

# 初次尝试——点和直线（下）

华中科技大学软件学院 万琳



## 提纲

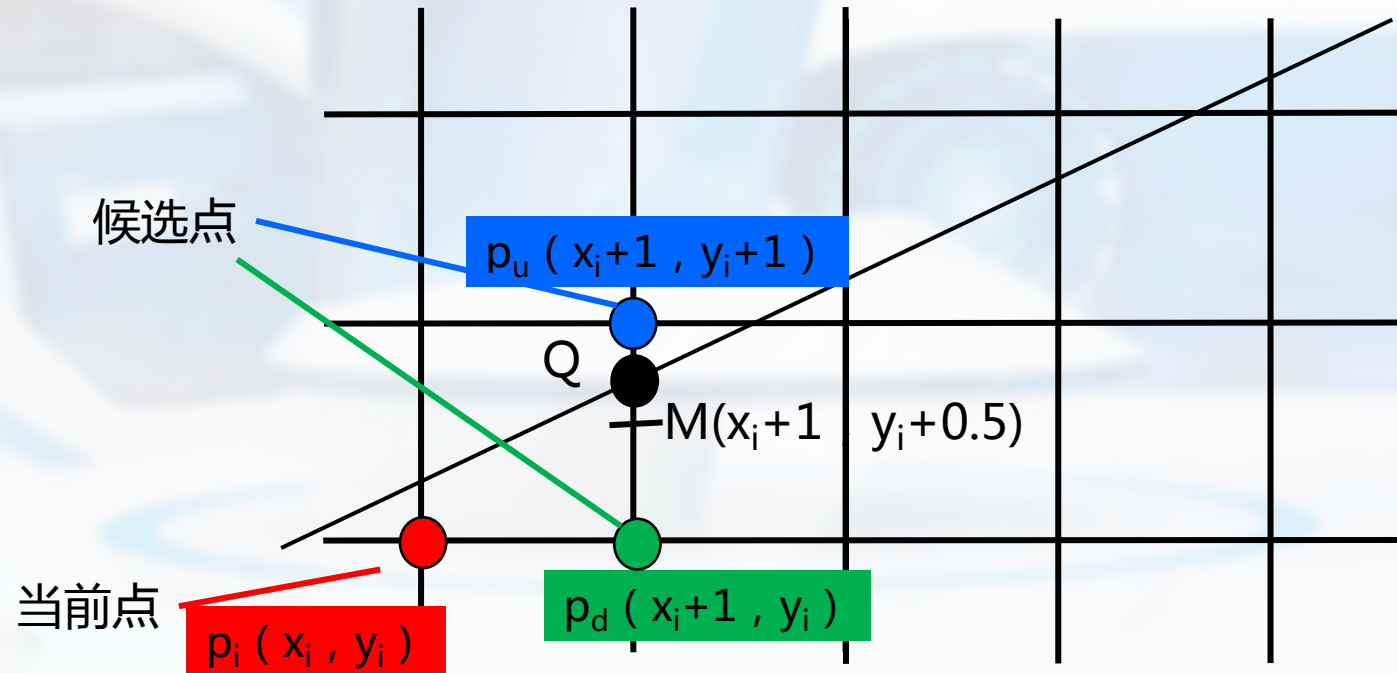
- ① 中点的Bresenham算法回顾
- ② 更直观的想法
- ③ 改进的Bresenham算法

1

## 中点Bresenham算法回顾

基本原理：

假定 $0 \leq k \leq 1$ ，x是最大位移方向

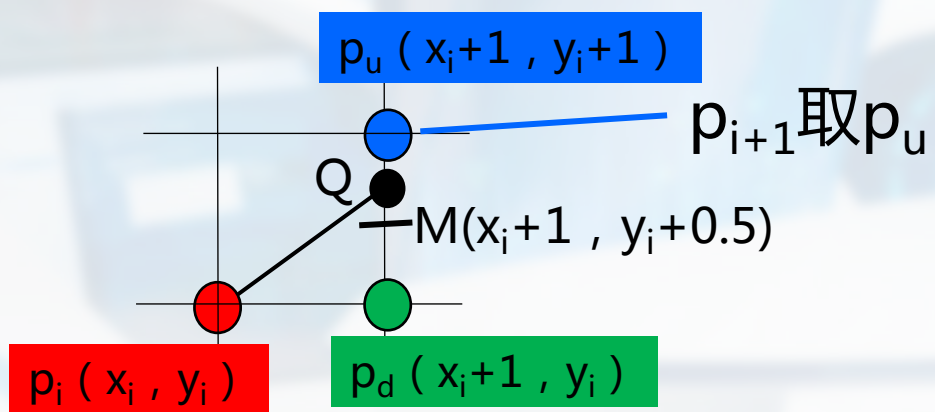


1

## 中点Bresenham算法回顾

基本原理：

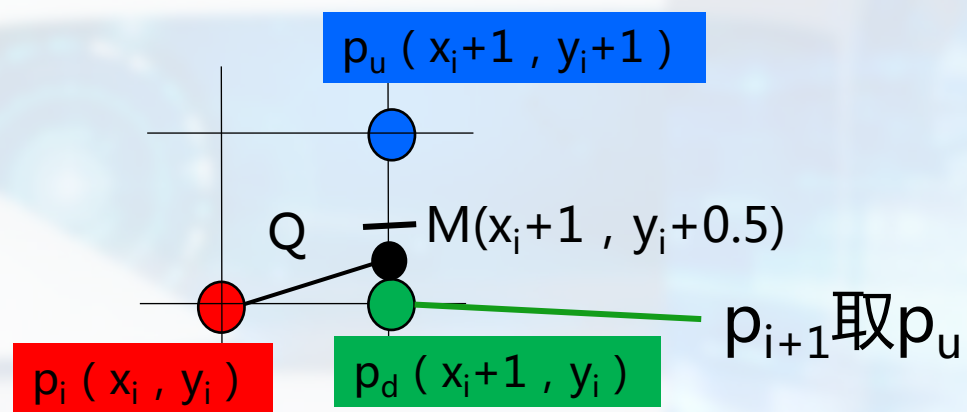
通过中点与直线的位置关系引入误差项 $d$ ，判断误差项的符号来选择候选点



M在Q的下方

$$d = F(M) < 0$$

取 $P_u$



M在Q的上方

$$d = F(M) \geq 0$$

取 $P_d$

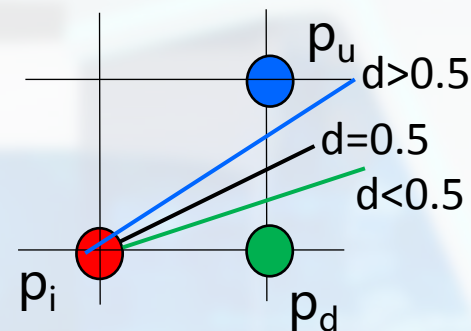
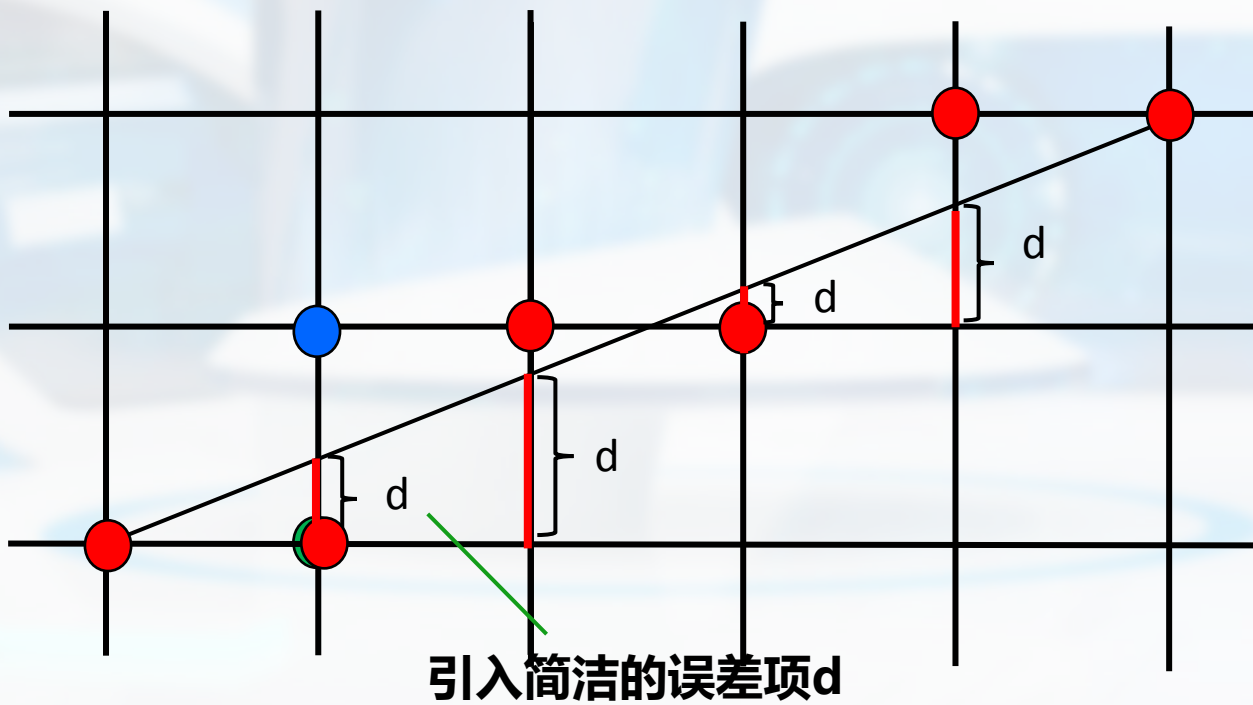


2

## 更直观的想法

基本原理：

假定 $0 \leq k \leq 1$ ， $x$ 是最大位移方向



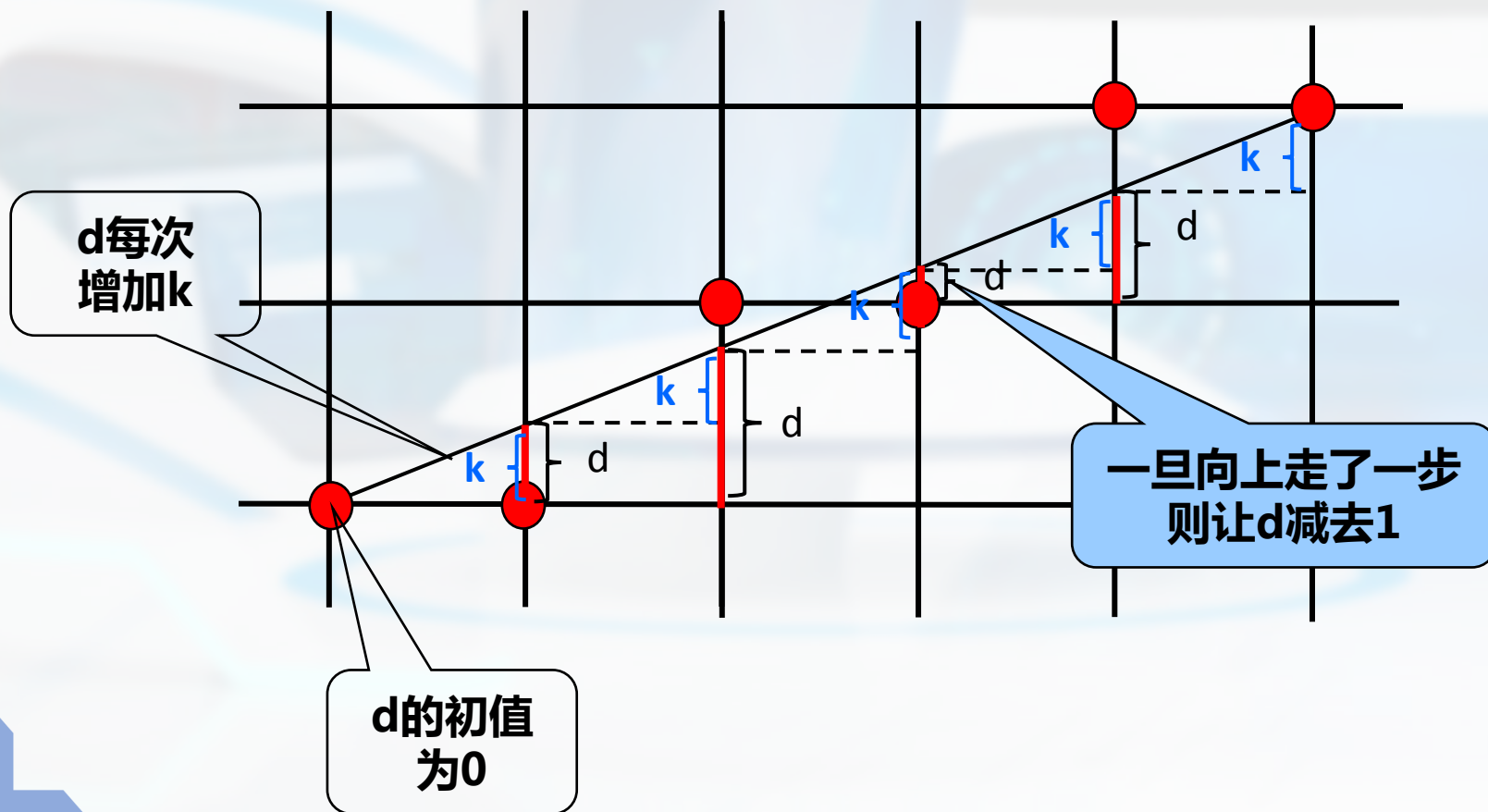
每次 $x_{i+1} = x_i + 1$

$$\begin{cases} d > 0.5 & y_{i+1} = y_i + 1 \\ d = 0.5 & y_{i+1} = y_i \\ d < 0.5 & y_{i+1} = y_i \end{cases}$$

2

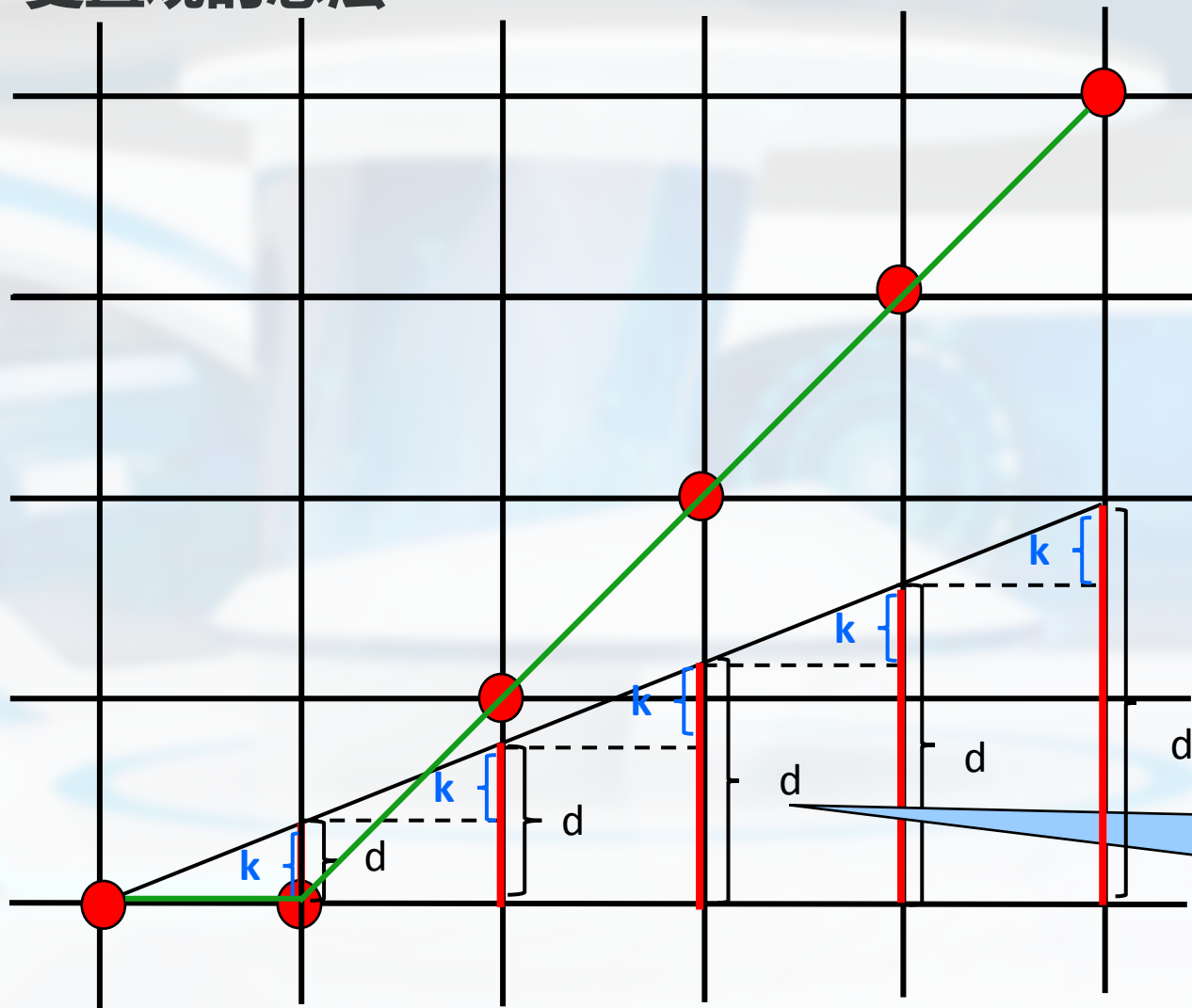
## 更直观的想法

寻找 $d$ 的变换规律



2

## 更直观的想法

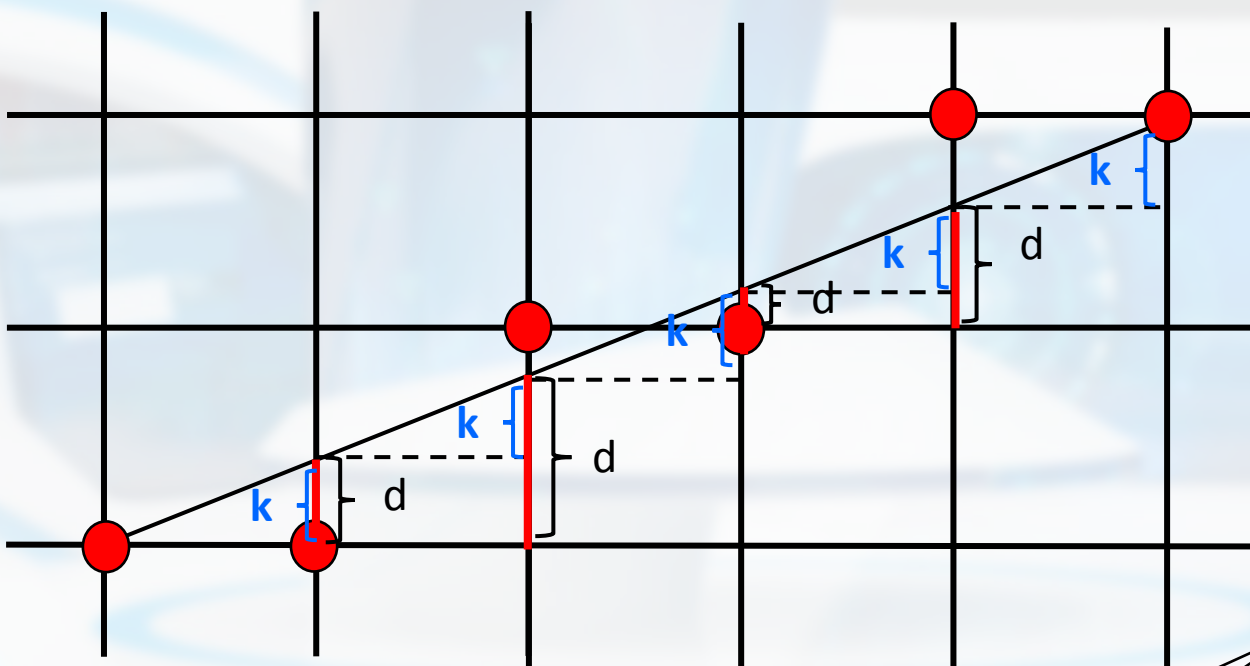


如果向上走了一步  
 $d$ 不减去1  
后果会如何？

3

## 改进的Bresenham算法

完备的算法



d的初值：

$$d_0 = 0$$

d的变换及如何取点：

$$d = d + k$$

浮点运算

$$x_{i+1} = x_i + 1$$

$$d > 0.5 \quad y_{i+1} = y_i + 1 \quad \text{此时 } d = d - 1$$

$$d \leq 0.5 \quad y_{i+1} = y_i$$

让人烦恼  
的比较



3

## 改进的Bresenham算法

改进一

d的初值：

$$d_0 = 0$$

$$\text{令 } e = d - 0.5$$

e的初值：

$$e_0 = -0.5$$

仍然有  
浮点数！

d的变换及如何取点：

$$d = d + k$$

$$x_{i+1} = x_i + 1$$

$$d > 0.5 \quad y_{i+1} = y_i + 1 \quad \text{此时 } d = d - 1$$

$$d \leq 0.5 \quad y_{i+1} = y_i$$



e的变换及如何取点：

$$e = e + k$$

$$x_{i+1} = x_i + 1$$

$$e > 0 \quad y_{i+1} = y_i + 1 \quad \text{此时 } e = e - 1$$

$$e \leq 0 \quad y_{i+1} = y_i$$

3

## 改进的Bresenham算法

改进二

e的初值：

$$e_0 = -0.5$$

$$e = e \times 2^{\Delta x}$$

e的初值：

$$e_0 = -\Delta x$$

e的变换及如何取点：

$$e = e + k$$

$$x_{i+1} = x_i + 1$$

$$e > 0 \quad y_{i+1} = y_i + 1 \quad \text{此时 } e = e - 1$$

$$e \leq 0 \quad y_{i+1} = y_i$$



e的变换及如何取点：

$$e = e + 2^{\Delta y}$$

$$x_{i+1} = x_i + 1$$

$$e > 0 \quad y_{i+1} = y_i + 1 \quad \text{此时 } e = e - 2^{\Delta x}$$

$$e \leq 0 \quad y_{i+1} = y_i$$

消除了  
浮点数！

3

## 改进的Bresenham算法

在 $0 \leq k \leq 1$ 情况下改进的Bresenham算法：

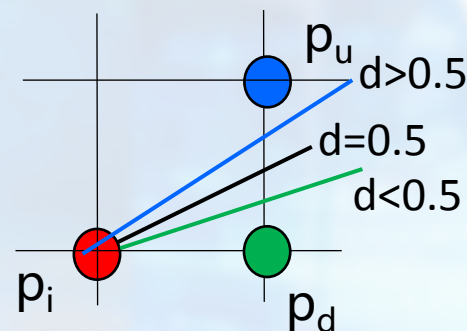
- (1) 输入直线的两端点 $P_0(x_0, y_0)$ 和 $P_1(x_1, y_1)$ 。
- (2) 计算初始值 $\Delta x$ 、 $\Delta y$ 、 $e = -\Delta x$ 、 $x = x_0$ 、 $y = y_0$ 。
- (3) 绘制点 $(x, y)$ 。
- (4)  $e$ 更新为 $e + 2\Delta y$

判断 $e$ 的符号

若 $e > 0$ ，则 $(x, y)$ 更新为 $(x+1, y+1)$ ，同时将 $e$ 更新为 $e - 2\Delta x$ ；

否则 $(x, y)$ 更新为 $(x+1, y)$ 。

- (5) 当直线没有画完时，重复步骤3和4。否则结束。

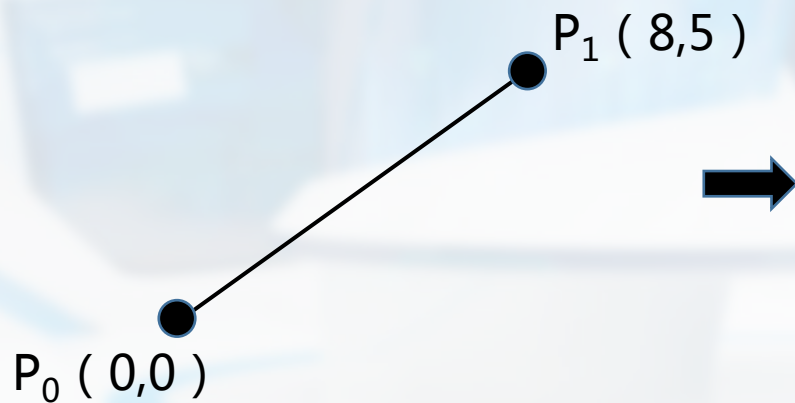


3

## 改进的Bresenham算法

输入：直线两个端点的坐标 $P_0(0,0)$ 和 $P_1(8,5)$

输出：最佳逼近这条直线的像素点集



起点： $P_0(x_0, y_0)$ 为  $(0,0)$   
终点： $P_1(x_1, y_1)$ 为  $(8,5)$

$$e_0 = -\Delta x = -8$$

$$e \text{ 增量为 } 2\Delta y = 2 \times 5 = 10$$

如果  $e > 0$  是  $y$  向上走一步

$$e = e - 2\Delta x$$

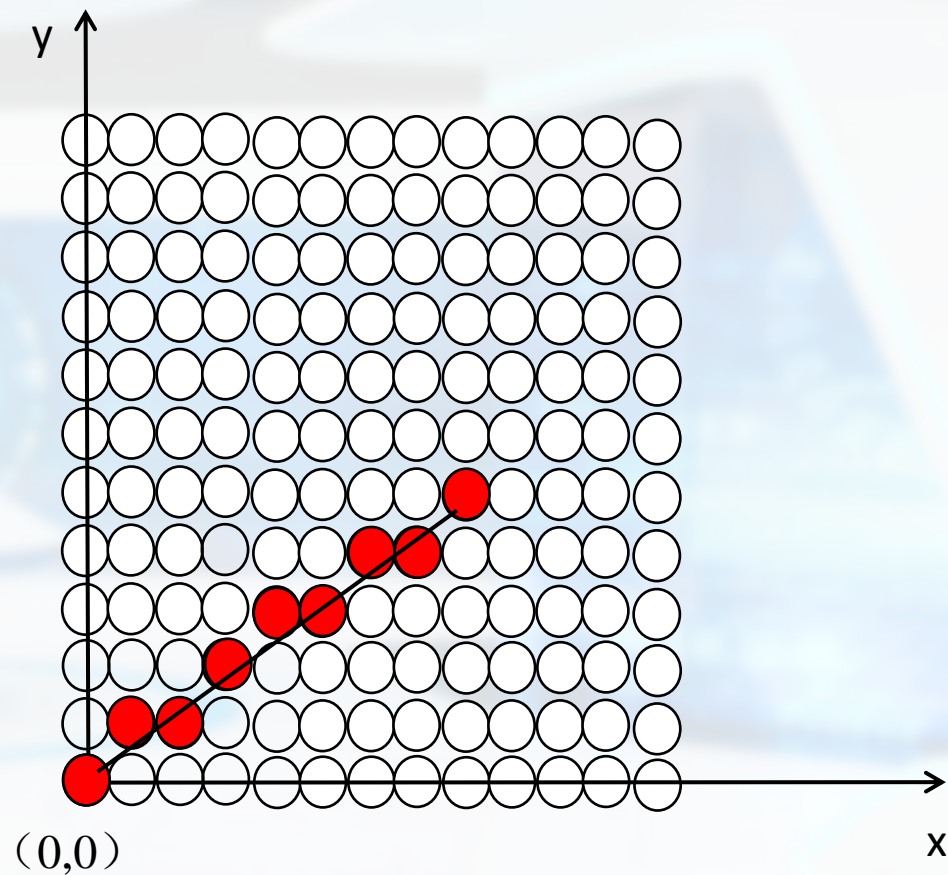
$$\text{其中 } -2\Delta x = -16$$



### 3

## 改进的Bresenham算法

x	y	e	$e+2\Delta y$
0	0	-8	2
1	1	-14	-4
2	1	-4	6
3	2	-10	0
4	2	0	10
5	3	-6	4
6	4	-12	-2
7	4	-2	8
8	5	-8	2
9	6	-14	





# 谢谢

软件学院 万琳