计算机图形学

第二章：光栅图形学算法

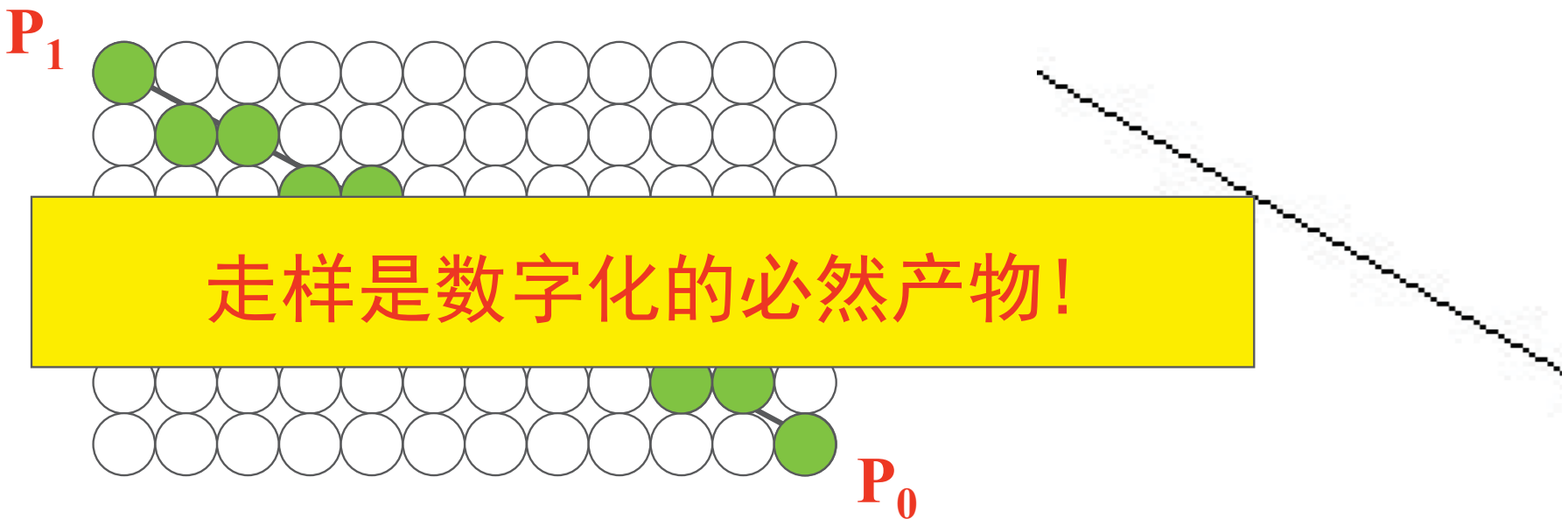
光栅图形学算法的研究内容

- 直线段的扫描转换算法
- 多边形的扫描转换与区域填充算法
- 直线裁剪算法
- 反走样算法
- 消隐算法

对直线、圆及椭圆这些最基本元素的生成速度和显示质量的改进，在图形处理系统中具有重要的应用价值

但它们生成的线条具有明显的“锯齿形”即会发生走样（Liasing）现象

一、走样



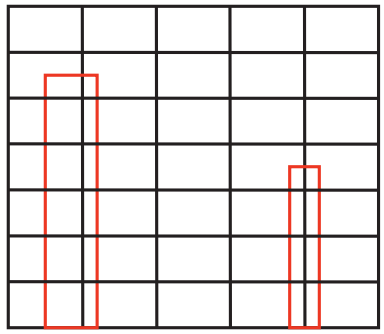
“锯齿”是“走样”（aliasing）的一种形式。而走样是光栅显示的一种固有性质。产生走样现象的原因是像素本质上的离散性

走样现象：

一是光栅图形产生的阶梯形（锯齿形）

二是图形中包含相对微小的物体时，这些物体在静态图形中容易被丢弃或忽略

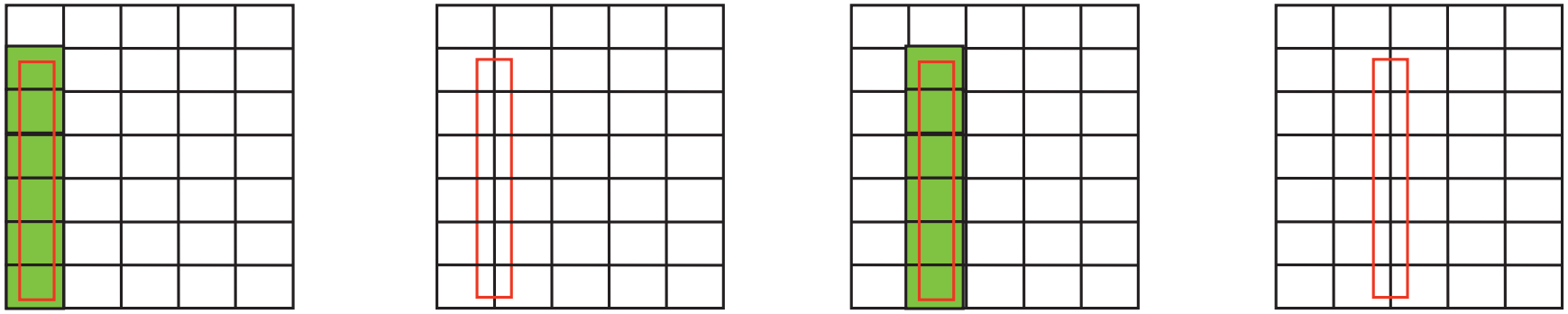
小物体由于“走样”而消失



(a) 需显示的矩形

(b) 显示结果

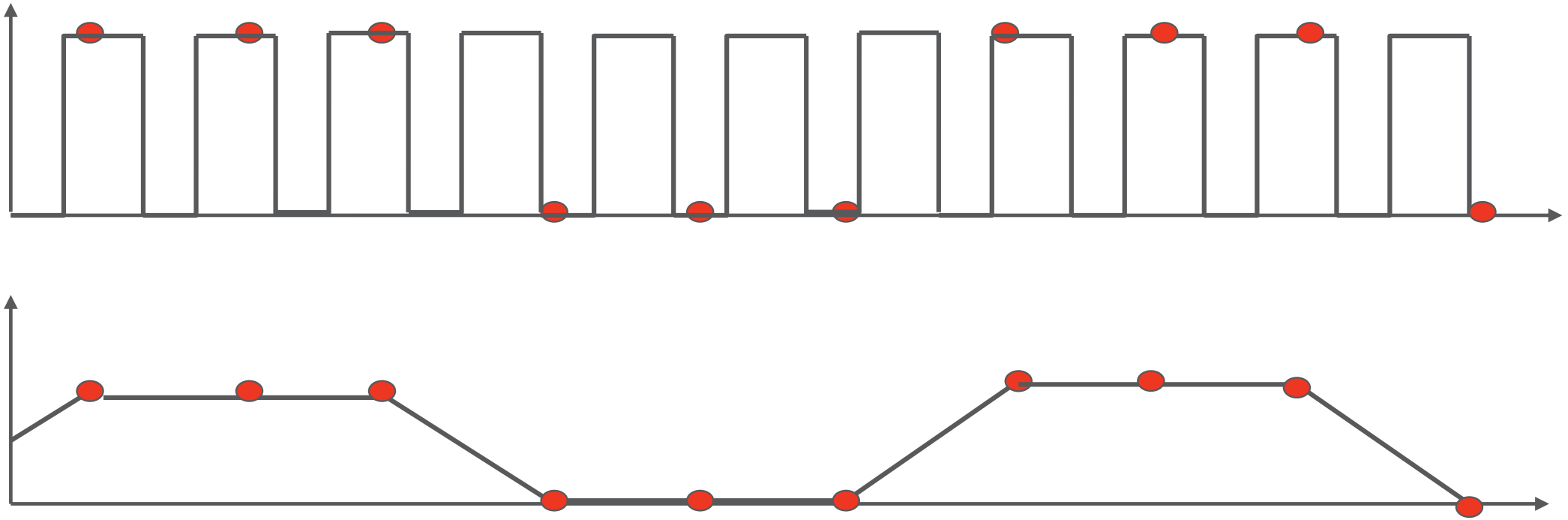
在动画序列中时隐时现，产生闪烁



(a) 显示　(b) 不显示　(c) 显示　(d) 不显示

矩形从左向右移动，当其覆盖某些像素中心时，矩形被显示出来，当没有覆盖像素中心时，矩形不被显示

简单地说，如果对一个快速变化的信号采样频率过低，所得样本表示的会是低频变化的信号：原始信号的频率看起来被较低的“走样”频率所代替



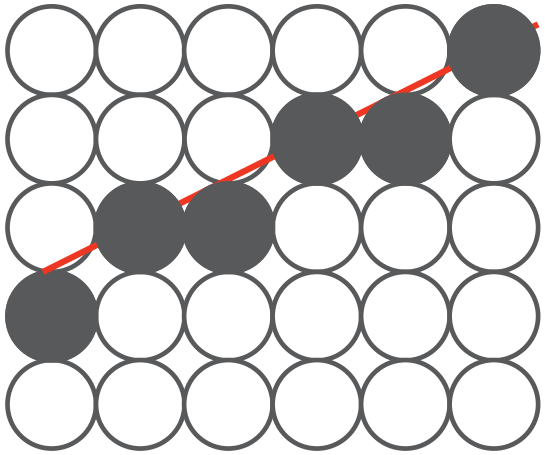
二、反走样技术

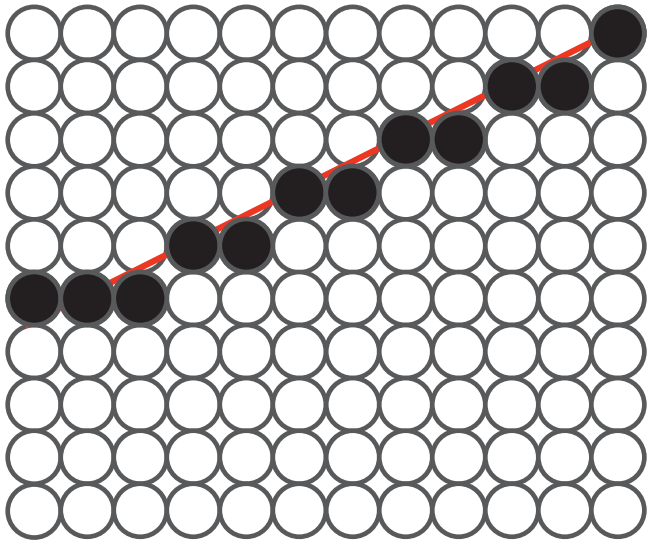
如何降低由于采样不足而产生的走样现象呢？

用于减少或消除走样效果的技术，称为反走样（Antialiasing）技术

由于图形的走样现象对图形的质量有很大影响，几乎所有图形处理系统都要对基本图形进行反走样处理

采用分辨率更高的显示设备，对解决走样现象有所帮助，因为可以使锯齿相对物体会更小一些





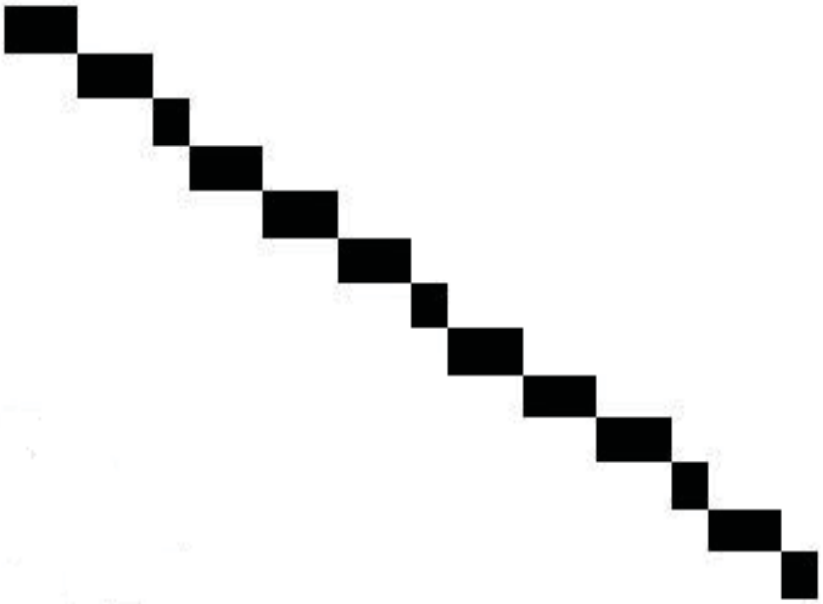
该反走样方法是以4倍的存储器代价和扫描转换时间获得的

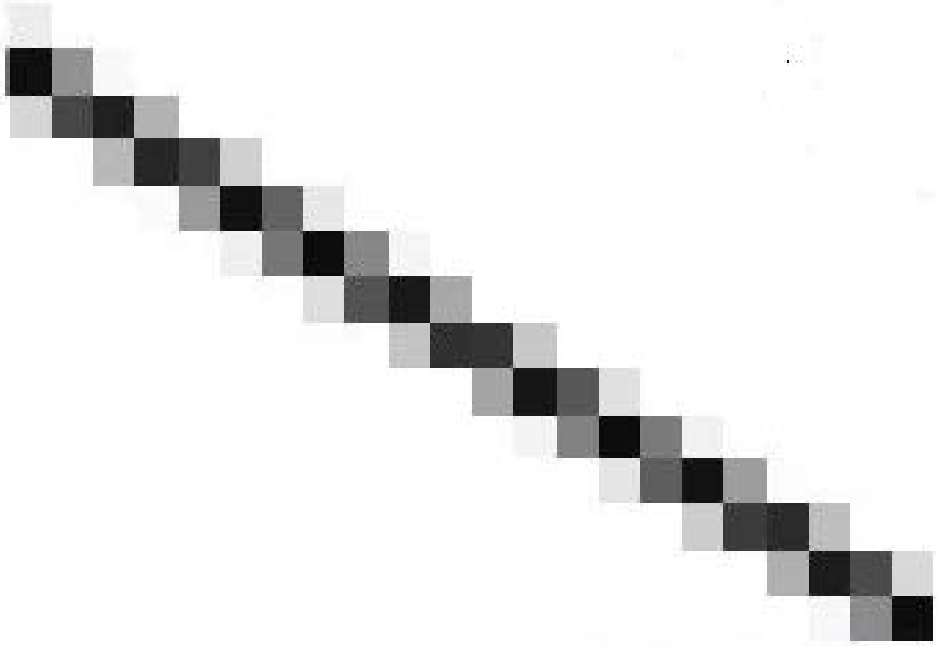
为了稳定屏幕上的图象，电子枪至少要1/24秒时间轰击屏幕上所有像素一次，如果像素提高一倍，电子枪就要快4倍！

反走样技术涉及到某种形式的“模糊”来产生更平滑的图像

对于在白色背景中的黑色矩形，通过在矩形的边界附近掺入一些灰色像素，可以柔化从黑到白的尖锐变化

从远处观察这幅图像时，人眼能够把这些缓和变化的暗影融合在一起，从而看到了更加平滑的边界





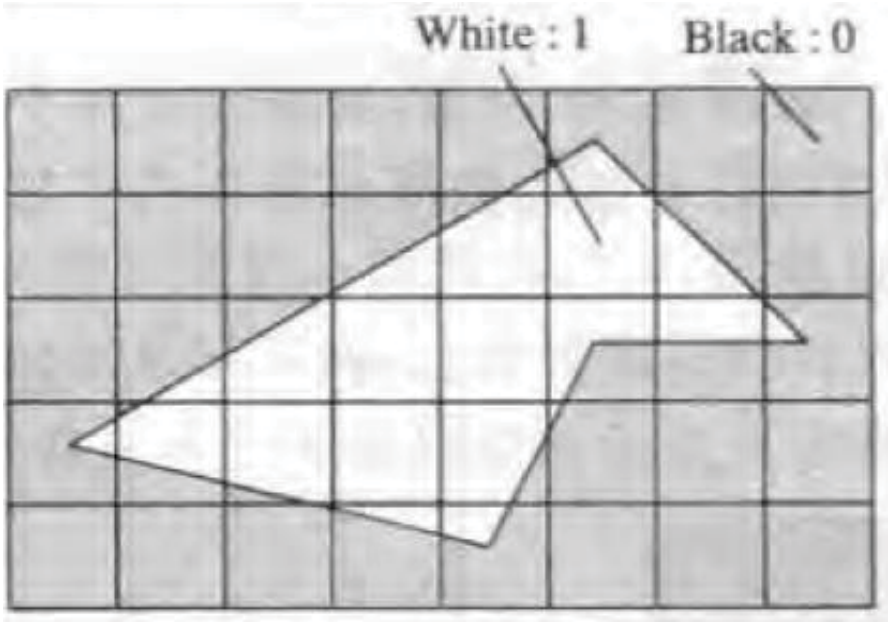
介绍两种反走样方法：

1、非加权区域采样方法

2、加权区域采样方法

1、非加权区域采样方法

根据物体的覆盖率（coverage）计算像素的颜色。覆盖率是指某个像素区域被物体覆盖的比例



0	0	0					
					15	8	
						7	

把这个多边形放在方格线中，其中每个正方形的中心对应显示器上一个像素中心。被多边形覆盖了一半的像素的亮度值赋为1/2, 覆盖三分之一的像素赋值为1/3；以此类推

如果帧缓冲区的每个像素有4个比特位，那么0表示黑色，。。。15表示白色

非加权区域采样方法有两个缺点：

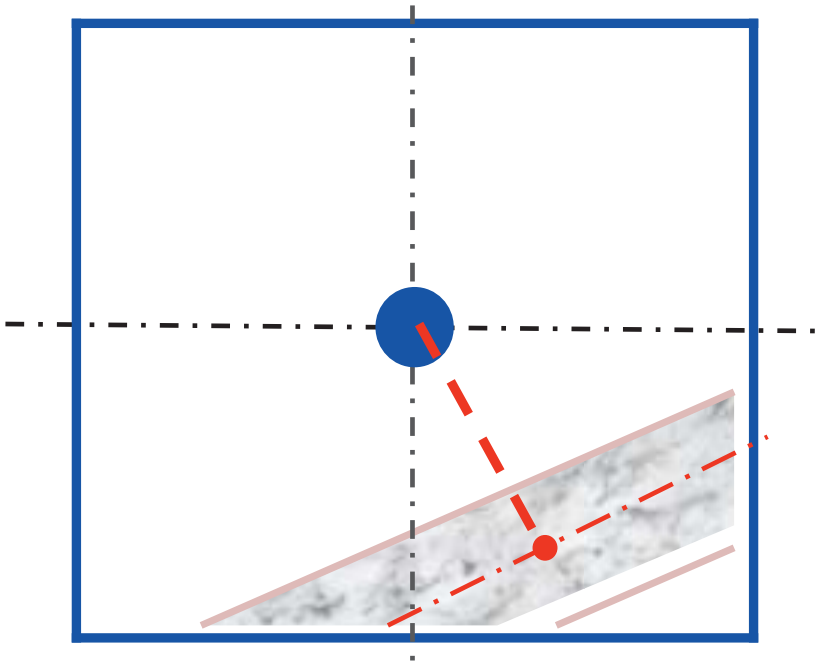
1、象素的亮度与相交区域的面积成正比，而与相交区域落在象素内的位置无关，这仍然会导致锯齿效应

2、直线条上沿理想直线方向的相邻两个象素有时会有较大的灰度差

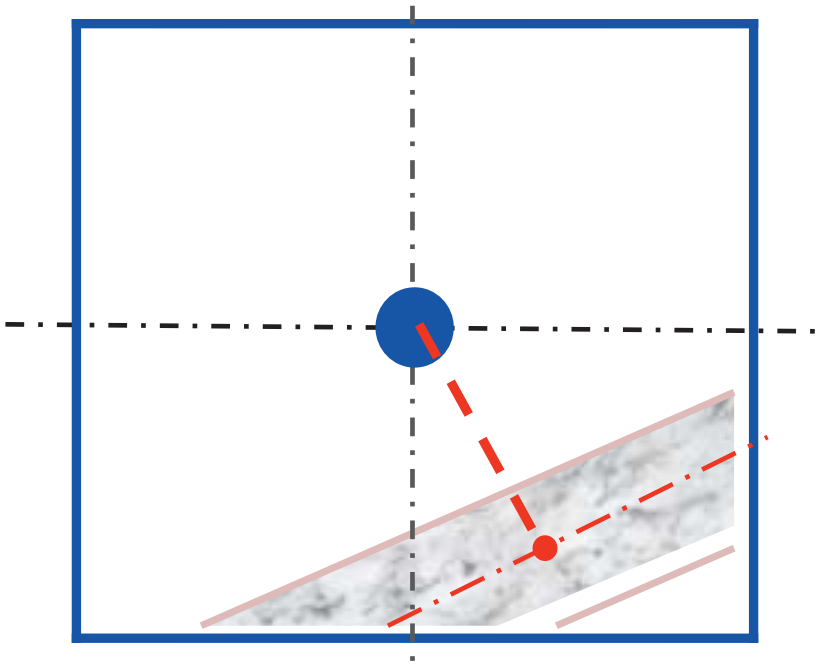
每个像素的权值是一样的，这是它的主要缺点。所以也称非加权区域采样方法

2、加权区域采样方法

这种方法更符合人视觉系统对图像信息的处理方式，反走样效果更好



将直线段看作是具有一定宽度的狭长矩形；当直线段与像素有交时，根据相交区域与像素中心的距离来决定其对象素亮度的贡献



直线段对一个像素亮度的贡献正比于相交区域与像素中心的距离

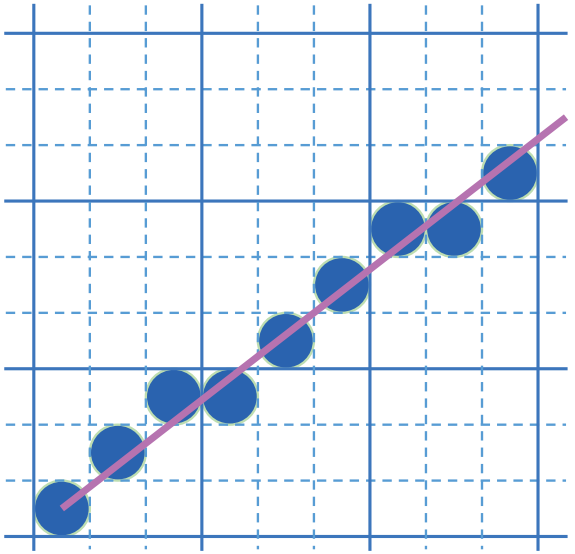
设置相交区域面积与像素中心距离的权函数（高斯函数）反映相交面积对整个像素亮度的贡献大小

利用权函数积分求相交区域面积，用它乘以像素可设置的最大亮度值，即可得到该像素实际显示的亮度值

可采用离散计算方法

将一个像素划分为 n = 3×3个子象素，加权表可以取作：

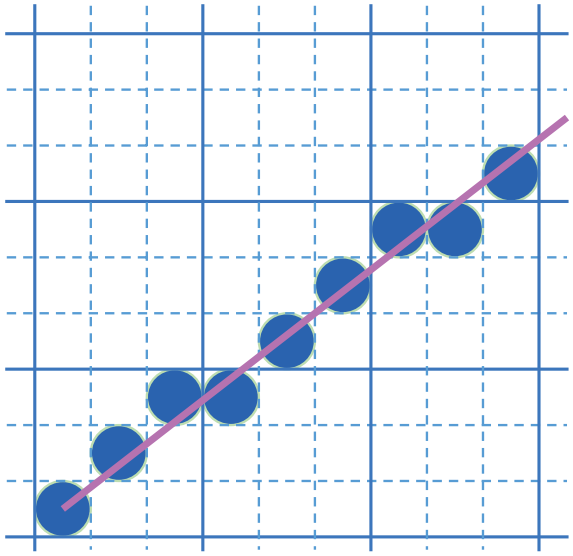
$$\begin{bmatrix} w1 & w2 & w3 \\ w4 & w5 & w6 \\ w7 & w8 & w9 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$



每个像素含9个子像素

加权方案：中心子像素的加权是角子像素的4倍，是其它像素的2倍，对九个子像素的每个网格所计算出的亮度进行平均

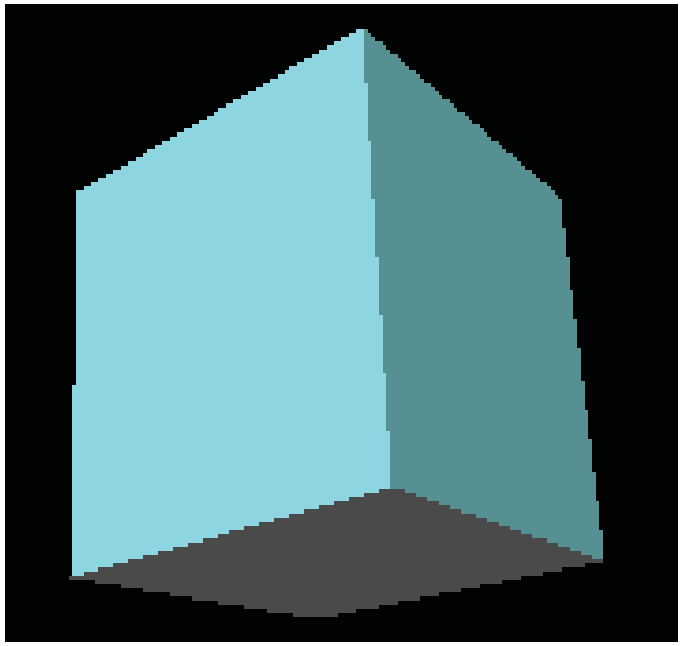
$$\begin{bmatrix} w1 & w2 & w3 \\ w4 & w5 & w6 \\ w7 & w8 & w9 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$



每个像素含9个子像素

- 然后求出所有中心落于直线段内的子象素。
- 最后计算所有这些子象素对原象素亮度贡献之和

反走样是图形学中的一个根本问题，不可能避免；是图形学中的一个永恒问题



原 图

反走样图