

基于顶点与邻边相关性的多边形填充算法

马辉 陆国栋 谭建荣

(浙江大学 CAD&CG 国家重点实验室, 杭州 310027)

吴良

(Brown University, Providence, Rhode Island, U. S. A)

摘要 为了加快多边形填充算法的运算速度,在深入挖掘顶点与相邻边关系对填充算法影响的基础上,提出了一种基于顶点与邻边相关性的多边形填充算法。该算法首先归纳了多边形顶点与邻边相关性的5种典型类型,然后依据顶点与邻边的相关性,对原有多边形进行了分割与重新组合,使其完全由简单的三角形和梯形这样的单元区域组成,这样就将复杂的多边形填充问题转化为这些单元区域的填充问题,并由此将扫描线与多边形边求交的乘除计算转化为加减运算。通过实验分析,新算法大大减少了运算的时间和复杂度,从而为多边形填充创造了一种有效的新途径。

关键词 多边形 填充 顶点与邻边相关性

中图法分类号: TP391.41 **文献标识码:** A **文章编号:** 1006-8961(2004)11-1336-06

A New Polygon Filling Algorithm Based on Pertinence Between Point and Its Abutting Sides

MA Hui, LU Guo-dong, TAN Jian-rong

(State Key Laboratory of CAD&CG at Zhejiang University, Hangzhou 310027)

WU Liang

(Brown University, Providence, Rhode Island, U. S. A)

Abstract Present polygon filling algorithms including scanline algorithm, seed filling algorithm and the algorithms based on the two classical filling theories, are analyzed and compared in this paper. A new filling algorithm is put forward, which is based on thorough analysis of the relations between point and its abutting sides. This new filling algorithm divides all points of a polygon into five types at firstly, and transforms the polygon into unit areas which are simple triangles and trapeziums by the line passing the points. Using the characteristics of bevel edges, the unit areas filling can replace the multiplication-division with the addition-subtraction. It decreases the time and complexity of filling the whole polygon. This paper explains the design of the algorithm and how it is going on, and also presents the data structures storing the information of points and unit areas. In the end, some experimental results show that the new algorithm has a high efficiency and a good stability.

Keywords polygon, fill algorithm, pertinence between point and its abutting sides

1 引言

众所周知,多边形的填充问题是计算机图形学的基本问题之一。近年来,随着多媒体技术的不断发展,对这一问题的研究更是引起了人们的关注。传统的多边形填充经典算法有扫描线填充算法和种子填充算法两种。前者是从多边形的顶端开始,到多边形

的底端为止,即先用一根根水平线进行扫描,并求得每一条水平扫描线与多边形各边的交点,然后将交点配成“交点对”,并在其间进行填充;后者是先给定区域内的一点为种子点,并对其赋予指定的颜色,然后将这种颜色扩充到区域内所有的点。

此后,在经典算法的基础上有了很多改进,如,对于扫描线填充算法,文献[1]归纳了奇偶扫描转换算法和有序边表扫描算法以及这两种算法的改进;

羊四清结合非零环绕规则与奇偶规则改进了活化边表结点结构^[2];Cordon 的关于斜扫描线的 CP 法也被引入到多边形填充中来^[3]。对于种子填充算法,Smith 曾在 1979 年提出了扫描线种子填充算法^[4],文献^[5,6]也从不同的角度对种子算法进行了有效的改进。

由于以上对扫描线填充算法的改进都沿用了扫描线逐条与多边形边求交的思想,虽然在一定程度上简化了算法流程与优化了数据结构,但是依然没有能够避免每根扫描线都与多边形的各个边进行比较、求交,还是存在较多的乘除运算和多余的判断,因此本文从分析多边形的顶点与邻边的相关特性入手,归纳了顶点与邻边相关性的 5 种典型类型,并提出了通过转化多边形为单元区域的新思路;然后,设计一种过且只过不同类型的顶点做扫描线的算法,并通过这些扫描线将多边形分割成由三角形和梯形单元区域构成的组合,使分割后的单元区域可以通过加减运算代替求交运算来得到其他所有扫描线与多边形边的交点,这样不仅显著提高了效率,同时也扩展了扫描线填充算法的运算途径。

2 多边形顶点与邻边相关性分析

2.1 顶点与邻边相关性类型

在多边形的填充过程中,顶点所具有的不同特性,对于填充算法的思路和实现都有很重要的影响。顶点与相邻边的相关性是指与顶点相邻的两条边与过顶点的水平直线的相对位置而言,对于所有可能出现的位置关系的 5 种类型定义如下(图 1 所示):

- 类型 1:与顶点相邻的两条多边形的边,分别位于过顶点水平直线的上方和下方;
- 类型 2:与顶点相邻的两条多边形的边,都位于过顶点水平直线的下方;
- 类型 3:与顶点相邻的两条多边形的边,都位于过顶点水平直线的上方;
- 类型 4:与顶点相邻的两条多边形的边,一条位于过顶点水平直线的下方,另一条为过顶点的水平直线;
- 类型 5:与顶点相邻的两条多边形的边,一条位于过顶点水平直线的上方,另一条为过顶点的水平直线。

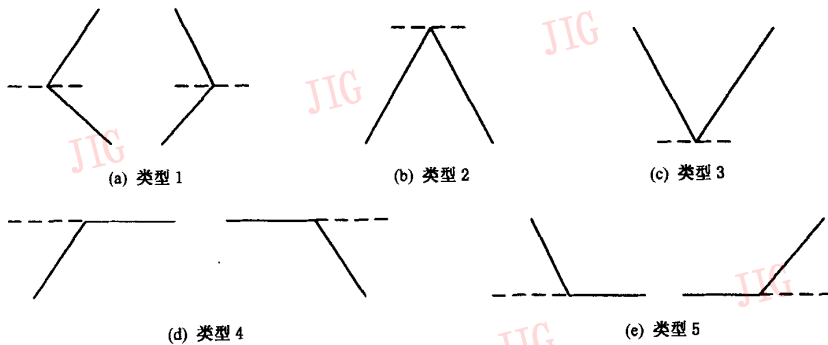


图 1 顶点与邻边相关性的各种类型

2.2 基于相关性的填充思路

通过分析可知,在经典的扫描线填充算法中,由于过 y 轴方向的每一像素点都要做扫描线,并要求扫描线与多边形每一条边的交点,因此有相当的计算量,而通过顶点的扫描线则可以把任意的多边形分割成由三角形和梯形组成的单元区域,从广义角度而言,由于此类底边为水平线段的三角形也可以看作是上边为零的梯形,因此分割后形成的单元区域在两条过顶点的水平扫描线之间的线段斜率不变(如图 2 所示)。

假设分割后形成的单元梯形的 4 个顶点分别为 $A = (x_A, y_A)$ 、 $B = (x_B, y_B)$ 、 $C = (x_C, y_C)$ 、 $D = (x_D,$

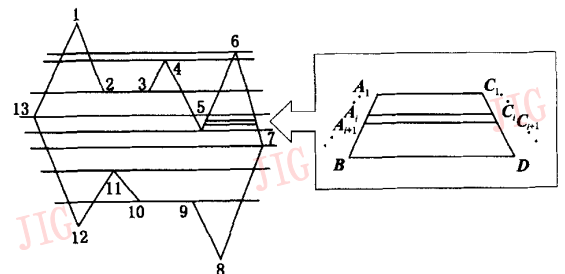


图 2 包含各类型相关性的多边形及分割后形成梯形区域示例

$y_D)$, AB 与 CD 为两条斜边,所需填充的像素点行间距为 d 。

定义 1 单元梯形的两条斜边的像素点分别为:①位于 AB 边的像素点 $\{A_1, A_2, \dots, A_n\}$, 记为 A_i , $i=1, 2, \dots, n$; 其中 $n=(y_B-y_A)/d$, $A_1=A$, $A_n=B$; ②位于 CD 边的像素点 $\{C_1, C_2, \dots, C_n\}$, 记为 C_i , $i=1, 2, \dots, n$, 其中 $n=(y_D-y_C)/d=(y_B-y_A)/d$, $C_1=C$, $C_n=D$ 。

定义 2 两条斜边的斜率记为 $k_{AB}=(y_B-y_A)/(x_B-x_A)$ 和 $k_{CD}=(y_D-y_C)/(x_D-x_C)$ 。

对于单元区域内的扫描线填色可以用起始点的位置 (x_{A_1}, y_{A_1}) 、 (x_{C_1}, y_{C_1}) 以及梯形斜边的斜率 k_{AB} 、 k_{CD} 来推导下一条扫描线与该单元区域的交点, 并按式(1)、式(2)依次推导:

$$\begin{cases} x_{A_{i+1}} = x_{A_i} - \frac{d}{k_{AB}} \\ y_{A_{i+1}} = y_{A_i} - d \end{cases} \quad (1)$$

$$\begin{cases} x_{C_{i+1}} = x_{C_i} - \frac{d}{k_{CD}} \\ y_{C_{i+1}} = y_{C_i} - d \end{cases} \quad (2)$$

在得到所有水平扫描线与单元区域的交点对后, 即可直接对此水平位置点对进行填色处理。

通过上述填充方法就避免了经典扫描线算法中, 每条扫描线与多边形的每一条边都要进行求交运算的问题。新算法只对过顶点的扫描线做与多边形边的求交运算, 并利用顶点与邻边的 5 种相关性来将原先复杂多边形转换为由多个三角形与梯形组成的单元区域, 这样对多边形的填充也随之转换为对这些单元区域的填充。由于单元区域在两条过顶点水平扫描线之间具有线段斜率不变的特性, 因此算法通过简单的加减运算即可推算出其他所有扫描线与单元区域的交点, 即扫描线与多边形各边的交

点, 这就大大减少了计算的复杂性。

3 填充算法设计与实现

3.1 算法设计

经过上述讨论后, 就可以针对多边形顶点与邻边不同的相关性来进行算法的设计, 以达到最少的计算量和最小的复杂性。

多边形可用有序点集 P 表示, $P=\{P_1, P_2, \dots, P_m\}$, 其中 m 为顶点个数, 并沿多边形边的顺时针方向记录顶点的 x 与 y 坐标信息。

顶点重排序后多边形可用有序点集 \hat{P} 表示, $\hat{P}=\{\hat{P}_1, \hat{P}_2, \dots, \hat{P}_m\}$, 其中 m 为顶点个数。点集需按照 y 坐标从大到小重新排列坐标信息, 以便于从上至下过顶点做扫描线。

通过分析可以发现, 对于单元区域分割来说, 顶点处于不同的位置将起到不同的作用, 并且和该顶点与邻边的相关性有密切联系。如图 3 所示, 若用过顶点 6 的扫描线与最高顶点 1 先对多边形进行分割, 则此时顶点 6 并不参与分割; 过顶点 4 的扫描线作出后, 过顶点 6 的扫描线将再次参与多边形的分割, 此时顶点 6 将参与分割。同样, 顶点 4 以及多边形中的其他顶点也都将遇到这样的情况。

为此, 建立双层次的数据结构数组 $Current$ 、 $Next$, 用以存放过顶点扫描线上所有有效点处于不同位置的信息, 并用以判断顶点与邻边的相关性:

(1) 类型 1, 如图 3 中的顶点 7、13, 这些顶点在 $Current$ 、 $Next$ 中各存放一次;

(2) 类型 2, 如图 3 中的顶点 1、4、6、11, 这些顶点在 $Current$ 中不存放, 在 $Next$ 中存放两次;

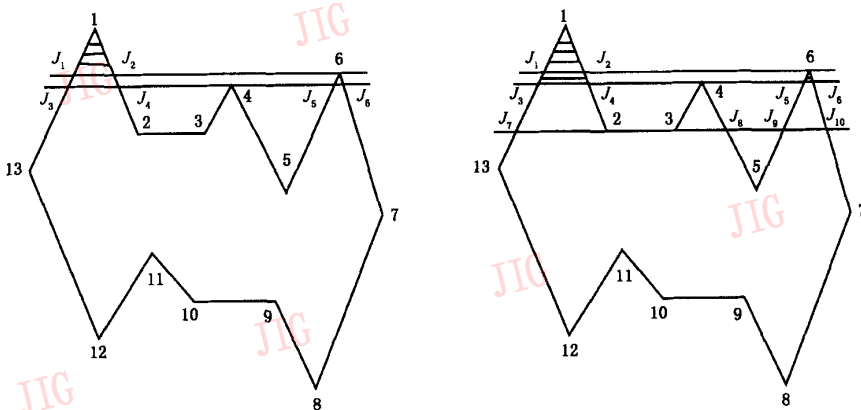


图 3 顶点位置不同在分割多边形中起不同作用的示意图

(3) 类型 3,如图 3 中的顶点 5、8、12,这些顶点在 *Current* 中存放两次,在 *Next* 中不存放;

(4) 类型 4,如图 3 中的顶点 2、3、10,这些顶点在 *Current* 中存放一次,在 *Next* 中不存放;

(5) 类型 5,如图 3 中的顶点 9,该顶点在 *Current* 中不存放,在 *Next* 中存放一次。

之所以个别类型顶点(4、6、8、12 等)处于某些位置须存放两次,是考虑将分割成的三角形也作为梯形处理,由于这样算法中由有效点构成的单元区域的点总是成对出现,因而便于处理。

为了便于填充计算,引入数组 *ForeNext*,用于存放上一条扫描线 *Next* 中有效点的信息,并采用先进先出的数据结构进行处理,即由每次读入新扫描线产生的有效点来形成以 *ForeNext*、*Current* 和 *Next* 组成的 3 层数组,并将 *ForeNext* 和 *Current* 中的有效点集合,以两点一组进行组合来构成单元区域,如图 4 即为当前扫描线过顶点 4 时,向下一条扫描线转换的过程。在形成单元区域后,在区域内就可以根据式(1)、式(2)推断出所有交点后,进行填充。

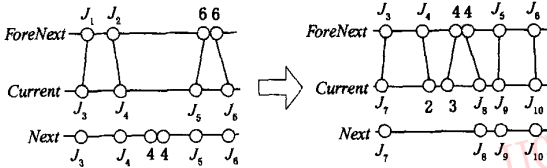


图 4 当前扫描线向下一条扫描线转换过程中的点的存储

3.2 算法描述

综合 3.1 节与 3.2 节的论述,即可以得到新算法的语言描述如下:

(1) 初始化

数组 x_p : 描述多边形边的有序点集的 x 坐标值;

数组 y_p : 描述多边形边的有序点集的 y 坐标值;

数组 $y_{\hat{p}}$: 根据 y 值大小对多边形顶点进行排序后各顶点 \hat{P} 的 y 坐标值;

数组 *Current*: 记录过当前顶点扫描线上所有有效点参与第 1 次区域组合的 x 坐标值;

数组 *Next*: 记录过当前顶点扫描线上所有有效点参与第 2 次区域组合的 x 坐标值;

数组 *ForeNext*: 记录过上一顶点扫描线上所有有效点参与第 2 次区域组合的 x 坐标值;

数组 *Temp*: 记录普通交点的 x 坐标值;

函数 *Index*() : 由 $x_{\hat{p}}$ 值对应到 x_p 值;

函数 *Sort*() : 排序函数;

函数 *Size*() : 返回数组列数;

函数 *Plot*() : 直线填色函数;

N : 多边形边数;

Step: 填充的像素点行间距,即扫描线算法中水平扫描线间距;

(2) 算法(伪码)

```

for n=2 to N //从次高顶点开始循环
  if  $y_{\hat{p}}(n)$  不等于  $y_{\hat{p}}(n-1)$ ; //对于同一水平线上的顶点
    只做一次扫描线
    break;
  else
    置 num 的初值为 0;
    for ii=1 to N
      if  $(y_p(ii) - y_{\hat{p}}(n)) * (y_p(ii+1) - y_{\hat{p}}(n)) < 0$  //找到该顶点有交点的非水平边
        num=num+1;
        Temp(num)=
           $(x_p(ii) - x_p(ii+1)) * (y_{\hat{p}}(n) - y_p(ii+1)) / (y_p(ii) - y_p(ii+1)) + x_p(ii+1)$ ;
        //求交并记录交点信息
        //对顶点进行判断
        if 交点为普通交点
          then 将 Temp 值直接存入 Current 和 Next
        if 交点为顶点类型 1
          then  $Current(num+1) = x_p(Index(n))$ ;
              $Next(num+1) = x_p(Index(n))$ ;
        else if 交点为顶点类型 2
          then  $Next(num+1) = x_p(Index(n))$ ;
              $Next(num+2) = x_p(Index(n))$ ;
        else if 交点为顶点类型 3
          then  $Current(num+1) = x_p(Index(n))$ ;
              $Current(num+2) = x_p(Index(n))$ ;
        else
          置  $cout_1$  初值为 0;
          置  $cout_2$  初值为 0;
          for jj=1 to N
            if  $y_{\hat{p}}(n) = y_{\hat{p}}(jj)$  //顶点为类型 4 或者类型 5
              if 交点为顶点类型 4
                then  $cout_1 = cout_1 + 1$ ;
                    $Next(num + cout_1) = x_p(Index(jj))$ ;
              else if 交点为顶点类型 5
                then  $cout_2 = cout_2 + 1$ ;
                    $Current(num + cout_2) = x_p(Index(jj))$ ;

```

```

end
//对 Next 和 Current 重新排序
Next=Sort(Next);
Current=Sort(Current);
end
for (mm=1 to Size(Current);mm+2) //以 2 为单位循环至
有效点的总个数
    xr1=ForeNext(mm);xr2=ForeNext(mm+1);
    xr3=Current(mm+1);xr4=Current(mm);//得到单元区
域 4 点的 x 坐标值
    yr1=yp(n-1);yr2=yp(n-1);
    yr3=yp(n);yr4=yp(n);//得到单元区域 4 点的 y 坐标
值
    k1=(xr1-xr4)/(yr1-yr4);
    k2=(xr2-xr3)/(yr2-yr3);//得到两条斜边的斜率
    for (yy=(yr1-Step) to yr3;yy+Step) //对单元区域
进行填色
        xr1=xr1-Step*k1;
        xr2=xr2-Step*k2;//依次推导 x 坐标
        Plot((xr1,yy),(xr2,yy));//直线两点间填色
    end

```

```

end
ForeNext=Next;//纪录 Next 的信息至 ForeNext
end

```

4 实验分析

为了解本文算法的效果,在 P III 933 微机上用 Matlab 平台对基于顶点与邻边相关性的多边形填充算法以及文献[2]中的改进后的经典扫描线算法有序边表算法进行了填充运算实验,并对两者效率进行了比较。

首先取如图 2 所示的 13 条边的多边形一个,通过设定顶点的坐标,使多边形的平均边长为 5.556 cm,其中最长边为 9.055 cm,最短边为 1 cm,然后取不同的填充像素点行间距,进行运算对比实验,并统计算法运算时间,用以考察不同填充精度要求的情况下算法的效率,其不同填充精度下两种算法的运算效率见表 1。

表 1 固定多边形的算法运算效率比较

算法	填充像素点间距(cm)					
	0.1	0.05	0.01	0.005	0.002	0.001
有序边表算法用时(s)	0.681	1.082	5.317	11.577	52.416	166.570
新算法用时(s)	0.591	0.981	4.416	8.993	22.122	50.903
有序边表算法用时/新算法用时	1.152	1.103	1.204	1.287	2.369	3.272

同时又将填充像素点行间距固定为 0.005,取不同边数,而边长大致相当的多边形进行对比计算,

以考察边数的变化情况下的算法效率,其两种算法运算时间见表 2。

表 2 固定填充像素点间距的算法运算效率比较

边数(条)	4	7	13	19	25	30	44
平均边长(cm)	5.693	5.047	5.556	5.744	6.312	7.742	8.617
有序边表算法用时(s)	2.264	2.815	5.317	8.086	31.428	77.091	157.787
新算法用时(s)	2.023	2.313	4.416	6.609	11.712	15.052	22.802
有序边表算法用时/新算法用时	1.119	1.217	1.204	1.223	2.683	5.122	6.920

时间计时从做第 1 条扫描线开始,至求出所有的扫描线交点结束,由于后面进行的填色判断和替换都是相同的过程,故运算时间就不再列入其中。

深入分析表 1、表 2 可以发现:

(1)新算法效率总体上比有序边表算法高,且效率提高显著。

(2)新算法充分体现了求交次数少的优势,即在多边形固定的情况下,当多边形平均边长为填充像素点行间距 100 倍的时候,效率的提高还不是很明显(为 1.103 倍),但当平均边长大约为填充像素点行间距 1 000 倍的时候,效率的提高就十分明显(为

3.272 倍)。

(3)在填充像素点行间距固定、多边形的平均边长大致相当的情况下,随着边数的增加,新算法运算效率的提高就愈加明显。

5 结 论

经典的多边形扫描线填充算法以及之后的很多改进算法都采用了扫描线逐条与多边形边求交的方式,虽然思路简洁易懂,但是尚有较多冗余。本文在挖掘多边形顶点与邻边相关特性的基础上,首先定

义了针对填充的典型顶点类型,并提出了新的填充思路;然后进一步设计了新的算法,即一方面以通过顶点的扫描线重新定义了多边形,另一方面构筑了不同类型顶点在算法中的存储方式,从而简化了求交过程。通过实验分析可见,新算法不仅对多边形填充的效率有显著的提高,而且扩展了计算机图形学领域中多边形填充的思路。

参 考 文 献

- 1 D. F. 罗杰斯著. 计算机图形学的算法基础[M]. 梁友栋等译. 北京:科学出版社,1987:73~101.
- 2 羊四清,李思昆. 改进的扫描线多边形填充算法的研究[J]. 数学理论与应用,1999,19(2):47~49.
- 3 Cordon D. Fast polygon scan conversion with medical application [J]. IEEE Computer Graphics and Applications,1994,14(6):20~29.
- 4 Smith Alvy Ray. Tint fill[J]. Computer Graphics, 1979,13(2):276~283.
- 5 倪玉山,林德生. 扩充堆栈的种子点区域填充算法[J]. 复旦学报(自然科学版),2000,39(1):99~103.
- 6 陈建勋,马恒太. 基于链码的种子填充算法[J]. 计算机工程与科学,1997,19(4):49~53.



陆国栋 1963年生,教授,博士生导师。1999年获浙江大学理学博士学位,现为浙江大学工程及计算机图形学研究所副所长。主要研究领域为计算机图形学、智能CAD、工程图样计算机理解等。



谭建荣 1954年生,教授,博士生导师。1992年获浙江大学理学博士学位,现为浙江大学工程及计算机图形学研究所副所长,浙江大学CAD&CG国家重点实验室副主任。主要研究领域为产品信息建模、计算机图形学、计算机辅助设计等。



吴良 1978年生,美国布朗大学博士研究生。主要研究领域为计算方法、凝聚态分析等。



马 辉 1977年生,浙江大学工程及计算机图形学研究所博士研究生。主要研究领域为计算机图形学、计算机辅助设计、产品信息建模等。

E-mail:hma@zju.edu.cn