

初次尝试——点和直线

华中科技大学软件学院 万琳



提纲

① 扫描转换的概念

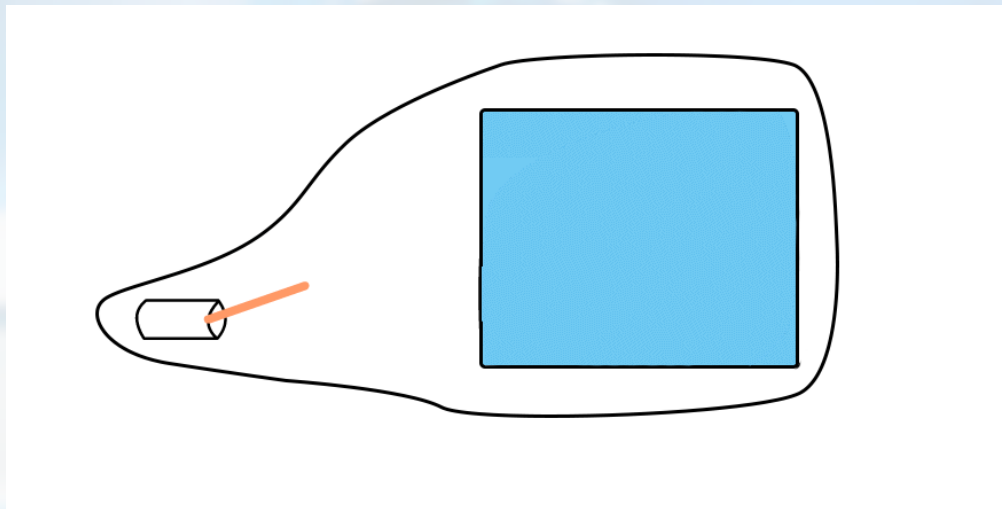
② 点

③ 直线

1

扫描转换的概念

光栅扫描式图形显示器：有一个点阵单元发生器，通过控制每个点阵单元的亮度来显示一副完整的图形。

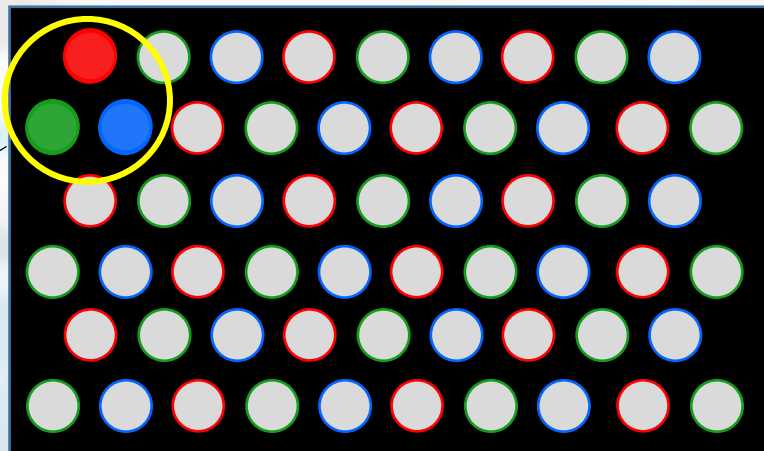


光栅扫描式图形显示器



和CRT相似，液晶显示器、LED显示器也都有一个点阵单元发生器，这个点阵单元其实就是像素点阵。

一组荧光小点

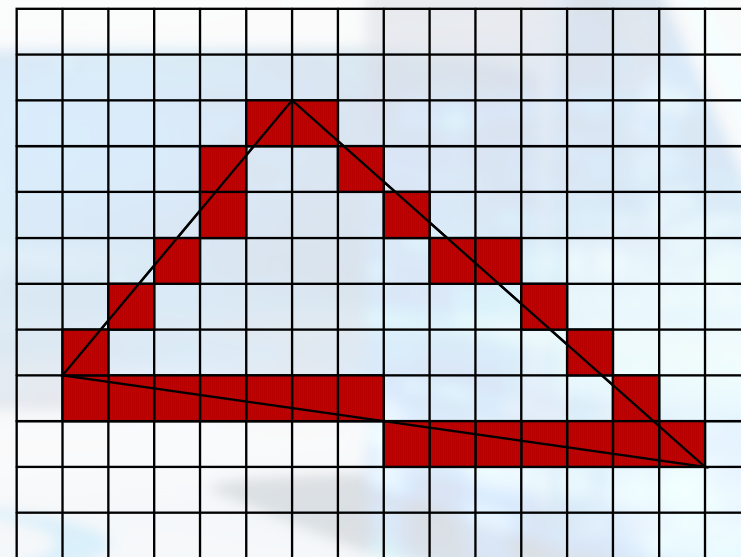


CRT的三色荧光屏



液晶显示屏的三色滤光片

一组亚像素



像素点阵

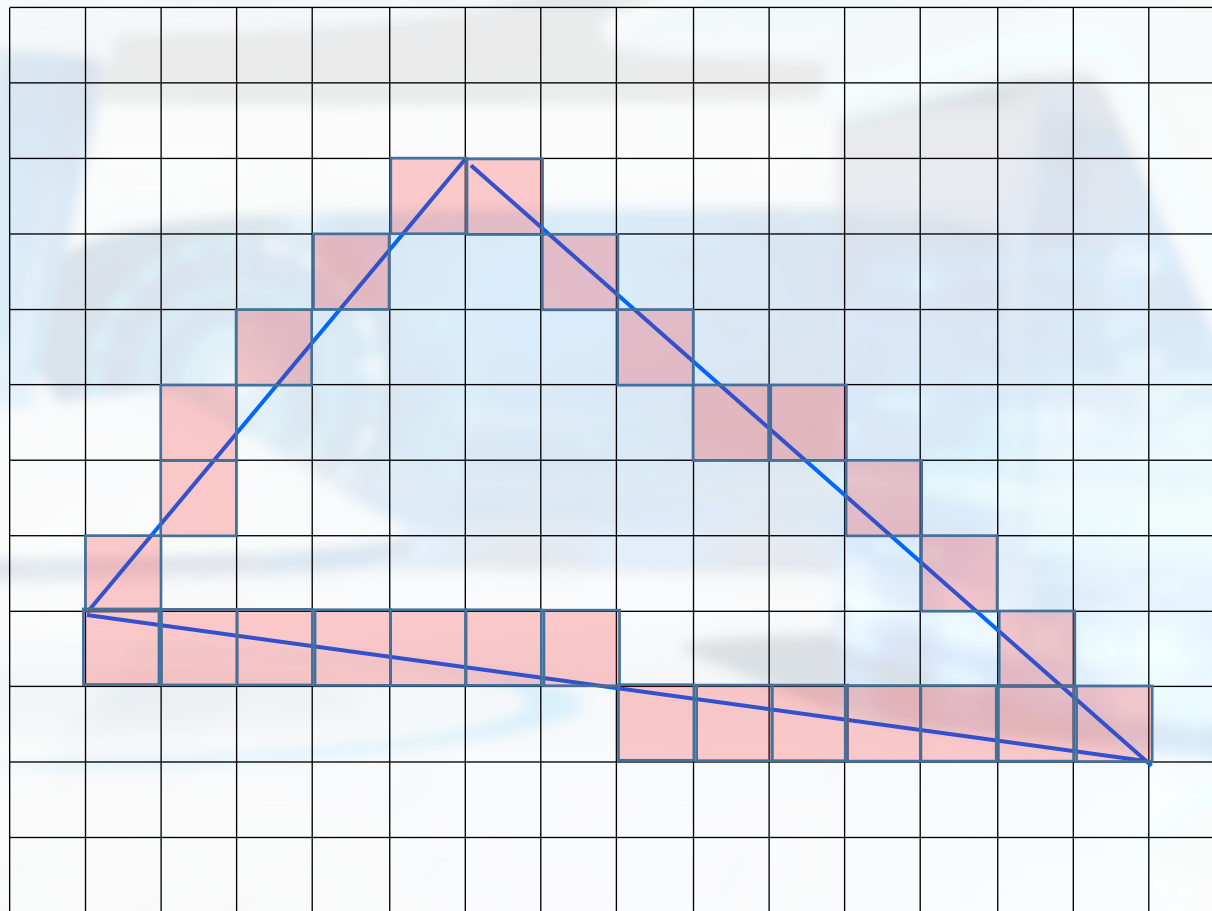
1

扫描转换的概念

图形的光栅化（图形的扫描转换）分成两步：

◆第一步：根据图形的定义在点阵单元上确定最佳逼近于图形的像素集

逼近的过程本质可以认为是
连续量向离散量的转换

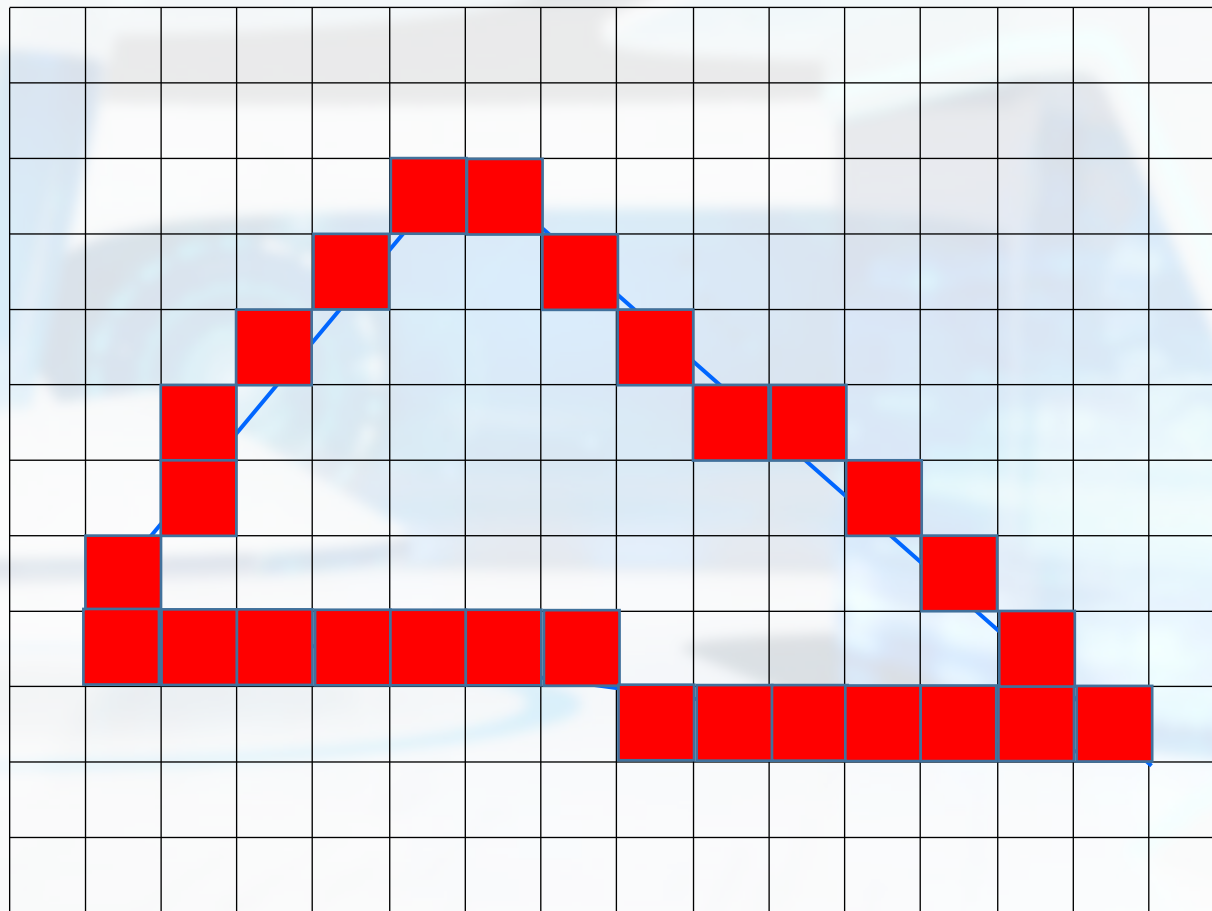


1

扫描转换的概念

图形的光栅化（图形的扫描转换）分成两步：

◆第二步：给像素指定合适的颜色值



1

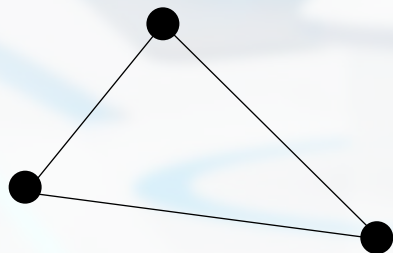
扫描转换的概念

这个过程叫做图形的光栅化（图形的扫描转换）：

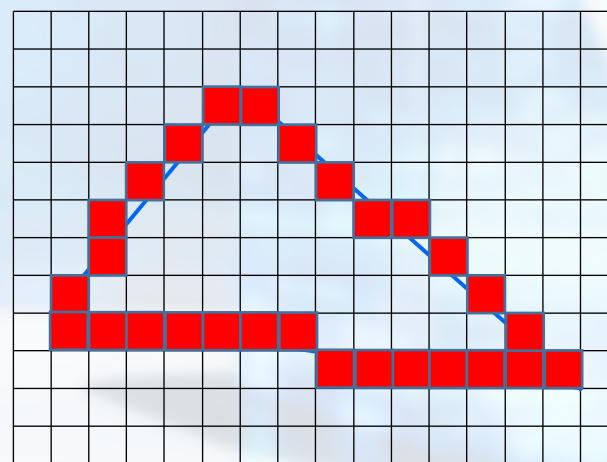
输入：三角形的定义

（1）几何信息：三个顶点的坐标

（2）属性信息：颜色、光照、纹理等。



找到最佳逼近的像素点

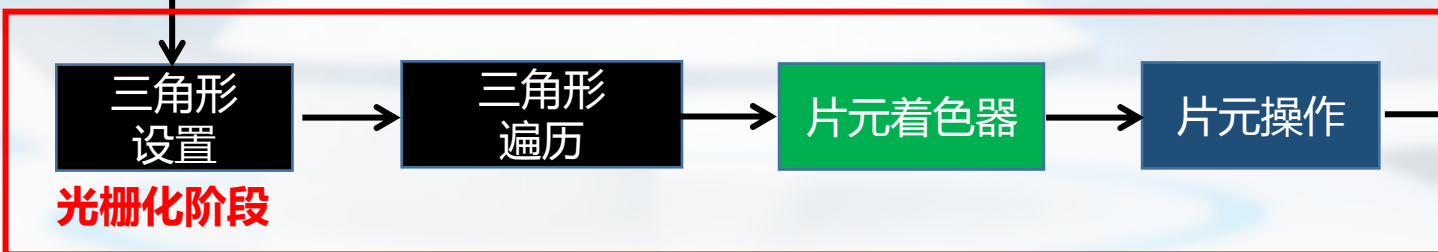
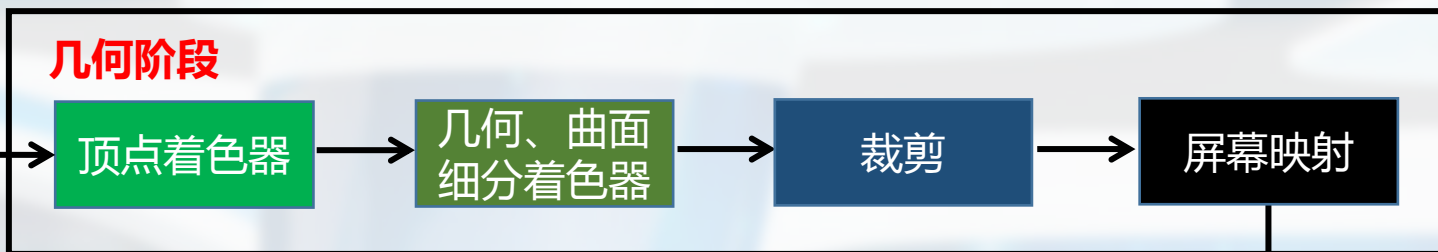


给像素指定合适的颜色值

1

扫描转换的概念

顶点数据
摄像机位置
光照纹理



说明：



可编程



可选



可配置



固定

```
0000000000000000
0000000000000000
0000011000000000
0000100100000000
0000100010000000
0001000001100000
0010000000010000
0100000000001000
0111111100000100
000000001111110
0000000000000000
0000000000000000
```

帧缓存

1

扫描转换的概念

顶点数据
摄像机位置
光照纹理



软光栅
(图形的扫描转换)



```
0000000000000000
0000000000000000
00000110000000000
0000100100000000
0000100010000000
001000001100000
0010000000010000
0100000000001000
0111111100000100
0000000011111110
0000000000000000
0000000000000000
```

帧缓存

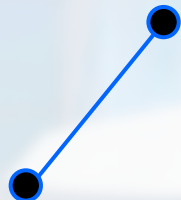
1

扫描转换的概念

最基本的图形的扫描转换算法，具体包括：



点



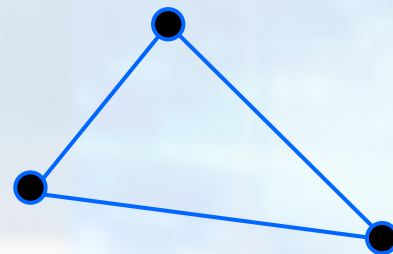
直线



圆



椭圆



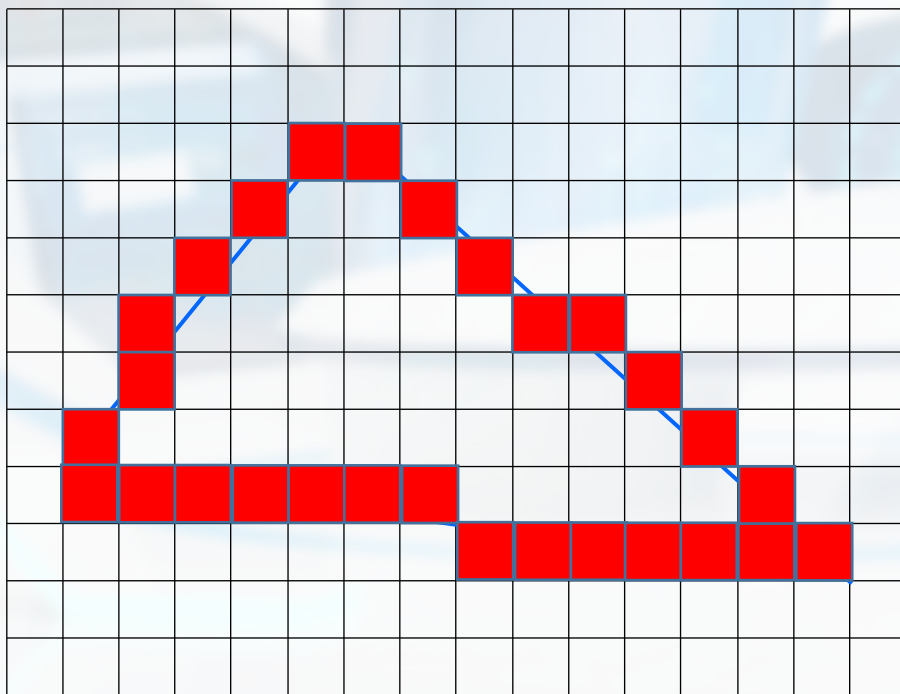
多边形

1

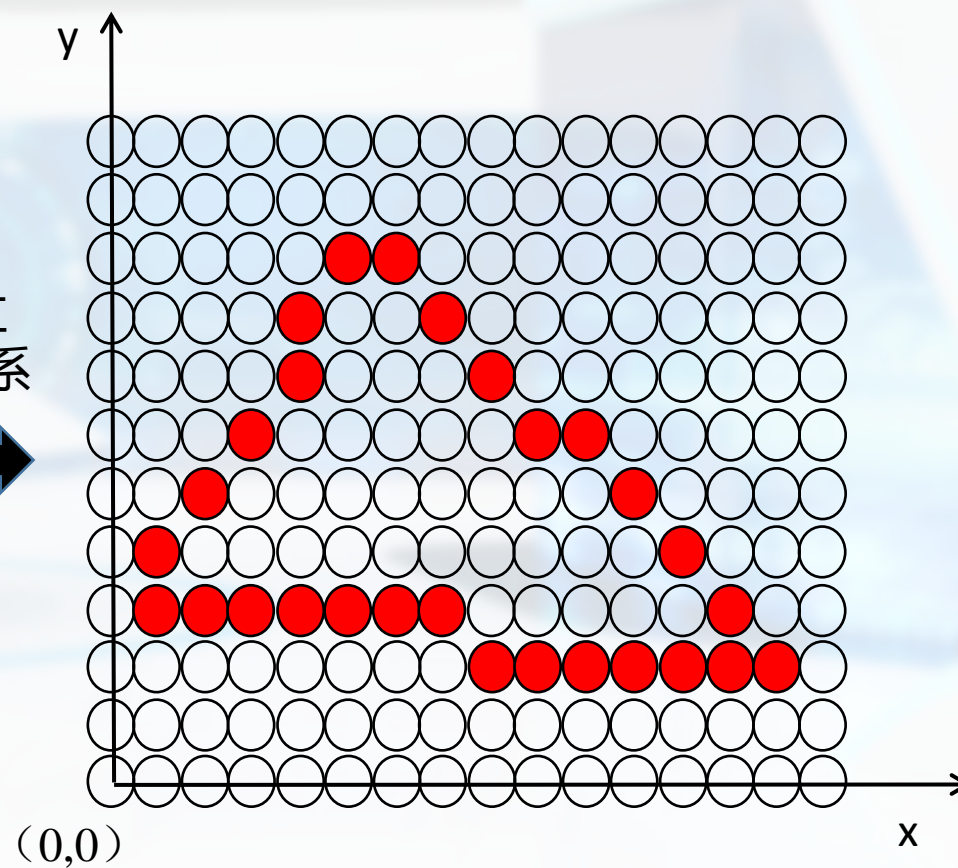
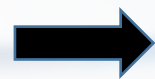
扫描转换的概念

约定

- ◆ 为像素点阵建立一个坐标系
- ◆ 先考虑二维图形的生成



建立
坐标系



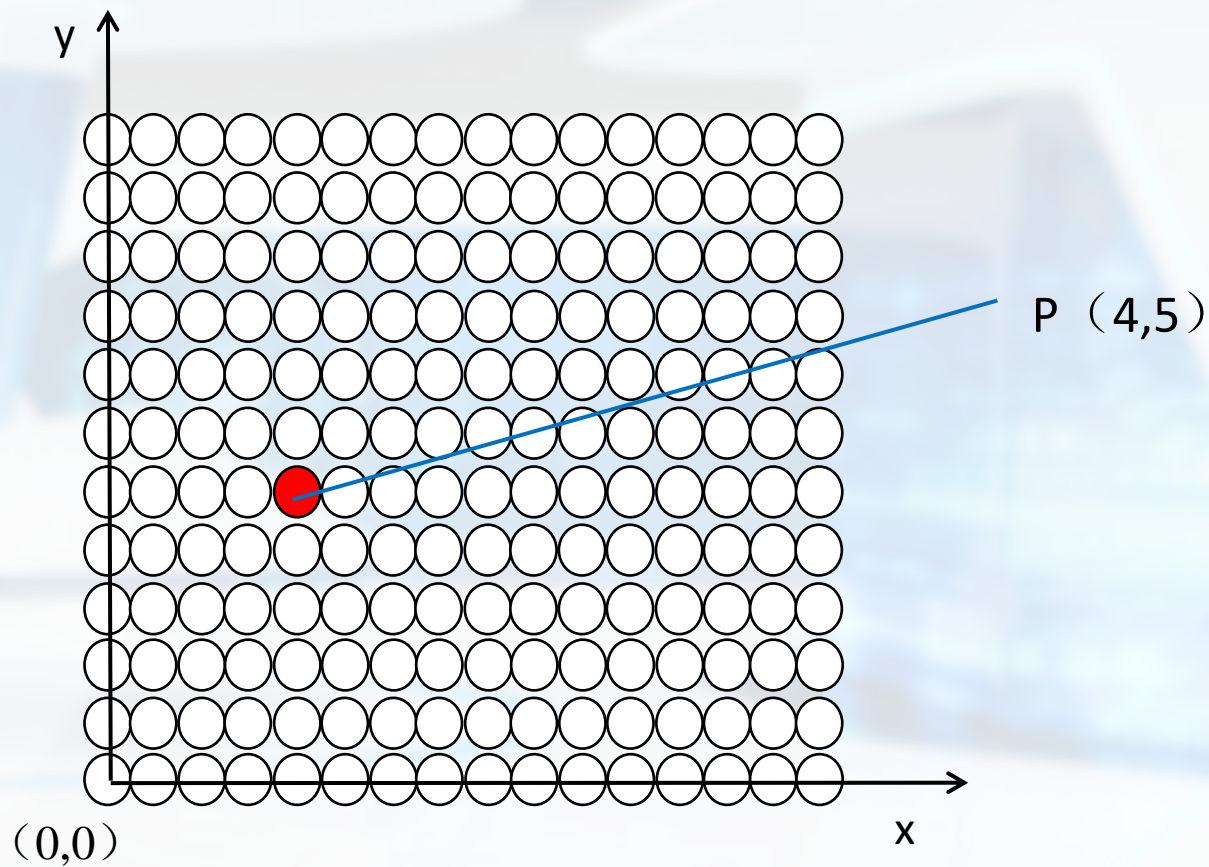
2

点

输入：点的坐标

输出：像素点的位置

$P(3.6, 4.7)$

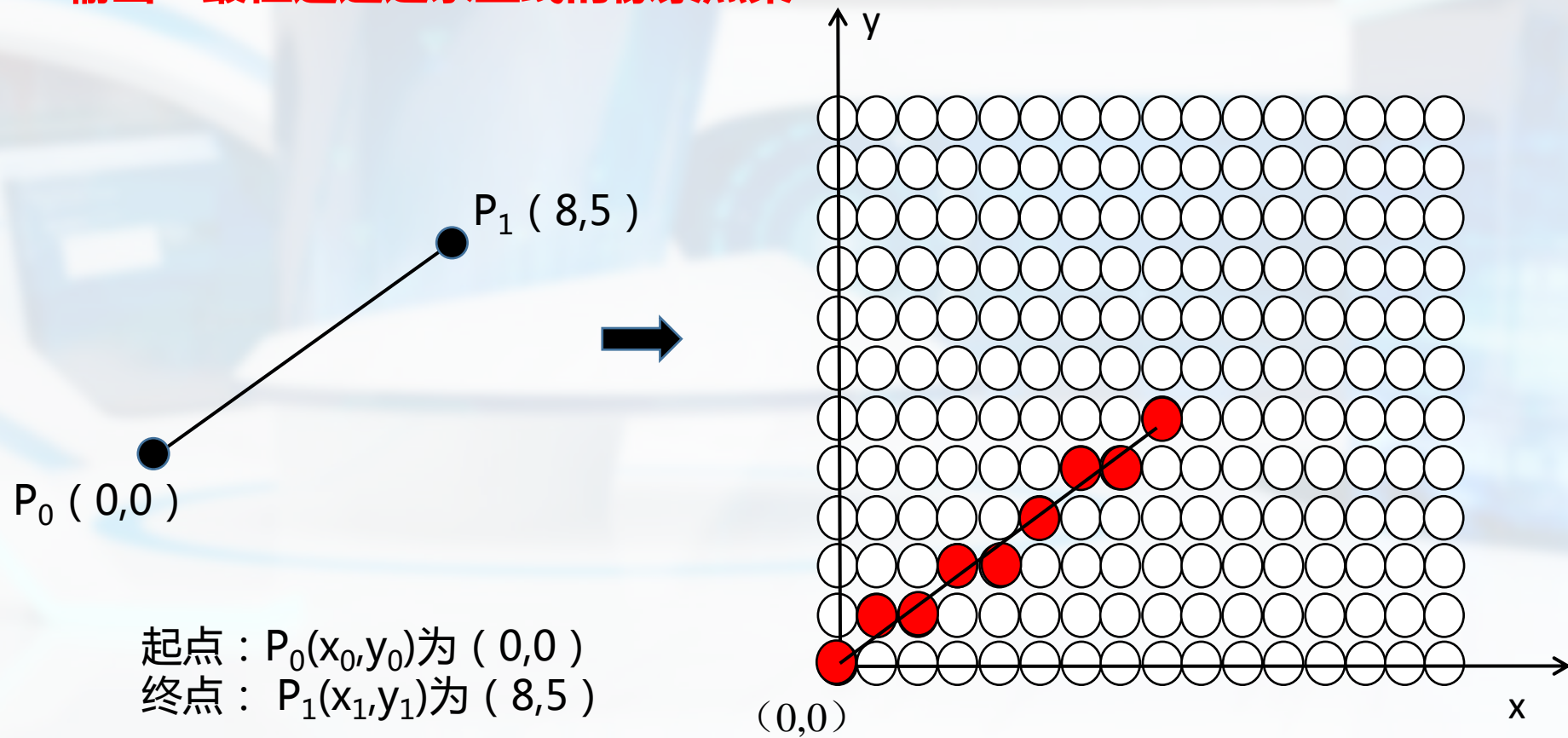


3

直线

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

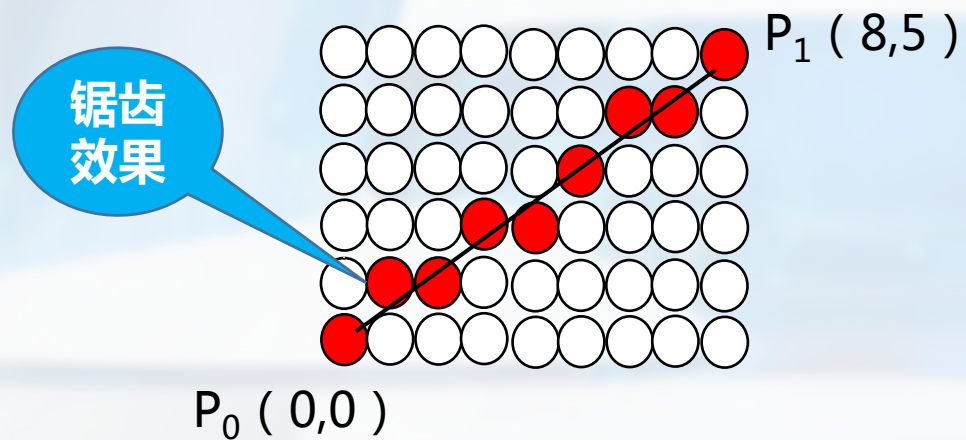


3

直线

高质量直线的要求

(1) 直线要直



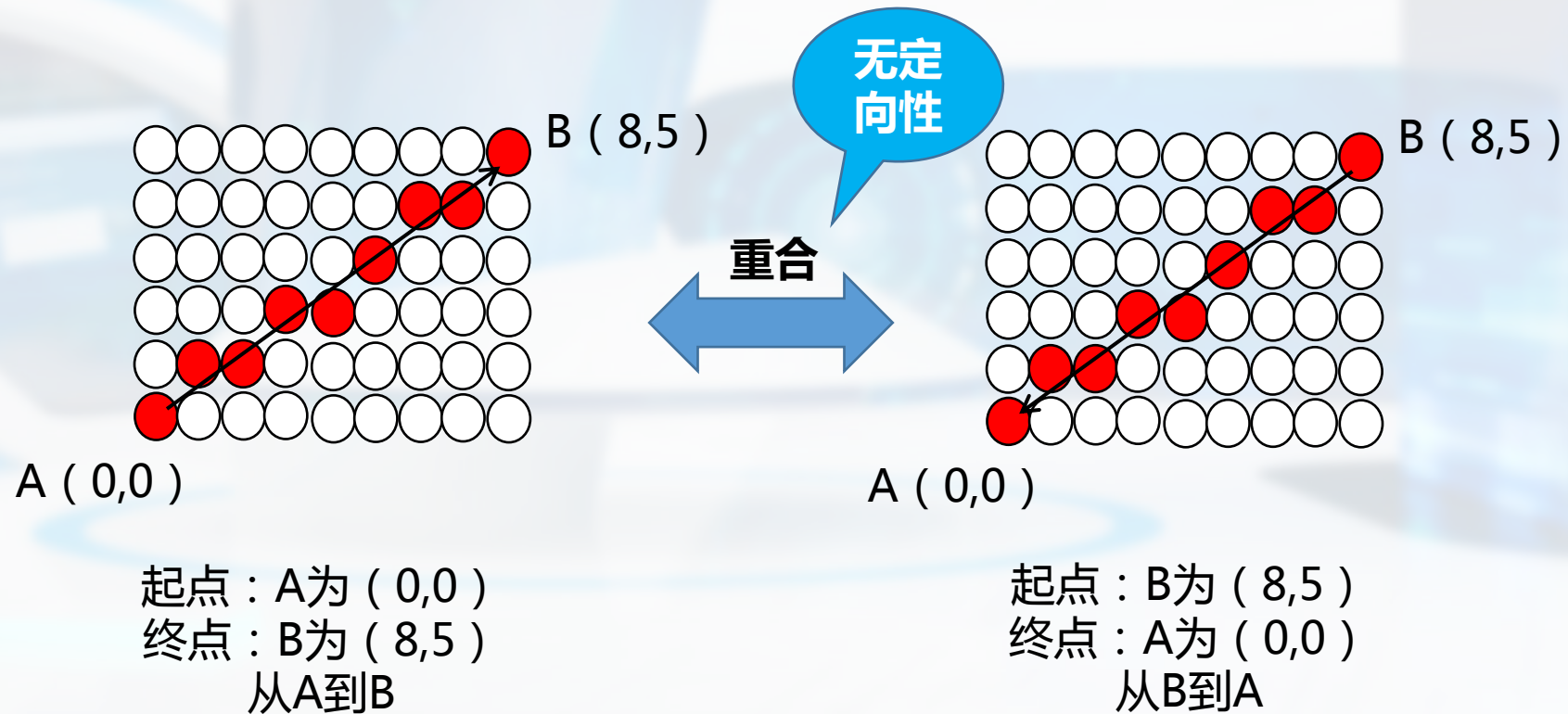
起点： $P_0(x_0, y_0)$ 为 $(0,0)$
终点： $P_1(x_1, y_1)$ 为 $(8,5)$

3

直线

高质量直线的要求

(2) 直线的端点要准确，即无定向性和断裂情况

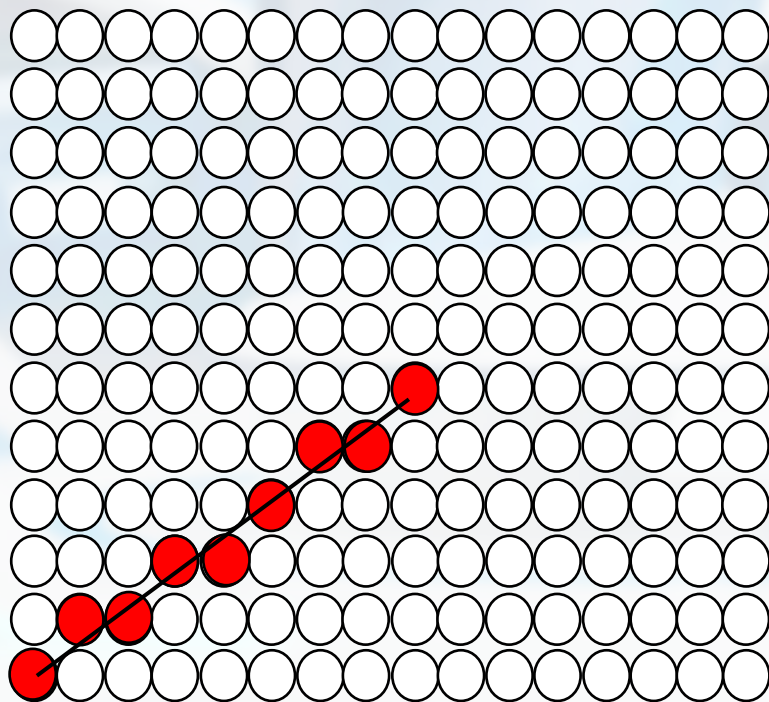


3

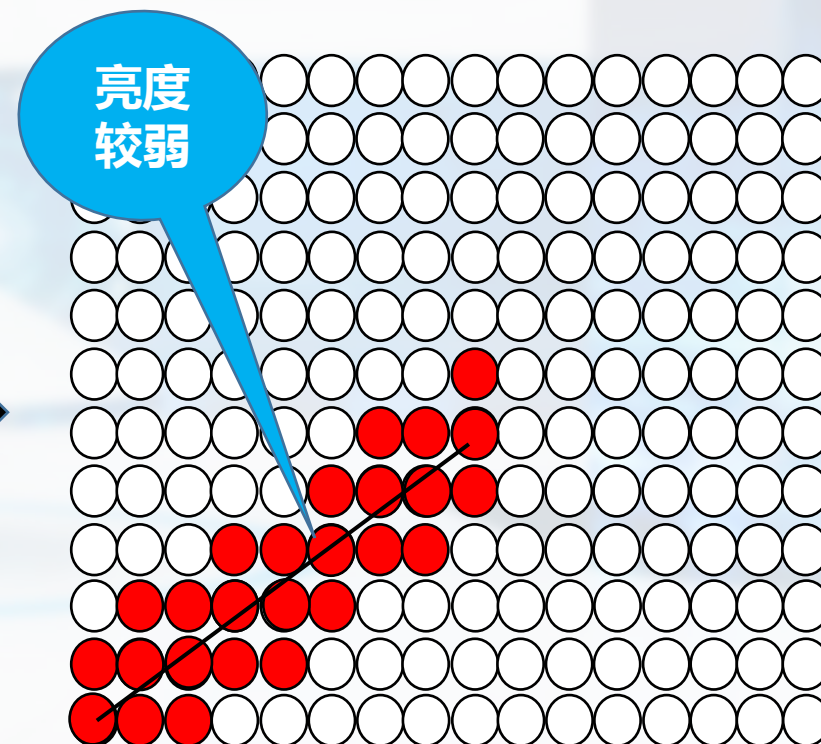
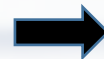
直线

高质量直线的要求

(3) 直线的亮度、色泽要均匀



线宽
处理

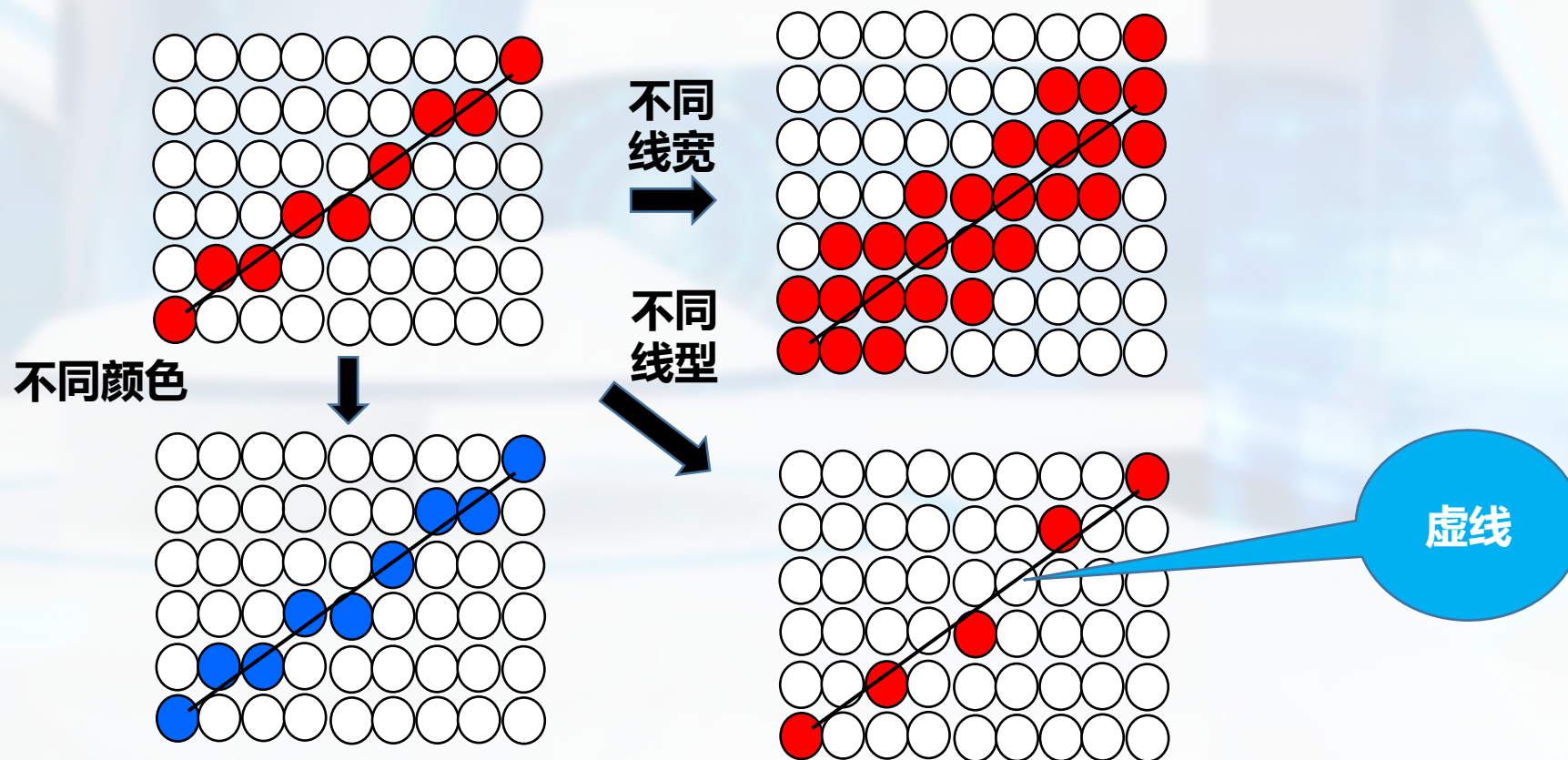


3

直线

高质量直线的要求

(4) 画线的速度要快，还能处理不同线宽、颜色、线型



3

直线

直线扫描转换算法：

逐点比较法

正负法

数值微分算法

Bresenham算法

3

直线

数值微分法 (Digital Differential Analyzer , 简称DDA)

一种直接从直线的微分方程生成直线的方法。

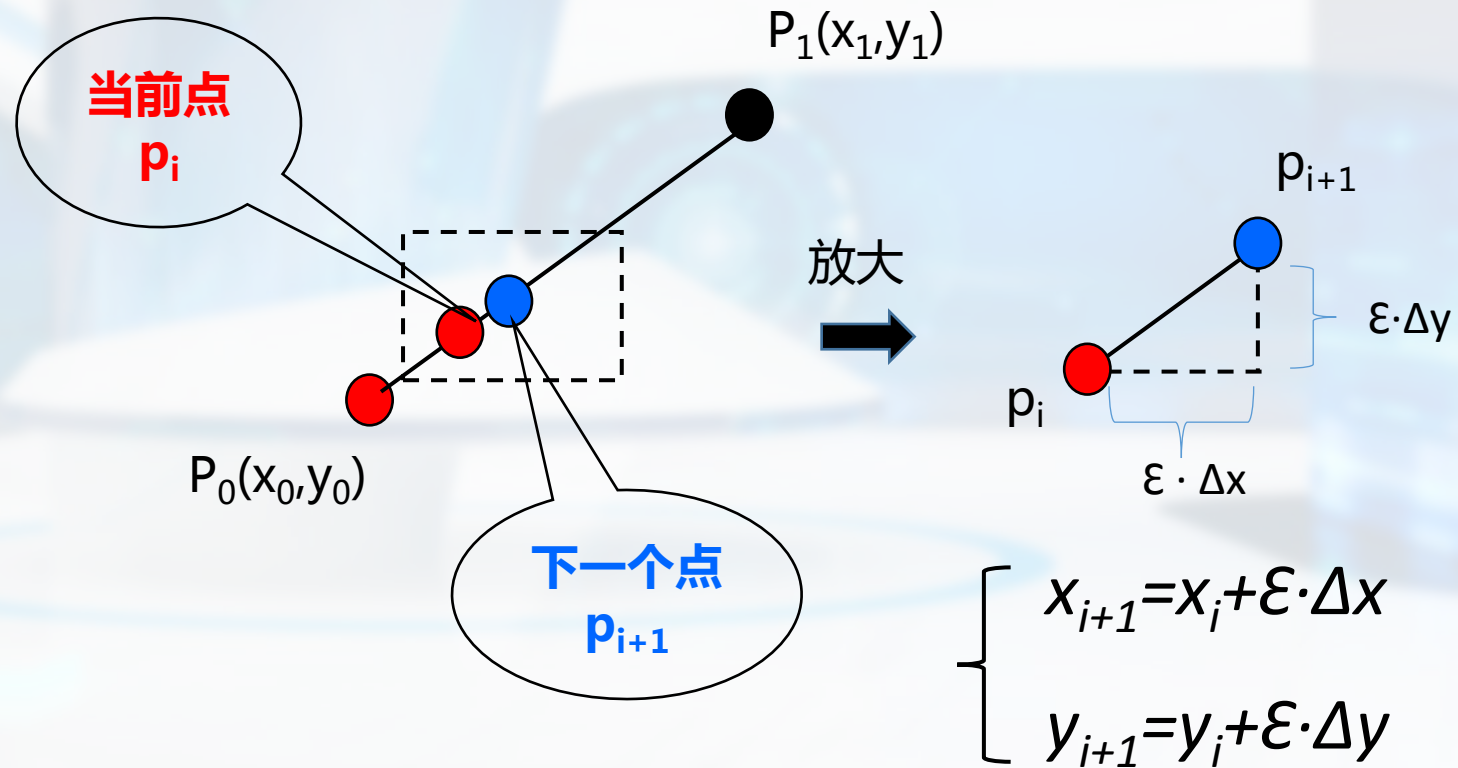
通过给定直线的两端点坐标 $P_0(x_0, y_0)$ 和 $P_1(x_1, y_1)$ ，我们可以得到直线的微分方程：

$$\frac{dy}{dx} = \frac{\Delta y}{\Delta x} = \frac{y_1 - y_0}{x_1 - x_0} = k$$

3

直线

数值微分法 (Digital Differential Analyzer , 简称DDA)



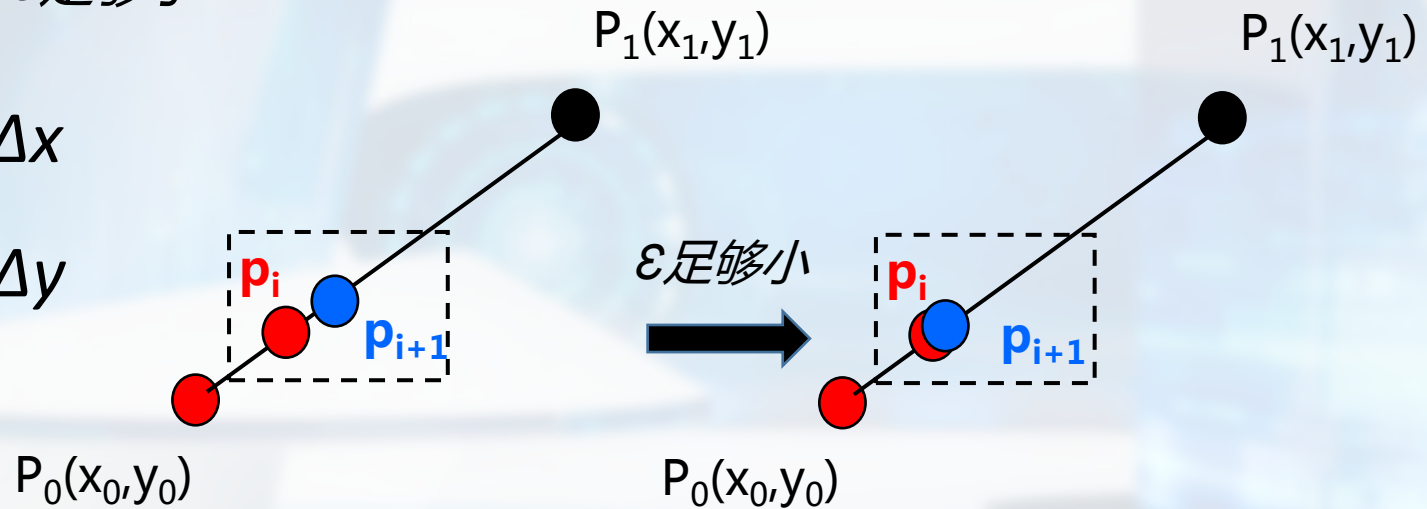
3

直线

数值微分法 (Digital Differential Analyzer , 简称DDA)

◆精度无限高： ε 足够小

$$\begin{cases} x_{i+1} = x_i + \varepsilon \cdot \Delta x \\ y_{i+1} = y_i + \varepsilon \cdot \Delta y \end{cases}$$

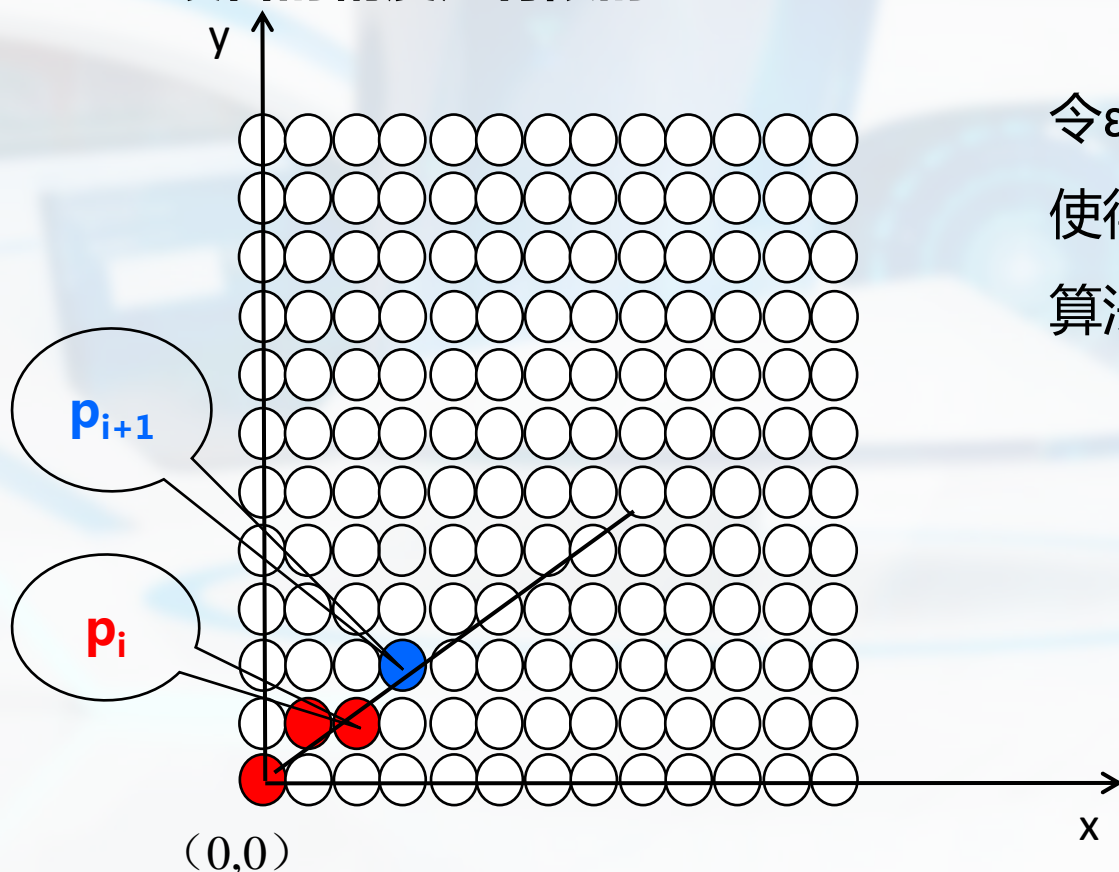


3

直线

数值微分法 (Digital Differential Analyzer , 简称DDA)

◆设备的精度是有限的：



令 $\epsilon = 1/\max(|\Delta x|, |\Delta y|)$

使得 $\epsilon \Delta x$ 或 $\epsilon \Delta y$ 中会有一个变成单位步长

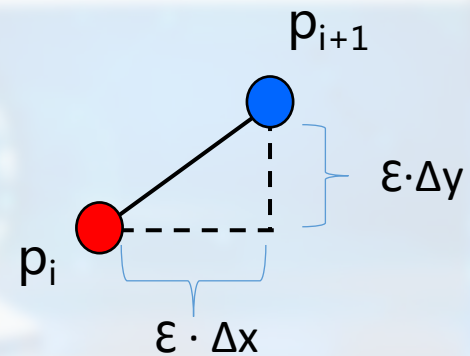
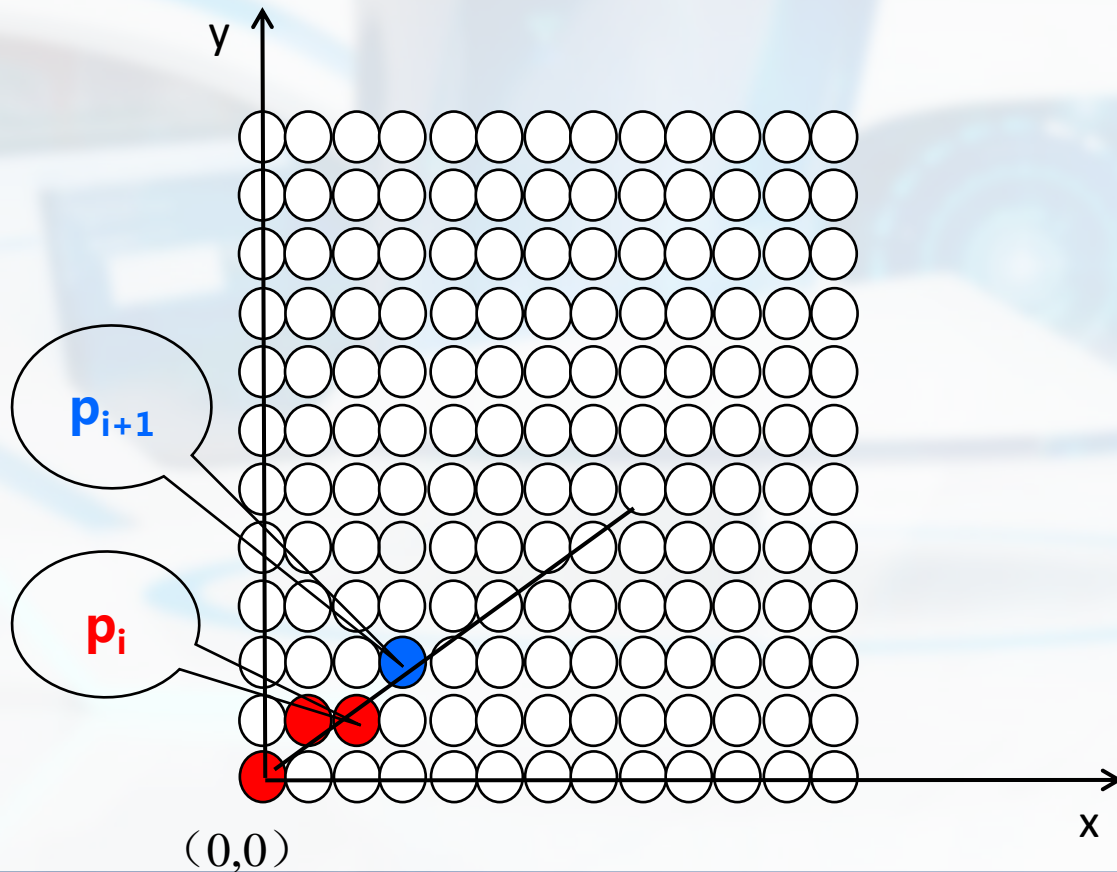
算法在**最大位移方向**上，每次总是走一步

3

直线

令 $\varepsilon = 1/\max(|\Delta x|, |\Delta y|)$

情况一 斜率绝对值小于1 : $\varepsilon = 1/|\Delta x|$



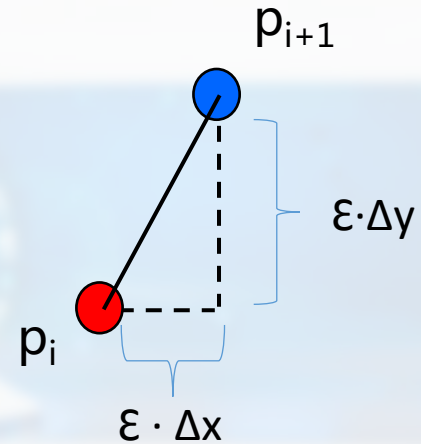
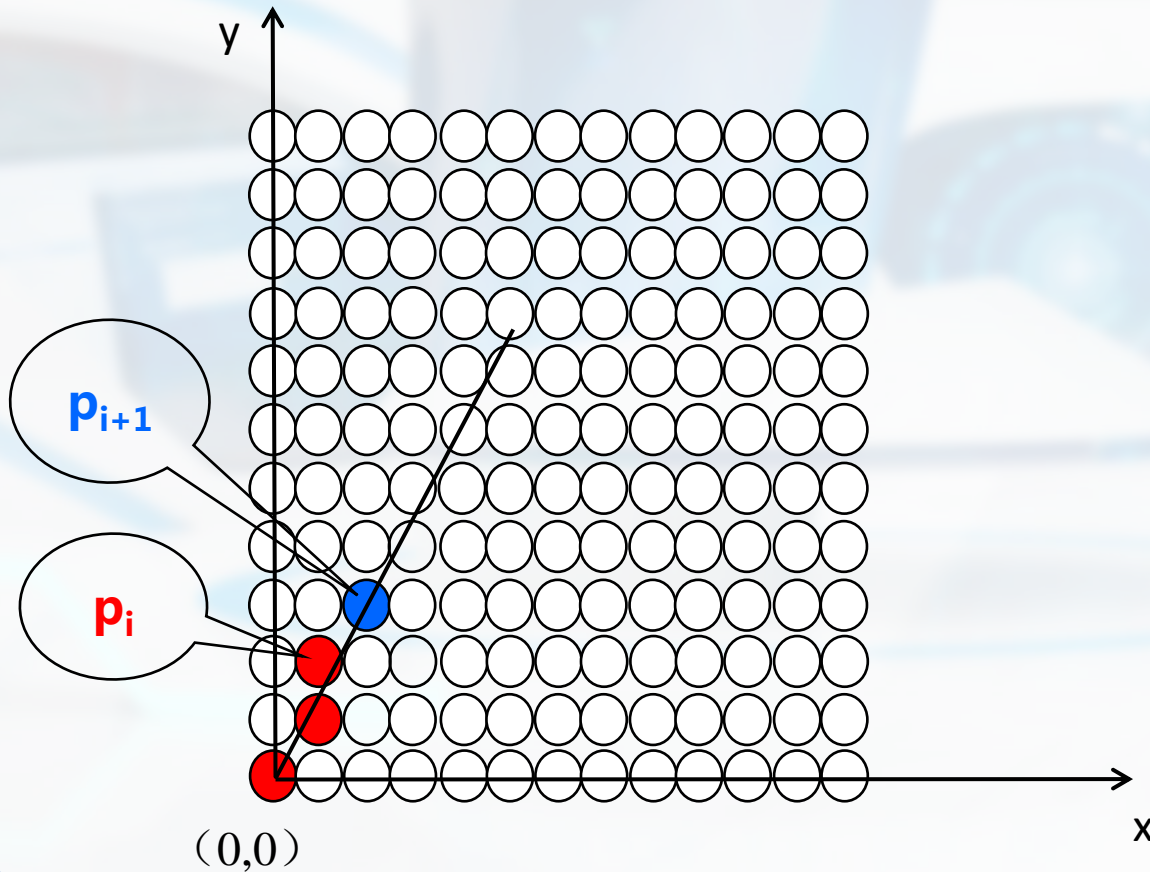
$$\begin{cases} x_{i+1} = x_i \pm 1 \\ y_{i+1} = y_i \pm k \end{cases}$$

3

直线

令 $\varepsilon = 1/\max(|\Delta x|, |\Delta y|)$

情况二 斜率绝对值大于1 : $\varepsilon = 1/|\Delta y|$

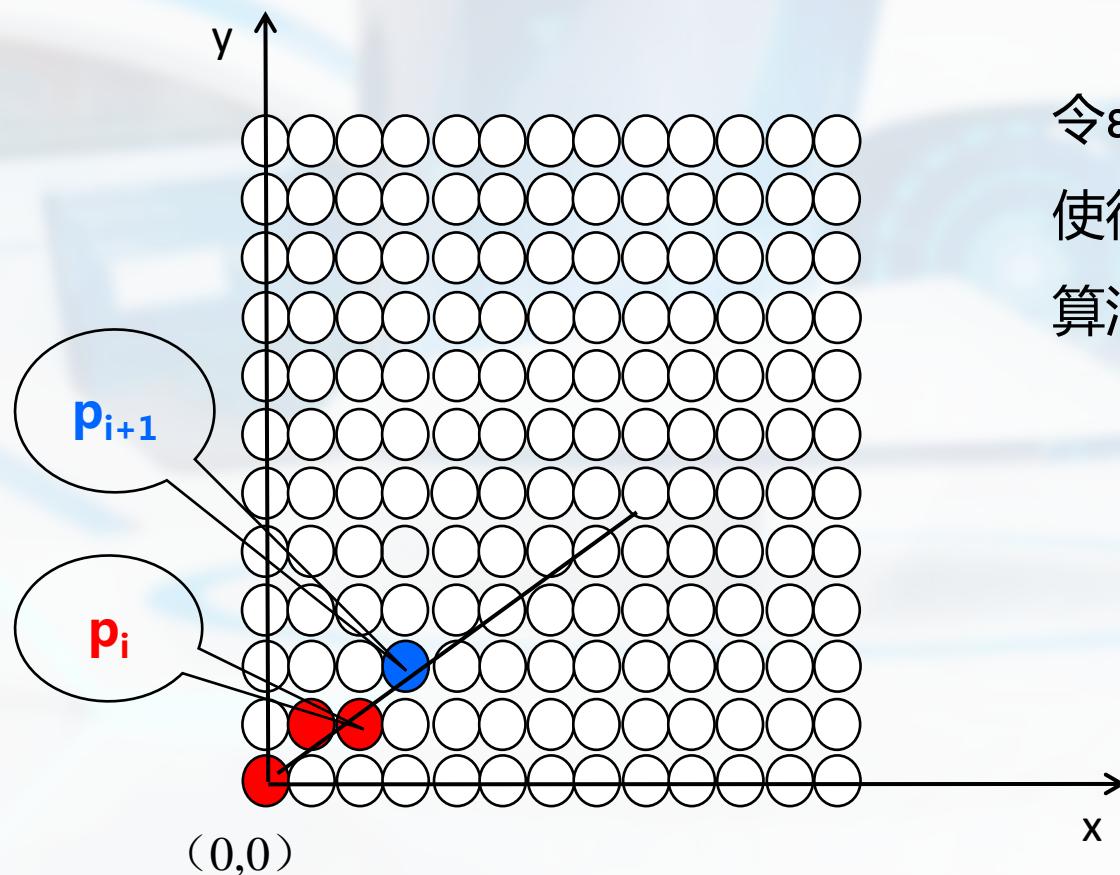


$$\begin{cases} x_{i+1} = x_i \pm 1/k \\ y_{i+1} = y_i \pm 1 \end{cases}$$

3

直线

数值微分法 (Digital Differential Analyzer , 简称DDA)



令 $\epsilon = 1/\max(|\Delta x|, |\Delta y|)$

使得 $\epsilon \Delta x$ 或 $\epsilon \Delta y$ 中会有一个变成单位步长

算法在**最大位移方向**上，每次总是走一步

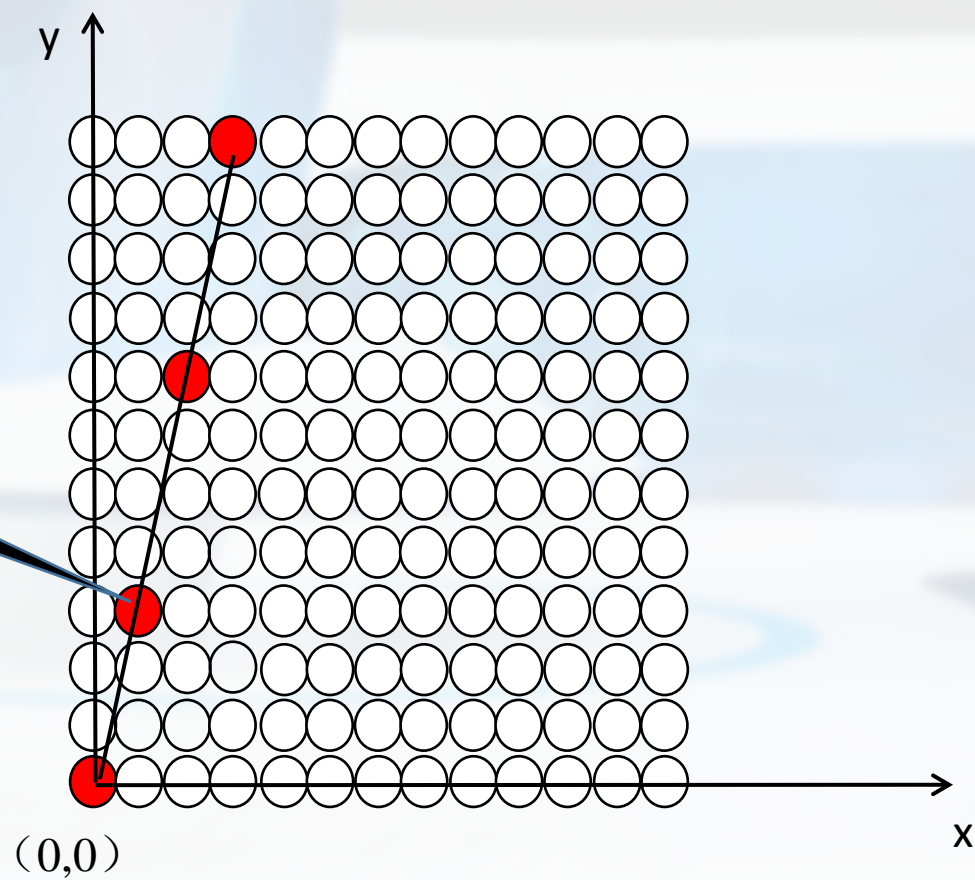
why

3

直线

最大位移方向的重要性

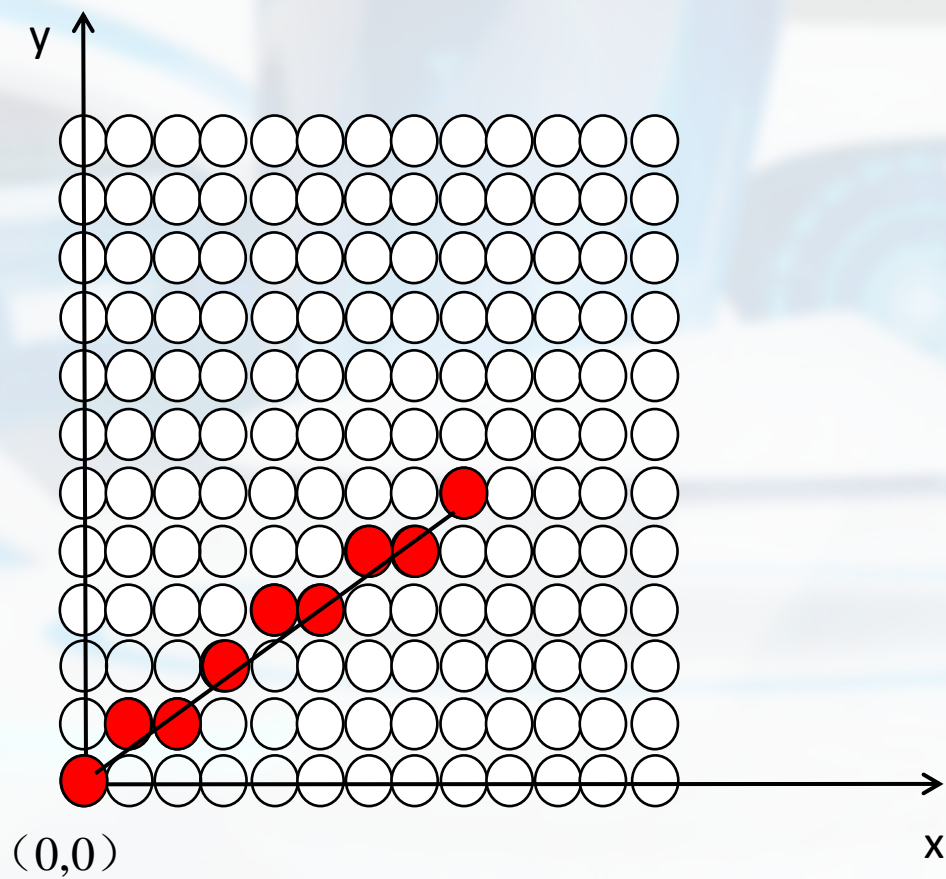
离散点



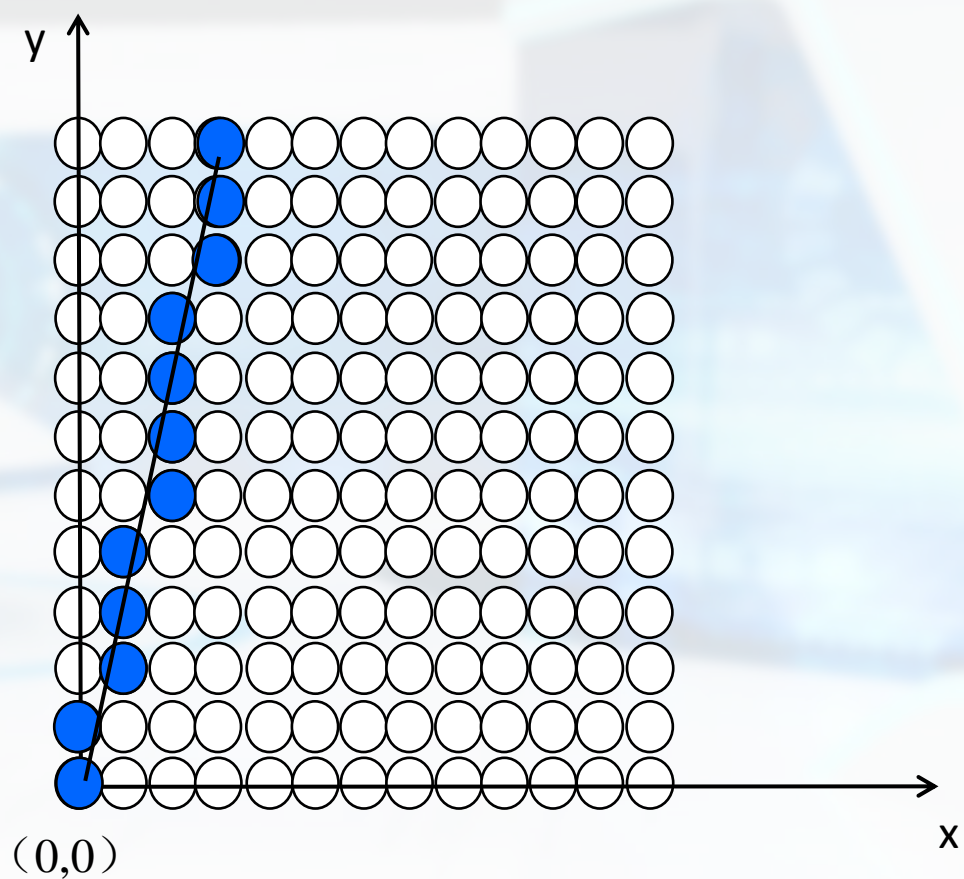
3

直线

情况一的直线生成



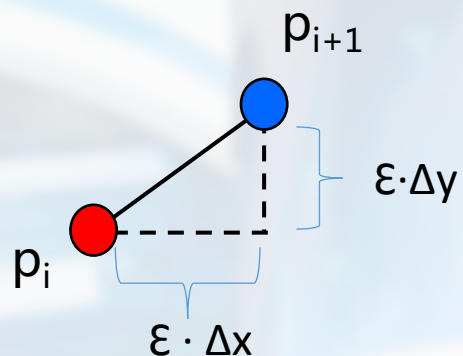
情况二的直线生成



3

直线

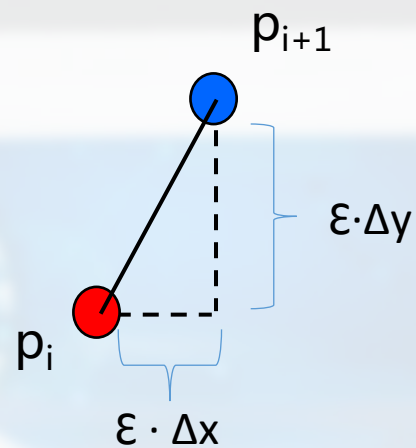
情况一的直线生成



$$\begin{cases} x_{i+1} = x_i \pm 1 \\ y_{i+1} = y_i \pm k \end{cases}$$

$\text{round}(y_{i+1}) \rightarrow (\text{int})(y_{i+1} + 0.5)$

情况二的直线生成



$$\begin{cases} x_{i+1} = x_i \pm 1/k \\ y_{i+1} = y_i \pm 1 \end{cases}$$

$\text{round}(x_{i+1})$
 \downarrow
 $(\text{int})(x_{i+1} + 0.5)$

四舍五入

3

直线

DDA直线生成算法(用大家易于理解的C语言代码)

```
void DDALine(int x0,int y0,int x1,int y1)
{
    int  dx,dy,eps1,k;
    float x,y,xIncre,yIncre;
    dx=x1-x0; dy=y1-y0;
    x=x0;    y=y0;
    If (abs(dx)>abs(dy)) eps1=abs(dx);
    else eps1=abs(dy);
    xIncre=(float)dx/(float)eps1;
    yIncre=(float)dy/(float)eps1;
    for (k=0;k<=eps1;k++)
    {
        putpixel((int)(x+0.5),(int)(y+0.5));//在对应坐标处输出像素点
        x+=xIncre;
        y+=yIncre;
    }
}
```

输出最佳逼近的像素点

最大位移方向的判断

x、y方向上增量的计算

3

直线

DDA直线生成算法特点

在一个迭代算法中，如果每一步的x、y值是用前一步的值加上一个增量来获得的，那么，这种算法就称为增量算法。因此，DDA算法是一个**增量算法**。

优点：DDA算法直观、易实现

缺点：有浮点数和浮点运算，效率不高

```
.....  
float x,y,xIncre,yIncre;  
.....  
xIncre=(float)dx/(float)eps1;  
yIncre=(float)dy/(float)eps1;  
.....
```



谢谢

软件学院 万琳