初次尝试——点和直线（中）

华中科技大学软件学院　万琳

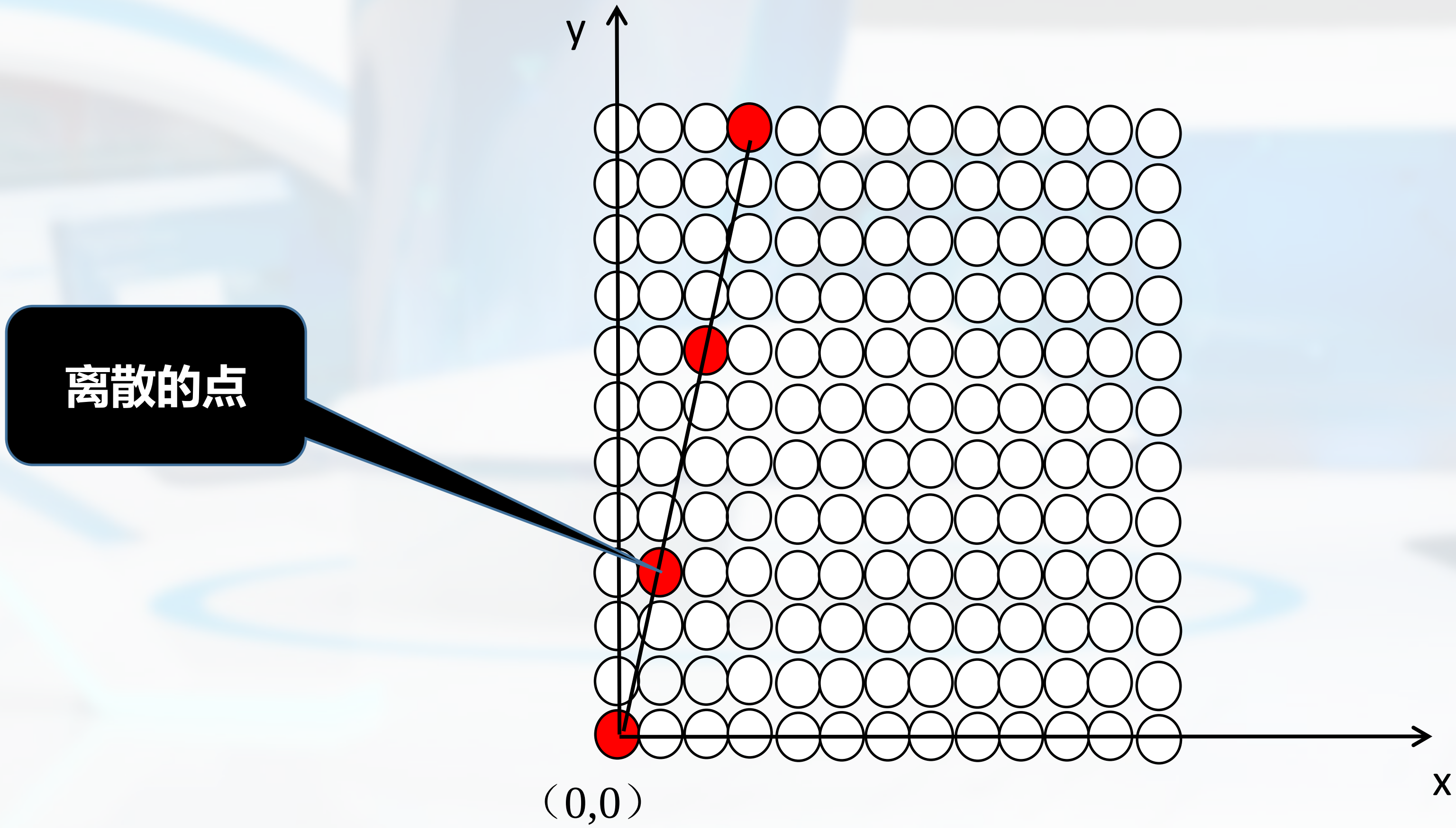
提纲

1. DDA算法回顾
2. 中点的Bresenham算法

1

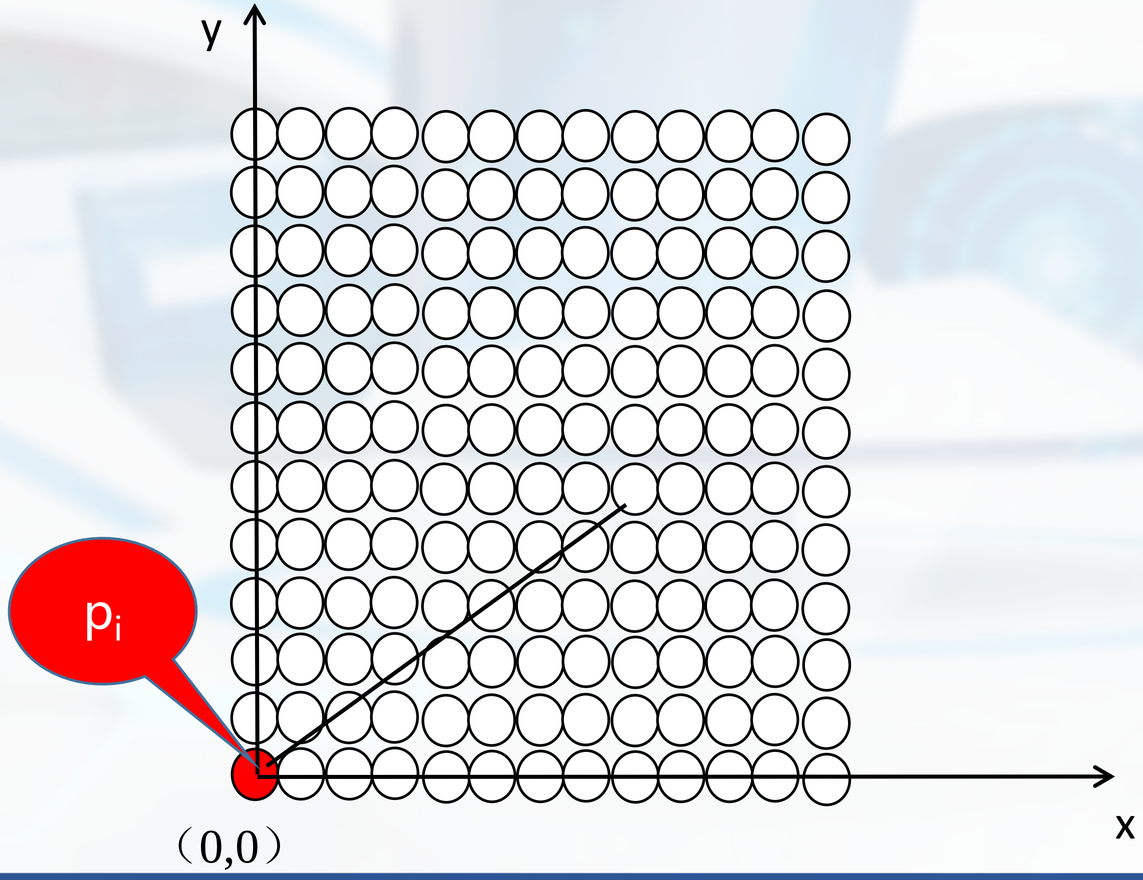
DDA算法回顾

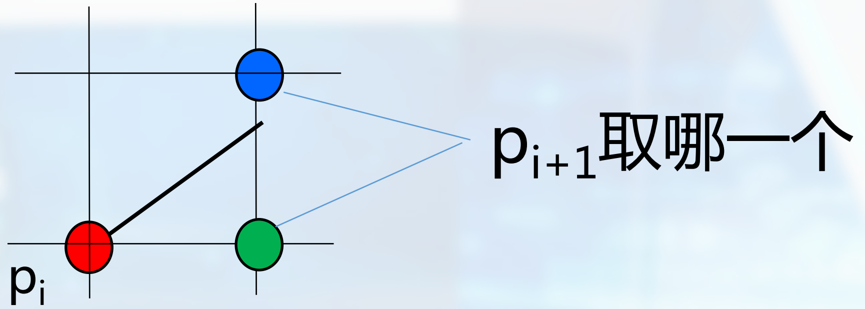
最大位移方向的重要性



1 DDA算法回顾

有了当前点，如何取下一点？



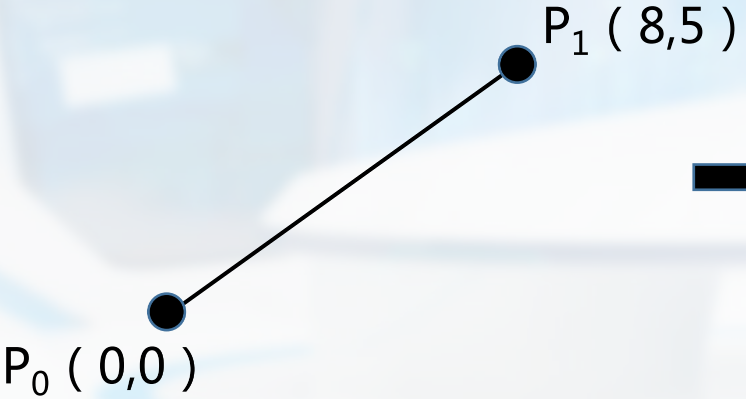


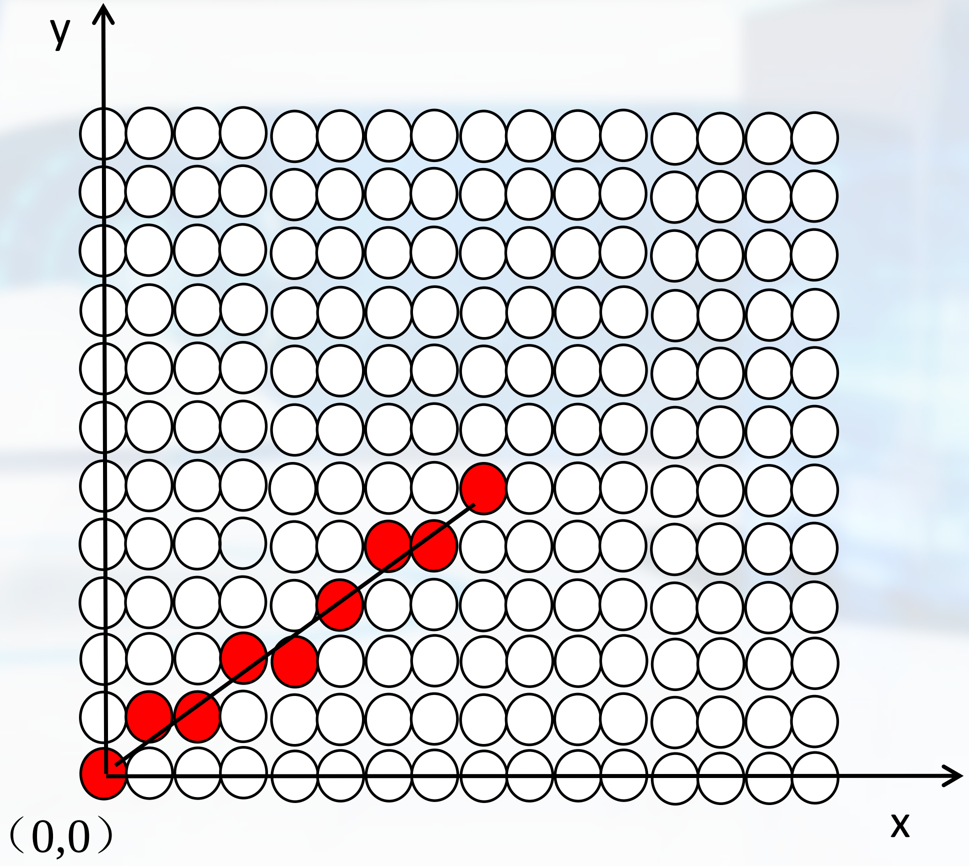
误差项？

2 中点Bresenham算法

输入：直线两个端点的坐标$P_0(x_0,y_0)$和$P_1(x_1,y_1)$

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）

2 中点Bresenham算法

通过给定直线的两端点坐标$P_0(x_0,y_0)$和$P_1(x_1,y_1)$，可得到直线斜截式表示：

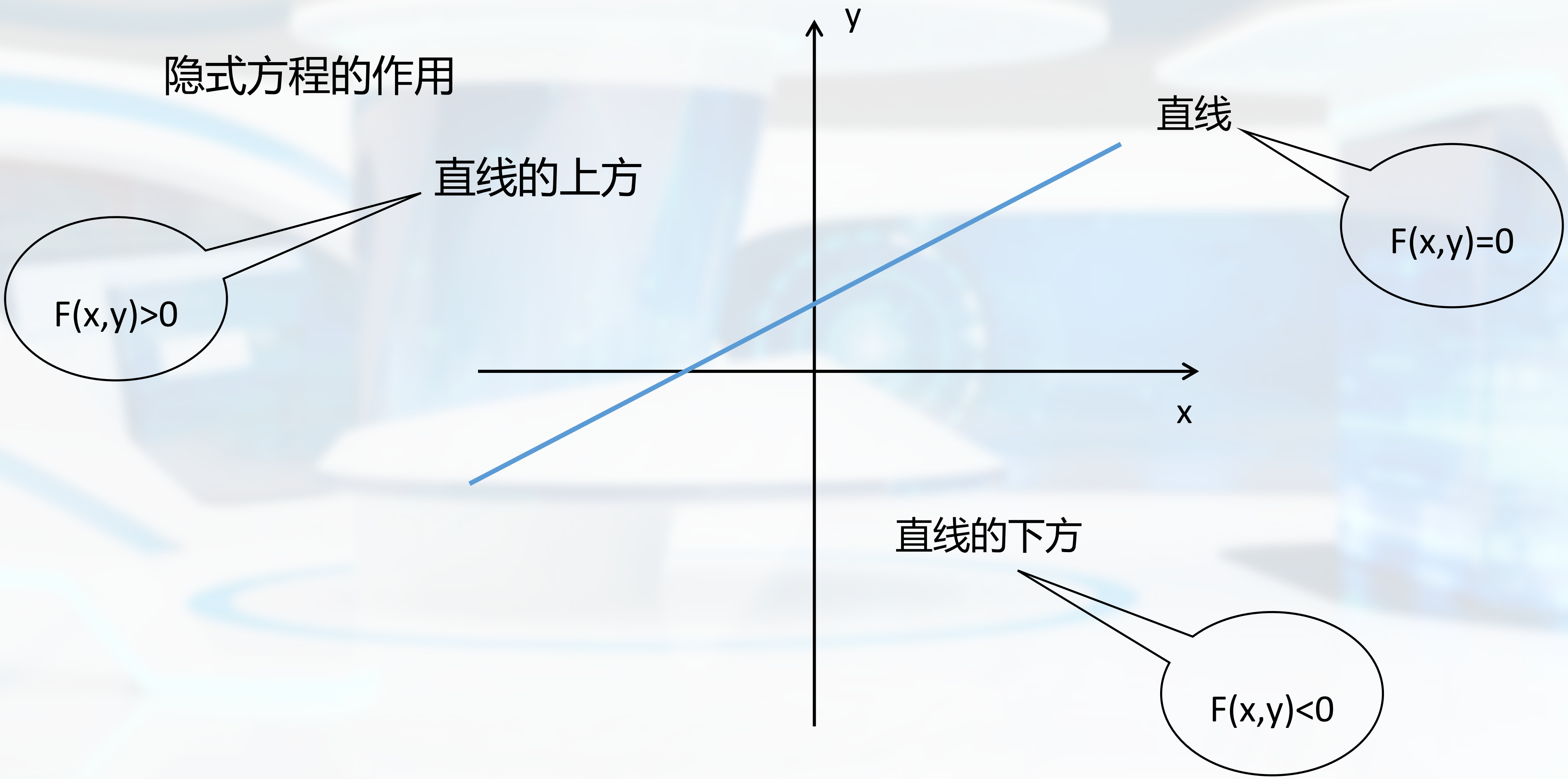
$$y=kx+b$$

我们还可以写出它的**隐式方程**：

$$F(x,y)=y-kx-b=0 \quad \text{其中} k=\frac{\Delta y}{\Delta x}=\frac{y_1-y_0}{x_1-x_0}$$

中点Bresenham算法

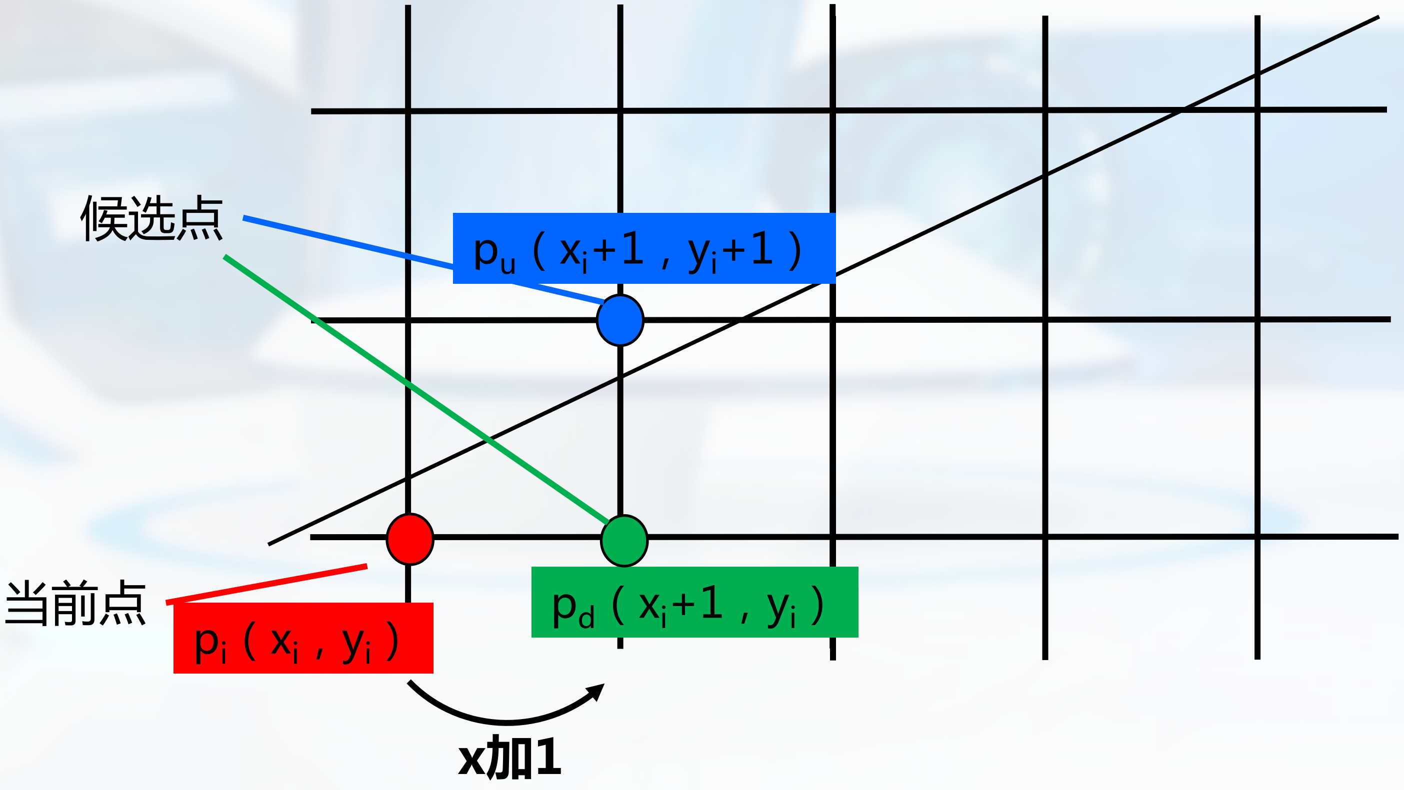
隐式方程的作用



2 中点Bresenham算法

基本原理：

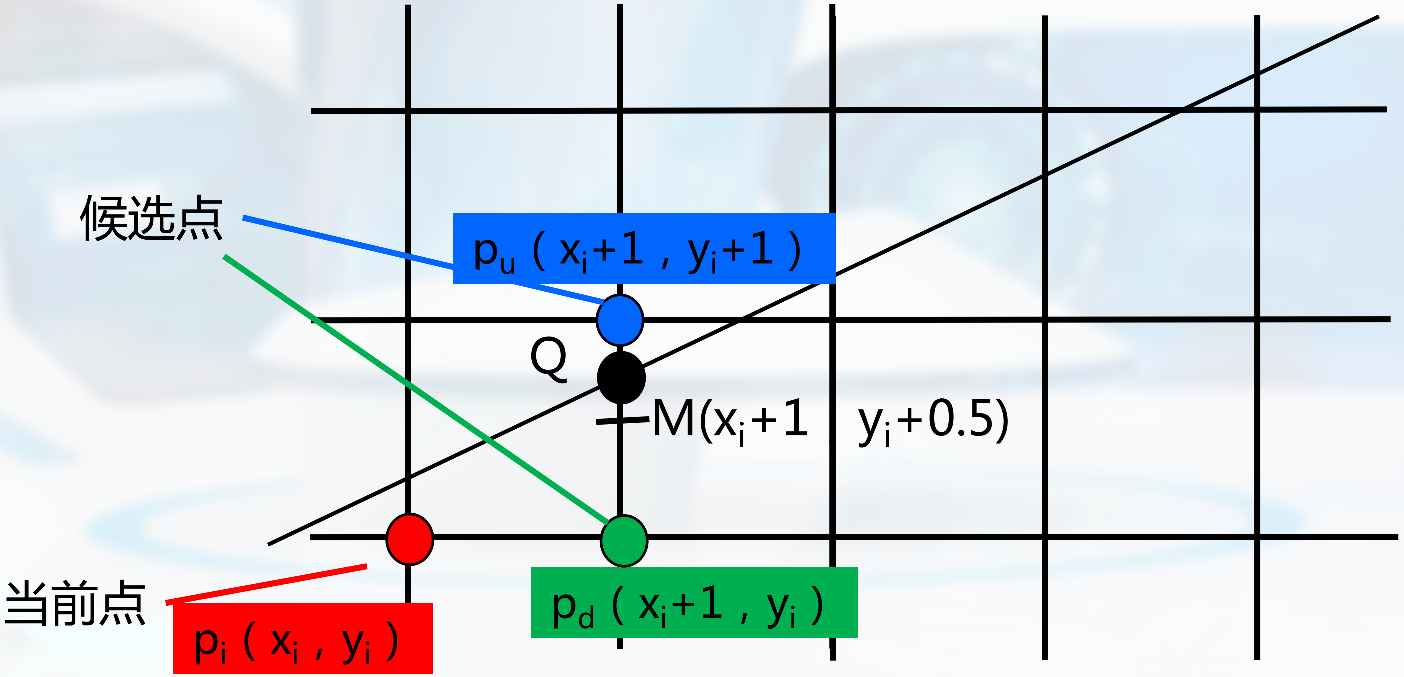
假定0≤k≤1，x是最大位移方向



2 中点Bresenham算法

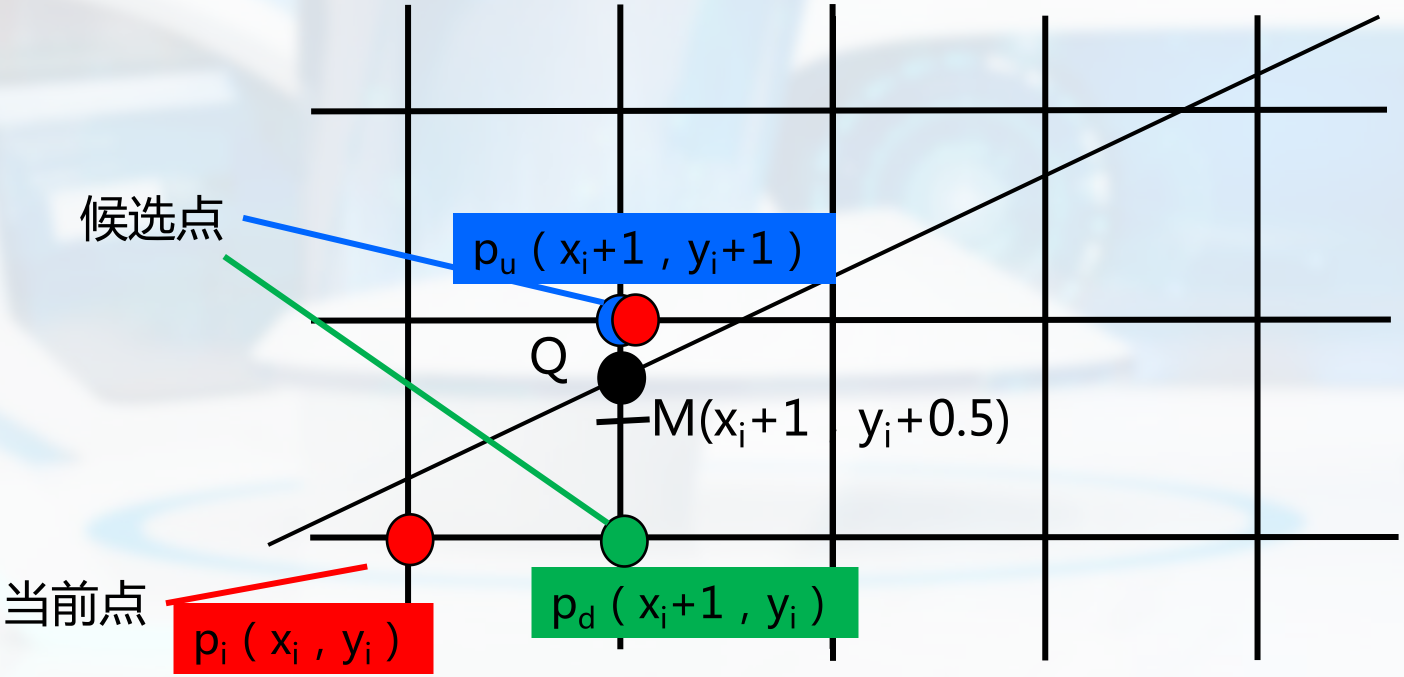
基本原理：

假定0≤k≤1，x是最大位移方向



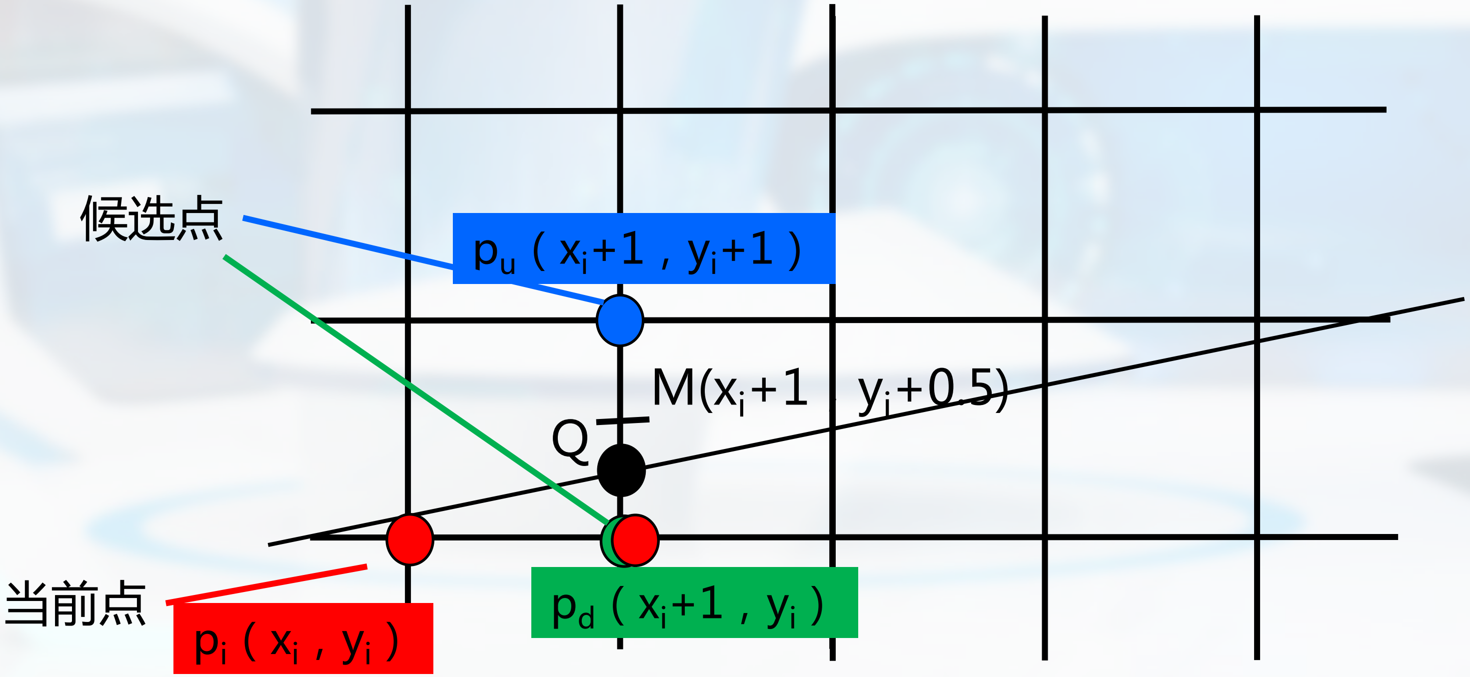
2 中点Bresenham算法

M在Q的下方，取P_u



2 中点Bresenham算法

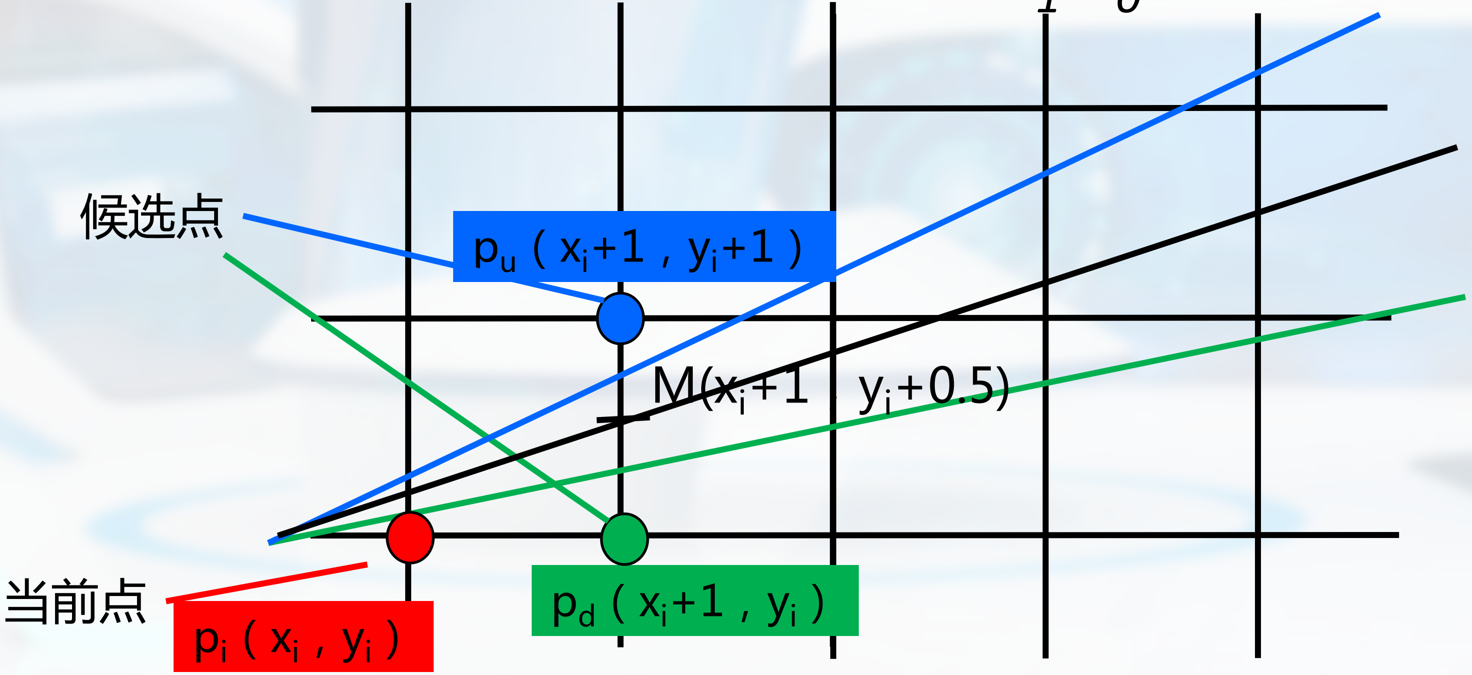
M在Q的上方，取P_d



2 中点Bresenham算法

直线隐式方程起作用

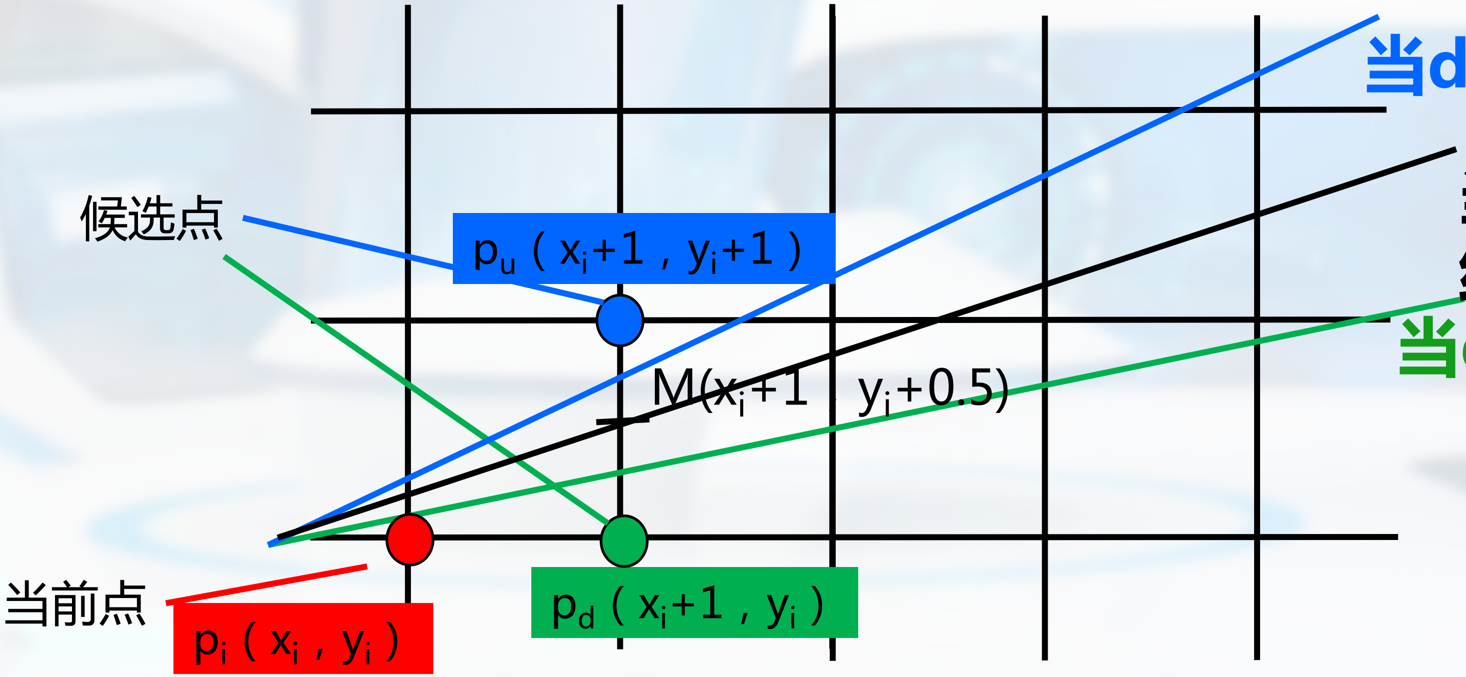
$$F(x,y)=y-kx-b=0 \quad \text{其中} k=\frac{\Delta y}{\Delta x}=\frac{y_1-y_0}{x_1-x_0}$$



2 中点Bresenham算法

构造判别式

$$d=F(x_M,y_M)=F(x_i+1,y_i+0.5)=y_i+0.5-k(x_i+1)-b$$



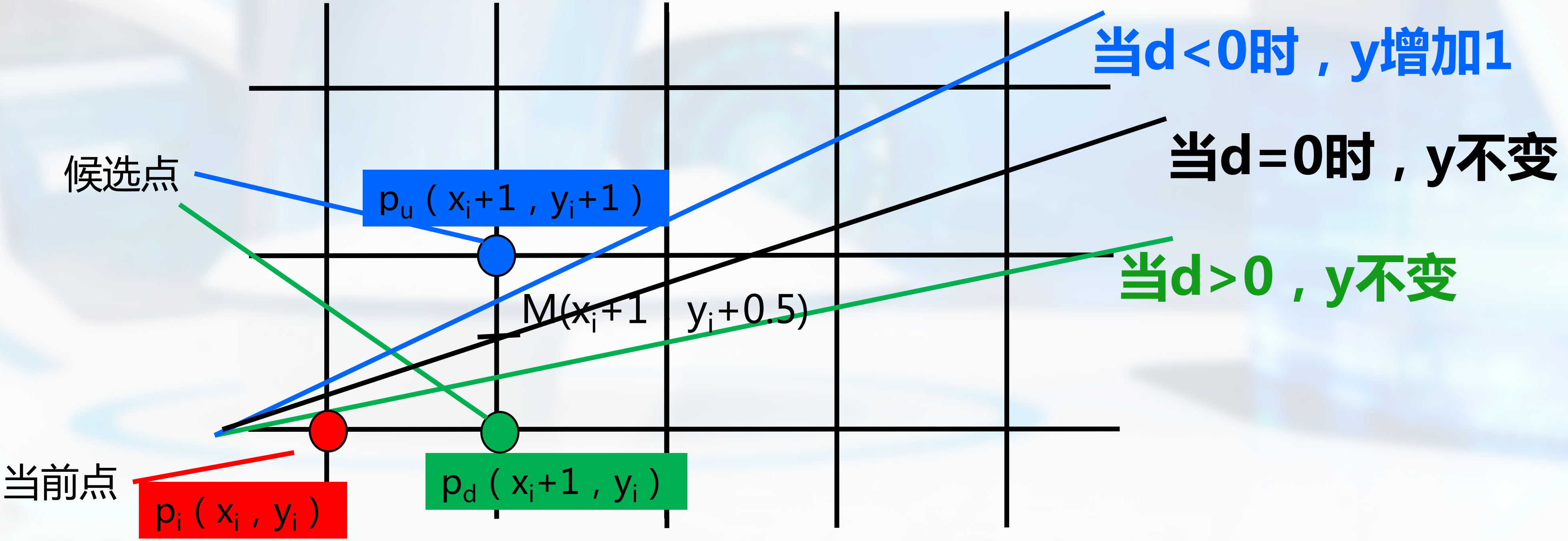
当d<0时，取右上方P_u

当d=0时，任取一个，约定取P_d

当d>0，取正右方的P_d

2 中点Bresenham算法

判别后如何取点？



2 中点Bresenham算法

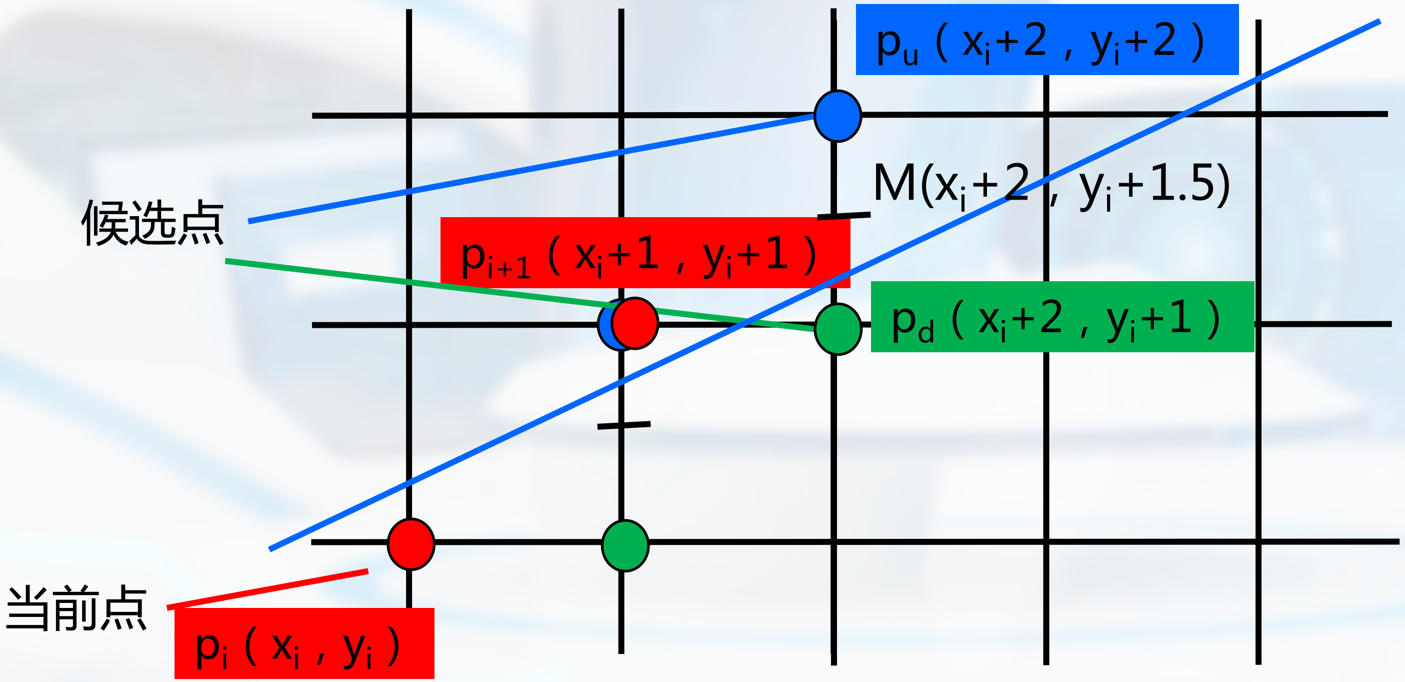
比DDA好吗？

$$d = F(x_M, y_M) = F(x_i + 1, y_i + 0.5) = y_i + 0.5 - k(x_i + 1) - b$$

复杂的
计算过程

2 中点Bresenham算法

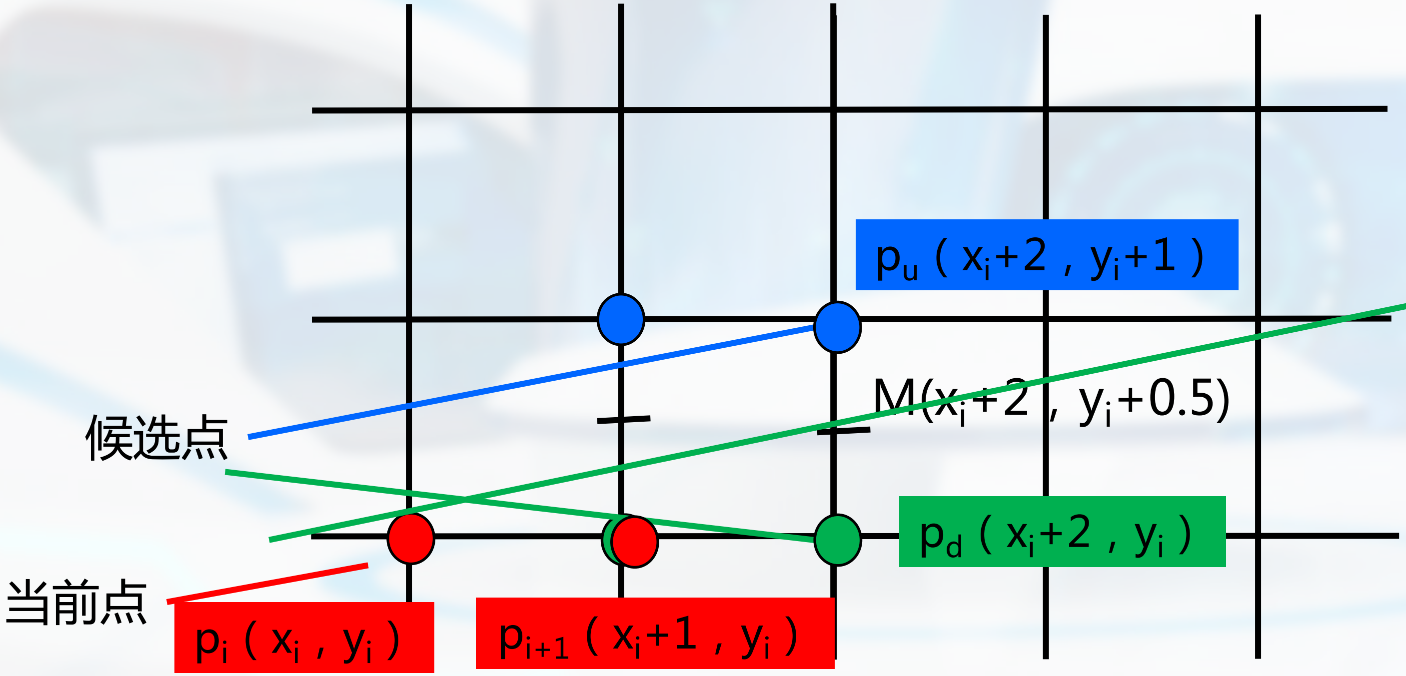
误差项递推：d < 0



$$
\begin{aligned}
d &= F(x_i+2, y_i+1.5) \\
&= y_i + 1.5 - k(x_i+2) - b \\
&= y_i + 1.5 - k(x_i+1) - b - k \\
&= d + 1 - k
\end{aligned}
$$

2 中点Bresenham算法

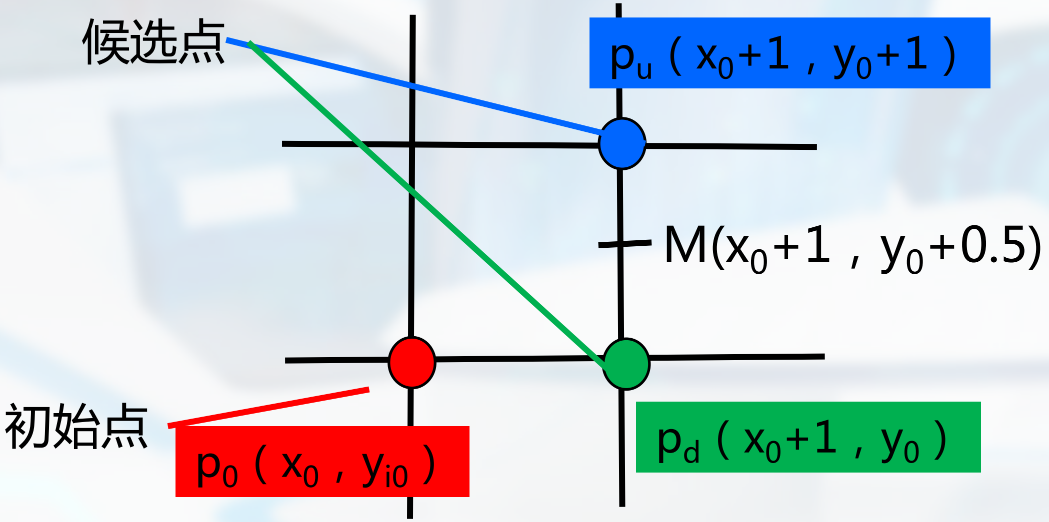
误差项递推：d≥0



$$
\begin{aligned}
d &= F(x_i+2, y_i+0.5) \\
&= y_i+0.5-k(x_i+2)-b \\
&= y_i+0.5-k(x_i+1)-b-k \\
&= d\boxed{-k}
\end{aligned}
$$

2 中点Bresenham算法

d的初值



$$
\begin{aligned}
d_0 &= F(x_0+1, y_0+0.5) \\
&= y_0+0.5-k(x_0+1)-b \\
&= y_0-kx_0-b-k+0.5 \\
&= 0.5-k
\end{aligned}
$$

2 中点Bresenham算法

比DDA好吗？

d的计算变成了增量计算

可是……

d的初值

$$
\begin{aligned}
d_0 &= F(x_0+1, y_0+0.5) \\
&= y_0+0.5-k(x_0+1)-b \\
&= y_0-kx_0-b-k+0.5 \\
&= 0.5-k
\end{aligned}
$$

浮点数

d < 0

$$
\begin{aligned}
d &= F(x_i+2, y_i+1.5) \\
&= y_i+1.5-k(x_i+2)-b \\
&= y_i+1.5-k(x_i+1)-b-k \\
&= d+1-k
\end{aligned}
$$

d ≥ 0

$$
\begin{aligned}
d &= F(x_i+2, y_i+0.5) \\
&= y_i+0.5-k(x_i+2)-b \\
&= y_i+0.5-k(x_i+1)-b-k \\
&= d-k
\end{aligned}
$$

2 中点Bresenham算法

0.5 → ×2

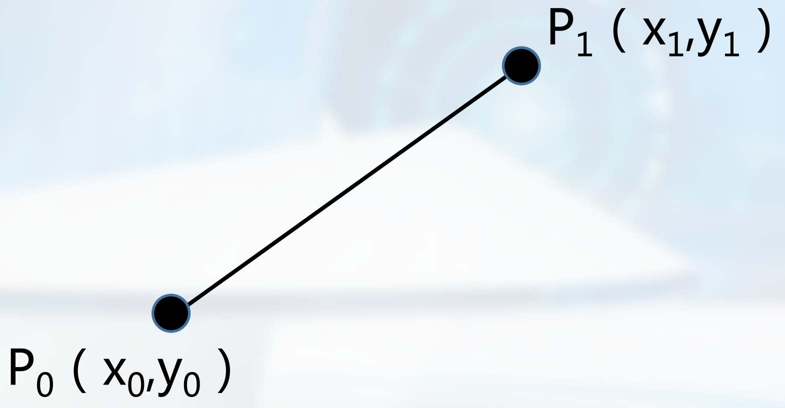
k=△y/△x → ×△x

→ 将d放大2△x倍

2 中点Bresenham算法

在$0 \le k \le 1$情况下的整数的中点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

中点Bresenham算法

在$0 \le k \le 1$情况下的整数的中点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Δx、Δy、$d=\Delta x-2\Delta y$、$x=x_0$、$y=y_0$。

中点Bresenham算法

在0≤k≤1情况下的整数的中点Bresenham算法：

（3）绘制点(x,y)。

判断d的符号

若d<0

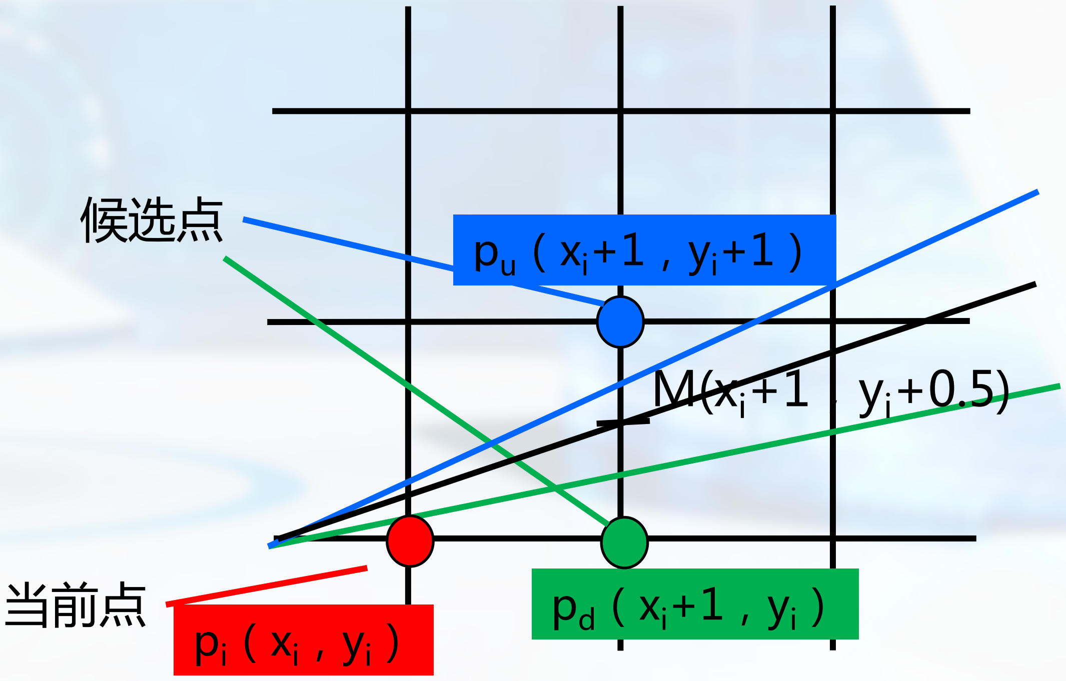
(x,y)更新为(x+1,y+1)

d更新为d+2△x-2△y

否则

(x,y)更新为(x+1,y)

d更新为d-2△y



2 中点Bresenham算法

在$0 \le k \le 1$情况下的整数的中点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$\triangle x$、$\triangle y$、$d=\triangle x-2\triangle y$、$x=x_0$、$y=y_0$。
（3）绘制点(x,y)。

判断d的符号

若$d<0$，则(x,y)更新为(x+1,y+1)，d更新为$d+2\triangle x-2\triangle y$；

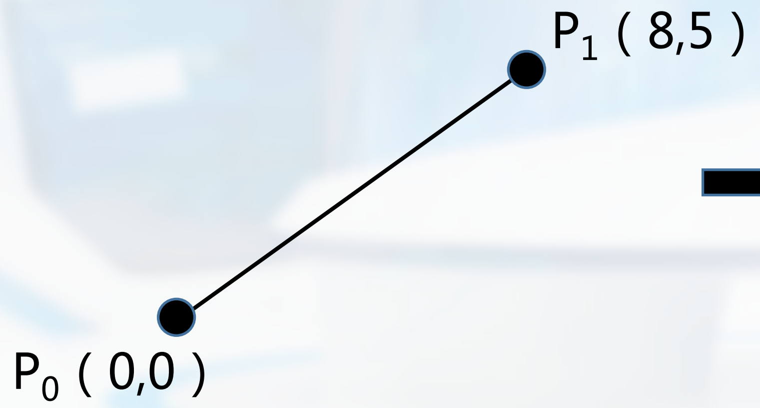
否则(x,y)更新为(x+1,y), d更新为$d-2\triangle y$。

（4）当直线没有画完时，重复步骤3，否则结束。

2 中点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



$\because$ k=5/8

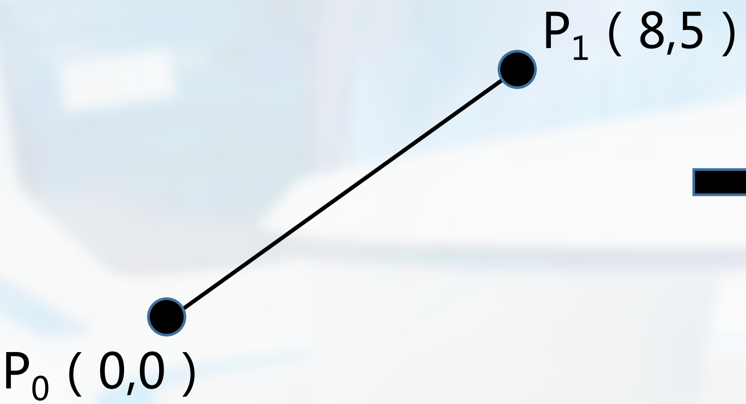
$\therefore$ 最大位移方向为x方向

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

2 中点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



$d_0=\triangle x-2\triangle y=8-2\times5=-2$

$d<0$往右上方时增量为

$2\triangle x-2\triangle y=2\times8-2\times5=6$

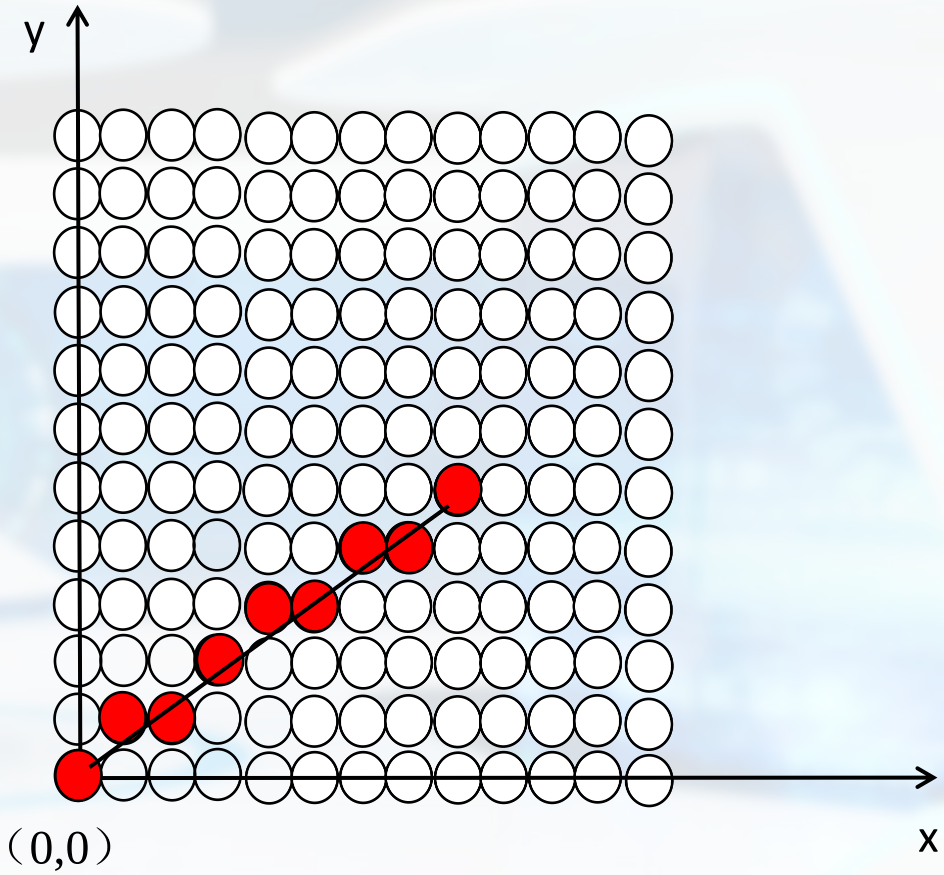
$d\geq0$ 往正右方向的增量为

$-2\triangle y=-2\times5=-10$

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

2 中点Bresenham算法

x	y	d
0	0	-2
1	1	4
2	1	-6
3	2	0
4	2	-10
x	y	d
5	3	-4
6	4	2
7	4	-8
8	5	-2



谢谢

软件学院　万琳