

复杂多边形的填充算法*

范慧琳 张全伙

(华侨大学计算机科学系, 泉州 362011)

摘要 概述区域填充中两类常用算法——扫描转换和种子填充的基本思想及其性能. 结合两类算法的设计思想, 给出一个复杂多边形的填充算法, 并用 TURBO PASCAL 加以实现.

关键词 像素, 扫描, 多边形, 填充, 算法

分类号 TP 391.72; TS 941

光栅扫描设备得到普遍应用的原因之一是它不仅能输出线条和字符, 还具有表示实区域的能力. 在真实图形生成、图象处理、计算机艺术, 高精度字体的显示等多种领域中有着重要的应用. 我们把根据边或顶点的简单描述生成实区域的过程称为实区域扫描转换或多边形填充或轮廓线填充. 常用的填充算法可分为两大类: 扫描转换和种子填充. 前者着重解决由多边形轮廓线所围的区域的填色问题. 后者适用于任意形状轮廓线所围成的区域, 但缺点是要求区域必须闭合且一次只能填充一个单连通区域, 对复杂多边形(指自相交多边形或经裁剪窗口裁剪后将形成一个或多个分离多边形的多边形)而言, 往往未能给予正确填充. 每一多边形仅其中一个区域被填充. 为此本文概述了扫描转换和种子填充算法设计思想及其性能, 结合二者的基本思想, 给出一个复杂多边形的填充算法并作简要讨论.

1 多边形的扫描转换

一个多边形的扫描转换包括寻找所有在多边形边界内的像素以及对这些像素按一定的亮度显示出来. 判定给定点是否位于多边形内的方法有多种. 如, 从该点作一条射线, 计算它与多边形边线的交点数. 当交点个数为奇数时该点位于边界之内, 否则该点在边界外. 显然如此逐点测试的效率是极低的, 而使用点相关性可以从本质上改善扫描转换算法的性能. 点相关性认为, 如果一给定的像素在多边形内, 则与之紧密相邻的像素一般也在多边形内, 除非多边形边界线正好通过相邻的两个像素. 因此, 对每一条扫描线, 若求出它与边界的交点, 并使这些交点按 x 值的大小排序, 那么, 扫描线就被分为若干个区间, 每个区间要么整个在多边形内, 要么整个在多边形外, 而无需逐点一一判别.

假设多边形已由适当的裁剪算法针对裁剪窗口进行了正确的裁剪, 且顶点坐标被转换成设备坐标. 那么, 按以下基本思想设计算法便可实现对多边形的扫描线填充: (1) 对多边形的

* 本文 1995-07-06 收到

每一条边,计算与扫描线的所有交点,建立所有 (x, y) 交点表;(2)将所有交点进行排序,使得同一扫描线的交点放在一起,并按 x 值递增排列;(3)同一扫描线的交点总是成对出现的,对每一对相邻的交点 $A(x_a, y_a)$,必有 $y_a = y_b, x_a \leq x_b$,对线段 AB 进行填充。

这里,应注意两个问题:首先,以上方法取决于扫描线与多边形边界之间的交点坐标,当扫描线与多边形顶点相交时,顶点需按交点计数规则计数,以保证交点总是成对出现^[1]。其次,对交点作不同的排序分类,可形成多种效率不同的具体算法。本文实现的复杂多边形填充算法是以其中效率较高的 $y-x$ 算法^[2]为基础的。在实现过程中把握了以下两个要点:(1)把所有交点先按 y 的值作桶分类(把 y 相同的放进一个桶里),再在每个桶中对 x 作插入分类,形成 y 桶表(即多边形边线表);(2)只有在每次一条扫描线地更新显示时,才把那些需要的交点产生出来,因此在从上到下每次用一条扫描线进行填充时,对当前扫描线建立一张以 x 分类的现行边线表,其中列出了与该扫描线相交的所有边线。以上作法使表处于按序排好的状态所需的工作量大大减少,从而提高了算法的效率。

2 种子填充

种子填充算法假定在多边形或任意线型构成的区域内部至少有一个象素是已知的,然后按一定规律从一个象素移到另一个象素,并改变区域内所有象素值,从而完成区域的填充。其缺点是要求区域必须闭合且一次只能填充一个单连通区域,对自相交多边形或经裁剪窗口裁剪后将开成一个或多个分离多边形的复杂多边形往往无法作正确的填充。如图1所示,每一多边形仅其中一个区域被填充。区域可采用内部定义或边界定义^[3]、种子填充适用于边界定义区域,可用于任意线型构成的一般区域,由于一般区域可用多边形区域来逼近,在此仅讨论具有一般意义的边界定义的多边形区域填充。

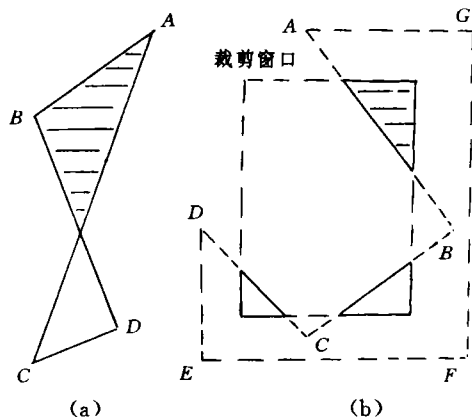


图1 不正确的复杂多边形填充

区域可分为四连通或八连通两种。如果区域是四连通的,那么区域内每一象素可通过上下左右四个方向的若干次移动到达。对于八连通区域,区域内的每一象素可通过两个水平方向、两个垂直方向和四个对角线方向的若干次移动到达。八连通算法可填充四连通区域,反之不然。采用堆栈的四连通区域种子填充算法的基本思想如下:

(1)种子象素压入堆栈;

(2)当堆栈非空时弹出一个象素,将该象素置成所要求的值,检查每个与当前象素邻接的四连通象素,若为边界象素或已置成所要求的值,则略而不计,否则把该象素压入堆栈;

(3)重复(2),直至栈空。

其中四连通边界定义区域可凸可凹,也可以包含一个或多个孔。该算法简单易实现,但填充过程中堆栈可能变得很大,且经常导致象素多次进栈,出现一些重复的、不必要的信息。使堆栈大小极小化的一种思想是在任意不间断扫描线区域中只取一个种子象素,该思想形成了

扫描线种子填充算法,改善了种子填充算法的效率。

3 复杂多边形的填充算法

一个复杂多边形区域可看成是由一个或多个四连通区域构成的。因而使用种子填充法正确填充复杂多边形的首要任务是在多边形的每个四连通区域中找寻一个种子点,完成了每一四连通区域的种子填充后也就完成了复杂多边形的填充^[4]。而种子的寻找则采用 $y-x$ 算法的基本思想。这里,设多边形已由裁剪窗口作正确裁剪且顶点的坐标被转换成设备坐标。

类同 $y-x$ 算法,建立多边形的边线表及与当前扫描线相关的现行边线表。对加入边线表中的每条边,检测它的每一顶点,以确定顶点是否具有局部最大值,即该顶点的 y 坐标是否大于或等于交于该顶点的两条边上另两个顶点的坐标。在边线表的边线记录中增加一个标志项,如果二条相邻边所交的顶点具有局部最大值,则在标志项中置入标记。然后,从最大 y 坐标开始,使用 $y-x$ 算法的思想自上而下每次用一条扫描线进行扫描。所不同的是,这里并不对扫描线上的像素值作填充,而只是搜索种子点的设置位置。对当前扫描成,如果:(1)加标记的边被加入现行边线表中,或(2)两条边相交于一非顶点的点(如图 2(a)中 BD 与 AC 的交点),那么它们在现行边线表中的位置将互换。一旦现行边线表排序后更改了边的原顺序,便设置一个标志。调用这一排序功能的操作若检测到该标志,则认为情形(2)出现。图 2(b)给出了一个自相交多边形,上部种子根据情形(1)设置,下部种子根据情形(2)设置。

以上的情形(1)可在构造边线表时,通过对顶点的检测加以判断。如图 2(a)所示,考虑多边形的局部最大值,设置了两个种子,确保裁剪后形成的两个分离四连通区域均被填充。情形(2)是依据现行边线表的排序情况而定的。如果二条边相交于一非顶点的点(如图 2(a)中 BD 与 AC 的交点),那么它们在现行边线表中的位置将互换。一旦现行边线表排序后更改了边的原顺序,便设置一个标志。调用这一排序功能的操作若检测到该标志,则认为情形(2)出现。图 2(b)给出了一个自相交多边形,上部种子根据情形(1)设置,下部种子根据情形(2)设置。

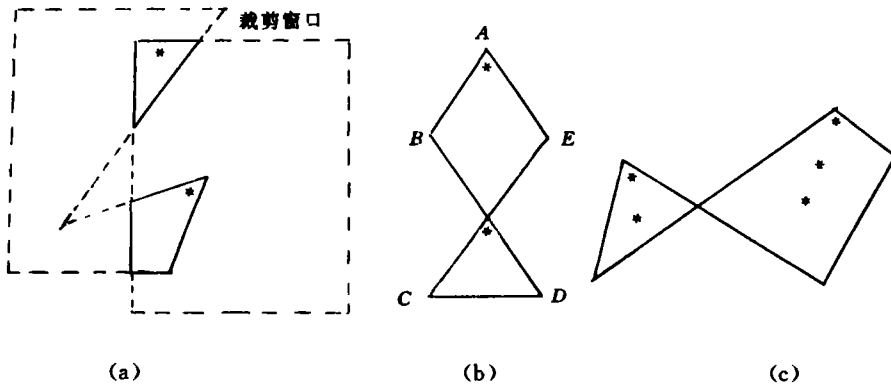


图 2 种子点的设置

算法的基本步骤描述如下。

- (1)对复杂多边形作裁剪,并将顶点坐标转换为设备坐标。
- (2)构造边线表。对包含局部最大值的边作标记。

(3)设 Seed-point 为种子点设置判别项,初值为假。从最大 y 坐标开始扫描,对边线表中的所有边线循环执行(i)~(iv):

- (i)若 Seed-point 为真,在现行边成表的每一对边线中间设置种子,将种子点压入堆栈,而后置 Seed-point 为假;
- (ii)建立当前扫描线的现行边线表,并对其排序;
- (iii)若现行边线表含有带标记边或现行边线表的排序操作改变了边顺序,则置 Seed-point 为真.
- (iv)扫描线前进一行.
- (4)用种子填充算法完成复杂多边形每一四连通区域的填充.

4 结 论

如文中所述,扫描线算法能完成对复杂多边形的正确填充,但速度较慢.种子填充算法相对而言速度较快,但却无法对复杂多边形作正确填充.为此,本文给出的算法仅用 $y-x$ 算法的基本思路确定复杂多边形的每一四连通区域的种子点,而每一四连通区域像素点的填充改用速度较快的种子填充方法实现.该算法使用 TURBO PASCAL 在 IBM 机上实现.实验表明,对小多边形,其效率与扫描线方法类同,而对尺寸较大的复杂多边形,其处理速度比扫描线方法要快.对凹的或自相交多边形的某些四连通区域,本算法可能生成多于一个的种子点,如图 2(c)所示.这些种子点均压入堆栈.设某一四连通区域已经用堆栈中弹出的某一种子点由种子填充法填充完毕,若从堆栈中弹出的当前种子点又落在该区域内,那么,从上述的种子填充算法步骤(2)可知,由于与该种子点邻接的四连通象素均已填充,则该区域不会再有象素点进栈,并不会产生无谓的重复填充.可见,多余的种子对处理速度的影响极其微小,因而本文的算法是可取的.

参 考 文 献

- 1 纽曼 W M, 斯普劳尔 R F. 对话式计算机图形显示原理. 易晓东等译. 北京:科学出版社,1984. 229~245
- 2 金廷赞. 计算机图形学. 杭州:浙江大学出版社,1988. 238~247
- 3 罗杰斯 D F. 计算机图形学的算法基础. 梁友栋等译. 北京:科学出版社,1987. 73~101
- 4 Richards J E. Filling complex polygons by region-fill methods on raster graphics terminals. Computer Graphics Forum, 1987, (6):49~54

An Algorithm for Filling Complex Polygon

Fan Huilin zhang Quanhao

(Dept. of Computer Science, Huaqiao Univ., 362011, Quanzhou)

Abstract In regional fill, scan conversion and seed fill as two kinds of algorithms in common use are discussed in reference to their key ideas and performance. Combining with their design thinking, an algorithm for filling complex polygon is given and implemented with Turbo Pascal.

Keywords image element, scan, polygon, fill algorithm