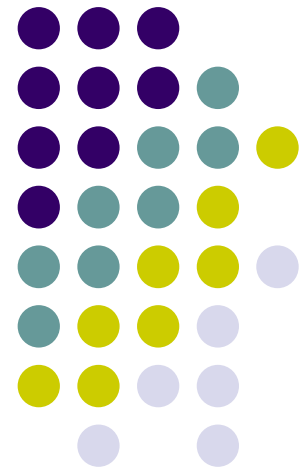


数字图像处理

第六讲 图像分割



提纲



- 基础知识
- 点、线、边缘检测
 - 背景知识
 - 孤立点的检测
 - 线检测
 - 边缘模型
 - 基本边缘检测
 - 高级边缘检测
- 边缘连接和边界检测

引言



- 图像分割
 - 把图像细分为构成它的区域或物体
 - 分割的粒度取决于应用问题
- 分割是图像处理最困难问题之一
 - 分割的精度决定了处理任务的成败
- 分割的基本原理
 - 灰度的不连续性：根据灰度的突变分割
 - 灰度的相似性：区域内的图像很相似

基础知识



- R 表示图像所占的区域
- 图像分割将 R 分割成 n 个区域 R_1, \dots, R_n :
 1. $\bigcup_{i=1}^n R_i = R$
 2. R_i 是一个连通集合, $i = 1, \dots, n$
 3. $R_i \cap R_j = \emptyset, \forall i \neq j$
 4. $Q(R_i) = True, i = 1, \dots, n$
 5. 对于任意的相邻区域 R_i 和 R_j , $Q(R_i \cup R_j) = False$
 - $Q(\cdot)$ 表示某个用于划分区域的函数

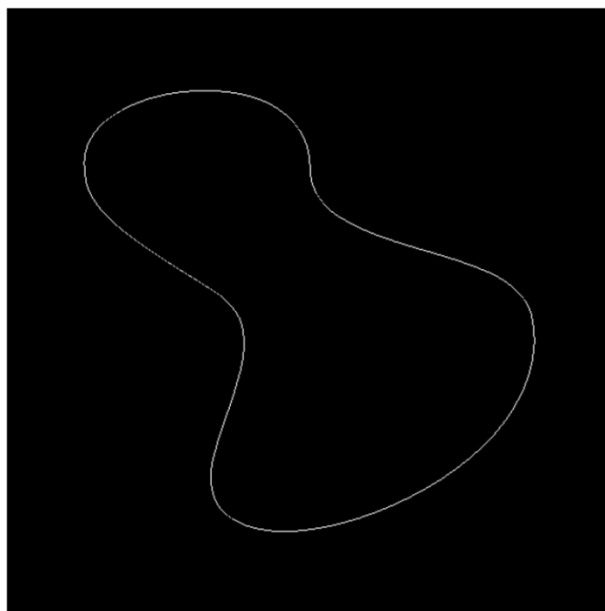
灰度的不连续性



- 区域的边界之间差异大，且和背景不同
 - 利用灰度的局部不连续性检测边界
- 基于边缘的分割



原图



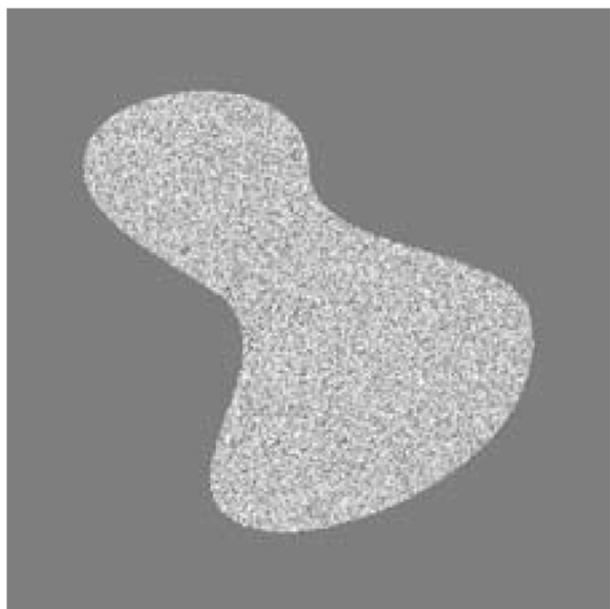
计算边界



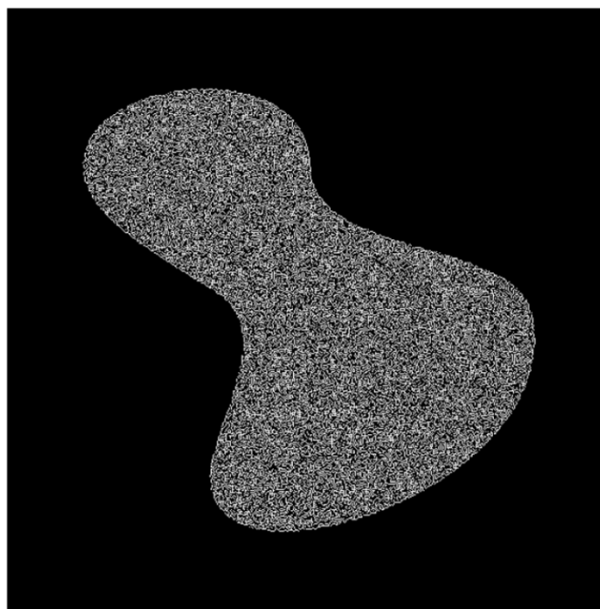
分割

灰度的相似性

- 基于区域的分割
 - 区域内的像素依据某些准则很相似



原图

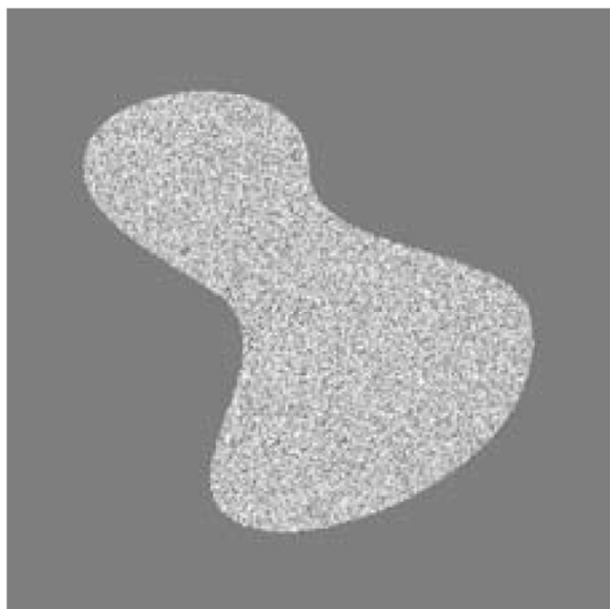


边缘信息

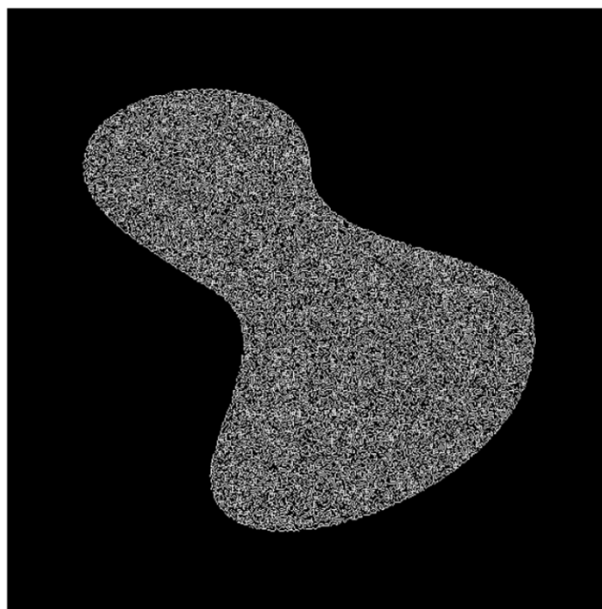
灰度的相似性

- 基于区域的分割
 - 区域内的像素依据某些准则很相似

分割成 4×4 区域，标准差非零标为白色



原图



边缘信息



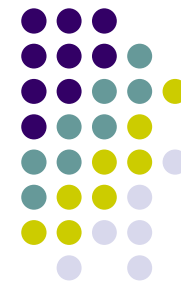
分割

提纲



- 基础知识
- 点、线、边缘检测
 - 背景知识
 - 孤立点的检测
 - 线检测
 - 边缘模型
 - 基本边缘检测
 - 高级边缘检测
- 边缘连接和边界检测

基本概念



- 边缘
 - 边缘像素：灰度发生剧烈变化
 - 边缘是连通的边缘像素集合
- 线
 - 一种特殊的边缘
 - 两侧的灰度值都很大或都很小
- 点
 - 长宽只有1个像素的线



提纲

- 基础知识
- 点、线、边缘检测
 - 背景知识
 - 孤立点的检测
 - 线检测
 - 边缘模型
 - 基本边缘检测
 - 高级边缘检测
- 边缘连接和边界检测

背景知识



- 1阶或2阶导数可以检测局部灰度突变
- 1阶导数的性质
 - 在恒定灰度区域为**零**
 - 在突变（斜坡、台阶）的起点**非零**
 - 沿着斜坡**非零**
- 2阶导数的性质
 - 在恒定灰度区域为**零**
 - 在突变（斜坡、台阶）的起点和终点**非零**
 - 沿着恒定斜率斜坡为**零**

数学基础



- 一维函数 $f(x)$

- 一阶导数

$$\frac{\partial f}{\partial x} = f'(x) = f(x + 1) - f(x)$$

- 数值是有限的
- 最小间隔1像素

- 二阶导数

$$\begin{aligned}\frac{\partial^2 f}{\partial x^2} &= \frac{\partial f'(x)}{\partial x} = f'(x + 1) - f'(x) \\ &= f(x + 2) - f(x + 1) - f(x + 1) + f(x) \\ &= f(x + 2) - 2f(x + 1) + f(x)\end{aligned}$$

数学基础



- 一维函数 $f(x)$

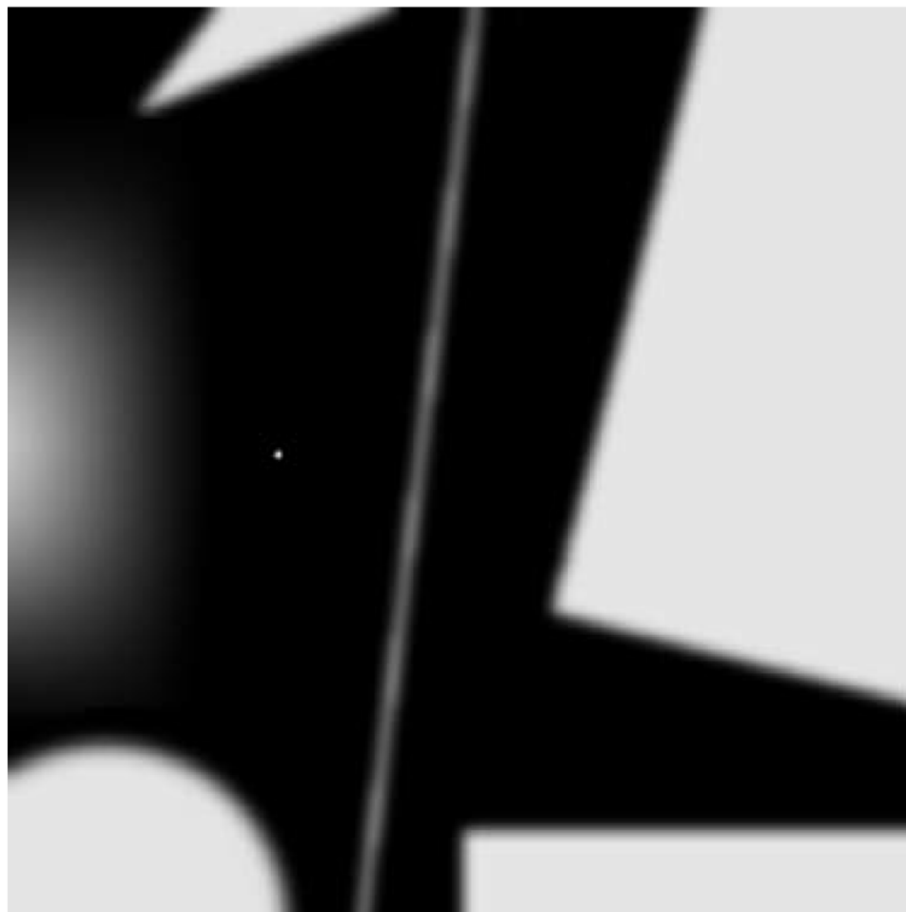
- 一阶导数

$$\frac{\partial f}{\partial x} = f'(x) = f(x + 1) - f(x)$$

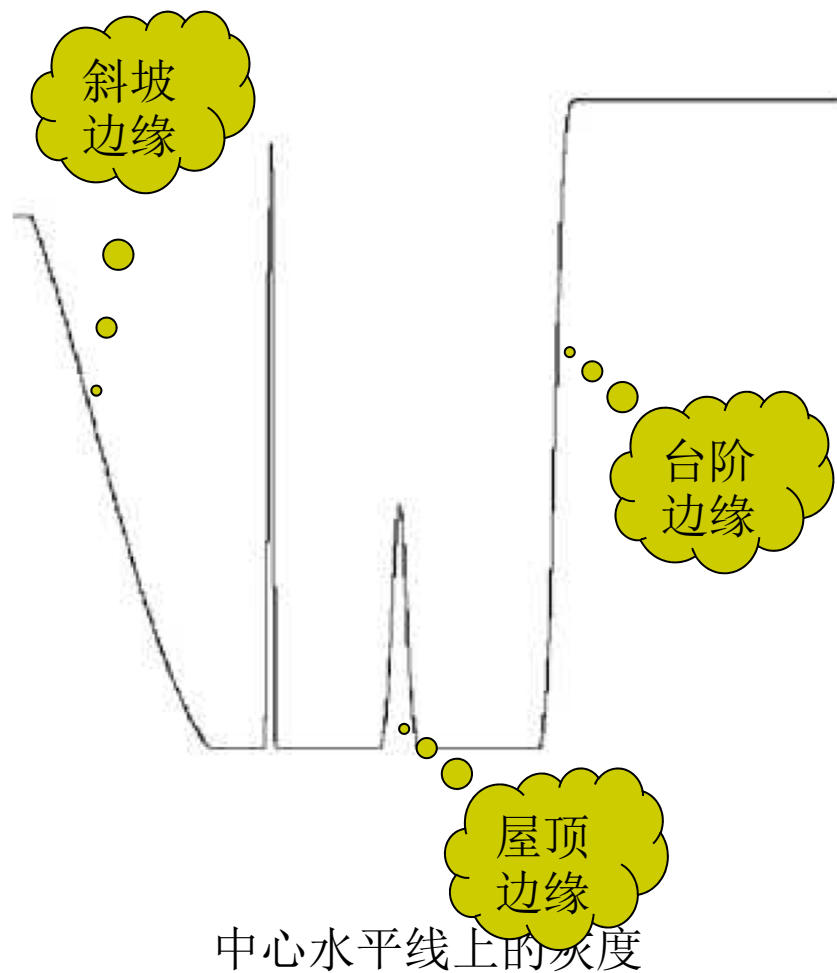
- 二阶导数

$$\frac{\partial^2 f}{\partial x^2} = f''(x) = f(x + 1) + f(x - 1) - 2f(x)$$

举例



原图



举例

一阶、二阶导数

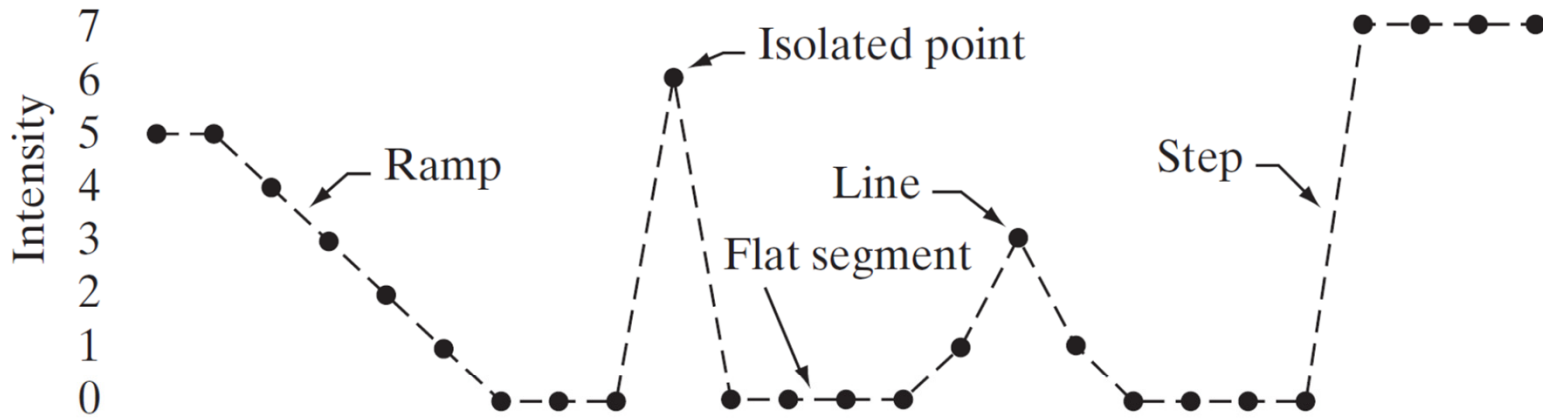
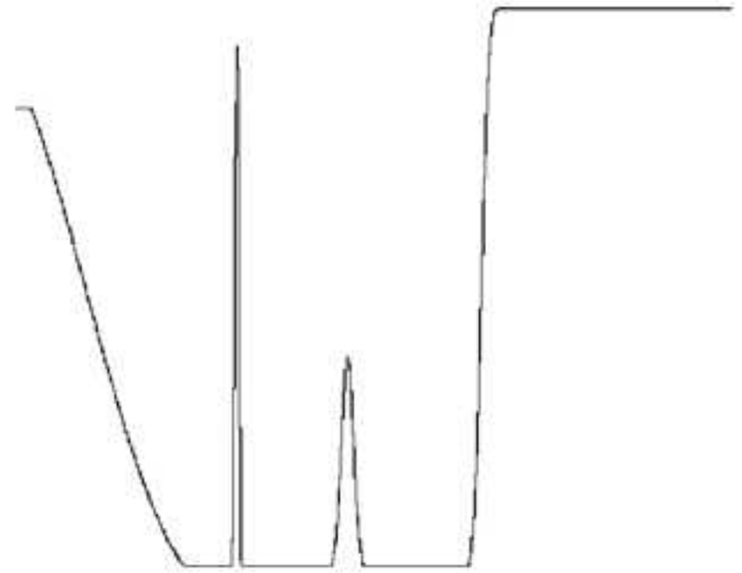


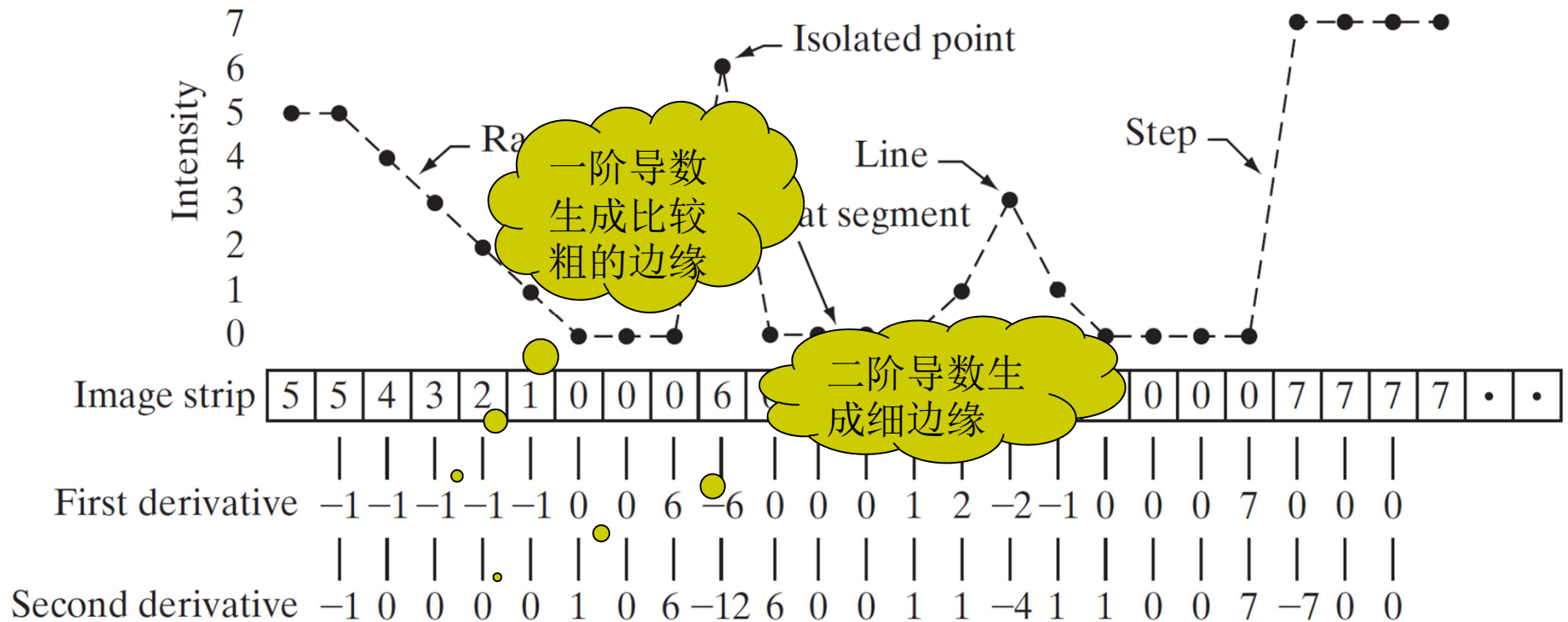
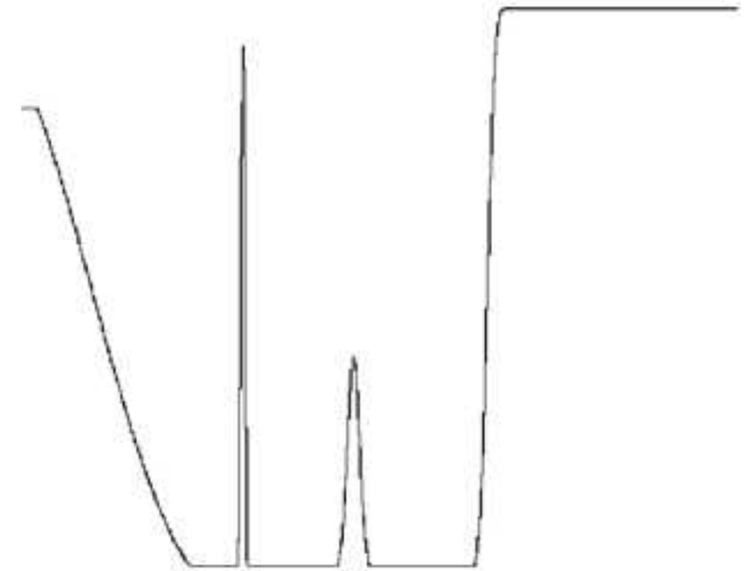
Image strip	5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	7	.	.
-------------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

First derivative	-1	-1	-1	-1	-1	0	0	6	-6	0	0	0	1	2	-2	-1	0	0	0	7	0	0	0
------------------	----	----	----	----	----	---	---	---	----	---	---	---	---	---	----	----	---	---	---	---	---	---	---

Second derivative	-1	0	0	0	0	1	0	6	-12	6	0	0	1	1	-4	1	1	0	0	7	-7	0	0
-------------------	----	---	---	---	---	---	---	---	-----	---	---	---	---	---	----	---	---	---	---	---	----	---	---

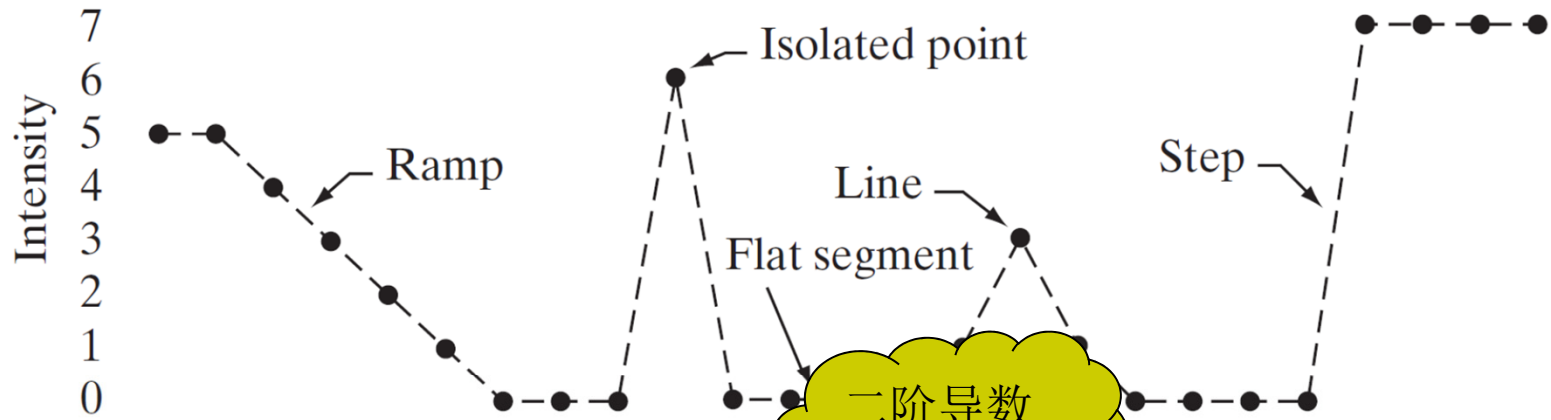
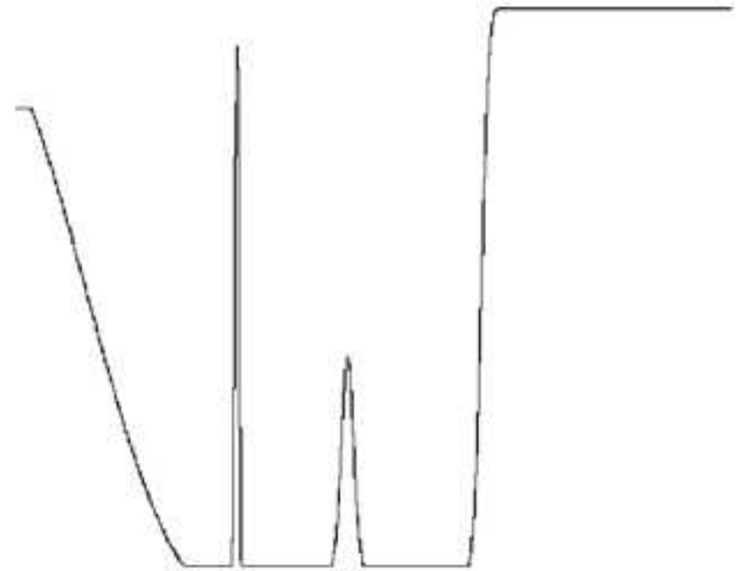
举例

一阶、二阶导数



举例

一阶、二阶导数



二阶导数的响应幅度更大

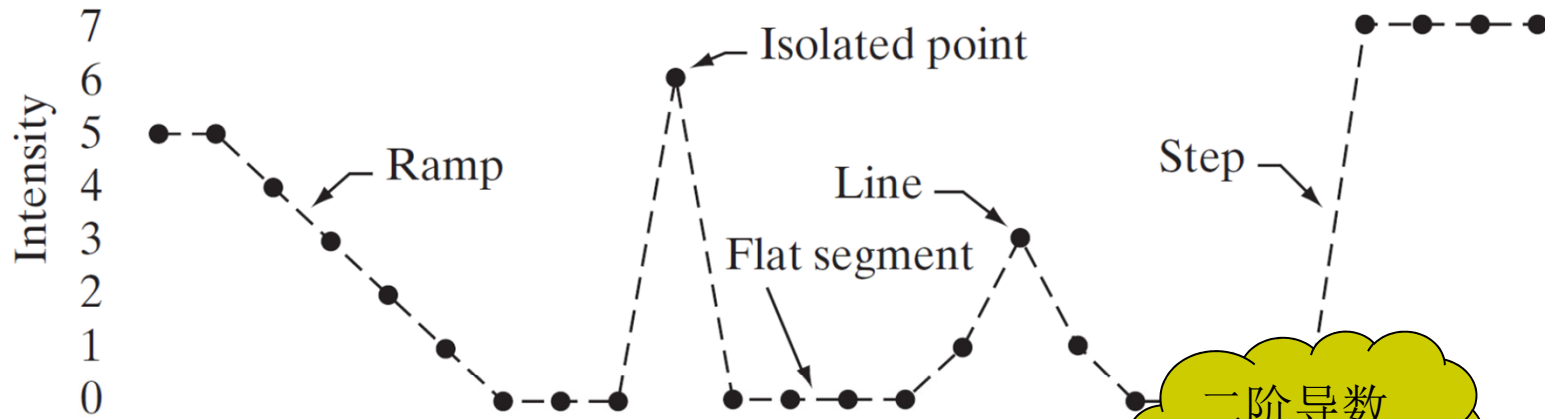
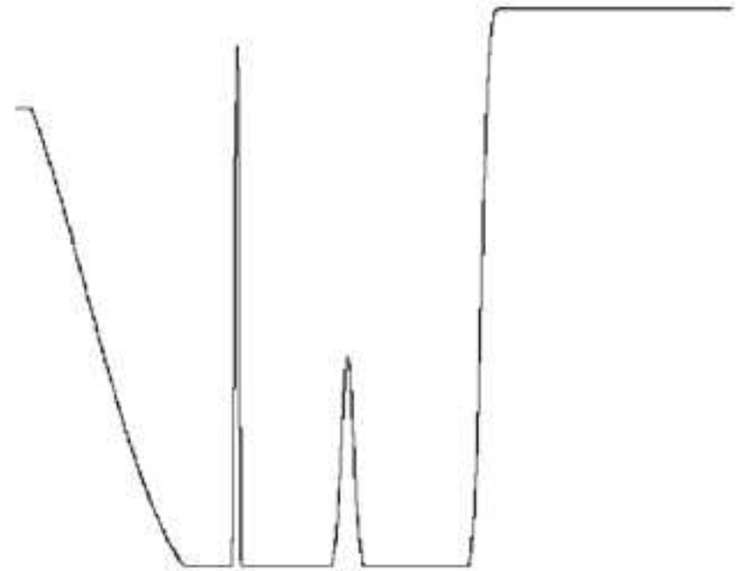
Image strip	5	5	4	3	2	1	0	0	0	6	0	0	0	0	0	0	0	0	7	7	7	7	•	•
-------------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

First derivative	-1	-1	-1	-1	-1	0	0	6	-6	0	0	1	2	-2	-1	0	0	0	7	0	0	0		
------------------	----	----	----	----	----	---	---	---	----	---	---	---	---	----	----	---	---	---	---	---	---	---	--	--

Second derivative	-1	0	0	0	0	1	0	6	-12	6	0	0	1	1	-4	1	1	0	0	7	-7	0	0	
-------------------	----	---	---	---	---	---	---	---	-----	---	---	---	---	---	----	---	---	---	---	---	----	---	---	--

举例

一阶、二阶导数



二阶导数的响应幅度更大

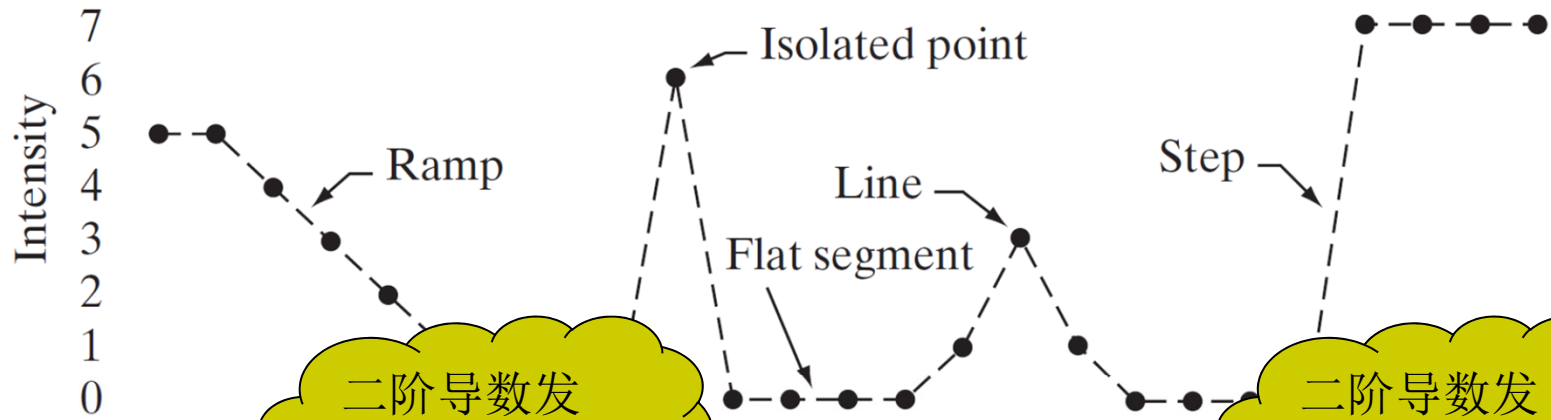
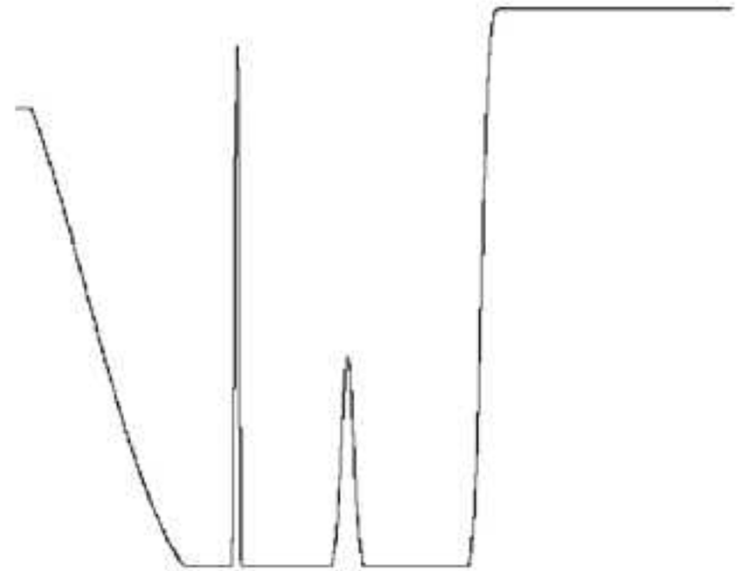
Image strip	5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	7	7	•	•
-------------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

First derivative	-1	-1	-1	-1	-1	0	0	6	-6	0	0	0	1	2	-2	-1	0	0	0	7	0	0	0
------------------	----	----	----	----	----	---	---	---	----	---	---	---	---	---	----	----	---	---	---	---	---	---	---

Second derivative	-1	0	0	0	0	1	0	6	-12	6	0	0	1	1	-4	1	1	0	0	7	-7	0	0
-------------------	----	---	---	---	---	---	---	---	-----	---	---	---	---	---	----	---	---	---	---	---	----	---	---

举例

一阶、二阶导数



二阶导数发生符号变化 (亮到暗)

二阶导数发生符号变化 (暗到亮)

Image strip	5	5	4	3	2	0	0	0	0	1	3	1	0	0	0	0	0	7	7	7	7
-------------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

First derivative	-1	-1	-1	-1	0	0	6	-6	0	0	0	1	2	-2	-1	0	0	0	7	0	0
------------------	----	----	----	----	---	---	---	----	---	---	---	---	---	----	----	---	---	---	---	---	---

Second derivative	-1	0	0	0	0	1	0	6	-12	6	0	0	1	1	-4	1	1	0	0	7	-7	0	0
-------------------	----	---	---	---	---	---	---	---	-----	---	---	---	---	---	----	---	---	---	---	---	----	---	---



一般结论

- 一阶导数通常产生较粗的边缘
- 二阶导数对细节有较强的响应
 - 细线、孤立点、噪声
- 二阶导数在斜坡和台阶产生双边缘响应
- 二阶导数的符号变化有指示意义
 - 灰度从亮到暗
 - 灰度从暗到亮



计算导数

- 空间滤波器
 - 模板

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

- 计算公式

$$\begin{aligned} R &= w_1z_1 + w_2z_2 + \cdots + w_9z_9 \\ &= \sum_{k=1}^9 w_k z_k \end{aligned}$$



提纲

- 基础知识
- 点、线、边缘检测
 - 背景知识
 - 孤立点的检测
 - 线检测
 - 边缘模型
 - 基本边缘检测
 - 高级边缘检测
- 边缘连接和边界检测



孤立点的检测

- 利用二阶导数检测孤立点
 - 响应更强
- 拉普拉斯算子

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

- 离散拉普拉斯算子

$$\frac{\partial^2 f(x, y)}{\partial x^2} = f(x + 1, y) + f(x - 1, y) - 2f(x, y)$$

$$\frac{\partial^2 f(x, y)}{\partial y^2} = f(x, y + 1) + f(x, y - 1) - 2f(x, y)$$



孤立点的检测

- 利用二阶导数检测孤立点
 - 响应更强
- 拉普拉斯算子

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

- 离散拉普拉斯算子

$$\begin{aligned} \nabla^2 f(x, y) = & f(x + 1, y) + f(x - 1, y) + f(x, y + 1) \\ & + f(x, y - 1) - 4f(x, y) \end{aligned}$$

拉普拉斯算子



- 标准形式

$$\begin{aligned}\nabla^2 f(x, y) &= f(x + 1, y) + f(x - 1, y) \\ &\quad + f(x, y + 1) + f(x, y - 1) \\ &\quad - 4f(x, y)\end{aligned}$$

0	1	0
1	-4	1
0	1	0

90度增量
各向同性

- 对角线形式

1	1	1
1	-8	1
1	1	1

45度增量
各向同性

检测方法



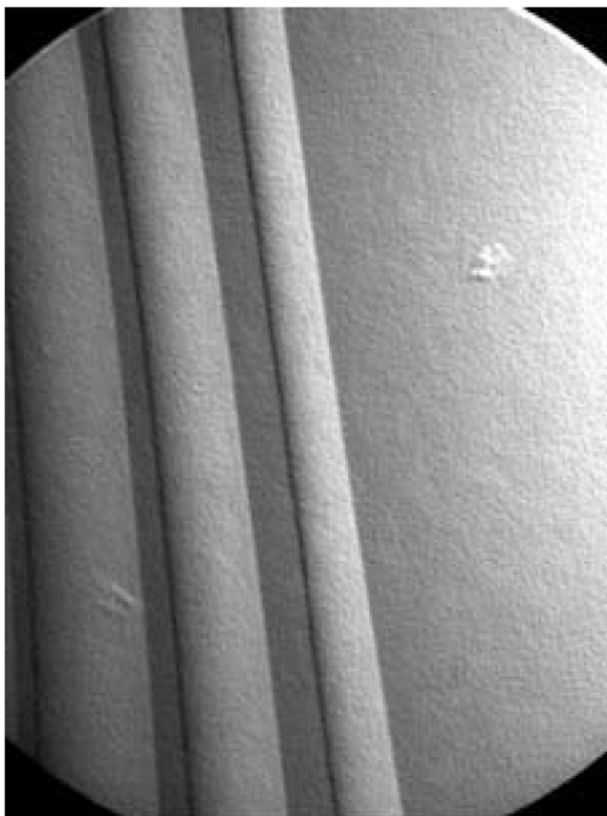
- 根据响应幅度是否大于某阈值 T

$$g(x, y) = \begin{cases} 1 & \text{if } |R(x, y)| \geq T \\ 0 & \text{otherwise} \end{cases}$$

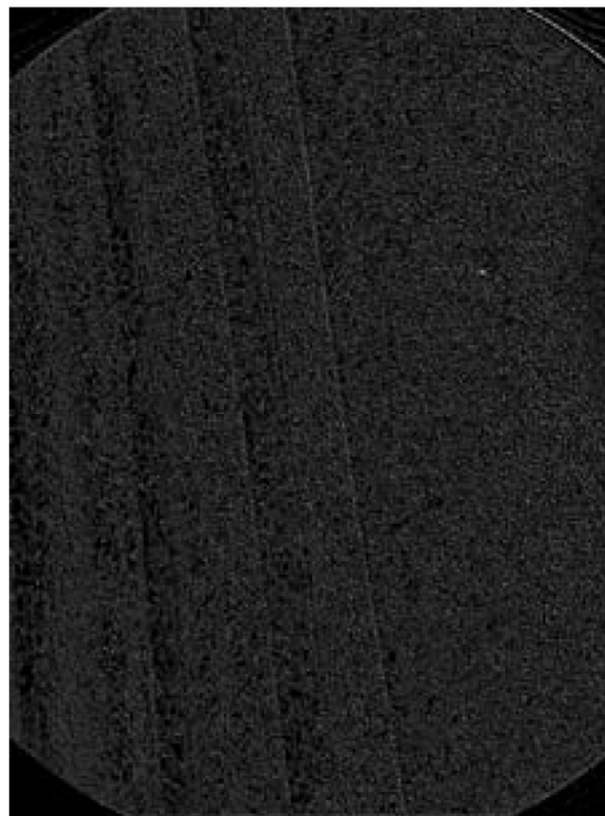
- 其中

$$\begin{aligned} R &= w_1z_1 + w_2z_2 + \cdots + w_9z_9 \\ &= \sum_{k=1}^9 w_kz_k \end{aligned}$$

举例

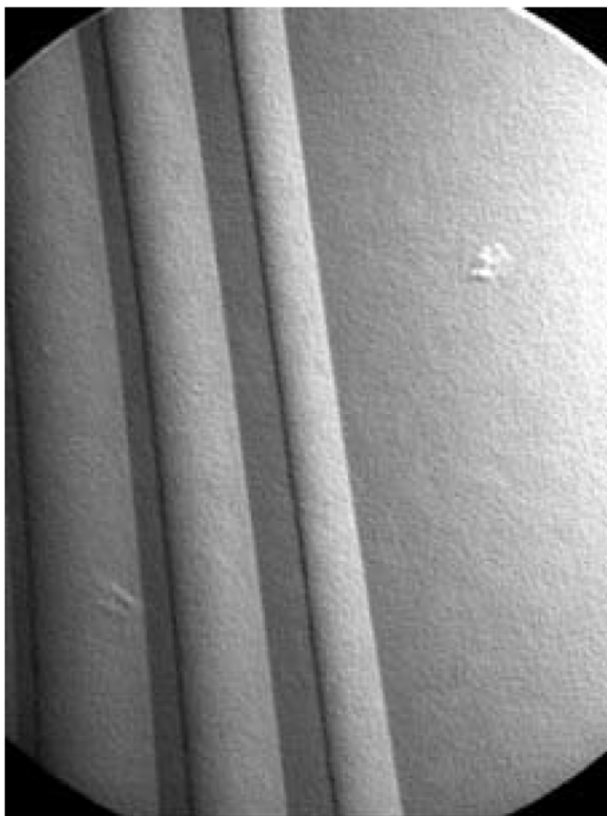


有一个孔的
涡轮叶片

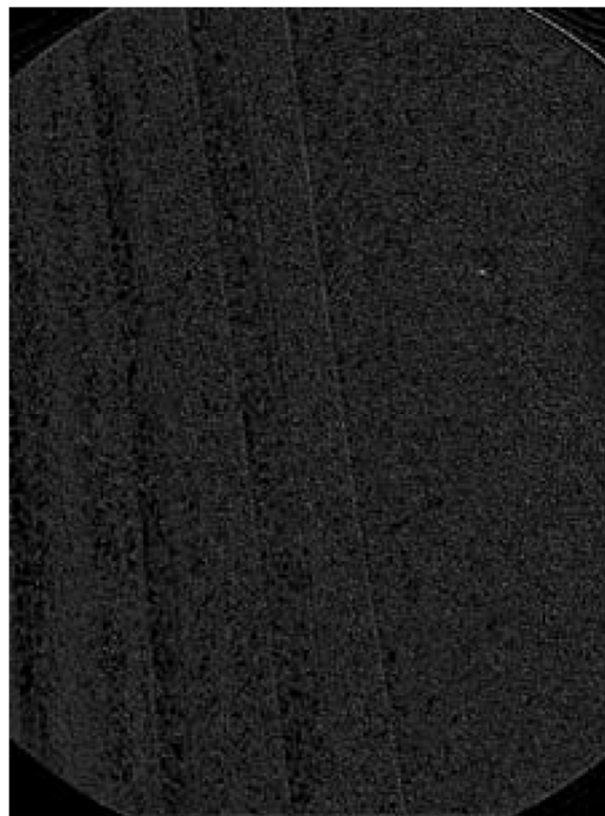


拉普拉斯滤波
后图像

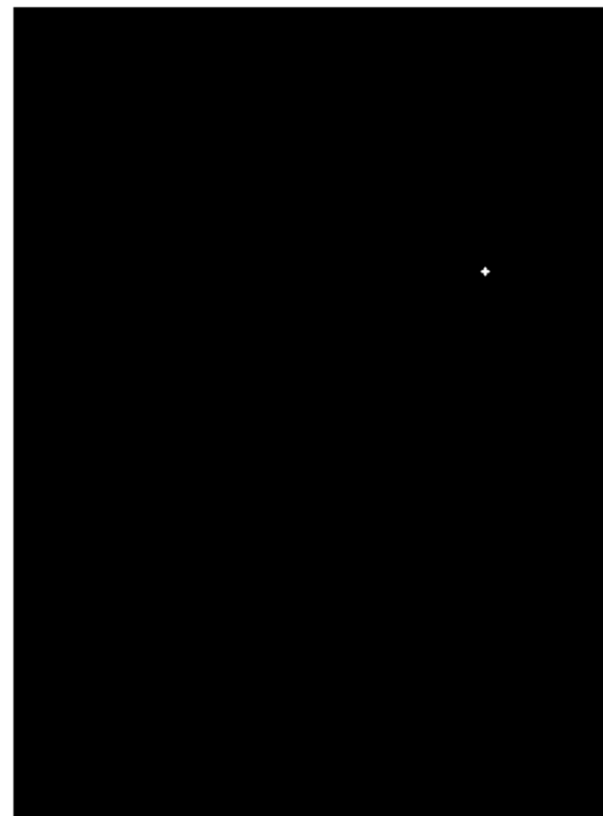
举例



有一个孔的
涡轮叶片



拉普拉斯滤波
后图像



阈值化
(最高亮度的90%)



提纲

- 基础知识
- 点、线、边缘检测
 - 背景知识
 - 孤立点的检测
 - 线检测
 - 边缘模型
 - 基本边缘检测
 - 高级边缘检测
- 边缘连接和边界检测



线检测

- 利用二阶导数检测线
 - 响应更强、更细的线
 - 需要留意双线效应

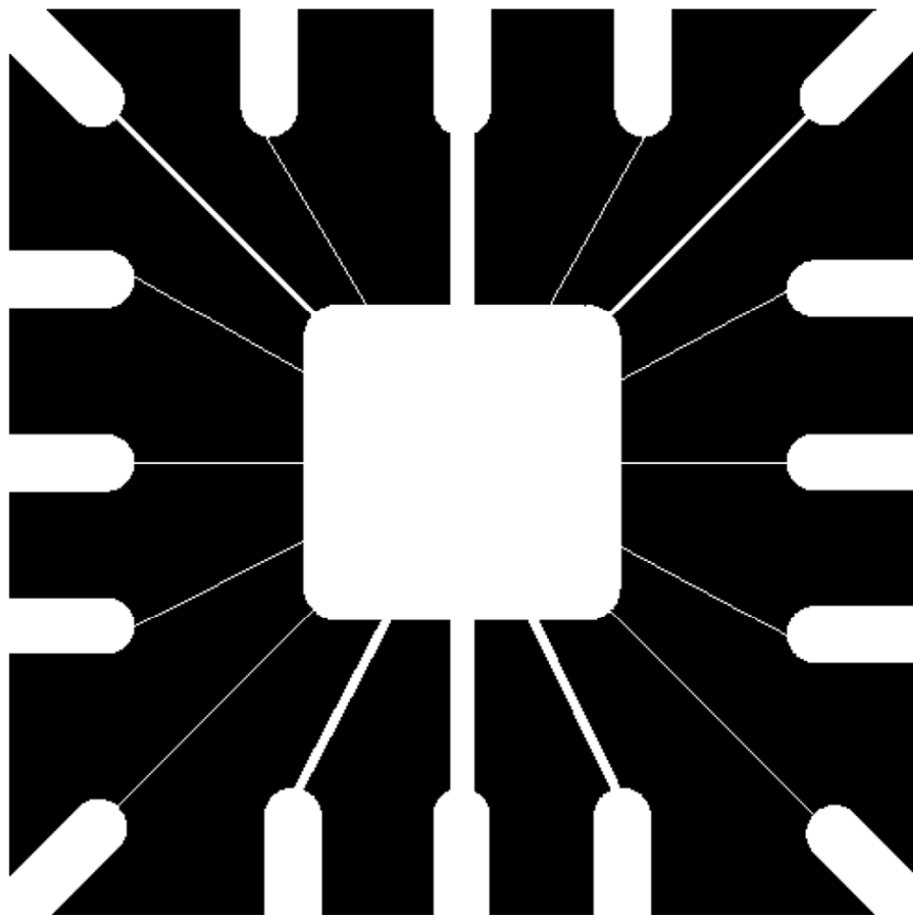
- 拉普拉斯算子

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

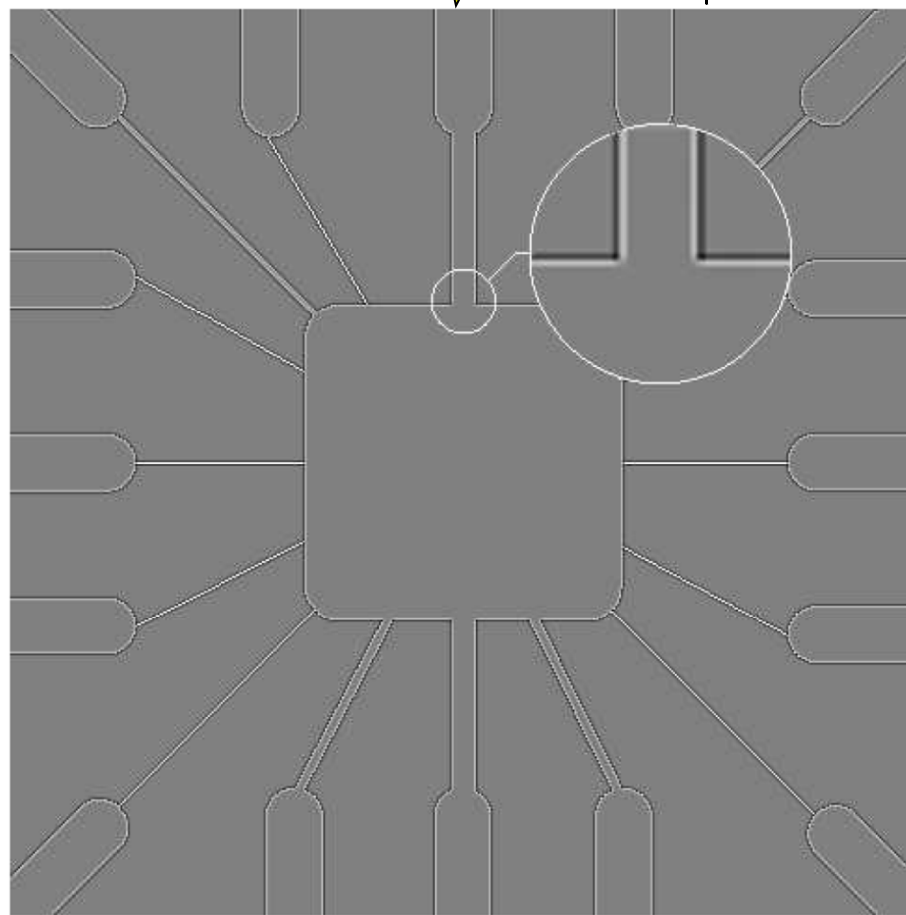
- 离散拉普拉斯算子

$$\begin{aligned} \nabla^2 f(x, y) = & f(x + 1, y) + f(x - 1, y) + f(x, y + 1) \\ & + f(x, y - 1) - 4f(x, y) \end{aligned}$$

举例

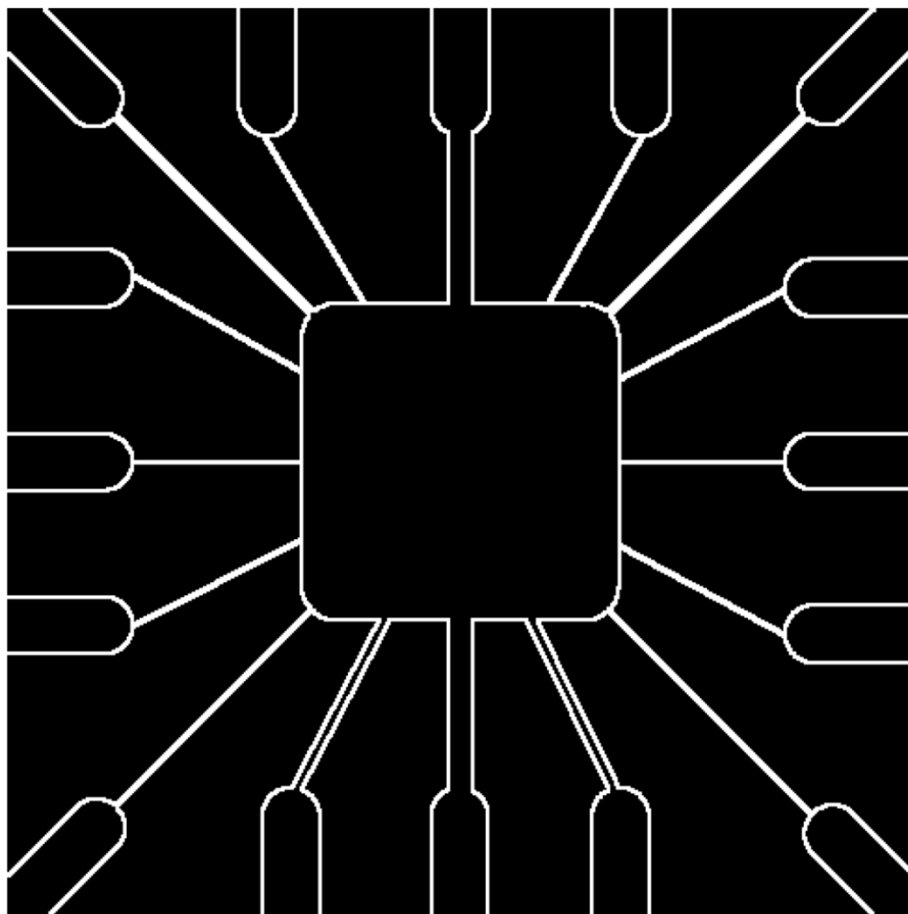


原图



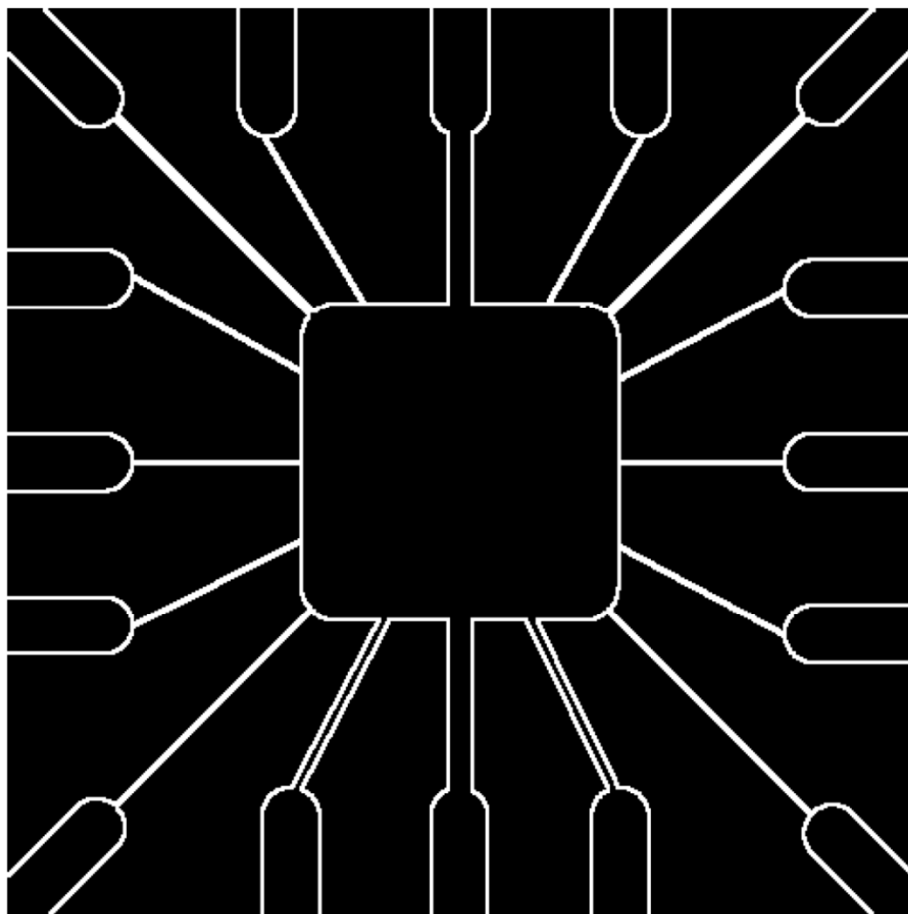
拉普拉斯滤波
后图像

举例

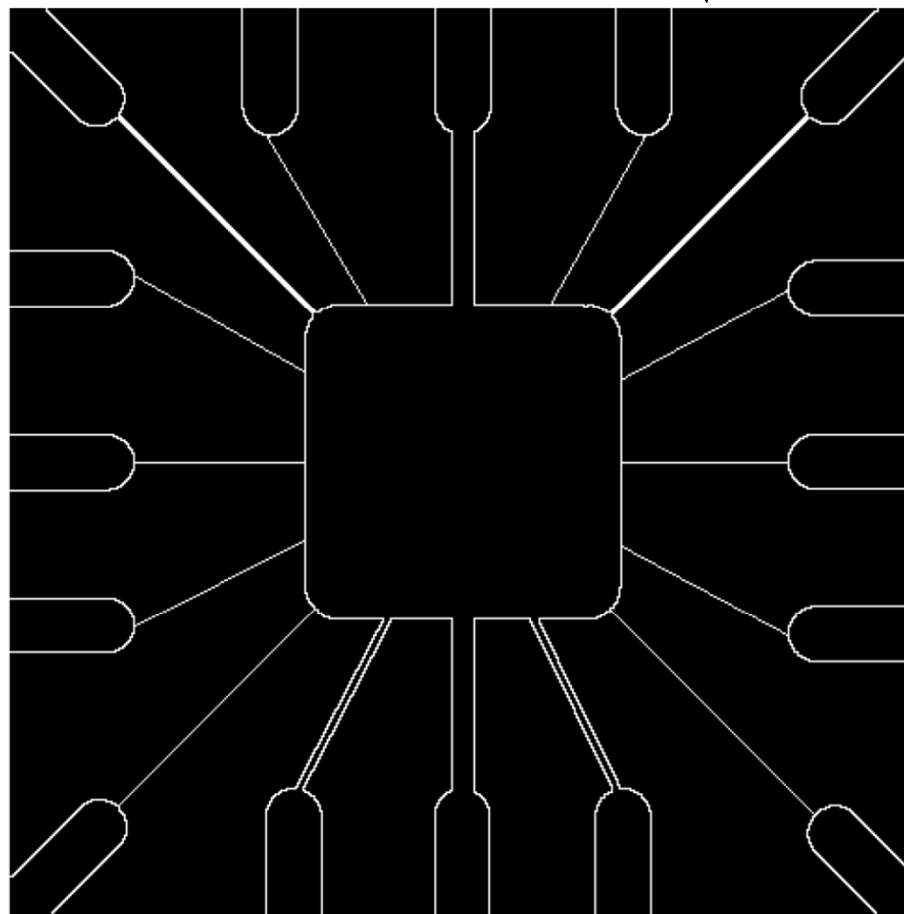


取绝对值

举例



取绝对值

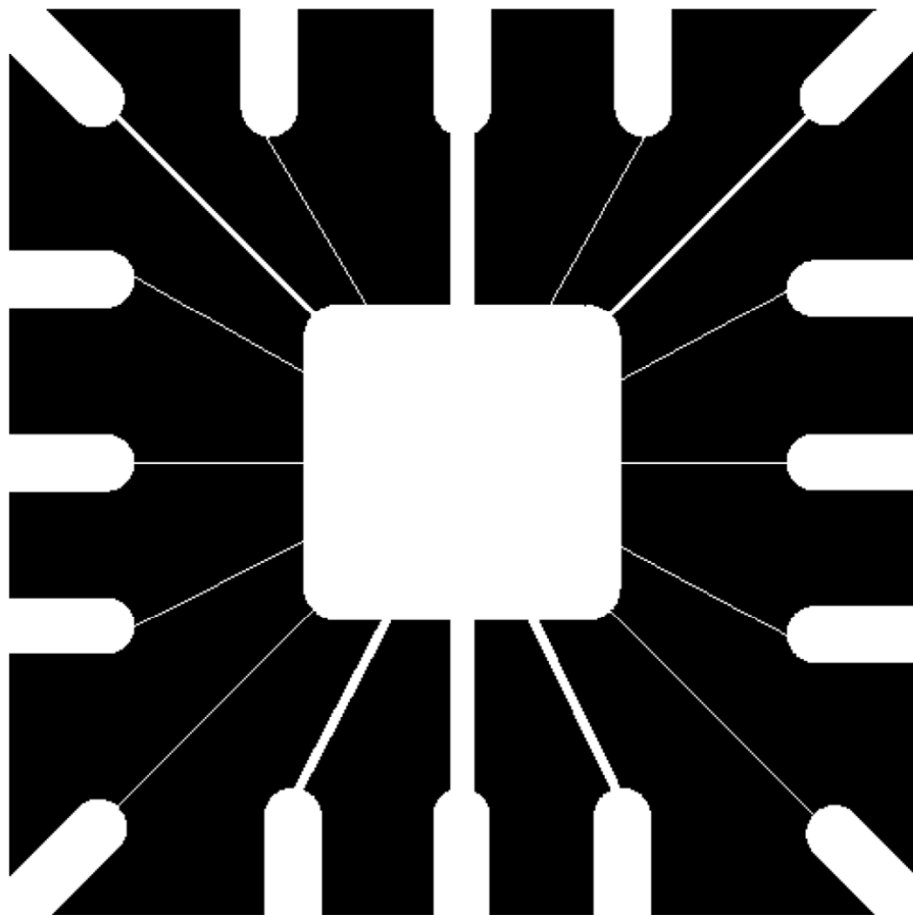


保留正数

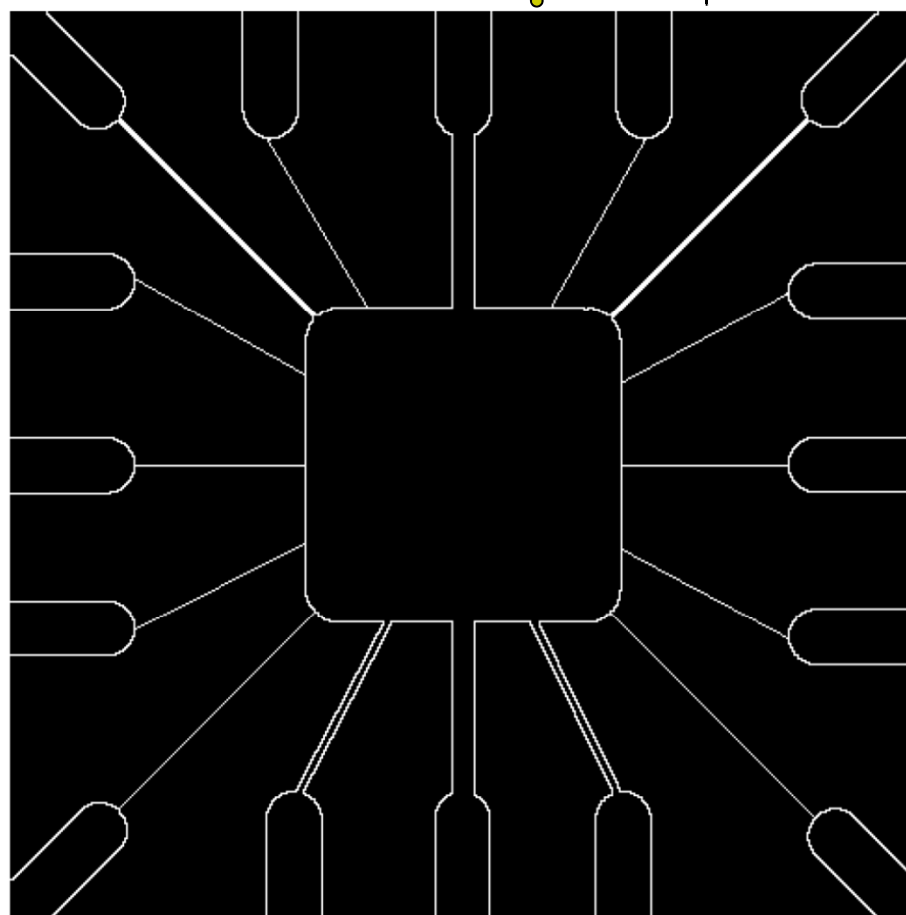
举例

宽线中间产生0值山谷

宽线最好被当做区域



原图



保留正数

检测特定方向的线



检测
水平线

-1	-1	-1	2	-1	-1
2	2	2	-1	2	-1
-1	-1	-1	-1	-1	2

检测
+45°线

检测
垂直线

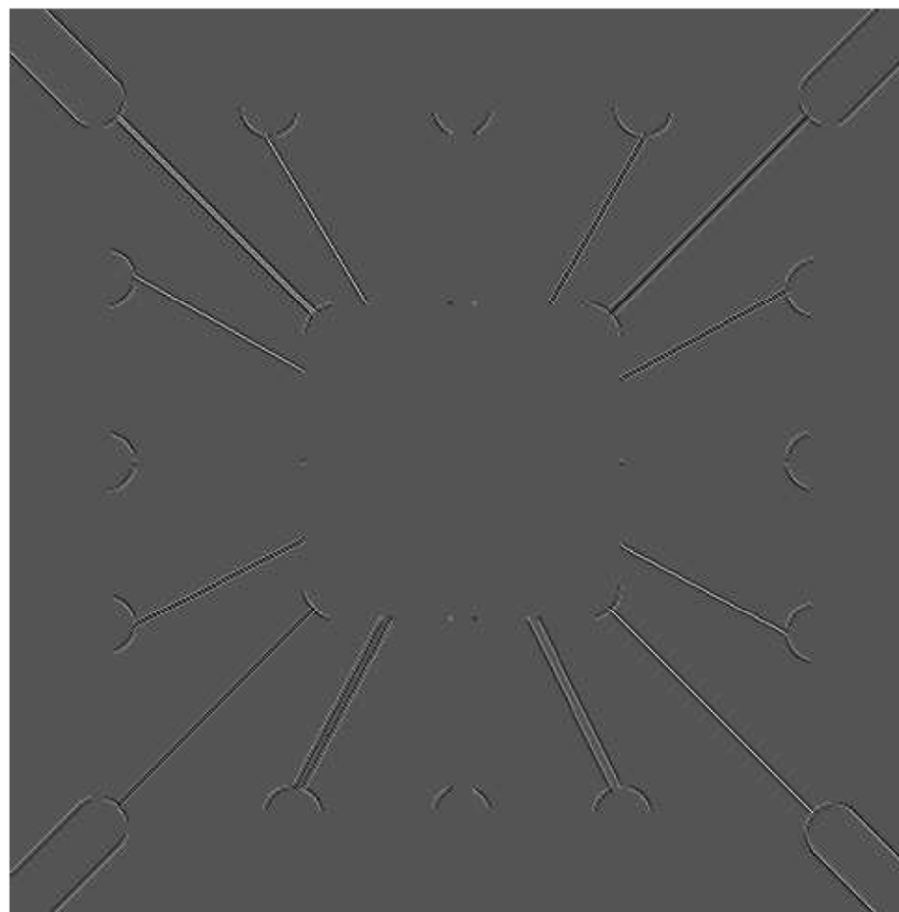
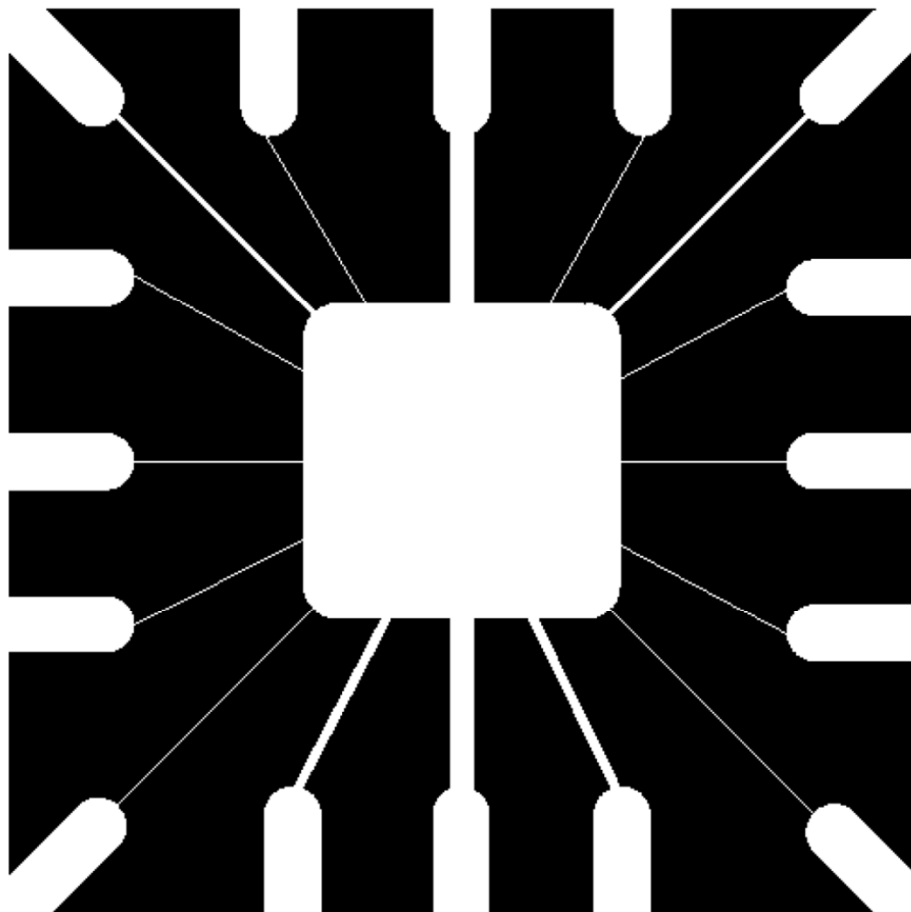
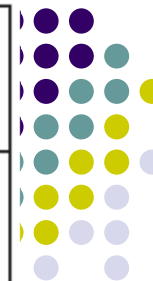
-1	2	-1	-1	-1	2
-1	2	-1	-1	2	-1
-1	2	-1	2	-1	-1

检测
-45°线

举例

检测
+45°线

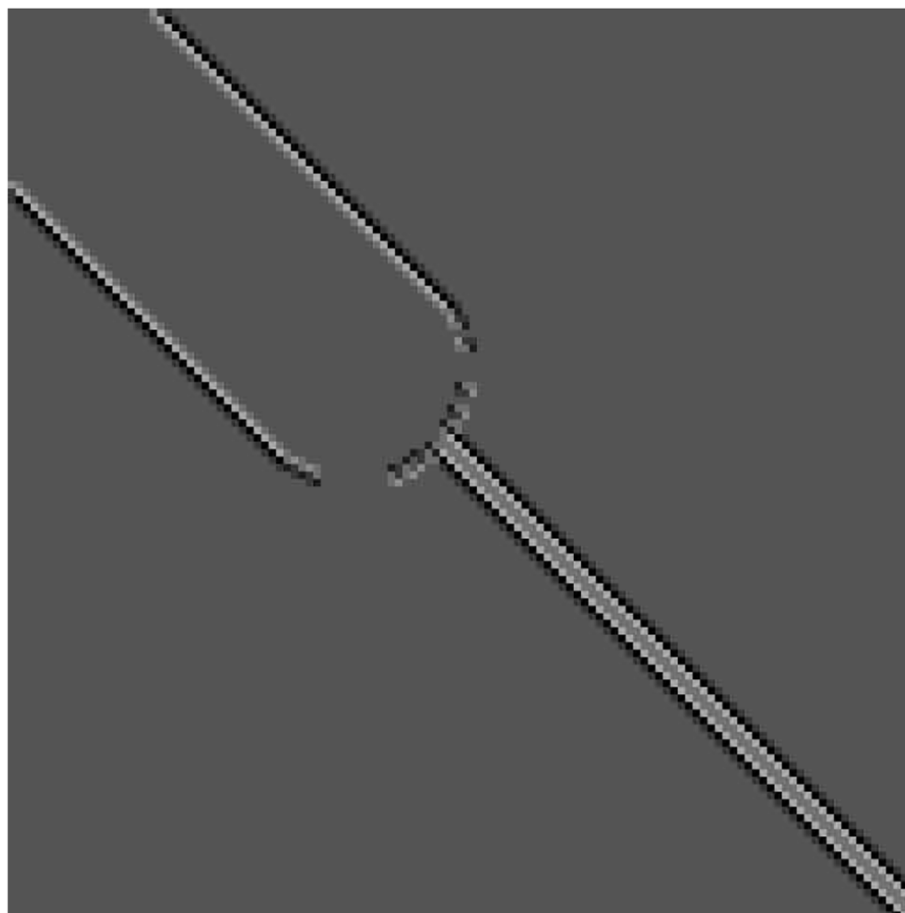
2	-1	-1
-1	2	-1
-1	-1	2



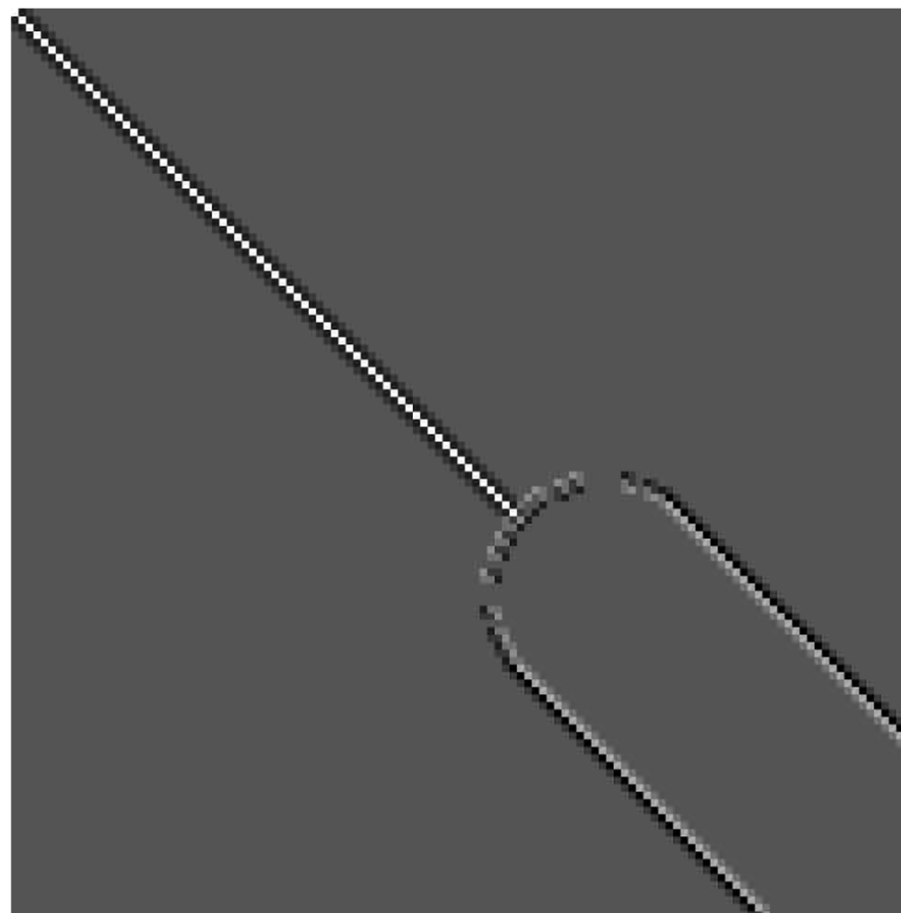
举例



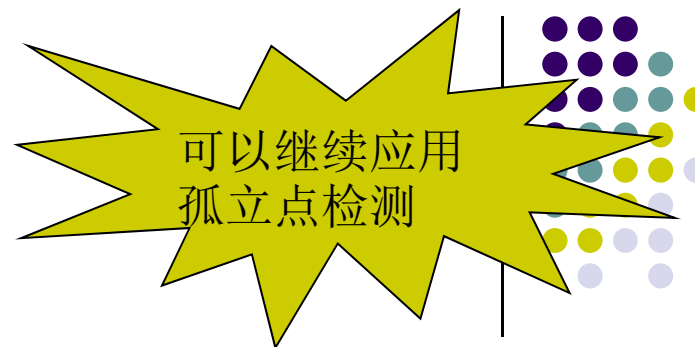
左上角放大的图



右下角放大的图

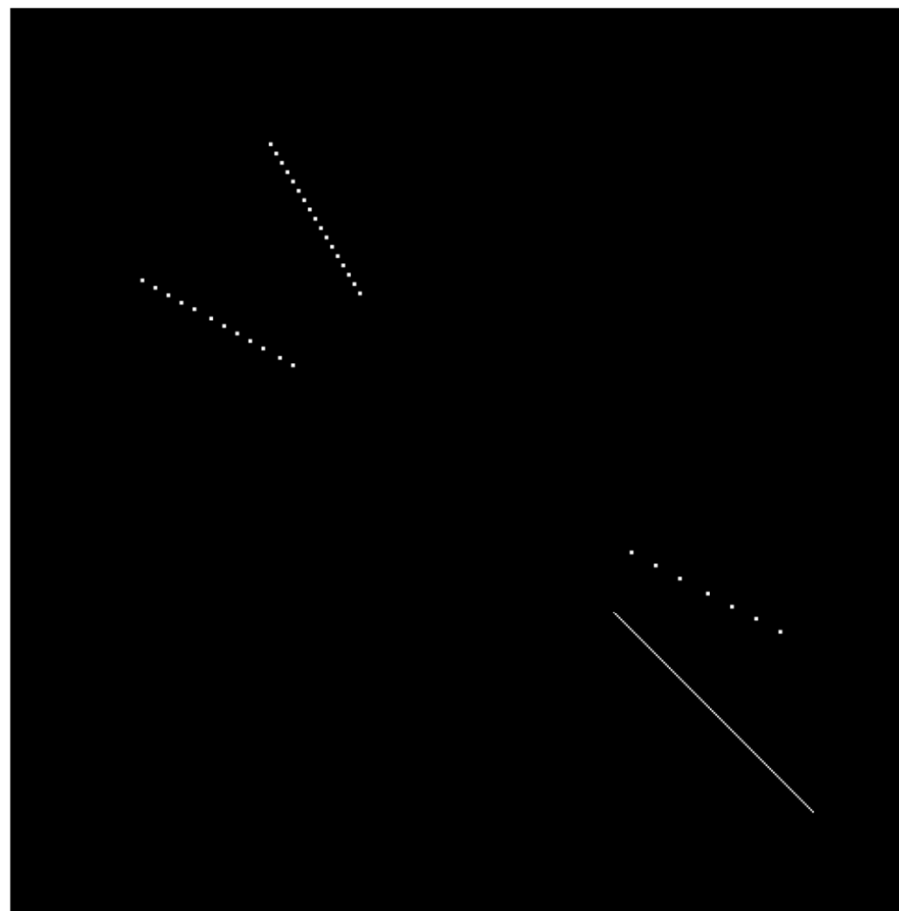
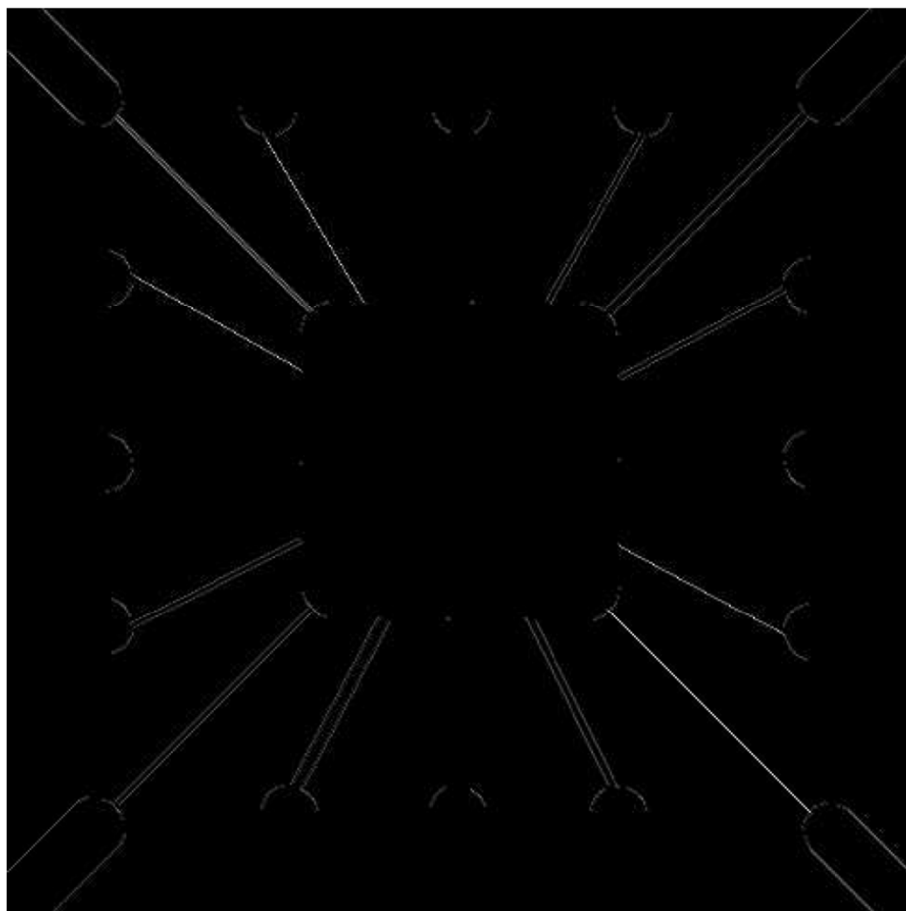


举例



保留正数

保留最大值



提纲



- 基础知识
- 点、线、边缘检测
 - 背景知识
 - 孤立点的检测
 - 线检测
 - 边缘模型
 - 基本边缘检测
 - 高级边缘检测
- 边缘连接和边界检测

边缘模型



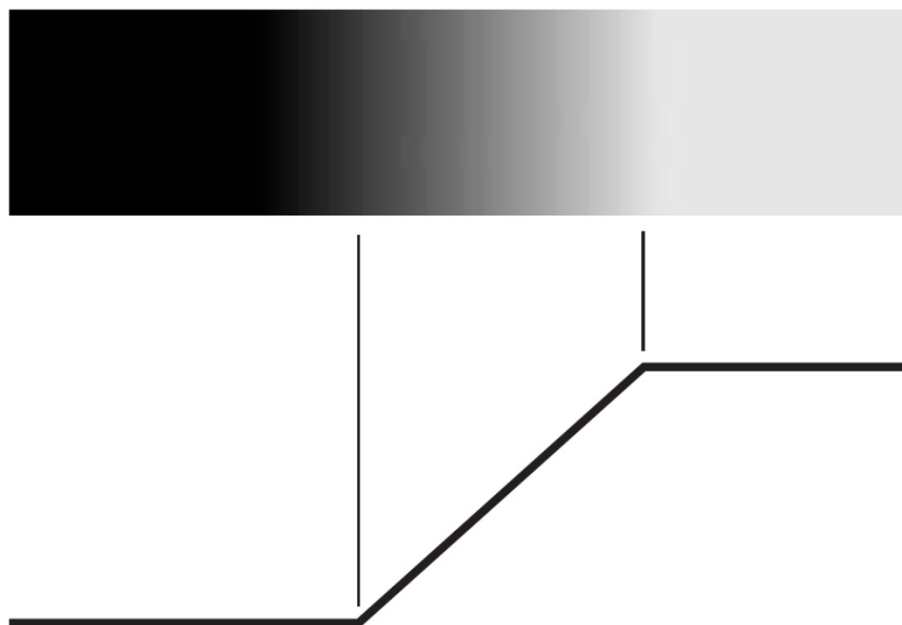
- 台阶边缘 (Step Edge)
 - 1个像素距离上发生灰度级的理想过渡
 - 经常出现在计算机生成的图像中



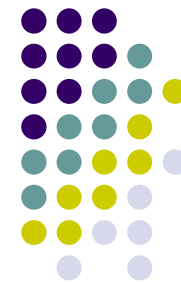
边缘模型



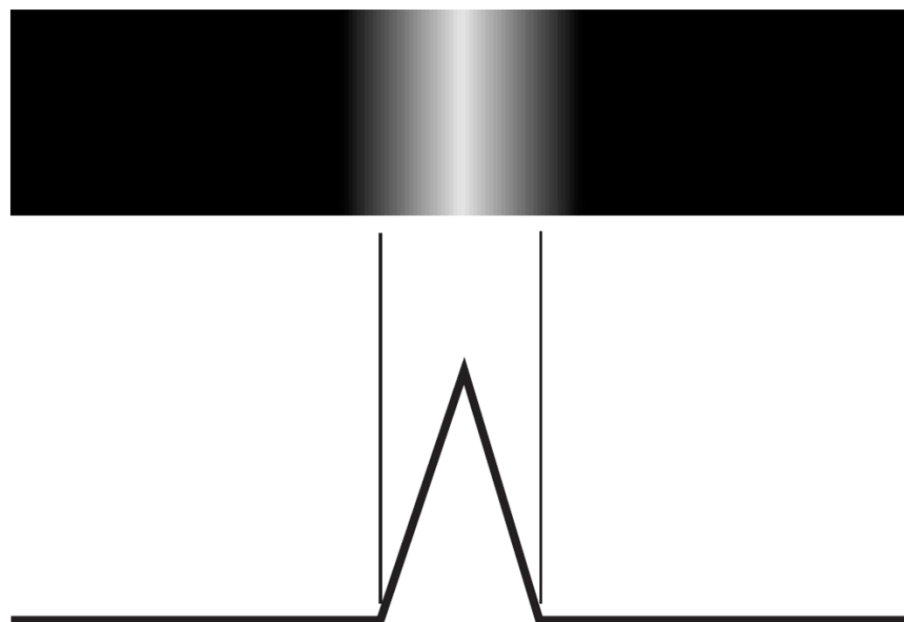
- 斜坡边缘 (Ramp Edge)
 - 实际边缘通常是模糊 (聚焦机制)、有噪声 (电子器件)
 - 斜率与模糊程度成反比



边缘模型



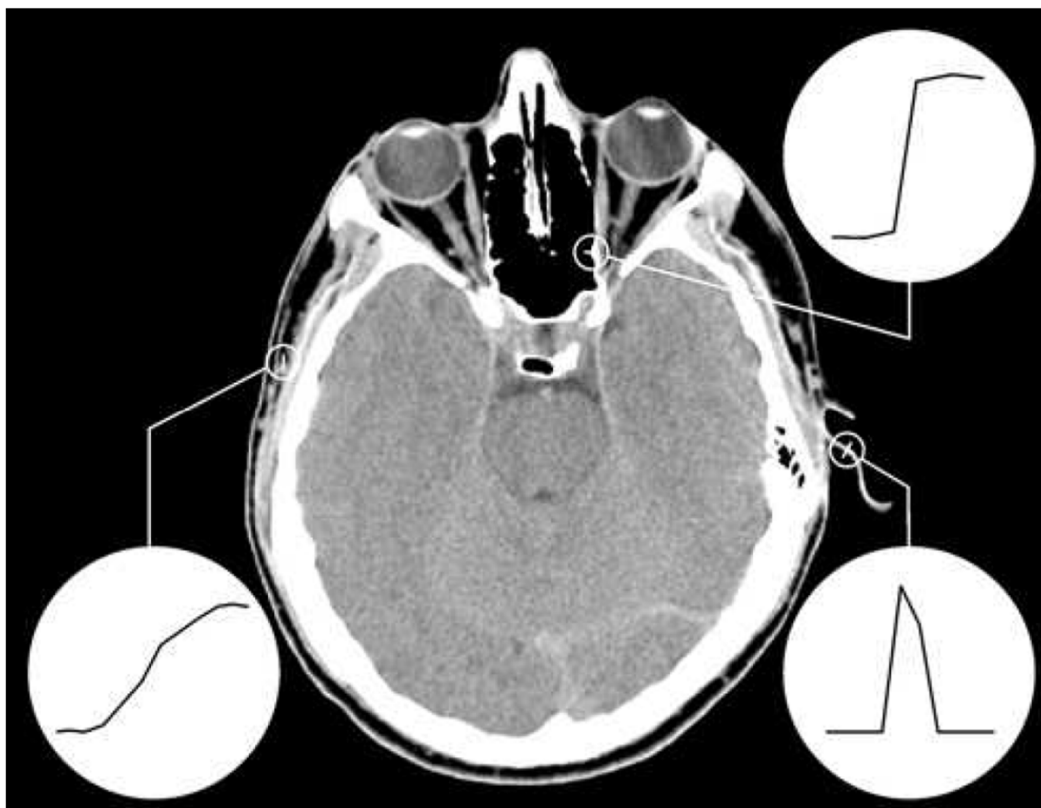
- 屋顶边缘 (Roof Edge)
 - 表示穿过区域的线
 - 出现在数字化的线条图、卫星图像中的道路





举例

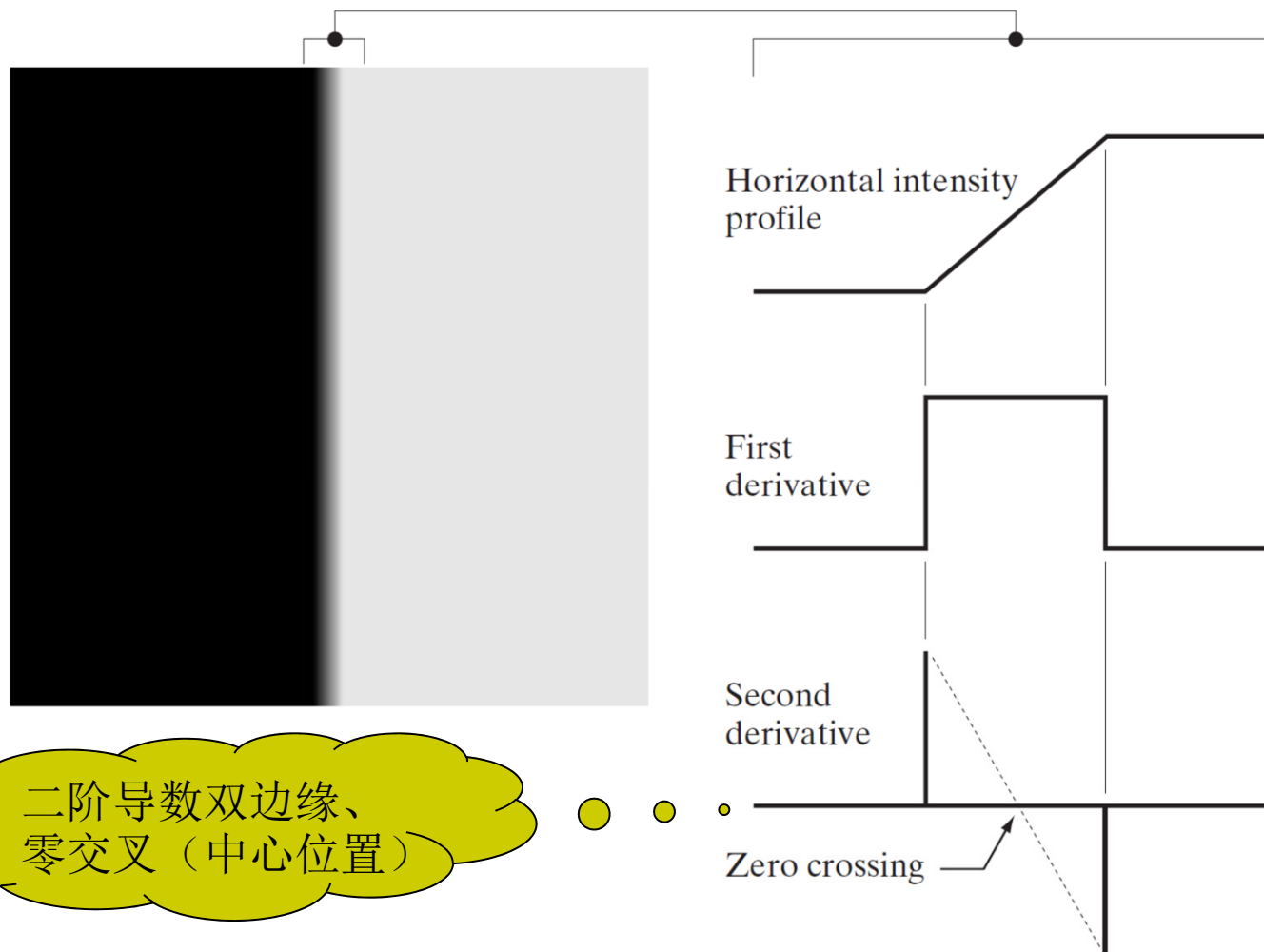
- 三种边缘通常同时出现
 - 陡峭的斜坡通常被认为是台阶



举例

● 斜坡边缘

- 1、一阶导数的大小用来检测某像素处是否存在边缘
- 2、二阶导数的符号用来确定一个边缘像素位于亮或暗区域

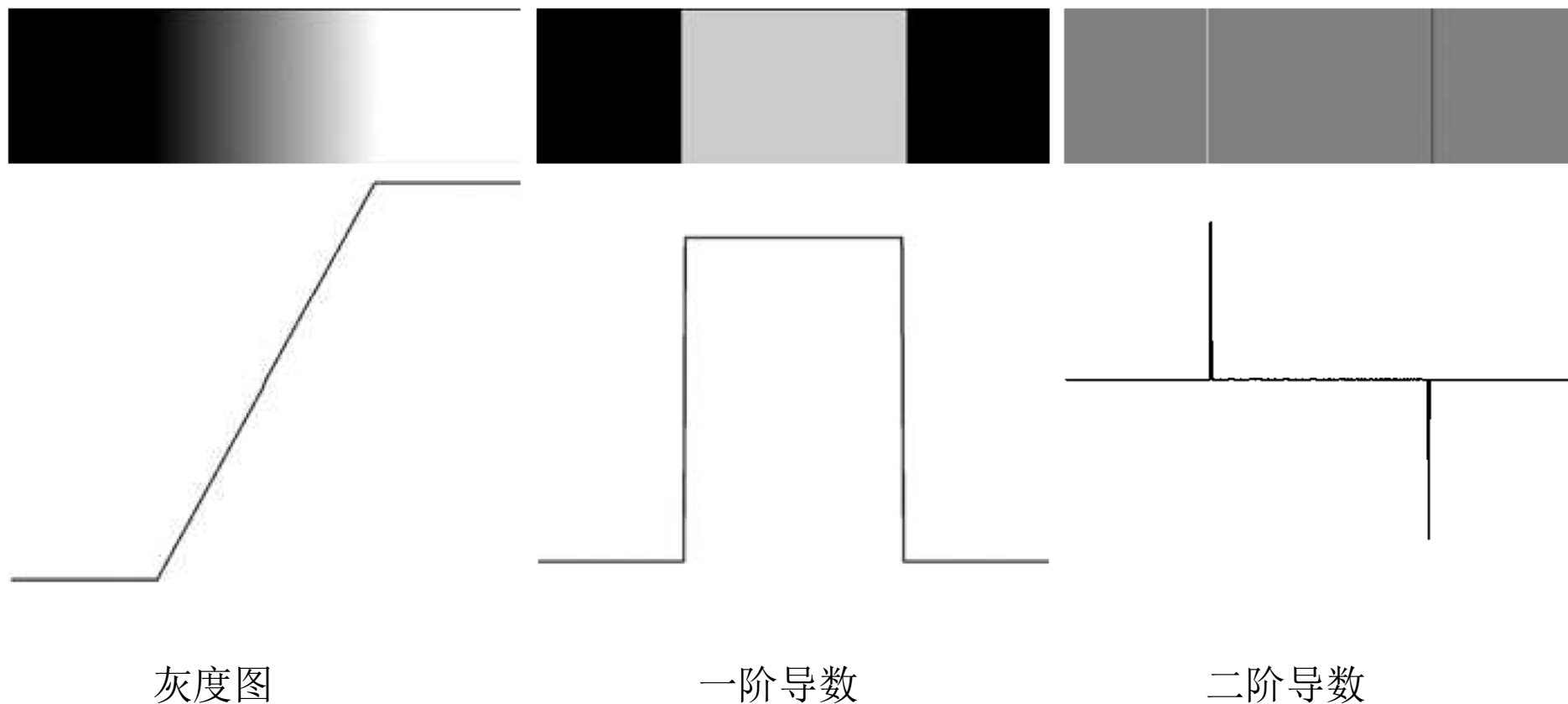


二阶导数双边缘、
零交叉（中心位置）

存在噪声的边缘



- 无噪声的情况

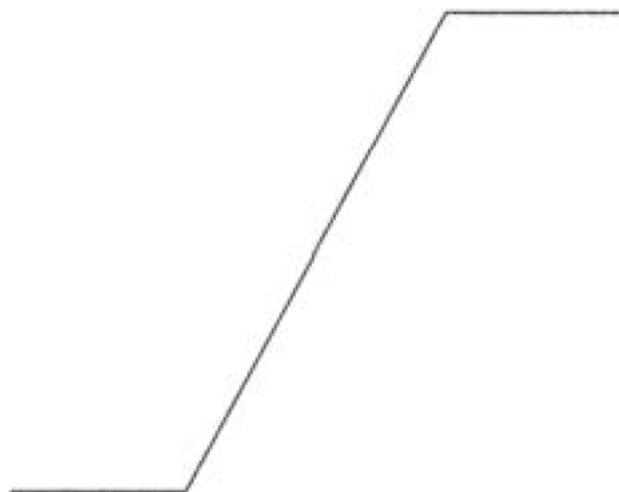
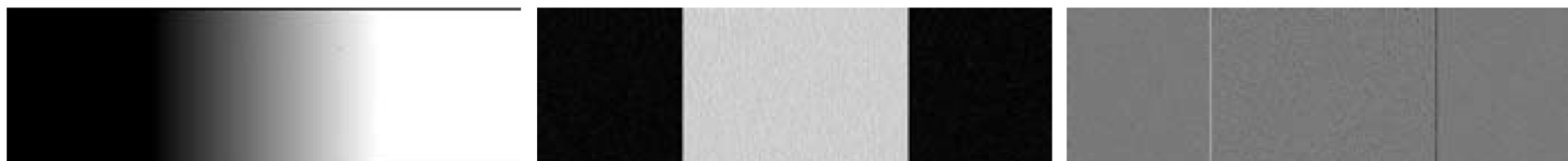


存在噪声的边缘

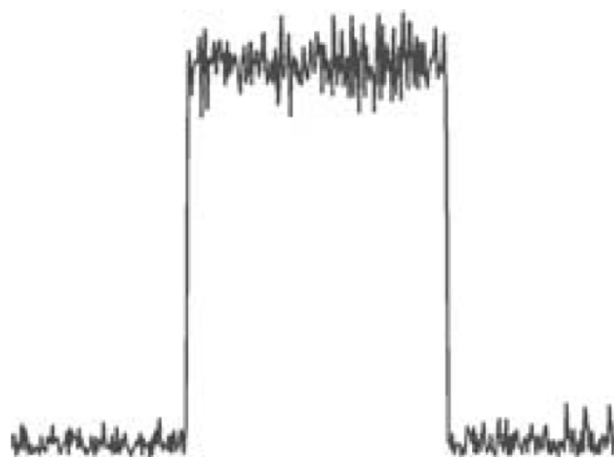
- 1、视觉上噪声并不明显
- 2、噪声对导数影响很大
- 3、二阶导数更敏感



- 均值为0，标准差为0.1的高斯噪声



灰度图



一阶导数



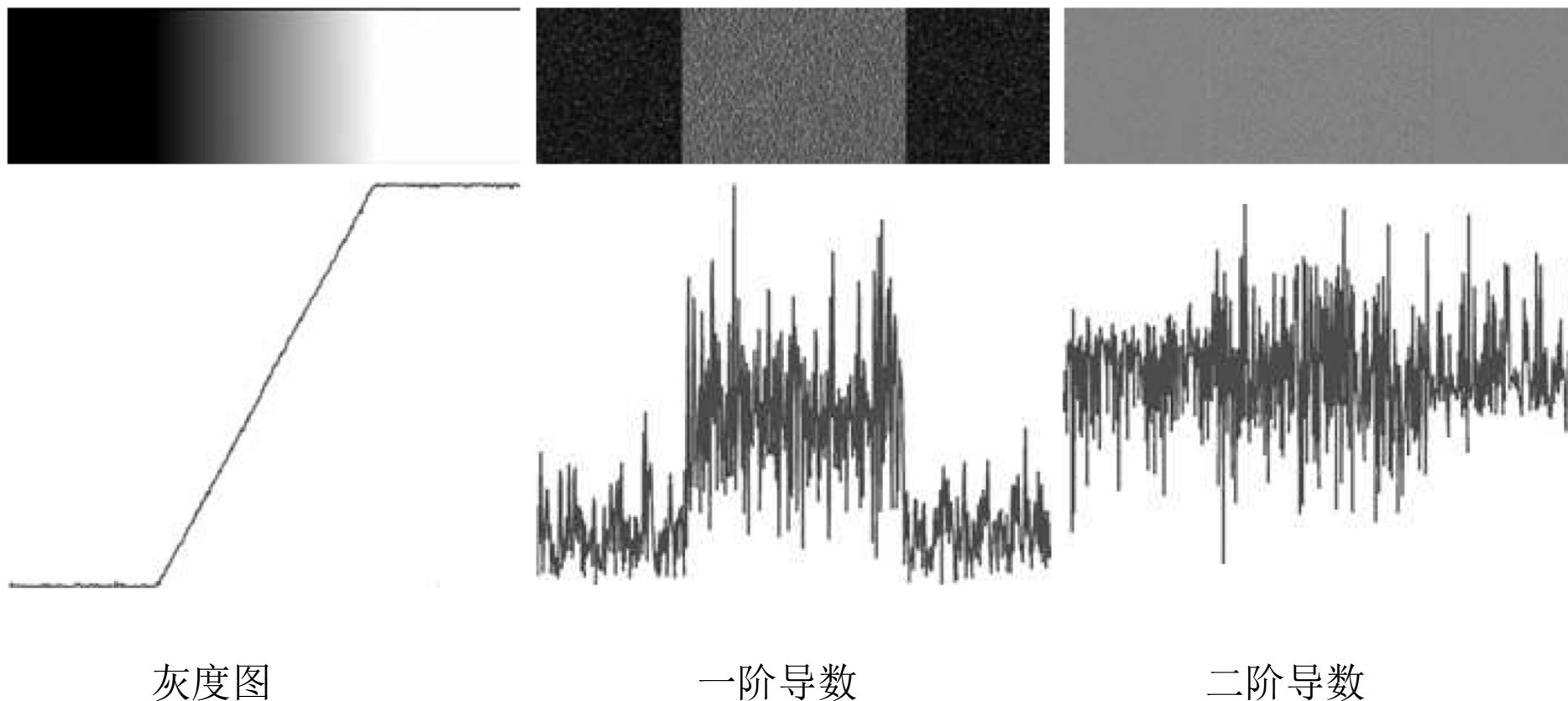
二阶导数

存在噪声的边缘

- 1、视觉上噪声并不明显
- 2、噪声对导数影响很大
- 3、二阶导数无法辨认



- 均值为0，标准差为1的高斯噪声

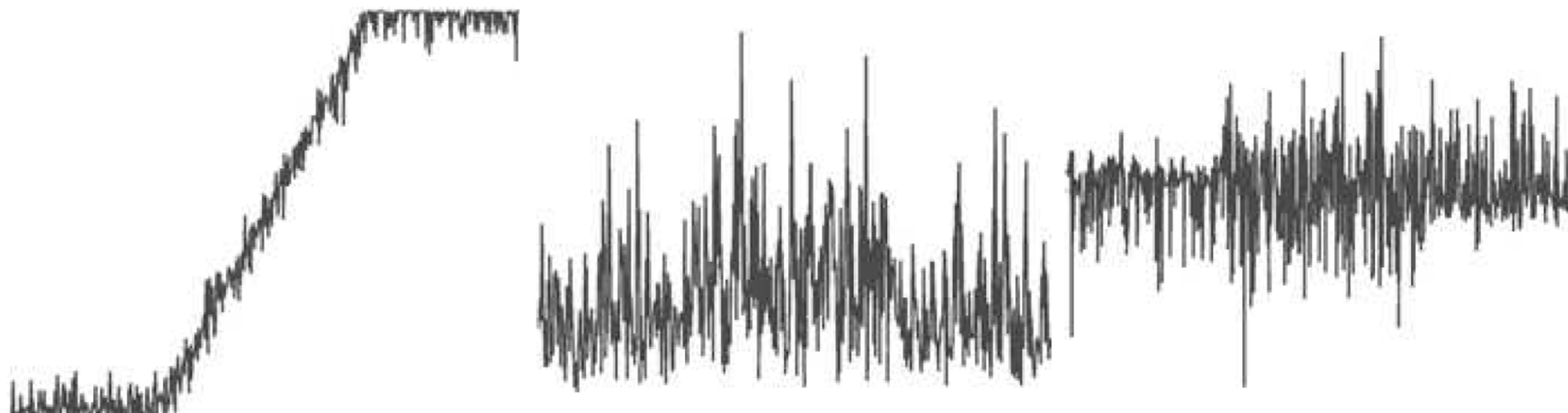


存在噪声的边缘

- 1、视觉上噪声并不明显
- 2、噪声对导数影响很大
- 3、导数无法辨认



- 均值为0，标准差为10的高斯噪声



灰度图

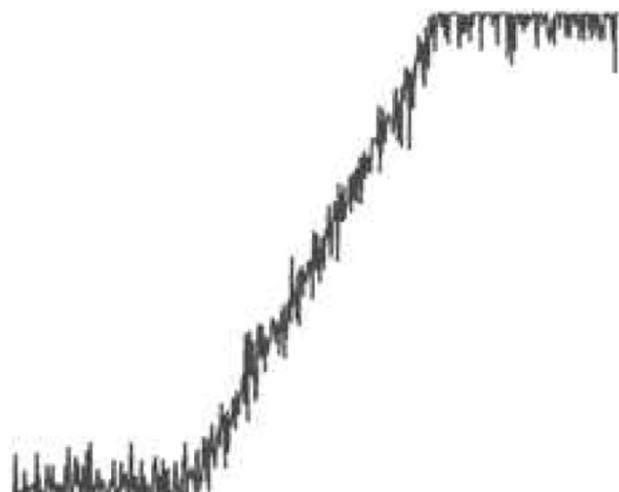
一阶导数

二阶导数

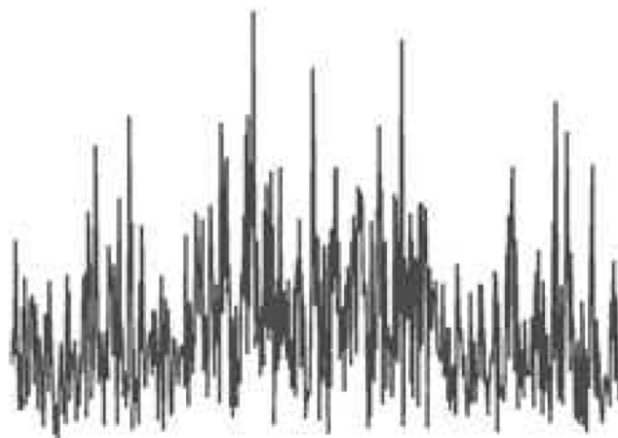
存在噪声的边缘

存在噪声时，图像平滑很必要

- 均值为0，标准差为10的高斯噪声



灰度图



一阶导数



二阶导数

边缘检测的三个基本步骤



1. 为降噪对图像进行平滑处理
 - 导数对噪声敏感
2. 边缘点的检测
 - 抽取所有的潜在边缘点
3. 边缘定位
 - 选出真正的边缘点



提纲

- 基础知识
- 点、线、边缘检测
 - 背景知识
 - 孤立点的检测
 - 线检测
 - 边缘模型
 - 基本边缘检测
 - 高级边缘检测
- 边缘连接和边界检测

图像梯度及其性质



- 利用梯度检测边缘

- 梯度：最大变化率的方向

- 线性算子

$$\nabla f \equiv \text{grad}(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

- 大小

- 非线性

$$M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2}$$

- 方向

$$\alpha(x, y) = \tan^{-1} \begin{bmatrix} g_y \\ g_x \end{bmatrix}$$

- 边缘的方向与梯度正交



图像梯度及其性质

- 利用梯度检测边缘

- 梯度：最大变化率的方向

- 线性算子

$$\nabla f \equiv \text{grad}(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

- 大小（近似计算）

- 计算简单

$$M(x, y) \approx |g_x| + |g_y|$$

- 方向

$$\alpha(x, y) = \tan^{-1} \begin{bmatrix} g_y \\ g_x \end{bmatrix}$$

- 边缘的方向与梯度正交

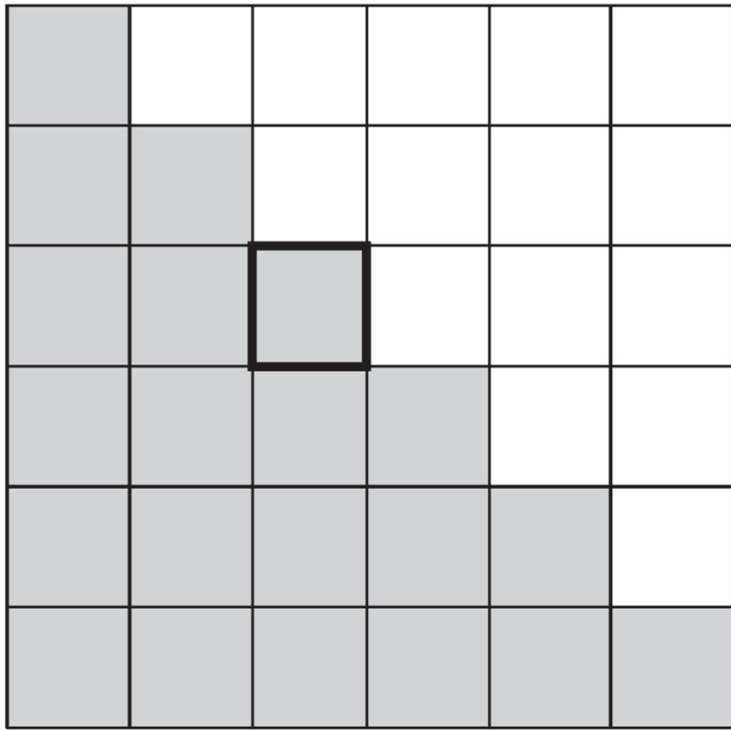
举例



灰色代表0
白色代表1

→ y

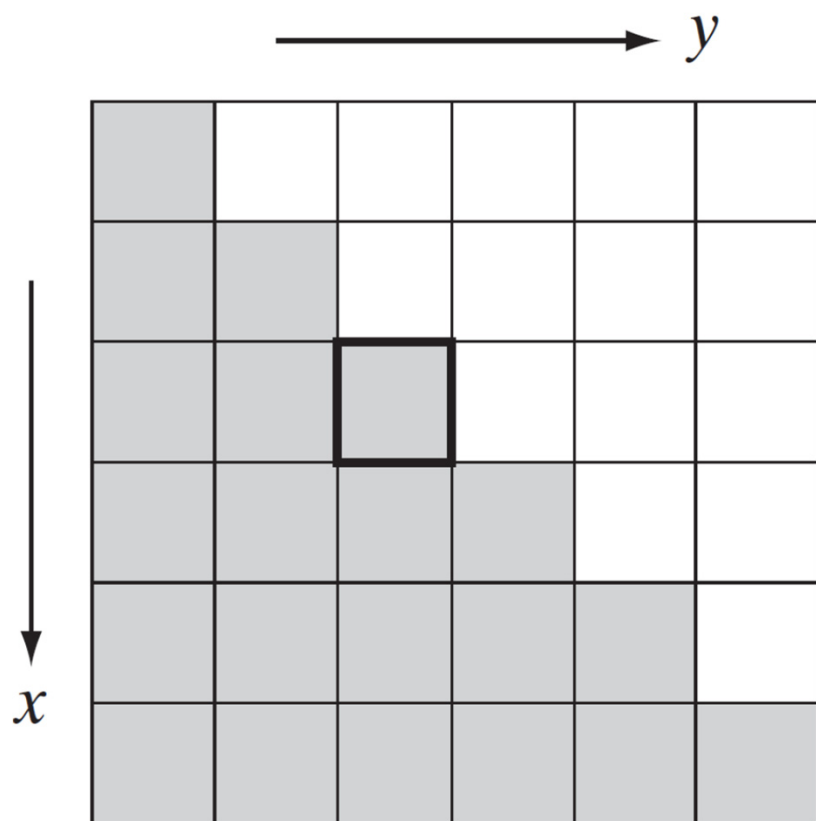
↓ x



-1	-1	-1	-1	0	1
0	0	0	-1	0	1
1	1	1	-1	0	1

梯度计算模板

举例

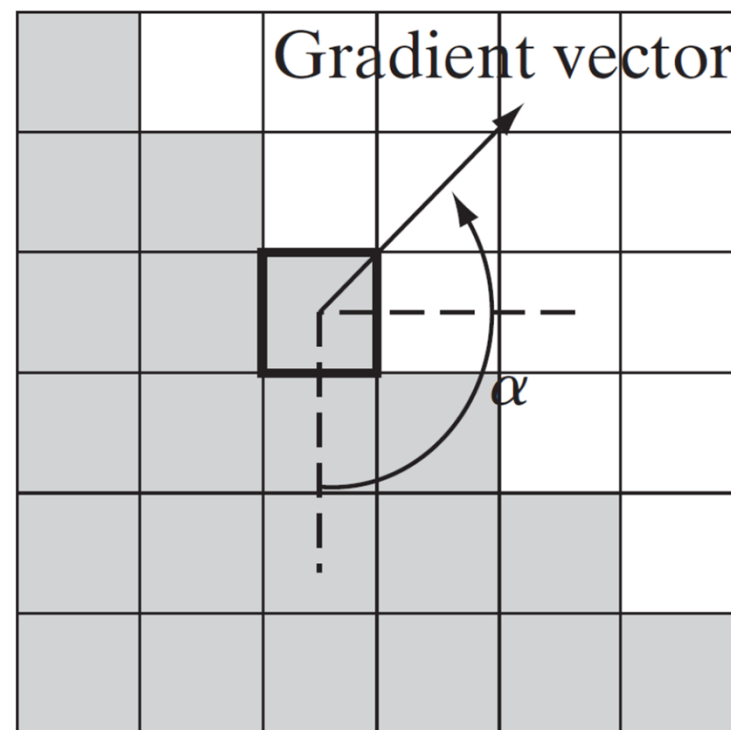
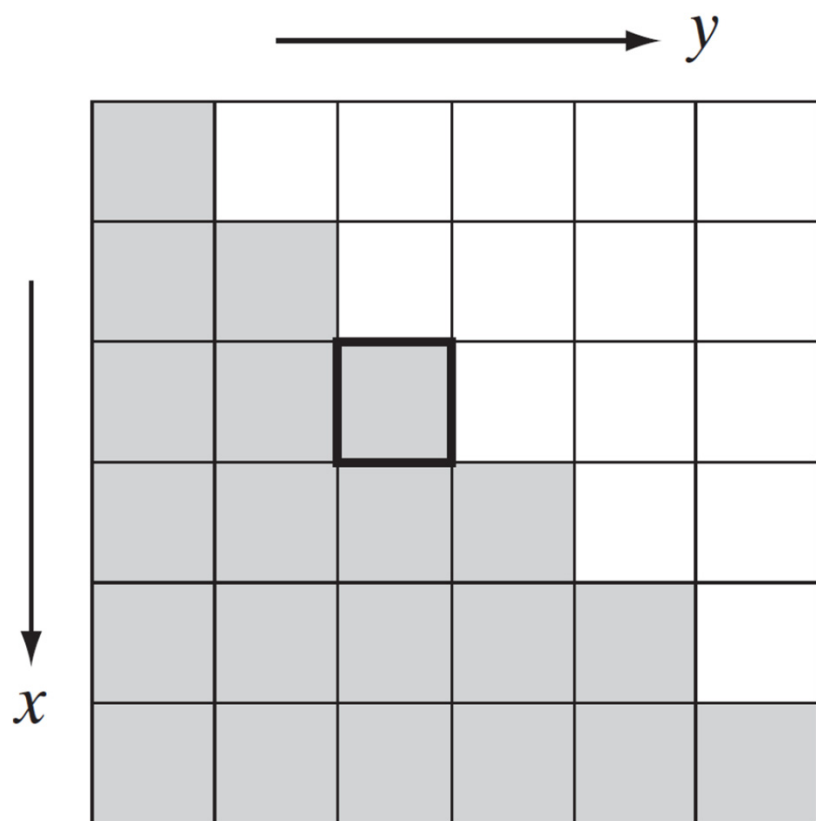


$$\nabla f = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} -2 \\ 2 \end{bmatrix}$$

$$M(x, y) = 2\sqrt{2}$$

$$\alpha(x, y) = \tan^{-1}(g_y/g_x) = -45^\circ$$

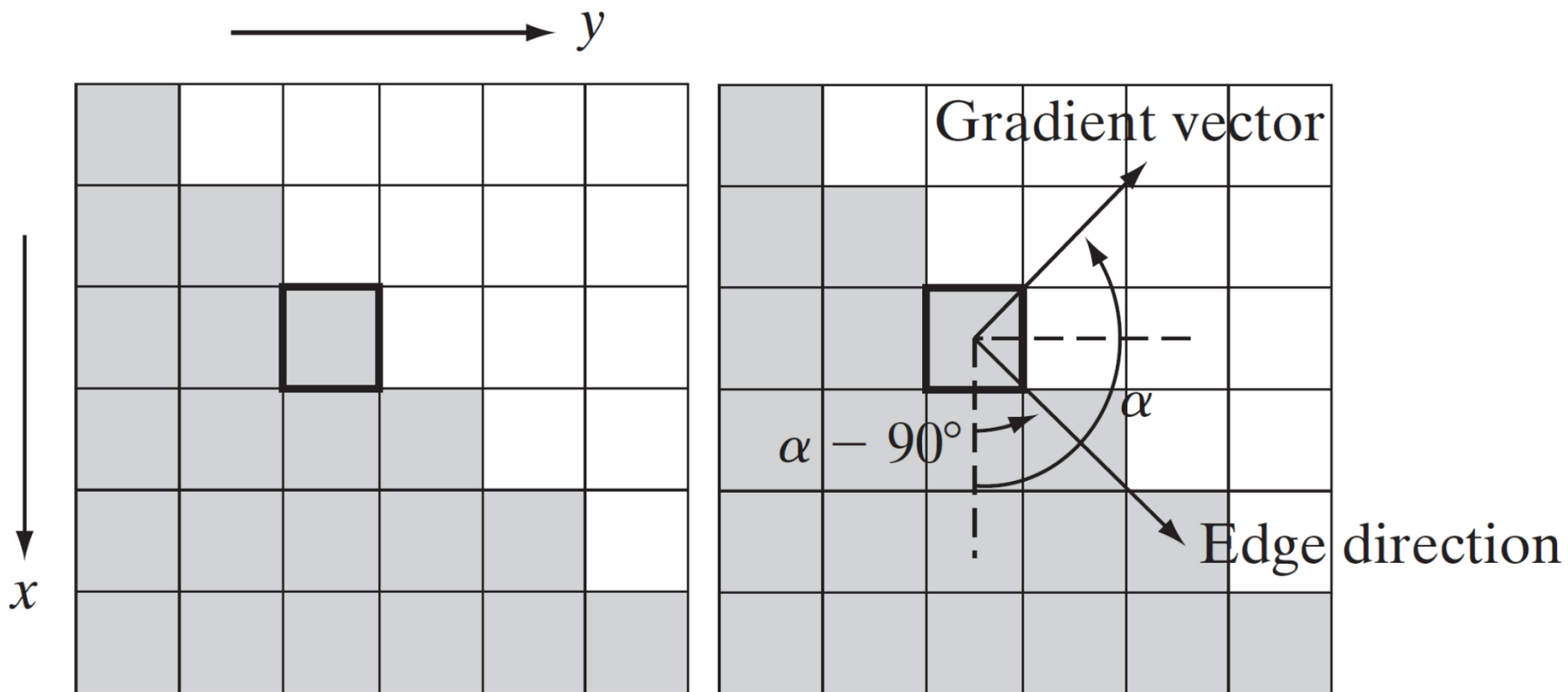
举例



举例



梯度向量也被称为边缘法线 (Edge Normal)



梯度算子



- 离散近似

$$g_x = \frac{\partial f(x, y)}{\partial x} = f(x + 1, y) - f(x, y)$$

$$g_y = \frac{\partial f(x, y)}{\partial y} = f(x, y + 1) - f(x, y)$$

- 1维模板

-1
1

-1	1
----	---



梯度算子

- 罗伯特交叉梯度算子

- 考虑对角方向

$$g_x = \frac{\partial f}{\partial x} = (z_9 - z_5)$$

$$g_y = \frac{\partial f}{\partial y} = (z_8 - z_6)$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- 2维模板

-1	0	0	-1
0	1	1	0

Roberts



梯度算子

- 普鲁伊特 (Prewitt) 算子

- 关于中心点对称

$$g_x = \frac{\partial f}{\partial x} = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$

$$g_y = \frac{\partial f}{\partial y} = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

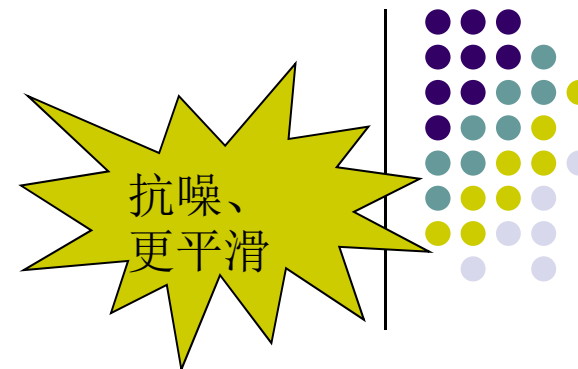
z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- 2维模板

-1	-1	-1	-1	0	1
0	0	0	-1	0	1
1	1	1	-1	0	1

Prewitt

梯度算子



- Sobel算子

- 关于中心点对称

$$g_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$g_y = \frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- 2维模板

-1	-2	-1	-1	0	1
0	0	0	-2	0	2
1	2	1	-1	0	1

Sobel

梯度算子



- 强调对角方向的算子

0	1	1	-1	-1	0
-1	0	1	-1	0	1
-1	-1	0	0	1	1

Prewitt

0	1	2	-2	-1	0
-1	0	1	-1	0	1
-2	-1	0	0	1	2

Sobel

举例

- 基于Sobel算子



原图

举例

- 基于Sobel算子

水平边缘
更加突出



原图



$|g_x|$

举例

- 基于Sobel算子

垂直边缘
更加突出



原图



$|g_y|$

举例

- 基于Sobel算子

称为边缘图
(Edge Map)



原图



$|g_x| + |g_y|$

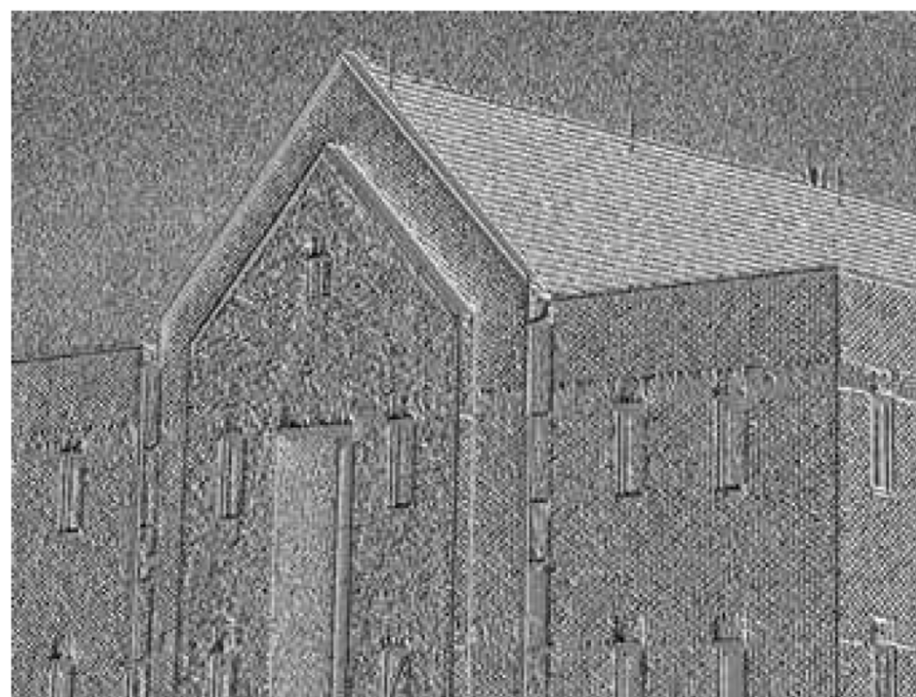
举例

- 基于Sobel算子

- 1、相对而言不是很有用
- 2、常数区域表示梯度方向一样



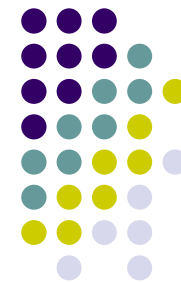
原图



角度

举例

- 去掉砖块的影响



平滑后的图像



$|g_x|$

举例

- 去掉砖块的影响



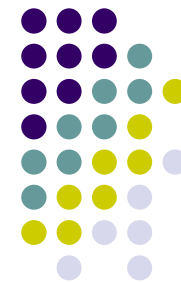
平滑后的图像



$|g_y|$

举例

- 去掉砖块的影响



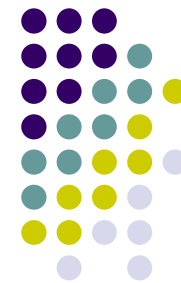
平滑后的图像



$|g_x| + |g_y|$

举例

- 对比



$$|g_x| + |g_y|$$



$$\text{平滑}|g_x| + |g_y|$$

举例

- 强调对角线上的边缘

0	1	2	-2	-1	0
-1	0	1	-1	0	1
-2	-1	0	0	1	2

Sobel



阈值化

- 保留超过最大值33%的像素
 - 初始图像未平滑处理



$$|g_x| + |g_y|$$



阈值化

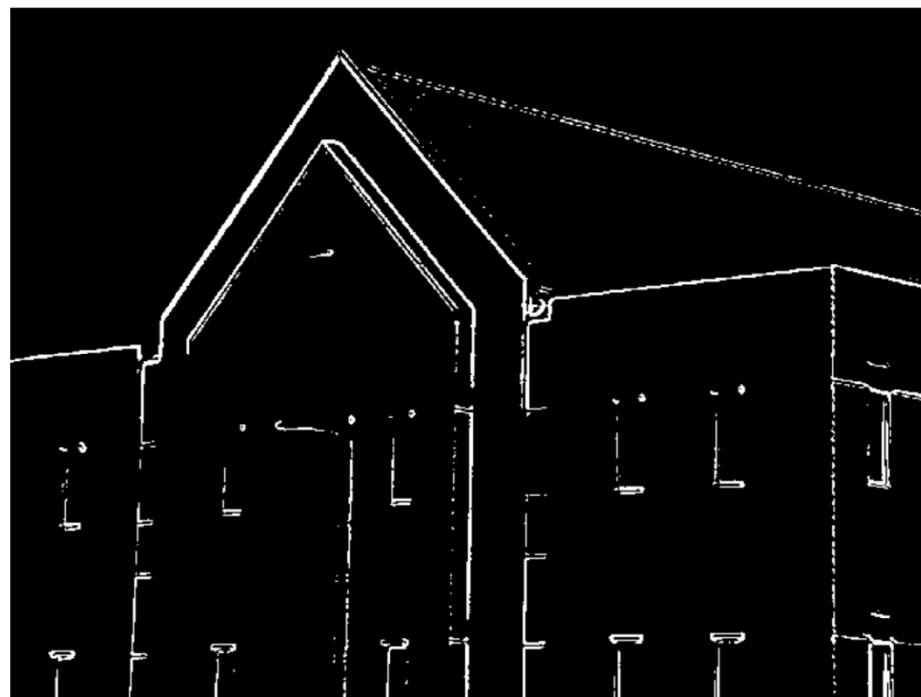
阈值化

- 保留超过最大值33%的像素
 - 初始图像平滑处理

边缘中断缓解
但部分边缘消失



$$|g_x| + |g_y|$$



阈值化

阈值化

- 保留超过最大值33%的像素

边缘中断缓解
但部分边缘消失



阈值化



平滑后阈值化



提纲

- 基础知识
- 点、线、边缘检测
 - 背景知识
 - 孤立点的检测
 - 线检测
 - 边缘模型
 - 基本边缘检测
 - 高级边缘检测
- 边缘连接和边界检测

高级边缘检测



- 基本边缘检测
 - 没有考虑边缘的性质
 - 没有考虑噪声模型
- Marr-Hildreth边缘检测器
- 坎尼 (Canny) 边缘检测器

Marr-Hildreth边缘检测器



1. 灰度变化和图像尺度有关
 - 需要用不同尺寸的算子
2. 灰度变化会影响导数
 - 一阶导数出现波峰或波谷
 - 二阶导数出现零交叉
- 理想的检测器具备如下功能
 1. 能够近似1阶或2阶导数
 2. 能够被调整以在不同尺寸上起作用
 - 大的算子检测模糊边缘、小的算子检测细节

Marr-Hildreth边缘检测器



- 滤波器 $\nabla^2 G$

- ∇^2 是拉普拉斯算子

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

- G 是2维高斯函数

$$G(x, y) = e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

- 其中 σ 是标准差

- 满足上页两个条件的最佳算子

Marr-Hildreth边缘检测器



- 滤波器 $\nabla^2 G$

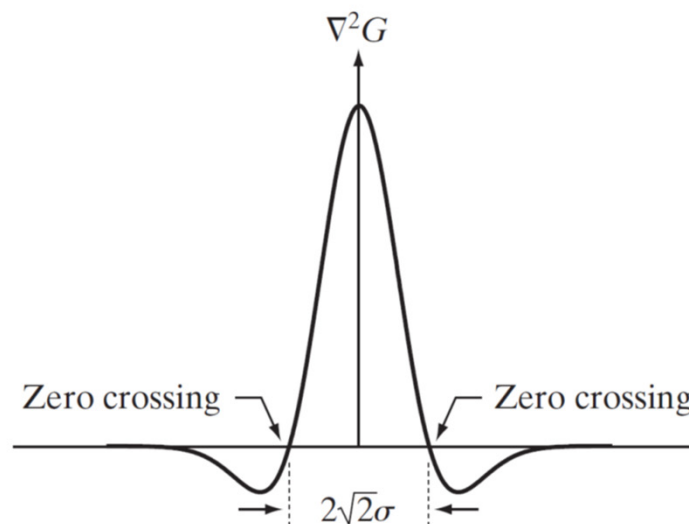
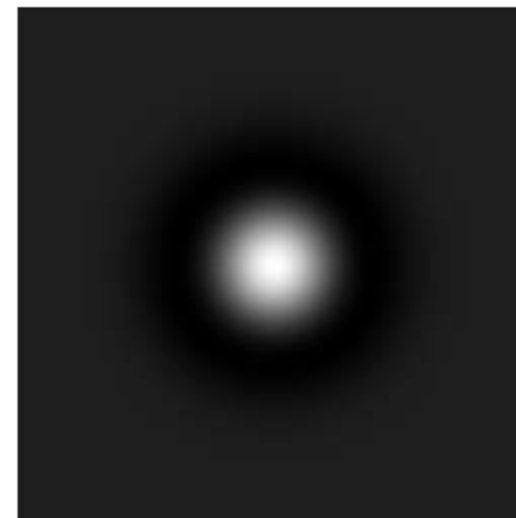
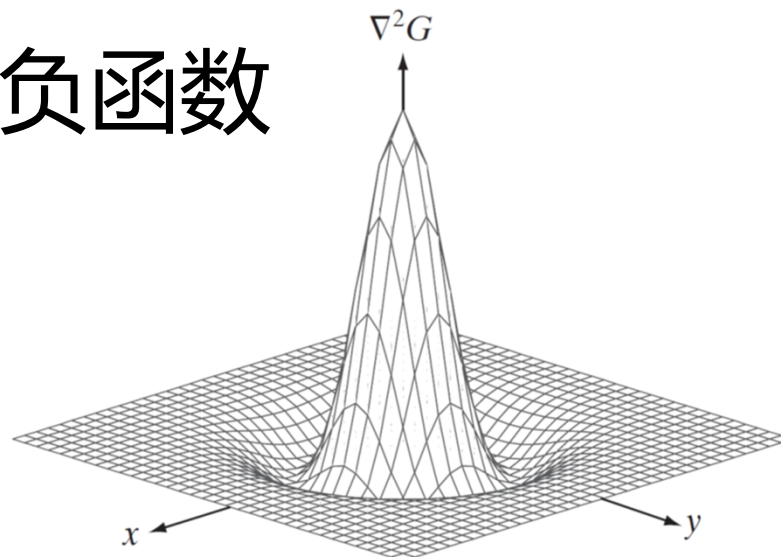
$$\begin{aligned}\nabla^2 G(x, y) &= \frac{\partial^2 G(x, y)}{\partial x^2} + \frac{\partial^2 G(x, y)}{\partial y^2} \\ &= \frac{\partial}{\partial x} \left[\frac{-x}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \right] + \frac{\partial}{\partial y} \left[\frac{-y}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \right] \\ &= \left[\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2} \right] e^{-\frac{x^2+y^2}{2\sigma^2}} + \left[\frac{y^2}{\sigma^4} - \frac{1}{\sigma^2} \right] e^{-\frac{x^2+y^2}{2\sigma^2}} \\ &= \left[\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \right] e^{-\frac{x^2+y^2}{2\sigma^2}}\end{aligned}$$

- 高斯的拉普拉斯 (LoG)

高斯的拉普拉斯 (LoG)



- LoG的负函数



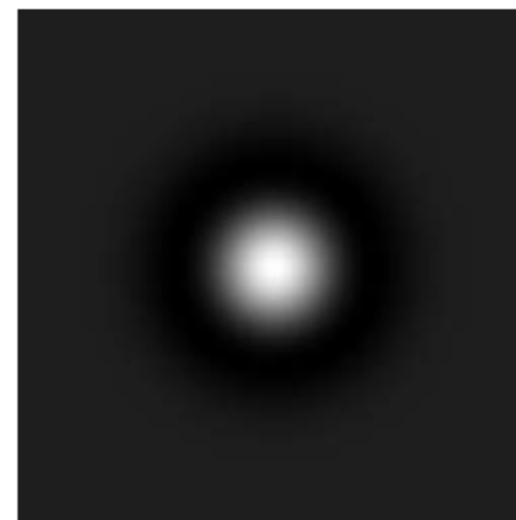
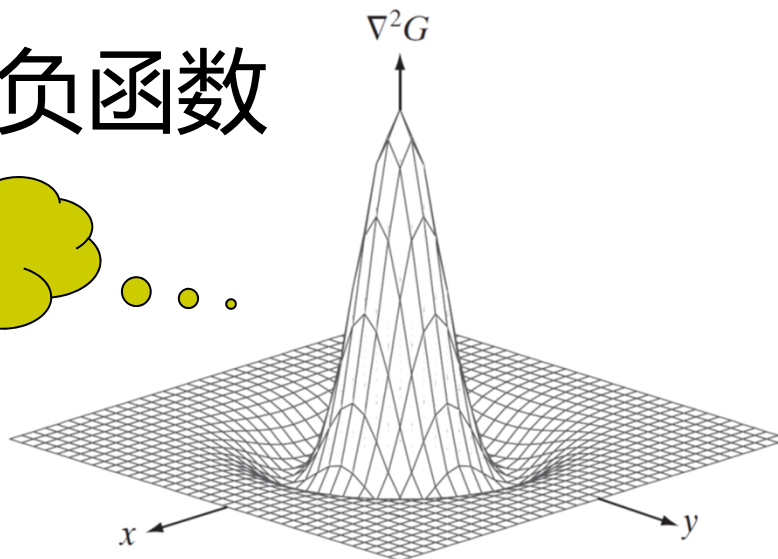
0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

高斯的拉普拉斯 (LoG)

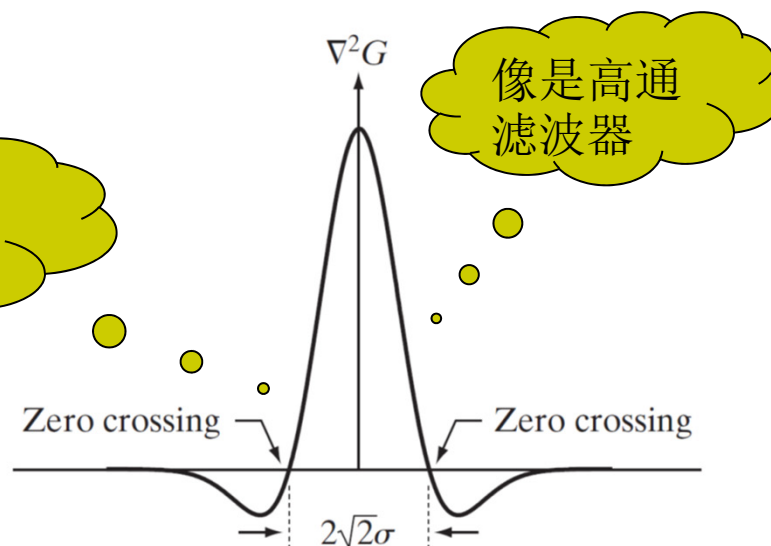


- LoG的负函数

墨西哥草帽算子



零交叉出现在 $x^2 + y^2 = 2\sigma^2$



0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

频率域滤波器→空间滤波器

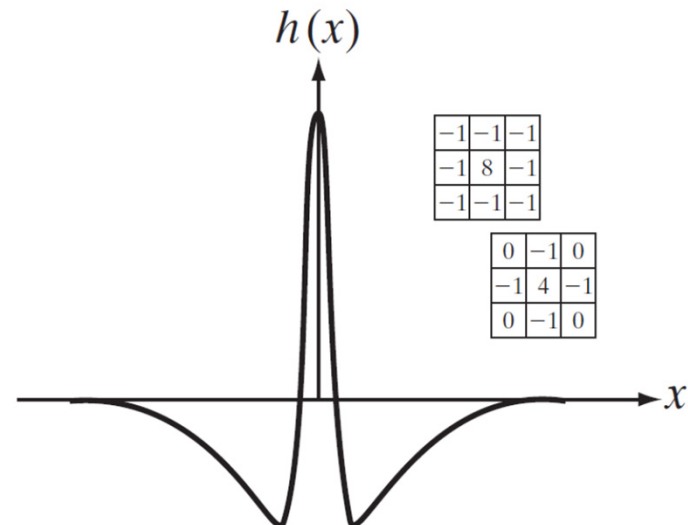
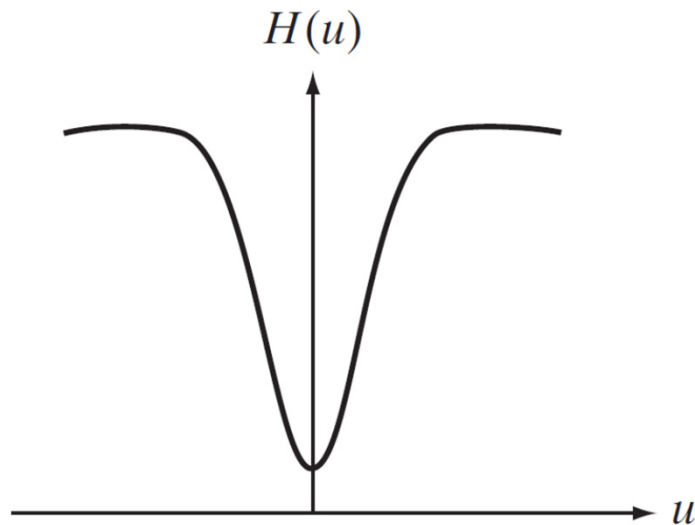


- 利用高斯函数构造高通滤波器

$$H(u) = A e^{-u^2/2\sigma_1^2} - B e^{-u^2/2\sigma_2^2}$$

- 空间域对应的滤波器

$$h(x) = \sqrt{2\pi}\sigma_1 A e^{-2\pi^2\sigma_1^2 x^2} - \sqrt{2\pi}\sigma_2 B e^{-2\pi^2\sigma_2^2 x^2}$$



LoG模板的生成



- 生成不同尺寸的模板

$$\nabla^2 G(x, y) = \left[\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

1. 对上式进行采样，得到 $n \times n$ 的模板
2. 缩放系数，保证求和为0

$$G(x, y) = e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

更加有效

- a) 对上式进行采样，得到 $n \times n$ 的模板
- b) 与拉普拉斯模板卷积
 - 系数之和自动为0

LoG的优势



$$\nabla^2 G(x, y) = \left[\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

● 优势

1. 高斯部分会模糊图像

- 可以去掉尺寸小于 σ 的细节，比如噪声
- 高斯函数曲线平滑，不会引入振铃等干扰

2. 二阶导数

- 各向同性，对任何方向的变化有相同的相应
- 符合人的视觉系统



两种实现方式

- 将LoG滤波器和图像卷积

$$g(x, y) = [\nabla^2 G(x, y)] \star f(x, y)$$

- 寻找零交叉来确定边缘位置

- 等价形式

- 卷积和二阶导是线性操作

$$g(x, y) = \nabla^2 [G(x, y) \star f(x, y)]$$

1. 先用高斯滤波器平滑图像
2. 再应用拉普拉斯算子

Marr-Hildreth边缘检测器



1. 用 $n \times n$ 的高斯低通滤波器平滑图像

- 滤波器通过对高斯函数采样得到

$$G(x, y) = e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

n 是大于等于 6σ 的最小奇数

2. 计算上述图像的拉普拉斯

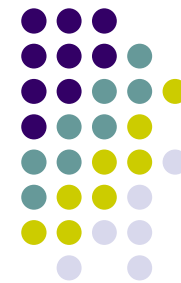
1	1	1
1	-8	1
1	1	1

1. 检查某像素两个相对邻域像素的符号
(上下、左右、两对角)
2. 符号相反, 并且差异大于某阈值

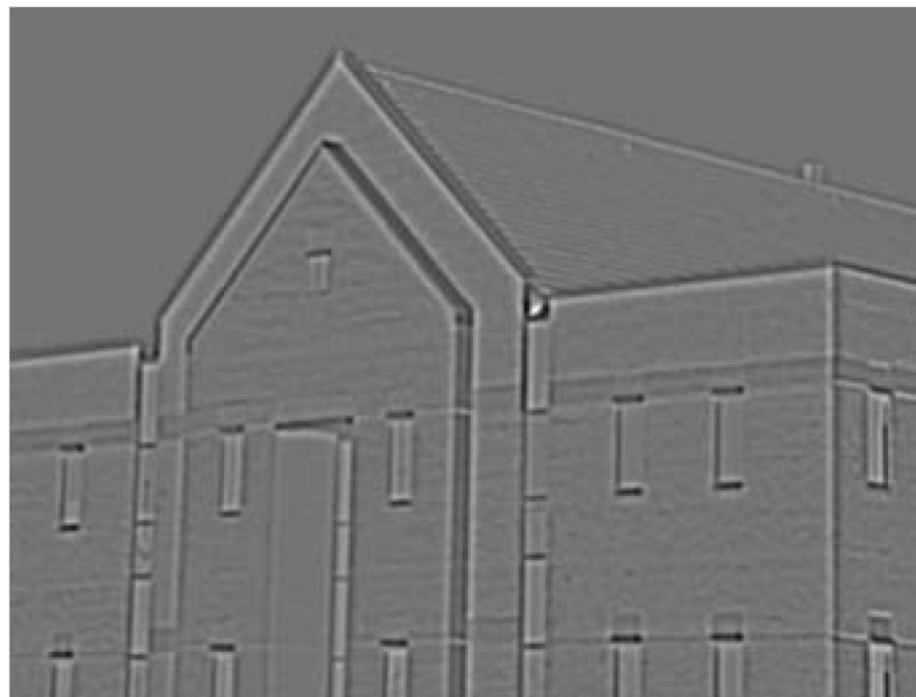
3. 寻找上述结果的零交叉

举例

- Marr-Hildreth边缘检测器



原图



前两个步骤的结果

举例

- Marr-Hildreth边缘检测器



产生闭环，
“意大利空心粉”效应

零交叉（阈值为0）

举例

- Marr-Hildreth边缘检测器

边缘的宽度为1



零交叉（阈值为0）



零交叉（阈值为最大值的4%）





扩展

1. 考虑不同的尺度
 - 尝试不同的 σ ，保留共同的零交叉
2. 使用高斯差分 (DoG) 来近似LoG

$$\text{DoG}(x, y) = \frac{1}{2\pi\sigma_1^2} e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{1}{2\pi\sigma_2^2} e^{-\frac{x^2+y^2}{2\sigma_2^2}}$$

- 人视觉系统的某些通道对方向和频率是有选择的 (标准差比值为1.75:1)
- 采用1.6:1，可以建模上述现象，并很好地近似LoG

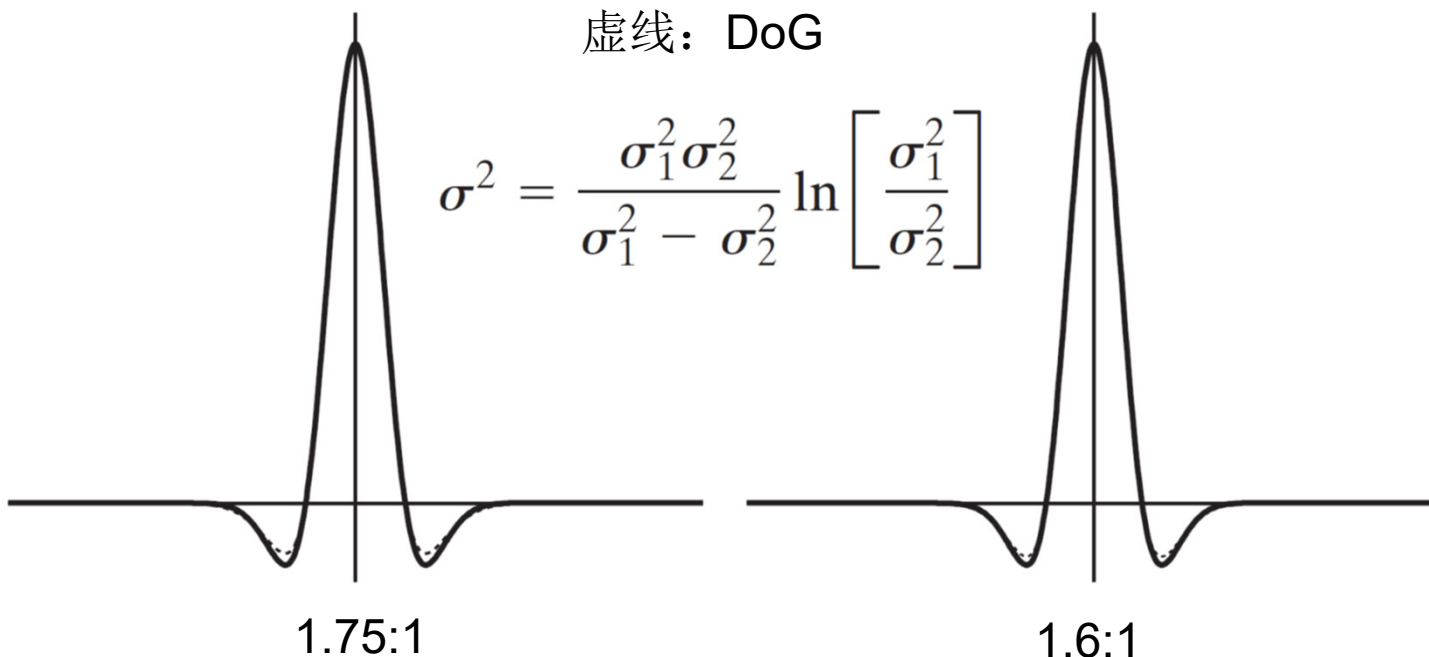


扩展

1. 考虑不同的尺度
 - 尝试不同的 σ ，保留共同的零交叉
2. 使用高斯差分 (DoG) 来近似LoG

实线: LoG
虚线: DoG

$$\sigma^2 = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 - \sigma_2^2} \ln \left[\frac{\sigma_1^2}{\sigma_2^2} \right]$$



高级边缘检测



- 基本边缘检测
 - 没有考虑边缘的性质
 - 没有考虑噪声模型
- Marr-Hildreth边缘检测器
- 坎尼 (Canny) 边缘检测器

坎尼（Canny）边缘检测器



1. 低错误率
 - 所有边缘都被找到，并且没有伪响应
2. 边缘点应被很好地定位
 - 已定位的边缘必须尽可能接近真实边缘
3. 单一的边缘点响应
 - 对每个真实边缘点，检测器仅返回1个点

坎尼（Canny）边缘检测器



- 数学分析
 - 考虑加性高斯白噪声污染的1维台阶边缘
 - 高斯一阶导数是近似最优的检测器
- 拓展到二维情况
 - 挑战：边缘可能是任意方向
 - 使用二维高斯函数平滑图像
 - 基于梯度寻找边缘的方向

坎尼（Canny）边缘检测器



1. 高斯函数平滑输入图像 f

$$f_s(x, y) = G(x, y) \star f(x, y)$$

- 其中

$$G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

2. 计算图像 f_s 的梯度

- 梯度大小 $M(x, y) = \sqrt{g_x^2 + g_y^2}$

- 梯度方向 $\alpha(x, y) = \tan^{-1} \left[\frac{g_y}{g_x} \right]$

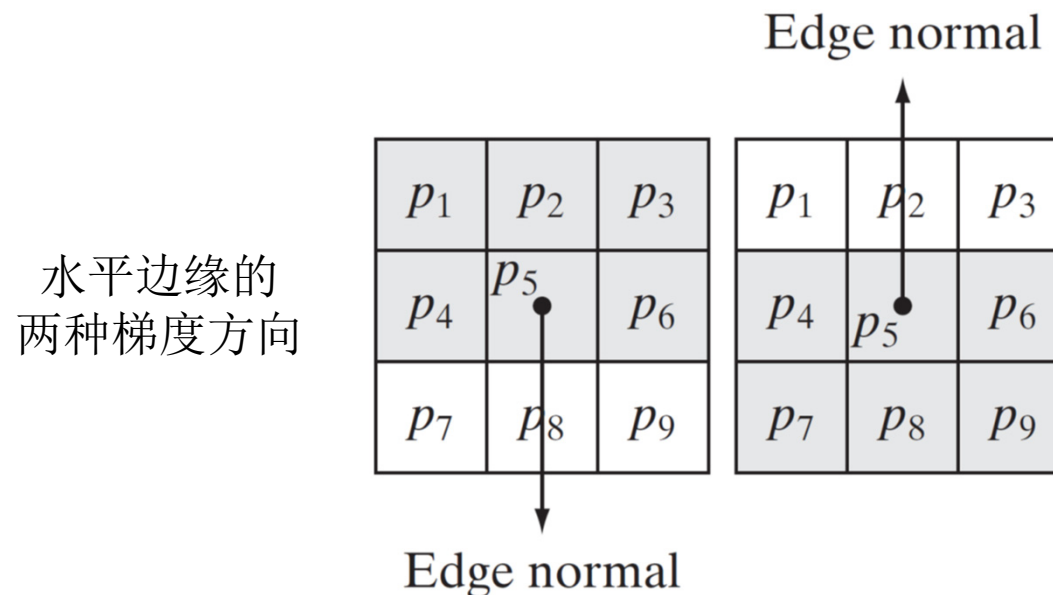
- 其中 $g_x = \partial f_s / \partial x$, $g_y = \partial f_s / \partial y$

坎尼（Canny）边缘检测器



3. 非最大抑制

- 目的：把梯度生成的粗边缘变细
- 指定梯度（边缘法线）的多个离散方向
 - 4种边缘：水平、垂直、 $+45^\circ$ 、 -45°



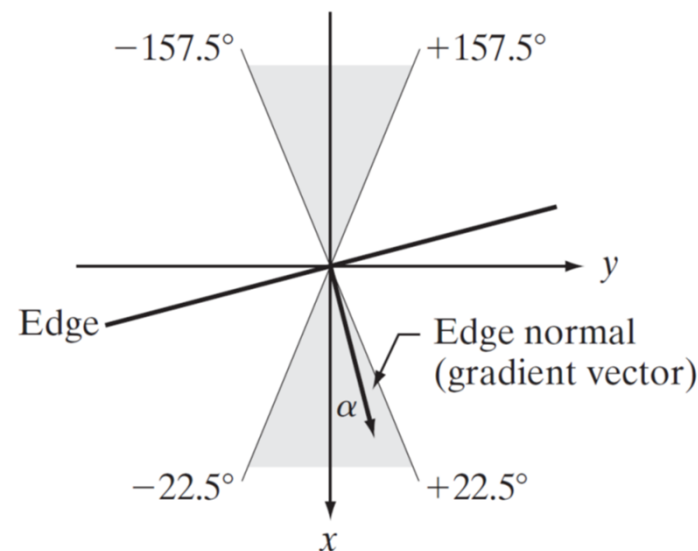
坎尼（Canny）边缘检测器



3. 非最大抑制

- 目的：把梯度生成的粗边缘变细
- 指定梯度（边缘法线）的多个离散方向
 - 4种边缘：水平、垂直、 $+45^\circ$ 、 -45°
- 根据梯度（边缘法线）的方向确定边缘的方向

水平边缘
的梯度范围



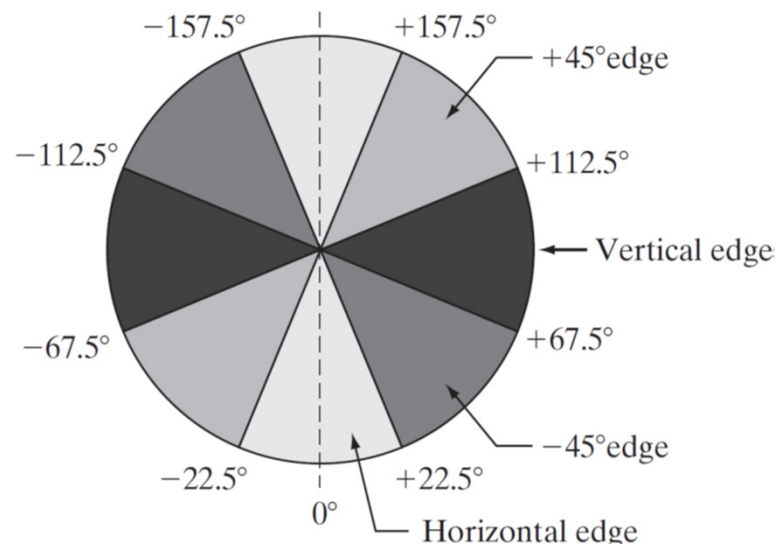
坎尼（Canny）边缘检测器



3. 非最大抑制

- 目的：把梯度生成的粗边缘变细
- 指定梯度（边缘法线）的多个离散方向
 - 4种边缘：水平、垂直、 $+45^\circ$ 、 -45°
- 根据梯度（边缘法线）的方向确定边缘的方向

4种边缘
的梯度范围



坎尼（Canny）边缘检测器



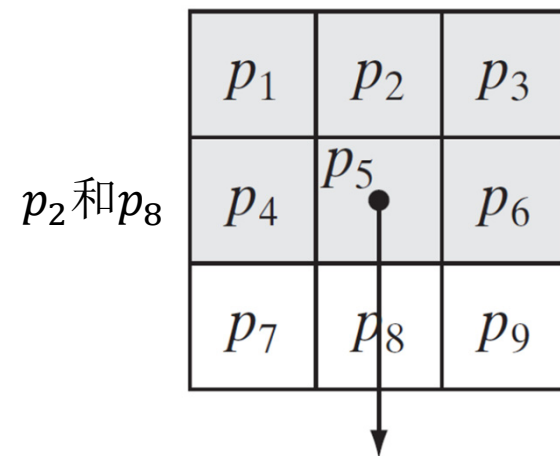
3. 非最大抑制

- 考虑 (x, y) 为中心的 3×3 区域
- 考虑4个方向：水平、垂直、 $+45^\circ$ 、 -45°
- 确定和梯度 $a(x, y)$ 最接近的方向 d_k
- 如果 $M(x, y)$ 的值比 (x, y) 在 d_k 方向的任一邻居数值小，对其抑制：

$$g_N(x, y) = 0$$

- 否则，保留：

$$g_N(x, y) = M(x, y)$$



坎尼（Canny）边缘检测器



4. 滞后阈值

- 目的：减少伪边缘点
- 两个阈值：低阈值 T_L 、高阈值 T_H
- 两个阈值的比值为：2:1或3:1
- 利用 T_H 阈值化
 - 强边缘点 $g_{NH}(x, y) = g_N(x, y) \geq T_H$
- 利用 T_L 阈值化
$$g_{NL}(x, y) = g_N(x, y) \geq T_L$$
 - g_{NL} 包含 g_{NH} 的所有非零元素

坎尼（Canny）边缘检测器



4. 滞后阈值

- 去掉 g_{NL} 中和 g_{NH} 重复的点

$$g_{NL}(x, y) = g_{NL}(x, y) - g_{NH}(x, y)$$

- 弱边缘点

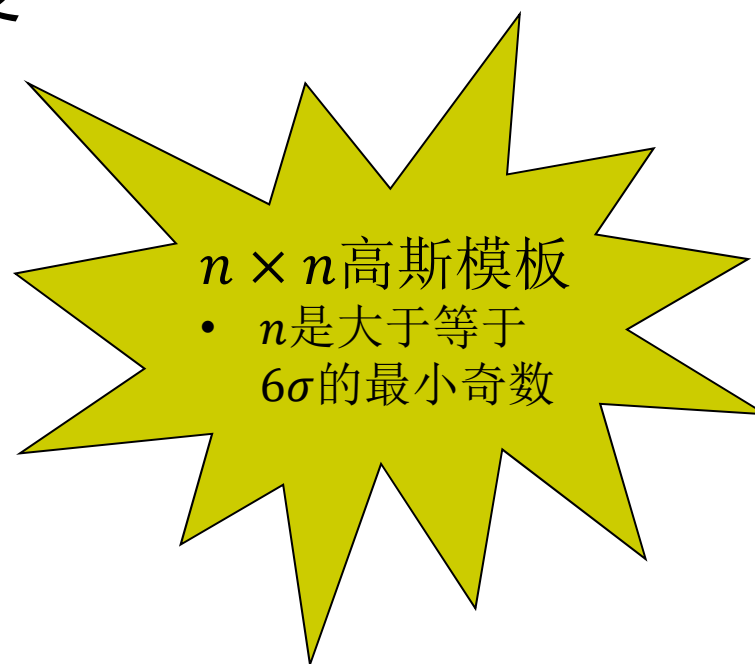
5. 连通性分析

- a. 遍历 g_{NH} 中的每一个点 p
 - ① 保留 g_{NL} 中和 p 连通（例如8连通）的点
- b. 去掉 g_{NL} 剩余的点
- c. 合并 g_{NH} 和 g_{NL}

坎尼（Canny）边缘检测器

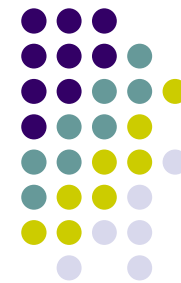


1. 高斯函数平滑输入图像
2. 计算图像的梯度
 - 梯度大小、梯度角度
3. 非最大抑制
 - 得到细边缘
4. 滞后阈值
 - 检测边缘
5. 连通性分析
 - 连接边缘

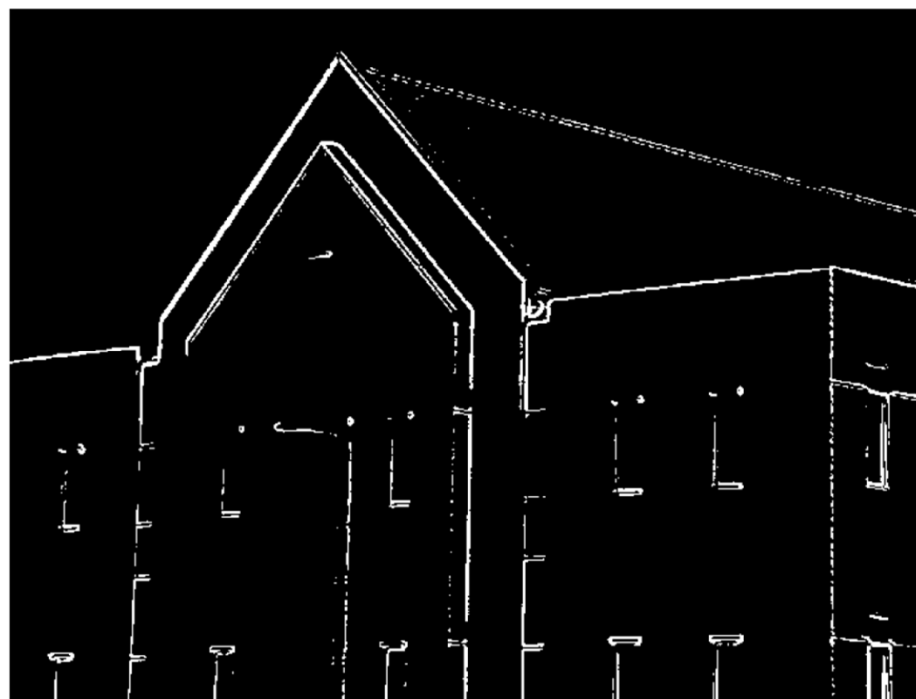


举例

- 基本边缘检测
 - 高斯平滑图像→梯度阈值化



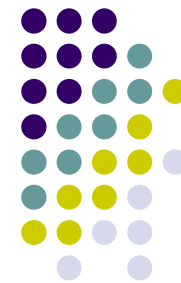
原图



梯度阈值化

举例

- 高级边缘检测



Marr-Hildreth边缘检测器



坎尼边缘检测器

举例

- 高级边缘检测



屋瓦产生的边缘完全消除



Marr-Hildreth边缘检测器



坎尼边缘检测器

举例

- 基本边缘检测



原图



梯度阈值化

举例

- 高级边缘检测

连通性更好
噪声更少



Marr-Hildreth边缘检测器



坎尼边缘检测器



提纲



- 基础知识
- 点、线、边缘检测
 - 背景知识
 - 孤立点的检测
 - 线检测
 - 边缘模型
 - 基本边缘检测
 - 高级边缘检测
- 边缘连接和边界检测

背景



- 边缘检测的结果不完美
 - 噪声
 - 不均匀照明导致的边缘间断
 - 虚假的灰度值不连续
- 边缘连接
 - 将边缘像素组合成有意义的边缘或区域边界
 1. 局部处理
 2. 区域处理
 3. 全局处理（使用霍夫变换）

提纲

- 边缘连接和边界检测
 - 局部处理
 - 区域处理
 - 全局处理



局部处理



1. 分析每个候选点 (x, y) 邻域内像素的特点
2. 将依据某准则相似的点连接起来

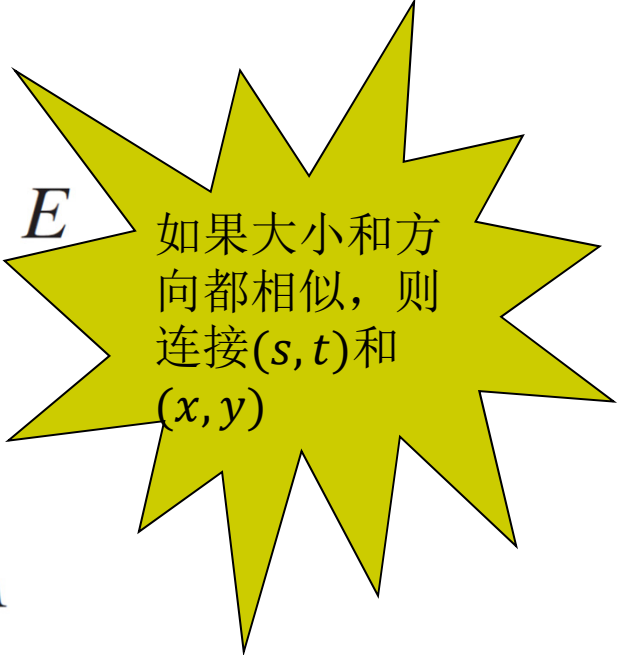
a. 基于梯度大小判断相似

$$|M(s, t) - M(x, y)| \leq E$$

- (s, t) 在 (x, y) 的邻域内

b. 基于梯度方向判断相似

$$|\alpha(s, t) - \alpha(x, y)| \leq A$$



如果大小和方向都相似，则连接 (s, t) 和 (x, y)

局部处理



- 简化算法（计算简单）

1. 计算输入图像 $f(x, y)$ 的梯度大小和方向
 - 梯度大小 $M(x, y)$, 梯度方向 $a(x, y)$
2. 依据下式生产二值图像

$$g(x, y) = \begin{cases} 1 & \text{if } M(x, y) > T_M \text{ AND } \alpha(x, y) = A \pm T_A \\ 0 & \text{otherwise} \end{cases}$$

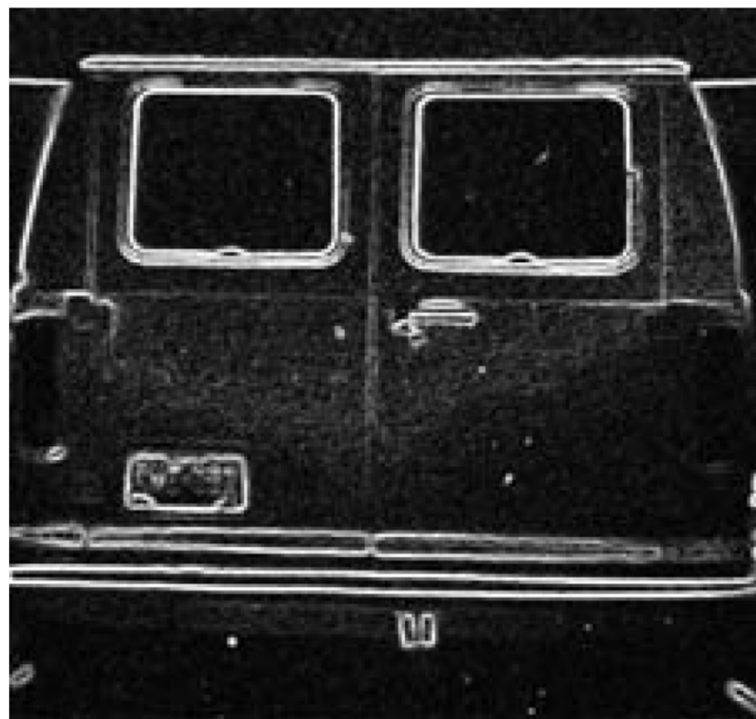
- T_M 为阈值、 A 为特定角度、 T_A 为允许的带宽
3. 逐行扫描, 填充长度不超过 K 的空隙
 4. 以角度 θ 旋转 $g(x, y)$, 重复第3步, 再反旋转

举例

- 寻找车牌



汽车尾部图像



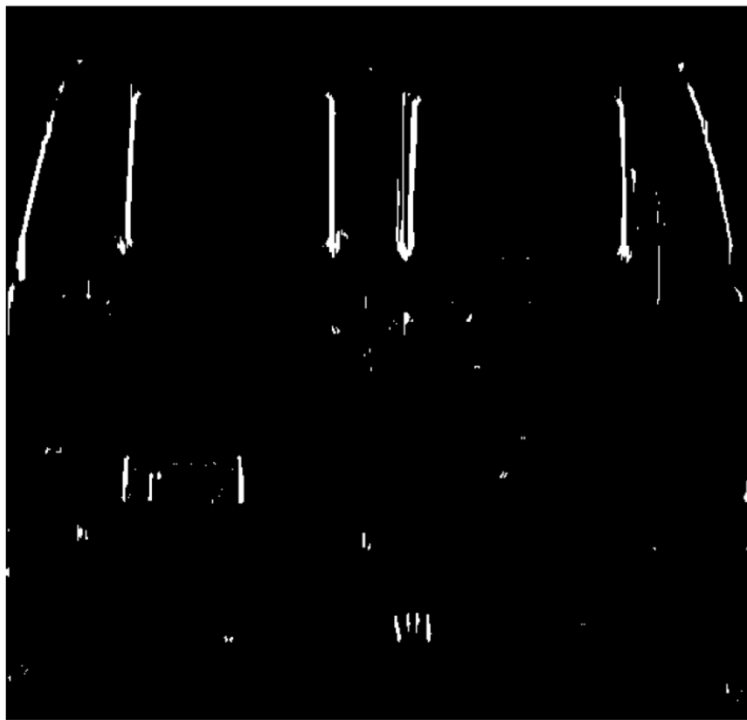
梯度大小图像

举例

- 寻找车牌



水平连接的边缘像素



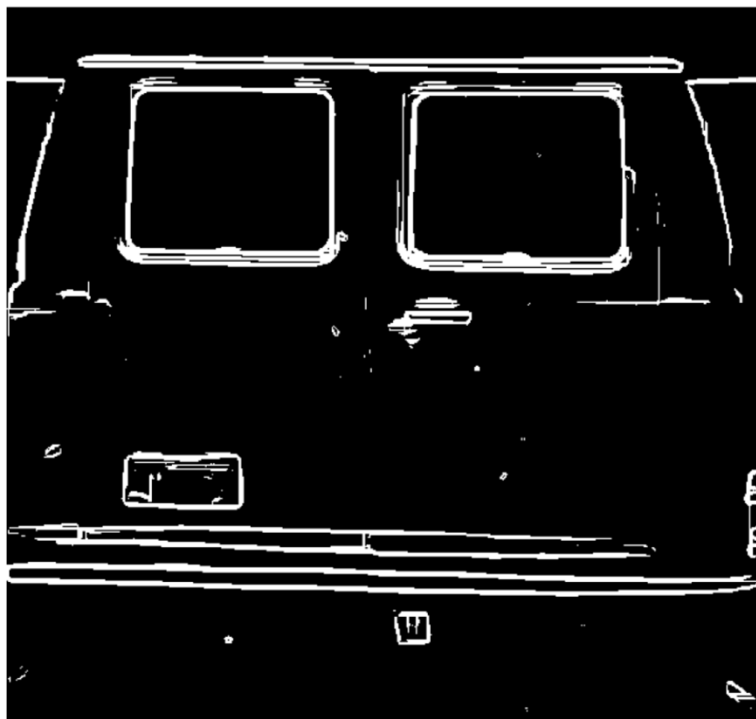
垂直连接的边缘像素

举例

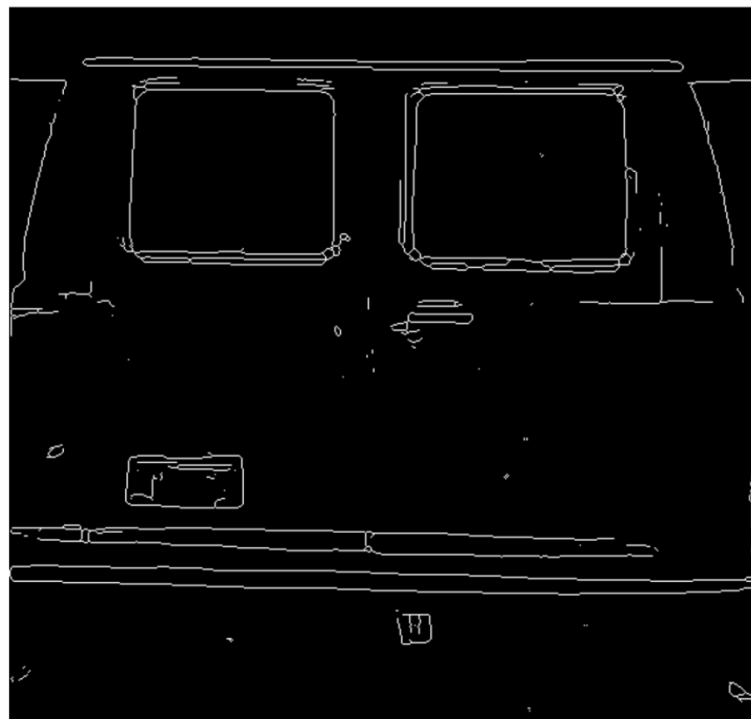
- 寻找车牌

采用比较大的
夹角范围

美国车牌的
长宽比是2:1



合并后的图像



细化后的图像



提纲

- 边缘连接和边界检测
 - 局部处理
 - 区域处理
 - 全局处理



区域处理

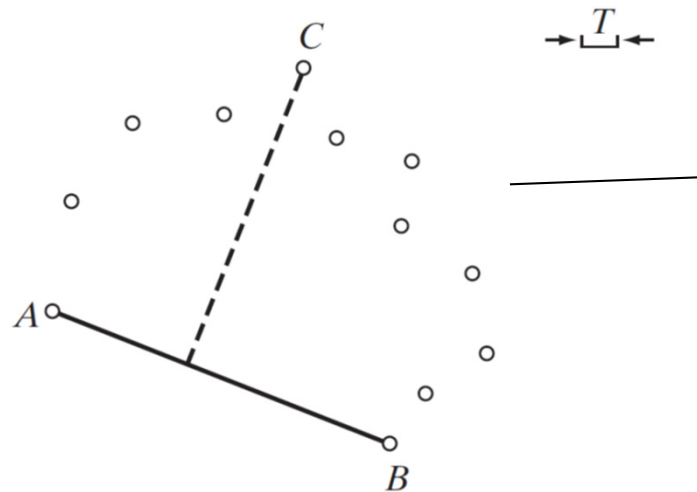


- 前提
 - 感兴趣区域的位置已知
 - 预先知道属于边界的像素点
- 目标：基于区域连接像素，近似区域边界
- 方法
 - 函数近似
 - 为已知点拟合一条2维曲线
 - 多边形近似
 - 实现容易、捕捉基本形状特征、表示简单

举例



- 已知 A 、 B 是曲线的端点

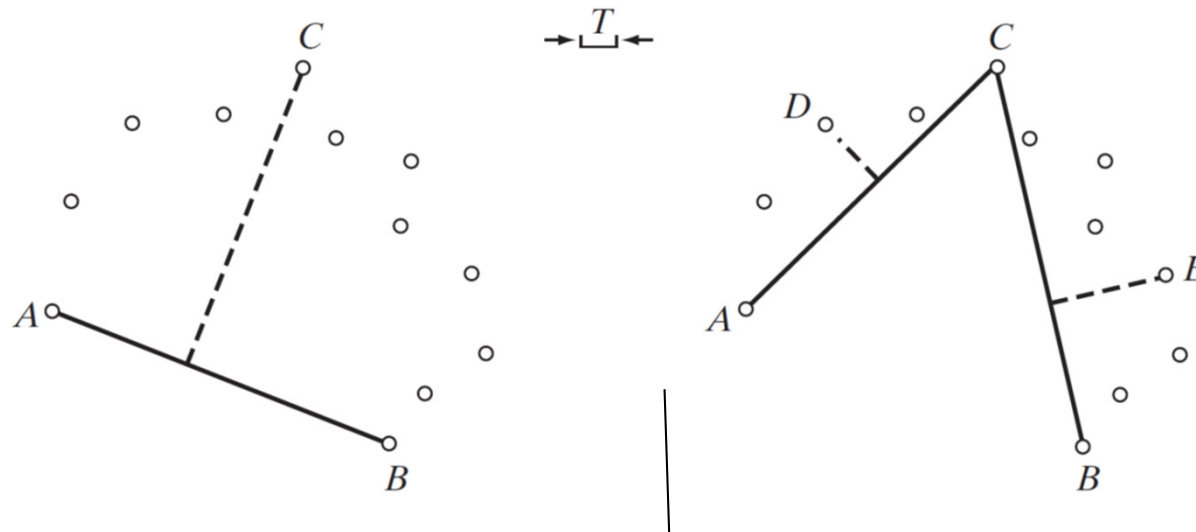


- 1、用直线连接 A 、 B
- 2、计算所有点离直线 AB 的距离
- 3、找到最远的点 C
- 4、如果距离大于阈值 T ，把 C 当做一个顶点

举例



- 已知 A 、 B 是曲线的端点

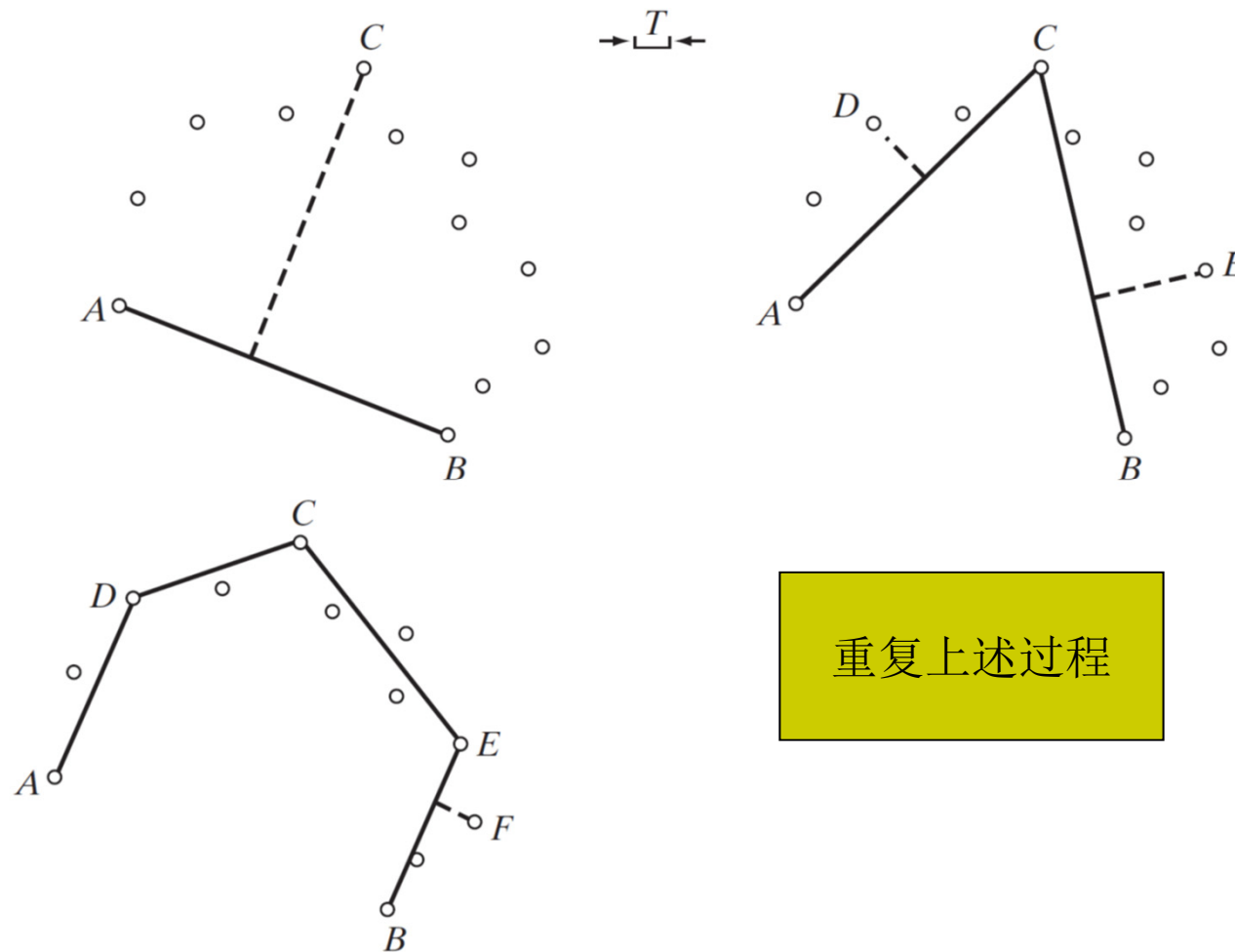


- 1、用直线连接 A 、 C
- 2、计算 AC 之间的点离直线 AC 的距离
- 3、找到最远的点 D
- 4、如果距离大于阈值 T ，把 D 当做一个顶点

举例



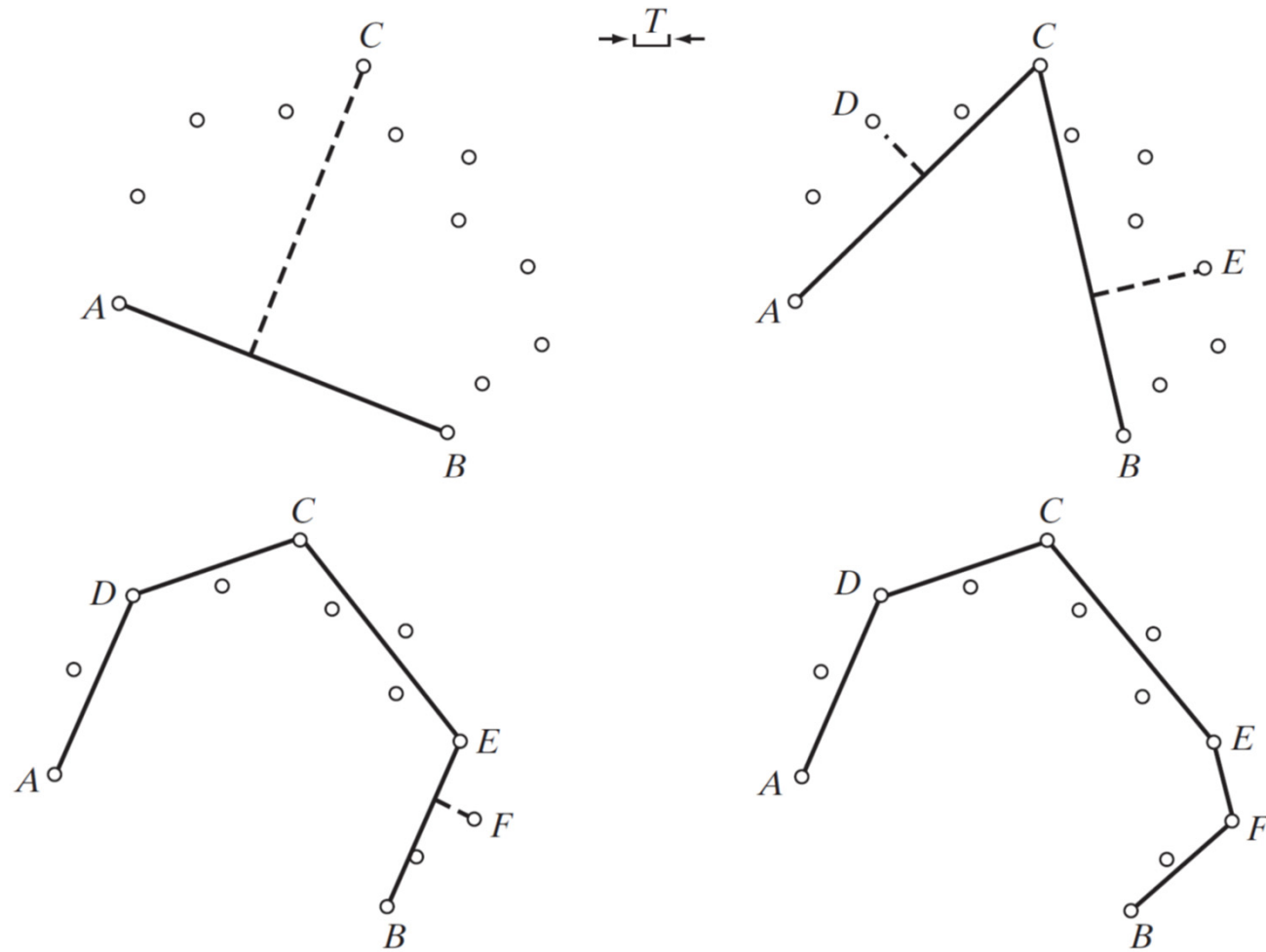
- 已知A、B是曲线的端点



举例



- 已知A、B是曲线的端点



算法设计



- 前提
 - 两个起始点
 - 所有的点必须排序
 - 顺时针、逆时针
- 判断曲线类型
 - 边界线段（开放曲线）
 - 存在两个间距较大的连续点（可作为起始点）
 - 边界（闭合曲线）
 - 连续点之间的距离比较均匀
 - 两端的点为起始点

区域处理算法



1. 令 P 是一个已排序、不重复的二值图像中的序列。指定两个起始点 A 和 B 。它们是多边形的两个起始顶点。
2. 指定一个阈值 T ，以及两个空堆栈“开” (OPEN) 和“闭” (CLOSED)。
3. 如果 P 中的点对应于一条闭合曲线，则把 B 放到“开”和“闭”中，并把 A 放到“开”中。
如果对应于一条开放曲线，则把 A 放到“开”中，而把 B 放到“闭”。
4. 计算从“闭”中最后一个顶点到“开”中最后一个顶点的线的参数。

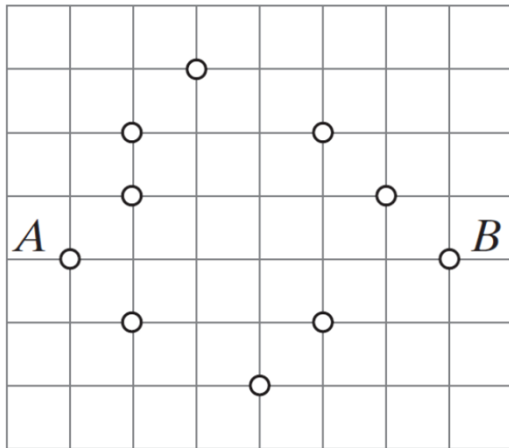
区域处理算法



5. 寻找属于序列 P 、且在步骤4中直线的两个顶点之间的点；计算这些点与直线的距离，选择具有最大距离 D_{max} 的点 V_{max} 。
6. 如果 $D_{max} > T$ ，则把 V_{max} 作为一个新顶点放在“开”堆栈的末尾。转到步骤4。
7. 否则，从“开”中移除最后一个顶点，并把它作为“闭”的最后一个顶点插入。
8. 如果“开”非空，转到步骤4。
9. 否则，退出。“闭”中的顶点就是拟合 P 中的点的多边形的顶点。

举例

- 闭合曲线
- 顺时针排序
- A 、 B 为起点



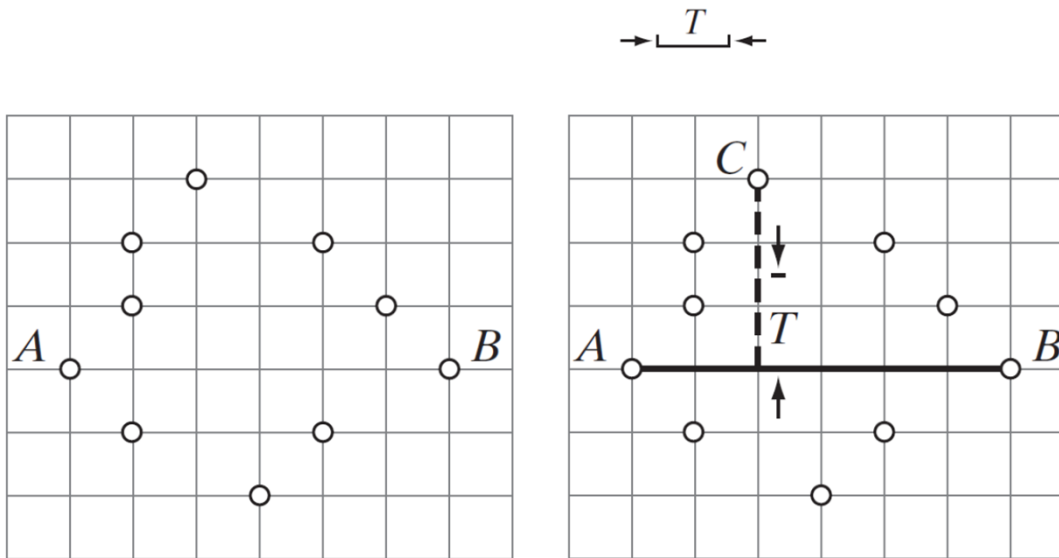
CLOSED	OPEN	Curve segment processed	Vertex generated
B	B, A	—	A, B

举例



- 闭合曲线
- 顺时针排序
- A 、 B 为起点

CLOSED	OPEN	Curve segment processed	Vertex generated
B B	B, A B, A	— (BA)	A, B C



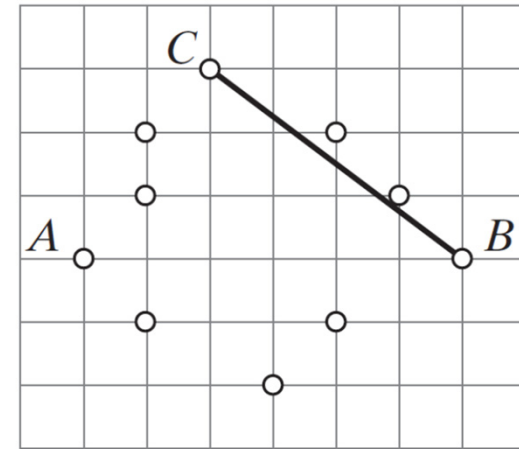
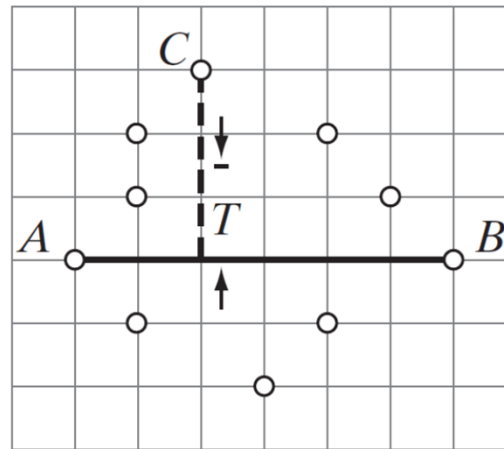
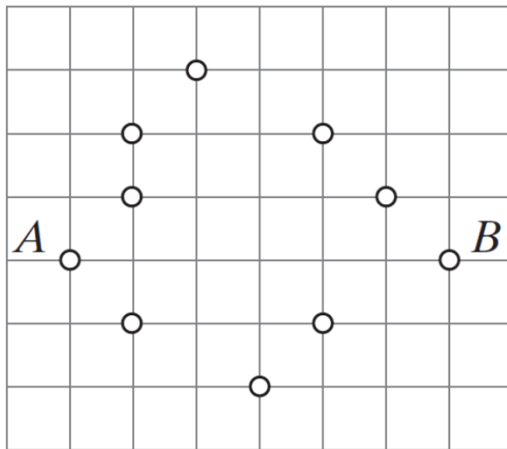
举例



- 闭合曲线
- 顺时针排序
- A 、 B 为起点

CLOSED	OPEN	Curve segment processed	Vertex generated
B	B, A	—	A, B
B	B, A	(BA)	C
B	B, A, C	(BC)	—

→ T ←



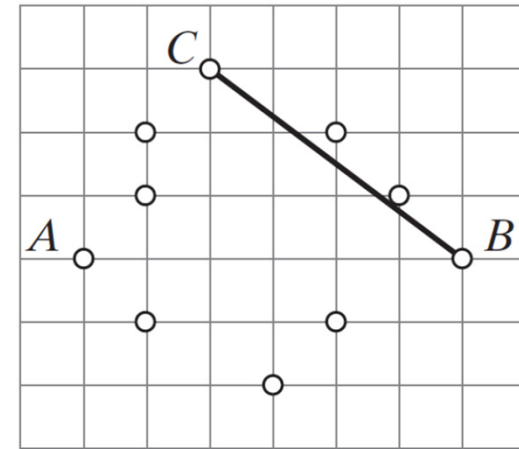
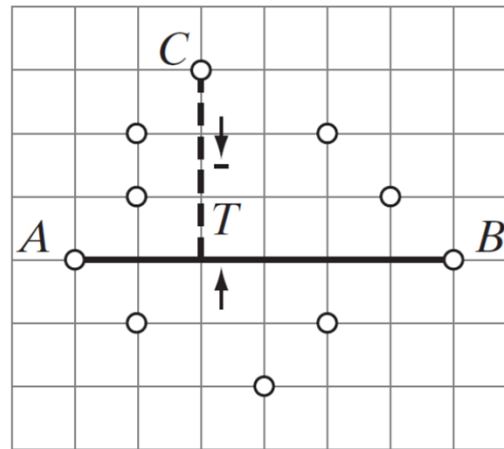
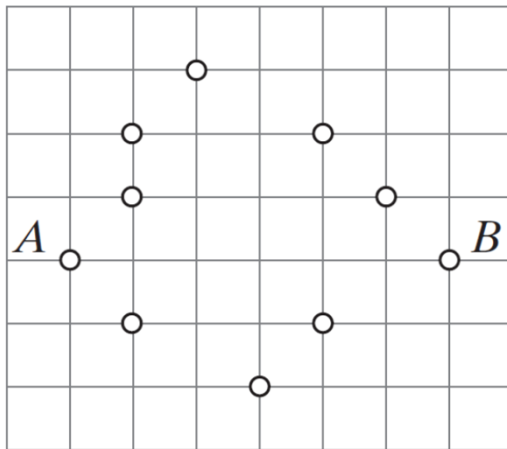
举例



- 闭合曲线
- 顺时针排序
- A 、 B 为起点

CLOSED	OPEN	Curve segment processed	Vertex generated
B	B, A	—	A, B
B	B, A	(BA)	C
B	B, A, C	(BC)	—
B, C	B, A	(CA)	—

$\rightarrow T \leftarrow$



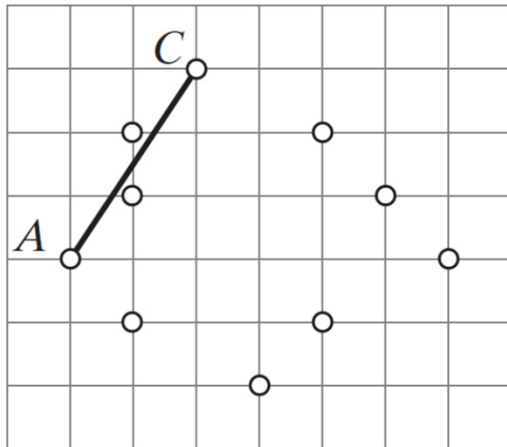
举例



- 闭合曲线
- 顺时针排序
- A 、 B 为起点

CLOSED	OPEN	Curve segment processed	Vertex generated
B	B, A	—	A, B
B	B, A	(BA)	C
B	B, A, C	(BC)	—
B, C	B, A	(CA)	—

→ T ←



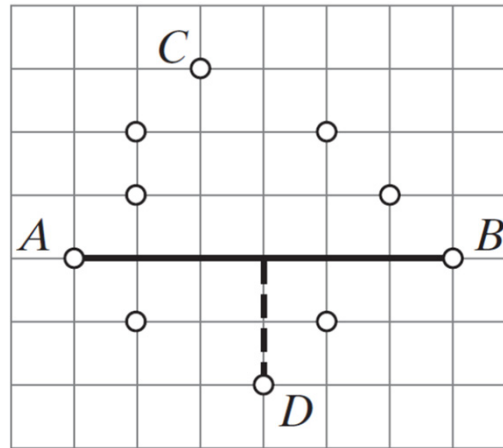
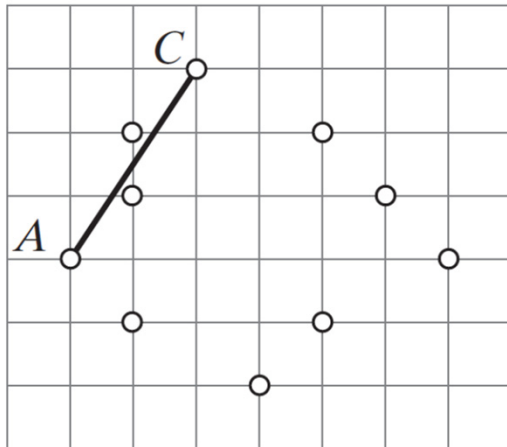
举例



- 闭合曲线
- 顺时针排序
- A 、 B 为起点

CLOSED	OPEN	Curve segment processed	Vertex generated
B	B, A	—	A, B
B	B, A	(BA)	C
B	B, A, C	(BC)	—
B, C	B, A	(CA)	—
B, C, A	B	(AB)	D

$\rightarrow T \leftarrow$



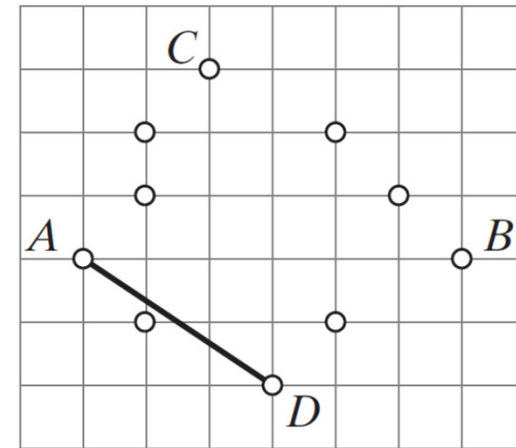
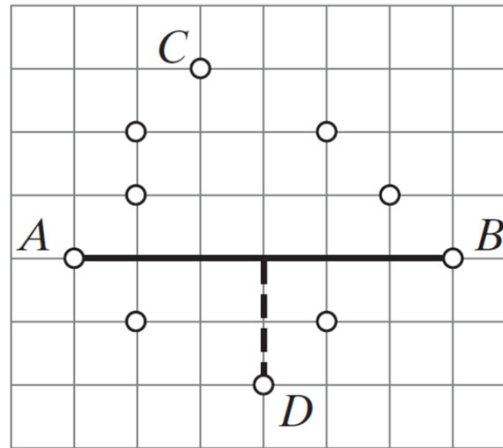
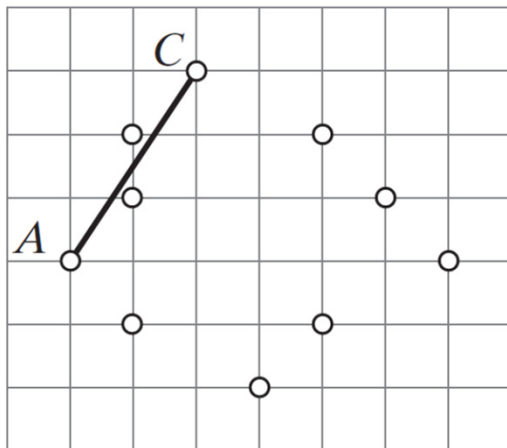
举例



- 闭合曲线
- 顺时针排序
- A、B为起点

CLOSED	OPEN	Curve segment processed	Vertex generated
<i>B</i>	<i>B, A</i>	—	<i>A, B</i>
<i>B</i>	<i>B, A</i>	(<i>BA</i>)	<i>C</i>
<i>B</i>	<i>B, A, C</i>	(<i>BC</i>)	—
<i>B, C</i>	<i>B, A</i>	(<i>CA</i>)	—
<i>B, C, A</i>	<i>B</i>	(<i>AB</i>)	<i>D</i>
<i>B, C, A</i>	<i>B, D</i>	(<i>AD</i>)	—

→ T ←



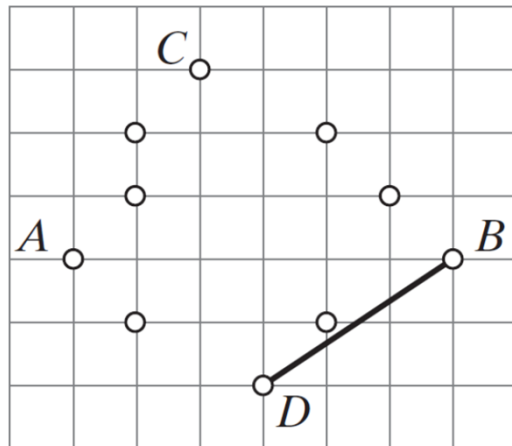
举例



- 闭合曲线
- 顺时针排序
- A 、 B 为起点

CLOSED	OPEN	Curve segment processed	Vertex generated
B	B, A	—	A, B
B	B, A	(BA)	C
B	B, A, C	(BC)	—
B, C	B, A	(CA)	—
B, C, A	B	(AB)	D
B, C, A	B, D	(AD)	—
B, C, A, D	B	(DB)	—

→ T ←



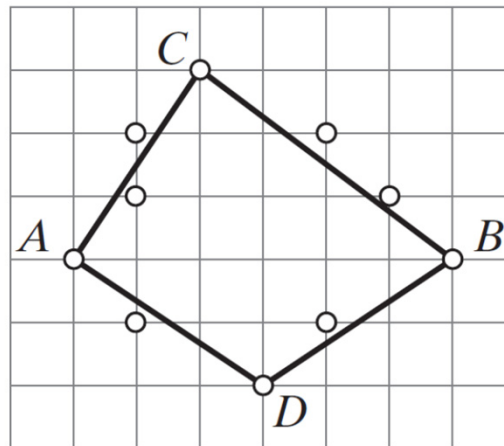
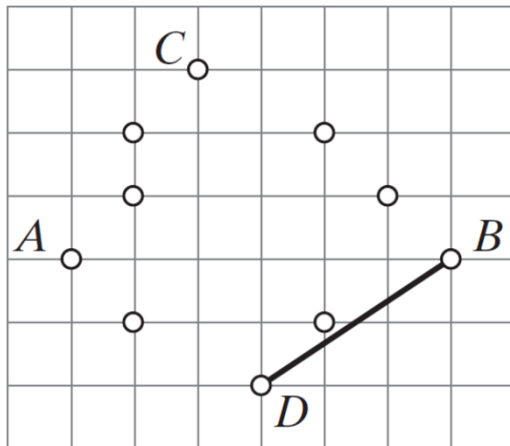


举例

- 闭合曲线
- 顺时针排序
- A、B为起点

CLOSED	OPEN	Curve segment processed	Vertex generated
<i>B</i>	<i>B, A</i>	—	<i>A, B</i>
<i>B</i>	<i>B, A</i>	(<i>BA</i>)	<i>C</i>
<i>B</i>	<i>B, A, C</i>	(<i>BC</i>)	—
<i>B, C</i>	<i>B, A</i>	(<i>CA</i>)	—
<i>B, C, A</i>	<i>B</i>	(<i>AB</i>)	<i>D</i>
<i>B, C, A</i>	<i>B, D</i>	(<i>AD</i>)	—
<i>B, C, A, D</i>	<i>B</i>	(<i>DB</i>)	—
<i>B, C, A, D, B</i>	Empty	—	—

→ T ←



提纲

- 边缘连接和边界检测
 - 局部处理
 - 区域处理
 - 全局处理



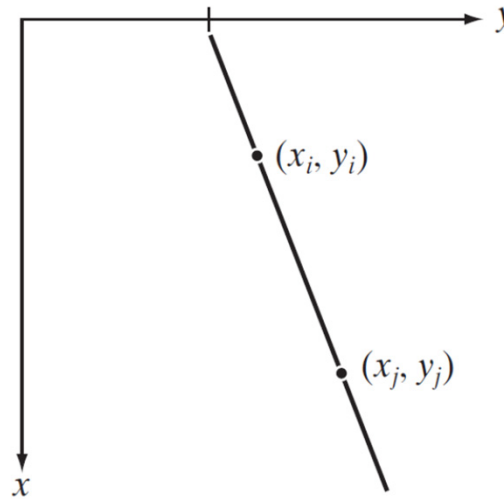


全局处理

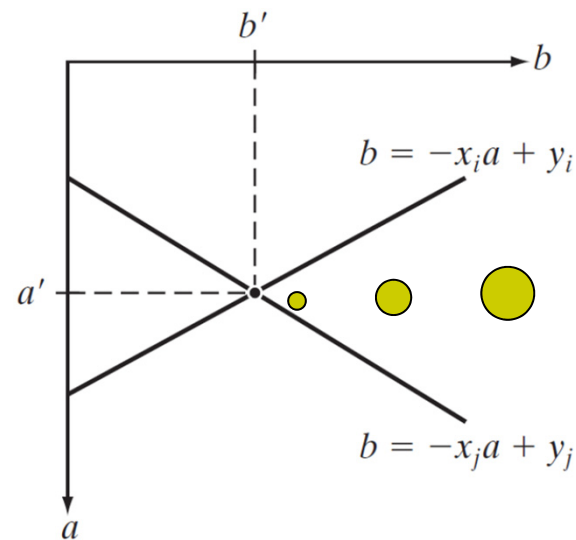
- 考虑没有边缘先验知识的情况
- 利用全局性质判断是否为边缘像素
 1. 指定感兴趣的几何形状
 2. 判断边缘像素集合是否满足该形状
- 问题：给定 n 个点，寻找共线的像素
 1. 考虑所有可能的直线 $n(n - 1)/2$
 2. 寻找靠近每一条直线的像素集合
 - 复杂度 $O(n^3)$

霍夫变换

- xy -平面
 - 直线方程
$$y_i = ax_i + b$$



- ab -平面
 - 参数方程
$$b = -ax_i + y_i$$
 - 每个点对应一条直线



交点对应于上图中直线的参数

霍夫变换

- xy -平面

- 直线方程

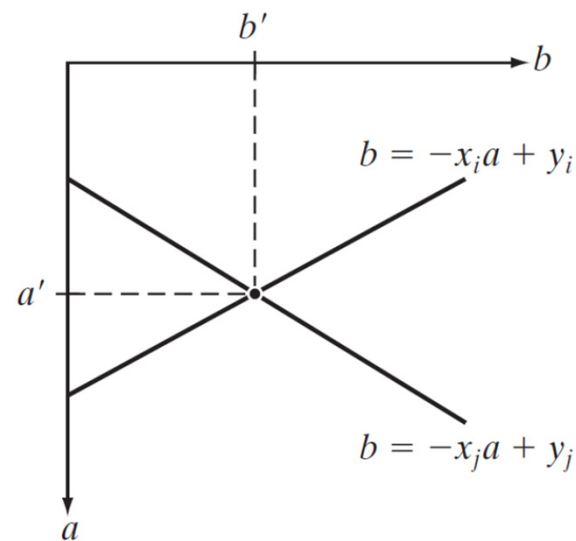
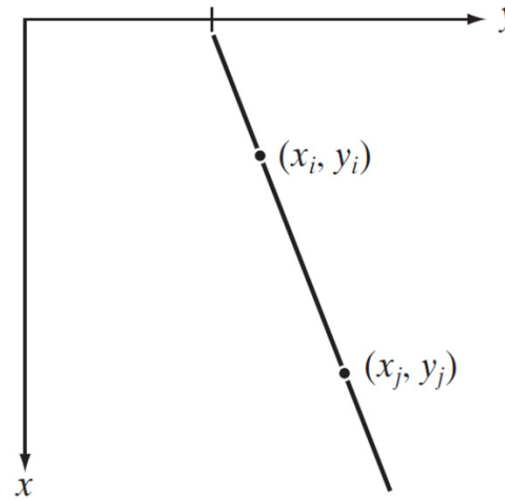
$$y_i = ax_i + b$$

- ab -平面

- 参数方程

$$b = -ax_i + y_i$$

- 每个点对应一条直线



简单的想法:

1. 画出所有 ab -平面中的直线
2. 寻找最多直线的交点

困难:

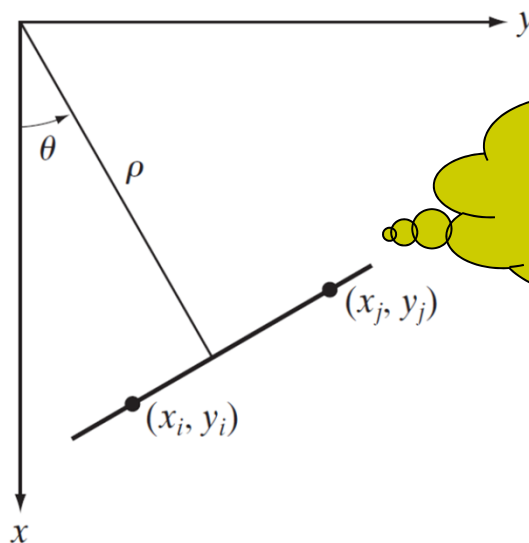
1. ab -平面是无界的

霍夫变换

- xy -平面

- 法线方程

$$x \cos \theta + y \sin \theta = \rho$$



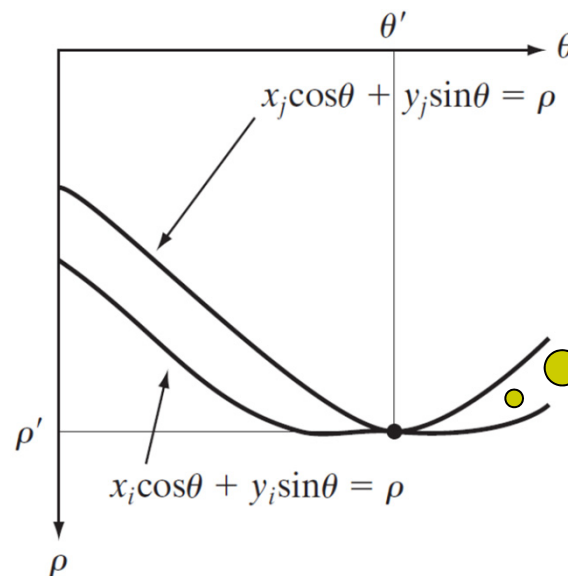
$$\langle (x, y), (\cos \theta, \sin \theta) \rangle = \rho$$

- $\rho\theta$ -平面

- 参数方程

$$\rho = x \cos \theta + y \sin \theta$$

- 每个点对应一条正弦曲线

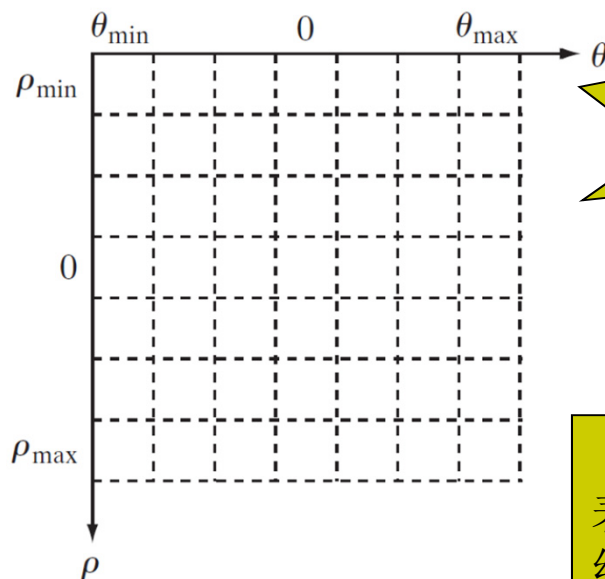


交点对应于上图中直线的参数



霍夫变换

- 划分累加单元
 - $-90^\circ \leq \theta \leq 90^\circ$
 - $-D \leq \rho \leq D$
 - D 是对角长度



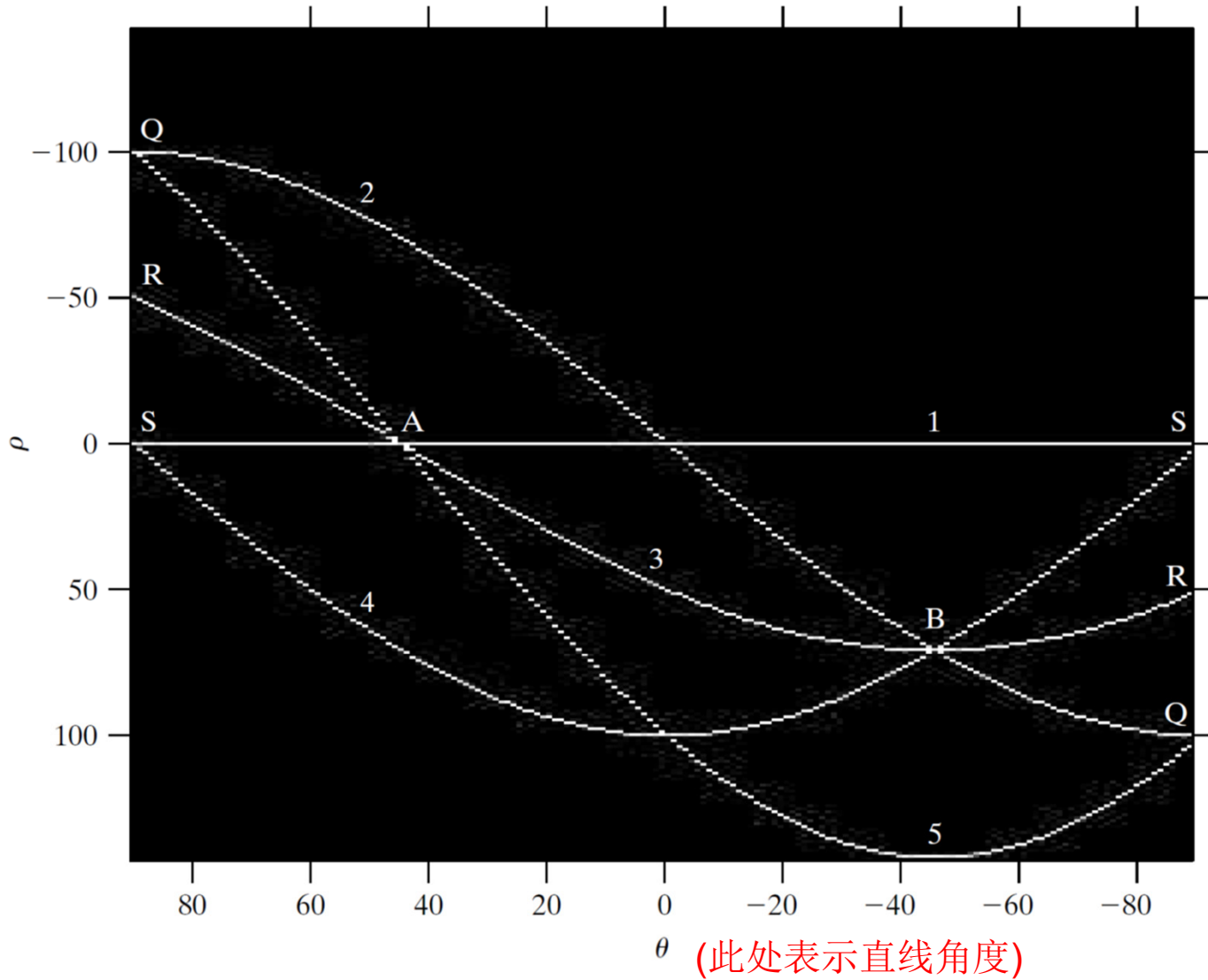
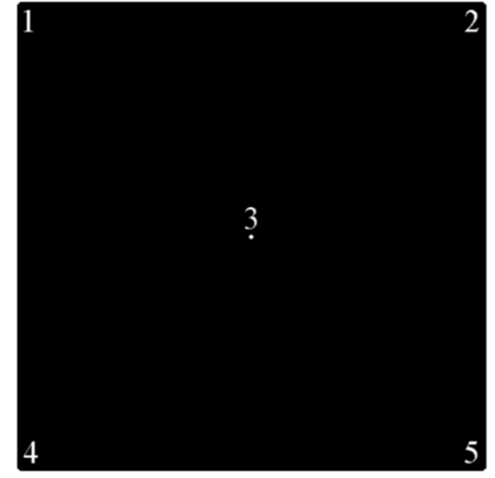
复杂度和
 n 呈线性

$A(i, j) = P$
表示有 P 个点属于直
线 $x \cos \theta_j + y \sin \theta_j =$
 ρ_i

- 统计每个单元内曲线的数目
 - (i, j) 位置单元内曲线数目记为 $A(i, j)$
 - (i, j) 位置单元对应的参数 (ρ_i, θ_j)
 - 计算 $\rho = x \cos \theta_j + y \sin \theta_j$, 并离散化

举例

A处有3个点相交
对应的参数是(0,45°)



B处有3个点相交
对应的参数是
(71, -45°)

Q, R, S在两端都出现

将霍夫变换用于边缘连接



1. 生成二值的边缘图像
 - 可采用之前介绍的任意算法
2. 划分 $\rho\theta$ -平面的累加单元
 - 粒度决定了精度、计算量
3. 统计每个累加单元的曲线数量
 - 寻找数值高的单元
4. 检验数值高累加单元对应的像素
 - 将距离小于某阈值的像素连接起来

可以扩展到直线以外的曲线

举例

- 寻找机场的主跑道
 - 中间位置、垂直方向



原图



举例

- 寻找机场的主跑道
 - 中间位置、垂直方向



原图

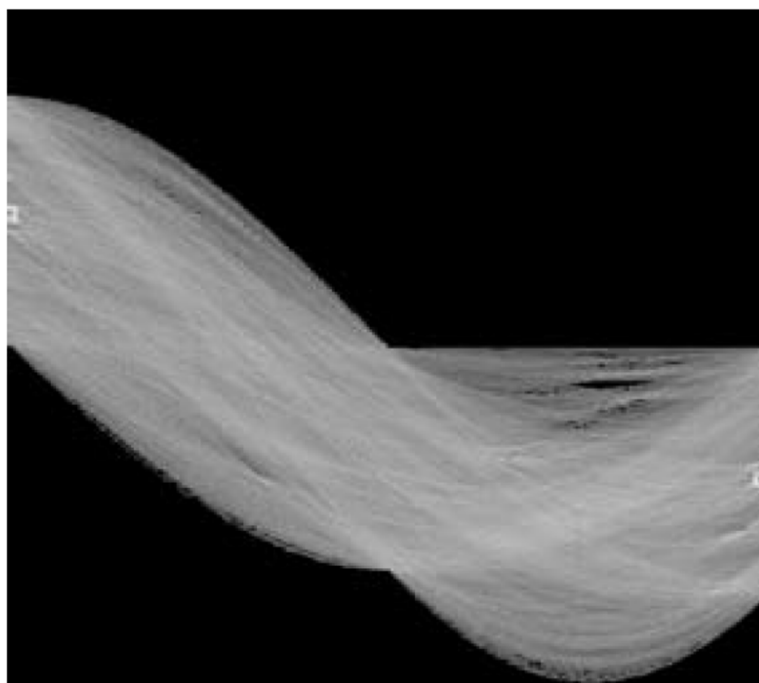


坎尼边缘检测器

举例

- 寻找机场的主跑道
 - 中间位置、垂直方向

方框为数值最大的单元



霍夫变换空间

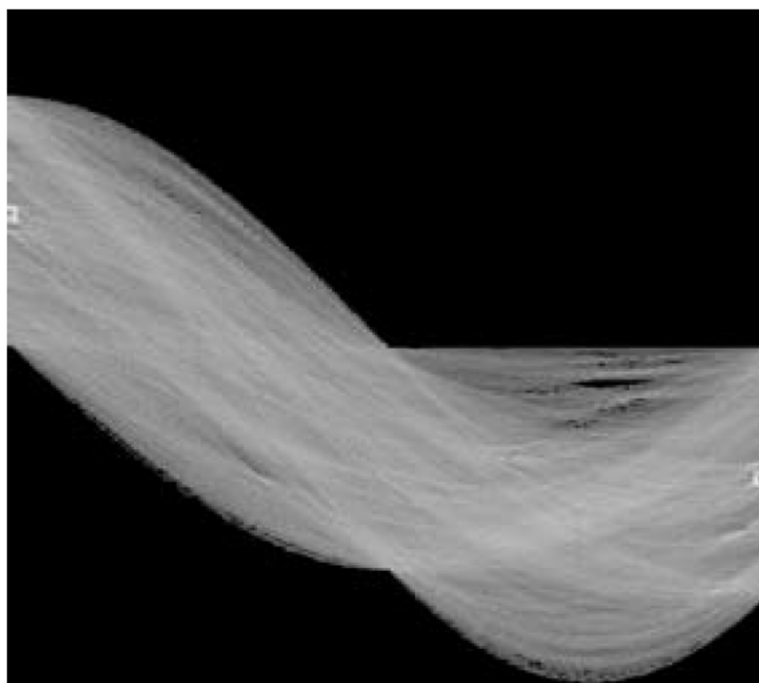


举例

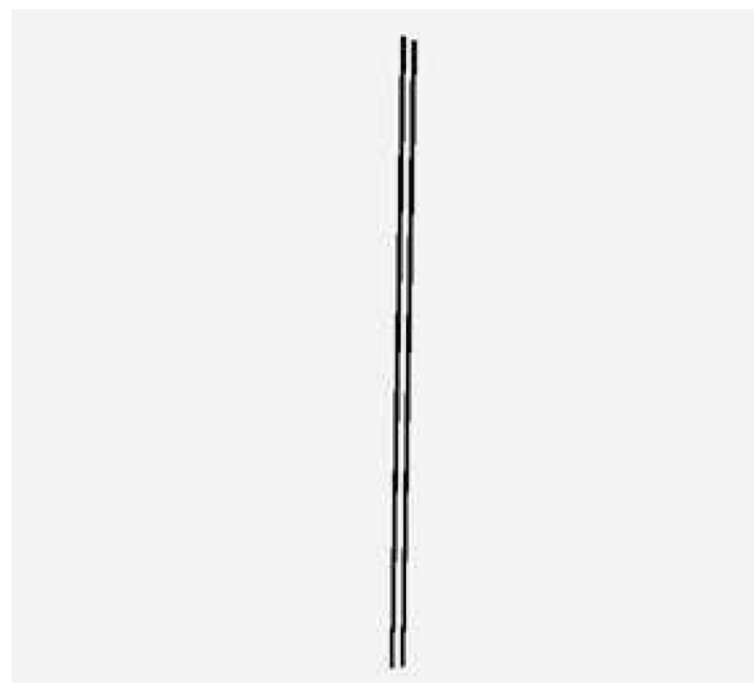
- 寻找机场的主跑道
 - 中间位置、垂直方向

方框为数值最大的单元

该单元对应的像素



霍夫变换空间



检测到的边缘

举例

- 寻找机场的主跑道
 - 中间位置、垂直方向



叠加到原图

