功的环链.注意,环链不一定在同一平面上,它是三维的.图 2 中,可以找到 3 条成功的环链,环链的顶点分别为 166′1′,544′5′和 788′7′.

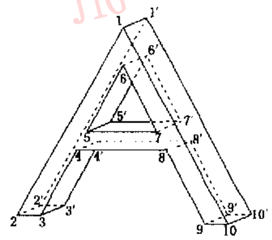


图 2 凹多面体的 3 条成功环链

引入环链的定义,是因为假如不添加顶点,那么对多面体进行剖分,剖分结果就会在多面体表面留下一个成功环链,这个环链构成的一系列折面可把多面体一分为二.假如得到了所有成功的环链,那么实质上即得到了这个多面体的所有可能剖分结果,然后可按某种要求选取其中的某些环链实现对多面体的凸剖分.

定理 1(两平面相交棱凹凸性判定定理)[8] 设 P_1P_2 是由平面 α 和 β 相交而成的棱,P_3、P_4 分别为 α 和 β 上的点,$P_1\to P_2\to P_3$ 所环绕的方向和 α 指向多面体外部法线符合右手规则,并且 $P_1\to P_2\to P_4$ 所环绕的方向和 β 指向多面体内部法线符合右手规则,P_i 的坐标分别为 (x_i,y_i),其中 $i=1$、2、3 、4,若

$$\boldsymbol{D}=\det(P_1,P_2,P_3,P_4)=\begin{vmatrix}x_1 & y_1 & z_1 & 1\\ x_2 & y_2 & z_2 & 1\\ x_3 & y_3 & z_3 & 1\\ x_4 & y_4 & z_4 & 1\end{vmatrix}\leq 0 \tag{1}$$

则棱 P_1P_2 为凸棱,否则为凹棱.

因为判断边的凸凹性非常重要,为提高运算速度,每一多面体专门建立一个链表,记录每条边的凸凹性,这个链表称为凸凹棱表.凸凹棱表中只记录凹棱的边号,不记录凸边,凸边可以在此多面体的边表中找到.

2 快速算法原理

显然对多面体进行剖分的快速算法关键在于快速找到一条符合条件的成功环链,因此,重点在于寻找成功的环链.不同的剖分要求会造成不同的成功

环链，剖分后的凸多面体的个数最少这一要求非常高，所以可寻找一种近似解法，即剖分后的凸多面体的个数接近最少，这样就可以大大提高计算速度.

2.1　求成功环链的快速算法

每个面中相邻的两点都构成一条棱，由定理1可得到所有棱的凸凹性，显然对多面体的凸剖分从一条凹棱开始较有利，接着找出这条凹棱所在的所有成功环链. 在这个问题中，需用到 N 叉树. 假定凹棱 l 的两个顶点为 A、B，从 A 点出发，那么 A 就构成了 N 叉树的根节点，A 的可链点构成了 N 叉树的第1层节点，第1层每个节点的可链点构成了它们的下一层，依此下去，直到其节点为 B 为止，这样就构成了一个 N 叉树. N 叉树的每一个分支就是经过 l 的一条环链，但在这个过程中，要剔除那些不成功的环链，因为不成功环链是环链上所有顶点围成的面为多面体表面的环链，其方法是看环链上相邻点所构成的面是否全部在多面体表面上，是，则剔除. 在 N 叉树的生成过程中，要注意，新生成的节点不应该是其分支上已有的点，因为这样会形成一个环路，而不构成树.

但是这个方法存在一个弊端，即内存花费太大，时间也过长，其原因是要建立环链的 N 叉树，并对 N 叉树进行遍历. 如果多面体有 N 个顶点，则剖分最坏的时间复杂度为 $O(N!)$. 改进的方法是对 N 叉树进行剪枝，实际上只需要记录 N 叉树的一个枝，即环链的顶点序列，这个顶点序列是 N 叉树中从根到叶子的一条最短路径. 采用深度优先搜索(DFS)策略，即最快地找出一条近似最短环链，然后再找出其他最短环链比较长度，求出较好的剖分环链.

选取环链的标准是环链所包含的面积最小，因为环链包含的面积最小意味着是一个较好的剖分，但面积计算复杂，因此选取环链的周长最小、即在选取环链时应选最短距离的可链点.

根据凹点的性质，如果环链上的所有顶点都是凹点，并且此环链不与多面体上其他边相交，则此环链是一个最好的剖分面，推广之，如果环链上的点大部分是凹点，则此环链是一个较好的剖分面. 因此，找下一个可链点时应优先选择凹点，如果找不到凹点，再找距离最近的凸点.

算法1　求成功环链的快速算法

(1) $k=0$，$minlength=10000$.

(2) 设环链的起点为 s，s 为凹棱的起点，环链的终点为 e，e 为此凹棱的终点，令 $length=1$，$loop=s$，$loop$ 为一字符串.

(3) 找 s 的最短距离凹可链点，如果找不到，则找 s 的最短距离凸可链点，设找到的可链点为 p，把边 sp 打上标记，$loop=loop+p$，$length=length+1$，$s=p$.

(4) 如果 $s=e$，则 $loop$ 是一个成功的环链，否则转第3步.

(5) 如果 $length<minlength$，则 $minlength=length$，$newloop[k]=loop$.

(6) $k=k+1$，if $k<T$ 转第2步，否则结束.

算法中的第3步是求成功环链的核心，第5步是找出顶点最少的 T 个成功环链，在 T 个成功环链中寻找环链所包含的面积最小的环链，这个面积最小的环链即为最好凸剖分环链，由这个环链对多面体进行一次凸剖分，此种方法既考虑了处理速度，也考虑了剖分后的最优结果，是一种综合较优的方法. 算法中 T 是一个阈值，由用户给定，通常 $T=5$，显然 T 越大，时间开销越大，$T=N$，则为 N 叉树搜索；$N=1$，则假定一次搜索找到的环链即为最好环链，此时时间开销最小，$minlength$ 的初值设为10000是因为实际中没有多于10000个点的环链. 显然，找成功环链的时间复杂度最好为 $O(N)$，通常情况下会接近 $O(N)$. 算法的核心为找下一个可链点时优先选择凹点，如果找不到凹点，则找距离最近的凸点，显然此种方法不能保证最优，但时间开销小，是一种实用算法.

2.2　用成功环链进行多面体的凸剖分

剖分完毕后，需要修改新多面体的边和面，对新剖分的边要用定理1重新判断凸凹性，继续剖分下去，直至所有的多面体都为凸.

算法2　多面体的凸剖分算法

(1) 对多面体的所有边判断其凸凹性，建立凸凹棱表.

(2) 从某一凹棱的起点开始，用算法1找出一个成功的环链.

(3) 用此环链对多面体进行凸剖分，判断剖分的结果是一个多面体还是两个多面体，并修改剖分后多面体的边、面和凸凹棱表的数据结构.

(4) 对剖分后新增的边，判断其凸凹性，把凹边加入到凸凹棱表中.

(5) 判断剖分后新的多面体的凸凹棱表是否为空，不为空则转2步.

(6) 如果所有多面体的凸凹棱表都为空，则算法结束，否则，从某一多面体的凸凹棱表中选出一凹

棱,转第2步.

显然,用此算法进行凸剖分产生的结果并不是凸多面体个数最少,而只是一种近似算法,但近似程度很好,用近似算法可以极大地降低时间复杂度.

由于找成功环链的时间复杂度为$O(N)$,凹棱的个数为$O(N)$,因此算法2的时间复杂度为$O(N^2)$,其中N为多面体的顶点数.

3 剖分后如何创建新的体

环链得到后,用所得环链将多面体剖分,剖分的结果有两种情况:一是剖分完后,还是一个多面体;二是将原多面体剖分为两个多面体.对于这两种情况要分别对待.

3.1 判断剖分结果是一个体还是两个体

如果原多面体是凹体,则一次剖分后可能还是一个体.

判断的基本思想是:首先将环链点所在的面分开,接着从中选一边的点;然后从这些点开始,寻找与这些点能连起来的所有点,最后看所得到点的数目是否比原多面体点数少,如果少,则说明剖分将原多面体一分为二,否则说明剖分结果还是一个体.

算法3 判断剖分结果是一个体或两个体

(1) 将所选出环链上的点加入数组 **W**;

(2) $i=0$;

(3) 判断i是否小于 **W** 的个数,小于,则取点$W[i]$,否则转第9步;

(4) 得到$W[i]$所在的一个面;

(5) 判断这个面是否已经判断过.已判断,则跳到第7步,否则继续;

(6) 求出这个面上所含环链点的个数n;

若$n=0$,则将这个面上所有的点都加入数组 **W**;

若$n=1$,如果这个面所含环链的点不是$W[i]$,则看这个面是否有点在数组 **W** 中,有,则将这个面中的点加入数组 **W**(注意不能重复),否则不加入.如果这个面所含环链的点正是 **W**,则将这个面标识为未判断,继续下一步.

若$n=2$,如图3所示,面是顺时针绕向,环的一条边$\overrightarrow{AB}$从这个面穿过,将这个面一分为二,则将边$\overrightarrow{AB}$左边的点加入数组 **W**;

(7) 得到$W[i]$的下一个面,转第5步.如果没有下一个面,则继续;

(8) $i++$,转第3步;

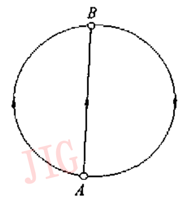


图3

(9) 判断 **W** 数组的个数是否小于原多面体点的个数,小于,则说明原多面体一分为二,否则还是一个体.

3.2 剖分后,原多面体还是一个多面体

在这种情况下,多面体上的有些点就要由原来的一个变成两个,即环链上的点就要一分为二,也就是说,原多面体增加了一些新点,把这些新点依次加在原多面体点数组的后面,构成了新多面体的点数组.进而要解决的问题是修改已有面,构造新的面.

要修改的面是环链上的点所在的面,因为这些面上的有些点可能要换成新ID.设这些面含有环链点的个数为n,n可能为1或者2.

当$n=2$时,如图3,边$\overrightarrow{AB}$将面分为左、右两个面,将右边面上的点A、B换成新ID.如果$\overrightarrow{AB}$的两边中有一边没有点,则说明那一边的面退化为一条边,将其删掉.

当$n=1$时,就比较麻烦,困难在于由一个点无法判断这个面是否在环链的右边,也就无法判断这个面上的环链中点是否应该换成新ID,如图4所示,解决方法如下,求出与环链$\cdots\to A\to B\to C\to\cdots$只有一个公共点$B$,且在环链右边的所有面$S_2$、$S_3$、$S_4\cdots$.

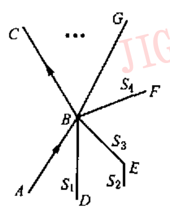


图4

算法4 构造新的面

(1) 求出对角棱AB所在的一个或者两个面,如果有两个面,则选出右边的面,这样就得到面S_1;

(2) 在面S_1中求出B的下一个点D;

(3) 求出B、D所在的两个面中非S_1的那个面S_2;

(4) 在面S_2中求出B的下一个点E;

(5) 这样依次循环下去,直到B的下一个点是C为止.

这样就得到了与环链只有一个公共点 B,且在环链右边的所有面 S_2、S_3、S_4…,将这些面上的 B 全部都换成新 ID.

3.3 剖分后,原多面体一分为二

这种情况下,点数组被分成了两个,原多面体中剩下的点再加上环链上的点就构成了另一个体的点数组.增加新面的方法与 3.2 节相同.

4 结　论

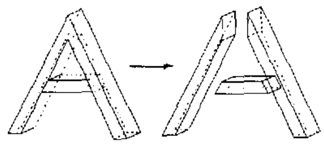


图 5　一个有孔的多面体的凸剖分

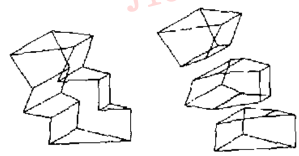
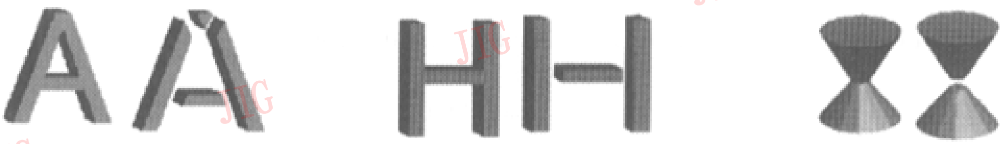


图 6　任意多面体的凸剖分结果

图 5 为对图 2 中的有孔多面体进行凸剖分的结果,图 2 中凹多面体可以找到 3 条成功的环链,其分别为 166′1′, 544′5′和 788′7′,剖分成 3 个凸多面体.图 6 为任意多面体的凸剖分结果,剖分成 3 个凸多面体.

图 7 是对各种多面体剖分的结果,其各项数据见表 1,实验在 PⅢ 460PC 机上进行,由表 1 可见,当多面体点数较少时,所花时间很少,剖分效果也很理想.



(a) 字母 A　　(b) 字母 H　　(c) 上、中、下面都是正十六边形

图 7　各种多面体剖分的结果

表 1 对多个多面体剖分结果的统计

图 7	点数	凹棱的条数	剖分次数	所花时间(ms)
(a)	21	5	4	280
(b)	24	4	2	65
(c)	48	16	1	110

本文提出了有孔多边形和有孔多面体的近似最优剖分快速算法,即凸剖分的多面体个数接近最少.对多面体,用深度优先搜索(DFS)策略找出一条近似最短环链,由最短环链构成的剖分面对多面体进行一次剖分,依次进行,直至所有的多面体都是凸体为止.这种凸剖分的时间算法复杂度为 $O(N^2)$,是一种高效的算法,其结果可以达到实用效果.

该算法的核心是找成功环链中的下一个可链点时应优先选择凹点,如果找不到凹点,则找距离最近的凸点,此种方法不能保证最优,并且对复杂的多面体产生的结果并不能达到剖分个数最少,因此寻找针对任意复杂的多面体快速的凸剖分算法仍然是一个具有十分挑战性的问题.

参 考 文 献

1 Lloyd EI. On triangulation of a set of points in plane[A]. In: proc. of 18th Annual Symposium on Foundations of Computer Science[C]. Providence,RI,USA, 1977:228～240.

2 Sibson R. Locally equiangular triangulation[J]. Computer Journal, 1997,21(3):243～245.

3 Avis D, Toussaint T. An efficient algorithm for decomposing a polygon into star-shaped polygon into star-shaped polygons[J]. Pattern Recognition, 1981,13(6):395～398.

4 Garey M R, Johnson D S, Preparata F P. Triangulating a simple polygon[J]. Inf. Proc. Lett, 1978,7:175～179.

5 Keil J M. Decomposing a polygon into simpler components[J]. SIAM J Computing. 1985,14(4):799～817.

6 陈向平,应道宁. 统一于 NIP 的多边形三角剖分算法[J]. 计算机学报,1989,12(3):194～199.

7 刘文予,朱光喜. 基于广义形态内插的非刚体运动描述方法[J]. 软件学报,2001,12(10):1544～1551.

8 徐明. 确定任意多面体的凸凹棱的快速算法及其应用[J]. 徐州师范大学学报(自然科学版),1999,17(2):21～23.

马　泳　1971 年生,1997 年获北京理工大学硕士学位,讲师,现为华中科技大学博士研究生.主要研究方向为图象处理和目标识别.

刘文予　1963 年生,博士,教授,博士生导师.主要研究方向为计算机图形学、计算机视觉、多媒体信息处理.