

遇见多边形（中）： Y向连贯性算法

华中科技大学软件学院 万琳



提纲

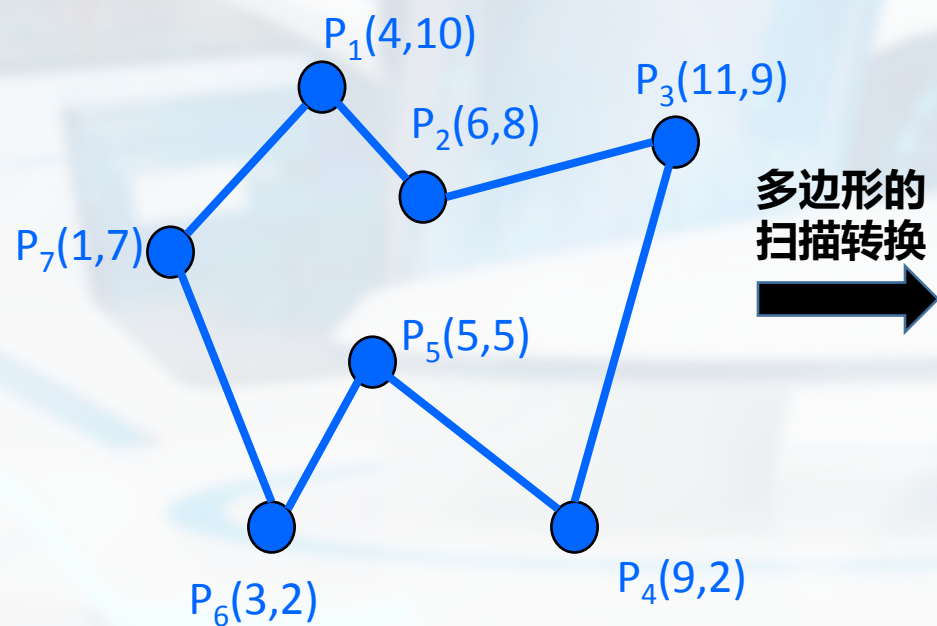
- ① 改进思想
- ② 数据结构
- ③ 算法过程

1

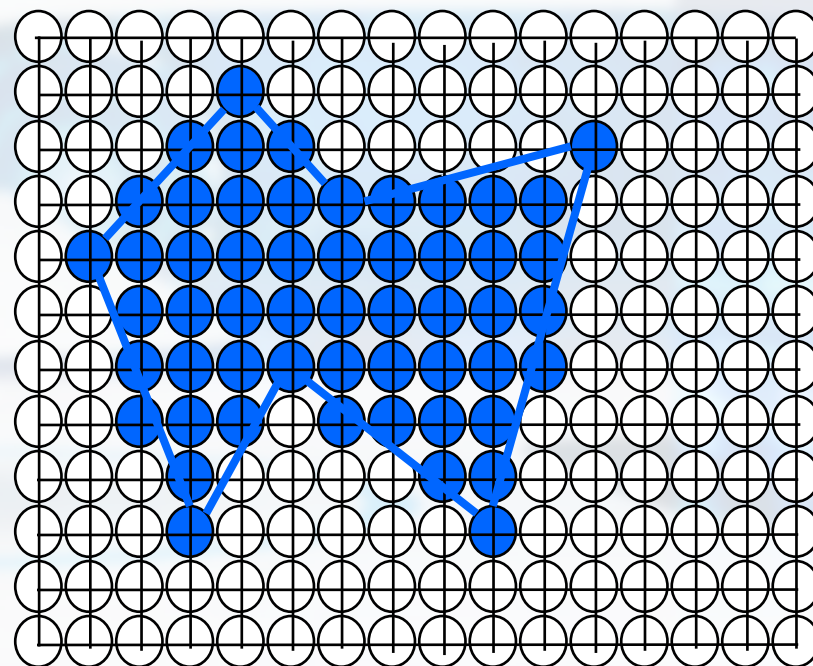
改进思想

输入：多边形顶点序列 $P_1(x_1, y_1)$ 到 $P_7(x_7, y_7)$

输出：最佳逼近这个多边形的像素点集



多边形的
扫描转换



1

改进思想

改进的出发点：

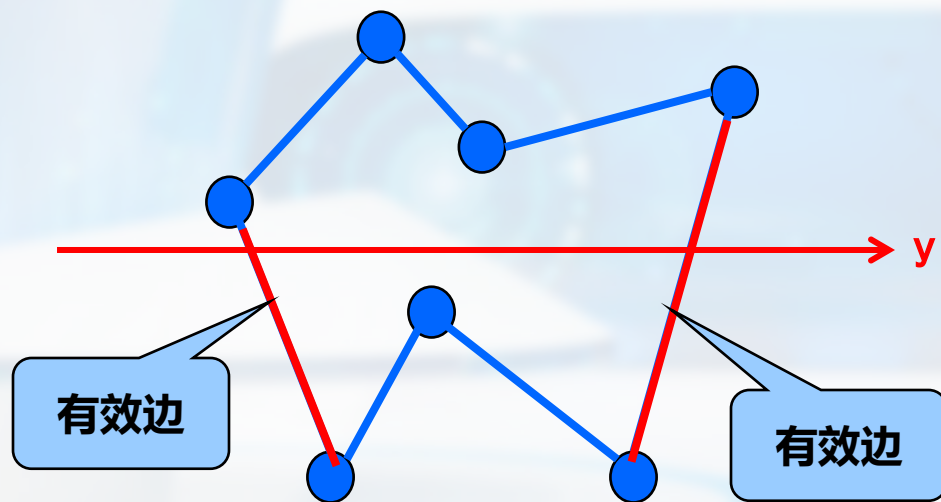
(1)对于某一条扫描线，需要与所有的边求交吗？

a.求交

b.排序

c.交点配对

d.区间填色



3

改进思想

改进的出发点：

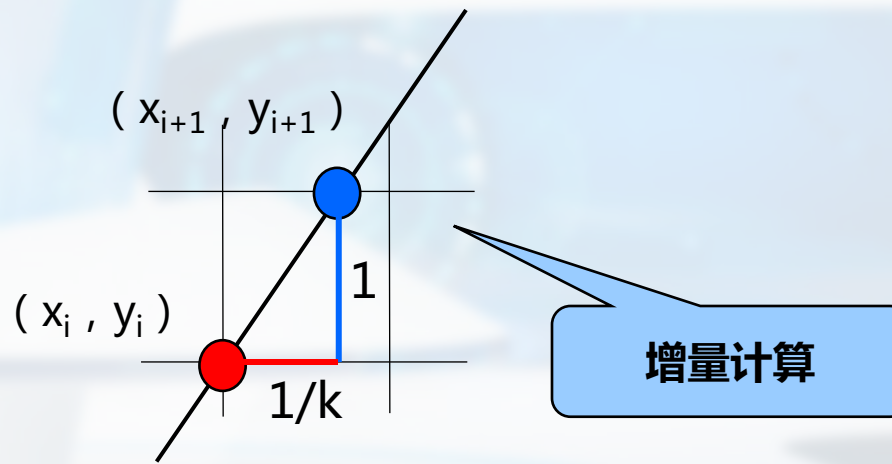
(2)扫描线和直线在Y方向上都有连贯性，那么交点可以怎么求？

a.求交

b.排序

c.交点配对

d.区间填色



1

改进思想

改进的出发点：

(3) 每次都需要排序吗？

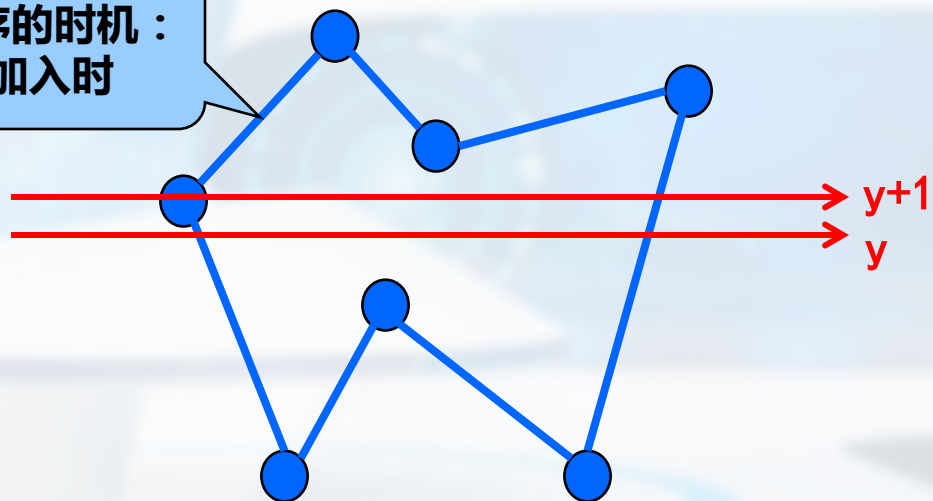
a. 求交

b. 排序

c. 交点配对

d. 区间填色

重新排序的时机：
新边加入时



2

数据结构

边表 (Edge Table) :

每条多边形的边放入首次出现的桶中。

桶

y=10
y=9
y=8
y=7
y=6
y=5
y=4
y=3
y=2

P_2P_1

P_2P_3

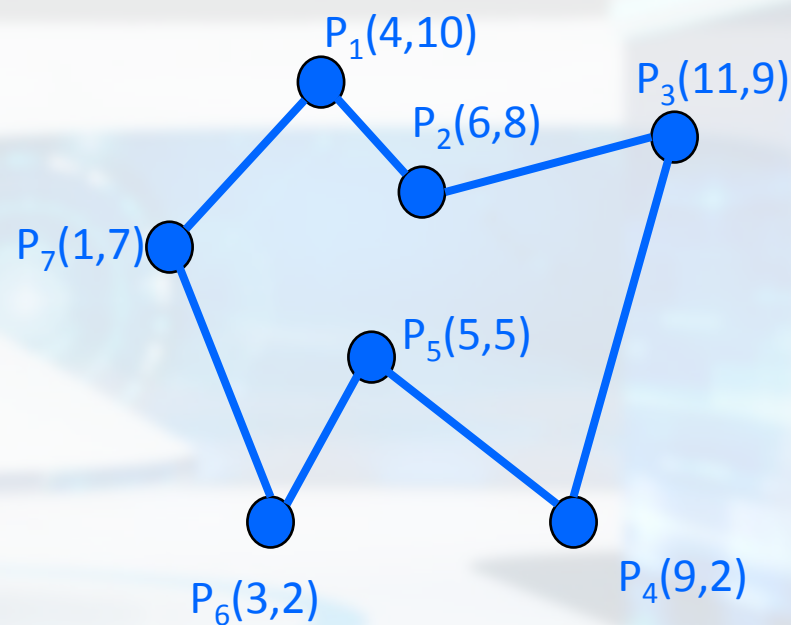
P_7P_1

P_6P_7

P_6P_5

P_4P_5

P_4P_3



2

数据结构

相关定义：

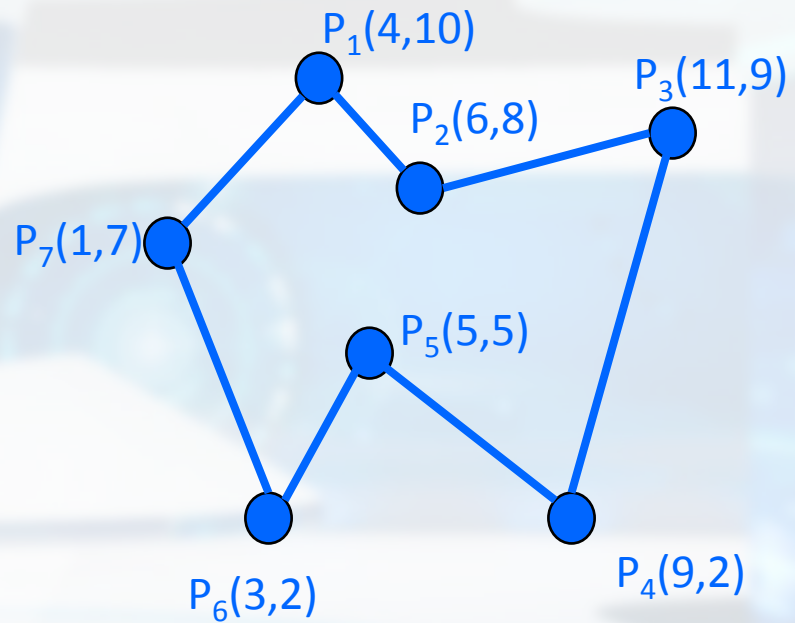
新边 (New Edge) :

每条新边对应一个 结点：

x	y_{\max}	$1/k$	next
---	------------	-------	------

新边结点排序原则：

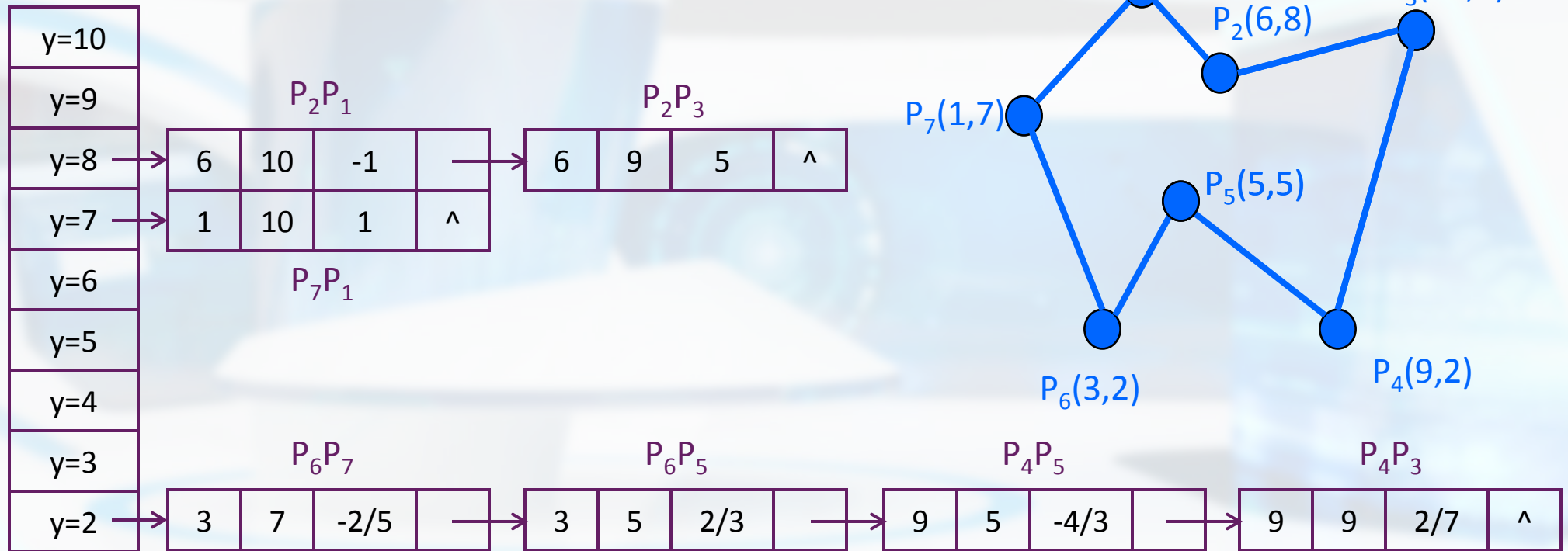
- (1) 交点递增
- (2) 交点相同，增量递增



2

数据结构

边表 (Edge Table) :



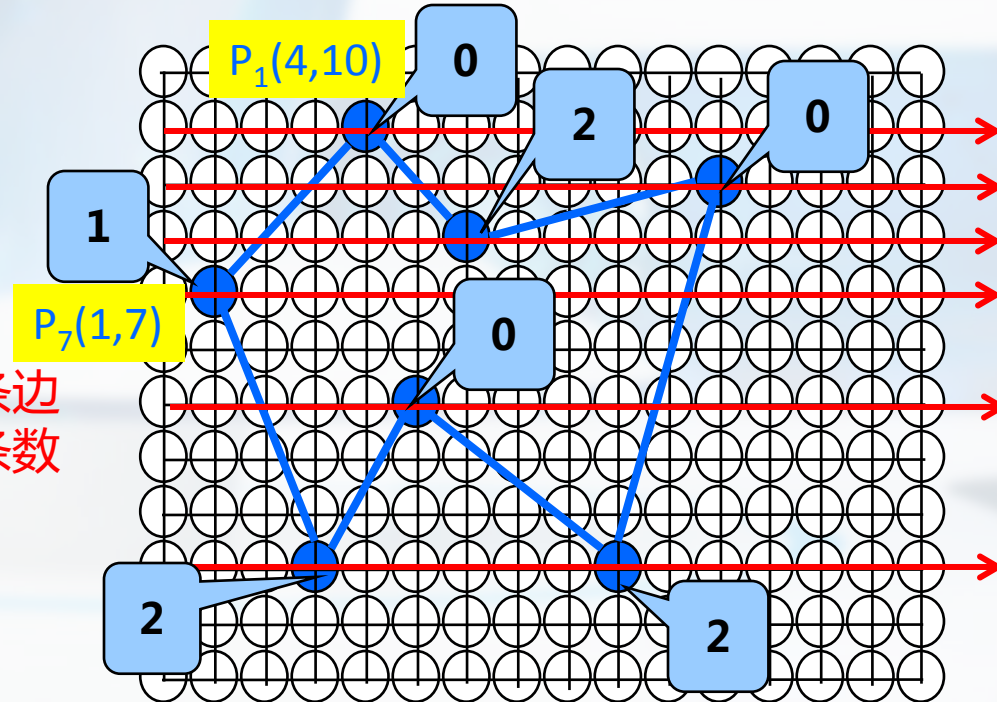
2

数据结构

如何体现：当扫描线与多边形顶点相交时，交点的取舍策略？

策略：

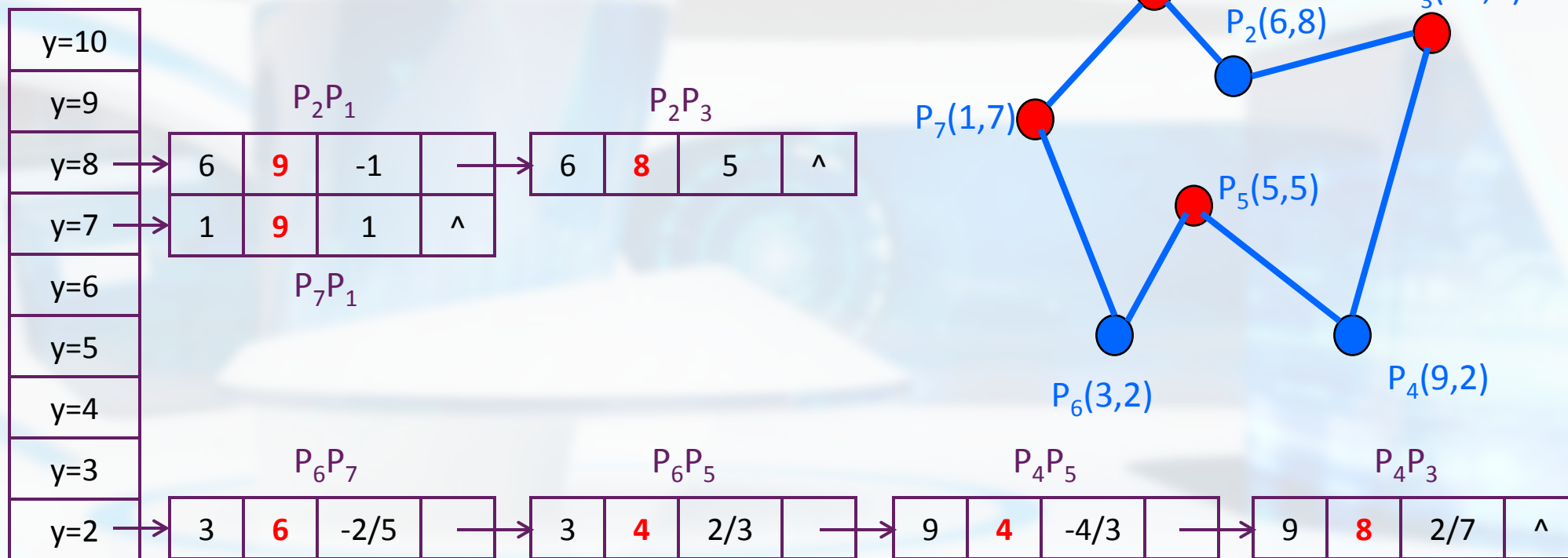
交点个数=
构成这个顶点的两条边
位于扫描线上方的条数



2

数据结构

边表 (Edge Table) :



2

数据结构

相关定义：

有效边 (Active Edge)：

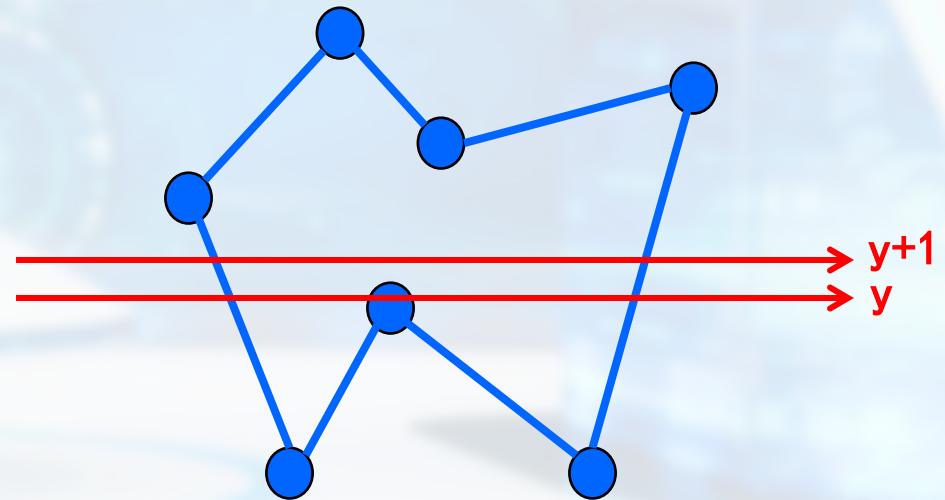
指与当前扫描线相交的多边形的边，也称为活性边。

有效边表 (Active Edge Table, AET)：

把有效边按与扫描线交点x坐标递增的顺序存放在一个链表中，此链表称为有效边表。

有效边表的每个结点：

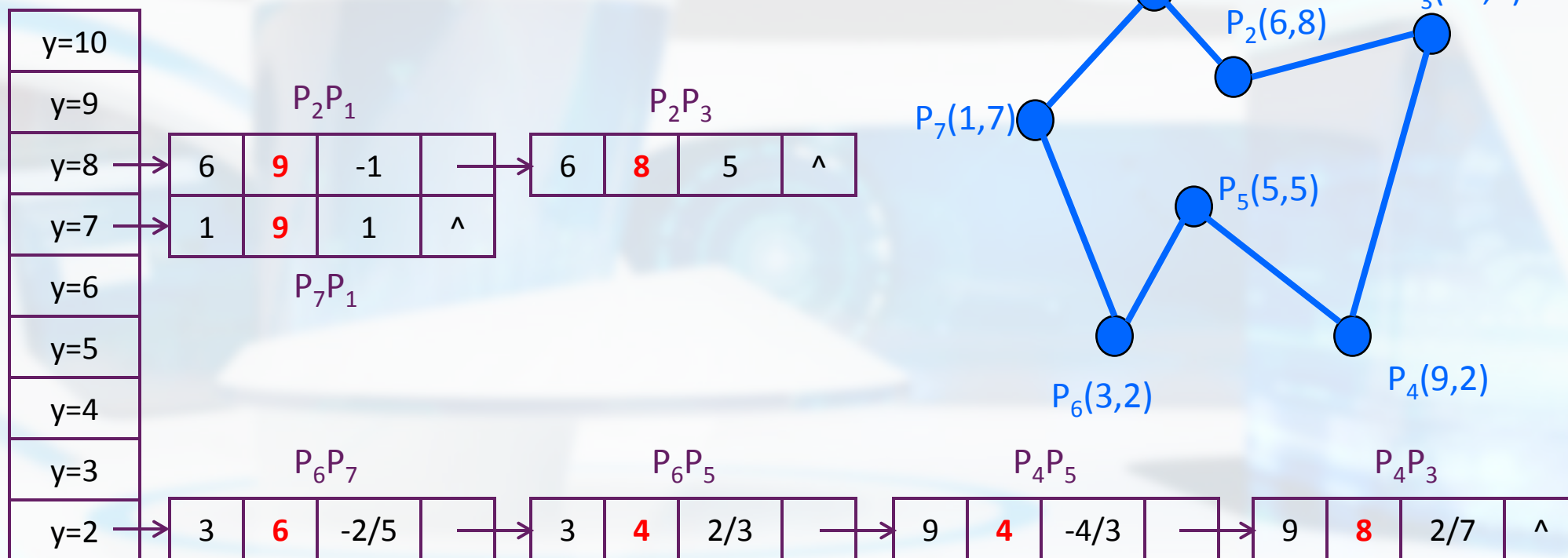
x	y_{\max}	$1/k$	next
---	------------	-------	------



3

算法过程

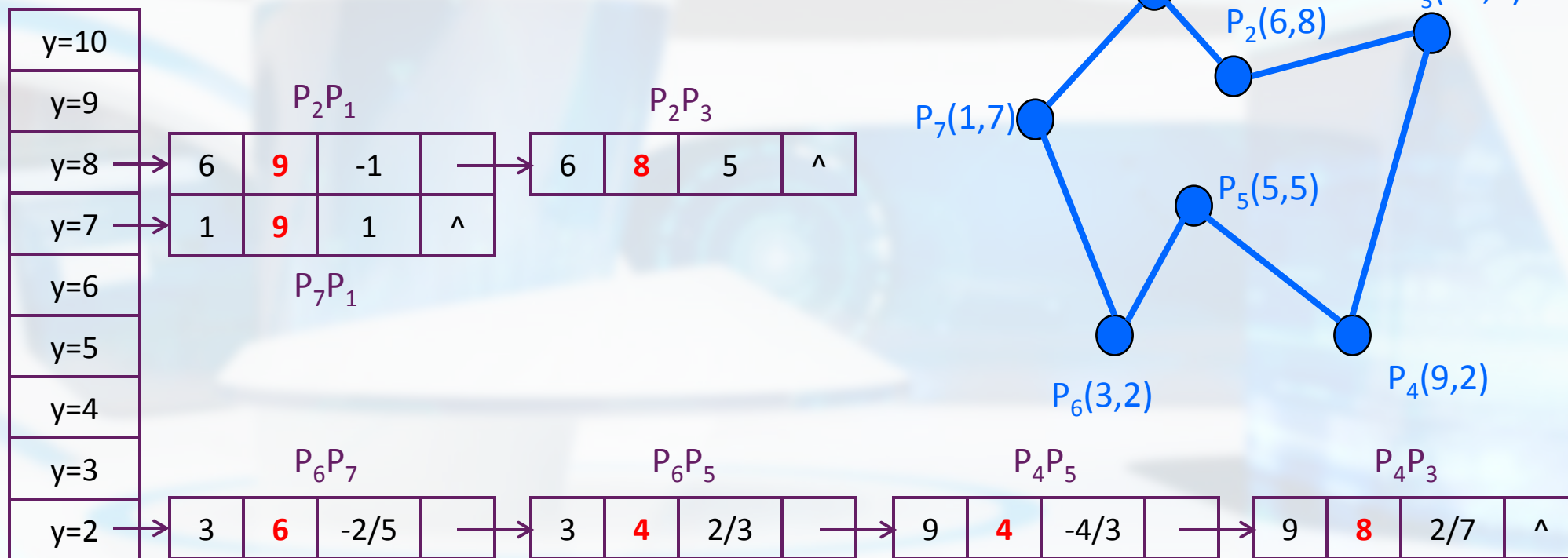
边表 (Edge Table) :



3

算法过程

边表 (Edge Table) :



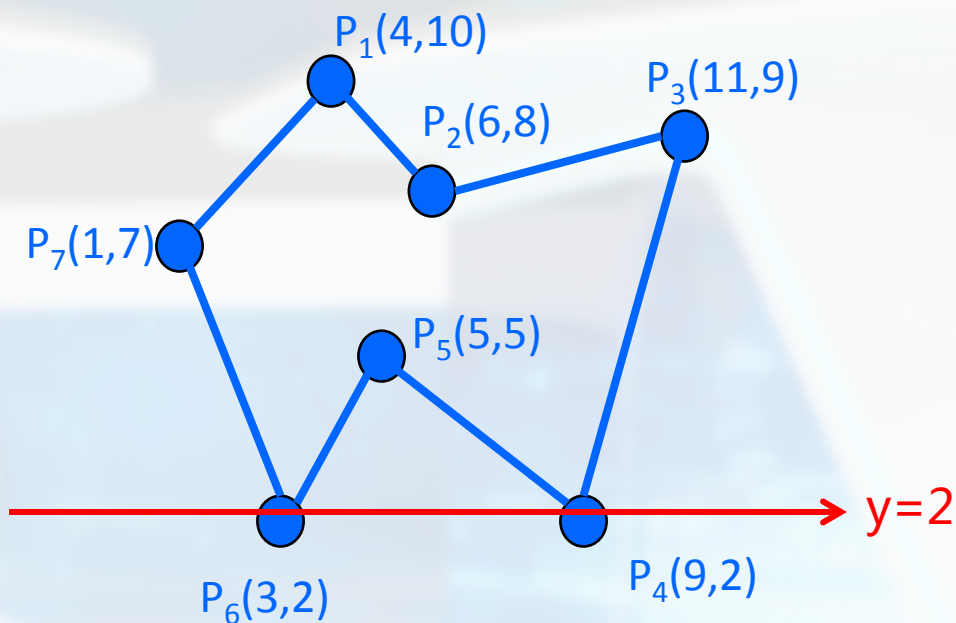
有效边表 (Active Edge Table, AET) : 开始为空

3

算法过程

边表 (Edge Table) :

y=10									
y=9									
y=8	→	6	9	-1	→	6	8	5	^
y=7	→	1	9	1	^				
y=6									
y=5									
y=4									
y=3									
y=2	→	3	6	-2/5	→	3	4	2/3	—

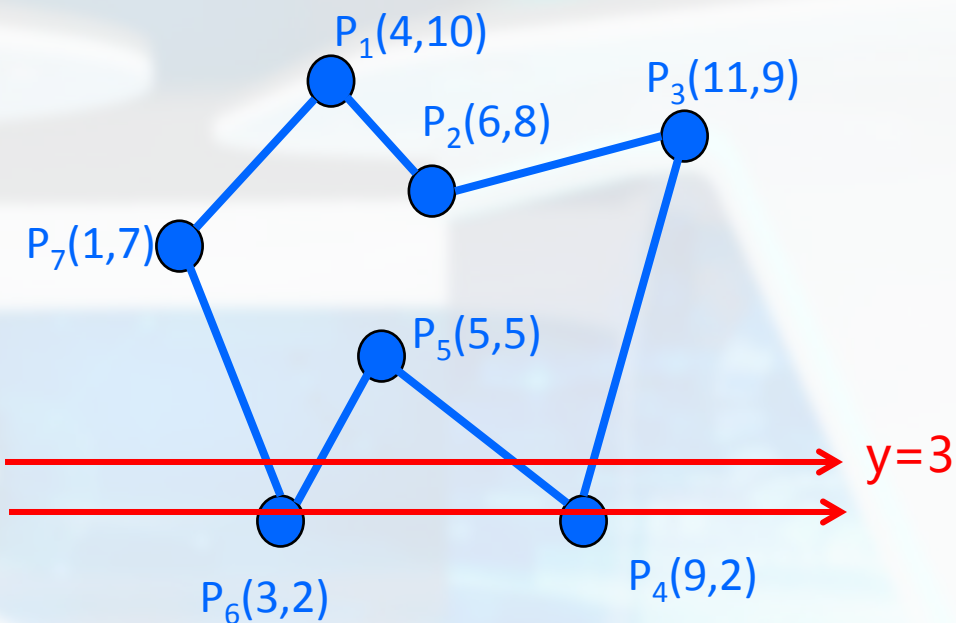
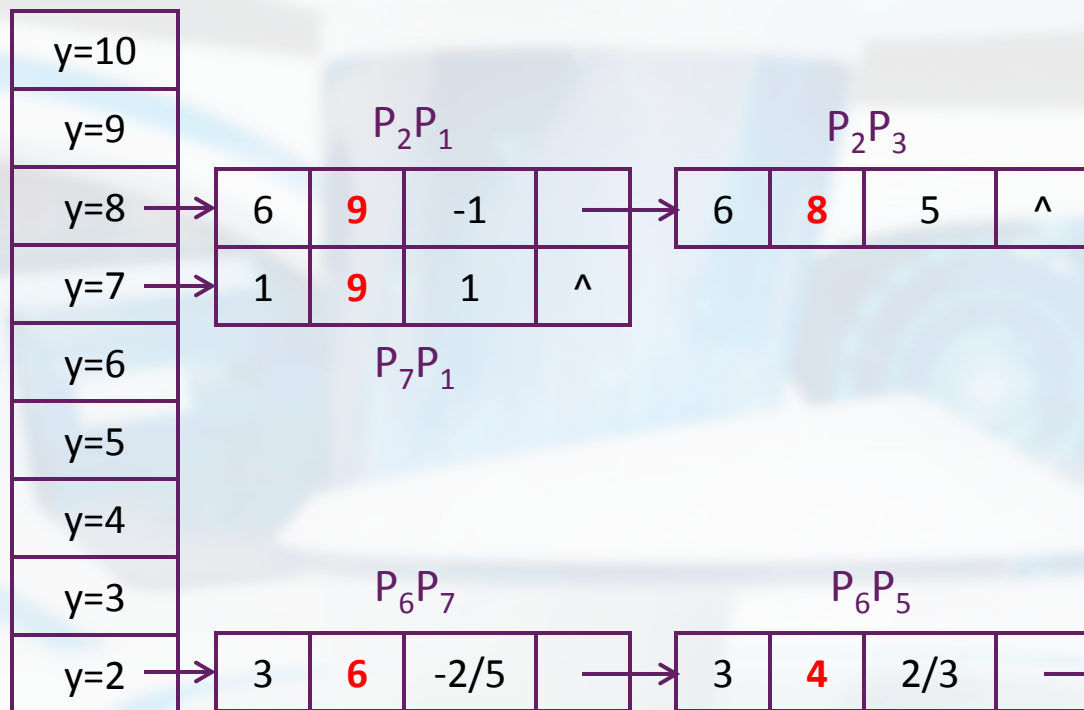


有效边表 (Active Edge Table, AET) : 与边表中 $y=2$ 处边表合并

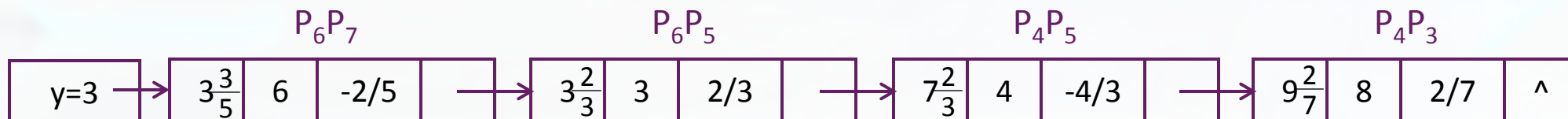
y=2	→	3	6	-2/5	→	3	4	2/3	→	9	4	-4/3	→	9	8	2/7	^

算法过程

边表 (Edge Table) :



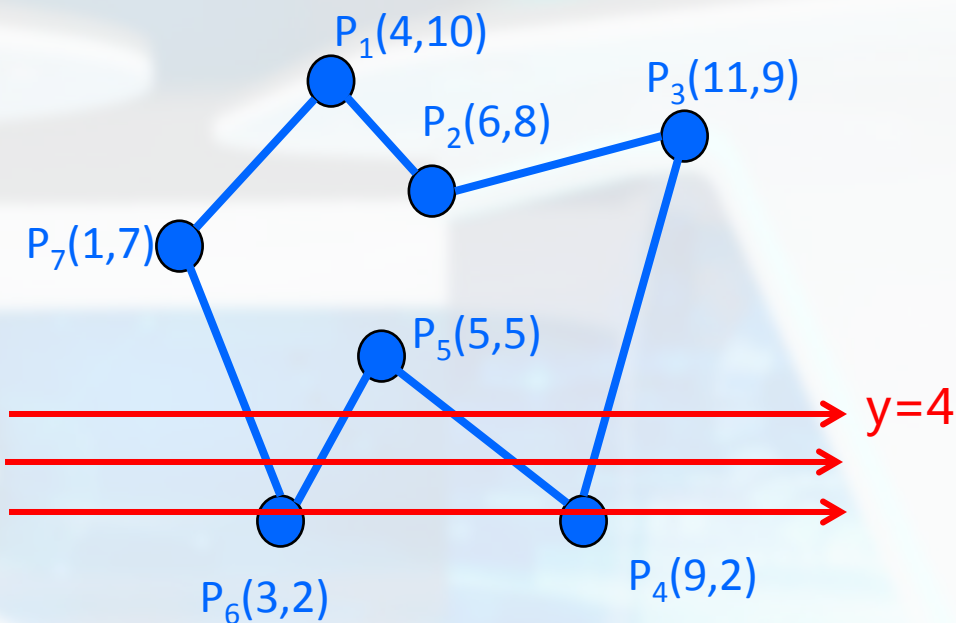
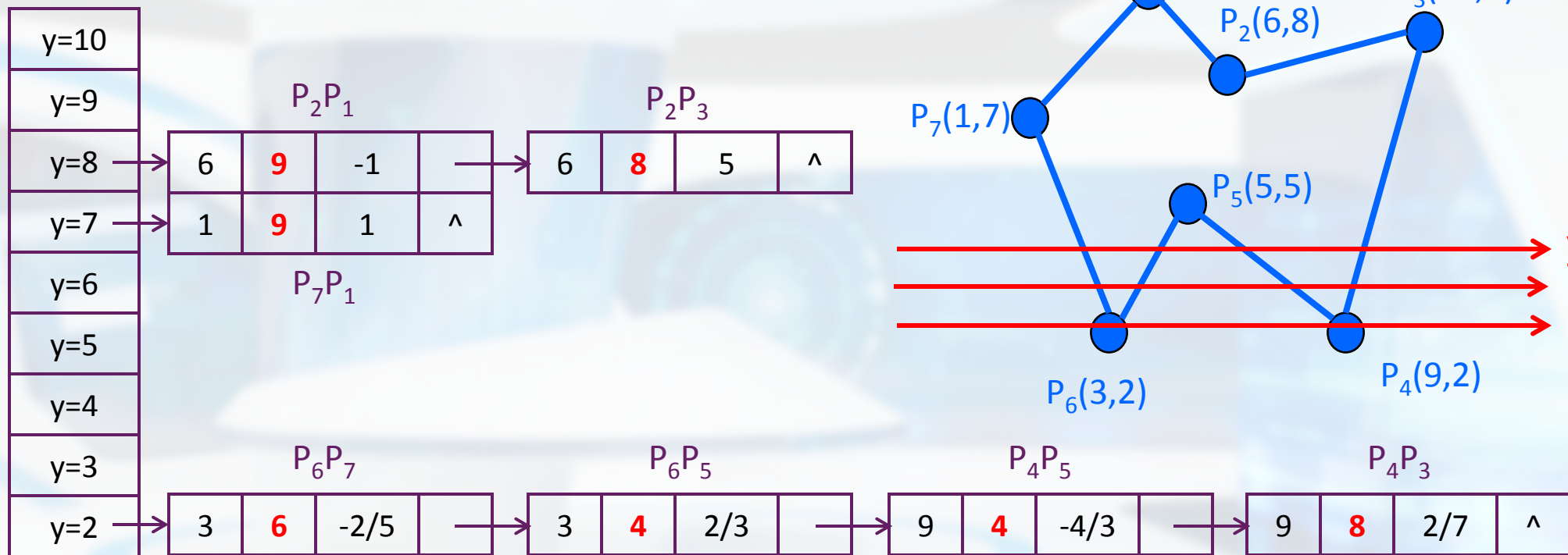
有效边表 (Active Edge Table, AET) : 增量计算



3

算法过程

边表 (Edge Table) :



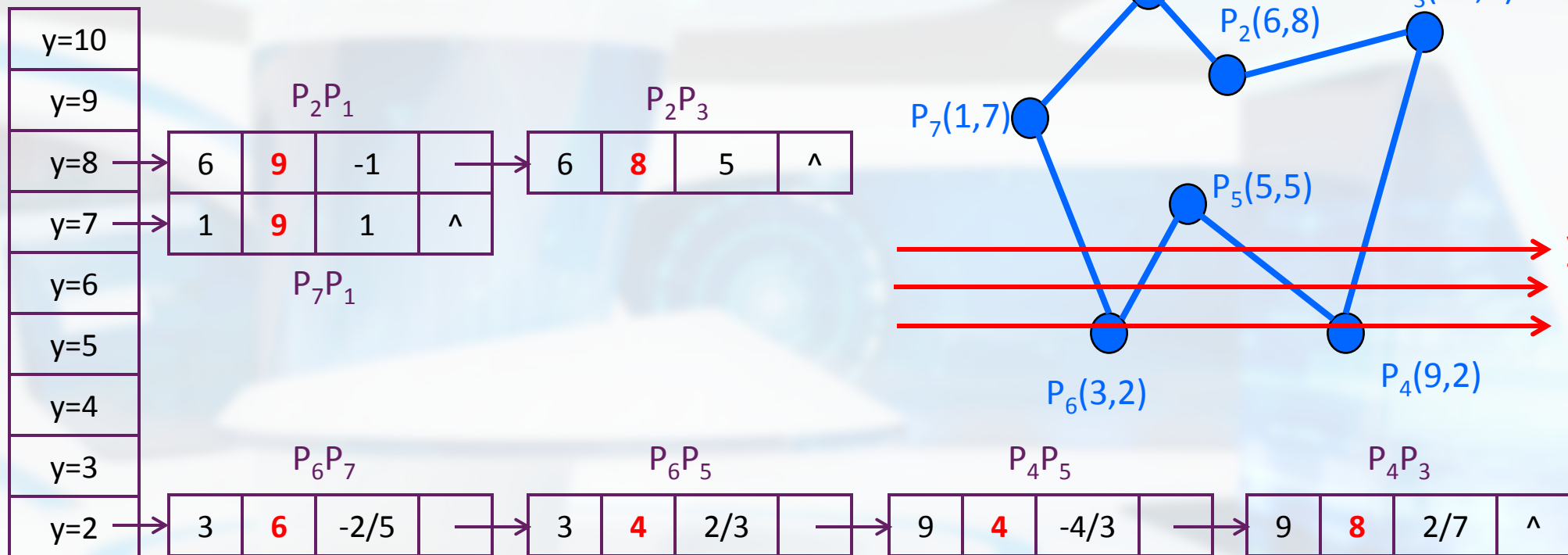
有效边表 (Active Edge Table, AET) : 增量计算

		P_6P_7				P_6P_5				P_4P_5				P_4P_3			
$y=4$	→	$3\frac{1}{5}$	6	$-2/5$	→	$4\frac{1}{3}$	3	$2/3$	→	$6\frac{1}{3}$	4	$-4/3$	→	$9\frac{4}{7}$	8	$2/7$	^

3

算法过程

边表 (Edge Table) :



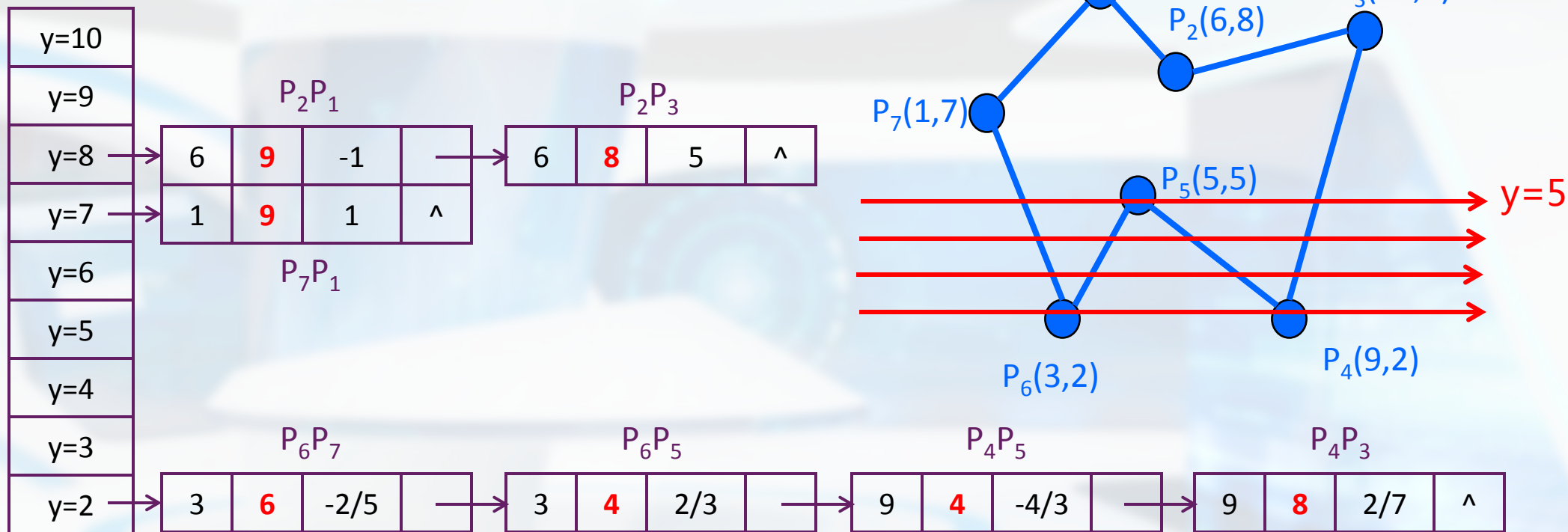
有效边表 (Active Edge Table, AET) : 删除之后的无效边



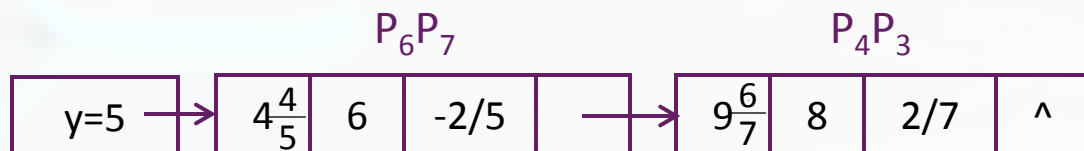
3

算法过程

边表 (Edge Table) :



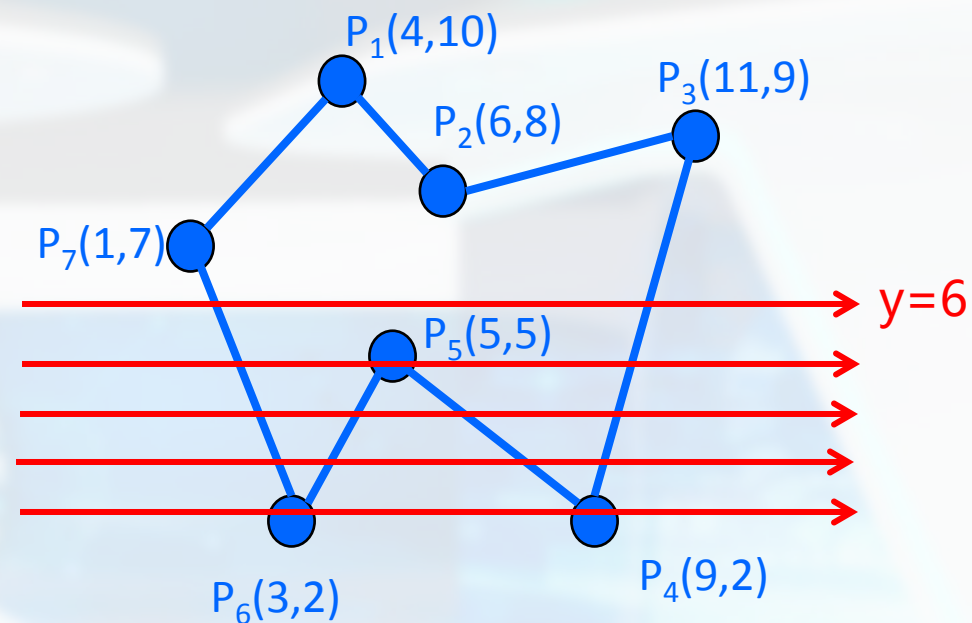
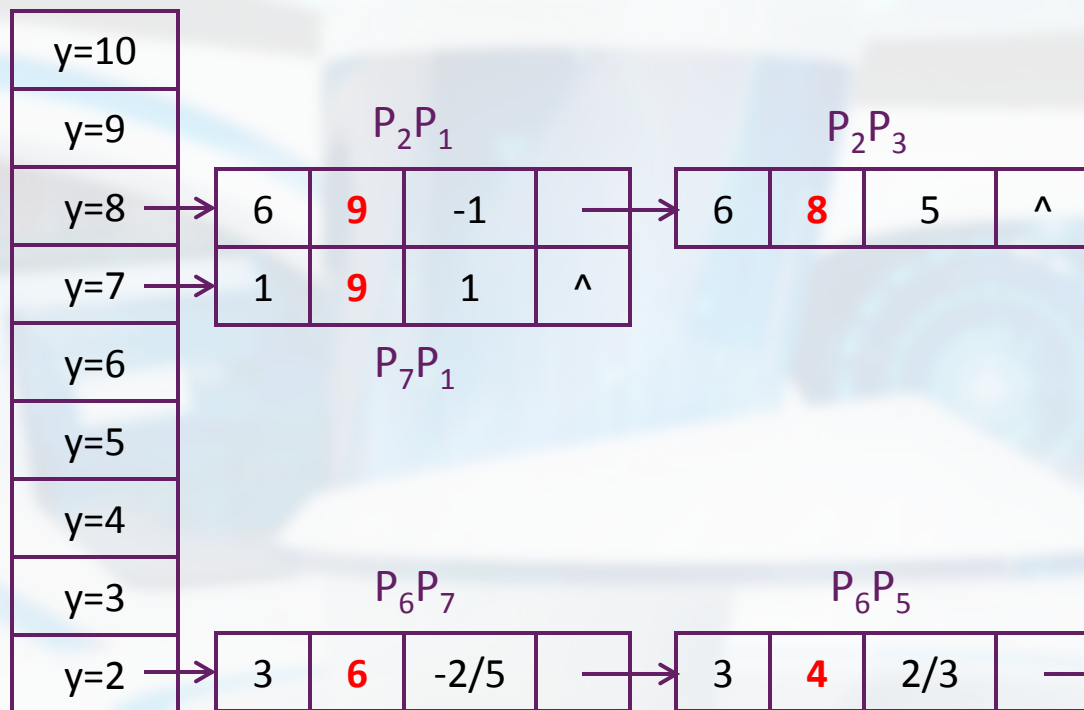
有效边表 (Active Edge Table, AET) : 增量计算



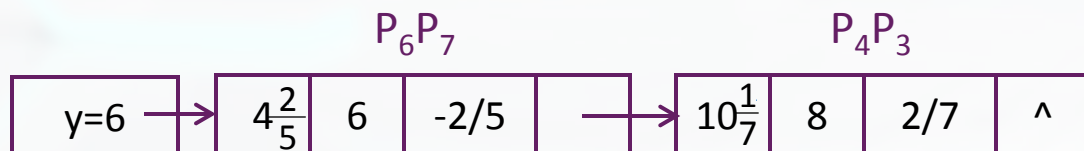
3

算法过程

边表 (Edge Table) :



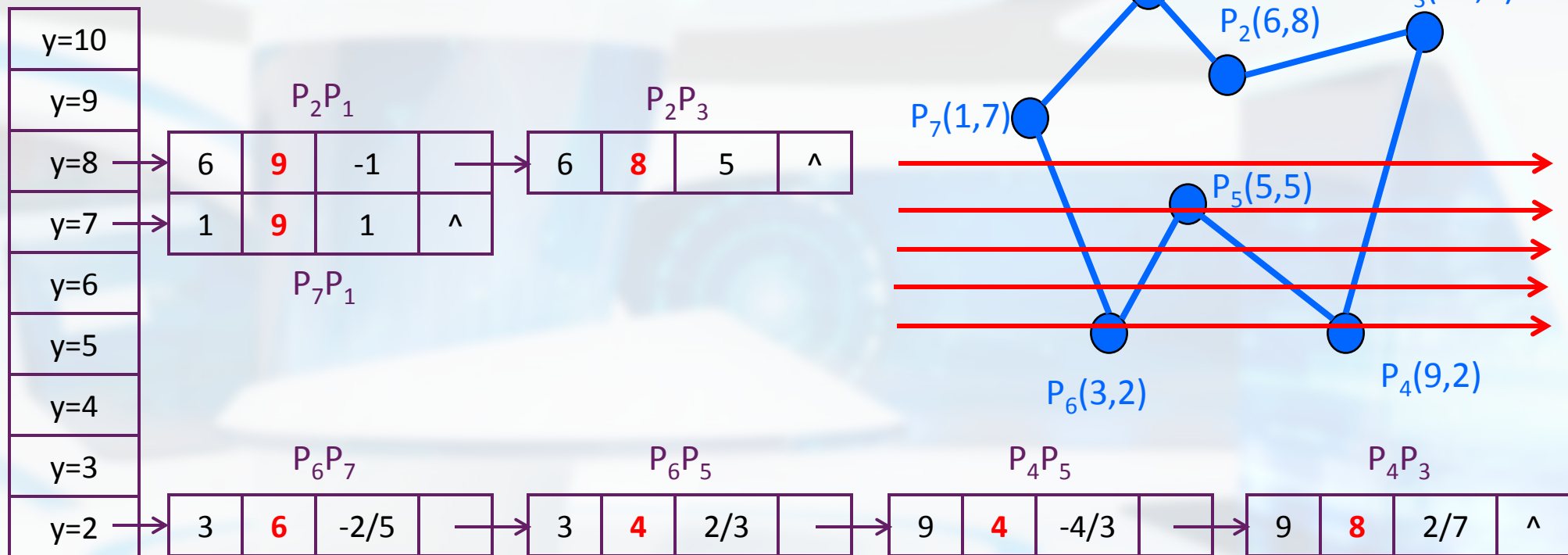
有效边表 (Active Edge Table, AET) : 增量计算



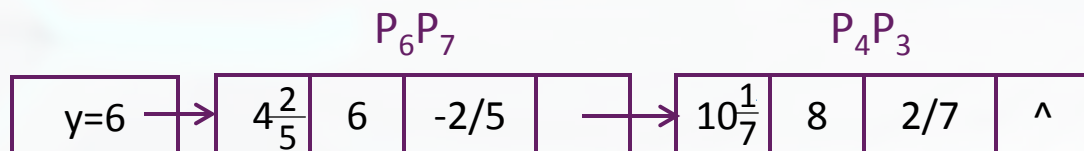
3

算法过程

边表 (Edge Table) :



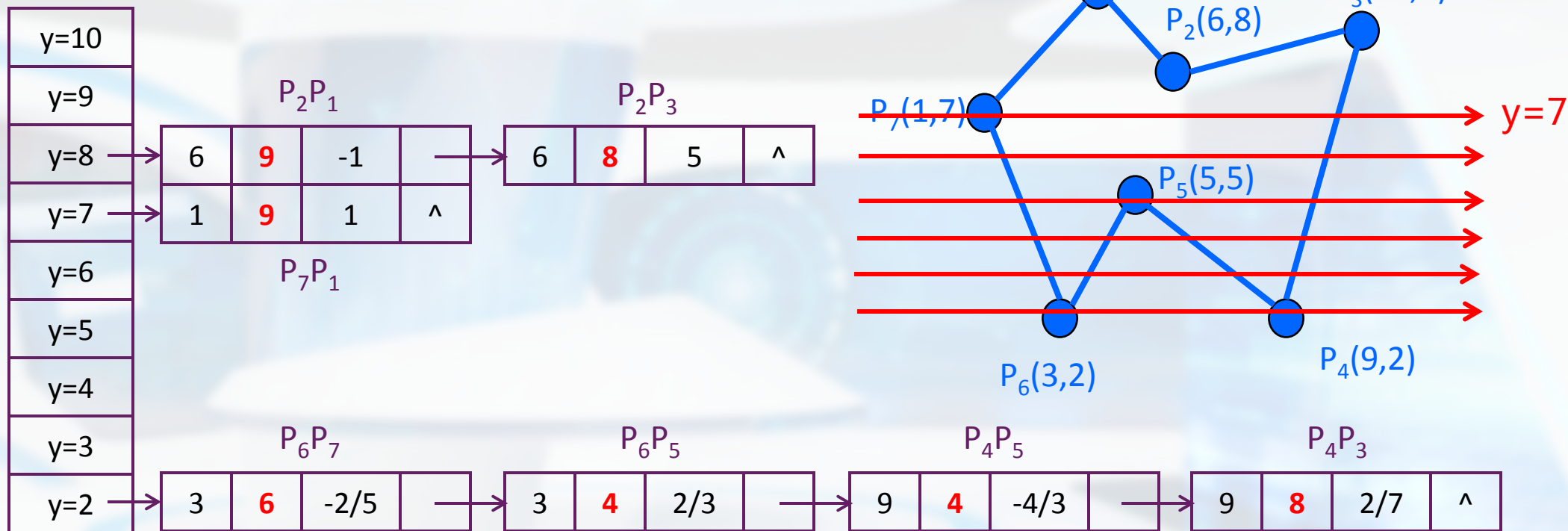
有效边表 (Active Edge Table, AET) : 删除之后的无效边



3

算法过程

边表 (Edge Table) :



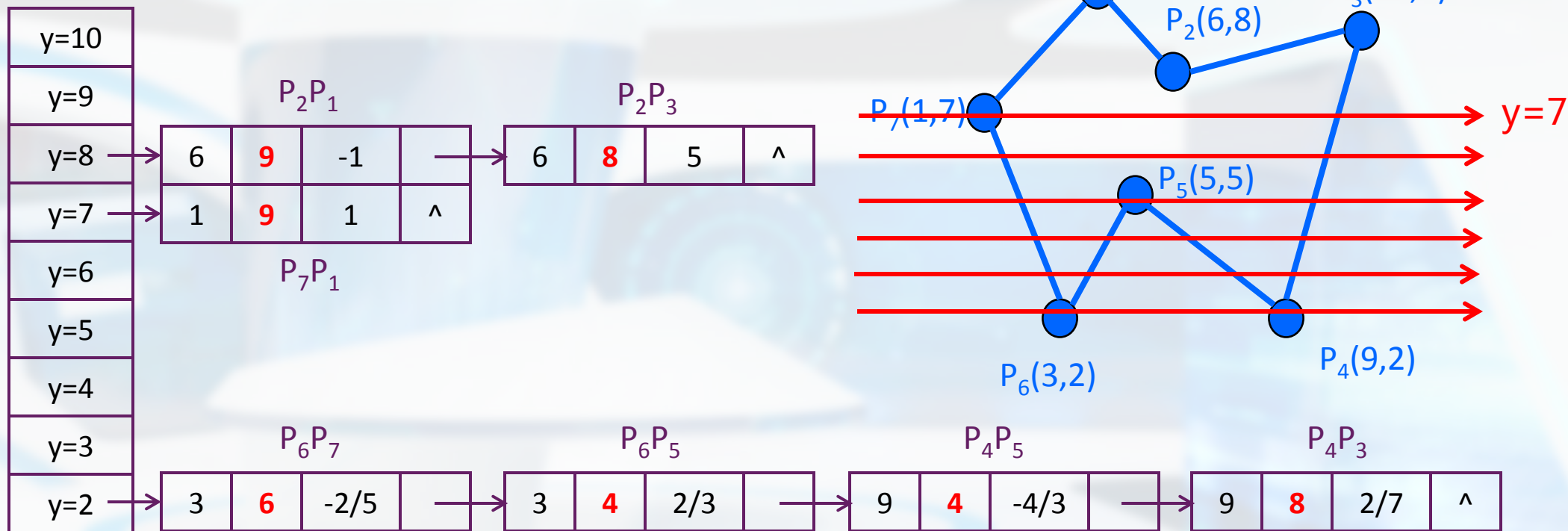
有效边表 (Active Edge Table, AET) : 增量计算



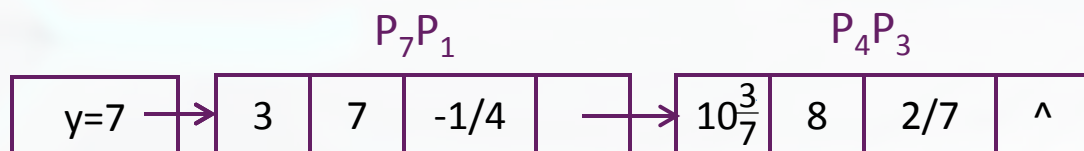
3

算法过程

边表 (Edge Table) :



有效边表 (Active Edge Table, AET) : 插入新边



3

算法过程

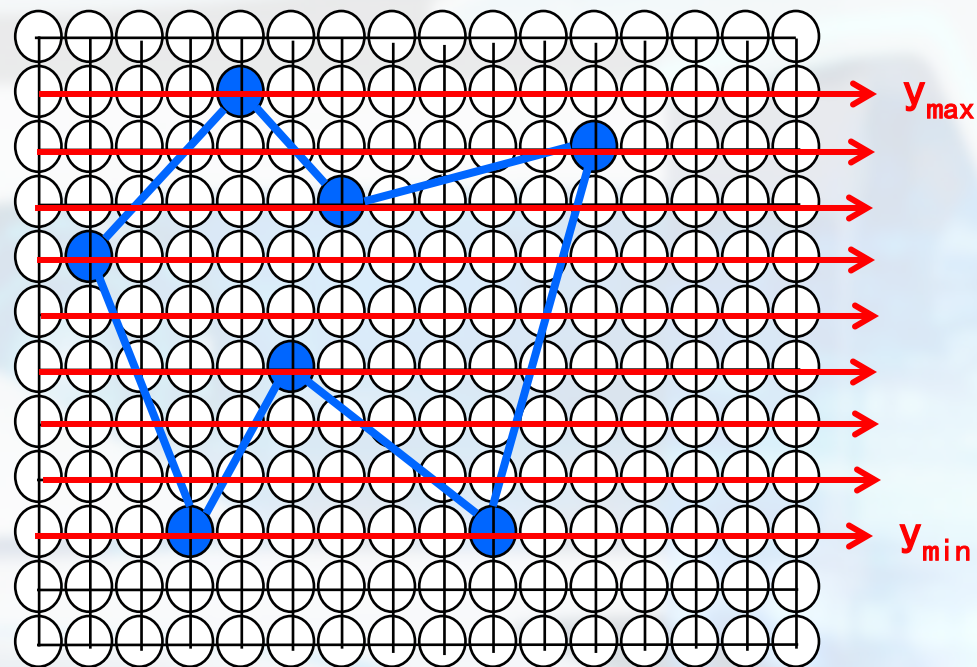
算法分析：

优点：

- ◆ 采用增量计算的方法进行交点计算
- ◆ 仅仅在新边加入时排序
(边数 \ll 扫描线数)

缺点：

桶表、链表的维护开销





谢谢

软件学院 万琳