从空心汉字到实心汉字－任意区域填充算法

杨学平 (中国科学院软件所 100080)

1.问题的提出

空心矢量式汉字与点阵式汉字相比有许多优点：节省存储空间，伸缩时不失真。但需要一种强大的区域填充算法，以使空心汉字变为实心汉字输出。一般的区域填充算法较难适应需要。如图 1 所示的由自相交、凹凸性不定的多边形区域形成的空心汉字，用通常的算法都会遇到困难。为此，本文提出一种改进算法，来适应由任意多边形区域形成的空心汉字的填充需要。

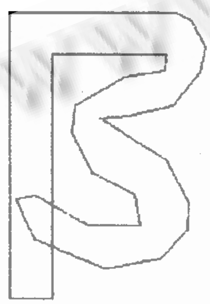

图 1 自相交的多边形区域

图 2 自相交的多边形区域

2.算法基本思想

我们将在设备坐标系下叙述这种算法，对其他坐标系显然不难推广。设有一个多边形区域 R，不论其凹凸及连通情况如何，我们仅要求每个多边形顶点都按相同顺序存放：或顺时针或逆时针，有空洞部分时，外环按一种顺序存放，内环则按另一种顺序存放。

一般填充算法的基本思想是：用一条扫描线与所有的棱边求交点，然后将此扫描线上的交点排序，则认为每对点之间的区域就是 R 的内部区域，这样图 1 填充的结果就是图 2，显然发生了错误。

新算法的基本思想是：对一个任意的一笔画完的封闭多边形（无论它是否自相交，也无论其凹凸情况如何），除水平边外，棱边可分为两类：Y 坐标是增加的和 Y 坐标是减少的。在一条扫描线与 R 的所有棱边求得交点后，将交点按属于 Y 增和 Y 减分为两类分别进行排序，在此扫描线上交点进行配对时，其方式是，Y 是增加的棱边上的交点必须逐一和 Y 是减少的棱边上的交点配对，而不是象传统那样绝对地按一条扫描线上所有交点排序的顺序配对。

这个算法的另一个关键是：当把一条非水平线棱边离散化为设备坐标系下的点（即与各扫描线求交点）时，必须保证在一条扫描线上最多只能留下一个点。

以下的算法是对区域 R 来实施的。

3.算法

按棱边顺序将 R 的每一条棱边与扫描线求交，这些交点将按下面的次序存入指定的数组中。为此我们设置 2n 个一维数组：Ao, A1, …, An, Bo, B1, …, Bn，每个数组的大小与设备 Y 方向（或多边形区域 R）的点数一致，任何一个数组的第 j 单元存放的都是扫描线 y＝j 与各棱边交点的 x 坐标，n 的情况可视区域 R 的复杂性而定。初始时置这些数组所有元素为－1。我们将用{Ai}存放 Y 增棱边与扫描线的交点，用{Bi}存放 Y 减棱边与扫描线的交点，并要求它们分别是排好序的。为此，我们定义一个操作"将一点送数组 Ai"：设一点为（X, Y），这操作意味着：i 从 1 开始。

(1)检查是否 Ai(Y)＝－1

(2)若 Ai(Y)≠－1，则检查是否 Ai(Y)＞X，若是则交换这两个数，然后 i＋1⇒i，转步骤 1。

(3)若 Ai(Y)＝－1，则令 Ai(Y)＝X。

同样有"将一点送数组 Bi"的操作。

实际上这操作就是一扫描线与 R 所有棱边交点的排序动作。

现在我们将区域 R 的各棱边作以下处理，设一棱边的两端点依顺序为(X_p, Y_p)，(X_{p+1}, Y_{p+1})，令

$$IFLAG_p = SIGN(Y_{p+1} - Y_p) = \begin{cases} -1 & Y_{p+1} < Y_p \\ 0 & Y_{p+1} = Y_p \\ 1 & Y_{p+1} > Y_p \end{cases} \quad (1)$$

被称为当前棱边的 Y 增减属性。这样，IFLAGp + 1 就是下一条棱边的 Y 增减属性，IFLAGp - 1 是上一条棱边的 Y 增减属性等等。在以下算法中，我们用数组集合{Ai}记录 Y 增属性的棱边与扫描线的交点，用数组集合{Bi}记录 Y 减属性的棱边与扫描线的交点：

(1)当 $Y_{p+1} > Y_p$时，则将此棱边与各扫描线的所有交点，除(X_p, Y_p)外，送数组 A_i；

(2)当 $Y_{p+1} < Y_p$时，则将此棱边与各扫描线的所有交点，除(X_p, Y_p)外，送数组 B_i；

(3)当 $Y_{p+1} = Y_p$时，若 IFLAGp + 1 > 0 时，将(X_{p+1}, Y_{p+1})再送数组 A_i；

若 $IFLAG_{p+1} < 0$ 时，将(X_{p+1}, Y_{p+1})再送数组 B_i；

(4)若 $IFLAG_p \cdot IFLAG_{p-1} < 0$ 时，将(X_p, Y_p)再送数组 A_i(当 $IFLAG_p > 0$ 时)或数组 B_i(当 $IFLAG_p < 0$ 时)。

这是尖点退化处理。

(5)若 $IFLAG_{p-1} = 0$ 并且 $IFLAG_{p-2} \cdot IFLAG_p > 0$，将$(X_p, Y_p)$送数组 A_i(当 $IFLAG_p < 0$ 时)或数组 B_i(当 $IFLAG_p > 0$ 时)。

这是水平线退化处理。

当区域 R 的所有棱边依顺序按上述算法处理完毕，我们把非 -1 的数组元素配对，即将扫描线 Y = J 上 $A_i(j)$ 与 $B_i(j)$ 按 J 从小到大配对，将所有$(A_i(j), J)$与$(B_i(j), J)$连线，那么填充过程全部完成。

值得注意的是：在上述过程中，我们只保证填充了区域的内部，而不能保证其所有棱边也被画出。如图 3 的水平棱边 AB 的某些水平点就被丢弃了，这需要通过追画 R 的所有棱边加以补足。

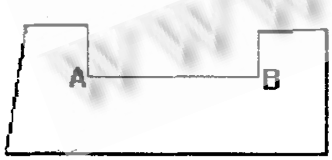

图 3 算法中漏掉的水平棱边

最后我们来实现上述的"当把一条非水平线棱边离散化为设备坐标系下的点时，必须保证在一条扫描线上最多只能留下一个点"。我们使用快速递推算法，这个算法可见有关文献。

设一条棱边两端点为(X_p, Y_p)，(X_q, Y_q)，且 $Y_p \neq Y_q$，那么此棱边可表为：

$$X = X_p + \frac{X_q - X_p}{Y_q - Y_p}(Y - Y_p) = X_p + K(Y - Y_p) \quad (2)$$

其中： $K = \frac{X_q - X_p}{Y_q - Y_p}$ (3)

故对扫描线 $Y = Y_m$与棱边 (2) 的交点 X_i 为

$X_m = X_p + K(Y_m - Y_p)$ (4)

对扫描线 $Y = Y_{m+1} = Y_m + 1$ 与棱边 (2) 的交点 X_{m+1}为

$X_{m+1} = X_p + K(Y_m + 1 - Y_p)$ (5)

将 (5) 减去 (4) 得：

$X_{m+1} = X_m + K$ (6)

当 $Y_q > Y_p$ 时，我们将用 (6) 从(X_p, Y_p)开始计算到(X_q, Y_q)；反之当 $Y_q < Y_p$ 时，我们将用

$X_{m+1} = X_m - K$ (7)

从(X_q, Y_q)计算到(X_p, Y_p)。

图 5 是图 4 用本算法填充的结果。

很显然，上述算法对于用曲线(例如 Bezier 曲线)描述的区域作填充也是适用的，只要保证"把一条曲线棱边离散化为设备坐标系下的点时，在一条扫描线上最多只能留下一个点"。

图 4 一个空心汉字

图 5 空心汉字填充的结果

(来稿时间：1997 年 5 月)