属性、字符处理、反走样

ray@mail.buct.edu.cn

2017/10/18

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
    - 优点：
      - 算法简单、效率高
    - 缺点：
      - 线的始末端是水平或垂直的
      - 斜率不同，线的宽度不同
  - 方形刷
    - 方形中心对准单象素宽的线条上各个象素
    - 存在重复写象素的问题
  - 区域填充方法

- 圆弧线宽的处理
  - 线刷
  - 方形刷
  - 区域填充方法

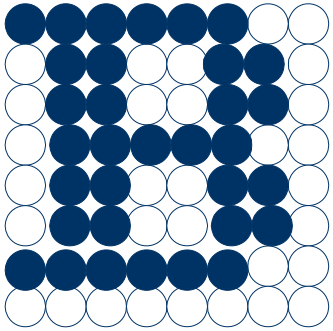
线型处理

- 线型
  - 实线、虚线、点化线
  - 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);

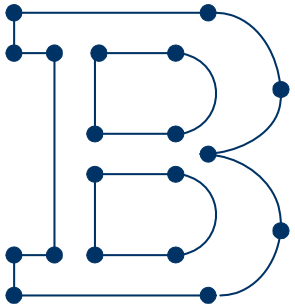
字符处理

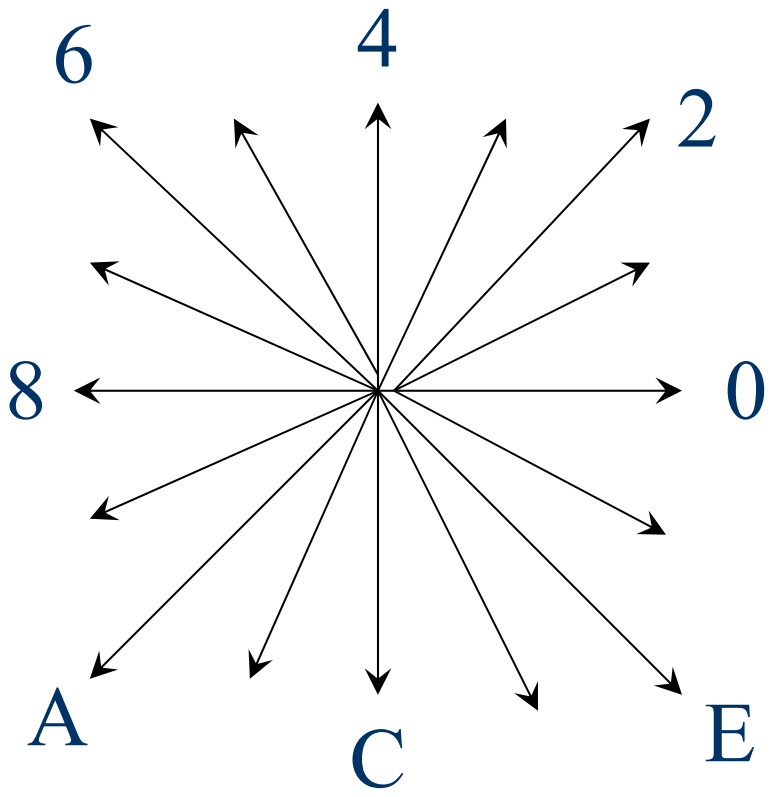
- 什么是字符？
  - 字符指数字、字母、汉字等符号。
  - 计算机中字符由一个数字编码唯一标识。
- ASCII
  - “美国信息交换用标准代码集”简称ASCII码。它是用7位二进制数进行编码表示128个字符。
- 汉字编码
  - 汉字编码的国家标准字符集。每个符号由一个区码和一个位码（2字节）共同标识。

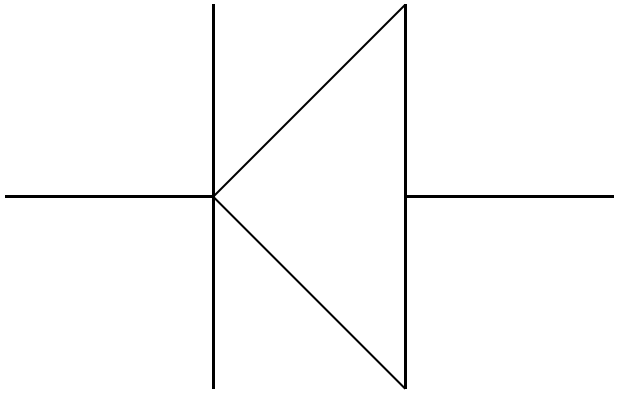
- 表示方法
  - 点阵字符
    - 每个字符由一个位图表示
  - 矢量字符
    - 记录字符的笔画信息



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







040 044 04c 042 04c 040 048 04c 046 04c

- 特点：
  - 点阵字符
    - 存储量大，易于显示。
  - 矢量字符
    - 存储量小，美观，变换方便，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

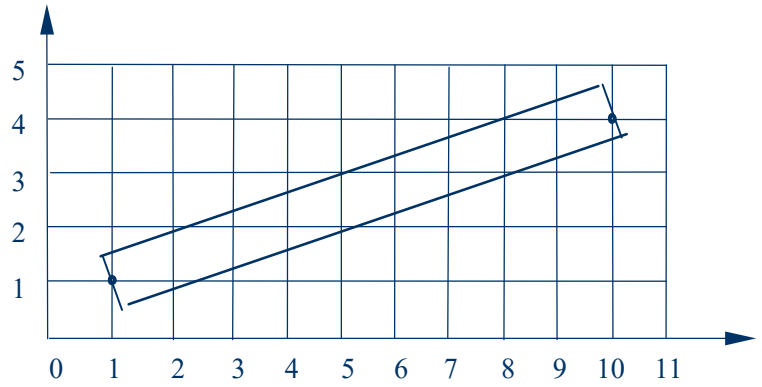
- 走样
  - 用离散量表示连续量引起的失真现象称之为走样(aliasing)
- 反走样
  - 用于减少或消除这种效果的技术称为反走样(antialiasing)

反走样方法

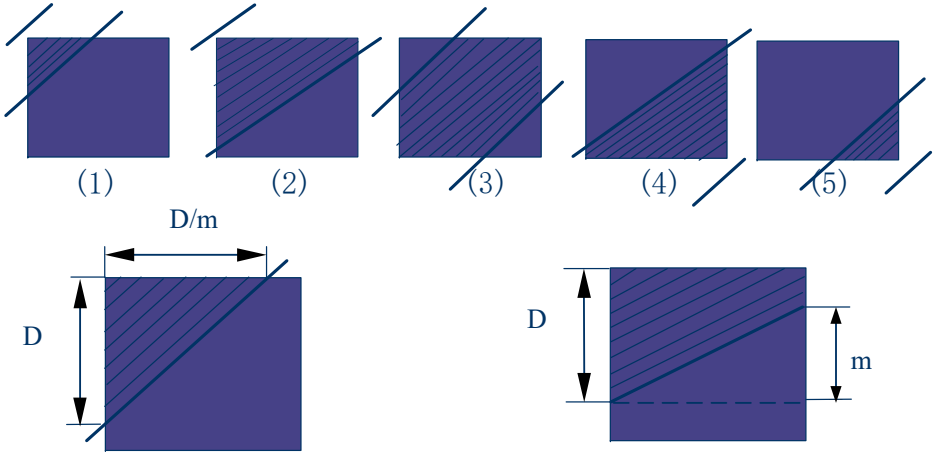
- 提高分辨率
  - 把显示器分辨率提高，显示出的直线段看起来就平直光滑了一些。
  - 阶梯的宽度减小了一倍。
  - 增加分辨率虽然简单，但是不经济的方法，而且它也只能减轻而不能消除锯齿问题

区域采样

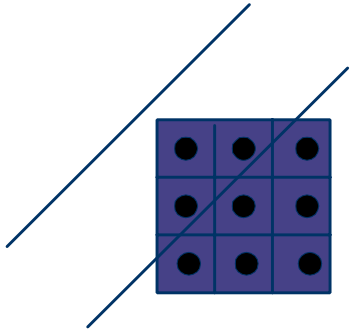
- 基本思想：
  - 每个象素是一个具有一定面积的小区域，将直线段看作具有一定宽度的狭长矩形。当直线段与象素有交时，求出两者相交区域的面积，然后根据相交区域面积的大小确定该象素的亮度值。



- 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
  - 首先将屏幕象素均分成n个子象素，
  - 然后计算中心点落在直线段内的子象素的个数k。
  - 将屏幕该象素的亮度置为相交区域面积的近似值可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：
  - 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。
  - 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。

加权区域取样

- 基本思想：
  - 使相交区域对象素亮度的贡献依赖于该区域与象素中心的距离
  - 当直线经过该象素时，该象素的亮度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。

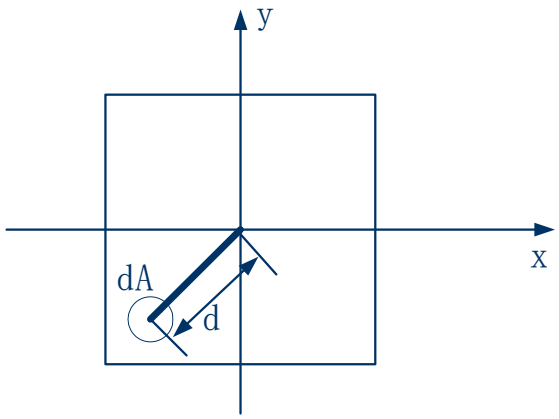
$$w(x,y)=\frac{1}{\sqrt{2\pi\sigma}}e^{-\frac{x^2+y^2}{2\sigma^2}}$$

高斯滤波器

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

- 离散方法
  - 将屏幕划分为n× n个子象素，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象素。
- 最后计算所有这些子象素对原象素亮度贡献之和$\sum w_i$乘以象素的最大灰度值作为该象素的显示灰度值。