

适用于凹多边形的 Cyrus-Beck 改进算法

陈 涛

(南京大学计算机科学与技术系 南京 210093)

摘要 本文对目前常用的二维线段裁剪算法进行分析,提出了一种基于 Cyrus-Beck 算法的改进算法,使其能够扩展到对凹多边形的处理,通过对线段与裁剪窗口位置关系的严格判断将求交次数减到最少,并且通过对交点性质的判断来识别出线段的可见部分。理论分析和实验结果均表明该算法优于目前处理任意多边形裁剪框的算法。

关键词 凹多边形,线裁剪,算法,改进

An Improved Algorithm of Cyrus-Beck Segment Clipping to Process Concave Polygon

CHEN Tao

(Department of Computer Science and Technology, Nanjing University, Nanjing 210093)

Abstract Based on the analysis of currently popular algorithms for segment clipping, in this paper, an improvement in the algorithm of Cyrus-Beck segment is proposed, which extends the algorithm to the range of concave polygon. The calculation of the intersection points is decreased to the minimum by the means of verifying the position of the line to be clipped against the polygon. And marking those intersection points with "in" or "out" to decide which sub-line between the two intersection points is visible. Both theoretic analysis and example testing show that the new algorithm is better than well known algorithms.

Keywords Concave polygon, Line clipping, Algorithm, Amelioration

1 引言

图形的裁剪是计算机图形学中许多重要问题的基础,裁剪的应用包括:从定义的场景中抽取出于观察的部分;在三维视图中标识出可见面;防止线段或对象的边界混淆;用实体造型来创建对象;显示多窗口的环境;允许选择图形的一部分来进行拷贝,移动或删除等绘图操作。裁剪对象可以是二维的点,线段或者是封闭的多边形,也可以是三维的多面体。由于三维裁剪可以由二维裁剪扩展得到,因此二维裁剪在图形裁剪中有重要的地位。

而在二维图像裁剪中,线段裁剪又是最基本的内容,很多适用于二维线段裁剪的算法通过扩展同样都可以用来处理多边形的裁剪。因此选择二维图形裁剪作为研究课题在图形裁剪领域有重要的意义。线段裁剪的基本目的是判断线段是否完全或部分位于所限定的区域(称为裁剪窗口)内,然后求出窗口内部分。对于线段裁剪,采用有效地算法在减少计算量的基础上,快速有效的确定给定线段与裁剪窗口的相对位置关系是其相关算法需要解决的问题。这些算法的速度和精度直接影响着图形裁剪的速度和精度,因此如何减少运算量,提高运行效率,成为这些算法设计中的核心问题。

线段的裁剪技术发展至今已出现了许多有效的算法,其中最具代表性的有:基于区域编码的 Cohen-Sutherland 算法^[1];易于实现的中点分割算法^[2];采用参数形式的梁友栋-Barsky 算法^[3];适用于多边形窗口的 Cyrus-Beck 算法^[4];基于区域分割的 Nicholl-Lee-Nicholl 算法^[5]等。然而这些算法在处理裁减问题的时候都有其局限性,即只适应于特定的裁减窗口,例如 CS 算法只适用于裁剪窗口是矩形的情况,梁友栋-Barsky 算法是 Cyrus-Beck 的特殊情况运用,是参数化的裁剪算法在矩形裁剪窗口的应用,Cyrus-Beck 算法是早期为处理凸多边形裁剪窗口而提出的裁剪算法。而对于凹多边形

形,目前没有通用的算法^[6,7],通常采取分割凹多边形为凸多边形的方法来处理,如延长线分割法和旋转分割法等等。而本文的目的就是提出一种比较快速有效的适用于任意多边形的裁剪算法。

一些典型的关于凹多边形窗口的解决方案是采用分割成凸多边形的方法(如图 1),利用现有的证明为有效的凸多边形算法来解决这个问题。但是,这种分割凹多边形的方法,在分割的时候需要人为地加上几条边,或者是几个点,这样,增加了额外的求交点的运算,假设一个锯齿形的凹多边形(如图 2),即使是最有效的分割方式也至少要求 4 个交点,若凹点再增加则分割产生的交点数就会随之而增加。并且,事实上,对于只穿过多边形内部,并不与锯齿相交的线段来说,利用分割法对其进行裁剪求交集显然是增加了许多裁剪次数,因而导致效率低下。

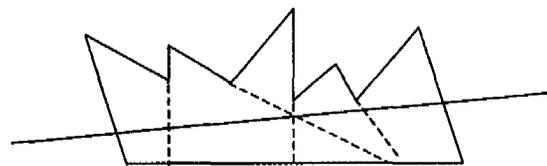


图 1 凹多边形的分割

而后有人提出一种整体裁剪法^[8]即直接将线段与多边形求交点(如图 2),然后顺次判断每一段交点之间的中点是否在多边形中,这种算法实现简单但难以高效地得到结果。一般判断一个点是否在一个封闭区内是通过该点做一条射线,计算与封闭区域的交点数量,奇数个则在封闭区内,偶数个则在外面。这种方法要进行大量的求交点运算,即便是在做与裁剪线段无关的判断点与区域关系时,仍需要通过构造射线来求交点数量的方法来实现。大量的求交运算使得该算法效率低下。另外当线段穿过的多边形的边数很多时,就需要相

应的大量的判断来识别中点和多边形区域的关系。

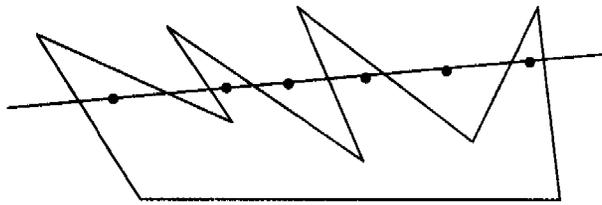


图2 整体裁减法

2 算法描述

为了能够处理各种多边形,特别是凹多边形,通过研究 Cyrus-Beck 算法的特点,并将其修改扩充,提出本算法使之能够适应任何多边形剪框。Cyrus-Beck 算法之所以不能运用到凹多边形,是由于它采用逐边裁剪的方法,求交的同时就进行裁剪,因此只能用来求线段与多边形两个交点,这个对于凸多边形来说是适合的。但是对于凹多边形来说,显然是不适应的。因此我们改进算法的相关步骤,引进数组存放多个交点的参数,重要的是要将这些交点的状态,也就是入点(即线段通过该点从多边形外部进入内部)还是出点同时进行标记并且保存,最后在判断哪些交点之间的线段是可见的哪些是不可见的时候,就利用这些交点的状态来进行判断。

2.1 判断凹多边形及其凹点

对于多边形进行判断,按照任意的一个方向遍历多边形的每一条边,找出并记录下凹点(所谓凹点就是使该多边形成为凹多边形的顶点,如图3)。

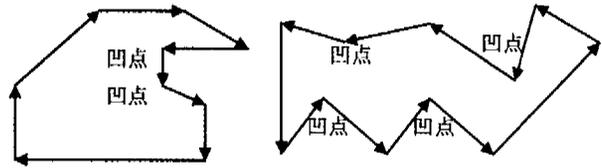
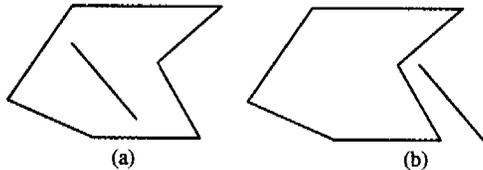


图3 凹点的识别

判断一个多边形是否是凹多边形只要绕多边形的边界计算相邻边界向量的叉乘结果:如果一些叉乘结果的z分量为



正,而另一些z分量为负,则该多边形为凹多边形;否则为凸多边形。

要找出相应的凹点,运用计算几何中向量乘法的性质,对于向量 $\vec{u}(u_x, u_y, u_z), \vec{v}(v_x, v_y, v_z)$

$$u \times v = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ u_x & u_y & u_z \\ v_x & v_y & v_z \end{vmatrix}$$

可以计算出: $(u_y v_z - u_z v_y)\vec{i} + (u_z v_x - u_x v_z)\vec{j} + (u_x v_y - u_y v_x)\vec{k}$

根据z分量的符号可以判断出向量u和v的位置关系。由于凸点相邻两条边向量叉乘结果的z分量总是与凹点相邻两条边向量叉乘结果的z分量异号的,则找到一个凸点,就能找到所有的凹点。而将任意多边形置于坐标系中时,根据多边形性质,X坐标最大的那个多边形端点一定是凸点,因此对于每一个多边形都可以迅速地找到一个凸点,也就能够找出多边形上的每一个凹点。

2.2 判断线段与多边形位置关系并计算交点

判断被裁剪线段与裁剪框的位置关系,通过分析对与任意的多边形裁剪框,线段与裁剪框的位置关系只有以下几种(如图4):

- (1) 线段与裁剪窗口不相交
 - ① 线段完全在窗口内 如图4(a)
 - ② 线段完全在窗口外 如图4(b)
- (2) 线段与裁剪窗口相交
 - ① 有向线段从外部进入多边形内部 如图4(c)
 - ② 有向线段从内部跨出多边形外部 如图4(d)

对于线段与裁剪框位置关系的确定,只需要依次遍历多边形的每一条边,判断线段与多边形边界的相交情况,而对于线段之间的相交情况计算几何中有相关的方法能够有效地作出判断。在本算法中采用快速排斥法和向量法相结合的方式:

首先用快速排斥法快速有效地排除明显不相交的线段,分别以两条线段的端点画矩形,若两个矩形没有公共区域,则两条线段没有交点,如图5。

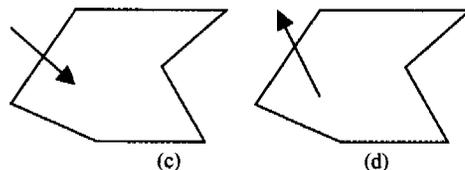


图4 线段与裁减框的位置关系

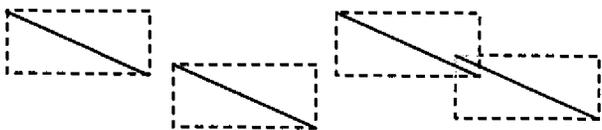


图5 快速排斥法排除线段相交

若两线段的位置关系不能用快速排斥法确定,则对于线段 P_1P_2, Q_1Q_2 ,判断两条线段是否互相跨越对方,也就是当且仅当 P_1, P_2 分别在线段 Q_1Q_2 两边;同样 Q_1, Q_2 分别在线段 P_1P_2 两边的时候,两条线段才有交点,如图6。

计算公式如下:

$$A = P_1Q_1 \times P_1P_2 \cdot P_1P_2 \times P_1Q_2$$

$$B = Q_1P_1 \times Q_1Q_2 \cdot Q_1Q_2 \times Q_1P_2$$

则相交的充要条件是 $A \geq 0$ 且 $B \geq 0$

在确定相交情况后,对于线段穿过裁剪窗口的情况利用参数化求交法(Cyrus-Beck方法),计算出交点的参数。计算交点参数的过程与Cyrus-Beck算法基本一致,需要注意的是在利用内法向量求交的过程中,对于内法向量的计算需要结合边E两个端点的凸凹性来区别处理。判断到若该点是凹点,则利用参数化求交法所求得的内法向量 N_i ,必须求反才是实际的内法向量,如图7。

与Cyrus-Beck不同的是,该算法计算出的交点是实际的交点,即是被裁剪线段与裁剪边界线段实际的交点,这样,对于每一个交点,在利用 $N_i \cdot (P_2 - P_1)$ 判断线段与多边形边的关系时,记录该交点的状态如下(假设被裁剪线段为有向

线段 P_1P_2 , 如图 8)。

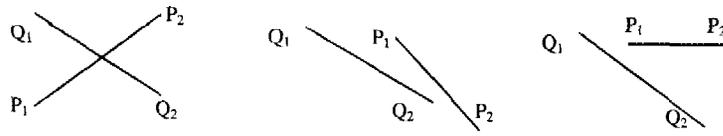


图 6 线段相交性判断



图 7 计算内法向量

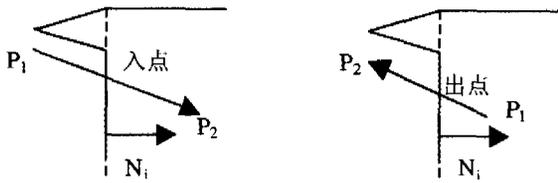


图 8 线段与边的关系

若 $N_i \cdot (P_2 - P_1) > 0$ 则该线段从多边形外部进入多边形内部, 相应的交点标记为入点;

若 $N_i \cdot (P_2 - P_1) < 0$ 则该线段从多边形内部离开多边形到外部, 相应的交点标记为出点。

2.3 线段裁剪

遍历多边形的每一条边, 将交点的参数和状态一起保存在一个数组中。把这个数组按照交点参数的大小从小到大排序, 得到一个交点参数从小到大的顺序(对于原来线段的起点和终点分别标为入和出)。则依次判断交点的状态, 取入出的一对交点作为一段可见线段的端点(如图 9)。

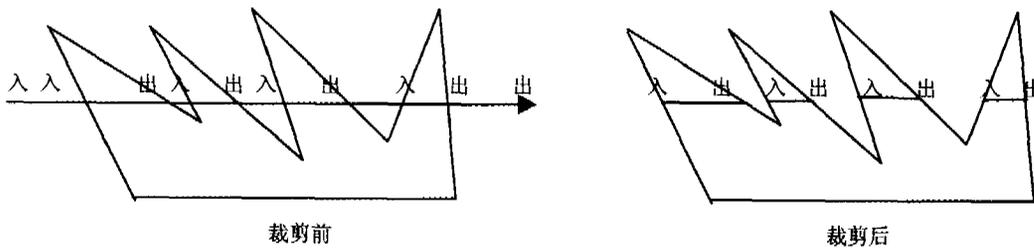


图 9 取可见线段

而对于一些特殊情况, 比如被裁剪线段经过多边形的顶点, 或者被裁剪线段经过多边形顶点, 在本算法中, 不需要单

独进行考虑, 按照以上步骤也可以进行裁剪, 这也是该算法不同于其它凹多边形裁剪算法之处(如图 10)。

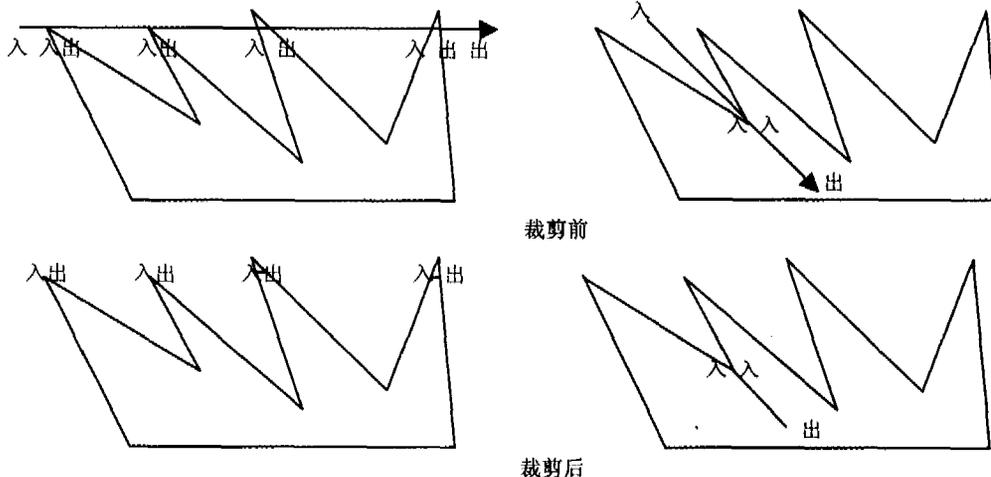


图 10 特殊情况

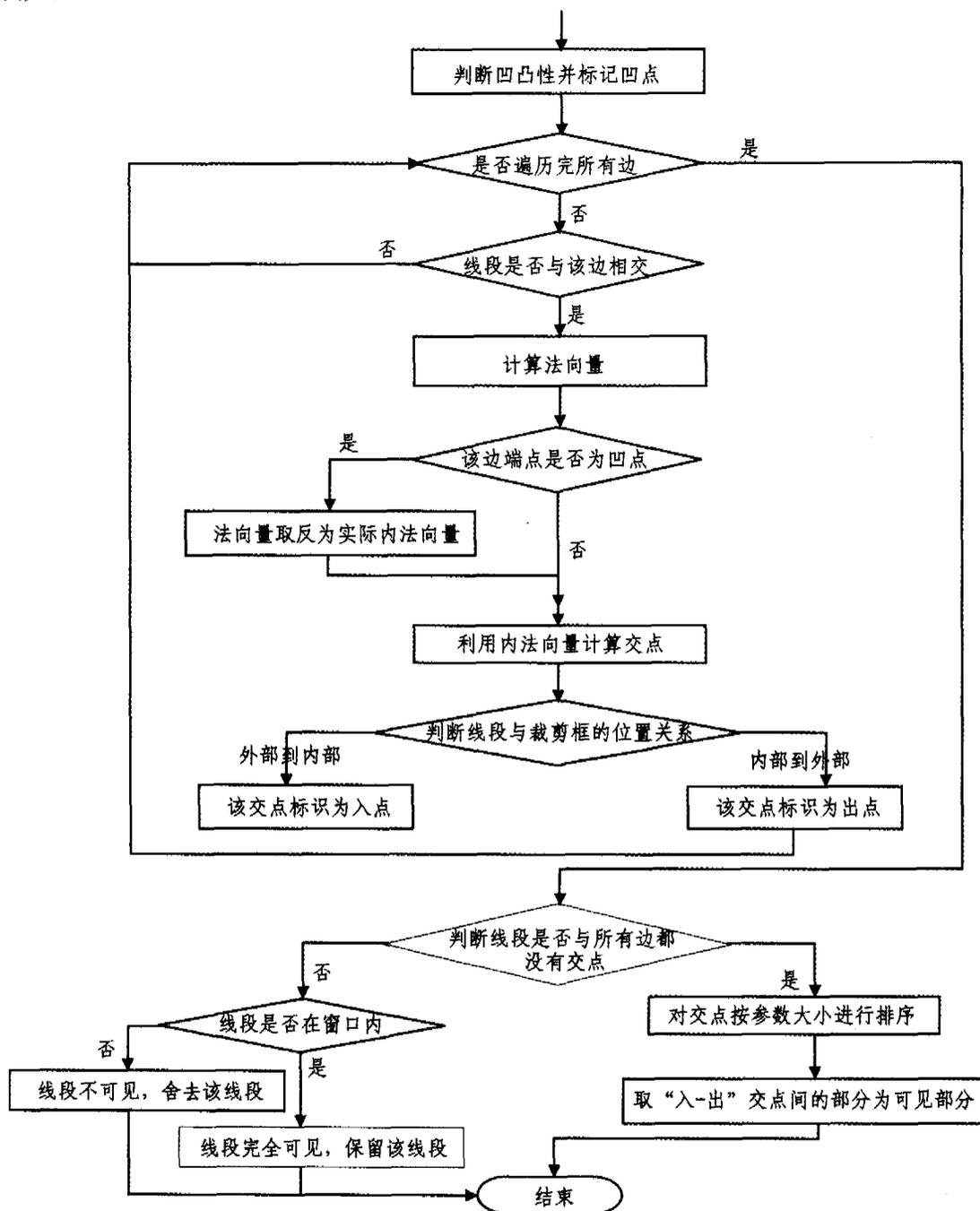
处理完线段与边界有交点的情况, 现在来讨论线段与边界没有交点的情况, 即线段完全位于多边形裁剪框内或者完全位于裁剪框外, 则只需要判断出线段上任意一点是否在多

边形裁剪框内就可以了。对于判断某点是否是在多边形内, 采用常用的方法来判断, 即过该点做一条射线, 若该射线与多边形窗口有奇数个交点, 则该点是在多边形内; 若是偶数个交点

则该点在多边形外。

3 算法实现

流程图如下：



4 算法的分析

在裁剪算法中,减少求交的次,尽量用判断来代替求交,是提高裁剪效率的主要途径。这主要是由于计算交点会多次用到除法,而除法在计算机中与乘法、加减法和比较运算相比效率要低许多。

本算法实际上是对 Cyrus-Beck 算法的扩展,因此继承了 Cyrus-Beck 算法的优点,利用参数化求交法,只用 1 次除法 2 次乘法就能解出交点的参数。本算法还通过相交性判断,避免了 Cyrus-Beck 算法中求虚交点参数的过程。因此即便当同样都处理凸多边形的时候,本算法也能够体现出优越性,例如对于图 11 所示的多边形,用 Cyrus-Beck 算法裁减的

过程要计算 6 次交点的参数值,而实际有用的参数交点个数仅为 2 个;而本算法的裁减过程中,通过简单的相交性判断只需要计算 2 次实际的交点就可以确定线段可见部分的端点。因此,当裁减窗口的边数为 n 时,Cyrus-Beck 算法需要对每条边都计算一次交点,即需要计算 n 个交点参数,而线段与裁减窗口的实际交点个数不会超过两个;而本算法无论裁减窗口有多少条边,都至多只需要计算两次交点参数值即可。

与现有的其它可以处理任意多边形窗口的算法比较,本算法也有其优点,与分割法相比,分割法需要先对裁减框进行

(下转第 224 页)

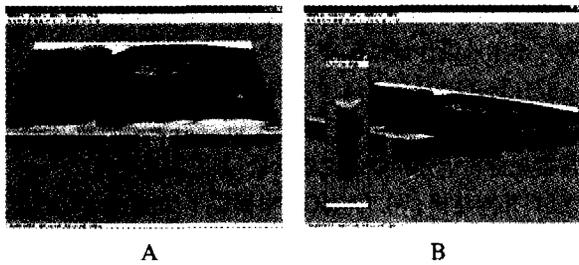


图6 三维模型数据表达

结论 分析了 CRUST51 全球地壳结构数据特征,提出了基于长方体与四棱柱相结合的三维可视化建模方法。利用 OpenGL 的强大的建模功能,建立了全球地壳结构数据的可视化模型,在 Delphi 中基于 OpenGL 三维图形库实现了 CRUST51 全球地壳结构数据的三维可视化表达。结合 OpenGL 的人机交互功能完成了多视角观察以及基于地壳剖切线的三维剖切功能,为仿真研究地震波传播规律以及精确地震走时计算提供了研究环境。

参考文献

- 1 Soller D R, Ray R D, Brown R D. A new global crustal thickness model. *Tectonics*, 1982(1): 125~149
- 2 Hahn A, Ahrendt H, Meyer J, Hufen J H. A model of magnetic sources within the Earth's crust compatible with the field measured by the satellite Magsat. *Geol Jahrb, Reihe A*, 1984, 75: 125~156
- 3 Tanimoto T. Crustal structure of the Earth. In: *Global Earth*

- Physics: A Handbook of Physical Constants, AGU Ref Shelf, vol 1, Ahrens T J. Ed. AGU, Washington, D C, 1995. 214~224
- 4 Nataf H C, Ricard Y. 3SMAC: An apriori tomographic model of the upper mantle based on geophysical modeling. *Phys Earth Planet Inter*, 1996(95): 101~122
- 5 Mooney W D, Laske G, Masters T G. CRUST 5. 1; A global crustal model at 5°×5°, *Journal of Geophysical Research*, 1998, 103(B1): 727~747
- 6 Serón F J. Visualization and finite element techniques for seismic interpretation. *Computer & Graphics in Spain*, 1992, 16(4): 383~394
- 7 Sheffield T M, Meyer D, Lees J, et al. Geovolume visualization interpretation; *Color in 3-D volumes*. The Leading Edge, 1999, 18(6): 668~674
- 8 Kidd G D. Fundamentals of 32D seismic volume visualization. *The Leading Edge*, 1999, 18(6): 702~709, 712
- 9 Dalley R M, 等著. 3D 图像/数据体处理新方法[J]. 王忠, 等译. 石油物探译丛, 1998, 1: 43~46
- 10 程朋根, 龚健雅, 史文中, 等. 基于似三棱柱体的地质体三维建模与应用研究[J]. *武汉大学学报(信息科学版)*, 2004, 29(4): 602~607
- 11 李清泉, 李德仁. 三维空间数据模型集成的概念框架研究[J]. *测绘学报*, 1998, 27(4): 325~330
- 12 健雅, 夏宗国. 矢量与栅格集成的三维数据模型[J]. *武汉测绘科技大学学报*, 1997, 22(1): 7~15
- 13 张焜, 白世伟. 一种基于三棱柱体体元的三维地层建模方法及应用[J]. *中国图形图象学报(A)*, 2001, 6(3): 285~290
- 14 朱亚平, 杨慧珠, 董渊, 等. OpenGL 技术在地震数据可视化中的应用[J]. *石油地球物理勘探*, 2000, 35(4): 403~414

(上接第 240 页)

相应的处理,将其分割成多个凸多边形,再用凸多边形的裁减法来裁减,这样即便是能利用计算几何中关于图形分割的方法找到最优的分割方案,在裁减过程中也要对线段进行多次不必要的裁剪,特别是对于在裁减框内部的线段,要进行多次不必要的裁剪,处理代价比较大;与中点法相比^[8],中点法在处理线段穿过多条边界时,判断中点是否在多边形内是一个多次求交的过程,而这些交点都是辅助交点,不能影响裁减结果。而本算法在处理凹多边形的时候只计算实际交点,因此不会对线段有多余的裁减和求交点过程。

(1)同样采用参数化方法求交,并且由于事先判断线段是否相交后对于有交点的才求交点,因此求交点的次数是最少的。虽然判断的过程中有一些运算,但是比起求交的运算来说还是比较快的,另外,当线段与多边形所交的点很少时,该算法的优势很明显,很快能够排除不在内部的部分。

(2)能够适用与任意的多边形,对于凸多边形,该算法的效率高于 Cyrus-Beck 算法的效率。

对于该算法,采用更简洁的线段相交性判断和减少判断凹点时的向量外积和内积的计算是进一步优化改进的方向所在。

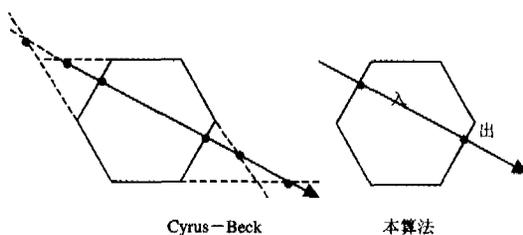


图 11

总结 本算法对多边形的各个顶点进行凹、凸判断,通过判断线段于多边形的交点是入点或出点的情况决定是否保留该线段,因此该算法可以对任意多边形窗口进行处理。算法中在对每条边进行判断的时候,先利用快速排斥法排除明显与边界不相交的线段,在通过相交判断(不要求交点)排除不相交的线段,只对相交的线段计算交点,有效地减少了交点的计算次数,并且不会产生参数的中间运算结果(即只计算了实际交点的次数,除去了延长线上交点的计算)。

该方法有以下几个优点:

参考文献

- 1 Newman W M, Sproull R F. *Principles of Interactive Computer Graphics*. New York: McGrawHill, 1979
- 2 Sproull R F, Sutherland I E. A clipping divider[M]. In: Washington; Thompson Books, 1968. 765~775
- 3 Liang Y D, Barsky B A. A new concept and method for line clipping[J]. *ACM Trans on Graphics*, 1984, 3(1): 12~22
- 4 Cyrus M, Beck J. Generalized two-dimensional clipping[J]. *Computer and Graphics*, 1978, 3(1): 23~28
- 5 Nicholl T M, Lee D T, et al. An Efficient New Algorithm for 2D Line Clipping. In: SIGGRAPH' 87, *Computer Graphics*, 1987, 21: 4
- 6 刘勇奎, 颜叶, 等. 一个有效的多边形窗口的线裁剪算法[J]. *计算机学报*, 1999, 22(11): 1209~1214
- 7 刘勇奎, 刘桂英. 一般多边形窗口的线裁剪[J]. *计算机辅助设计与图形学学报*, 1993, 5(4): 269~274
- 8 赵平, 冯春, 等. 一种一般多边形窗口的有效线裁剪[J]. *西南交通大学机械工程学院*, 2002, 29(4): 34~48