



# 线宽线型、字符、反走样

王坤峰 教授

信息科学与技术学院



# 内容

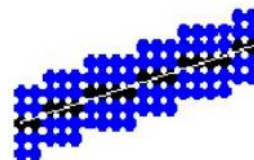
- 线宽处理
- 线型处理
- 字符处理
- 走样与反走样

# 线宽处理

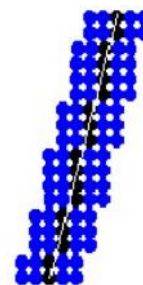
## ■ 直线线宽的处理

### — 线刷

- 算法简单、效率高
- 线的始末端是水平或垂直的
- 斜率不同，线的宽度不同
- 对于宽度为偶数个像素的直线会产生偏移



$m \in (-1, 1)$



$m \notin (-1, 1)$

### — 方形刷

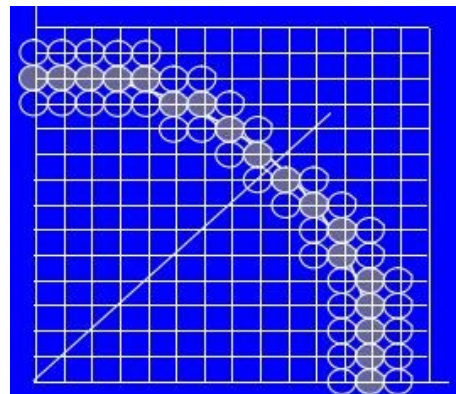
- 方形中心对准单像素宽的线条上各个像素
- 存在重复写像素的问题
- 线条末端是水平或垂直的，且线宽与线条方向有关
- 对于宽度为偶数个像素的直线会产生偏移

### — 区域填充方法

## ■ 圆弧线宽的处理

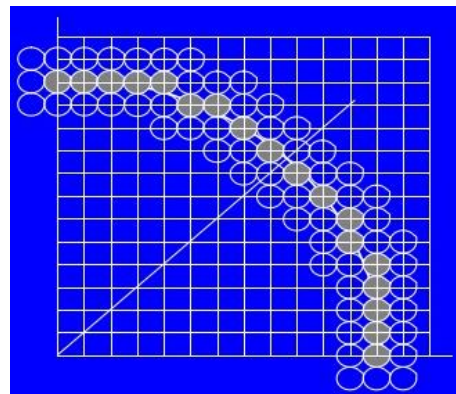
### — 线刷

- 在小于 $45^\circ$ 的弧时用水平刷子，在大于 $45^\circ$ 的弧时用垂直刷子，即在经过曲线斜率为 $\pm 1$ 的点时刷子的方向要改变



### — 方形刷

- 不需要改变刷子方向。只需沿着单像素宽的轨迹，把正方形中心对准轨迹上的像素，把方形内的像素全部置成线条颜色



### — 区域填充方法

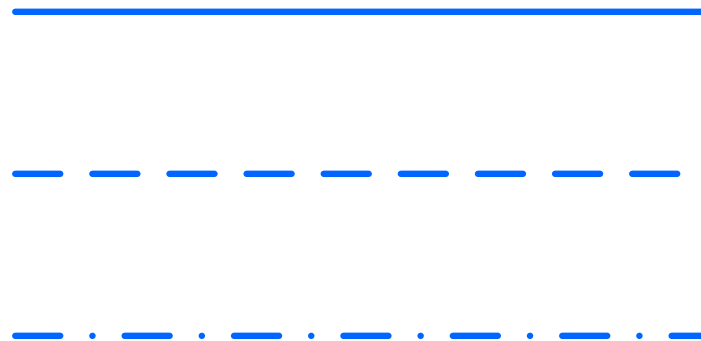


# 线型处理

## ■ 线型

- 实线、虚线、点划线
- 线型的表示
  - 一个布尔值的序列表示
- 实现

```
if (位串[i % 32])  
    drawpixel(x,y,color);
```





# 字符处理

## ■ 什么是字符？

- 字符指数字、字母、汉字等符号。
- 计算机中字符由一个数字编码唯一标识。

## ■ ASCII码

- “美国信息交换用标准代码集”简称ASCII码。它  
是用7位二进制数进行编码表示128个字符，包括字母、标点、运算符以及一些特殊符号。

## ■ 汉字编码

- 汉字编码的国家标准字符集GB 2313-80。每个符号  
由一个区码和一个位码（2字节）共同标识。

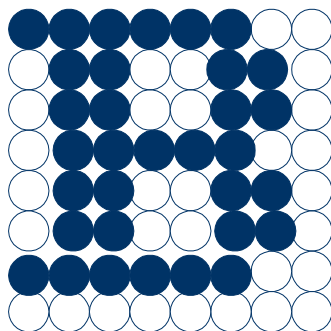
## ■ 表示方法

### — 点阵字符

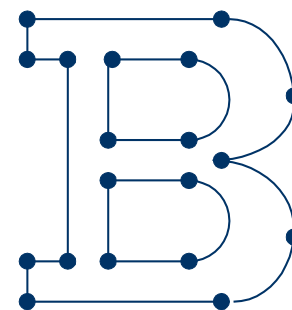
- 每个字符由一个位图表示

### — 矢量字符

- 由直线段和曲线段组成，记录字符的笔画信息



|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |





## ■ 特点:

### — 点阵字符

- 每个字符由一个位图表示
- 存储空间大，但易于显示
- 显示时，首先从字库中将位图检索出来，然后将位图写到帧缓冲器中

### — 矢量字符

- 记录字符的笔划信息而不是整个位图
- 存储量小，美观，变换方便，但是需要光栅化后才能显示



## ■ 字符属性

### — 字体

- 宋体 仿宋体 楷体 黑体 隶书

### — 字高

### — 字宽

### — 字倾斜角

- 倾斜

### — 字色

- 红色 绿色 蓝色



# 走样与反走样

## ■ 走样

- 用离散量表示连续量引起的失真现象称之为走样 (aliasing)

## ■ 反走样

- 用于减少或消除这种效果的技术称为反走样 (antialiasing)

# 反走样方法

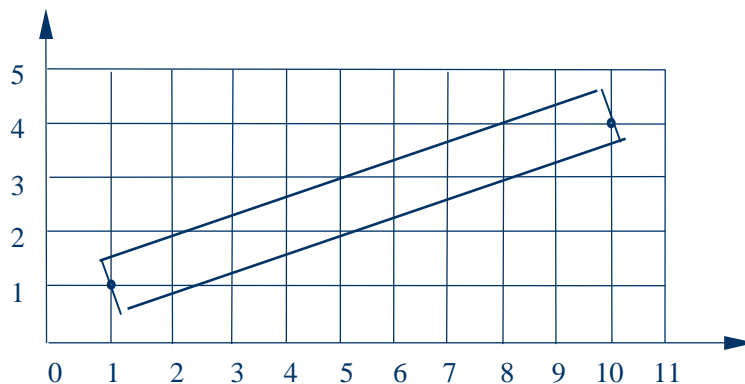
## ■ 提高分辨率

- 把显示器分辨率提高，显示出的直线段看起来就平滑光滑了一些。
- 阶梯的宽度减小了一倍。
- 导致4倍的存储器代价和2倍的扫描转换时间。
- 增加分辨率虽然简单，但不是经济的方法，而且它也只能减轻而不能消除锯齿问题

# 区域采样

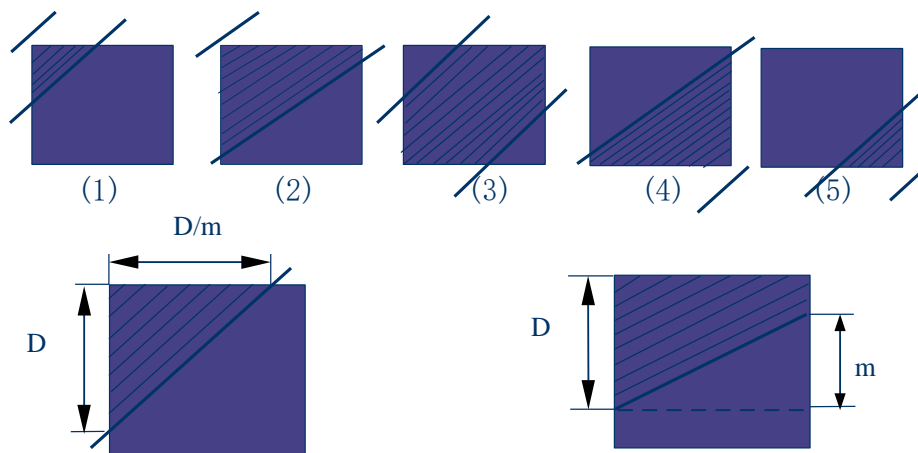
## ■ 基本思想：

- 每个像素是一个具有一定面积的小区域，将直线段看作具有一定宽度（比如一个像素单位）的狭长矩形。
- 当直线段与像素有交时，求出两者相交区域的面积，然后根据相交区域面积的大小确定该像素的亮度值。



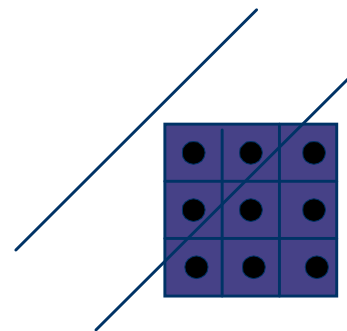
## ■ 面积计算

- 情况(1) (5)      阴影面积为:  $D^2/2m$ ;
- 情况(2) (4)      阴影面积为:  $D - m/2$ ;
- 情况(3)          阴影面积为:  $1 - D^2/m$



## ■ 离散方法

- 首先将屏幕像素均分成 $n$ 个子像素，
- 然后计算中心点落在直线段内的子像素的个数 $k$ 。
- 将屏幕该像素的亮度置为最大灰度值乘以相交区域面积的近似值 $k/n$ 。
- 例：  $n=9, k=3$  近似面积为 $1/3$



- 简单区域采样方法有两个缺点：
  - 像素的亮度与相交区域的面积成正比，而与相交区域落在像素内的位置无关，这仍然会导致锯齿效应。
  - 直线条上沿理想直线方向的相邻两个像素有时会有较大的灰度差。

# 加权区域取样

## ■ 基本思想：

- 使相交区域对像素亮度的贡献依赖于该区域与像素中心的距离
- 当直线经过该像素时，该像素的亮度**F**是在两者相交区域**A'**上对滤波器（函数**w**）进行积分的积分值。

$$w(x, y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

高斯滤波器

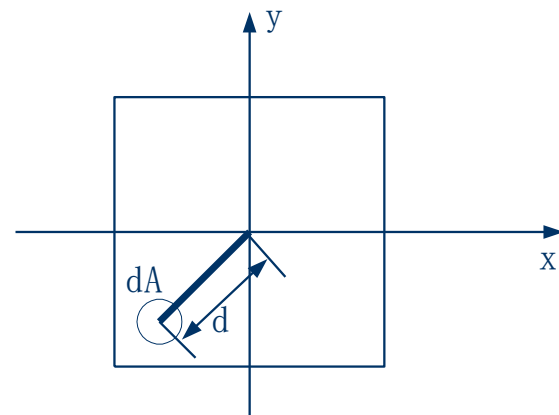
$$F = \int_{A'} w(x, y) dA$$



## ■ 离散方法

— 将屏幕划分为 $n \times n$ 个子像素，加权表可以取作：

$$\begin{bmatrix} w1 & w2 & w3 \\ w4 & w5 & w6 \\ w7 & w8 & w9 \end{bmatrix} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$



- 然后求出所有中心落于直线段内的子像素。
- 最后计算所有这些子像素对原像素亮度贡献之和  $\sum w_i$  乘以像素的最大灰度值作为该像素的显示灰度值。