

# 基于 MC 思想的布尔操作研究\*

孙立鏊<sup>a</sup>, 金瑛浩<sup>a</sup>, 孙大松<sup>b</sup>

(哈尔滨理工大学 a. 计算机科学与技术学院; b. 计算中心, 哈尔滨 150080)

**摘要:** 为了提高布尔运算的执行效率及系统的鲁棒性, 提出一种基于细胞元模型的布尔运算算法。该算法利用 MC 思想为每个特征实体的三角片建立 cube, 用最值空间和交互的充分性原理来快速检测发生相交的三角片, 并根据每个 cube 顶点的颜色来判断三角片的内外关系、所属关系以及三角片的剖分方式, 最后生成新的特征实体。该算法不仅提高了布尔运算的速度和性能, 还避免了以往算法中出现孔洞的缺陷。实验表明, 该算法具有更强的适应性和实用性。

**关键词:** MC 算法; 布尔操作; 细胞元模型; 语义特征; 特征建模; 最值空间; 等值面抽取; 三角剖分

**中图分类号:** TP391      **文献标志码:** A      **文章编号:** 1001-3695(2010)07-2754-03

doi:10.3969/j.issn.1001-3695.2010.07.100

## Research of MC-based Boolean operation

SUN Li-juan<sup>a</sup>, JIN Ying-hao<sup>a</sup>, SUN Da-song<sup>b</sup>

(a. College of Computer Science & Technology, b. Computer Center, Harbin University of Science & Technology, Harbin 150080, China)

**Abstract:** In order to improve the efficiency and robustness of Boolean operation, this paper proposed a new Boolean operation algorithm based on the cellular model. This algorithm made use of the thinking of MC created cubes for each triangle of feature entities. It tested the interacted triangle rapidly through the maximum value space and the sufficiency principle of feature interaction and estimated the inside or outside of the triangle, the relations of belongingness and the method of triangulations by the colors of the cube's vertexes, then built the new feature entity. The new algorithm of Boolean operation can increase the speed and the performance of computing and avoid the errors about holes in the old entity. The experimental results show that the new algorithm is more adaptable and practicable.

**Key words:** marching cubes; Boolean operations; cellular model; semantic feature; feature modeling; maximum value space; isosurface approximation; triangulation

布尔运算<sup>[1,2]</sup>是计算几何、计算机图形学中的基础问题之一,它在各种造型领域都有广泛的应用,是最主要的造型手段之一。通常意义上的三维布尔运算都是针对封闭实体之间进行的,但在实际的三维可视化建模应用中往往存在非封闭实体与封闭实体、非封闭实体与非封闭实体之间的空间布尔运算<sup>[3,4]</sup>。目前国内外采用的方法多是通过取近似值或分类计算的方式进行,算法复杂且低效,尤其对复杂模型更是如此。很多系统在进行布尔操作时不仅会降低模型的精确度,甚至还会出现孔洞的错误<sup>[5,6]</sup>。本文结合自主开发的基于细胞元模型的语义特征建模系统 HUST-CAID,讨论了以细胞元为基础的特征造型技术,提出了基于 MC 思想的布尔运算算法,不仅提高了系统布尔运算的速度,还使其适用于有非封闭实体参与的运算。

### 1 细胞元语义特征造型

语义特征造型<sup>[7]</sup>的基本思想就是在一个特征模型中把形状信息与功能信息有机地结合在一起,共同构成特征的语义,其最终目的就是维护特征模型的有效性。语义特征造型最重要的特点是所有特征的语义在整个造型过程中对所有的特征操作来说都不改变。

细胞元语义特征造型系统(如 HUST-CAID)就是以细胞元模型为基础的语义特征造型系统,它分为三层结构(图 1):底层是数据模型,即设计者全面设计特征实体,并且把所有的特征信息及相互之间的联系保存在数据模型中;中间层是细胞元几何模型,它能够全面完整地把特征显示出来,但它不是面向用户的,只是一种过渡性几何模型;最上层是视图模型,视图是根据用户的需要(如面向生产或面向加工等)从细胞元几何模型映射出来的模型,它面向终端用户,最接近真实效果。这种分层结构不仅可以保留语义特征造型系统的所有优点,还可以通过特征依赖图(FDG)使特征修改不依赖于历程树,因而在造型过程中能更有效地维护特征的语义(特征模型有效性维护)。

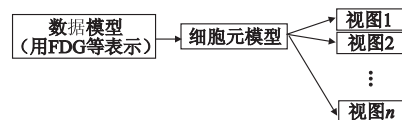


图1 细胞元语义特征造型系统结构

细胞元模型将实体描述成相互邻接或离散的细胞元集合。细胞元是一个关于点集合的描述,这个点集包含于模型各个面的所有形状特征中,因此,每个形状特征都是由一组相关细胞元的子集所表示的。各子集之间的关系由各自的所属列表来维护。

收稿日期: 2009-11-28; 修回日期: 2010-01-11      基金项目: 国家自然科学基金资助项目(60173055)

作者简介: 孙立鏊(1944-),男,教授,博导,主要研究方向为计算机图形学与 CAD; 金瑛浩(1978-),男,讲师,博士研究生,主要研究方向为计算机图形学与 CAD(chinajyh@yeah.net); 孙大松(1972-),男,副教授,主要研究方向为计算机图形学与 CAD。

## 2 基于 MC 思想的布尔运算

MC(marching cubes)算法<sup>[8-10]</sup>是基于体元的一种典型的面绘制算法。自 Lorensen 等人于 1987 年提出该算法以来,就因为其简单、高效和适用性强博得了众多研究人员的关注,被广泛应用于医学图像三维重建、隐式曲面绘制、分子表面显示、地质矿藏分布构造等诸多标量场中可视化问题的解决,被认为是目前最流行的等值面抽取算法之一,对科学计算可视化领域的研究产生了深远的影响。

### 2.1 基于 MC 思想的 cube

MC 算法虽然是一种等值面的绘制技术,但其算法思想在布尔运算中仍然有两点值得借鉴,一是通过建立 cube 来简化处理的复杂度,提高运算的速度;二是充分利用 cube 的八个顶点来确定等值面的“内”“外”关系。

a) 为每个实体的三角片建立一个 cube。为了方便 cube 的构建与以后的相关计算,同时引入最值空间的概念。最值空间是由特征或外形的点集在坐标轴上最远投影所确定的空间。例如, $R$  是某一特征上的点集, $R_x, R_y$  和  $R_z$  分别是  $R$  中所有点的横坐标、纵坐标及竖坐标的集合,若  $x_1 = \min \{x_i | x_i \in R_x\}, x_2 = \max \{x_i | x_i \in R_x\}, y_1 = \min \{y_i | y_i \in R_y\}, y_2 = \max \{y_i | y_i \in R_y\}, z_1 = \min \{z_i | z_i \in R_z\}, z_2 = \max \{z_i | z_i \in R_z\}$ ,那么最值空间八个顶点的坐标分别为  $V_1(x_1, y_1, z_1), V_2(x_2, y_1, z_1), V_3(x_1, y_2, z_1), V_4(x_2, y_2, z_1), V_5(x_1, y_1, z_2), V_6(x_2, y_1, z_2), V_7(x_1, y_2, z_2), V_8(x_2, y_2, z_2)$ 。若把一个三角片的三个顶点作为最值空间的点集,那么这个最值空间就称为三角片的 cube。很容易证明,一个三角片无论其方向和形状如何,它的三个顶点必然位于 cube 的面或边上。

b) 根据三角片顶点的法线来确定 cube 顶点的颜色。处于物体内的顶点标记为黑色,处于物体外的顶点标记为白色。这样通过 cube 顶点的颜色就能确定某个三角片的“内”“外”关系(图 2)。这样在布尔运算时只需根据 cube 的顶点就可以确定相交部分中某些区域的取舍,而不需要每次都计算每个顶点的法线方向,大幅度地降低了运算的复杂度并节省了时间。

### 2.2 相交检测

采用最值空间的方式来建立 cube 的一个最大好处就是所有 cube 的边与各个坐标轴都是正交或平行的,这就使得判断任意两个 cube 之间的相交关系变得十分简单,只要通过两个 cube 的顶点的大小关系就可以实现。而 cube 相交是三角片相交的必然条件,也就是说只要三角片相交,那么它的 cube 也必然相交,反之则未必。因此只需要检测一个 cube 的八个顶点是否存在于另外一个 cube 中就可以了。若一个 cube 的顶点中有一个或多个顶点处于另外一个 cube 中,则两个三角片可能相交,需要做进一步的相交检测;若没有顶点处于另外一个 cube 中,则说明两个三角片不可能相交,也就没有必要进行相交检测。

Cube 的相交检测首先可以确定哪些三角片可能发生相交,也就是可以确定哪些三角片处于两个物体相交的边缘上。这是进行布尔运算的第一步,也是最重要的一步。通过 cube 的筛选使得实际上能够参与布尔运算的三角片明显减少,从而为降低布尔运算的复杂度和提高布尔运算的执行速度奠定了基础。

### 2.3 面片求交与三角剖分

设参加布尔运算的两个实体分别为  $A$  和  $B$ 。若  $A$  中可能发生相交的三角片组成的集合为  $T_A, B$  中可能发生相交的三角片组成的集合为  $T_B(T_A$  和  $T_B$  中的三角片仍然按照其在原实体中的顺序来存储)。首先从  $T_A$  中选择第一个三角片  $t_{a1}$ ,

那么  $T_B$  中可能与  $t_{a1}$  发生相交的三角片的集合  $I_1$  必然满足  $I_1 \subseteq T_B$ 。若经过计算, $I_1$  中的所有三角片与  $t_{a1}$  都无交点,那就说明这个三角片不是被分割面上的三角片,因此直接从  $T_A$  中删除它,然后从  $T_A$  中选择下一个三角片。若  $t_{a1}$  与  $I_1$  中的某一个或某几个三角片有交点,则将这个三角片中的交点记录到由  $A$  和  $B$  共享的链表  $L$  中。当  $t_{a1}$  处理完后,从  $T_A$  中选择下一个三角片,直到  $T_A$  中的所有三角片都被遍历。很容易理解,对于  $T_A$  和  $T_B$  的处理与具体的操作(如交、并或差)没有关系,与实体选择也没有关系,所以无论是选择哪个集合处理,其最终结果都是一样的。又因为链表  $L$  是由  $A$  和  $B$  共享的,所以没有必要再对  $T_B$  中的三角片进行检测,这就使得对布尔操作的最终计算变得更加简便。

如图 3 所示,当两个三角片  $a$  和  $b$  相交时,两个三角片交点  $C$  和  $D$  的连线  $L$  就是两个三角片的交线。当一个三角片与多个三角片相交时,只要遍历所有与此三角片相交的三角片,并将其交点相连就是这个三角片上的角线。对于多对多的相交情况也是如此。因此采用这种存储方式不需要记录交线的信息就可以满足布尔运算的要求,这不仅节省了布尔运算的存储空间也提高了它的运算速度。很容易证明,遍历计算完  $T_A$  中所有的三角片时, $T_B$  中的所有三角片也都会被遍历完。因此,虽然通过 cube 相交检测得到了两个集合  $T_A$  和  $T_B$ ,但是只需要遍历计算其中任意一个集合就可以满足要求。在实际过程中,本文选择  $T_A$  和  $T_B$  中较小的集合,以减少相交计算的次数。

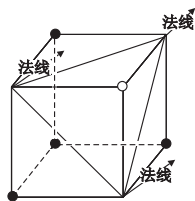


图2 三角片的cube及其顶点颜色

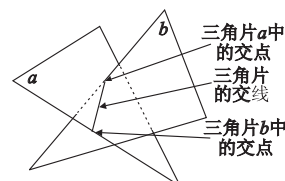


图3 相交三角片的交点与交线

相交检测的主要工作就是建立并维护由两个实体共享的用于标记被分割三角片及相交交点的链表  $L$ ;而检测后的另外一个重要工作就是根据链表  $L$  对被分割的三角片进行正确的三角剖分,以保证布尔运算后的实体不会产生错误的面或歧义面。错误面或歧义面是导致最终实体孔洞的主要原因。因此,根据链表  $L$  的三角剖分将对布尔运算的最终效果产生重要的影响。

在对被分割的三角片进行三角剖分时,要遵循下面三个步骤:

- a) 由被分割三角片与其相交三角片上的交点确定其交线。
- b) 如果三角片上无交点,说明这个三角片被另外一个三角片完全分割(图 4),那么它的交线则由分割三角片上的与被分割三角片对应的交点来标记。这种情况的三角剖分只需要对包含原来两个角的三角片部分进行剖分,而对只含一个角的部分仍然是三角形,因此无须继续剖分。剖分时只要通过其中的一个交点(在实际应用中,本文选择在链表  $L$  中靠前的交点)与对角的三角片顶点确定即可,如图 5(a)和(b)所示。

c) 如果三角片上的交点数量大于一个,说明这个三角片被至少一个三角片分割,由其产生的两个实体间的交线也是一段折线段。这种情况下,对于被分割的三角片的两个部分都要进行剖分。对于含有一个角的部分可以通过顶点与各个交点来确定,而对于含有两个角的部分则使用 Delaunay 三角剖分算法。其剖分结果如图 5(c)和(d)所示。

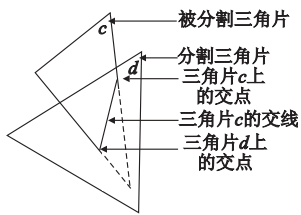


图4 被另外一个三角片完全分割的三角片

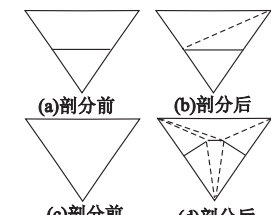


图5 被分割三角片的三角剖分

2.4 区域分类与生成实体

三角片求交与三角剖分为生成最后的实体做好了准备,但是根据具体的运算,同样一个三角片可能会属于不同的实体。比如一个三角片在交运算的时候可能属于新生成的实体 C,而在并运算的时候就可能被直接删除而不属于任何一个实体。因此在布尔运算中如何确定实体各元素的所属关系也是一个重要的问题。很容易理解,对于任意两个相交的三角片,其 cube 也必然相交。无论两个三角片如何相交, cube 空间都被分割成为三个部分: a) 只属于实体 A 的空间  $S_A$ ; b) 只属于实体 B 的空间  $S_B$ ; c) 实体 A, B 相交区域的空间  $S_{AB}$ 。对于  $S_A$  与  $S_B$  中的三角片的取舍处理相对比较简单: 若进行并操作,那么处于  $S_A$  和  $S_B$  中的三角片都要保留; 若进行差操作(假设是  $A - B$ ),那么实体 B 中除了  $S_{AB}$  中的三角片以外全部删除,而实体 A 中除了处于  $S_{AB}$  中的三角片外全部保留; 若进行交操作,那么除了  $S_{AB}$  中的三角片以外,实体 A 和 B 中所有的三角片都要删除。

然而对于处于  $S_{AB}$  中的三角片的取舍则相对麻烦些,需根据所生成实体的组成和三角片的内外关系来决定。在  $S_{AB}$  中,根据三角片的内外关系可将  $S_{AB}$  分成四个部分:  $S_A$  的内部与  $S_B$  的内部、 $S_A$  的内部与  $S_B$  的外部、 $S_A$  的外部与  $S_B$  的内部以及  $S_A$  的外部与  $S_B$  的外部。需要指出的是,由于相交的两个三角片的大小、方向都是不确定的,这四个部分的大小和形状也都是不确定的。但是无论三角片如何相交,这四个部分也都是必然存在的(只是大小和形状有所不同),对这些三角片的处理方式也是一致的。因此为了方便说明具体的处理方式,下面给出了一个比较特殊的模型,如图 6 所示。

若实体 A 与 B 并运算,则保留图 6 中①和③中的三角片,删除②④中的三角片; 若实体 A 与 B 差运算(假设  $A - B$ ),那么只保留①中的三角片,删除②③④中的三角片; 若实体 A 与 B 交运算,那么保留②中的三角片,删除①③④中的三角片。

最后将所有被保留的三角片添加到新产生的实体中的所有者列表中。由于细胞元模型对模型各个元素的组织是通过所有者列表来实现的,对面片的删除和添加其实就是对所有者列表的操作。如果新生成的实体叫做 C,那么只需要顺序合并实体 A 和 B 的所有者列表即可。因为需要删除的三角片已经在先前的操作中从各自的所有者列表中删除了,所以三种不同的布尔运算(交、并、差)最终都转换成了一种运算,即实体的所有者列表的合并操作,可以大幅度提高布尔运算的计算速度。

3 实验结果与分析

图 7 是在 HUST-CAID 上对两个封闭实体进行交、并和差运算的模拟; 图 8 是对封闭实体与非封闭实体布尔运算的模拟; 图 9 是非封闭实体与非封闭实体布尔运算的模拟。

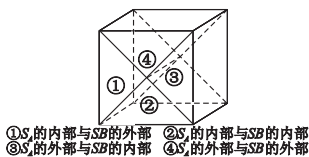


图6  $S_{AB}$ 空间的四个部分

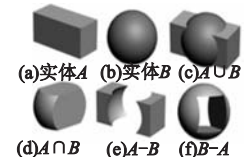


图7 封闭实体间的布尔运算

在 HUST-CAID 系统中,为了保持特征实体的语义和运算的一致性,对于有非封闭实体参与的布尔运算不作分割处理(在 HUST-CAID 中另有分割运算)。因此封闭实体与非封闭实体的差运算以及非封闭实体间的交、差运算前后变化无法从实体渲染图中看出变化。在图 8 和 9 中未给出这些运算结果的渲染图。

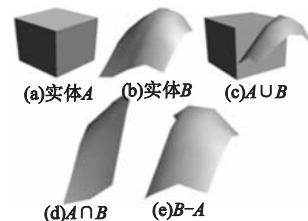


图8 封闭实体与非封闭实体间的布尔运算

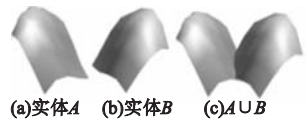


图9 非封闭实体与非封闭实体间的布尔运算

由于本算法在三角剖分及区域分类过程中没有进行近似的操作,生成的实体会保持原实体的精度; 另外这种布尔操作由于改进了相交检测和区域分类的方法,在速度方面的优势比较明显。通过实验发现,每次被检测的三角片一般不会超过实体全部三角片的 13%,而被剖分的三角片不会超过实体全部三角片的 11%,检测的准确率一般可以达到 84.6% 以上(表 1)。而且算法的稳定性受实体复杂度的影响较小,所以这种检测方法对性能提升的帮助很大。

表1 被检测三角片与被剖分三角片的比例表

| 实体三角片数 | 被检测三角片数 | 被剖分三角片数 | 检测准确率/% |
|--------|---------|---------|---------|
| 100    | 13      | 12      | 92.31   |
| 200    | 25      | 22      | 88.00   |
| 500    | 67      | 55      | 82.90   |
| 1 000  | 122     | 102     | 83.61   |

4 结束语

从拓扑学的角度看,传统的布尔运算就是通过拓扑运算和几何运算重新生成一个新的拓扑结构,它只是对拓扑实体的拓扑信息和几何信息进行的操作。基于细胞元模型的语义特征造型系统在修改实体几何和拓扑结构的同时,可以很好地保持各种工程信息,因此更具实际应用价值。本文提到的布尔运算策略吸取了 MC 算法的优点,不仅提高了相交面的检测速度,还简化了区域分类与生成实体的计算,将多种不同的运算归并成一种运算,可以在不降低原实体精度的前提下提高布尔运算的速度。该算法虽然在速度、精度及使用范围上取得了一定的进步,但是对于布尔操作时语义的保持还是采用了原来的算法,笔者下一步将针对这一问题进行深入研究。

参考文献:

- [1] 郭开波, 张李超, 王从军. STL 模型布尔运算的实现[J]. 华中科技大学学报: 自然科学版, 2006, 34(7): 97-99.
- [2] MANTYLA M. Boolean operations of 2-manifolds through vertex neighborhood classification[J]. ACM Trans on Graphics, 1986, 5(1): 1-29.
- [3] SCHNEIDER P J, EBERLY D H. Geometric tools for computer graphics [M]. 北京: 电子工业出版社, 2004.
- [4] TOMAS M. A fast triangle-triangle intersection test[J]. Journal of Graphics Tools, 1997, 2(2): 25-30.
- [5] 刘红军, 王从军, 黄树槐. 带有孔洞的多边形的布尔算法[J]. 华中科技大学学报: 自然科学版, 2003, 31(8): 18-20.
- [6] RIVERO M, FEITO F R. Boolean operations on general planar polygons[J]. Computer & Graphics, 2000, 24(6): 881-896.
- [7] BIDARRA R, BRONSVOORT W F. Semantic feature modeling[J]. Computer Aided Design, 2000, 32(3): 201-225.
- [8] LORENSEN W E, CLINE H E. Marching cubes: a high resolution 3D surface construction algorithm[J]. ACM Computer Graphics, 1987, 21(4): 163-169.
- [9] 刘迎, 蔡康颖, 王文成. 基于 marching cubes 重组的外存模型渐进压缩[J]. 计算机学报, 2004, 27(11): 1457-1463.
- [10] 梁秀霞, 张影明, 刘毅. 拓扑结构正确的三线性插值曲面的三角片逼近[J]. 计算机研究与发展, 2006, 43(3): 528-535.