

海量数据多边形布尔运算的区域分割算法

姚辉学 卢章平

(江苏大学机械工程学院图形技术研究所, 镇江 212013)

摘要 研究了一种大数据量的2维多边形,称为海量数据多边形,其边界由直线段和圆弧段组成,顶点很多,形状复杂。本文设计并实现了海量数据多边形之间的布尔运算算法。首先用区域划分的方法快速求解出两多边形的交点,然后判断各交点的出点、入点性质,最后沿着多边形的边界跟踪,并根据并、交、差的不同要求切换跟踪路线,直到形成封闭的回路。运算结果仍为2维多边形。该算法从实用的角度出发,避免繁琐的数学推理,易于程序实现,在实际应用中显著提高了布尔运算的效率。

关键词 多边形 布尔运算 LCD设计 交集

中图法分类号: TP391.9 文献标识码: A 文章编号: 1006-8961(2007)03-0552-06

Area Partitioning Algorithm of Boolean Operations on Highly Complicated 2D Polygons

YAO Hui-xue, LU Zhang-ping

(Institute of Graphics Technology, School of Mechanical Engineering, Jiangsu University, Zhenjiang 212013)

Abstract In this paper, Boolean operations on a kind of highly complicated 2D polygons having a large amount of data is concerned. Each polygon is composed of line segments and arc segments, which has a great number of vertices and is of a complicated geometric shape. This algorithm of Boolean operations is introduced specially for these 2D polygons. By area partitioning, the algorithm calculates the intersection points of two polygons. Then it determines the crossing condition at the position of each intersection point, and traces the edges of polygons to form a closed cycle according to the requirement of union, intersection or subtract operation. The results are also 2D polygons. The algorithm is designed from the aspect of practicality, avoiding the complex mathematic reasoning. It can be coded conveniently and improves the efficiency of Boolean operations greatly in practice.

Keywords polygons, Boolean operation, LCD design, intersection

1 引言

大数据量的2维多边形布尔运算在工程中并不多见,但在一些特殊行业中却成为必须要解决的问题,如在游戏机液晶显示器(LCD)计算机模拟系统中,上下电极板中的电极图形就使用这种大数据量的2维多边形来表示。图1所示为一个相对简单的LCD上电极板中的一个电极(多边形A)和下电极

板中的一个电极(多边形B),当两电极通电时,其重叠部分之间的液晶将发光,这里所谓的重叠部分,就是两电极多边形的交集。

根据实际情况,这种多边形(代表电极)不允许自交,但多边形的数据量很大,其边界由直线段或圆弧段组成,顶点和边数很多,形状复杂且没有规律,本文称其为海量数据多边形。例如,在图1中,多边形A由333条边组成,包括288条直线段和45条圆弧段,多边形B由388条边组成,包括188条直线段

收稿日期:2005-05-23; 改回日期:2005-12-05

第一作者简介:姚辉学(1973~),男,讲师。2002年于甘肃工业大学获机械设计及理论专业工学硕士学位。主要研究方向为计算机图形学和CAD及工程图纸处理。E-mail: yhx_jxy@ujs.edu.cn

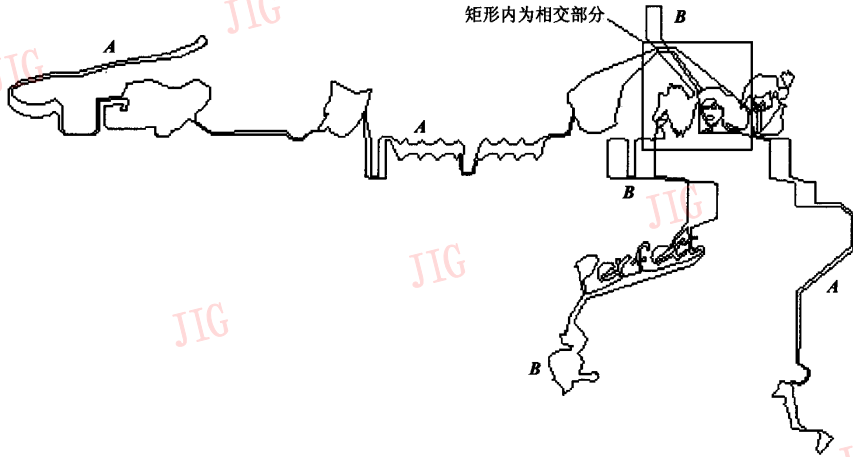


图 1 多边形 A 和 B

Fig. 1 Polygon A and B

和 200 条圆弧段,并且 A 和 B 的形状都很复杂。

对于 2 维多边形的布尔运算,已经有很多成熟的算法^[1-6],还有商业化图形软件中的布尔运算等。这些算法的执行时间主要消耗在多边形的三角片划分^[1,4]或求解交点^[1-5]的运算中,所涉及的多边形数据量不能太大;或者对多边形加了种种限制条件,如多边形必须为凸多边形,多边形的边必须为直线段^[1,5]等等。而对于图 1 所示的海量数据多边形,这些算法的时间复杂度将会很大,在实际应用中运算时间过长,占用内存过大,甚至不能完成运算。

本文针对海量数据多边形,充分研究了算法的高效率和低存储量要求,利用分治法的算法设计思想,设计出一种能够实现这种大数据量 2 维多边形之间布尔运算的算法。

2 算法设计和实现

根据算法设计的有穷性、正确性和可行性要求,本文所研究的布尔运算算法,其基本设计思想如下:

- (1) 求解两多边形之间的交点,形成交集;
- (2) 判断各交点处一个多边形相对于另一个多边形的出点、入点性质;
- (3) 从一个多边形的起点出发,跟踪边界,根据并、交、差的不同要求,构造结果多边形。

上述设计思想对于任意简单的多边形都适应,但本文的研究对象是海量数据多边形(图 1),必须考虑算法的高效率与低存储量要求,所以,这里使用区域划分的方法来求解两多边形之间的交点。

2.1 用区域划分的方法求交点

仔细研究图 1 中的多边形 A 和 B,会发现:虽然多边形的形状很复杂,所占据的空间区域也很大,但是两多边形相交的部分并不复杂,相交的区域也不大,交点就位于这个并不大的有效区域内。把多边形 A 和 B 的相交部分放大,如图 2 所示。

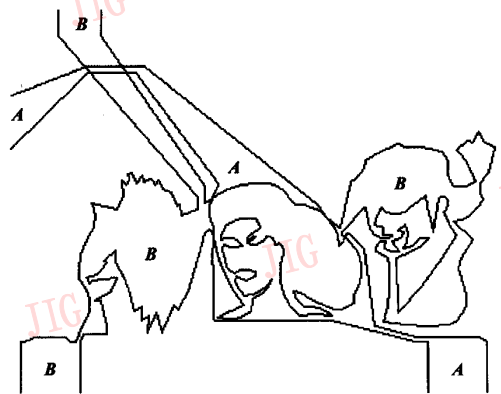


图 2 多边形 A 和 B 相交部分的放大图

Fig. 2 Magnifying of intersection area of polygon A and B

(1) 矩形网格区域划分

为了迅速接近有效区域,首先计算出两多边形的包围矩形^[7,8],然后计算出两包围矩形的重叠区域,即公共矩形,那么求交点的运算就只涉及到两多边形位于公共矩形内的边(包括与公共矩形边界相交的边),这样就首先去除了两多边形明显不相交的部分,一定程度上减小了求交点的运算量。但是由于多边形的数据量很大,位于公共矩形内的边仍然很多,所以,本算法对公共矩形进行网格区域划

分,目的是把一个公共矩形内的较大规模的求交点运算分解成多个矩形网格内的小规模的求交点运算

来完成,小问题解决了,大问题也就解决了。区域划分的示意图如图 3 所示。

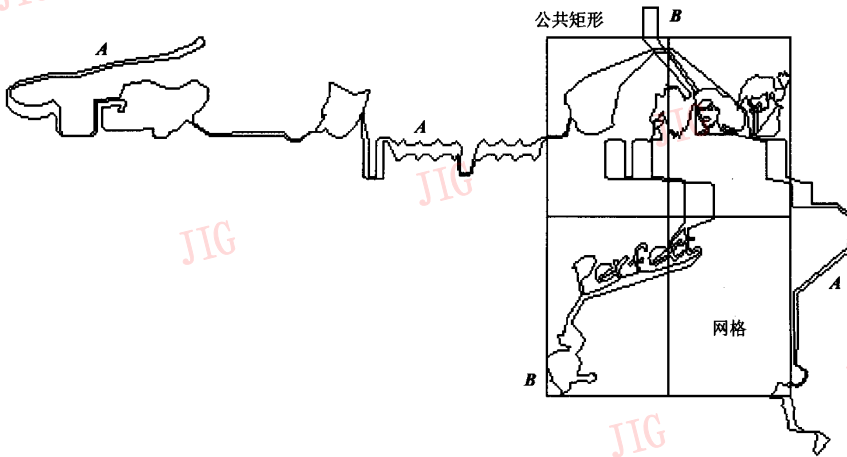


图 3 区域划分示意图

Fig.3 Example of area partitioning

为了方便,假定求交点运算在公共矩形内是均匀分布的,所以对公共矩形进行等分即可。由于划分网格后要计算多边形落在各个网格内的边的集合,所以网格并不是划分得越多越好,如果网格划分得过多,计算多边形落在各个网格内的边的运算就会消耗过多的时间,从而失去了算法设计的意义。经过测试和实践,网格内的平均边数阈值为 80 ~ 100 时,运算达到最佳效果。这样,A 和 B 较大的一个边数除以边数阈值所得的商(整数化)就是网格的个数。

网格划分后,求出两多边形落在各个矩形网格内的边的集合,并记录。需要注意的是,为了避免交点被遗漏,网格内边的集合既要包括完全落在该网格内的边,还要包括与网格边界相交的边。为了节省判断一条边是否落在一个网格内的计算时间,只需要判断边的包围矩形与网格矩形是否重叠即可。

(2) 依次在网格内求交点,构造交点集

网格内求交点,就是网格内多边形 A 的边集 E_A 与多边形 B 的边集 E_B 之间求交点,这里使用边边求交算法来计算交点。由于多边形的边可以是直线段或圆弧段,所以边与边的求交包括 3 个基本的求交算法:直线段与直线段求交、圆弧段与圆弧段求交和直线段与圆弧段求交。这 3 个基本算法已经十分成熟^[7],本程序把它们作为模块来使用。

交点集可以保存在链表中。在边与边的求交运

算中,如果边与边重合,则认为两边之间没有交点。上述方法计算出的交点集中可能会出现重复交点,所以需要对交点集进行整理,消除重复交点。如果交点的几何坐标相等,则认为这几个交点是重复的,根据重复交点在链表中的位置,只保留第一个交点,把其后的重复交点全部删除。

2.2 判断各交点的出点、入点性质

交点的出点、入点^[5]性质,是指在交点处,多边形 A 是进入多边形 B 还是从多边形 B 中出来等的相交情况。本算法使用微圆法完成判断。微圆法示意图如图 4 所示。

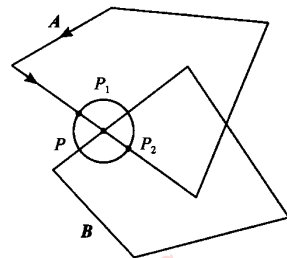


图 4 微圆法示意图

Fig.4 Example of small-circle method

假设多边形 A 和多边形 B 的一个交点为 P,以 P 点为圆心,以 radius 为半径画一个微圆,该微圆与多边形 A 的边交于 P_1 和 P_2 两点,并且 P_1 点在 P_2 点的前面(根据点所在的边的序号大小或两点到顶点的距离大小来判断),然后分别判断 P_1 和 P_2 两点

相对于多边形 B 的位置关系(在多边形内,在多边形外,在边界上),根据位置关系就可以确定交点的出点、入点性质。

微圆法与多边形的方向(顺时针还是逆时针)没有关系,只与多边形的顶点顺序有关。

微圆的半径 $radius$ 称为微圆法的精度,是一个很小的 $double$ 型数值,它必须小于单个多边形上任意两条不相邻的边之间的最小距离,才能保证判断的正确性。不相邻两边的最小距离在实际应用中具有具体的要求,例如,对于电极图形,这个最小距离用来保证电极的电阻不能太大,以使电流畅通。微圆法中还用到了点与平面图形位置关系的判断算法,这类算法已有很多^[7,8],本文不再详述。

2.3 交点集排序

交点集在多边形上进行排序,是指从多边形的入口点开始,根据交点在多边形上的边的序号由小到大进行排序,有时一条边上出现多个交点,则依据交点到该边起点之间的边的长度的大小进行排序。排序的算法比较简单,排序的结果是原交点集的两个副本,一个副本是对多边形 A 排序的,另一个副本是对多边形 B 排序的。

排序是为后面的边界跟踪做准备的。

2.4 跟踪边界,构造结果多边形

边界跟踪算法的基本思想是:在交点集排序的基础上,从一个多边形上的第一个交点出发,根据并、交、差运算的不同要求,选择正确的方向沿着多边形的边跟踪,当到达另一个交点时,判断是否把跟踪路线切换到另一个多边形,并选择正确的方向沿着相应多边形的边继续跟踪,直至又回到第 1 个交点,形成一条完整的回路^[5,8]。如果交点集中仍有未被经过的剩余交点,则继续按照同样的方法处理剩余的交点,直到所有的交点都被经过。处理完毕得到的所有多边形就是布尔运算的结果。

在边界跟踪算法中,跟踪方向的选择是最关键的,跟踪方向是由交点出点、入点性质所控制的。

下面以图 5 所示的多边形 A 和多边形 B 为例,来说明边界跟踪法的具体实现。已知条件是多边形 A 和多边形 B ,点 A_0 为 A 的入口,点 B_0 为 B 的入口。

则,一个结果多边形为

(1) 并运算 $A \cup B, P_0 \rightarrow A_1 \rightarrow A_0 \rightarrow A_4 \rightarrow P_3 \rightarrow B_5 \rightarrow B_6 \rightarrow B_0 \rightarrow B_1 \rightarrow P_0$;

(2) 交运算 $A \cap B, P_0 \rightarrow A_2 \rightarrow P_1 \rightarrow B_2 \rightarrow P_0$;

(3) 差运算 $A - B, P_0 \rightarrow A_1 \rightarrow A_0 \rightarrow A_4 \rightarrow P_3 \rightarrow B_4 \rightarrow P_2 \rightarrow P_1 \rightarrow B_2 \rightarrow P_0$ 。

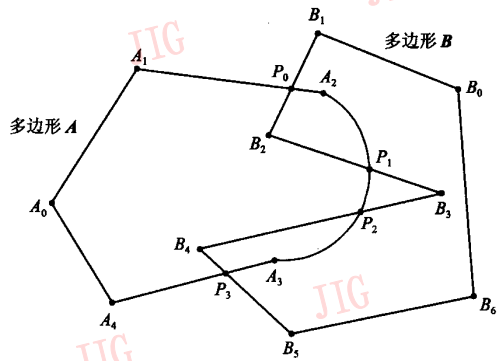


图 5 边界跟踪法示例

Fig. 5 Example of edge tracting method

检查是否有未被经过的剩余交点,若有,继续使用边界跟踪法处理剩余交点,直到所有交点都被经过。对于图 5,另一个结果多边形为

(1) 并运算 $A \cup B, P_1 \rightarrow P_2 \rightarrow B_3 \rightarrow P_1$;

(2) 交运算 $A \cap B, P_2 \rightarrow A_3 \rightarrow P_3 \rightarrow B_4 \rightarrow P_2$;

(3) 差运算 $A - B$,所有交点都已经过,结束。

所有的结果多边形成为布尔运算的结果,图 6 所示为图 5 的布尔运算结果。

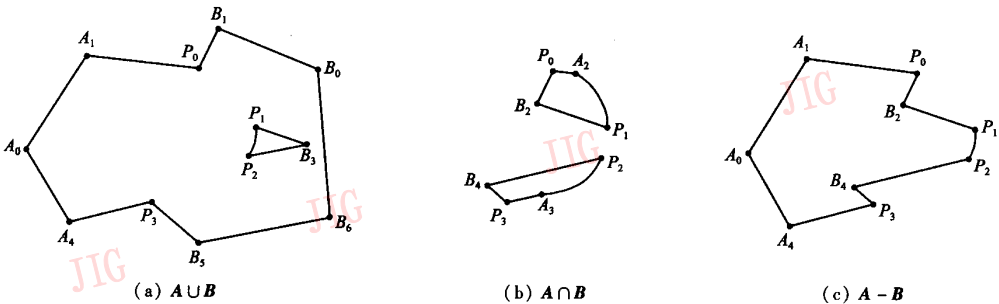


图 6 布尔运算的结果

Fig. 6 Results of Boolean operations

3 算法分析

3.1 时间复杂度

从算法的实现过程可知,算法的执行时间(T)主要由以下 5 部分决定:一是把多边形的边分配到各个网格中(时间记为 T_1);二是在各个网格中执行边与边的求交运算并循环各个网格(时间记为 T_2);三是微圆法判断交点的出点、入点性质(时间记为 T_3);四是交点集排序(时间记为 T_4);五是边界跟踪(时间记为 T_5),则

$$T = T_1 + T_2 + T_3 + T_4 + T_5,$$

其中,由于交点的个数很少,所以 T_3 和 T_4 相对较小,在算法时间复杂度中可以不考虑。

假设多边形 A 的顶点个数(边数)为 m ,多边形 B 的顶点个数(边数)为 n ,网格划分的个数为 g , T_1 中的基本操作是判断一条边(直线或圆弧)是否落在一个矩形网格内,假设这一判断所需要的执行时间为 K_1 ,则

$$T_1 = (m + n) \times K_1 \times g$$

假设多边形的边被平均分配到 g 个网格中,那么每个网格中的边数分别为 $\frac{m}{g}$ 和 $\frac{n}{g}$, T_2 中的基本操作是边与边(直线或圆弧)的求交运算,假设这一基本操作所需要的执行时间为 K_2 ,那么每一个网格内的执行时间为 $\frac{m}{g} \times \frac{n}{g} \times K_2$,则时间 T_2 为

$$T_2 = g \times \frac{m}{g} \times \frac{n}{g} \times K_2 = \frac{mn}{g} \times K_2$$

T_5 中的基本操作是提取多边形 A 和 B 的一条边(顶点数据)来构造结果多边形的一条边,并且对多边形只循环一次,假设这一基本操作所需要的执行时间为 K_5 ,则

$$T_5 = (m + n) \times K_5$$

相对于 T_1 和 T_5 , T_2 是主要时间,这是因为边与边的基本求交运算其时间复杂度要大得多,所以,可以把 T_5 省略,在网格没有划分得过小过多得情况下,可以把 T_1 省略,则整个算法的时间复杂度为

$$T(mn) = O\left(\frac{mn}{g}\right)$$

3.2 空间复杂度

从算法的实现过程可知,算法的空间复杂度与时间复杂度是类似的,算法执行时主要的内存占用

是在各个网格中执行边与边的求交运算并循环各个网格。整个算法的空间复杂度为

$$S(mn) = O\left(\frac{mn}{g}\right)$$

4 应用实例分析

本文算法的一个应用实例是 LCD 计算机模拟系统,用海量数据多边形表示电极图形,分别放置在两个图层(上下电极板)中。电极图形是包含孤岛的,所以要对本文算法进行简单的扩展,使其可以解决带孔洞的多边形之间的交集运算,此时,海量数据 2 维多边形为外部轮廓,内部包含多个孤岛(也是海量数据多边形),则

(1) 两外部轮廓进行交集运算,得到交集结果多边形;

(2) 孤岛与孤岛之间进行并集运算,如果孤岛结果多边形之间还有相交,继续并运算;

(3) 交集结果多边形与孤岛并集结果多边形之间进行差运算,得到最终结果。

LCD 计算机模拟是 LCD 计算机辅助设计的最关键的环节,其实现方法就是电极板之间的电极进行两两布尔运算。根据模拟显示结果,可以观察 LCD 的实际运行效果,检查设计中的错误,从而肯定或否定设计方案。本文的布尔运算算法十分理想地完成了这一工作。下面为电子表 LCD(图 7)和格斗游戏 LCD(图 8)等两个实例。

在 AMD-K6-300-64M-RAM 的计算机中,完成 LCD 模拟显示并对运算结果进行填充,图 7 的电子表模拟用时 5 ~ 6min,图 8 的格斗游戏模拟用时 12 ~ 15min,模拟结果完全正确。而使用已有的布尔运算算法基本上无法完成模拟显示,试对已有的布尔运算算法进行试验,运行了一个晚上结果还没出来,检查内存发现内存占用 100%。这也是设计本文算法的原因。

5 结论

本文所设计的算法,充分考虑了处理较大数据量的问题所需要的高效率与低存储量要求,很好地解决了海量数据多边形的布尔运算问题。对于图 1 所示的多边形 A 和 B ,本文算法的交集运算结果如图 9 所示。

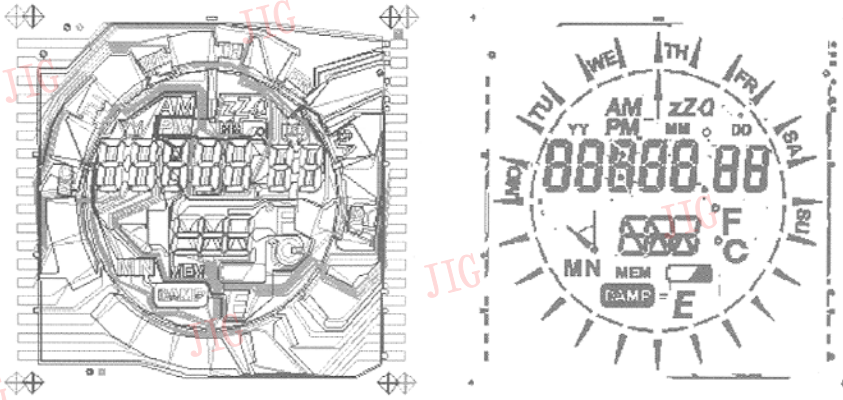


图 7 电子表 LCD 及其模拟结果填充

Fig. 7 LCD drawing of a electronic watch and a hatch of its display imitation

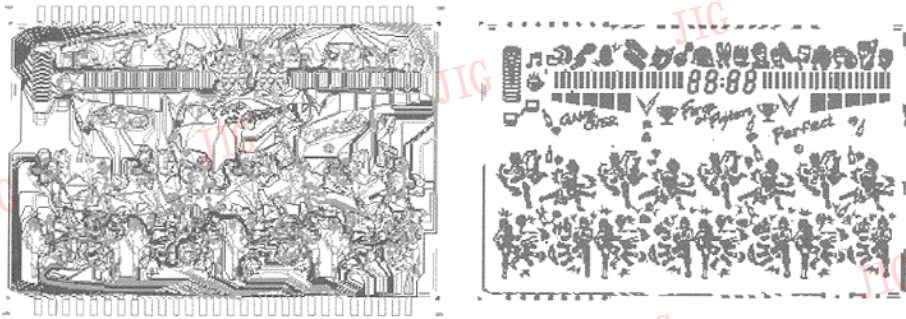


图 8 格斗游戏 LCD 及其模拟结果填充

Fig. 8 LCD drawing of a playing machine screen and a hatch of its display imitation



图 9 海量数据多边形的交集运算结果

Fig. 9 Intersection of the two highly complicated 2D polygons

参考文献 (References)

1 Rivero M, Feito F R. Boolean operations on general planar polygons [J]. Computer & Graphics, 2000, 24(6): 881 ~ 896.

2 Ruiz J, de Miras, Feito F R. Inclusion test for curved-edged polygons [J]. Computers & Graphics, 1997, 21(6): 815 ~ 824.

3 Feito F, Rivero M L, Rueda A J. Boolean representations of general planar polygons [A]. In: Proceedings of the 7th International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media [C], Plzen-Bory, Czech Republic, 1999: 87 ~ 92.

4 Dong Wei-ming, Maira Ba-bang, Zhou Deng-wen, et al. Boolean operations on general planar curved-Edge Polygons [J]. Journal of Computer-aided Design & Computer Graphics, 2003, 15 (9): 1134 ~ 1140. [董未名, 玛依拉·巴榜, 周登文等. 平面扩展简单多边形的布尔运算[J]. 计算机辅助设计与图形学学报, 2003, 15(9): 1134 ~ 1140.]

5 Xie Bu-ying, Zhang Yan. The boolean operation for 2D figure with segment-based method and linked-list method [J]. Journal of Engineering Graphics, 2003, 24(2): 78 ~ 84. [谢步瀛, 张岩. 用分段法与链表法的二维布尔运算[J]. 工程图学学报, 2003, 24(2): 78 ~ 84.]

6 Liu Hong-jun, Wang Cong-jun, Huang Shu-huai. Boolean operation for polygon with holes [J]. Journal of Huazhong University of Science & Technology (Nature Science Edition), 2003, 31(8): 18 ~ 20. [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.]

7 Sun Jia-guang. Computer Graphics (3rd ed) [M]. Beijing: Tsinghua University Press, 1998: 393 ~ 463. [孙家广等编著. 计算机图形学(第3版)[M]. 北京:清华大学出版社, 1998: 393 ~ 463.]

8 Zhou Peide. Computational Geometry-The Analysis of Algorithms and Design [M]. Beijing: Tsinghua University Press, 1999: 133 ~ 176. [周培德著. 计算几何——算法分析与设计[M]. 北京:清华大学出版社, 1999: 133 ~ 176.]