第6卷第11期
1995年11月

软 件 学 报
JOURNAL OF SOFTWARE

Vol. 6, No. 11
Nov. 1995

三维布局中八叉树节点的快速分解算法*

戴 佐 查建中 倪仲力

(天津大学机械系智能工程研究室,天津 300072)

摘要 本文在对八叉树子节点的顶点类型及构成三维实体精确CSG模型的半空间的类型进行系统地分类的基础上,结合不同类型半空间的性质给出了一个三维实体的精确CSG模型转换到八叉树模型的综合算法.通过对若干个不同的实体进行实际计算,结果表明该算法能够满足三维实体自动化布局的需要.

关键词 三维布局,八叉树,节点,快速分解,算法.

八叉树是表达三维实体模型或三维图象的一种重要方法,在计算机图形学、计算机视觉、图象处理、CAD/CAM、机器人手臂运动路线规划等方面获得了广泛的应用.根据八叉树应用的不同,八叉树本身具有很多变种,八叉树的构造算法也多种多样[1].

文献[2,3]成功地将八叉树数据结构引入到三维实体的计算机仿真布局及智能最优布局领域,用以表示任意形状的三维实体及三维布局空间.三维实体及空间的精确描述通常采用结构实体几何法(CSG)及边界描述法,而将物体的CSG描述转化为八叉树描述的算法,文献[1]已作了综述.但由于应用领域的不同,所有现有的CSG或B_rep表达到八叉树表达的转化算法并不适用于三维布局问题的需要,其主要不适应点有:

①三维实体布局中八叉树的节点采用动态变精度分解策略,即对于全满、全空节点,不分解;对于半满节点,根据其它附加的条件,可能分解,也可能不分解.即在布局中需要分解时才分解,而半满节点的分解需求随着分解精度、布局物体的形状、相互位置及具体应用的变化而变化,可以说是随机的.因此,布局当中八叉树的生成问题最终归结为将某一个半满节点分解,求其8个子节点的状态的问题.以往的转化算法采用的都是静态分解的策略,即按某种顺序将所有的半满节点一次性分解为全满、全空节点直至达到分解精度为止.

②由于B_rep表达方式是将所有的实体均近似为多面体加以表达,因此现有边界描述到八叉树的转化算法实质上是多面体的边界描述到八叉树的转化.同样,对于CSG而言,现有的所有算法都是针对凸多面体设计的,对于曲面体都是通过将其转化为凸多面体来完成的;对于布局问题而言,其所涉及的实体形状多种多样,其中绝大部分是曲面体,将这些曲面

* 本文1994-07-02收到,1994-10-18定稿

本研究得到国家自然科学基金的资助.作者**戴佐**,1968年生,讲师,主要研究领域为智能工程,CAD,布局.**查建中**,1947年生,教授,博士导师,主要研究领域为优化设计,智能工程,CIMS.**倪仲力**,1970年生,硕士生,主要研究领域为布局自动化.

本文通讯联系人:戴佐,天津300072,天津大学机械系智能工程研究室

体转化为凸多面体本身就是一个复杂的推理和计算过程,且转化后的凸多面体具有庞大的数据规模及有限的近似精度.因此,不可能也不必要将所有实体先转化为凸多面体近似表达,再将这个近似表达转化为八叉树.

针对上述情况,本文首先对构成八叉树某一节点的8个子节点的所有顶点进行分类,设计了检验部分半满子节点的27点法;然后对构成实体CSG表达的半空间进行分类,设计了检验全满及全空节点的半空间法;通过两个算法的有机结合,形成了一个面向任意形状实体的节点分解的快速算法.

1 三维实体的精确CSG表达

任意三维实体都是由一些基本形体构成的(不同层次的交、并、差),基本形体又可表示为许多半空间的交,而半空间通常可表示为$f(x,y,z)\geqslant 0$的形式,由此形成一个二叉——多叉树混合结构,叫作CSG精确代数模型,如图1所示.

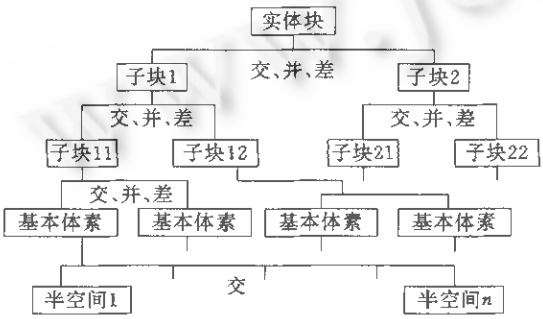

图1 三维实体的CSG代数模型

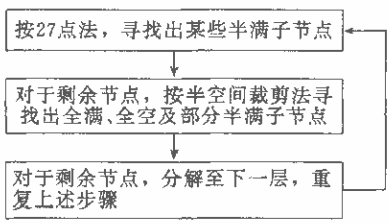

图2 总体流程图

2 八叉树简介

八叉树是从一个立方体开始,将其均分为8个子立方体,按照一定的分解规则,某些子立方体再均匀分为8个子立方体,如此递归分解下去,直到满足一定的条件为止.整个分解过程可用一个树状结构来表达,一个根节点有8个子节点,每个子节点又可有8个子节点.

根据应用背景不同,八叉树有许多表现形式.应用于三维布局中的八叉树结构定义如:每个立方体有3种状态:该立方体全部被布局对象填满、部分被布局对象填满、无布局对象在内.这3种状态分别对应着八叉树节点的全满、半满、全空状态.节点的分解原则为:若节点的状态为半满且没有达到分解精度,在布局过程需要时,分解至下一层节点,否则不分解.

3 节点快速分解算法总体介绍

由于三维实体之间的交、并、差计算直接对应于八叉树之间的交、并、差计算[4],因此本章重点讨论基本形体的八叉树节点分解算法.

基本形体由一组半空间(即三元不等式函数)的交构成,每一个半空间由一个不等式表示,因此设:$f1(x,y,z)\geqslant 0,\cdots,fn(x,y,z)\geqslant 0$为一基本体素;

$x1\leqslant x\leqslant x2$, $y1\leqslant y\leqslant y2$, $z1\leqslant z\leqslant z2$为待分解的半满节点的6个面,则算法总体流程如图2所示.

4　27 点法介绍

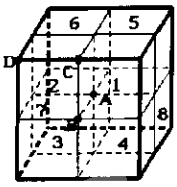

图 3 为八叉树的某一个待分解节点,其每一个子节点有 8 个顶点,八个子节点的所有顶点合并后共有 27 个不同的点.这 27 个点可分为如表 1 所示的 4 种类型:

表 1　八叉树的待分解节点

类型	示例	个数	涉及子节点数	在父节点中位置
Ⅰ	A 点	1	8	体中心
Ⅱ	B 点	6	4	面中心
Ⅲ	C 点	12	2	边中心
Ⅳ	D 点	8	1	顶点

由表 1 可以看出,Ⅰ类点是 8 个子节点的公共顶点,即它的状态可以被 8 个子节点所公用,利用效率最高;Ⅱ类点是 4 个子节点的共同点,所以利用效率次高,依此类推.又,对于一个子节点来说,如果它有一个顶点在实体内,另一个顶点在实体外,那么可以肯定地说,该子节点为半满节点.据此,设计出"27 点"算法(若点 M 在实体内,称其为黑点,否则为白点).

①查Ⅰ类点,若为黑点,则将八个子节点的颜色均置为黑,否则置为白(在这里,顶点的颜色与子节点的颜色是两个概念);

②按前、后、左、右、上、下的顺序查Ⅱ类点的颜色,若和与其相关的 4 个子节点的颜色相反(如 B 点对应于子节点 1,2,3,4),则该四个子节点状态为半满,否则继续.

③对于剩余的未检测出状态的子节点,查相应Ⅲ类点,若与子节点颜色相反,则为半满状态,否则继续.

④对于剩余子节点,查相应Ⅳ类点,若与子节点颜色相反,则为半满状态,否则输出剩余子节点.

实体由 N 个不等式组成,查一个点的颜色所需计算的方程个数在 1 到 N 之间;而对于最好的情况而言,仅查一个Ⅰ类点(是黑点)、两个Ⅱ类点(白点)即可判断出所有 8 个子节点均为半满状态,此时方程的最少计算次数为 $N+1+1=N+2$ 次;对于最坏情形而言,有可能计算出的所有 27 个点均为黑点,方程计算次数为 $27N$ 次,且此时只知道 8 个子节点中无全空节点存在,并不能判断出子节点的状态.由此可知,27 点法对于绝大多数半满子节点(即子节点的 8 个顶点中至少有一个的颜色和其余的不一样)具有很高的判断效率.若 8 个顶点的颜色都一样,同为黑或白,则该子节点的状态用 27 点法仍不能判明,我们暂时定该子节点的颜色为黑或白.但是,当基本体素为凸形体时,8 个顶点均为黑即表明该子节点为全满状态.证明如下:

8 个顶点同为黑表明它们是基本体素内或面上的点.求这 8 个点的三维凸壳,恰好为子节点所对应的子立方体.而点集的凸壳是包含点集的最小的凸形体,所有其它包含该点集的凸形体必包含凸壳,也就是说,子立方体(凸壳)内的所有点都在凸基本体素内,为全满状态.

在 27 点法结束后,将所有状态不明的子节点取出,组成剩余子节点集 L,用半空间裁剪法判明其最终状态为全空、全满还是半满.

5 半空间裁剪法介绍

文献[1]描述了一个计算凸多面体(即所有形成半空间的不等式为三元一次方程)的八叉树节点状态的半空间求交法,其基本思想是将每一个半空间看作一个封闭的三维实体,首先求出子节点在这些半空间中的状态,然后利用八叉树的求交算法算出所有这些半空间的交.在此基础上,本文设计了面向凸多面体及某一类曲面的半空间裁剪法.

设 L 表示 27 点法未判断出状态的子节点集,x 表示 L 中的一个元素;M 表示组成实体的半空间集(不等式集),b 为 M 的一个元素.

x 在第 i 个半空间 bi 中的状态有三种类型:全空,全满,半满.

对于全空状态,x 判定为全空,从 L 中裁减下去;

对于全满状态,x 的状态不变,保留在 L 中;

对于半满状态,分 3 种情况讨论:①若 x 的颜色为黑,x 判定为半满,从 L 中裁剪下去;②若 x 的颜色为白,且在前 $i-1$ 个半空间中有一个的状态为半满,保留在 L 中,则称其为半满不定状态,因为它既可能为半满,也可能为全空;③若 x 的颜色为白,且在前 $i-1$ 个半空间中的状态均为全满,则称 x 为暂时半满状态,保留在 L 中.

当 M 中的半空间均检查完后,L 中的全满状态即可最终判定为全满状态,暂时半满状态可判定为半满状态;而半满不定状态仍无法判断,需将其分解至下一层节点,继续调用 27 点法来确定.

实施上述算法的前提是计算某一个节点在半空间中的状态,文献[1]给出了平面半空间的最大值——最小值算法,本文将其推广到一类曲面中.

首先将不等式划分为 3 种类型.

①形如 $f(x,y,z)=ax+by+cz+d\geqslant 0$ 的不等式称为 I 类,其中,a、b、c、d 为常数,该方程代表坐标系中由一个平面分割出的一个半空间.

②形如 $f(x,y,z)=A(x-a)^{n1}+B(y-b)^{n2}+C(z-c)^{n3}+R\geqslant 0$ 的不等式称为 II 类,其中 A、B、C、R、a、b、c 为常数,$n1$,$n2$,$n3\geqslant 2$,该方程代表坐标系中处于某一特殊位置的某一曲面分割出的一个半空间.

③不能用上述两种形式表达的所有其它曲面,统称为 III 类不等式.

设 $x1<x<x2$,$y1<y<y2$,$z1<z<z2$ 为 L 中的当前子节点,$f(x,y,z)>0$ 为 M 中的半空间方程,现欲判断该子节点在半空间中的状态,基本思路如下:

若当前节点颜色为黑,则计算出 $f(x,y,z)$ 在域 $x1<x<x2$,$y1<y<y2$,$z1<z<z2$ 内的最小值 $V\min$,若 $V\min\geqslant 0$,则可判定在该半空间中为全满状态;若 $V\min<0$,则可判定当前节点的最终状态为半满状态;

若当前节点颜色为白,则先计算出 $V\max$,若 $V\max>0$,则可判定在该半空间中为半满状态;若 $V\max<=0$,则可判定当前节点的最终状态为全空状态;

对于 I 类半空间,$f(x,y,z)=ax+by+cz+d>0$,其最大最小值必定出现在 $x=(x1,x2)$,$y=(y1,y2)$,$z=(z1,z2)$ 上,所以

$$V\max=\max(ax1,ax2)+\max(by1,by2)+\max(cz1,cz2)+d;$$

$$V\min=\min(ax1,ax2)+\min(by1,by2)+\min(cz1,cz2)+d;$$

对于Ⅱ类半空间

$$f(x,y,z)=A(x-a)^{n1}+B(y-b)^{n2}+C(z-c)^{n3}+R\geqslant 0$$

设 OPT 代表最大或最小值,则

$$OPT(f(x,y,z))=OPT(A(x-a)^{n1})+OPT(B(y-b)^{n2})+OPT(C(z-c)^{n3})+D$$

而 $OPT(A(x-a)^{n1})$ 只可能出现在 $x1,a,x2$ 三个值上(如果 $x1<a<x2$)或 $x1,x2$ 上(如果 a 不在 $x1,x2$ 之间),所以

当 $x1<a<x2$ 时,$OPT(A(x-a)^{n1})=OPT(A(x1-a)^{n1},0,A(x2-a)^{n1})$;

否则,　　$OPT(A(x-a)^{n1})=OPT(A(x1-a)^{n1},A(x2-a)^{n1})$

对于Ⅲ类半空间,由于其最优值的计算困难,无法应用最大——最小值法,本文采用如下两种方案处理:

①在建立物体的 CSG 模型时,用Ⅰ类和Ⅱ类面来近似Ⅲ类面;

②若①不行,可将状态不明节点往下分解,然后递归地调用 27 点法和Ⅰ、Ⅱ类面的半空间法进行裁剪,直至达到某一精度为止.

6　三维实体八叉树表达的实验特性

为了检验上述算法的效率、准确性及八叉树表达的近似精度,本文按照深度优先法则完全分解一棵八叉树至预定精度,同时利用正方体轴侧投影的堆砌对三维实体进行造型显示,以此检验分解结果的正确性及有效性.

选取如下基本形体:

①圆锥　　$(x-64)^2-(y-128)^2+(z-64)^2\leqslant 0;y\leqslant 128;$

②球　　$(x-64)^2+(y-64)^2+(z-64)^2\leqslant 4096;$

③单叶双曲面　$(x-64)^2-0.64(y-64)^2+(z-64)^2\leqslant 576;0\leqslant y\leqslant 128;$

在 HP386QS/16 微机上对以上实验对象进行实验,得出数据示于表 2—4:

表 2

分解	圆锥					平均　体积
层数	外型	分解节点数	体积下界	分解时间	体积上界	误差 %
2	图 4	53	245760	0.93″	999424	+13.4
3		237	382976	4.18″	757760	+3.89
4		969	460800	17.36″	646400	+0.83
5		3869	503808	1′9.5″	595968	+0.016
备　注	真实体积　549016.92					

表 3

分解	球					体积　相对
层数	外型	分解节点数	体积下界	分解时间	体积上界	误差 %
2	图 5	65	557056	0.93″	1671168	+1.46
3		337	802816	5″	1396736	+0.16
4		1497	946176	21.8″	1250816	+0.04
5		6257	1021056	1′42.45″	1174912	−0.004
备　注	真实体积　1098033.83					

表 4

分解层数	单叶双曲面					平均 体积误差 %
	外型	分解节点数	体积下界	分解时间	体积上界	
2	图 6	65	229376	1.31″	1048576	+9.58
3		265	376836	5.4″	819200	+2.56
4		1129	476160	23.1″	696832	+0.58
5		4577	529280	1′34.08″	638336	+0.12
备 注		真实体积 583112				

图 4、5、6 给出了利用本文算法对 3 个实体进行不同分解精度完全分解的结果显示，造型效果证明了分解算法的计算结果是准确的. 从平均分解时间来看，在 16MHZ 微机上，每秒可分解 59 个半满节点. 而从文献[5]的两个干涉检验实验来看，当检验精度为 5 层时，实验 1 分解了 110 个节点，实验 2 分解了 160 个节点；文献[3]8 个块的自动化布局全过程共分解了 52 个半满节点；由此可知，自动化布局中所需分解的半满节点数的数量级也就在百位上，应用本文算法在 16MHZ 微机上(不附加协处理器)只需几秒或十几秒的时间，基本可达到布局过程对八叉树节点分解的实时性要求. 由于八叉树是三维实体在空间体积分布的近似表达，因此，八叉树计算出的体积与真实体积的误差就基本上反映了八叉树表达对三维实体真实情况的近似程度. 从实验结果及实体图中可以看出，当完全分解物体至第 4 层时，无论从视觉效果还是体积误差看，都已经非常接近于真实情况了.

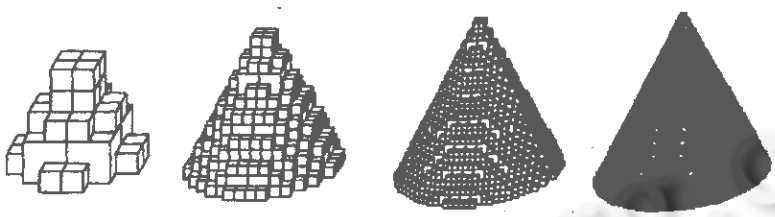

图 4

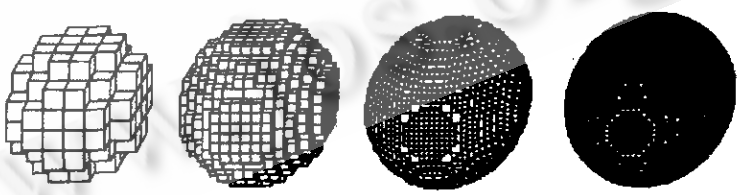

图 5

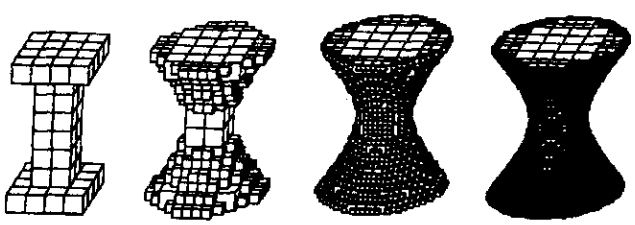

图 6

7 结 论

八叉树在三维实体布局中的应用主要涉及实体造型、干涉检验、基于节点匹配的启发式算法及物性计算等等.这一切,尤其是实体造型和干涉检验,对算法的实时性要求很高,而影响上述算法实时性的一个主要因素就是节点的分解速度.

理论分析及实验特性研究证明本文所提出的节点分解算法的平均计算效率较高,完全可以实现分解精度与时间的合理折衷,满足布局对实时性及计算精度的要求.

参考文献

1 Samet H. The design and analysis of spatial data structures. Reading Massachusetts: Addison Wesley, 1990.

2 戴佐,查建中.面向对象的三维布局仿真系统 OO3DP.计算机仿真,1993,1(3):16—21.

3 戴佐,袁俊良,查建中.一种基于八叉树结构表达的三维实体布局启发式算法.软件学报,1995,6(10):629—636.

4 Donald Meagher. Geometric modeling using octree encoding. Computer Graphics and Image Processing. 1982,12(19):129—147.

5 袁俊良.三维实体布局仿真系统的研究[硕士论文].天津大学,1994.

A FAST DECOMPOSITION ALGORITHM OF OCTREE NODE IN 3D—PACKING

Dai Zuo　Cha Jianzhong　Ni Zhongli

(*Intelligence Engineering Laboratory, Department of Mechanical Engineering, Tianjin University, Tianjin* 300072)

Abstract　Based on the classification of the vertices of 8 subcubes and Half—spaces of CSG model, this paper presents a compound algorithm to convert the CSG model to Octree representation of 3D objects. The computation results on several objects show that the algorithm can meet the needs of automation of 3D—packing.

Key words　3D—packing, octree, node, fast decomposition, algorithm.