

快速绘制剖面线的区域边界搜索算法*

秦圣峰

(基础课部)

摘 要

现有的剖面线算法大都需要输入边界参数,使用起来很不方便。本文根据光栅图形显示器的显示特性,提出一种区域边界的自动识别搜索算法。该方法将几何模型分为向上做区和向下做区,并把区域边界简单地分为两种状态来处理,对若干曲线相交而构成的封闭区域,只要给出区域内的一个种子点和剖线方向及间隔三个参数,即可绘制区域内的剖面线。该算法已由 TURBOC 语言编程实现,实践表明,它快速、稳定且易于实现。

关键词:剖面线;区域边界搜索;算法

0 引 言

在 CAD、CG 中,经常要遇到剖面线的绘制问题。以往的做法是建立区域边界的数学模型,输入大量的边界参数和剖面线参数,并通过求交,判断和排序等一系列步骤来绘制剖面线。由于零件的剖面形状大多是由若干曲线相交而构成的复杂区域,故这种方法在使用上多有不便。文[1]给出了一种不输入边界参数的通用剖面线绘制方法,它将区域分为向上做区和向下做区,把边界细分为十四种状态来处理,对区域内的所有点进行识别、判断和处理。该算法的缺点是效率太低且程序实现较复杂。寻求不输入边界参数且快速简捷的剖面线绘制算法,是本文提出和所要解决的问题。

本文所讨论的算法——区域边界自动搜索算法,其基本思想是:利用语言中读某一指定点色彩值的函数功能,根据种子点首先在剖线方向上搜索第0条剖线的始终点,然后自此剖线的终点开始,用7个可能方向顺序搜索边界点,当新搜索到的边界点满足间隔要求时,再据此点在区域内部搜索剖线,直到所有边界都搜索完毕。

本文于1992年4月13日收到

* 在全国第七届 CAD 和图形学学术会议上宣读。

1 区域边界的几何模型建立

1.1 几个概念的定义

(1) 读点: 设显示屏上当前点的位置为 (x, y) , 读 (X, Y) 点的色彩值; 在 TurboC 中用 $\text{getpixel}(x, y) = \text{color}$ 表示这一过程。

(2) 前读: 因剖面线常用 $\pm 45^\circ$ 两种方向, 以下均以 45° 方向讨论。从当前点 (x, y) 读新点 $(x+1, y+1) = \text{color1}$ 和 $(x+1, y) = \text{color2}$ 的过程。

(3) 回读: 从当前点 (x, y) 读 $(x-1, y-1) = \text{color3}$ 和 $(x-1, y) = \text{color4}$ 的过程。

(4) 剖线的始终点: 规定左边的端点为始点, 右边的端点为终点。

1.2 区域模型与边界分类

(1) 向上做区与向下做区的划分。根据已知区域内的种子点 (x_0, y_0) , 利用读点过程即可建立目标区域的底色 DS。从 (x_0, y_0) 出发, 循环执行前读过程, 直到 $(\text{color1 OR color2}) \neq \text{DS}$ 为止, 当前点即为 0 号剖面线的终点 (边界上的点), 记为 Q_0 ; 再从 (x_0, y_0) 出发, 利用回读过程, 同样可找到 0 号剖面线的始点 P_0 。

在求得 0 号剖面线之后, 我们定义: 位于 0 号剖面线上部的区域为向上做区, 其下部区域为向下做区。见图 1 所示。

前读过程只在求 0 号剖面线时使用一次。

(2) 起点边与终点边的划分。区域的边界曲线, 在光栅显示器上显示时, 无论是由 DDA 算法还是由 Bresenham 算法生成, 其结果都将是由 8 个方向的单位矢量组合而成的线串 [2], 如图 2 所示。

在向上做区, 从 Q_0 点逆时针搜索边界, 只有 1, 2, 3 三个方向上的矢量边才可能存在剖面线终点。此时 1, 2, 3 三个方向上的边我们称为终点边。其余方向上的边都称为起点边。

在向下做区, 从 Q_0 点顺时针搜索边界时, 5, 6, 7 三个方向上的边为终点边, 其余为起点边。见图 1 和图 2。

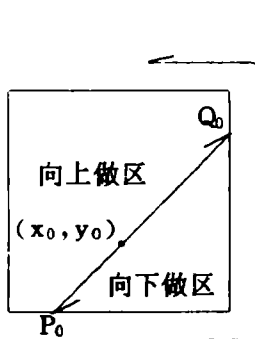


图 1

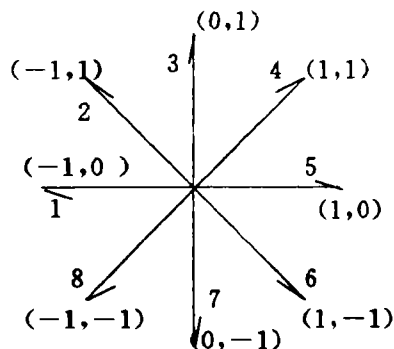


图 2

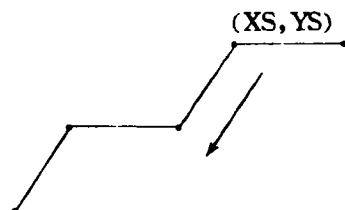


图 3

2 边界搜索算法

2.1 算法的过程描述

为简便起见, 现假定绘制区域为单连通域。算法过程如下:

(1) 从 (x_0, y_0) 出发, 找 P_0, Q_0 两点, 并存贮 $P_0 \rightarrow P[0], Q_0 \rightarrow Q[0]$ 。

(2) 设 Q_0 为当前点 (x_s, y_s) , $(fx=1, fy=1)$ 为初始方向, 开始向上做区的边界搜索。对7个可能边界搜索方向进行顺序搜索, 当某方向上存在边界点时, 把新点赋给 (x_s, y_s) , 当前搜索方向赋给 (fx, fy) ; 若 (fx, fy) 位于终点边上且 (x_s, y_s) 距第 K 条剖面线的间隔大于等于给定值时, 则以 (x_s, y_s) 为据, 回读找新一条剖线的始点, 并存贮新一条剖线的始终点于 $P[K], Q[K]$ 中, 且让变量 $K=K+1$ 。否则, 继续边界搜索过程, 直到 (x_s, y_s) 与 P_0 重合, 此时表明向上做区处理完毕。

(3) 设 Q_0 为当前点 (x_s, y_s) , $(fx=1, fy=1)$ 为初始方向, 开始向下做区的边界搜索, 边界搜索过程同向上做区。

(4) 显示全部求出的剖面线。

2.2 边界搜索方向和顺序的确定

对向上做区, 为了搜索区域的最短边界, 搜索方向可以如此决定: 从当前已知方向 (f_x, f_y) , 逆时针旋转 135° 角 (即向前推进三个方位), 得到第一个搜索方向。从此位置开始每次顺时针旋转 45° 角, 旋转6次共可得7个可能的顺序搜索方向。假定当前已知方向为 $(+1, +1)$, 则得到的7个可能方向如图2所示。这样一来, 宏观上的边界搜索方向是逆时针。

对向下做区, 搜索方向和顺序为: 从当前已知方向 (f_x, f_y) , 顺时针旋转 135° 角, 得到第一个搜索方向。从此位置开始, 每次逆时针旋转 45° 角, 旋转6次共可能7个可得方向。结果对向下做区的边界总体上来看为顺时针搜索。

2.3 几点说明

(1) 终点内缩处理。由于图形线段的光栅化, 致使部分线段 (主要是 $1/8$ 圆域内的线段) 会出现如图3所示的光栅化结果。从图中可明显看出, 如果用 (x_s, y_s) 直接回读找始点, 就会与当前线段相自交, 以致于丢失剖线。为此需将 (x_s, y_s) 向区域内部内缩到 (x_1, y_1) , 以此作为剖线终点, 并回读求始点, 在屏幕坐标系内, 终点内缩处理为:

向上做区: 取 $dx=-1, dy=1$:

If $(fx=0 \text{ AND } fy=-1) \{x_1=x_s-1; y_1=y_s\}$

else $\{x_1=x_s-dx; y_1=y_s+dy\}$

向下做区: dx, dy 同上。

If $(fx=0 \text{ AND } fy=1) \{x_1=x_s-1; y_1=y_s\}$

else $\{x_1=x_s+dx; y_1=y_s+dy\}$

(2) 复连通域的处理。当区域内部有孔洞存在时, 0号剖线的始终点有可能不在同一条边界上, 此时的边界搜索将遍及整个外边界或某个内边界且只有向上做区存在。这通过适当修改判断条件 (x_s, y_s) 与 P_0 或 Q_0 重合即结束向上做区来实现。

对复连通区域的处理, 其一可对第一次绘制后留下的空白区作再次区域搜索; 其二是在

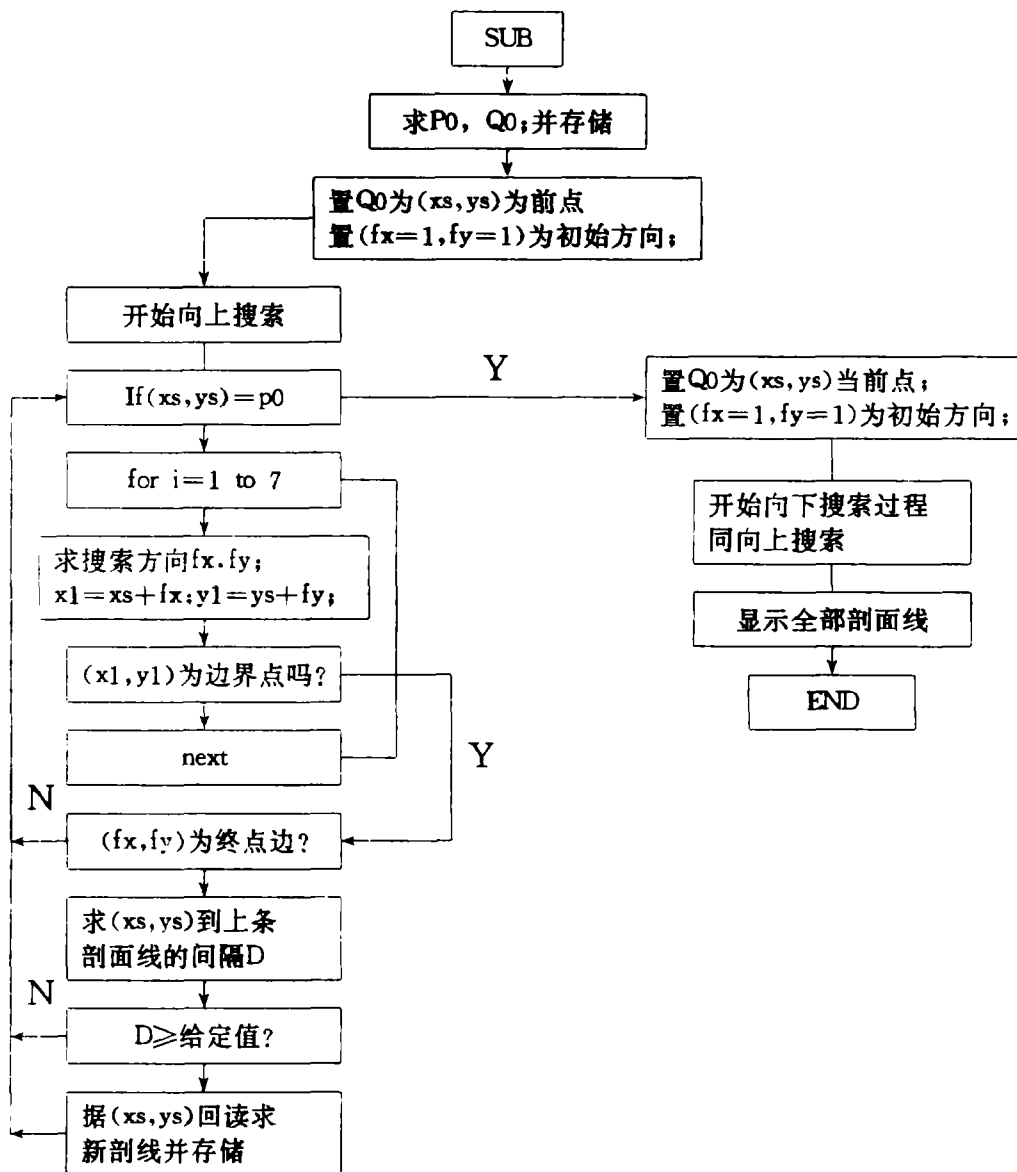


图4 流程图

剖面绘制之前,用一条平行于剖面线方向的辅助直线将区域的内外边界连通,使复连通域转换为单连通域,即可一次进行处理。这里对辅助直线不要求其端点位于内外边界上,只要与内外边界相交即可。

在工程制图中,绝大多数的剖面线绘制区域都是单连通域。

3 程序框图及实例

根据以上介绍的算法,作者用 TurboC 语言编制了通用的不需输入边界参数的剖面线程序,源程序约70个程序行。图4是简要的程序流程图。图5是 -45° 方向单连通域的绘制实例。图6是 45° 方向复连通域的绘制实例,它是用辅助直线来处理的。

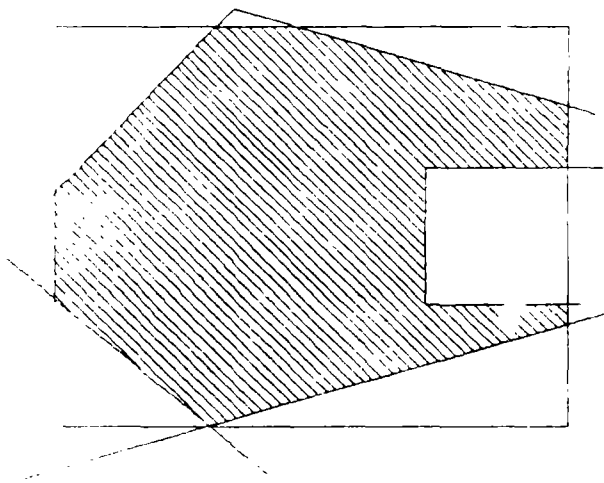


图5

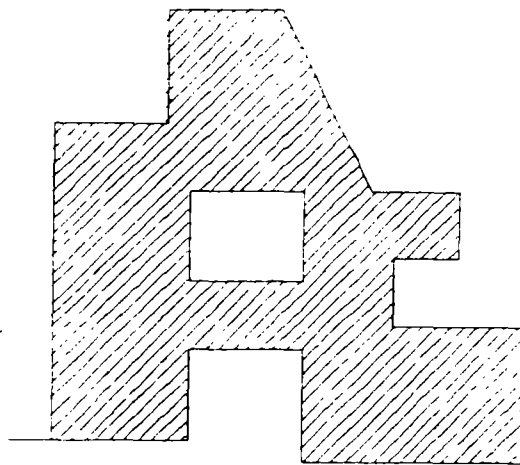


图6

4 结 论

1、区域边界搜索算法是一种新的编程方法,它不仅适于绘制剖线,且可用于几何图案的区域填充。

2、该算法能够自动寻找包围种子点的最短边界,因而可应用于图形识别。

3、计算机实际操作运行表明,本文提出的算法是快速、简捷的。

4、由于边界搜索过程中,对起点边只是走过而没作特殊处理,故在终点边与起点边交替出现的区段,可能出现间隔不严格一致的现象,但相差不多。实践表明,绘制的剖线是能够满足使用要求的。

5、该算法与文〔1〕的算法相比,效率可提高 d 倍,这里 d 为剖面线间隔。

参 考 文 献

- [1] 李奋强. 不输入边界参数的剖面线程序设计. 工程图学学报, 1992; 2: 1~5
- [2] D·F 罗杰斯著, 梁友栋译. 计算机图形学的算法基础. 北京: 科学出版社, 1987: 30~112

The Region Boundary Searching Algorithm For Fast Drawing Section Lines

Qin Shengfeng

ABSTRACT

In this paper, the new algorithm for fast drawing section lines is presented, basing on the pixel displaying and disting function of computer upon the monitor. Using this programme of section lines designed by this algorithm, it is sufficient to input only the corrdinates of one point with in region along with the direction and interval parameter of the section lines to draw the section lines within whole close region. The algorithm is proved to be simple and practical. And had accomplished on micro-computer with Turboc.

Key words: Section Lines; Region boundary searching; Algorithm