

《数字图像处理》

第11讲 图像分割 (1)

冯建江

清华大学 自动化系

2017.12.7

内 容

- 基础
- 点检测
- 线检测
- 边缘检测
- 霍夫变换

内 容

- 基础
- 点检测
- 线检测
- 边缘检测
- 霍夫变换

图像分割的定义



输入指纹图像



指纹分割结果
(指纹区为0, 背景区为1)



脊线分割结果
(脊线为0, 背景及谷线为1)

- 利用集合概念定义图像分割
- 令集合 R 表示整个图像区域。图像分割将 R 切分为 n 个子区域（子集） R_1, R_2, \dots, R_n ，这些子区域满足以下属性：
 1. $\bigcup_{i=1}^n R_i = R$
 2. 对 $i \neq j$, 有 $R_i \cap R_j = \emptyset$

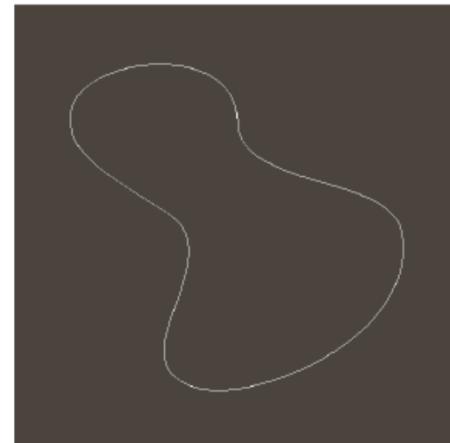
两种方法

- 两种图像分割方法：基于边缘和基于区域

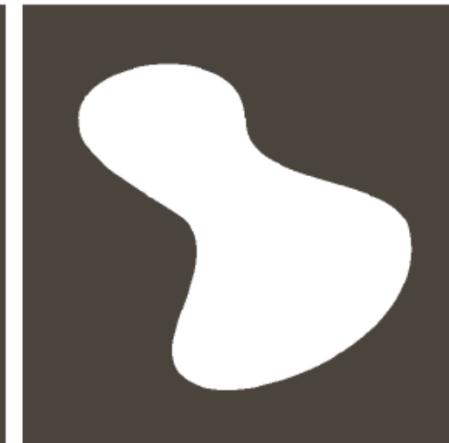
简单图像



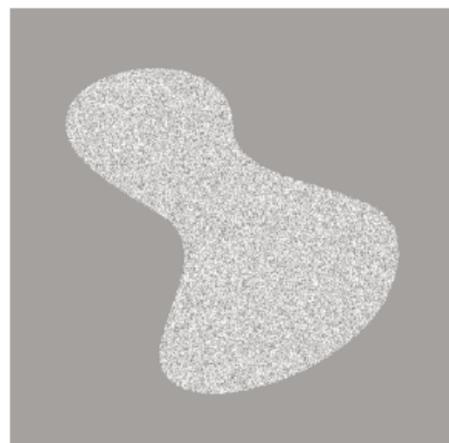
边缘检测结果



基于边缘的分割结果



有噪声图像



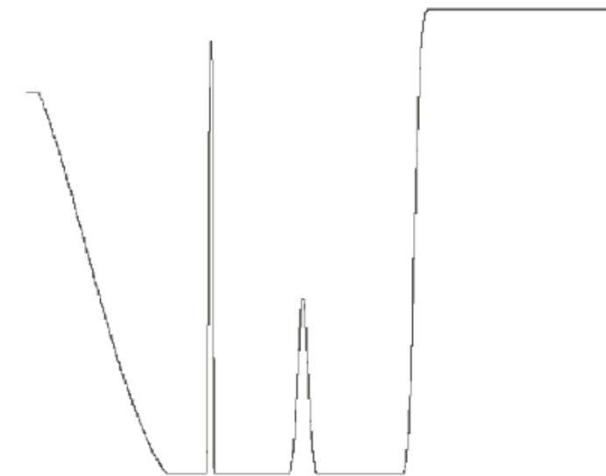
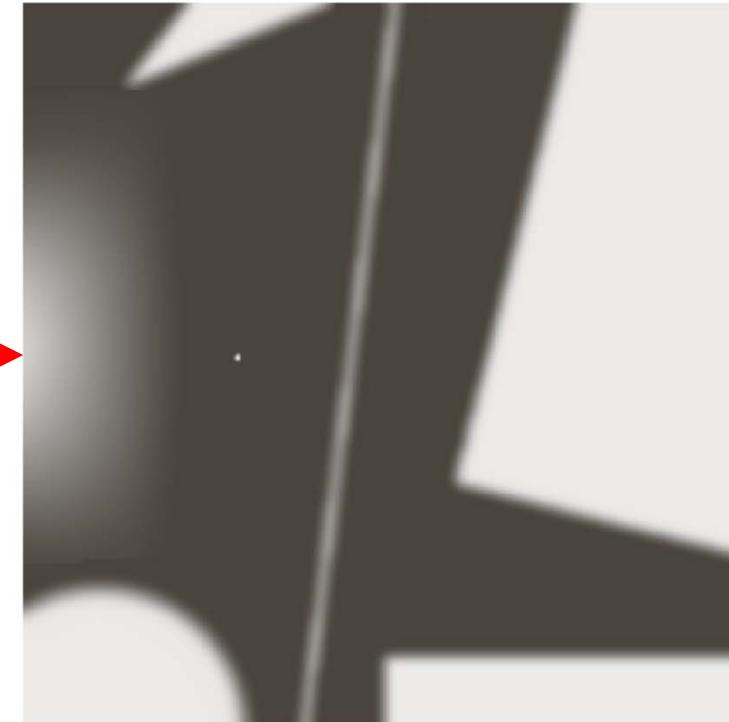
边缘检测结果



基于区域的分割结果



点、线、边缘



- 点、线、边缘是图像中灰度局部发生较大变化的地方
- 使用图像函数的一阶、二阶导检测灰度的局部变化
- 对于数字图像，只能近似计算导数

一阶、二阶导近似计算的要求

- 一阶导近似计算需满足：

1. 在灰度不变区为0
2. 在坡的起始和结束不为0
3. 在坡上不为0

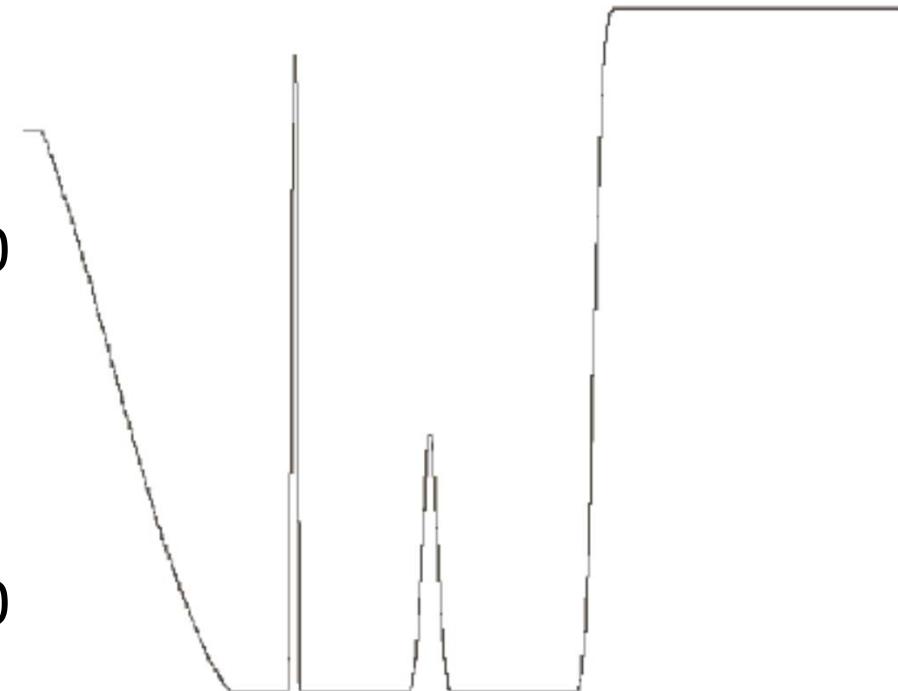
- 二阶导近似计算需满足：

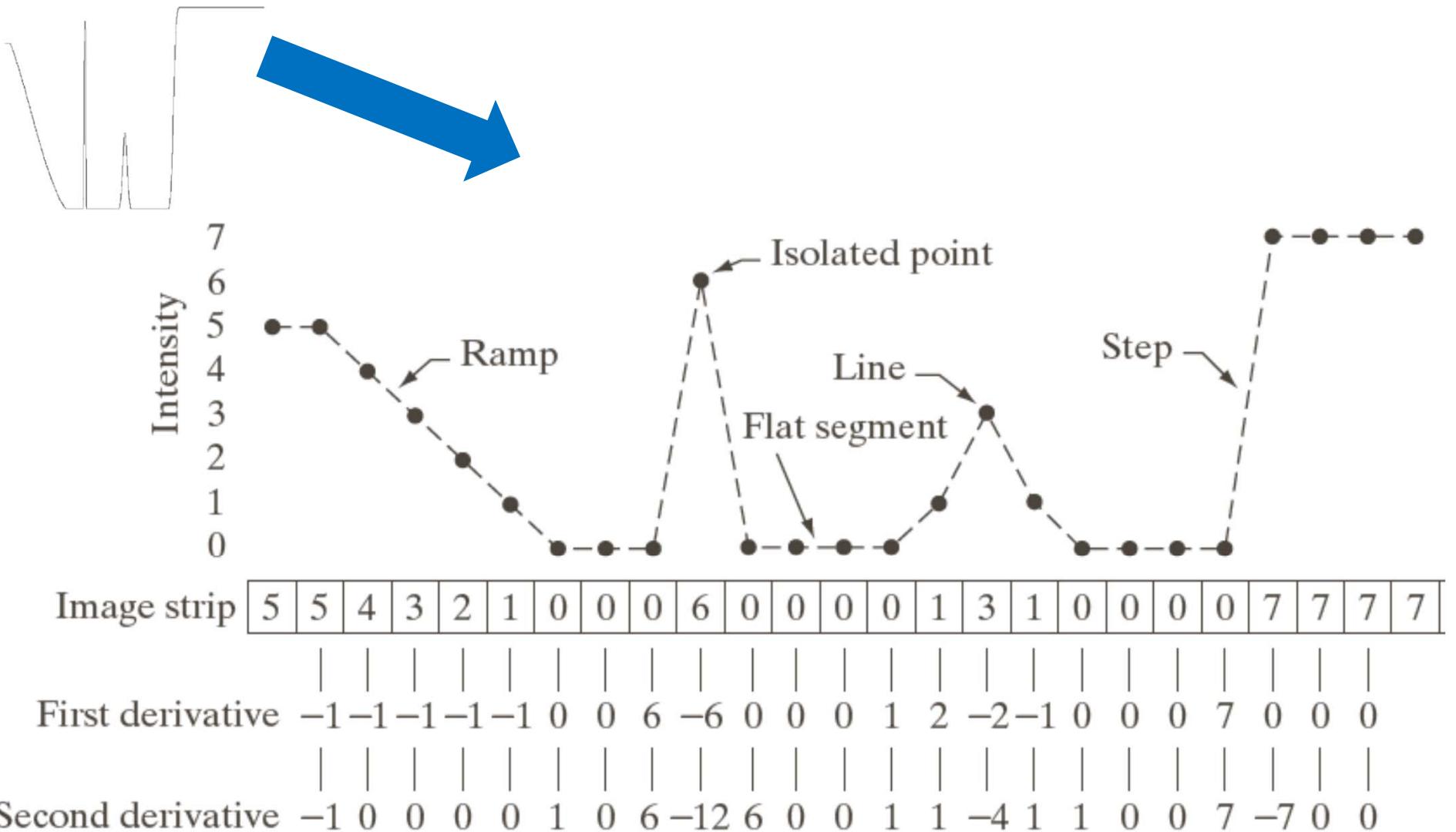
1. 在灰度不变区为0
2. 在坡的起始和结束不为0
3. 在坡上为0

- 一种常用的近似是（以一维信号为例）：

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

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





一阶、二阶导的特点

根据前面的例子，总结如下：

- 一阶导产生较粗的边缘（整个斜坡都不为0）
- 二阶导在斜坡和台阶处产生双边缘，而且边缘较细
- 二阶导对细节（细线、孤立点）的响应较强
- 二阶导的符号能确定变化是从亮到暗，还是从暗到亮

图像一阶、二阶导的计算

- 前面为了观察方便，使用一维信号（图像的一行像素）来分析一阶、二阶导。
- 对于图像，使用空域滤波器实现一阶、二阶导的计算。
- 要计算某位置的导数，将各系数与其对应像素值相乘，求和。
- 如果要得到图像全部位置的导数，可用滤波器对图像做卷积。

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

内 容

- 基础
- 点检测
- 线检测
- 边缘检测
- 霍夫变换

点的检测

- 利用二阶导检测图像中的点特征

- 拉普拉斯算子: $\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$

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

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

$$\nabla^2 f(x, y) = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

另一种离散近似:

$$\nabla^2 f(x, y)$$

$$= f(x+1, y+1) + f(x, y+1) + f(x-1, y+1) \\ f(x+1, y) + f(x-1, y) + f(x+1, y-1) \\ + f(x, y-1) + f(x-1, y-1) - 8f(x, y)$$

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

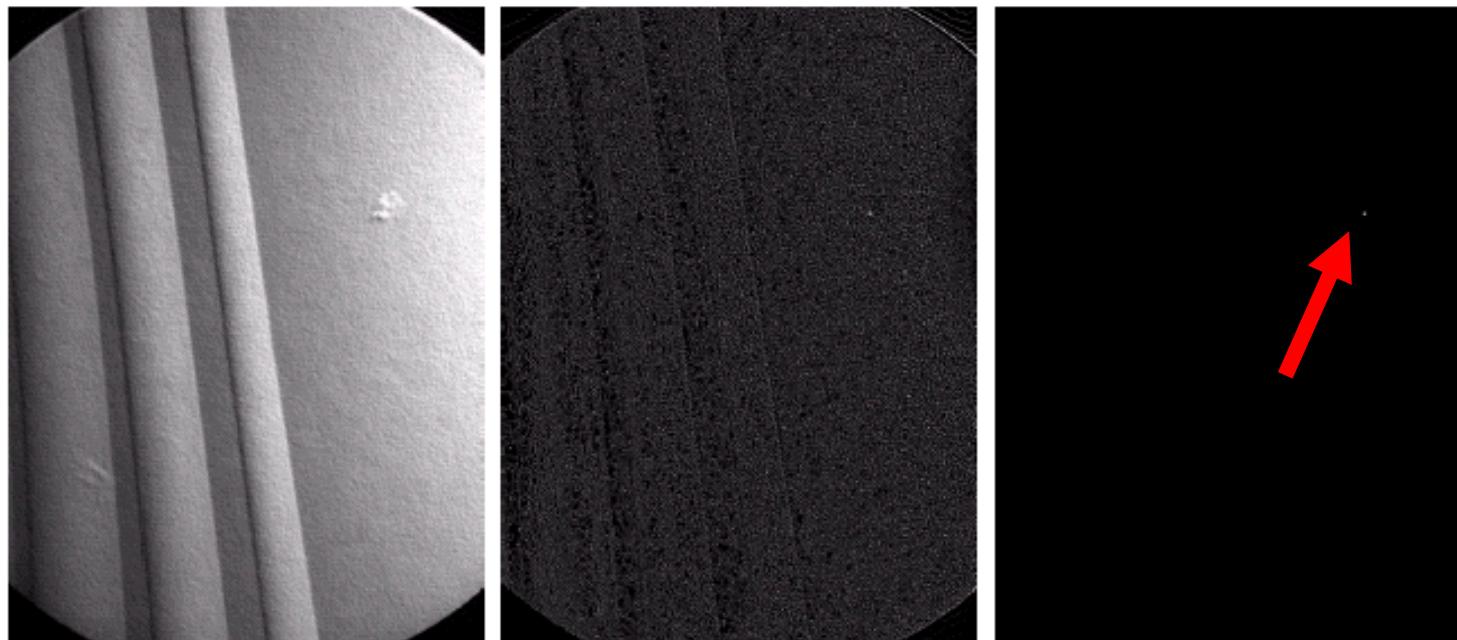
点的检测

- 将拉普拉斯算子 h 与输入图像 f 进行卷积得到响应图 R

$$R = f \star h$$

- 当 $|R(x, y)| \geq T$ 时，输出为 1；否则，输出为 0
- 拉普拉斯点检测器的物理意义：如果某像素与周围 8 近邻（或者 4 近邻）的差异超过某阈值，就认为是点特征。
- 在灰度恒定的区域，输出为 0

点检测例子

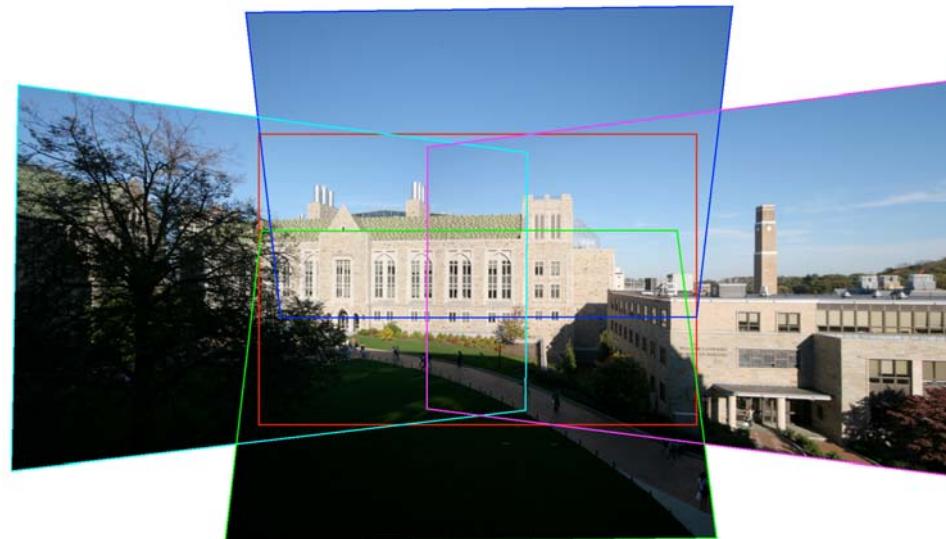


- 问题：检测飞机涡轮机叶片上的缺陷
- 左边为叶片的x光图像，右上方有一个很不明显的孔
- 利用拉普拉斯点检测器能检测出该孔

```
% ex1001_point
I = imread('..\data\Fig1004(b)(turbine blade black_dot).tif');
I = im2double(I);
h = [1 1 1; 1 -8 1; 1 1 1];
J = imfilter(I, h);
T = max(J(:))*0.9;
BW = J>=T;
close all
figure, imshow(I), ax(1)=gca;
figure, imshow(J,[ ]), ax(2)=gca;
figure, imshow(BW), ax(3)=gca;
linkaxes(ax);
```

点检测的应用

- 斑点检测（blob detection）是点检测的扩展，在不同尺度空间上查找点特征
- 著名的SIFT就是先检测出尺度空间中的斑点，然后用梯度直方图对周围区域进行描述
- SIFT等特征描述子的应用很广，包括图像拼接、图像检索、图像分类等



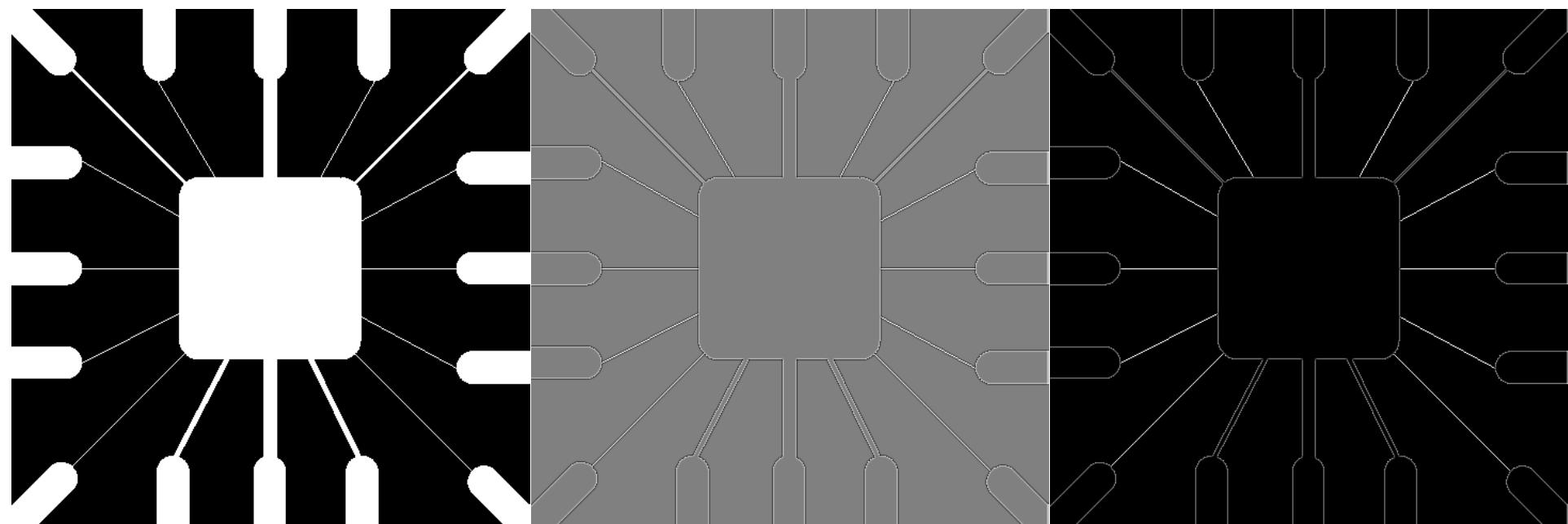
内 容

- 基础
- 点检测
- 线检测
- 边缘检测
- 霍夫变换

线的检测

- 一连串点组成直线
- 可用拉普拉斯算子检测线

-1	-1	-1
-1	8	-1
-1	-1	-1



输入图像

拉普拉斯图像

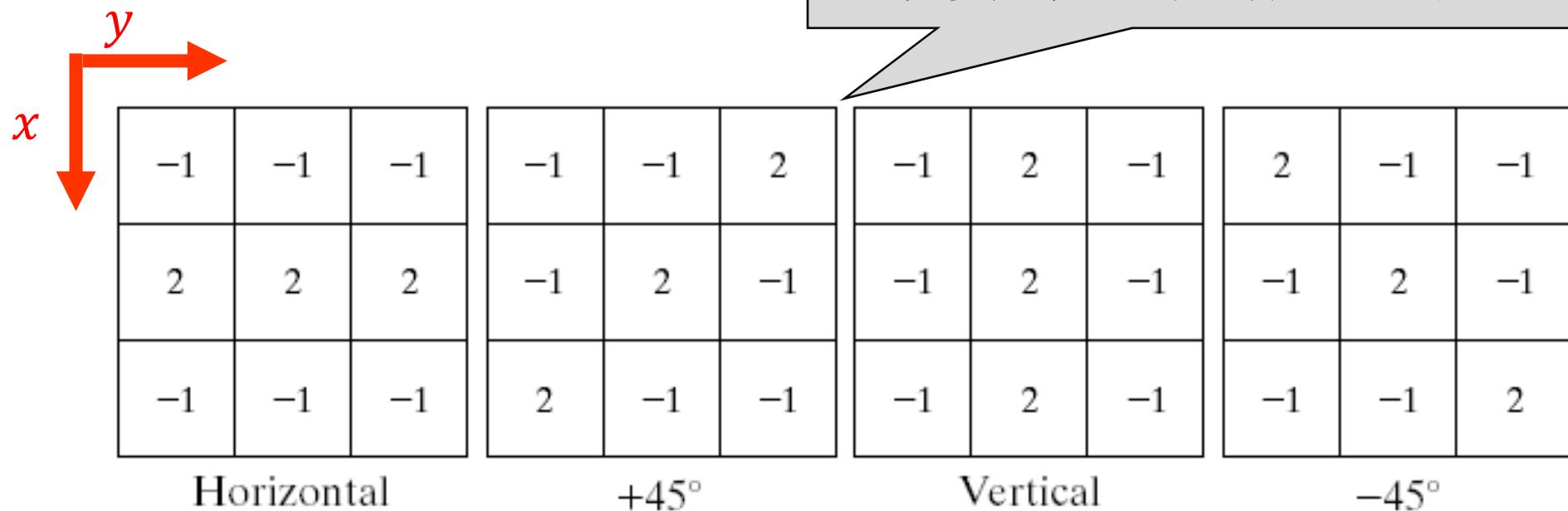
拉普拉斯
图像的正值

```
% ex1002_lineLaplacian
I = imread('..\data\Fig1005(a)(wirebond_mask).tif');
I = im2double(I);
h = -[1 1 1; 1 -8 1; 1 1 1];
J = imfilter(I, h);
J_positive = J;
J_positive(J<0) = 0;
close all
figure, imshow(I), ax(1)=gca;
figure, imshow(J,[ ]), ax(2)=gca;
figure, imshow(abs(J),[ ]), ax(3)=gca;
figure, imshow(J_positive,[ ]), ax(4)=gca;
linkaxes(ax);
```

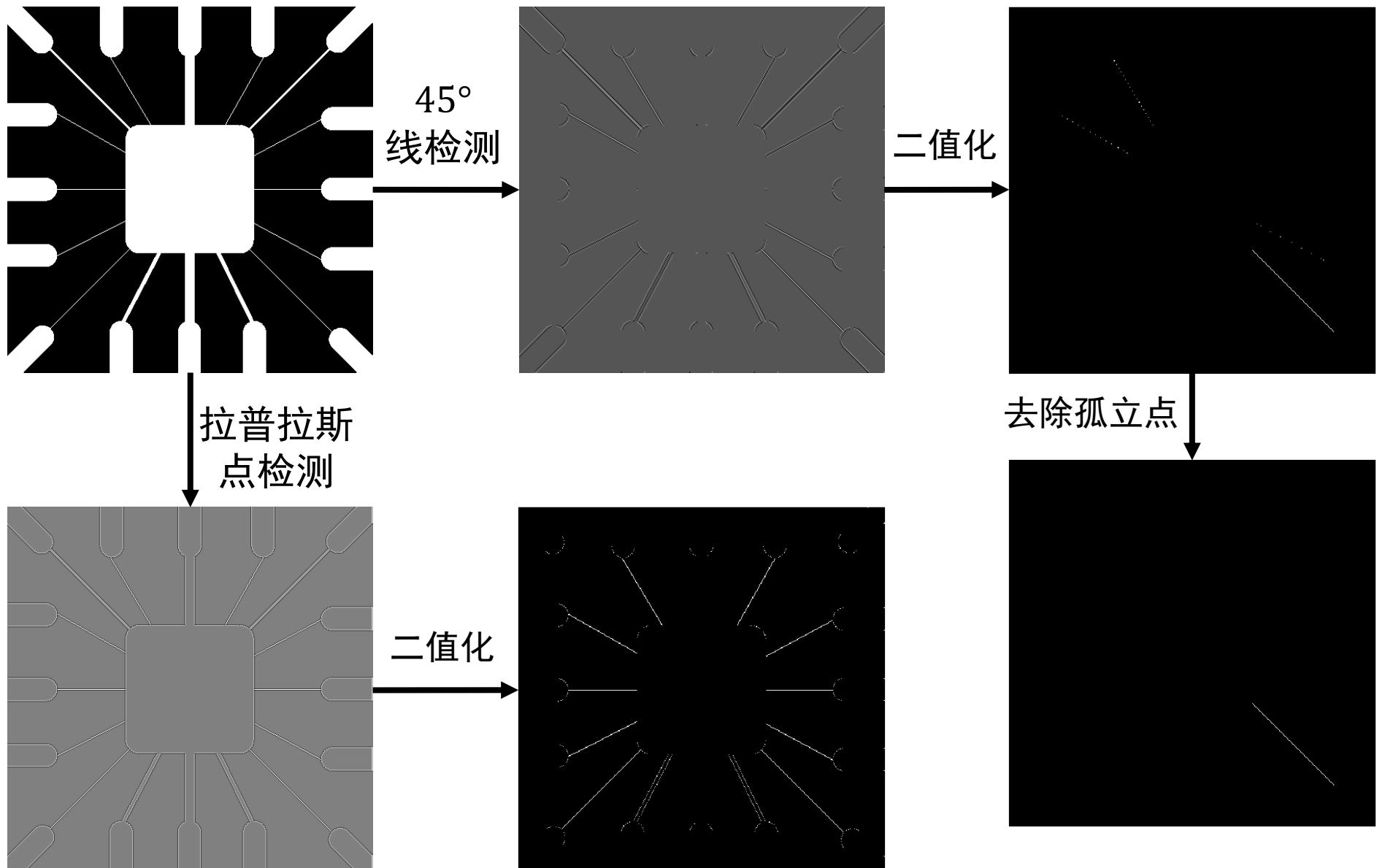
检测特定方向的线

- Laplacian算子是点检测器，对各种方向的线（宽度为1）的响应都大。
- 有些应用中，我们只想检测特定方向的线，或者想知道线的方向。
- 使用下面的线检测滤波器

- 过中心的特定方向的系数为2，其他系数为-1
- 每个滤波器的系数之和为0



点检测 vs 特定方向线检测



```
% ex1003_lineSpecificDirection
I = imread('..\data\Fig1005(a)(wirebond_mask).tif');
I = im2double(I);

h1 = -[1 1 1; 1 -8 1; 1 1 1];
J1 = imfilter(I, h1);
t1 = max(J1(:))*0.5;

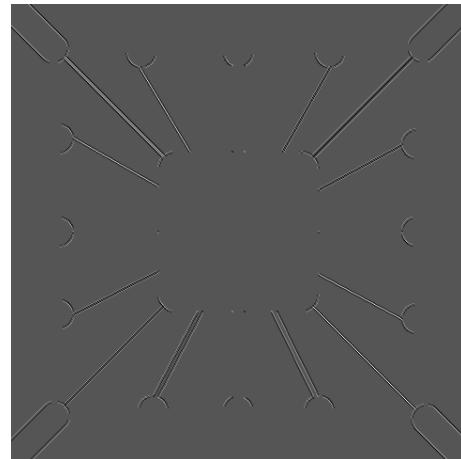
h2 = [2 -1 -1; -1 2 -1; -1 -1 2];
J2 = imfilter(I, h2);
t2 = max(J2(:))*0.5;
BW2 = bwmorph(J2>t2, 'clean');

close all
figure, imshow(I), ax(1)=gca;
figure, imshow(J1, []), ax(2)=gca;
figure, imshow(J2, []), ax(3)=gca;
figure, imshow(J1>t1), ax(4)=gca;
figure, imshow(J2>t2), ax(5)=gca;
figure, imshow(BW2), ax(6)=gca;
linkaxes(ax);
```

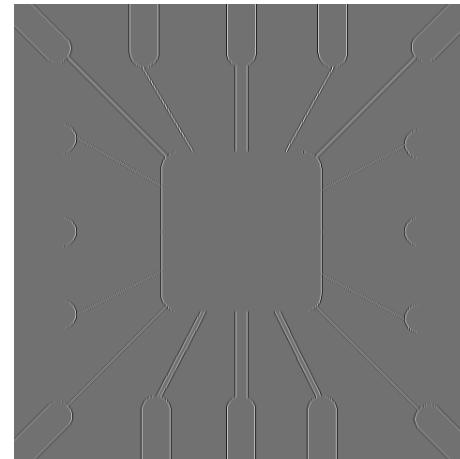
检测各方向的线



0°响应图



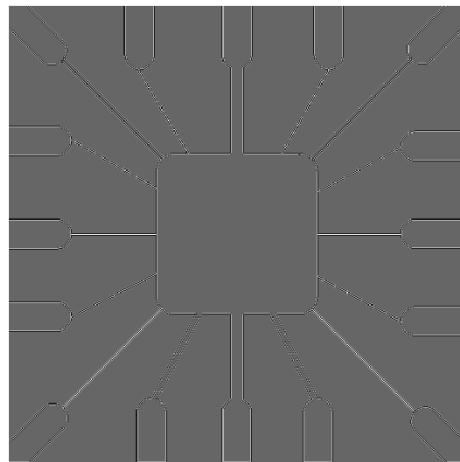
45°响应图



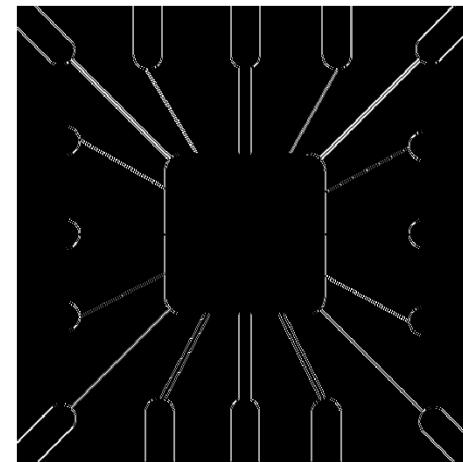
90°响应图



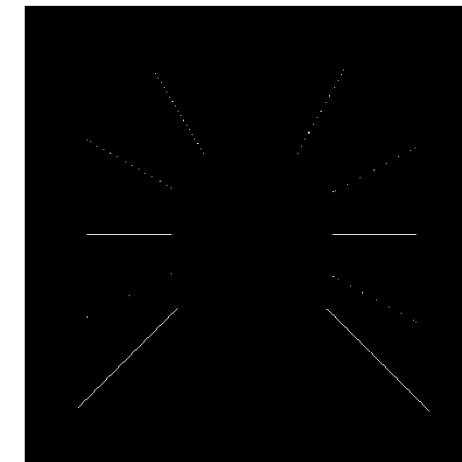
-45°响应图



最大响应图



响应最大的方向



最大响应图二值化

```

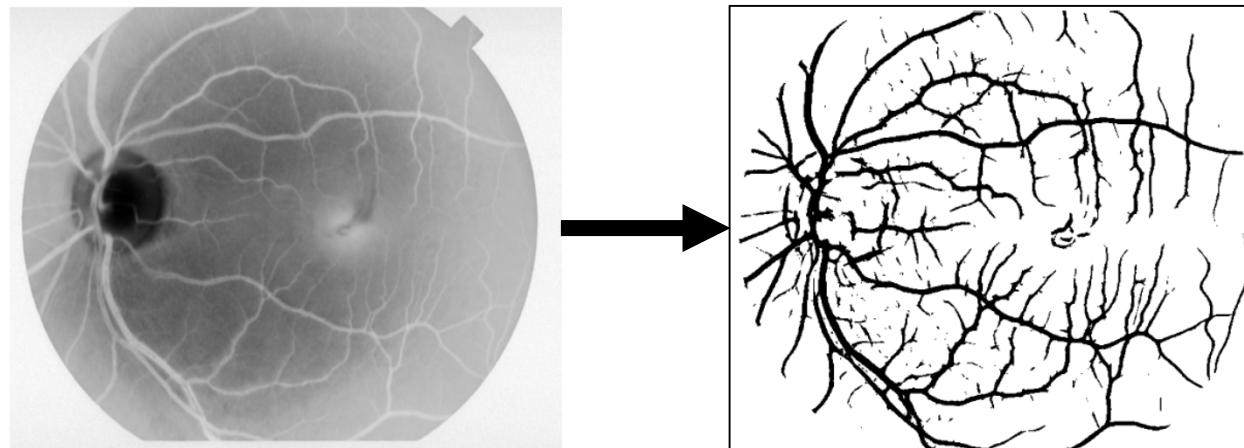
% ex1003_line4Directions
clear all
close all
I = imread('..\data\Fig1005(a)(wirebond_mask).tif');
I = im2double(I);
[M,N] = size(I);

h{1} = [-1 -1 -1; 2 2 2; -1 -1 -1];
h{2} = [2 -1 -1; -1 2 -1; -1 -1 2];
h{3} = [-1 2 -1; -1 2 -1; -1 2 -1];
h{4} = [-1 -1 2; -1 2 -1; 2 -1 -1];
figure, imshow(I);
J = zeros(M,N,4);
for k = 1:4
    J(:,:,k) = imfilter(I, h{k});
    figure, imshow(J(:,:,k),[]);
end
[val, line_direction] = max(abs(J),[],3);
idx = [1:M*N]' + (line_direction(:)-1)*M*N;
M = reshape(J(idx), [M N]);
figure, imshow(M,[]);
t = max(M(:))*0.5;
figure, imshow(M>t,[]);
figure, imshow(line_direction,[]);

```

线检测的应用

医学图像中血管提取、遥感图像中道路提取等



内 容

- 基础
- 点检测
- 线检测
- 边缘检测
- 霍夫变换

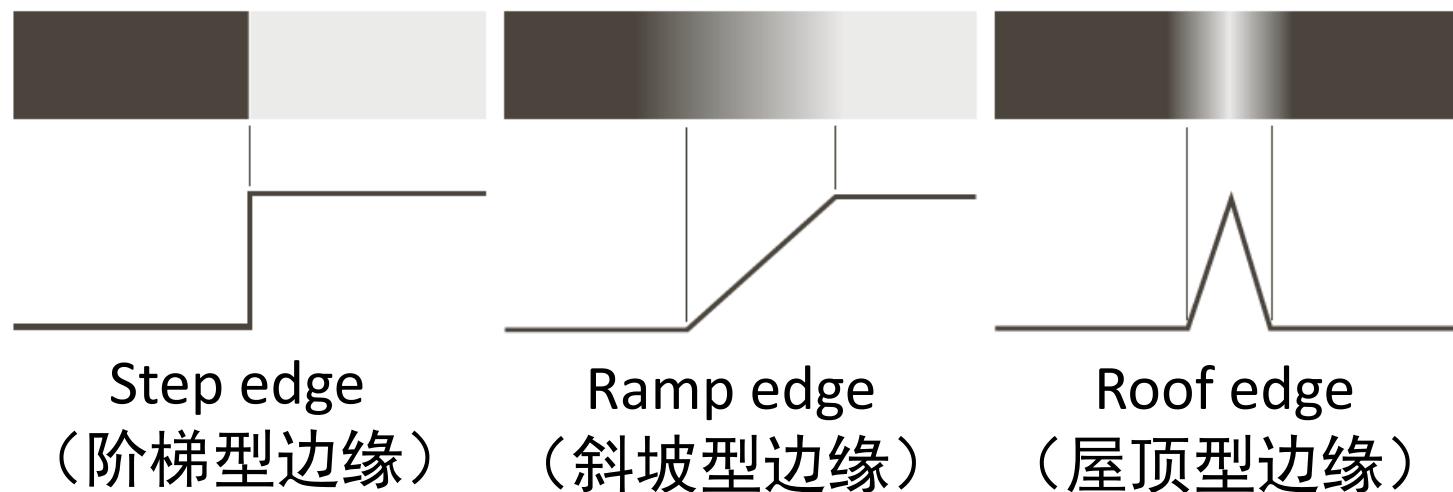
边缘检测

- 前面介绍的点和线的检测，是一直比较特殊的图像分割（将图像中的点、线和背景分割开）
- 下面介绍的边缘检测更为通用

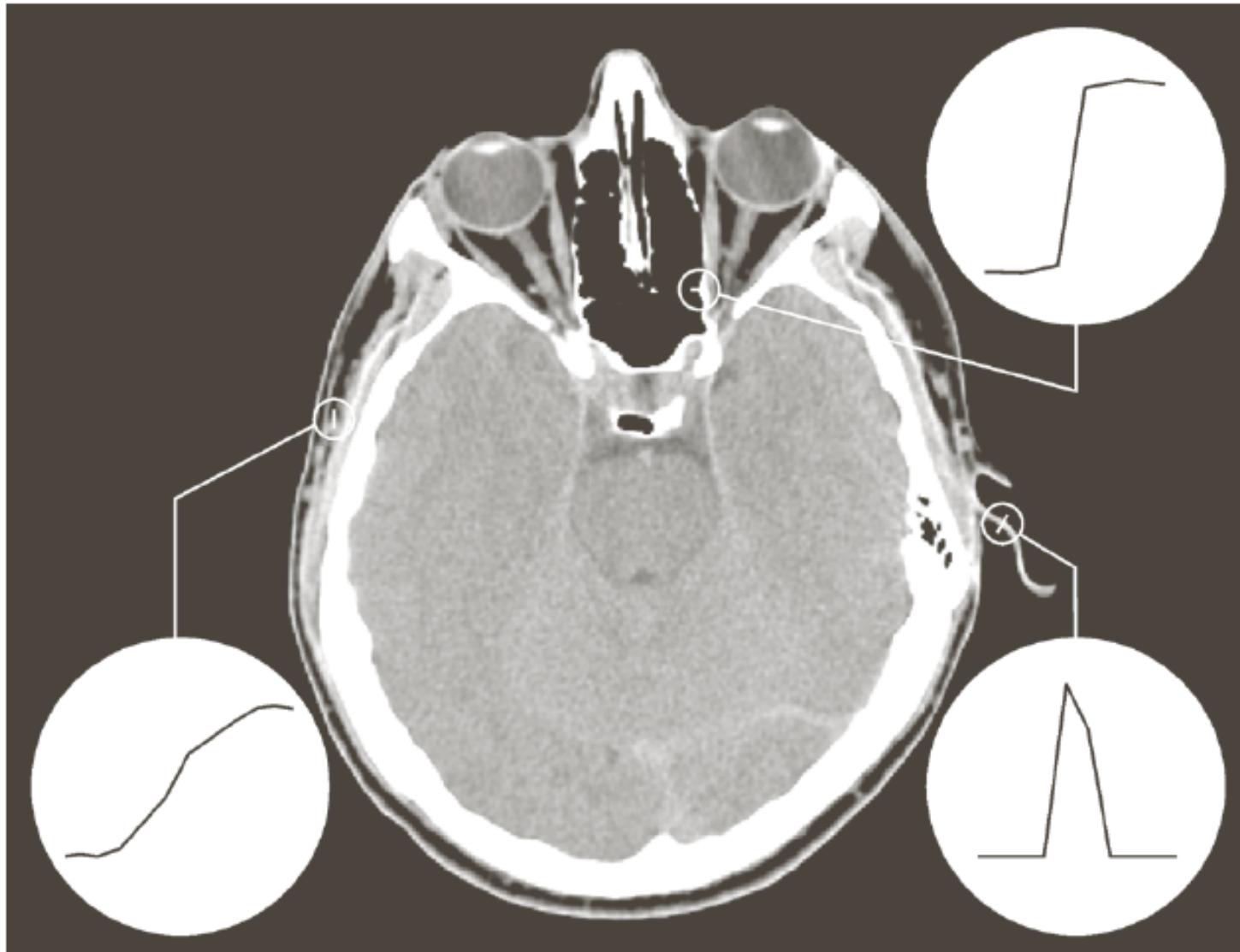


边缘模型

- 用边缘模型来近似实际图像中的边缘；基于这些模型设计边缘检测算法
- 按照灰度剖面图，边缘模型可分为三种
- 阶梯型边缘是理想的边缘，灰度变化很快
- 当实际成像系统的对焦、电子器件不理想时，斜坡型边缘更接近实际的边缘
- 屋顶型边缘类似于线特征

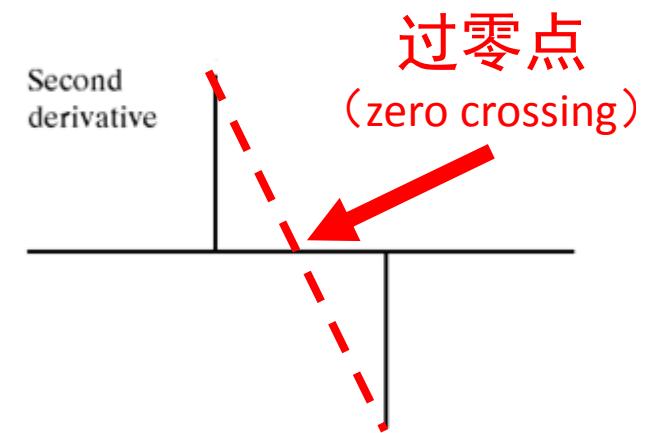
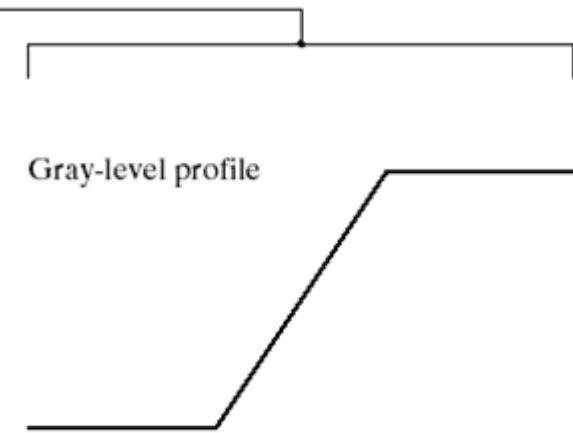
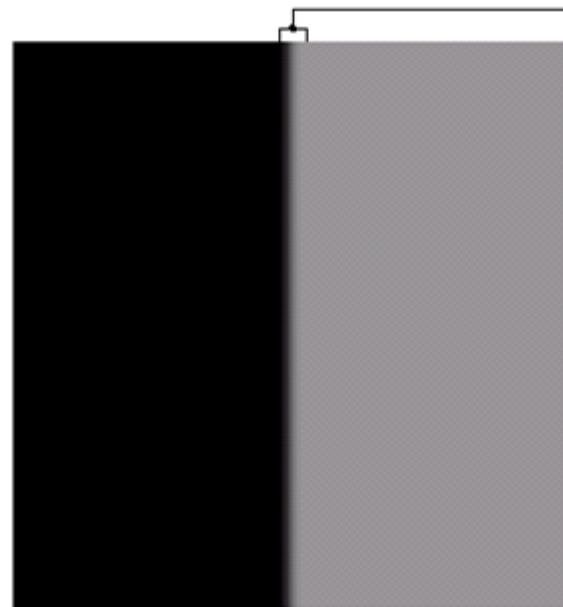


实际图像中的边缘



头部CT的一张切片

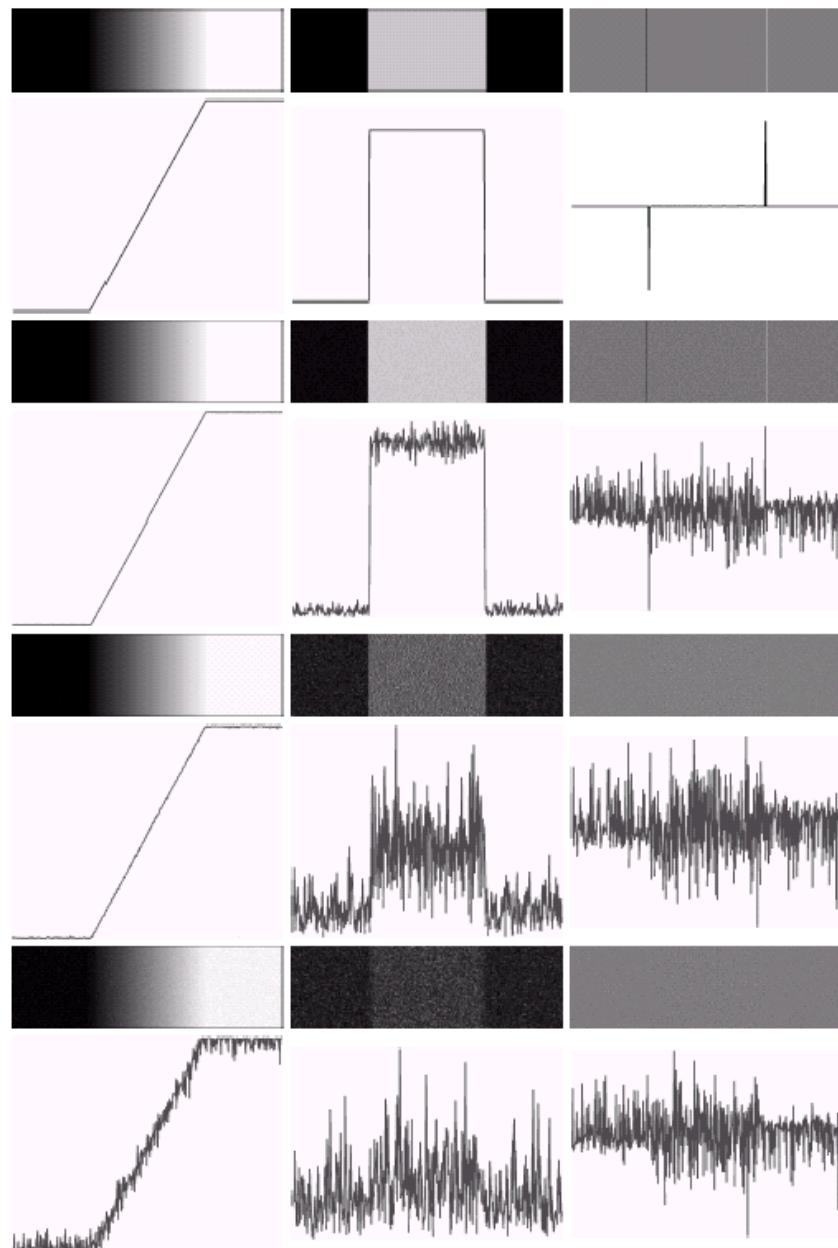
斜坡型边缘的一阶、二阶导



图像某位置的一阶二阶导的特点：

- 一阶导可检测是否有边缘
- 二阶导出现一正一负两个峰值
- 二阶导可判断其位于亮侧还是暗侧
- 两个峰值连线与灰度0轴的交点为过零点，可作为粗边缘的中点

噪声边缘的一阶、二阶导



- 第1列4幅图像的高斯噪声标准方差分别为0, 0.1, 1, 10
- 第2列为一阶导
- 第3列为二阶导
- 一阶导对噪声敏感
- 二阶导对噪声更敏感
- 因此，求导前，对图像平滑

边缘检测算法

- 基本的边缘检测

- 梯度定义
- 梯度算子

- 高级的边缘检测

- Marr-Hildreth方法
- Canny方法

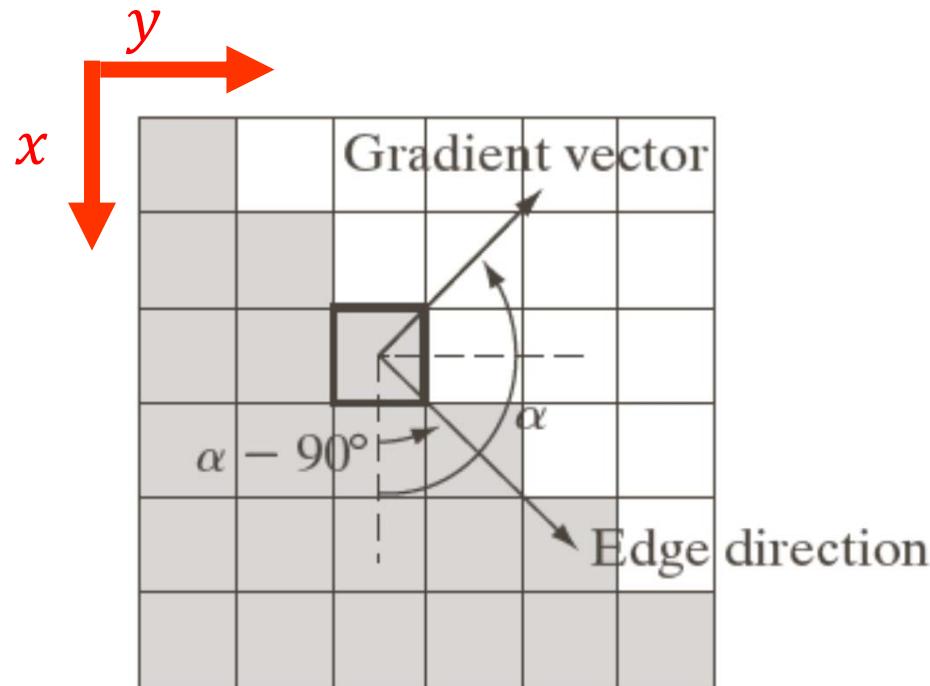
梯度

- 利用图像梯度来测量边缘的强度和方向
- 2D连续函数 $f(x, y)$ 在位置 (x, y) 处的梯度

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

- 梯度是向量，指向 f 变化率最大方向
- 梯度的幅度 $M(x, y) = \sqrt{g_x^2 + g_y^2}$
- 梯度的方向 $\alpha(x, y) = \arctan2(g_x, g_y)$, 范围 $[-\pi, \pi]$

梯度的例子



6 × 6图像，白1，黑0
计算黑框像素处的梯度

使用的梯度算子为

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

x 方向

y 方向

$$\text{梯度 } \nabla f = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} -2 \\ 2 \end{bmatrix}$$

梯度算子

最基本的梯度算子：

$$g_x = f(x + 1, y) - f(x, y)$$

$$g_y = f(x, y + 1) - f(x, y)$$

-1
1

-1	1
----	---

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

x 方向

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

y 方向

Prewitt算子

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

x 方向

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

y 方向

Sobel算子

梯度的例子

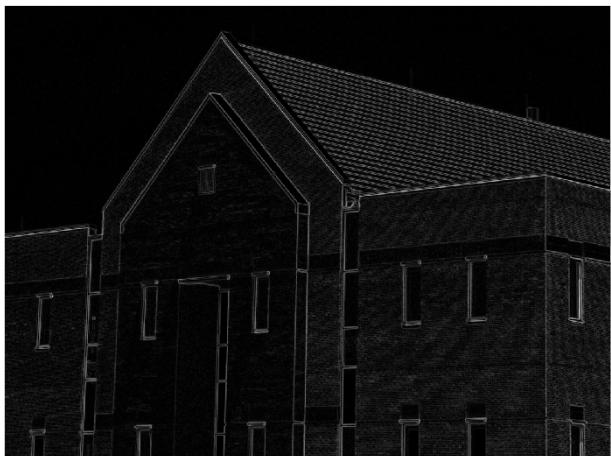
Sobel算子



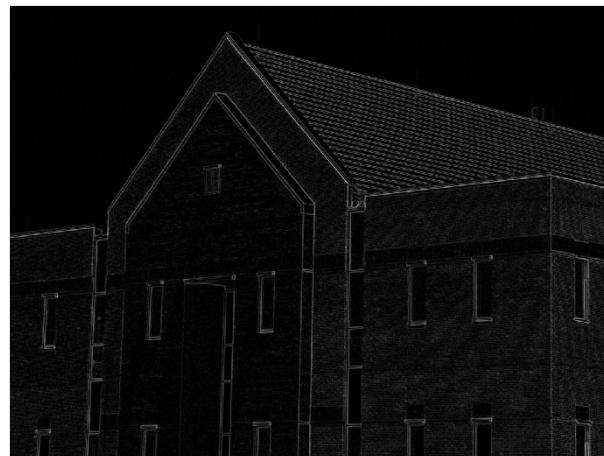
$$|g_x|$$



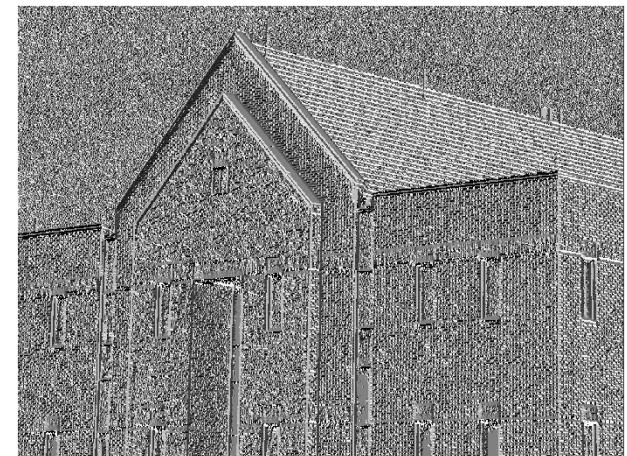
$$|g_y|$$



梯度幅度图



$$|g_x| + |g_y|$$



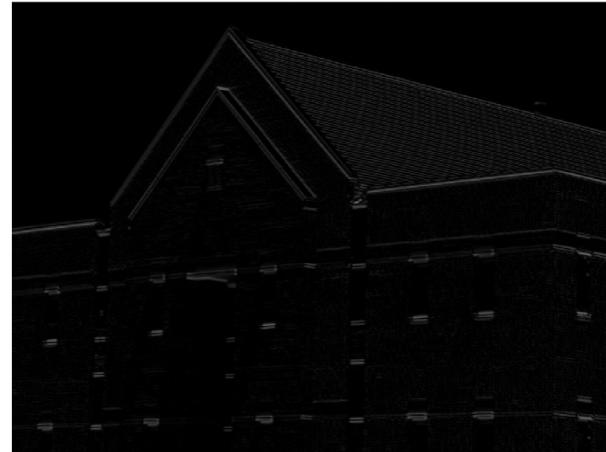
梯度方向图

平滑后求梯度

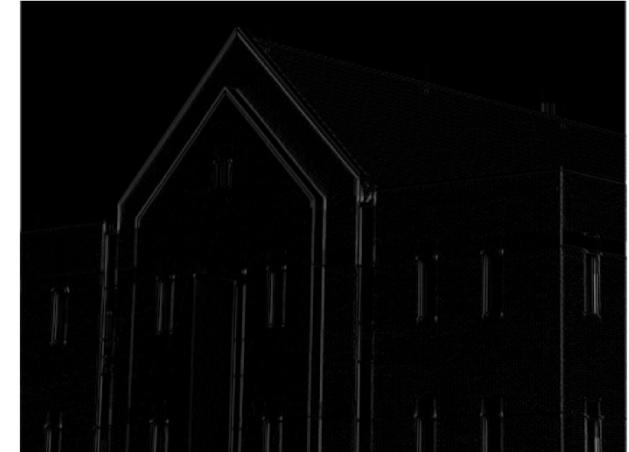
Sobel算子



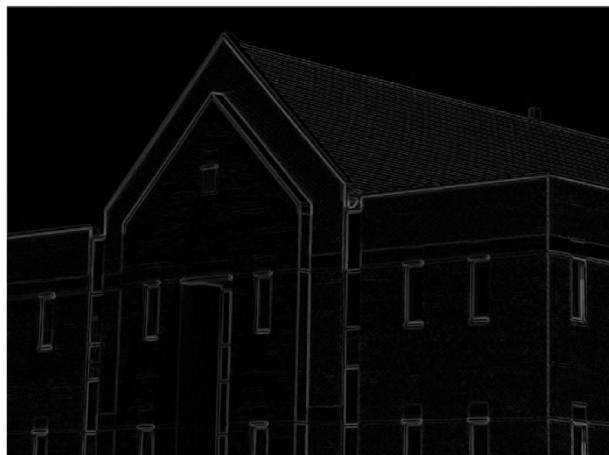
5 × 5 均值滤波后的图像



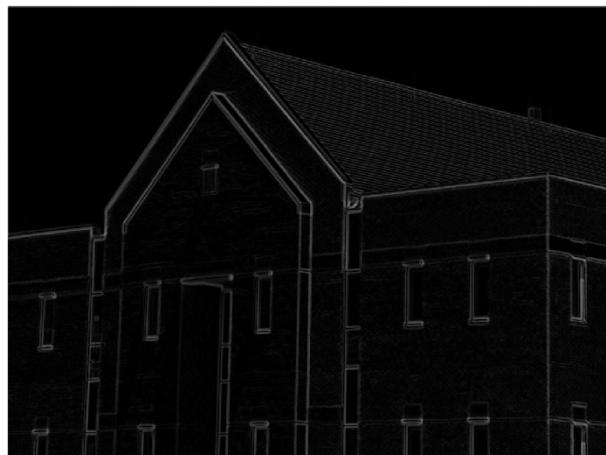
$$|g_x|$$



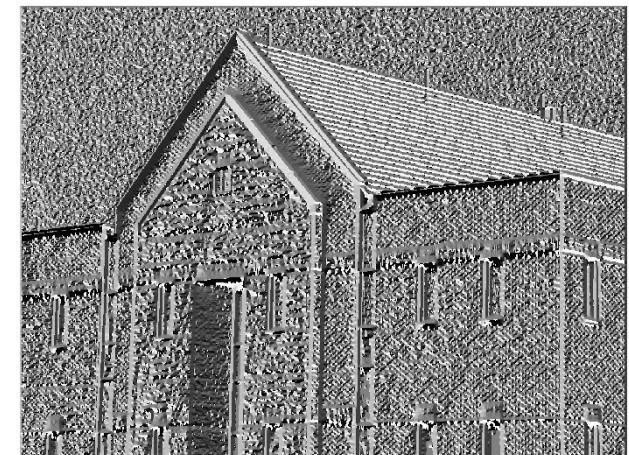
$$|g_y|$$



梯度幅度图



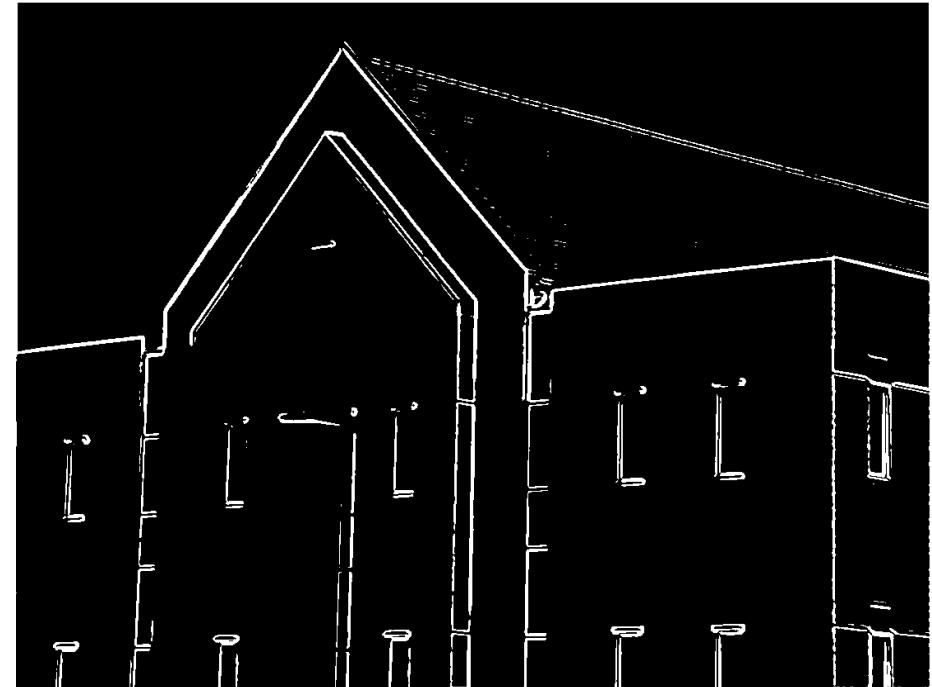
$$|g_x| + |g_y|$$



梯度方向图

梯度阈值化

将幅度图与某阈值相比得到二值图



边缘检测算法

- 基本的边缘检测

- 梯度定义
- 梯度算子

- 高级的边缘检测

- Marr-Hildreth方法
- Canny方法

Marr-Hildreth边缘检测

- 边缘检测滤波器为 Laplacian of Gaussian (LoG)
 $\nabla^2 G$, G 为高斯函数

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

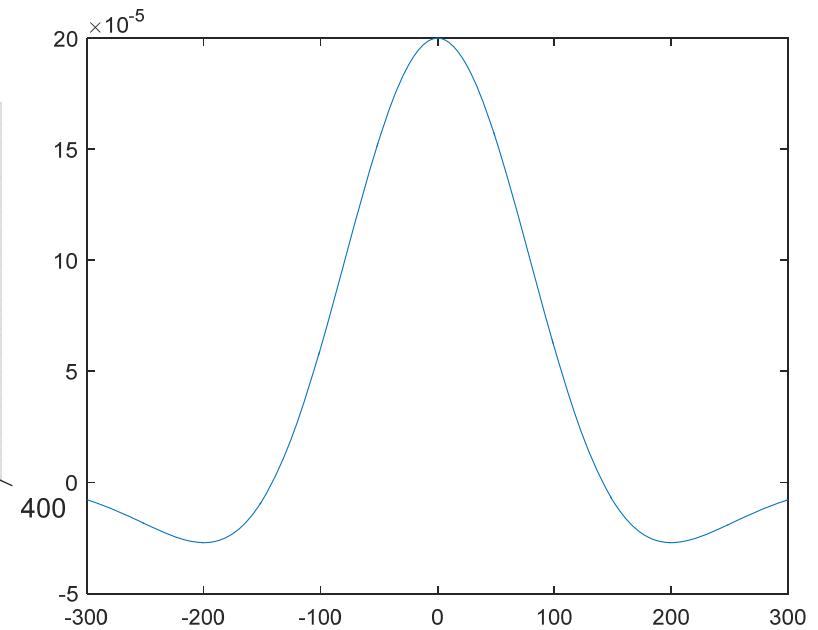
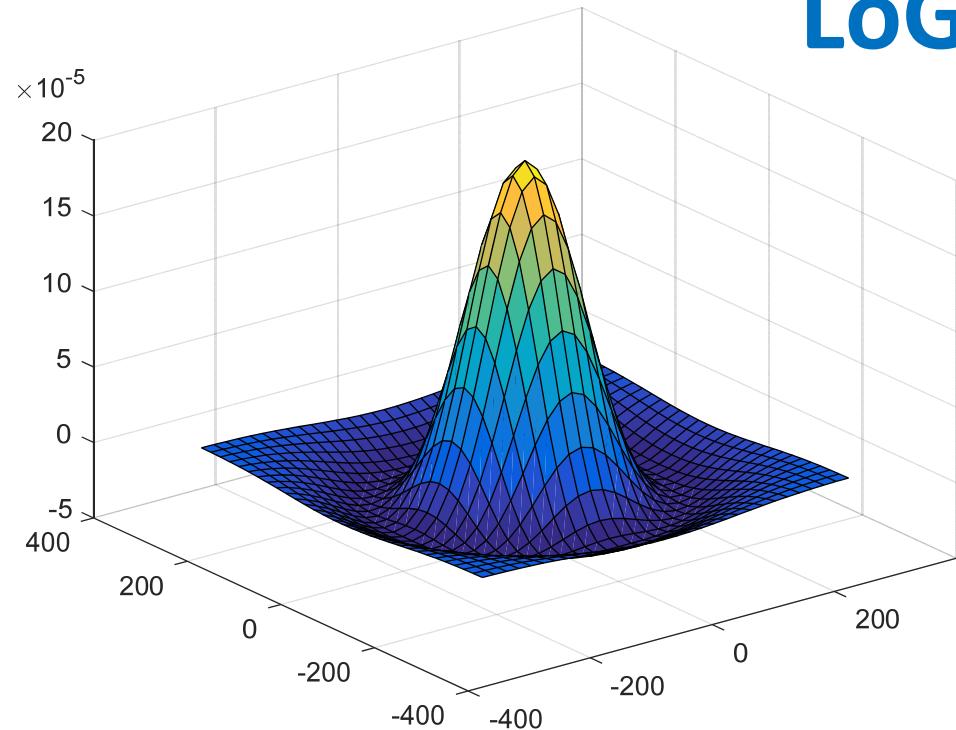
- LoG 滤波

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

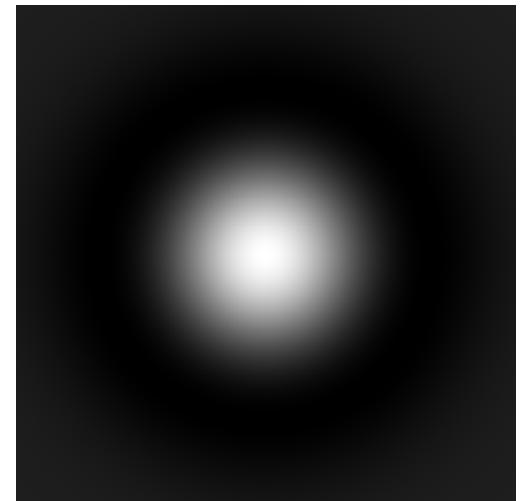
- 由于拉普拉斯和卷积均为线性运算，等价于

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

LoG



$$\sigma = 100$$
$$-\nabla^2 G = - \left[\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$



```
% LoG_show
sigma = 100;
r = 3*sigma;
[X, Y] = meshgrid(-r:r);
LoG = -(1/(sigma^4))*(X.^2+Y.^2-2*(sigma^2)).*exp(-
(X.^2+Y.^2)/(2*(sigma^2)));
b = 20;
figure,
surf(X(1:b:end,1:b:end),Y(1:b:end,1:b:end),LoG(1:b:end,1:b:end));
figure, imshow(LoG, []);
figure, plot(-r:r,LoG(r+1,:)));
```

Marr-Hildreth边缘检测

算法步骤：

1. 对图像做高斯低通滤波（可用2次1D高斯低通滤波）
2. 计算滤波后图像的拉普拉斯
3. 检测过零点

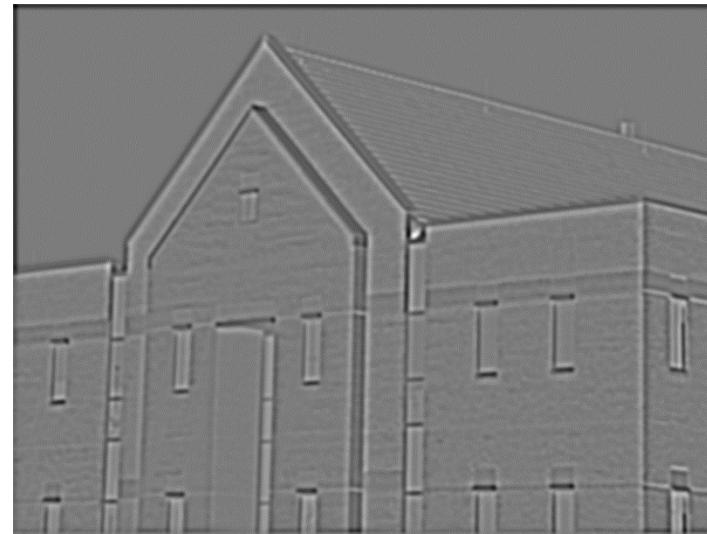
检测过零点：

- 整数坐标 $g(x, y)$ 为0且为过零点的情况不多
- 对于每个像素检查其 3×3 邻域，如果至少有一对（上下、左右、2对对角线）近邻像素的值符号相反，该像素就是过零点
- 为了抑制噪声，可要求像素值差异的绝对值超过某阈值

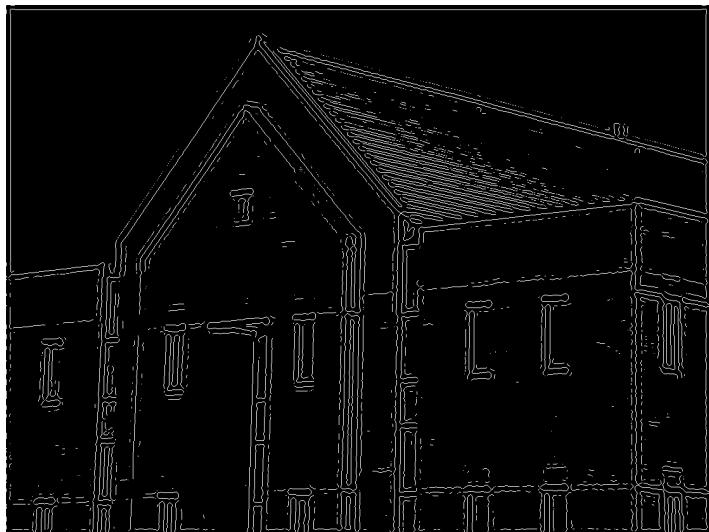
Marr-Hildreth边缘检测的例子



输入图像



LoG滤波结果



阈值为0



阈值不为0

```
% ex1007_log
I = imread('..\data\Fig1022(a)(building_original).tif');
I = im2double(I);
sigma = 4;
hsize = 25;
h = fspecial('log', hsize, sigma);
J = imfilter(I, h);
E = edge(J, 'zerocross');
t = max(J(:))*0.01;
E2 = edge(J, 'zerocross', t);
close all
figure, imshow(I);
figure, imshow(J, [ ]);
figure, imshow(E, [ ]);
figure, imshow(E2, [ ]);
```

Canny边缘检测

边缘检测的目标：

- 错误率低：假阳性和假阴性的比例低
- 边缘定位准确
- 单一响应

Canny, J., *A Computational Approach To Edge Detection*, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986

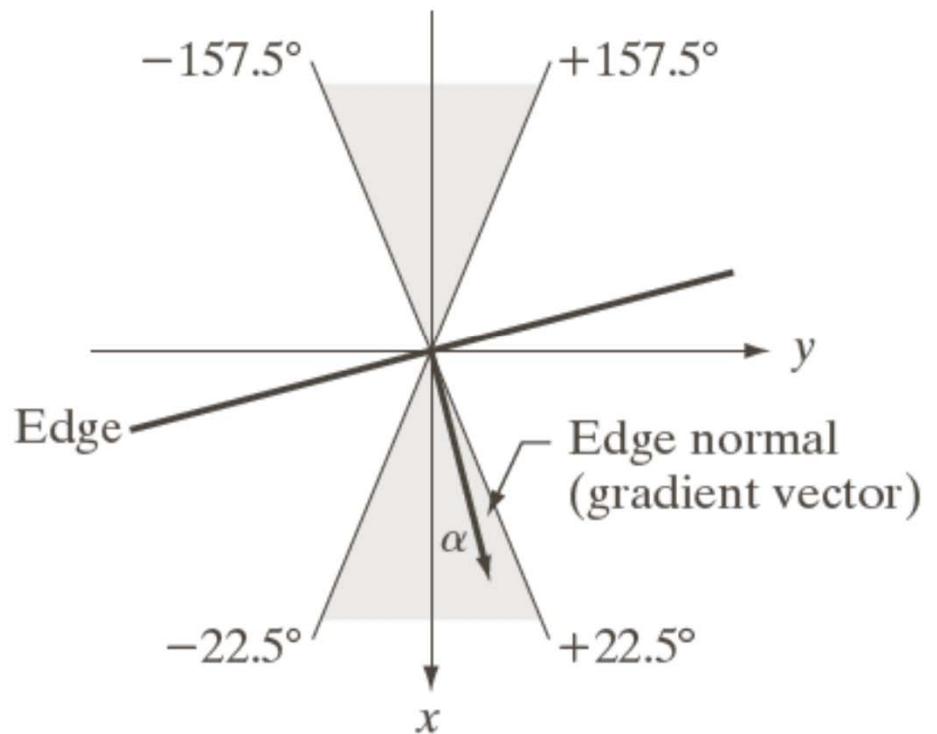
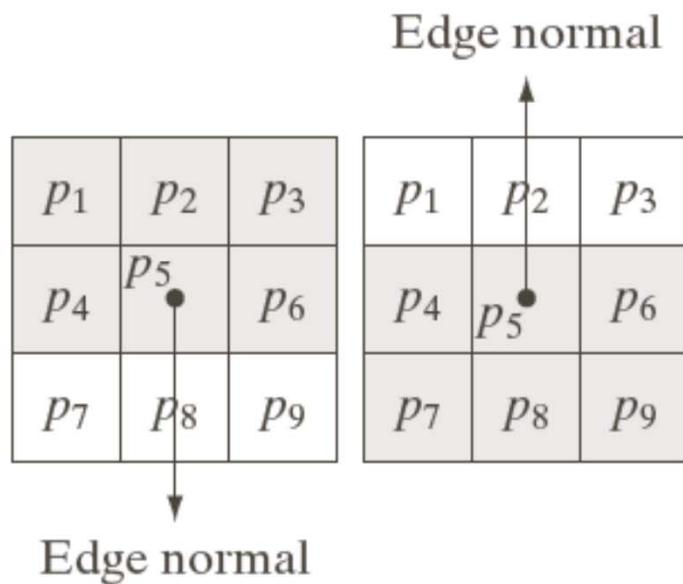
Canny边缘检测算法

步骤：

1. 用高斯低通滤波器对输入图像进行平滑
2. 计算梯度的幅度图和角度图
3. 对梯度幅度图做非极大抑制（nonmaxima suppression）
4. 使用双阈值和连通性分析来检测和连接边缘

非极大值抑制

- 梯度图并不足以精确确定边缘，须保留梯度局部最大的点，而抑制非极大值。
- 判断当前像素在垂直边缘方向是否局部极大（可能需要插值）



双阈值与连通性分析

- 对非极大值抑制图像作用两个阈值 th_1 和 th_2 ，两者关系 $th_1=0.4th_2$ 。
- 我们把梯度值小于 th_1 的像素的灰度值设为0，得到图像1。然后把梯度值小于 th_2 的像素的灰度值设为0，得到图像2。
- 由于图像2的阈值较高，去除大部分噪音，但同时也损失了有用的边缘信息。而图像1的阈值较低，保留了较多的信息，我们可以以图像2为基础，以图像1为补充来连结图像的边缘。

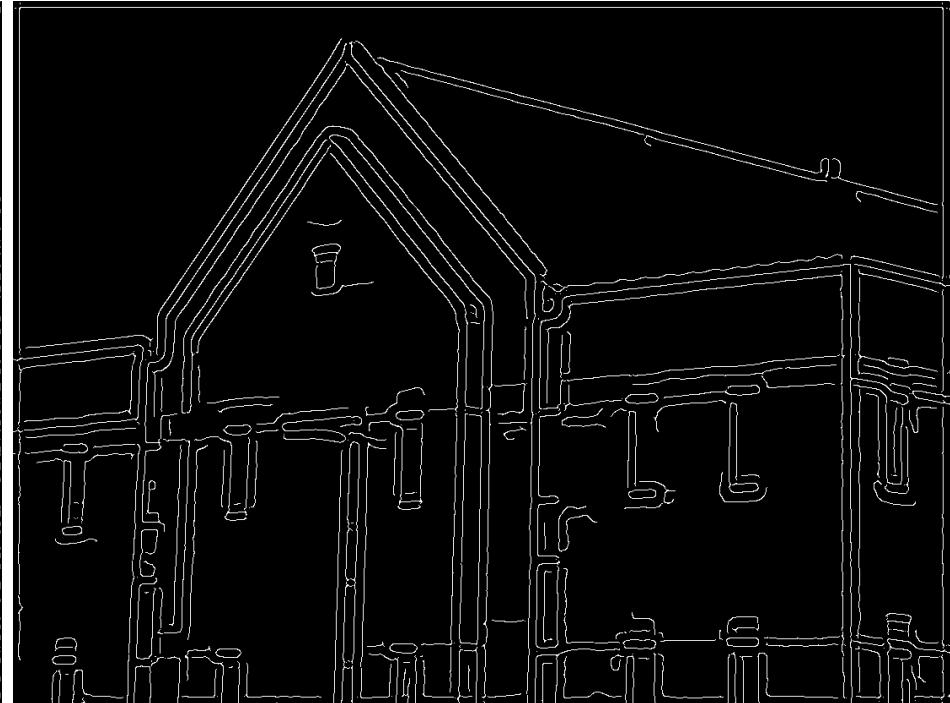
双阈值与连通性分析

- 对图像2进行扫描，当遇到一个非零灰度的像素 $p(x,y)$ 时，跟踪以 $p(x,y)$ 为开始点的轮廓线，直到轮廓线的终点 $q(x,y)$ 。
- 考察图像1中与图像2中 $q(x,y)$ 点位置对应的点 $s(x,y)$ 的8邻近区域。如果在 $s(x,y)$ 点的8邻近区域中有非零像素 $s(x,y)$ 存在，则将其包括到图像2中，作为 $r(x,y)$ 点。从 $r(x,y)$ 开始，重复第一步，直到我们在图像1和图像2中都无法继续为止。
- 当完成对包含 $p(x,y)$ 的轮廓线的连结之后，将这条轮廓线标记为已经访问。回到第一步，寻找下一条轮廓线。重复第一步、第二步、第三步，直到图像2中找不到新轮廓线为止。

Canny边缘检测的例子

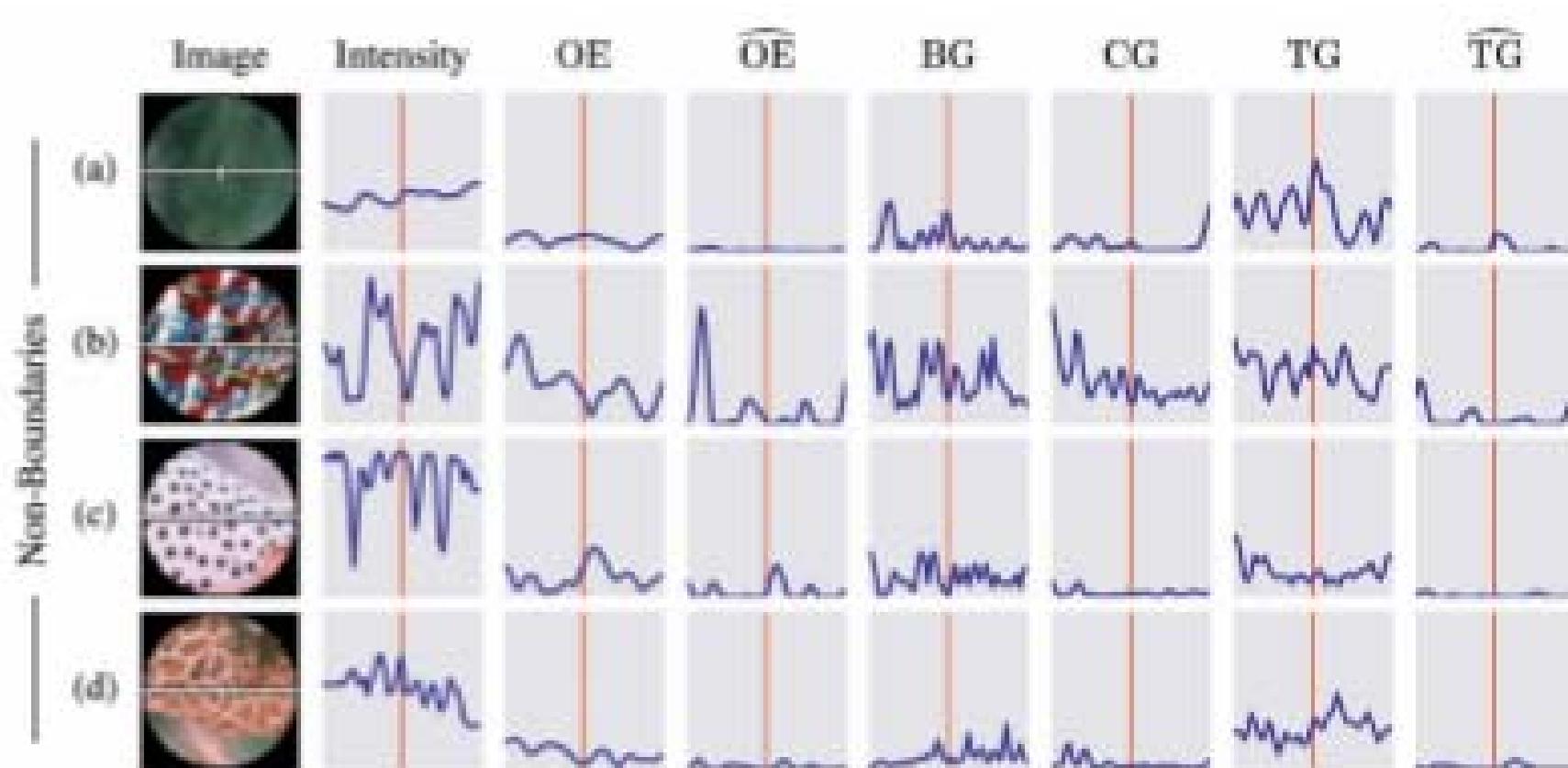


Marr-Hildreth

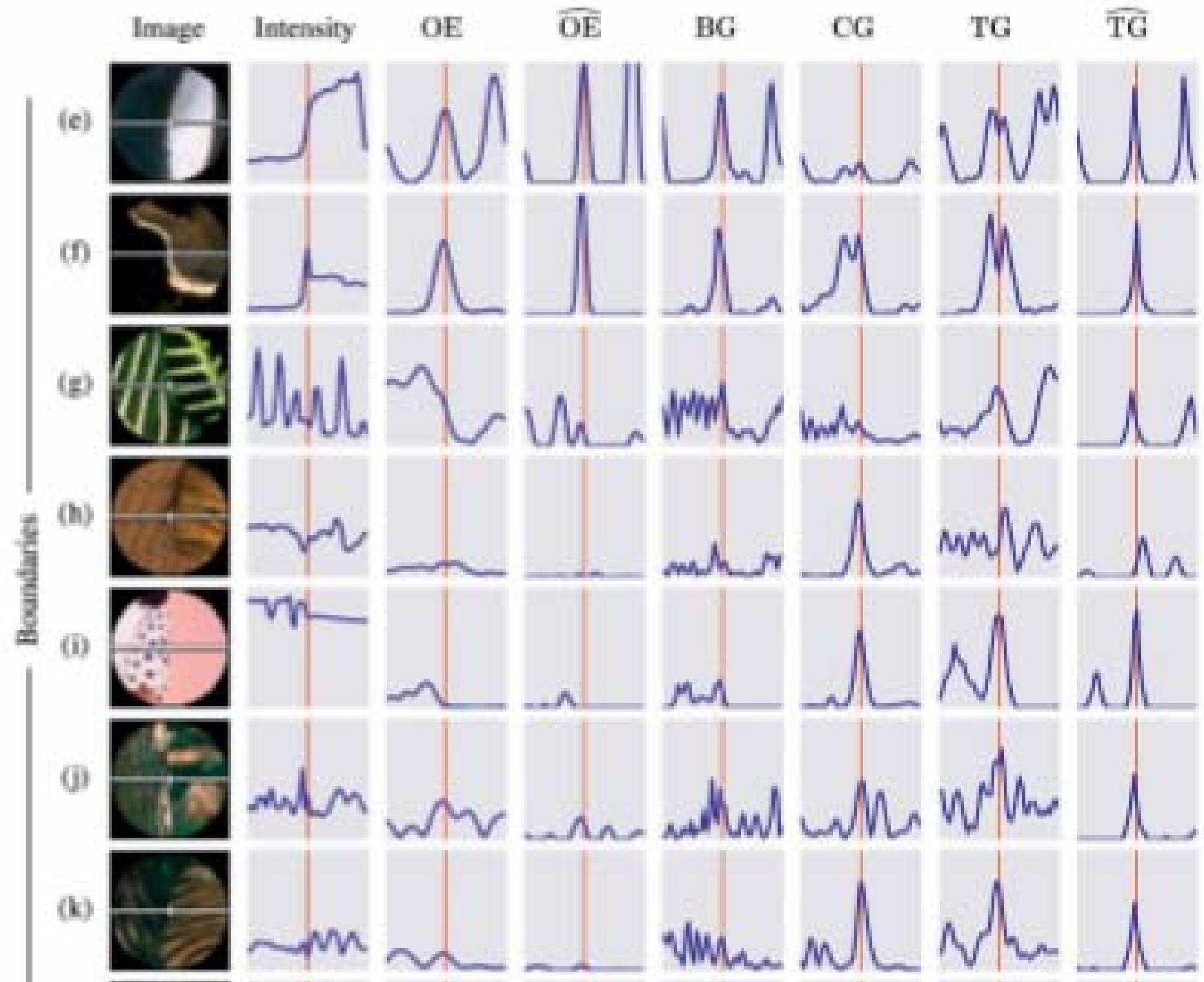


Canny
threshold 0.1,
 $\sigma=4$

更高级的边缘检测



David R. Martin, Charless C. Fowlkes, Jitendra Malik:
Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and
Texture Cues. IEEE Trans. PAMI. 26(5): 530-549 (2004)



内 容

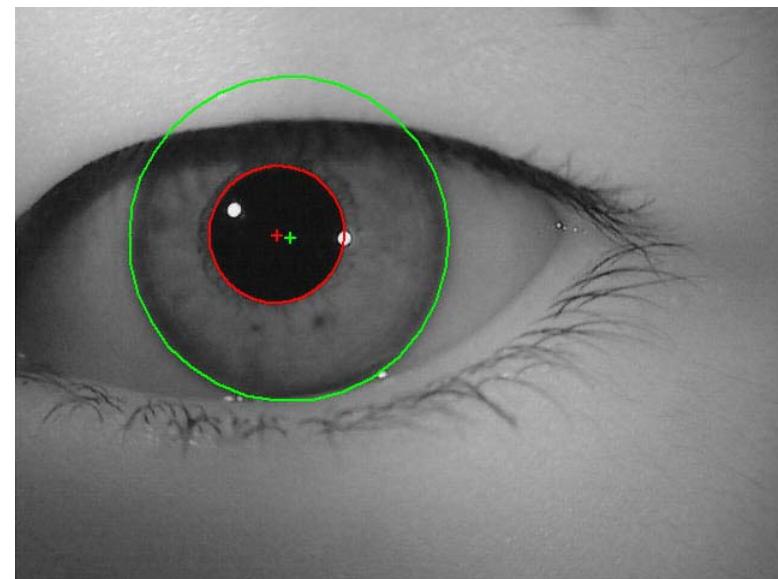
- 基础
- 点检测
- 线检测
- 边缘检测
- 霍夫变换

霍夫变换（Hough Transform）

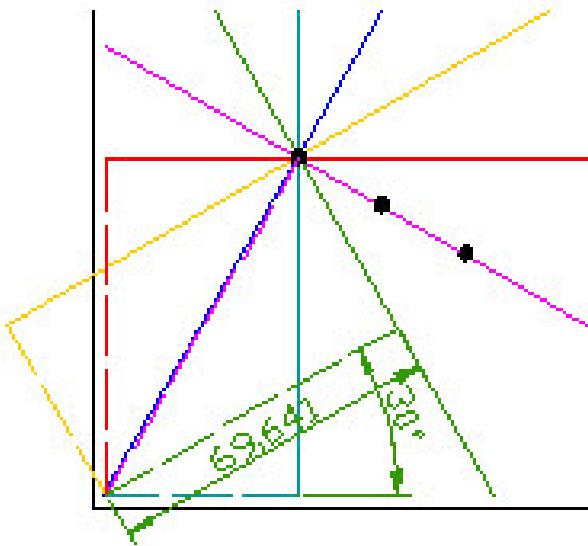
- 边缘检测算法输出的边缘点往往不够连续，或包含许多错误边缘，或不适合后续的图像分析任务
- 如果我们对边缘构成的形状有某种先验知识（例如直线、圆、椭圆等），可以利用霍夫变换估计形状的参数，提高边缘的质量，或者用于图像分析任务



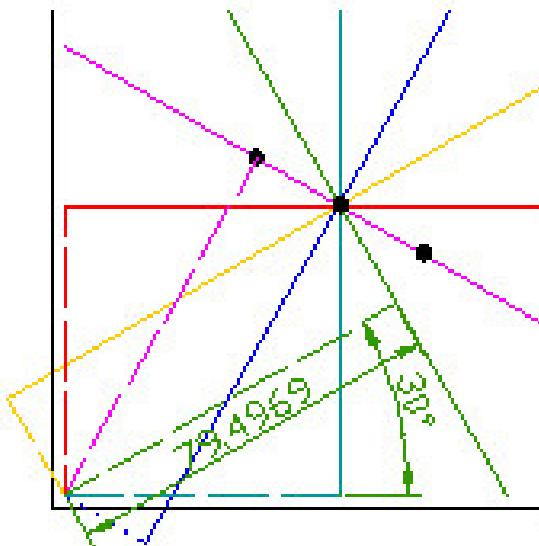
无人驾驶中的车道线检测



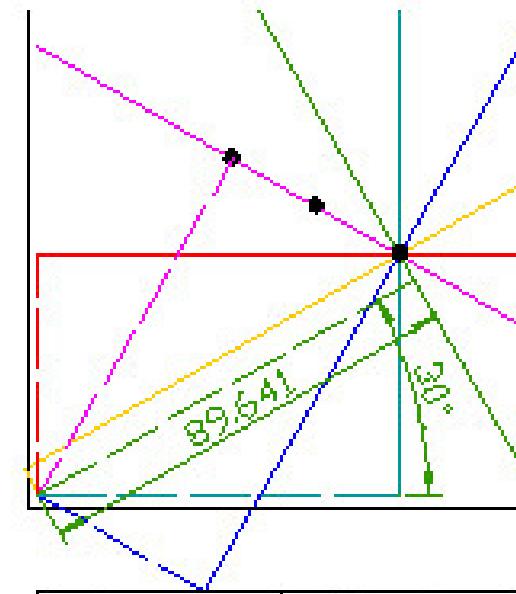
虹膜分割



Angle	Dist.
0	40
30	69.6
60	81.2
90	70
120	40.6
150	0.4



Angle	Dist.
0	57.1
30	79.5
60	80.5
90	60
120	23.4
150	-19.5



Angle	Dist.
0	74.6
30	89.6
60	80.6
90	50
120	6.0
150	-39.6

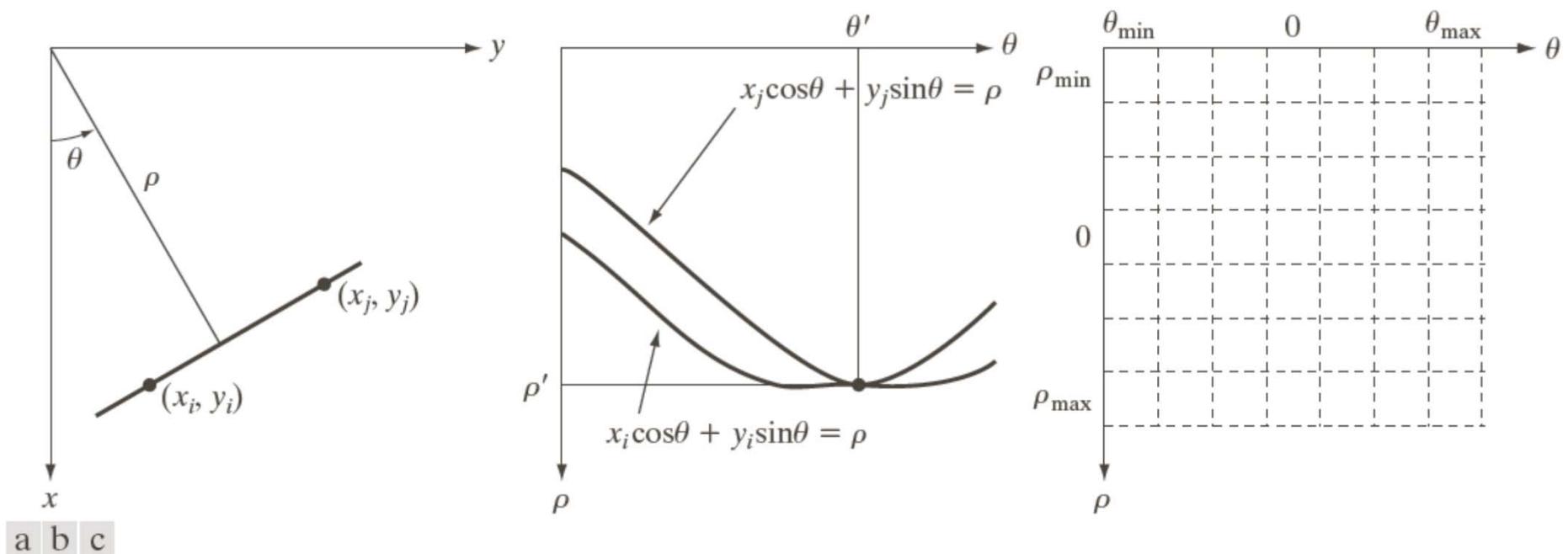
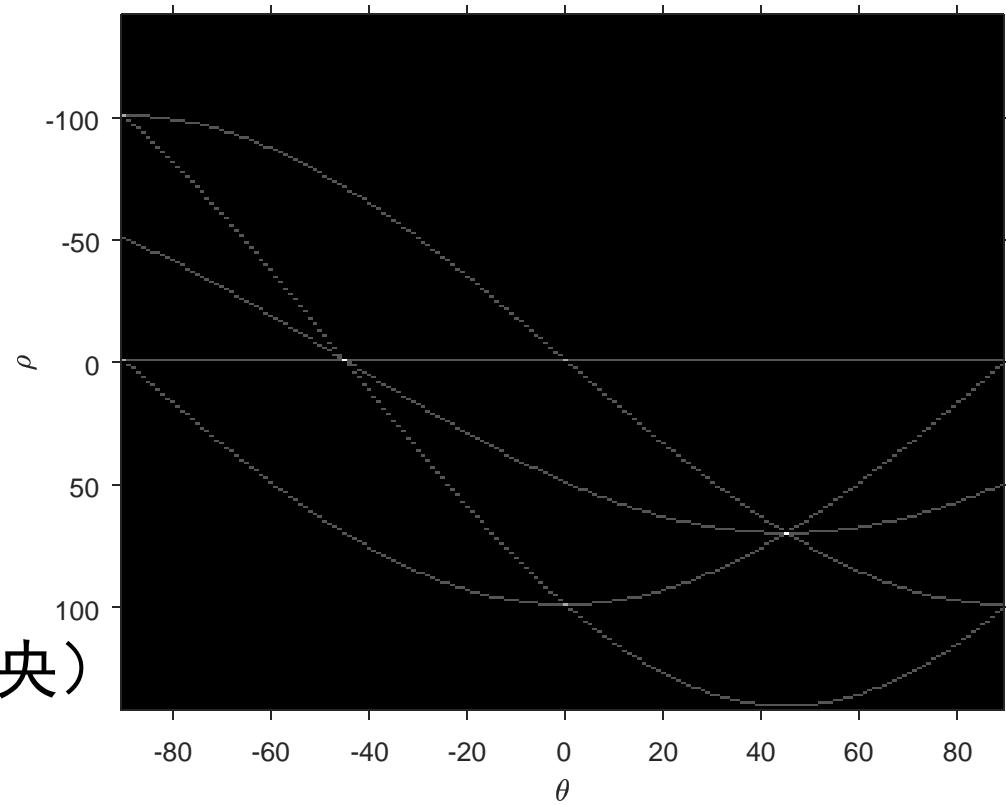


FIGURE 10.32 (a) (ρ, θ) parameterization of line in the xy -plane. (b) Sinusoidal curves in the $\rho\theta$ -plane; the point of intersection (ρ', θ') corresponds to the line passing through points (x_i, y_i) and (x_j, y_j) in the xy -plane. (c) Division of the $\rho\theta$ -plane into accumulator cells.

霍夫变换例子



图像（5个白像素：4个角和中央）

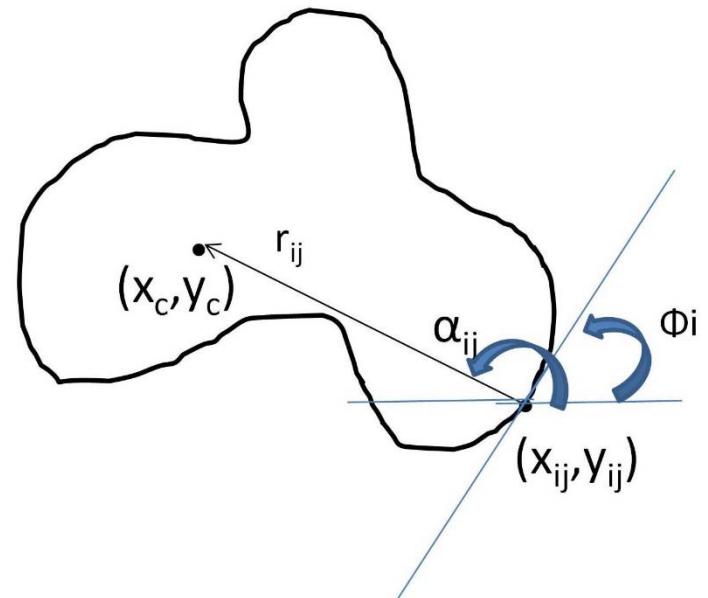
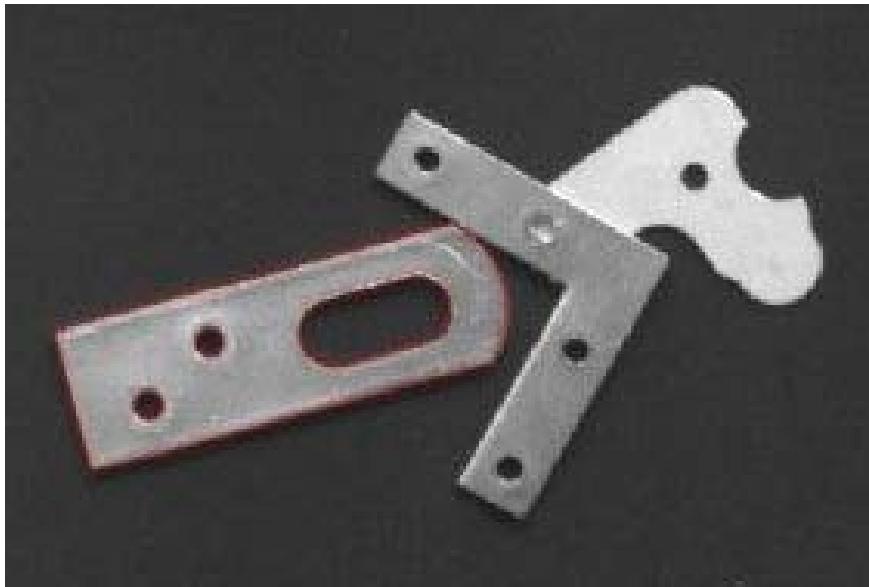


参数空间图像

推广霍夫变换（Generalized Hough Transform）

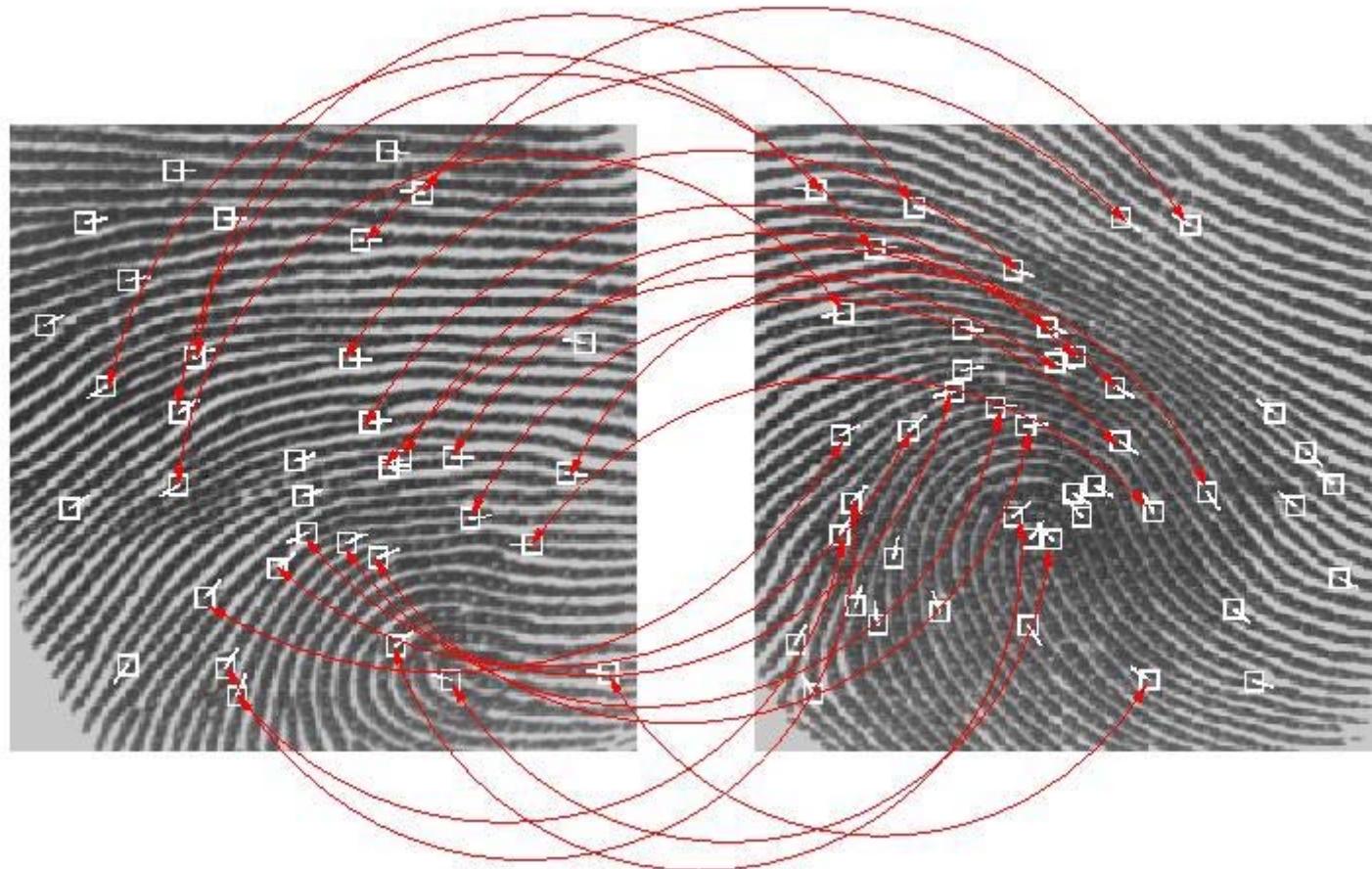
- 基本的霍夫变换用于检测具有显示数学表达式的形状
- 推广的霍夫变换（GHT）用于工件检测、点匹配、物体检测等；核心是模板匹配

工件检测



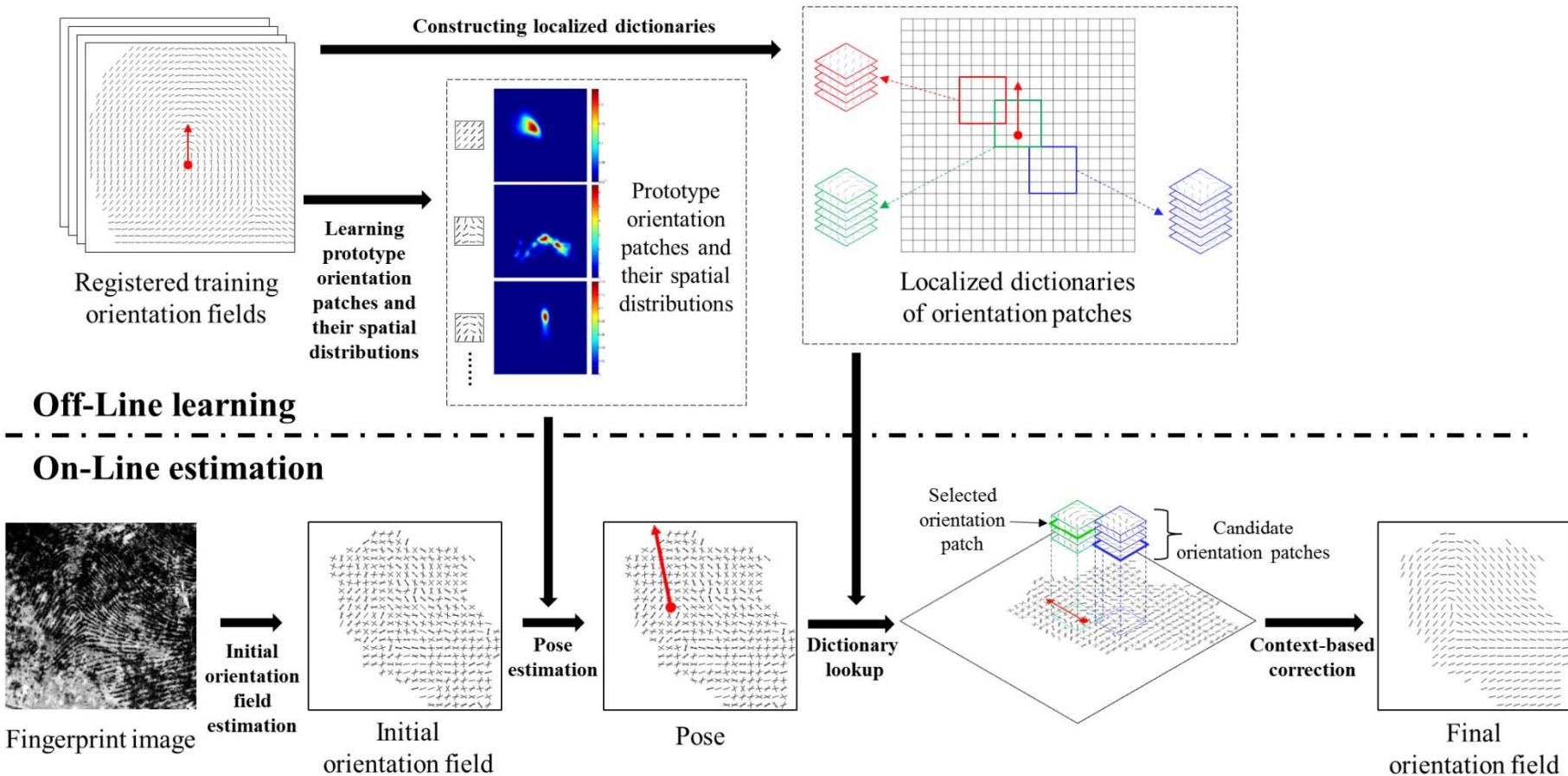
D.H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes",
Pattern Recognition, Vol.13, No.2, p.111-122, 1981

GHT指纹细节点匹配



N. K. Ratha, K. Karu, S. Chen, and A. K. Jain, "A Real Time Matching System for Large Fingerprint Database", IEEE Trans. PAMI, 18(8), 1996, pp. 799-813.

GHT指纹姿态估计



Xiao Yang, Jianjiang Feng, Jie Zhou.

Localized Dictionaries Based Orientation Field Estimation for Latent Fingerprints.

IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI). 36(5): 955-969 (2014)