

# 计算几何试验报告

王佟	014952	<a href="mailto:only.rain@263.net">only.rain@263.net</a>
郭小波	014905	<a href="mailto:lengbing@263.net">lengbing@263.net</a>
王玉刚	014891	<a href="mailto:myzdf@263.net">myzdf@263.net</a>

## 问题的背景:

问题的背景来自 VLSI 中电路参数提取。

近年来, 我们课题组和 Synopsys 公司合作开发了三维寄生电容提取软件 B3D。B3D 是一个交互式的高效的电容计算软件, 它包括寄生器件描述语言 PDDL, 三维寄生参数提取, 和与用户交互的图形界。目前 B3D 系统对于复杂的形体结构的描述不是很方便。整个系统描述的基本的形体是平行六面体, 对于复杂的形体结构我们就要用平行六面体来构建, 在构建过程中遇到的最大的问题是平行六面体和平行六面体之间的关系。目前 B3D 不允许两个形体之间存在重叠的关系, 这样对于很多复杂的结构很难描述好。

鉴于此, 我们提出了允许两个形体之间存在重叠的关系以方便描述复杂的结构。提出重叠功能后, 有两个方案实现它。

第一个方案是沿着以前的处理方法, 在预处理时处理形体间的重叠部分, 使处理完后的形体不存在重叠的关系, 然后再按以前的程序做。

这样做的一个难点是如何处理形体间的重叠部分。

第二个方案是重新寻求一整套形体运算的方法, 充分利用形体间关系的不确定性, 在时间上和空间上加以控制, 实现重叠功能。

这样做的难度很大, 需要寻找一些快速复杂的形体运算算法。

但是这样做的一个最大的好处在于我们不再处于被动的局面 (工艺上出现一种新的结构我们就得为这种新的结构设计一套新的做法)。

## 问题的提出:

我们把实际的问题抽象成数学模型为:

输入  $m$  个同方向柱体, 给出他们之间的关系, 包括包含, 相交, 以及相交的具体信息。我们把问题简化一步, 即这  $m$  个柱体的底面都位于  $OXY$  平面, 向  $Z$  正方向延伸一个任意高度。

我们的 Project 的内容就是:

输入  $m$  个凸多边形, 在比较快的时间内计算出这些多边形的相互关系。

这些相互关系包括一个多边形与那些多边形相交, 相交在什么位置, 一个多边形的一部分被哪些其他多边形所覆盖。

## 问题的解决:

输入  $n$  个凸多边形, 可以在  $O(m \cdot \log m + I)$  时间复杂度内计算出任何

一个多边形和哪些多边形相交，以及这些交点的具体位置，每个交点由哪些多边形的哪些边产生。以及任何多边形边的一点由哪些多边形覆盖它。其中  $I$  为交点的数目,  $m$  为输入多边形的边的个数。

## 数据结构:

我们采用类似 DCEL 的结构。

Point	屏幕上的点
PolygonVertex	多边形的顶点
PolygonEdge	多边形的边
PolygonArc	多边形的段
InterPoint	不同多边形边与边的交点

所有多边形都是按顺时针方向。这样我们就定义了多边形边的方向，多边形段的方向。每个多边形边都有起点和终点之分，它的起点和终点是多边形的顶点。

同时记录它上面的交点和它所属的多边形。

每一多边形的段都有起点和终点之分，它的起点和终点是边与边的交点，并且在其内再没有其他交点。同时记录它上面的顶点和它所属的多边形。

以及覆盖这个段的所有其他多边形的列表。

多边形的顶点记录以他开始的边和结束的边，同时记录它所属的多边形和多边形的段。交点为不同多边形边与边的交点，记录产生这个交点的边列表，以及多边形段的列表。每个多边形按顺时针记录它所有的顶点，边，段。

这样我们可以通过任何一个元素快速的访问到与其相关的任何元素。

## 设计模式:

程序以 Java 语言开发，分 Application 和 Applet 两种模式。

Application 可以访问本地文件系统，但必须由 Sun 公司的 JDK1.2 以上的版本作支持。

Applet 不能访问任何本地的资源。但可以用 IE 直接浏览。

整个程序分为两层，GUI 控制和算法控制。即界面和数据完全分开。

再算法部分主要分为三部分。

- 一 是求出所有的交点，构造多边形的段。
- 二 是求这些多边形的并（比目前可查文献的算法好的很多[1]）。
- 三 是构造段与多边形的覆盖关系。

同时程序分运行模式和演示模式两种。演示模式可以清楚的看到算法的每一个过程。

## 算法描述:

求交点部分我们使用典型的 Bentley and Ottmann's 扫描线算法。

然后再  $O(I)$  的时间内构造出所有的多边形的段。

在求多边形的并时我们采用扫描线算法和沿段扫描的算法。  
用扫描线算法检测新的连通域和判断新的有效连通域，  
用沿段扫描算法求给定连通域的外包。这里的细节可以再演示模式下观看。

构造段与多边形的覆盖关系的算法是这次 Project 的核心。  
我们采用扫描线算法和沿段扫描的算法。  
用扫描线算法检测新的连通域和判断新的连通域，  
用沿段扫描算法求每个段和其他多边形的关系。  
这里的沿段扫描算法要比上面提到的沿段扫描算法求并的负责的多。  
它采用类似 Jarvis March 求极线和 Lee83 Scan 求凸包的扫描算法。  
这里的细节可以再演示模式下观看。

## 算法的实现：

目前算法基本实现，但没有实现所有的退化情况。  
退化情况中多条边交于一点的情况还没有实现，但在实际应用中，  
由于给定的点不是整点，这种情况几乎不存在。  
在演示中，对每个用鼠标选择的点我们只要给一个小的扰动，就可以避免这种情况。