

2.3 多边形的扫描转换与区域填充

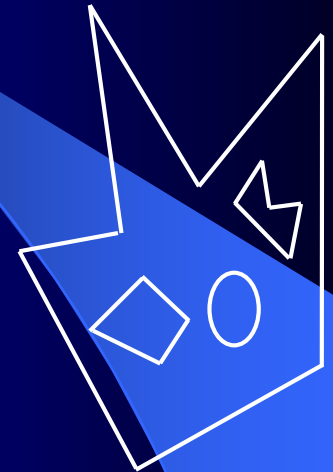
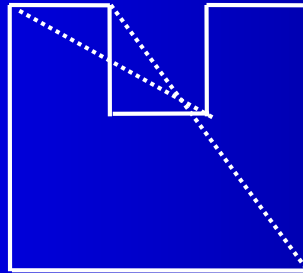
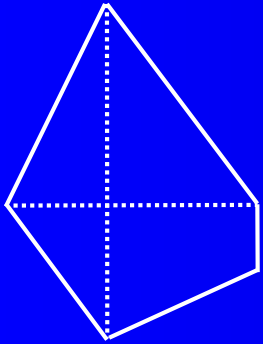
多边形有两种重要的表示方法：顶点表示和点阵表示。

多边形的扫描转换：把多边形的顶点表示转换为点阵表示。

区域可采用内点表示和边界表示两种表示形式。

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。

多边形分为凸多边形、凹多边形、含内环的多边形。



两条对角线的交点可能在多边形的外部

2.3.1 多边形的扫描转换

2.3.1.1 扫描线算法

– 基本思想:

y坐标间隔为1的水平线。扫描线
纵坐标为y，也称为第y号扫描线

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，即完成填充工作。

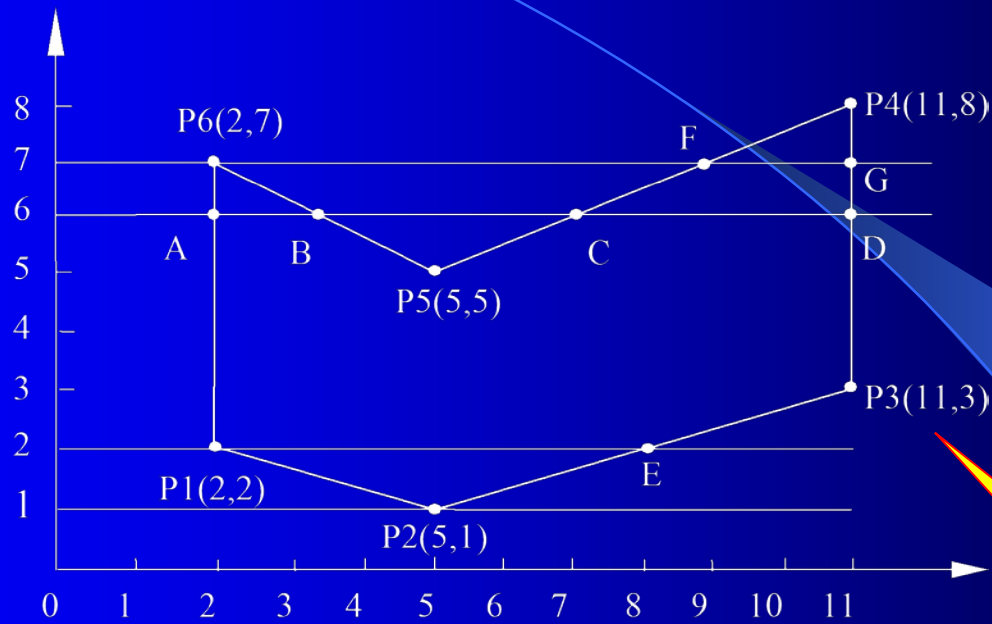
– 对于一条扫描线填充过程可以分为四个步骤:

(1) 求交

(2) 排序

(3) 配对

(4) 填色



一个多边形与若干扫描线

假定从下
向上扫描

数据结构

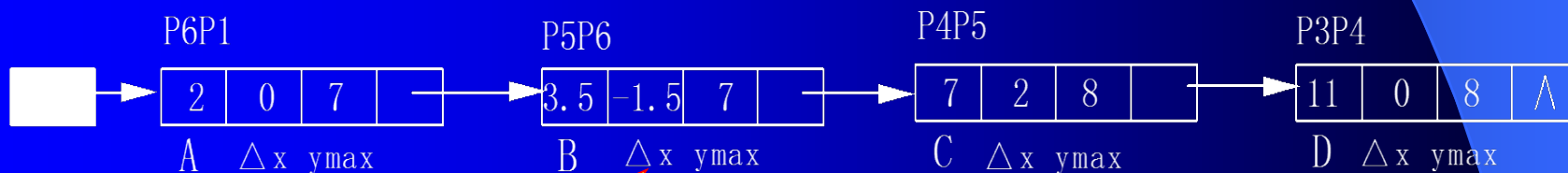
- 活性边表(AET: Active Edge Table): 把与当前扫描线相交的边称为活性边, 并把它按与扫描线交点 x 坐标递增的顺序存放在一个链表中

- 结点内容

x : 当前扫描线与边的交点横坐标

Δx : 从当前扫描线到下一条扫描线间 x 的增量

y_{max} : 通过该边的最高扫描线(y 坐标最大的扫描线)的坐标



可用增量法计算

假定当前扫描线与多边形某一条边的交点的横坐标为 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 Δx 。

设该边的直线方程为：

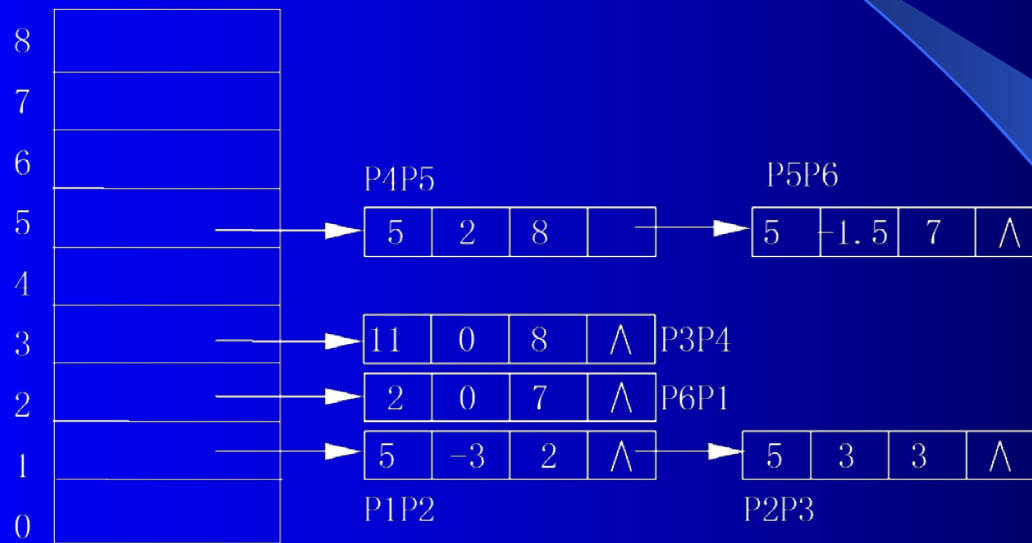
$$ax + by + c = 0$$

若 $y = y_i$ 时 $x = x_i$ ，则当 $y = y_{i+1}$ 时，

$$x_{i+1} = \frac{1}{a}(-b \cdot y_{i+1} - c) = x_i - \frac{b}{a};$$

所以 $\Delta x = -\frac{b}{a}$ ，且为常数

- 新边表 (NET: New Edge Table) : 存放与扫描线第一次相交的边。若某边的较低端点(即纵坐标较小的端点)为 $ymin$, 则该边就在扫描线 $ymin$ 的新边表中



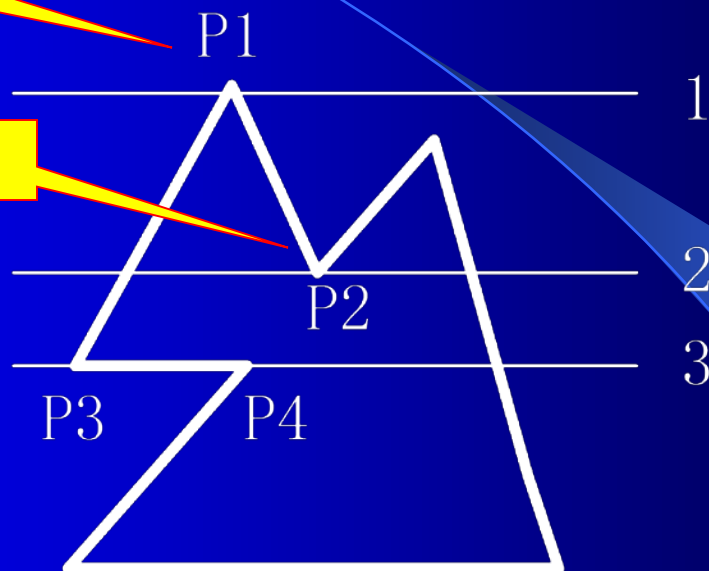
上图所示各条扫描线的新边表 NET

扫描线与多边形的顶点或边界相交时，必须正确地取舍交点。

- 与扫描线相交的边的另外两个端点分别位于扫描线的两侧，则计一个交点
- 与扫描线相交的边的另外两个端点均位于扫描线的同侧：
 - (1) $y_i < y_{i-1}$ 且 $y_i < y_{i+1}$ ，则计2个交点(填色)，如P2
 - (2) $y_i > y_{i-1}$ 且 $y_i > y_{i+1}$ ，则计0个交点(不填色)，如P1
- 水平边不起作用，在预处理中去掉

$P1$ 计0个交点

$P2$ 计2个交点



算法过程

void polyfill (多边形 polygon, int color)

{

for (各条扫描线*i*){

NET是一维数组

初始化新边表头指针NET [*i*];

建立新
边表

把 $y_{min} = i$ 的边放进边表NET [*i*];

}

即多边形各顶点
中最小的y坐标

$y =$ 最低扫描线号;

初始化活性边表AET为空;

AET是指针

每一次循环，给配对的交点区间填色，并得到下一个活性边表

for (各条扫描线 i) {

(1) 把新边表NET [i] 中的边结点插入AET表，并按 x 坐标递增顺序排列；

(2) 遍历AET表，把配对交点区间(左闭右开)上的象素(x, y)，用drawpixel (x, y, color) 改写象素颜色值；

(3) 遍历AET表，把 $y_{\max} = i$ 的结点从AET表中删除，并把 $y_{\max} > i$ 的结点的 x 值递增 Δx ；

}

} /* polyfill */