

一种基于八叉树的三维实体内部可视化技术

朱响斌 唐敏 董金祥

(浙江大学 CAD&CG 国家重点实验室, 杭州 310027) (浙江大学人工智能研究所, 杭州 310027)

摘要 鉴于传统的图形学——面图形学只能表达三维实体的表面的形状和属性,不能表达实体内部的属性,如纹理、密度场以及温度场等,因而在计算机图形学、CAD 以及有限元分析等许多领域都需要一种新的可视化技术——三维实体的可视化技术,以表达实体内部的属性。三维实体的可视化技术是新兴的图形学——体图形学的一个重要组成部分。为了实现三维实体的可视化,针对 CAD 造型系统中的实际要求,根据体图形学的理论和八叉树的特点,提出了一种基于八叉树的实体内部可视化技术。该技术采用八叉树算法对边界数据结构表达的实体进行体元化。由于实体内部属性变化的不均匀,算法采用了不规则体元,以充分表达实体内部的细节。实际应用效果表明,该算法不但能充分表达实体内部属性,而且也具有一定的造型功能。

关键词 体图形学 八叉树 可视化

中图分类号: TP391.41 **文献标识码**: A **文章编号**: 1006-8961(2002)03-0229-05

A Technology Based on Octree for Interior Visualization of 3D Objects

ZHU Xiang-bin, TANG Min, DONG Jin-xiang

(State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou 310027)

(Institute of Artificial Intelligence, Zhejiang University, Hangzhou 310027)

Abstract The traditional graphics——surface graphics only can display the surface of the 3D objects. It can not display interior properties of the 3D objects, such as texture, density field and temperature field. So many fields such as computer graphics, CAD and finite element analysis, need a new visualization technology for displaying the interior of 3D objects. The visualization technology of 3D object is an important part of a new graphics——volume graphics. Lots of information of solids, such as texture and density, are often hidden in the interior of 3D Objects. Based on the volume graphic theory and the characteristic of octree, this paper proposes a technology for interior visualization of the 3D objects for the demand of CAD system. The octree arithmetic is applied for voxalization which converts geometric objects into a set of voxels that best approximates the continuous object. As a result of uneven changes of interior properties, the technology uses the unequal voxels to best represent the details of interior properties. According to many practices, the technology not only can represent interior properties sufficiently, but also has some modeling functions.

Keywords Volume graphic, Octree, Visualization

0 引言

在 CAD 的造型系统当中,经常要涉及到表达几何实体的内部属性这一问题,因为在实体内部有着较复杂的属性(如温度场、三维纹理以及材料密度分布等等),但从点集拓扑模型^[1]的思想出发,可将

这些属性看作是三维实体占据空间位置的函数,其中,典型的,如模拟三维纹理的过程函数。

由于现有的 CAD 的造型系统往往采用边界数据结构,这种结构虽便于面绘制、存储与传输,但不便于表达实体内部属性及结构,因此,需要把以边界表示的实体剖分成体元(Voxel)集合,然后再进行体绘制,以实现三维实体内部的可视化,其实现的流程如

图 1 所示,且大致可分为以下 3 步:

- (1) 几何实体的面绘制.
- (2) 将多面体模型剖分为体元的集合.
- (3) 体绘制.

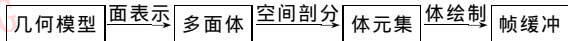


图 1 三维实体可视化流程

由此可见,这个过程是与 Kaufman 提出的体图形学(Volume Graphic)^[2]的内容相一致的,但它不同于体视化技术,体视化过程可用图 2 来表示,由图 2 可以看出,体视化是基于离散化和采样后数据集的,而体图形学是基于几何实体的,且体图形学需要进行几何实体的剖分操作,也即体元化,而体视化技术则有类似相反的过程,即需要首先从体元模型中构造出等值面,再把体数据转换成面数据,以实现面绘制.体图形学也不同于面图形学(Surface Graphic),由于体图形学能够方便地显示出实体内部的信息特征,也便于几何实体的干涉检查、集合操作以及块操作,因此体图形学在 CAD 造型系统中有着很好的应用前景.

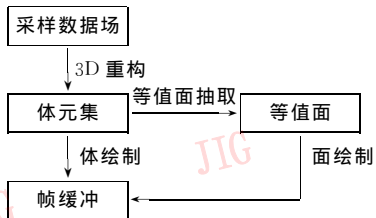


图 2 体视化流程

体图形学中的几何模型,可以采用文献[3]提出来的体几何模型,以便于有限元的剖分,但是现有的 CAD 系统一般采用了较成熟的 CSG 模型以及边界表示模型,它们一般要求在原有数据结构(如半边数据结构)的基础上来实现三维实体内部的可视化.由于上述过程的第 1 步操作,即面表示,已在一般 CAD 系统实现,因此,本文在由现有边界表示的多面体模型的基础上,针对体图形学中的后两步工作,提出了一种相应的基于八叉树的实现技术.

1 多面体的体元化

所谓多面体的体元化是把面图形学中表达的多面体转换成体图形学的一个多面体,且该多面体在体图形学中是一个由体元组成的集合;而面图形学中用来表达一个多面体常用的表示方法则有 CSG

表示、边界表示法等,其中,边界表示法,由于它的灵活性和精确性,使得它是一种应用得较多的表示方法,为此,本文考虑采用边界表示法中常见的由半边数据结构表达的几何实体来进行多面体的体元化.

1.1 基本概念

体元(Voxel) 也称为体素或体单元,即每个体元是一个体单位,并且附带有表示实体属性空间分布的数据,它是二维像素在三维空间的引伸.

体元化(Voxelization) 也称体素化,指在离散体元空间中,把一个几何实体转化为体元集合的过程,该集合是由许许多多的体元组成的,而且它在离散体元空间中,应能尽可能精确地表达该几何实体.

1.2 半边数据结构

边界表示法的几何模型是一种多面体模型,该边界模型采用体、面、环、边、点的空间坐标和拓扑关系来表达实体的形状,这种边界模型不仅可以十分精确地描述实体,并且与其他图形软件的接口也容易,表达范围也广泛.将多面体模型剖分为体元或三维网格,也就是将面数据转换成体数据,可以采用很多种方法,譬如三维扫描转换算法.另外,也可采用八叉树方法,把实体剖分成由一个个体元(Voxel)组成的集合,由于采用八叉树方法可以与造型系统中的空间数据模型相结合,并使得 CAD 系统可以很方便地进行布尔操作、集合操作以及干涉检查,因此本文就以八叉树剖分为基础来实现三维实体体元化.

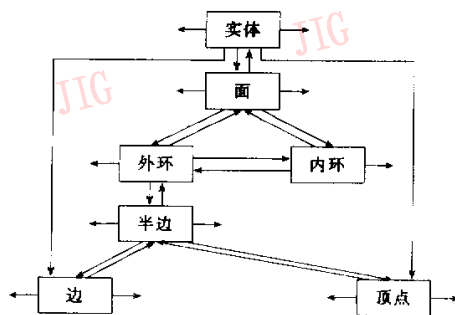


图 3 半边数据结构模型

边界表示法的实体造型系统中,常用的数据结构为如图 3 所示的半边数据结构,并且具体的半边数据结构,一般要比图 3 模型稍为复杂一些,譬如,本文采用的实体(Solid)数据结构如下:

```

struct _TSolid {
    IdSeeds    ids;           /* 多面体标志 */
    Box3D     box;           /* 多面体的包围盒 */
    TShell    * shells;      /* 组成多面体的壳链表 */
    TShell    * outer_shell; /* 外壳 */
}

```

```

TSolid * next; /* 下一个多面体 */
TSolid * prev; /* 上一个多面体 */
};

```

其中,TSolid 中的 box 为实体的包围盒.

1.3 体元数据结构

由以上半边数据结构可以看出,半边数据结构虽有很完全的多面体边界信息,但要进行八叉树剖分,最好把多面体的每个面都进行三角剖分,以利于八叉树剖分中立方体与边界面的求交.

如前所述,尽管本算法采用八叉树剖分技术,可把多面体剖分为由一个个体元组成的集合或体缓冲,但是由于用八叉树剖分技术剖分后,生成的是一个树形结构,因此必须把该链表结构转换为合适的体元数据结构,以便进行存储、传输及操作.这种体元的数据结构可以采用链表结构,且每一个链表结点表示一个体元,这样链表结构其实就构成了一个体缓冲(Volume Buffer),由于体元链表可以采用线性八叉树编码,因而不仅节约了内存空间,而且也能快速进行空间索引,由此可见,在体图形学里,体缓冲的操作是一个关键问题.这里,首先要考虑的是体元的大小问题,由于实体内部的属性或纹理,其在空间的分布是不均匀的,为了实体内部显示的真实性,八叉树剖分的体元大小应该尽量小,即接近于像素大小,但这样不但对内存空间要求大,并且体绘制的计算时间也长.在实际情况中,往往有些连续空间体元由于具有相同的颜色值和不透明度值,或者非常接近,因此可以充分利用颜色值和不透明度值的空间连续性,即对于颜色值和不透明度值空间变化不大的区域,采用较大的体元;而对于颜色值和不透明度值变化较大的区域采用较小的体元.由于八叉树剖分,是从大的立方体往小立方体剖分,因此做到这一点较为容易.具体做法是利用八叉树的递归停止条件来确定采用哪种剖分,停止条件为下列两个条件之一:

- (1) 立方体的长度小于某一阈值及立方体的 8 个角点处的颜色值和不透明度值的方差小于某一阈值.
- (2) 虽然立方体的 8 个角点处的颜色值和不透明度值的方差还大于某一阈值,但立方体的长度已小于某一较小的阈值.

这样对于需要再剖分的结点,体元数据结构在原链表基础上,必须要形成一个子体元的链表,可称为子链表(如图 4 所示).这一思想也可用于边界面上,在八叉树剖分中,最关键的是判断出哪些立方体与边界面相交,为了使利用立方体逼近的边界面更为光滑,立方体应越小越好,但这样占用空间和时间的

开销都大,因此,可以在边界面上的八叉树剖分中,采用提高分辨率的方法,即使实体内部剖分后的立方体分辨率小(立方体大),使边界面上的立方体分辨率大(立方体小),但为了方便以后的八叉树的几何操作,仍采用上述方法,保存原有体元结点,再在原体元结点下,形成子体元链表(如图 4 所示).

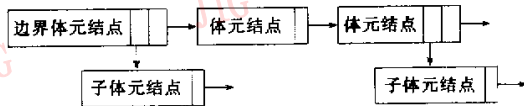


图4 体元结点与子体元链表

具体的体元数据结构如下:

```

typedef struct VoxelType{
    boxType boxclass; //体元的类型
    TVertex voxel; //体元坐标
    char * code; //八叉树线性编码,用于实体的几何操作
    int l; //体元的分辨率,l=2^i,其中i为剖分次数
    RGBA color; //体元的颜色及不透明度.
    struct VoxelType * next; //指下一体元.
    struct VoxelType * child; //指向子体元链表.
    Tlist * allTrimesh; //相交三角面片指针链表.
}VoxelType;

```

其中,体元链表中的线性八叉树编码是为了便于进行几何变换以及为了体绘制时的深度排序,或者为了进行体元数据的快速索引.这种线性八叉树编码采用八进制编码,且每个结点具有相同的编码长度.根据某结点的编码可知,其所有父结点的编号以及该结点的局部原点(即立方体 8 个顶点中具最小(x,y,z)坐标值的顶点)坐标就是体元(Voxel)坐标.具体原理可参考文献[4].

有了体元数据后,就可以进行体绘制,但由于体绘制往往需要突出边界面,因此必须要有多面体的面信息,由于三角面片的优点及其具有成熟的各种处理方法,因此在本文算法中,采用了与三角面片相结合的方式来进行处理,以便实现整个场景的三维可视化绘制.另外,采用三角面片也是为了使得再进行体绘制时,可采用多种绘制方法.图 5 是该数据结构的示意图.

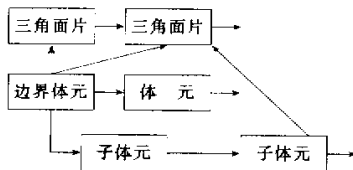


图5 体元数据结构示意图

1.4 基于八叉树的体元化

八叉树的体元化过程,主要分为如下两个步骤:一是进行八叉树的剖分;二是把树形的数据结构转化为体元链表结构.其中,主要的工作是八叉树剖分,而在进行八叉树剖分之前,一般首先需对多面体进行三角剖分,以形成一个三角面片表.具体的八叉树剖分算法如下:

```
void COctree::make_tree(TList * allTrimesh, TList *
primitive, octreeType * tree, int depth)
{
    int i, j;
    switch (classify (allTrimesh, primitive, tree)) //判断
    当前八叉树结点,是否与其父结点相交的三角片相交,从而
    得出其是 WHITE、BLACK 或 GRAY 结点.
    {
        case WHITE: tree->boxclass = WHITE; break;
        case BLACK: //如果是 BLACK 结点,按照上述的两个
            停止条件,进行剖分.
            tree->boxclass = BLACK;
            depth++;
            setson(tree, depth);
            break;
        case GRAY: //如果是 GRAY 结点,继续进行八叉树
            剖分
            if (depth == d1) //d1 为边界结点的分辨率.
                tree->boxclass = BLACK;
            else {
                tree->boxclass = GREY;
                setson(tree);
                depth++;
                for (i = 0; i < 8; i++)
                    make_tree(allTrimesh, tree->ctrlList,
                    tree->octree[i], depth);
            }
    }
}
```

2 体绘制

由于场景中既有面数据,又有体数据,因此需要将面绘制和体绘制结合进行混合绘制.混合绘制可以采用混合光线跟踪算法和组合绘制算法.本文采用了组合绘制算法,也就是将面数据和体数据分开绘制,并将所得图象作为中间结果再进行组合,以形成混合绘制的图象.绘制的主要步骤如下:

(1) 对多面体模型进行面绘制,譬如采用扫描线算法.

(2) 将面绘制中各象素颜色及不透明度存放在帧缓存中,象素的浓度深度信息存放在 Z-Buffer 中.

(3) 用相关性体元投射进行体绘制的象素,可以在绘制过程中,组合原先帧缓存中的象素信息.

由于采用八叉树将多面体剖分为体元,其实体的空间信息已保存在混合数据结构中,并且能够方便进行空间索引,因此,本文算法采用体元投射法来实现体绘制,虽然体元投射法可以再次利用三维空间各个 Voxel 的空间相关性,以减少计算量,但是它又不同于一般的体元投射法,其中原因在于,有的体元有子体元.而 Wilhelms 等人提出的体元投射法^[4]是基于规则数据场的,其中体元的大小和形状都是相同的,而采用本文算法形成的各体元形状虽是相同的,但是大小却不一样.

其解决的方法是再剖分,由于八叉树剖分后的体元分辨率都是可知的,因此,可以重新定义一个统一的分辨率进行再剖分.由于再剖分的各角点值非常接近,所以在进行体元投射时,进行相应的处理也能达到同样剖分效果.现假设剖分后的体元分辨率为 l_1 ,子体元分辨率为 l_2 ,现要求的分辨率为 l_3 ($l_3 > l_2 > l_1$),这样对体元在图象合成时,若对于该体元重复处理 ($l_3 - l_1$) 次,对子体元重复处理 ($l_3 - l_2$) 次,则图象合成公式如下:

$$\alpha_{out} = \alpha_{in} (1 - \alpha_{now})^k + \alpha_{now}^2 \sum_{i=1}^{k+1} (1 - \alpha_{now})^{k-i-1}$$

$$C_{out} = C_{in} (1 - \alpha_{now})^k + C_{now} \alpha_{now} \sum_{i=1}^{k+1} (1 - \alpha_{now})^{k-i-1}$$

这种处理就相当于进行了再剖分,而实际却没有剖分,这样就达到了节约内存空间的目的.

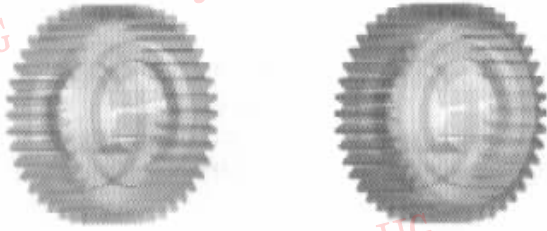
由于采用八叉树剖分成的各个体元,具有相同的形状,且具有相同的角度,因此各体元在图象平面上,平行投影几何体的形状也是一样的,而且当三维数据场的投影方向给定以后,即可选定一种体元投射几何模型来进行所有体元的运算,而其余的处理方法与体元投射法一样.

3 实现结果

为了检验本文算法的效果,利用一齿轮实体对该算法进行了绘制试验.大家知道,采用传统算法是把实体剖分成均匀大小的体元集合,而采用本文的算法是把实体剖分成不均匀大小的体元集合.由于齿轮内部有 3 种颜色,因此在大致相同的绘制条件下,如用传统算法,空间开销为 7 571 个体元,绘制效

果如图6(a)所示,本文算法空间开销为13 752个体元,效果图见图6(b),效果显然以图6(b)较好,如果采用传统算法要达到相同效果,则体元数要达

32 059个.可见,采用本算法可以在较小的空间开销的情况下,达到较好的显示效果.图7是一个减速机箱盖的绘制效果比较.



(a) 传统算法绘制结果

(b) 本文算法绘制结果

图6 不同算法绘制的齿轮三维可视化图

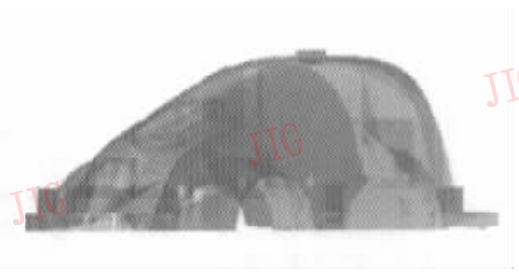


图7 本文算法绘制的减速机箱盖三维可视化图

4 结 论

实体内部的可视化技术属于体图形学的内容^[5],随着体图形学技术及硬件技术的发展,该技术在CAD及有限元分析领域将有着广泛的应用前景.本文针对体图形学在具体实际情况中的应用提出了一种实现方法,该方法可应用于大多数的CAD造型系统当中.

参 考 文 献

- 1 Rosenblum L J. Research issues in scientific visualization[J]. IEEE Computer Graphics and Application, 1994, 14(2): 70~73.
- 2 Kaufman A, Cohen D, Yagel R. Volume graphics [J]. Computer, 1993, 26(7): 51~64.
- 3 王利生, 谈正. 三维实体的体几何模型[J]. 计算机学报, 1999, 22(7): 777~780.
- 4 彭群生, 鲍虎军, 金小刚. 计算机真实感图形和算法基础[M]. 北京: 科学出版社, 1999: 175~180.
- 5 Kaufman A. Research issues in volume visualization[J]. IEEE Computer Graphics and Application, 1994, 14(2): 63~167.

朱响斌 1971年生,1993年毕业于东南大学,现在浙江大学攻读工程硕士学位.主要研究方向为计算机图形学与辅助设计、体视化以及图象处理.

唐敏 1974年生,1999年获博士学位,副教授.主要研究方向为计算机图形学与辅助设计、几何造型.

董金祥 1945年生,教授,博士生导师.主要研究方向为计算机图形学与辅助设计、先进制造技术与自动化、DBMS.