

边缘检测

Edge Detection

§ 3-3 边缘和线的检测

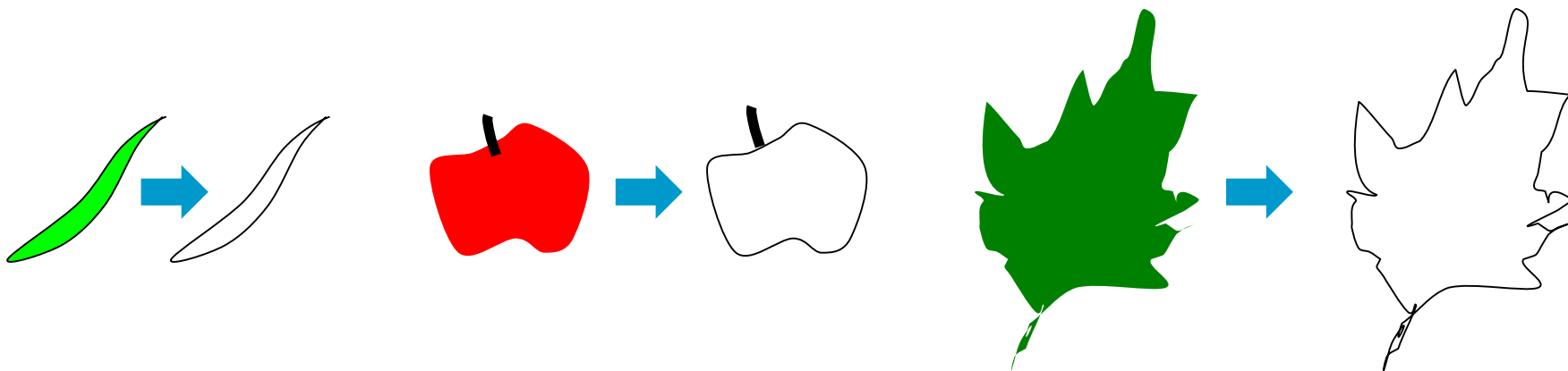
边缘检测是图像处理和计算机视觉中的基本问题，边缘检测的目的是标识数字图像中亮度变化明显的像素点。图像亮度的显著变化通常反映了场景内对象的变化：

- (1) 深度上的不连续(如立体图像中对象与背景的深度不同)
- (2) 表面方向不连续(如屋脊)
- (3) 物质属性变化(如人上装与下装的颜色不同)
- (4) 场景照明变化

边缘检测是图像处理和计算机视觉中，尤其是特征提取中的一个重要内容。

§ 3-3 边缘和线的检测

- **景物轮廓即边缘**，它和区域一样能用来表示有意义的景物的范围，而且两者是共存的；
- **区域**利用**特性相似**得到，而**边缘**利用**特性差异**得到；
- 边缘和线有着密切的关系，边缘可以看成是线的一种，即封闭线；而线可以看作是边缘的组成部分。
- 检测边缘，完成图像分割的处理可以在灰度图像、二值图像、彩色图像等多种图像中进行。



基本术语

边缘点(Edge point): 在灰度显著变化的位置上的点。

边缘段(Edge segment): 对应于边缘点坐标及其方位。

边缘检测器(Edge detector): 从图像中抽取边缘集合的算法。

轮廓(Boundary): 边缘列表或一条表示边缘列表的拟合曲线。

边缘连接(Edge linking): 从无序边缘表形成有序边缘表的过程。

边缘跟踪(Edge tracking): 一个用来确定轮廓的图像搜索过程。

边缘检测算法基本步骤：

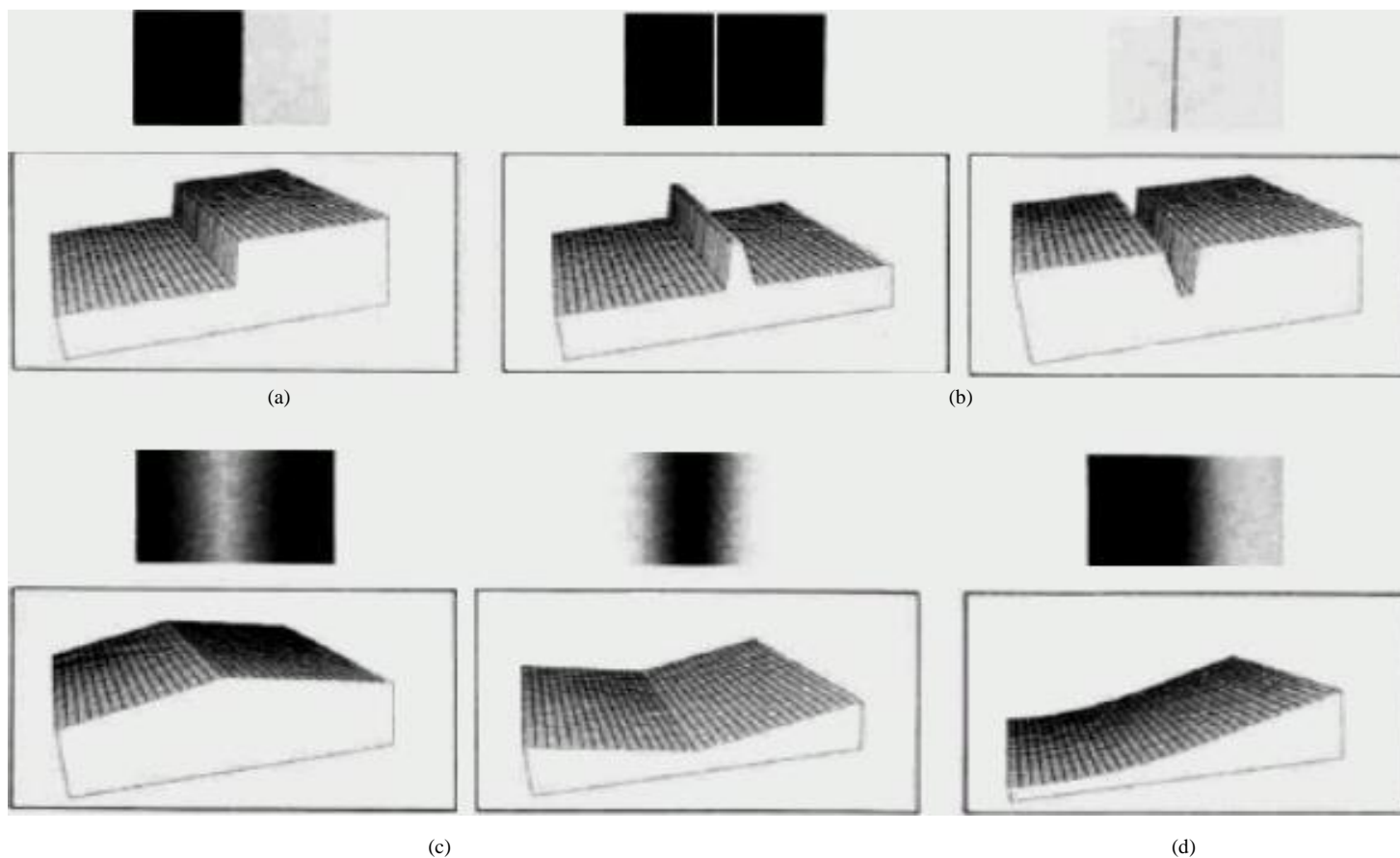
(1) 滤波：边缘检测算法主要基于图像强度的一阶和二阶导数，但导数的计算对噪声很敏感，因此必须使用滤波器来改善与噪声有关的边缘检测器的性能。需要指出，大多数滤波器在降低噪声的同时也导致了边缘强度的损失，因此，增强边缘和降低噪声之间需要折衷。

(2) 增强：增强边缘的基础是确定图像各点邻域强度的变化值。增强算法可以将邻域（或局部）强度值有显著变化的点突显出来。边缘增强一般是通过计算梯度幅值来完成的。

(3) 检测：在图像中有许多点的梯度幅值比较大，而这些点在特定的应用领域中并不一定都是边缘，所以应该用某种方法来确定哪些点是边缘点。最简单的边缘检测判据是梯度幅值阈值判据。

(4) 定位：如果某一应用场合要求确定边缘位置，则边缘的位置可在子像素分辨率上来估计，边缘的方位也可以被估计出来。

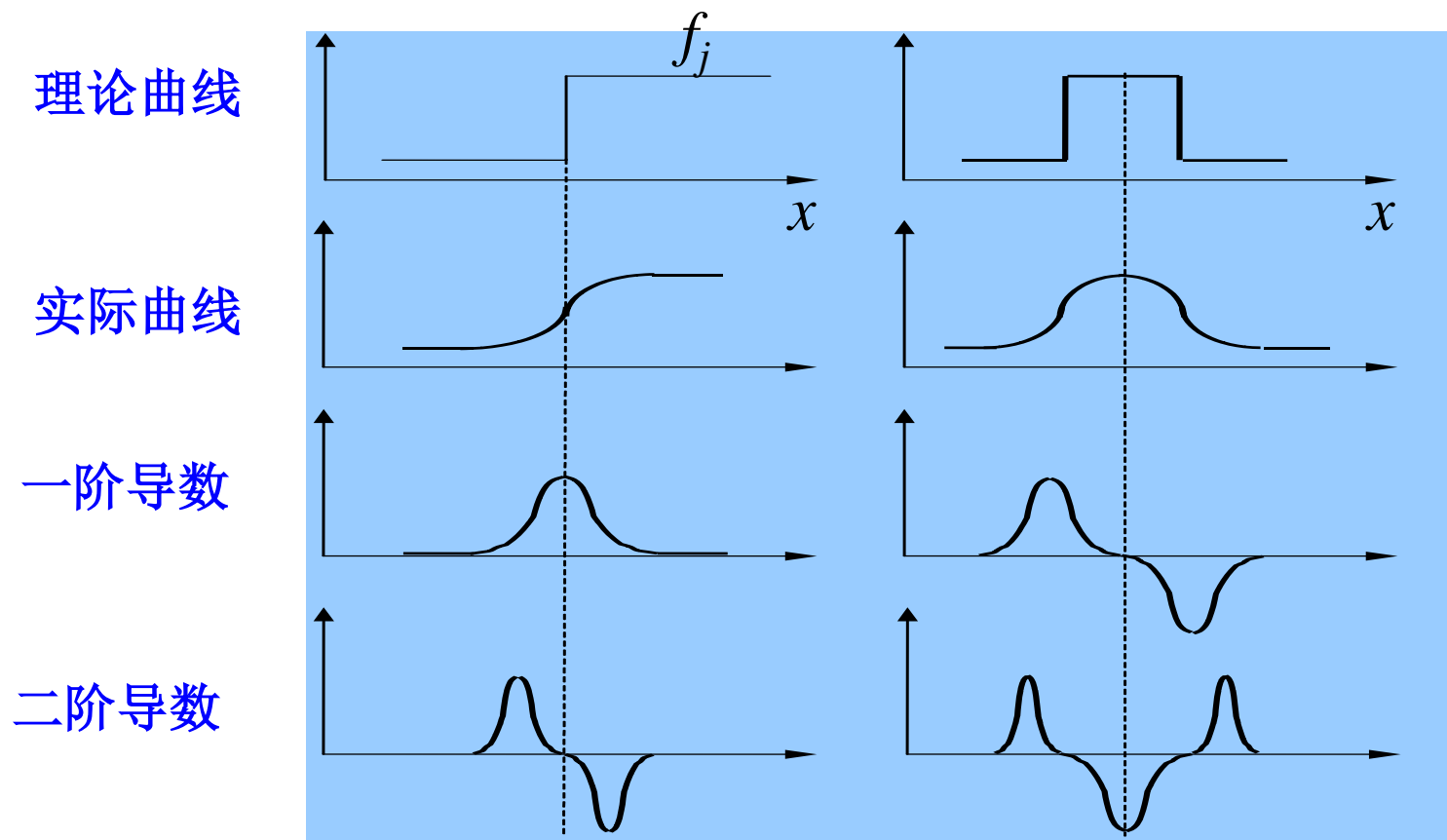
一、边缘特点



图像中不同类型的边界

(a) 边界; (b) 线; (c) 折线变化; (d) 缓慢的平滑变化

两种常见的边缘一阶导数和二阶导数示意图



(a)阶跃函数(阶梯状)

(b)线条函数(脉冲状)

§ 3-3 边缘和线的检测

实际的图像往往是复杂的，灰度的变化也不一定仅限于上面这些标准形式，并且常常伴随噪声，给边缘检测带来困难。

因此，边缘检测方法的研究也一直深受重视。

§ 3-3 边缘和线的检测

二、常用的边缘检测算子

边缘检测的基本做法是，用一定的算子如Roberts算子、Sobel算子、Laplacian算子等得到的差分值，然后作阈值运算：

$$g(i, j) = \begin{cases} g_1 & E(i, j) \geq T_E \\ g_2 & E(i, j) < T_E \end{cases}$$



差分值

§ 3-3 边缘和线的检测

1、梯度差分算子

用垂直方向和水平方向的一阶差分值计算最大差分方向的一阶差分算子 $E_1(i, j)$

$$E_1(i, j) = \sqrt{[\Delta_1 f(i, j)]^2 + [\Delta_2 f(i, j)]^2}$$

$$\Delta_1 f(i, j) = f(i, j) - f(i-1, j)$$

$$\Delta_2 f(i, j) = f(i, j) - f(i, j-1)$$

梯度差分算子是基础的方法，方向性强、计算不方便，已经很少应用了，但其思想还是很多其他算法的基础。

§ 3-3 边缘和线的检测

2、Roberts算子

Robert算子是一种梯度算子，它用交叉的差分表示梯度，是一种利用局部差分算子寻找边缘的算子，对具有陡峭的低噪声的图像效果最好

$$R(i, j) = |f(i+1, j+1) - f(i, j)| + |f(i+1, j) - f(i, j+1)|$$

$$R(i, j) = \sqrt{[f(i+1, j+1) - f(i, j)]^2 + [f(i+1, j) - f(i, j+1)]^2}$$

$$R(i, j) = \max(|f(i+1, j+1) - f(i, j)|, |f(i+1, j) - f(i, j+1)|)$$

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

演示

§ 3-3 边缘和线的检测

3、Sobel算子

Sobel算子是滤波算子的形式来提取边缘。 X ， Y 方向各用一个模板，两个模板组合起来构成1个梯度算子。 X 方向模板对垂直边缘影响最大， Y 方向模板对水平边缘影响最大。

梯度的大小

$$G = \sqrt{G_x^2 + G_y^2}$$

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * A$$

梯度方向

$$\Theta = \arctan\left(\frac{G_y}{G_x}\right)$$

$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A$$

演示

§ 3-3 边缘和线的检测

4、Laplace算子

Laplace算子是一种各向同性算子，在只关心边缘的位置而不考虑其周围的象素灰度差值时比较合适。**Laplace**算子对孤立象素的响应要比对边缘或线的响应要更强烈，因此只适用于无噪声图象。存在噪声情况下，使用**Laplace**算子检测边缘之前需要先进行低通滤波。

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

$$g(i, j) = 4f(i, j) - f(i+1, j) - f(i-1, j) - f(i, j+1) - f(i, j-1)$$

$$H_1 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

演示

5、高斯-拉普拉斯(LOG)算子

噪声点对边缘检测有较大的影响，效果更好的边缘检测器是高斯-拉普拉斯(LOG)算子。它把高斯平滑滤波器和拉普拉斯锐化滤波器结合起来，先平滑掉噪声，再进行边缘检测，所以效果更好。

常用的LOG算子是 5×5 的模板：

$$\begin{bmatrix} -2 & -4 & -4 & -4 & -2 \\ -4 & 0 & 8 & 0 & -4 \\ -4 & 8 & 24 & 8 & -4 \\ -4 & 0 & 8 & 0 & -4 \\ -2 & -4 & -4 & -4 & -2 \end{bmatrix}$$

6、Canny 边缘检测器

检测阶跃边缘的基本思想是在图像中找出具有局部最大梯度幅值的像素点。检测阶跃边缘的大部分工作集中在寻找能够用于实际图像的梯度数字逼近。图像梯度逼近必须满足两个要求：

- (1) 逼近必须能够抑制噪声效应
- (2) 必须尽量精确地确定边缘的位置。

抑制噪声和边缘精确定位是无法同时得到满足的，也就是说，边缘检测算法通过图像平滑算子去除了噪声，但却增加了边缘定位的不确定性；反过来，若提高边缘检测算子对边缘的敏感性，同时也提高了对噪声的敏感性。

有一种线性算子可以在抗噪声干扰和精确定位之间提供**最佳折衷方案**，它就是高斯函数的一阶导数。

*图像梯度，其实就是对二维离散函数的求导（差分）

Canny边缘检测算子是John F. Canny于 1986 年开发出来的一个多级边缘检测算法。更为重要的是Canny创立了边缘检测计算理论(Computational theory of edge detection)解释这项技术如何工作。

Canny 的目标是找到一个最优的边缘检测算法：

- ① 低误判率，即尽可能少地把边缘点误认为是非边缘点；
- ② 高定位精度，即准确地把边缘点定位在灰度变化最大的像素上；
- ③ 抑制虚假边缘。

为了满足这些要求，Canny 使用了变分法，这是一种寻找满足特定功能的函数的方法。最优检测使用四个指数函数项的和表示，但是它非常近似于高斯函数的一阶导数。

算法4.1 Canny 边缘检测

- (1) 用高斯滤波器平滑图像；
- (2) 用一阶偏导有限差分计算梯度幅值和方向；
- (3) 对梯度幅值应用非极大值抑制；
- (4) 用双阈值算法检测和连接边缘。

$$G[i, j] = e^{-\frac{(i^2 + j^2)}{2\sigma^2}}$$

1) 求图像与高斯平滑滤波器卷积:

$$S[i, j] = G[i, j; \sigma] * I[i, j]$$

2) 使用 2×2 一阶有限差分计算偏导数的两个阵列P与Q:

$$P[i, j] \approx (S[i, j+1] - S[i, j] + S[i+1, j+1] - S[i+1, j]) / 2$$

$$Q[i, j] \approx (S[i, j] - S[i+1, j] + S[i, j+1] - S[i+1, j+1]) / 2$$

3) 幅值和方位角:

$$M[i, j] = \sqrt{P[i, j]^2 + Q[i, j]^2}$$

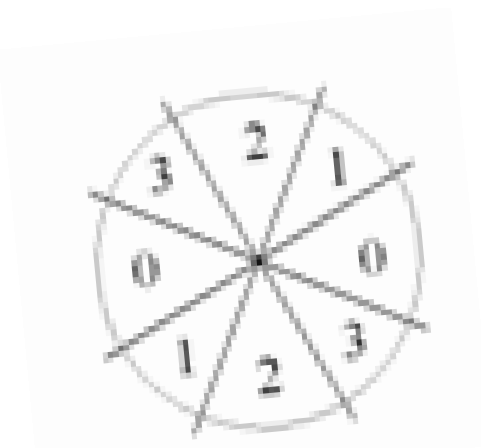
图像梯度

$$\theta[i, j] = \arctan (Q[i, j] / P[i, j])$$

4) 非极大值抑制(Non-Maxima Suppression, NMS)

幅值图像阵列 $M[i,j]$ 的值越大，其对应的图像梯度值也越大，但这还不足以确定边缘，因为这里仅仅把图像快速变化的问题转化成求幅值阵列 $M[i,j]$ 的局部最大值问题。为确定边缘，必须细化幅值图像中的屋脊带(Ridge)，即只保留幅值局部变化最大的点。这一过程叫非极大值抑制(Non-Maxima Suppression, NMS)，它会生成细化的边缘。

非极大值抑制NMS：通过抑制梯度线上所有非屋脊峰值的幅值来细化 $M[i,j]$ 中的梯度幅值屋脊。这一算法首先将梯度角 $\theta[i,j]$ 的变化范围减小到圆周的四个扇区之一，如图所示。四个扇区的标号为0到3，对应着 3×3 邻域内元素的四种可能组合，任何通过邻域中心的点必通过其中一个扇区。



*方向角：

$$\alpha[i, j] = \text{Sector}(\theta[i, j])$$

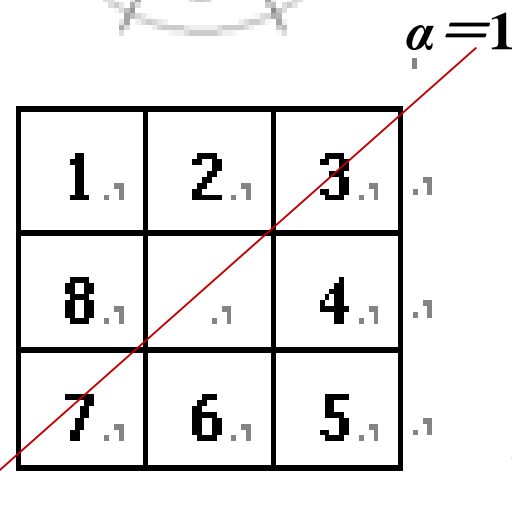
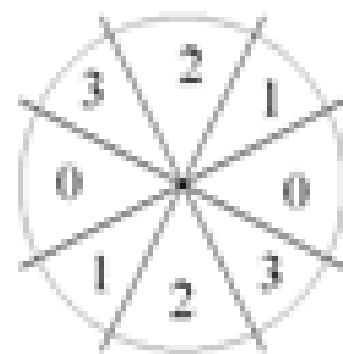
1.	2.	3.	.
8.	.	4.	.
7.	6.	5.	.

算法使用一个 3×3 邻域作用于幅值阵列 $M[i,j]$ 的所有点。在每一点上，邻域的中心像素 $M[i,j]$ 与沿着梯度线的两个元素进行比较，其中梯度线是由邻域的中心点处的扇区值 $\alpha[i,j]$ 给出的。如果在邻域中心点处的幅值 $M[i,j]$ 不比沿梯度线方向上的两个相邻点幅值大，则 $M[i,j] = 0$ 。这一过程可以把 $M[i,j]$ 宽屋脊带细化成只有一个像素点宽。在非极大值抑制过程中，保留了屋脊的高度值。

*幅值: $N[i,j] = \text{NMS}(M[i,j], \alpha[i,j])$

表示非极大值抑制过程。 $N[i,j]$ 中的非零值对应着图像强度阶跃变化处的对比度。尽管在边缘检测的第一步对图像进行了平滑，但非极大值抑制幅值图像 $N[i,j]$ 仍会包含许多由噪声和细纹理引起的假边缘段。

减少假边缘段数量的典型方法是阈值法。



5) 阈值化

- * 将低于阈值的所有值赋零值，得到图像的边缘阵列。
- * 阈值 T 太低→假边缘。
- * 阈值 T 太高→部分轮廓丢失。
- * 选用两个阈值：更有效的阈值方案。

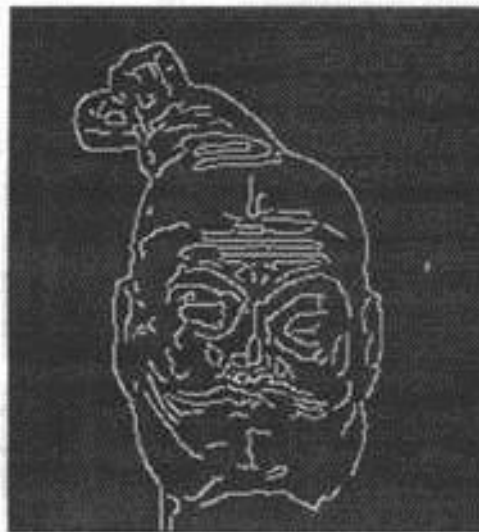
高阈值边缘图像含有很少的假边缘，但有间断（不闭合）

基本思想（双阈值法）：

- ① 取高低两个阈值 t_1, t_2 作用在幅值图 $N[i, j]$ ， $t_2 = 2t_1$ ；
- ② 得到两个边缘图 $N_1[i, j]$ 和 $N_2[i, j]$ ：低阈值和高阈值边缘图；
- ③ 在高阈值边缘图中将边缘连接成轮廓，出现断点时，在低阈值边缘图中的8邻点域搜寻边缘点，直到将 $N_2[i, j]$ 连接起来为止。



(a)



(b)



(c)

7×7 高斯滤波模板

13×13 高斯滤波模板

§ 3-3 边缘和线的检测

三、模板匹配：

用预先给定的几种模板在图像上各局部区域进行匹配运算的方法

1、模板类型：

■ 方向模板（边缘检测算子）

应用原理：

将一组方向模板分别在图像上按光栅顺序移动，并在每一个像素的相应邻域内作匹配处理。

当该像素处在某一方向的边缘，则与相应的方向模板间有最高的相似度，即匹配程度最高，而与其它方向模板的相似度较小，当该像素不处在边缘，则与所有的方向模板间的相似度都较小。

由各像素上相似度的大小可知边缘存在与否，并由最高相似度所对应的模板可求得该处的边缘方向。

§ 3-3 边缘和线的检测

线的检测

通过比较典型模板的计算值，确定一个点是否在某个方向的线上

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

水平

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

45°

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

垂直

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

135°

方向模板

■ 线的检测

1	1	1	1	1	1	1	1	1
5	5	5	5	5	5	5	5	5
1	1	1	1	1	1	1	1	1

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

水平

$$R_1 = -6 + 30 = 24$$

水平方向存在边缘

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

45°

$$R_2 = -14 + 14 = 0$$

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

垂直

$$R_2 = -14 + 14 = 0$$

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

135°

$$R_2 = -14 + 14 = 0$$

请在实验课上实现

演示

§ 3-3 边缘和线的检测

线模板:

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

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

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

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

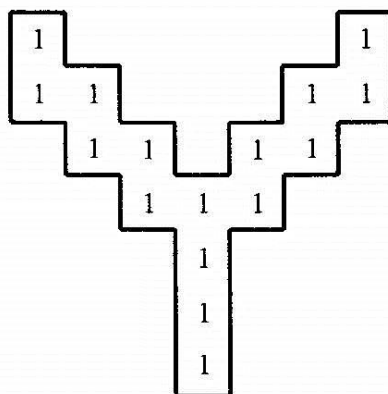
(d)

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

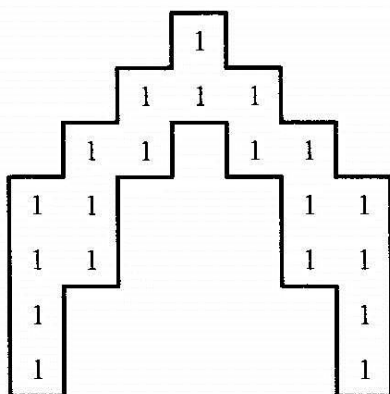
图 点模板

1	2	3	2	1
2	-1	-2	-3	-3
3	-2			
2	-3			
1	-3			

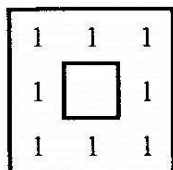
(d),



(c)



(b)



(a)

图 特定形状模板

(a) 空心正方形; (b) Λ 形; (c) Y 形; (d) Γ 形

形状模板:

§ 3-3 边缘和线的检测

形状模板检测原理：

通过求出形状模板在图像各位置的匹配值，来判断对象物存在的位置。

2、模板匹配处理：（以点模板为例说明模板匹配的基本概念）

a、模板匹配处理是一种邻域运算形式；

b、本例中采用邻域边界不处理方法，即保留原值；

c、采用匹配度量方式：指用某一关系式来计算模板上的权值与被检测图像上某像素的邻域内灰度的对应关系，从而来形容两者的相似度，即匹配程度。

点的检测

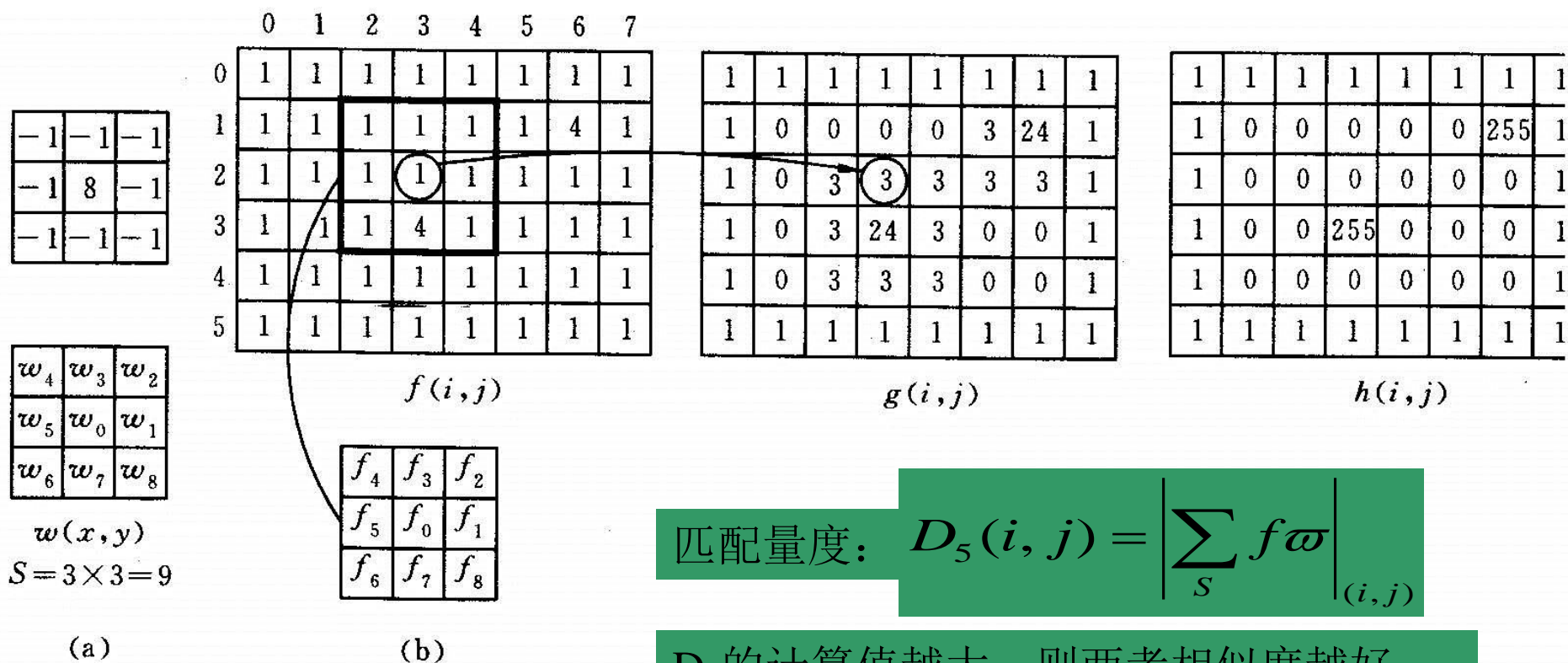


图 模板匹配处理

(a) 点模板; (b) 原图像; (c) 匹配计算后图像; (d) 点检测结果



■ 编程实现多种边缘检测方法