

Vector to Raster Conversion

# 矢量-栅格转换



# 中国人民解放军战略支援部队 信息工程大学—曹一冰讲师

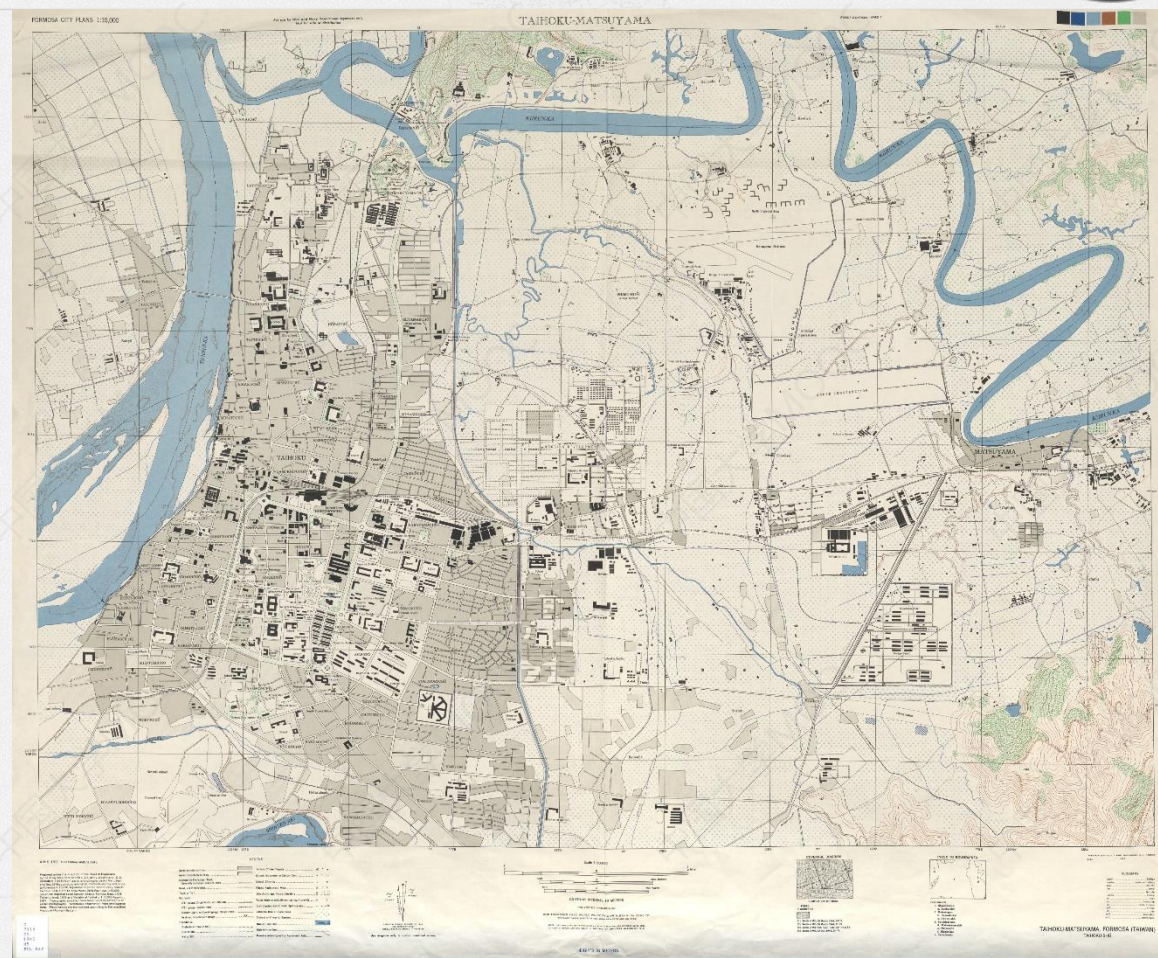
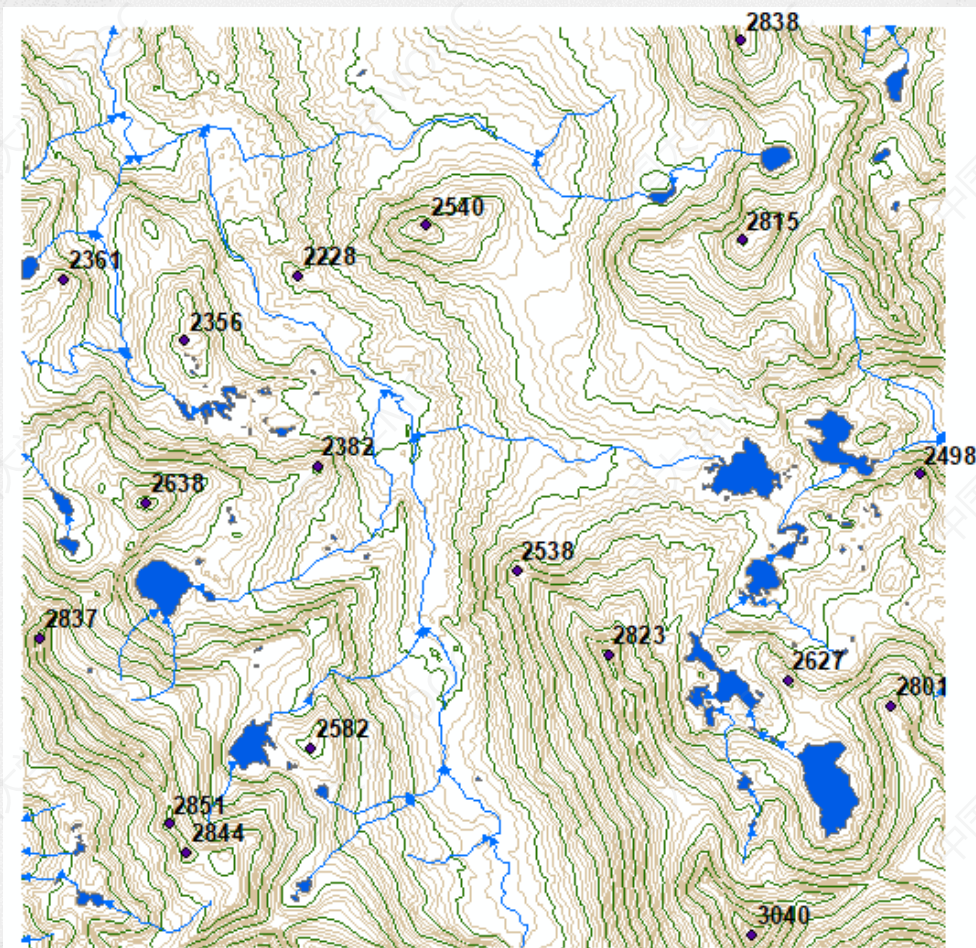
PLA Strategic Support Force Information Engineering University——Lecturer, Yibing Cao

- 获军队科技进步二等奖1项、三等奖1项。
- 获第五届全国高校GIS青年教师讲课比赛一等奖，指导第九届全国大学生GIS应用技能大赛获特等奖。
- 近五年来，主持国家重点研发计划项目子课题2项，发表学术论文10篇，受理国家发明专利9项，获得计算机软件著作权7项。



# 矢量-栅格转换

Vector to Raster Conversion



矢量数据

栅格数据



# 矢量-栅格转换

Vector to Raster Conversion



作用：

- (1) 矢量-栅格转换主要用于地图输出、栅格数据空间分析、DEM联合分析等场景；
- (2) 栅格-矢量转换主要用于数字地图更新、矢量数据空间分析等场景。



以应用为导向，根据需求实现高效转换



# 目录

## Part 1 点的栅格化

Rasterization of Points

## Part 2 线的栅格化

Rasterization of Polyline

## Part 3 面的栅格化

Rasterization of Polygon

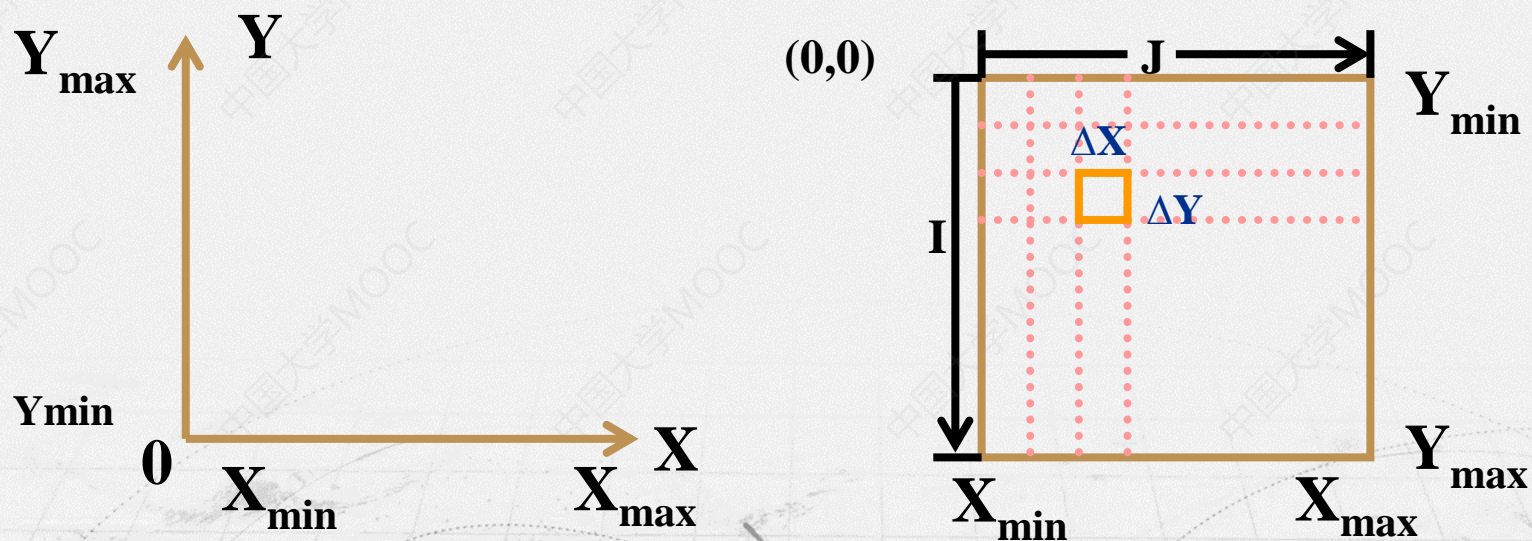


# 点的栅格化





点的栅格化实质上就是将点的坐标转换成栅格数据中的行、列值I和J，从而得到点所在栅格元素的位置。





行  $I = 1 + \text{integer}((Y_{\max} - Y)/\Delta Y)$

列  $J = 1 + \text{integer}((X - X_{\min})/\Delta X)$

X、Y为所求点在矢量坐标系下的坐标值。

$\Delta X$ 、 $\Delta Y$ 分别为每个栅格单元对应边长。

$X_{\min}$ 、 $Y_{\max}$ 标识矢量数据的X最小值及Y最大值。

I、J为所求点所在栅格坐标系中的行列值。





# 线的栅格化

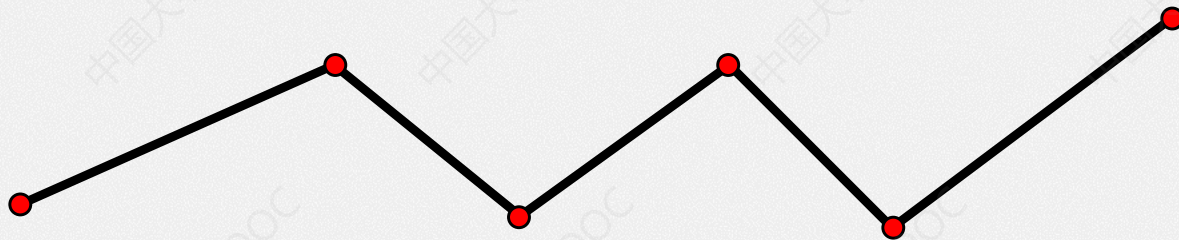




# 02

## 线的栅格化

Rasterization of Polyline



线是由多个直线段组成，线段由两点之间的连线组成，因此，线栅格化的核心是直线段的栅格化。





## 02

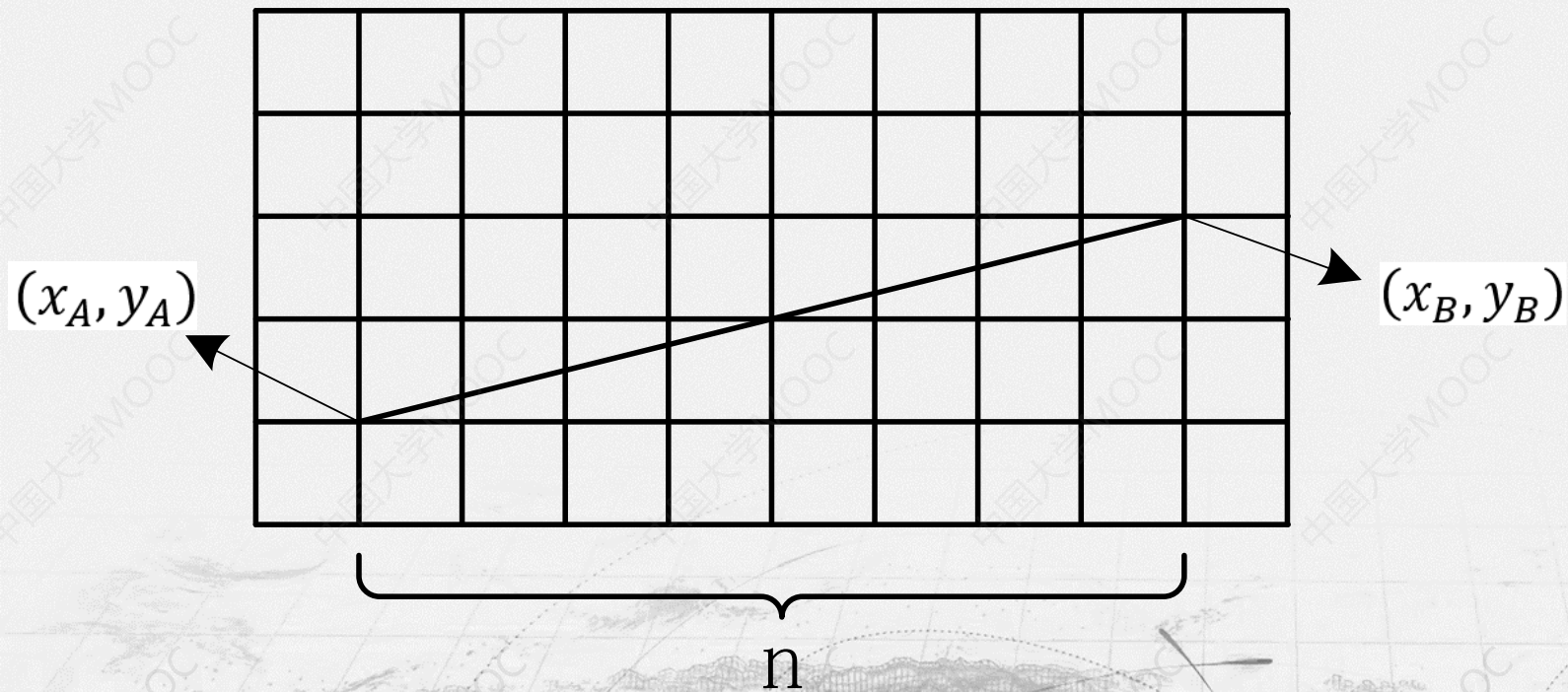
## 线的栅格化

Rasterization of Polyline



## 数字微分分析法

(Digital Differential Analyzer, DDA)



$$\begin{cases} x_{i+1} = x_i + \frac{x_B - x_A}{n} = x_i + \Delta x \\ y_{i+1} = y_i + \frac{y_B - y_A}{n} = y_i + \Delta y \end{cases}$$

$$n = \max(|x_B - x_A|, |y_B - y_A|)$$

$$\Delta x = \frac{x_B - x_A}{n}; \Delta y = \frac{y_B - y_A}{n}$$

$$x_0 = x_A; y_0 = y_A; x_n = x_B; y_n = y_B$$



# 02 线的栅格化

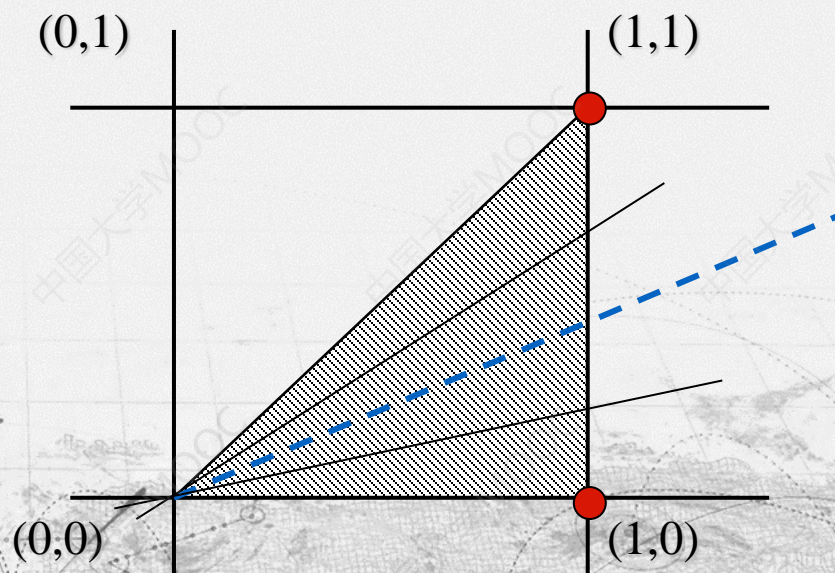
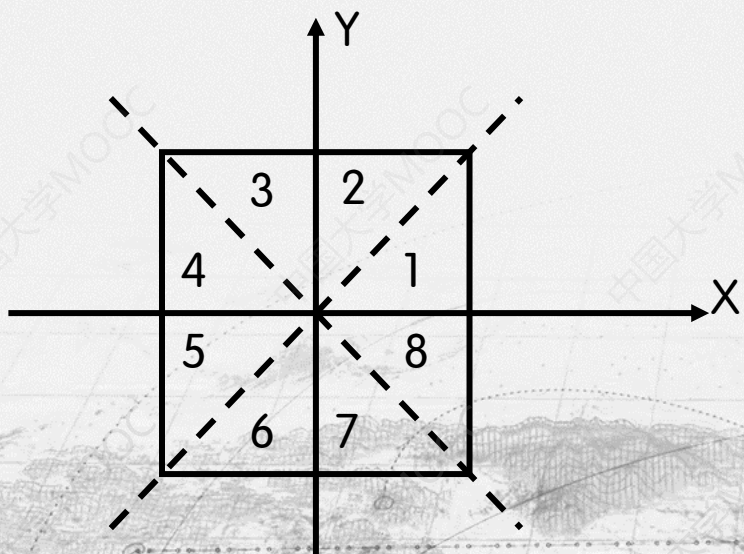
Rasterization of Polyline



## Bresenham算法:

根据直线的斜率，把直线分为8个象限；

以斜率在第一象限的直线段为例，若直线的斜率为  $1/2 \leq \Delta y / \Delta x \leq 1$ ，则下一点取(1, 1)点，若  $0 \leq \Delta y / \Delta x < 1/2$ ，则下一点取(1, 0)点。





## Bresenham算法:

在算法实现时，令起始的误差项为

$e = -1 / 2$ ，然后在推断出下一点后，令

$e = e + \Delta y / \Delta x$ ，

若 $e \geq 1 / 2$ 时， $e = e - 1$ 。这样只要根据 $e$ 的符号就可确定下一点的坐标值，即：

若 $e \geq 0$ ，取 $(1, 1)$ 点；

若 $e < 0$ ，取 $(1, 0)$ 点；

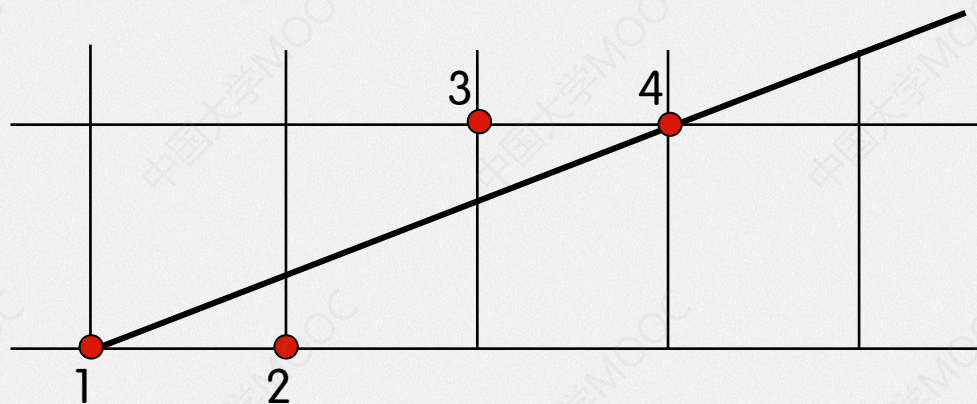


# 02 线的栅格化

Rasterization of Polyline



## Bresenham算法:



例如，一直线的斜率为 $1/3$ 。

起始点： $e = -1/2$ ，取点1

第2点： $e = -1/2 + 1/3 = -1/6 < 0$ ，取点2

第3点： $e = -1/6 + 1/3 = 1/6 > 0$ ，取点3

第4点： $e = 1/6 + 1/3 = 1/2 > 0$ ，取点4

因 $e \geq 1/2$ ，所以， $e = 1/2 - 1 = -1/2$ 。

依次进行，直到到达直线的另一端点。







# 面的栅格化





面矢量数据转成栅格数据要分两步走，一方面要把矢量多边形的边界转成栅格数据，另一方面还必须对其面域进行填充。

- 内部点扩散法
- 扫描法
- 边填充法



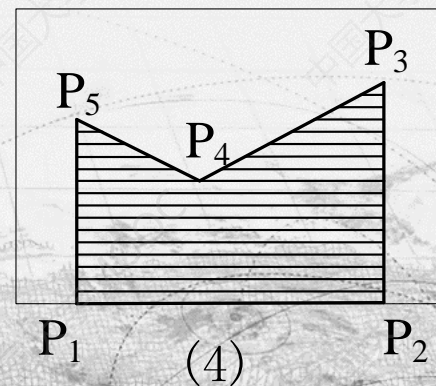
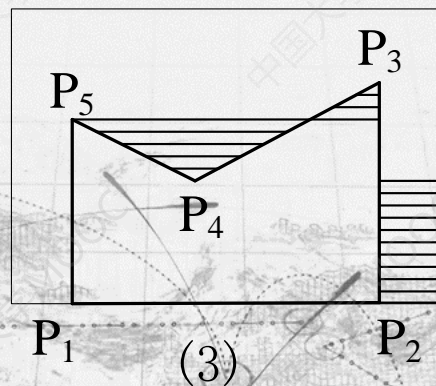
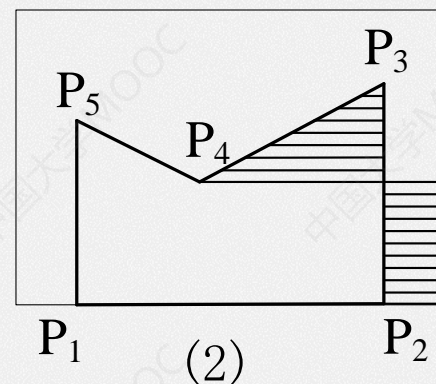
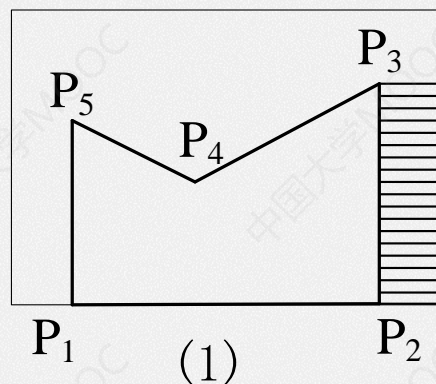
# 03 面的栅格化

Rasterization of Polygon



**基本思想：** 对于每条扫描线和每条多边形边的交点，将该扫描线上交点右方的所有像素取补。对多边形的每条边作此处理，与边的顺序无关。

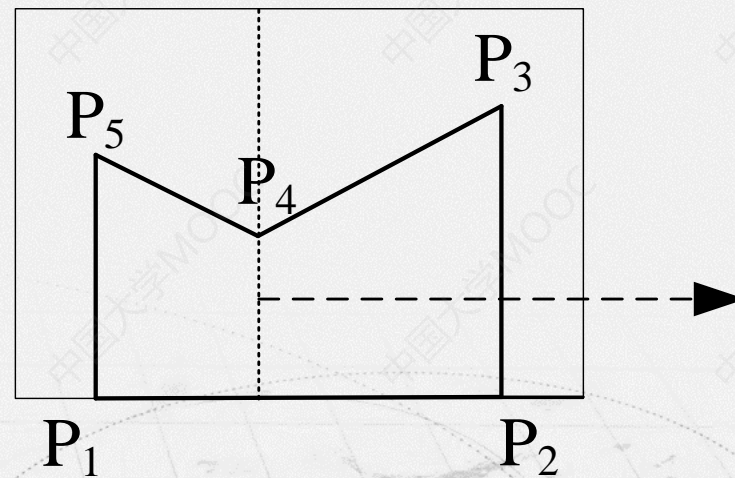
栅栏





所谓**栅栏**指的是一条与扫描线垂直的直线，栅栏位置通常取多边形的顶点，且把多边形分为左右两半。

**栅栏填充算法的基本思路：**对于每条扫描线与多边形的交点，将交点与栅栏之间的像素用多边形的属性值取补。若交点位于栅栏的左边，则将交点右边，栅栏左边的所有像素取补；若交点位于栅栏的右边，则将栅栏右边，交点左边的像素取补。



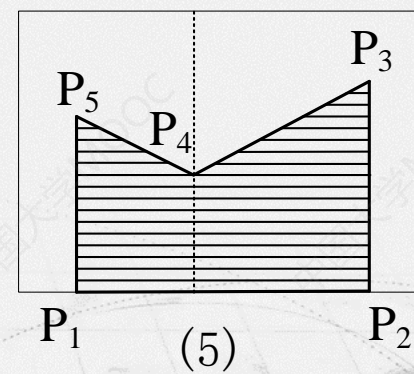
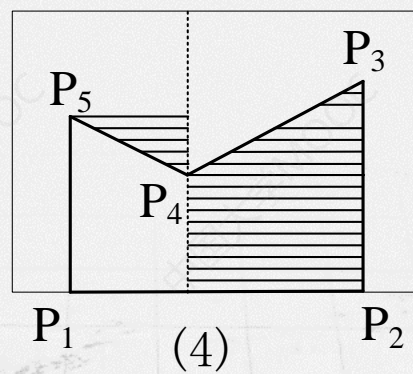
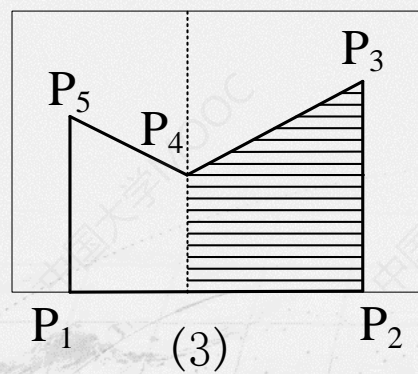
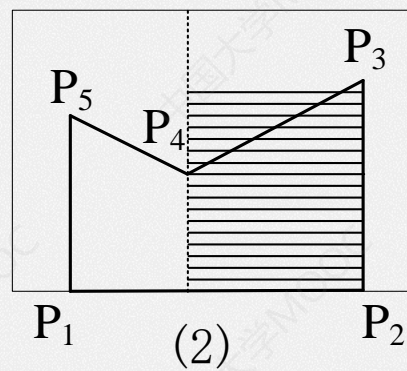
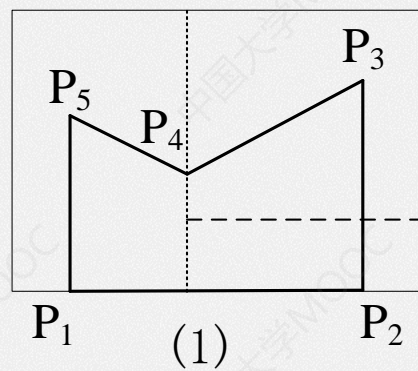


# 03 面的栅格化

Rasterization of Polygon



## 作业题





# 矢量-栅格转换

Vector to Raster Conversion



小结

空间数据结构转换的作用

矢量-栅格转换

点的栅格化

线的栅格化

面的栅格化



谢谢观看