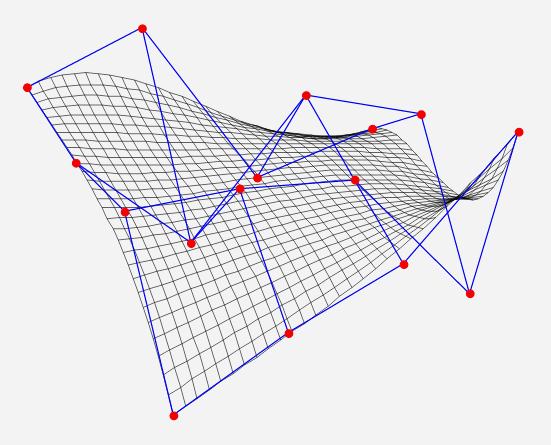
# 计算机图形学

实验4



学号：0x091395

姓名：uknowho

一、 实验目的

1．掌握多边形的扫描转换；

2．掌握区域填充算法。

二、 实验原理

一 多边形的扫描转换

多边形可分为凸多边形、凹多边形、含内环多边形。

（1）凸多边形：任意两顶点间的连线均在多边形内。

（2）凹多边形：任意两顶点间的连线有不在多边形内的部分。

（3）含内环多边形：多边形内包含有封闭多边形。

扫描线多边形区域填充算法是按扫描线顺序，计算扫描线与多边形的相交区间，再用要求的颜色显示这些区间的像素。区间的端点可以通过计算扫描线与多边形边界线的交点获得。对于一条扫描线，多边形的填充过程可以分为4个步骤。

（1）求交：计算扫描线与多边形各边的交点。

（2）排序：把所有交点按x值递增顺序排序。

（3）配对：第一个与第二个，第三个与第四个等，每对交点代表扫描线与多边形的一个相交区间。

（4）填色：把相交区间内的像素置成多边形颜色，把相交区间外的像素置成背景色。

具体实现方法：为多边形的每一条边建立一边表；为了提高效率，在处理一条扫描线时，仅对与它相交的多边形的边进行求交运算。把与当前扫描线相交的边称为活性边，并把它们按与扫描线交点递增的顺序存放在一个链表中，称此链表为活性边表。另外使用增量法计算时，需要知道一条边何时不再与下一条扫描线相交，以便及时把它从扫描线循环中删除出去。为了方便活性边表的建立与更新，为每一条扫描线建立一个新边表（NET），存放在该扫描线第一次出现的边。为使程序简单、易读，这里新边表的结构应保存其对应边如下信息：当前边的边号、边的较低端点（xmin，ymin）与边的较高端点（xmax，ymax）和从当前扫描线到下一条扫描线间x的增量x。

相邻扫描线间x的增量x的计算，假定当前扫描线与多边形某一条边的交点的X坐标为x，则下一条扫描线与该边的交点不要重计算，只要加一个增量x。设该边的直线方程为ax+by+c=0；若y=yi，x=xi；则当y = yi+1时

其中 为常数。

扫描线与多边形顶点相交的处理方法如图1所示。

（1）扫描线与多边形相交的边分别处于扫描线的两侧，则记为一个交点，如点P5，P6。

（2）扫描线与多边形相交的边分别处于扫描线同侧，且yi<yi–1，yi<yi+1，则计两个交点（填色），如P2，若yi>yi–1，yi>yi+1，则计0个交点（不填色），如P1。

（3）扫描线与多边形边界重合（当要区分边界和边界内区域时需特殊处理），则计1个交点。

具体实现时，只需检查顶点的两条边的另外两个端点的y值。按这两个y值中大于交点y值的个数是0,1,2来决定。

算法步骤如下：

（1）初始化：构造边表。

（2）对边表进行排序，构造活性边表。

（3）对每条扫描线对应的活性边表中求交点。

（4）判断交点类型，并两两配对。

（5）对符合条件的交点之间用画线方式填充。

（6）下一条扫描线，直至满足扫描结束条件。

二 区域填充算法

这里的区域指已表示成点阵形式的填充图形，是像素的集合。区域有两种表示形式：内点表示和边界表示，如图2所示。内点表示，即区域内的所有像素有相同颜色；边界表示，即区域的边界点有相同颜色。区域填充指先将区域的一点赋予指定的颜色，然后将该颜色扩展到整个区域的过程。

区域填充算法要求区域是连通的。区域可分为4向连通区域和8向连通区域，如图3所示。4向连通区域指的是从区域上一点出发，可通过四个方向，即上、下、左、右移动的组合，在不越出区域的前提下，到达区域内的任意像素；8向连通区域指的是从区域内每一像素出发，可通过8个方向，即上、下、左、右、左上、右上、左下、右下这八个方向的移动的组合来到达。

1．区域填充的递归算法

上面讨论的多边形填充算法是按扫描线顺序进行的。种子填充算法则是假设在多边形内有一像素已知，由此出发利用连通性填充区域内的所有像素。一般采用多次递归方式。

2．区域填充的扫描线算法

算法的基本过程如下：给定种子点（x，y），首先填充种子点所在扫描线上给定区域的一个区段，然后确定与这一区段相连通的上、下两条扫描线上位于给定区域内的区段，并依次保存下来。反复这个过程，直到填充结束。

区域填充的扫描线算法可由下列3个步骤实现。

（1）初始化：确定种子点元素（x，y）。

（2）判断种子点（x，y）是否满足非边界、非填充色的条件，若满足条件，以y作为当前扫描线沿当前扫描线向左、右两个方向填充，直到边界。

（3）确定新的种子点：检查与当前扫描线y上、下相邻的两条扫描线上的像素。若存在非边界、未填充的像素，则返回步骤（2）进行扫描填充。直至区域所有元素均为填充色，程序结束。

扫描线填充算法提高了区域填充的效率。

三、 实验内容及结果

（1）创建应用程序框架，以上述单文档程序框架为基础，创建如图4所示应用程序界面。

（2）编辑菜单资源。

（3）添加消息处理函数。

（4）添加程序结构代码。

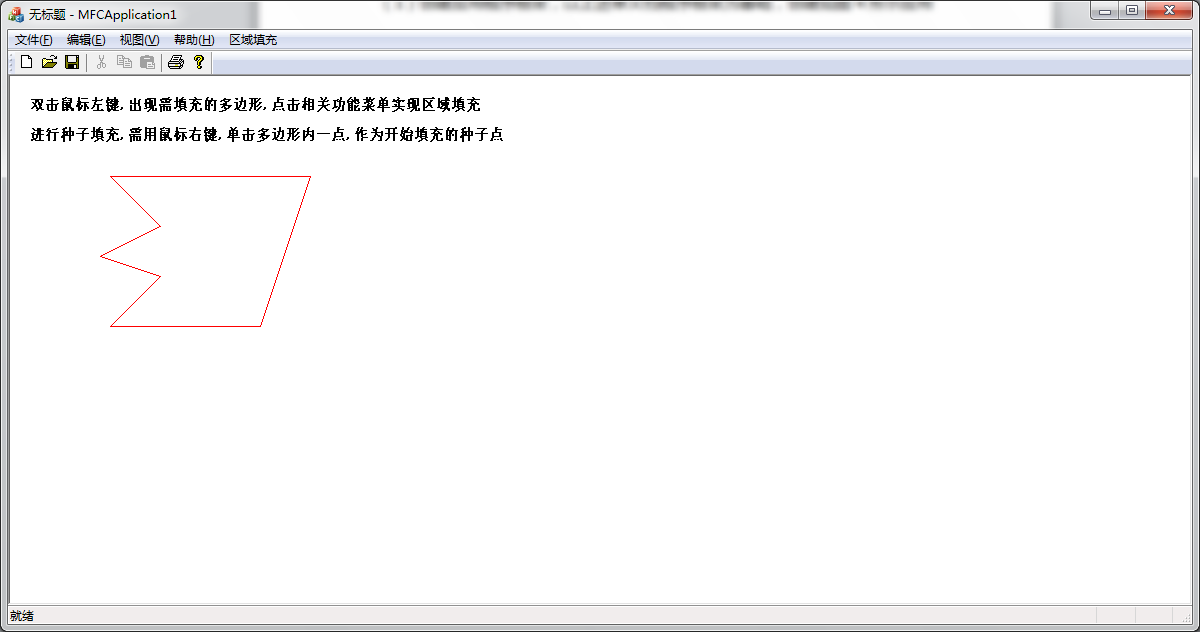


图1 实验结果

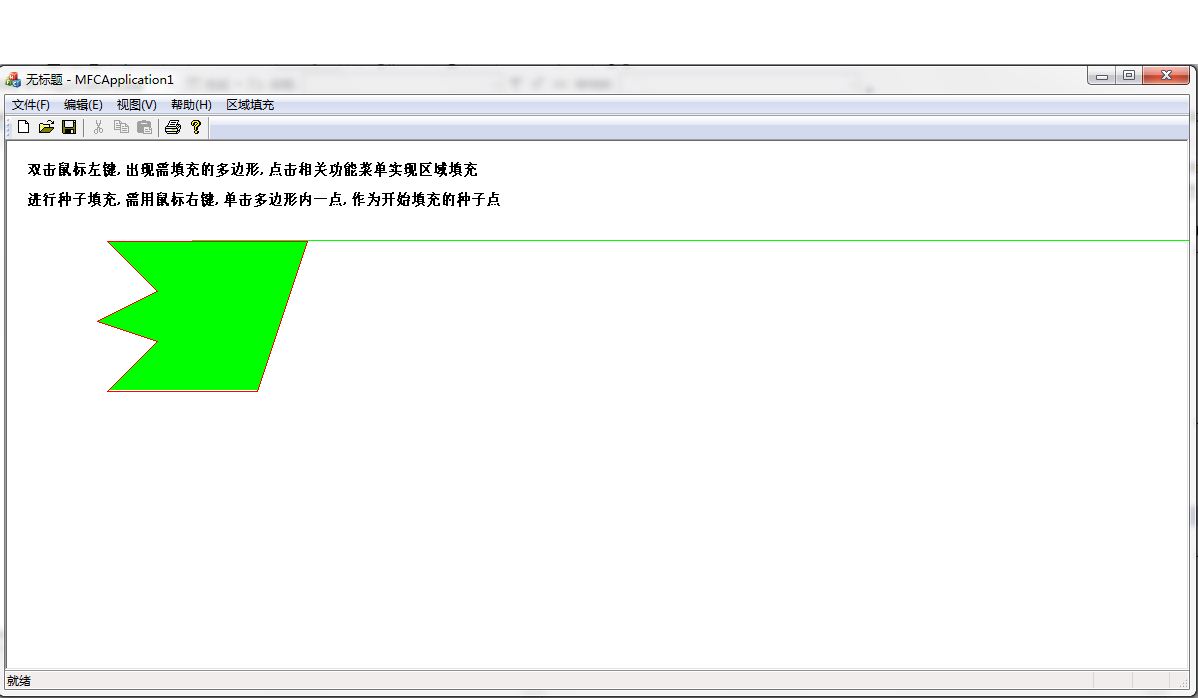


图2 实验结果