

## 空域滤波增强

空域滤波增强也称为模板操作,主要以像素邻域为基础对图像进行增强,增强函数  $E()$  定义在像素点  $(x, y)$  的某个邻域上。模板是指滤波器、核、掩模或窗口。邻域可以是任意形状,通常采用正方形或矩形阵列。

空域滤波增强的基本原理是利用图像与模板的卷积来进行。模板通常取尺寸为  $n \times n$  的小图像,  $n$  一般为奇数,此时,模板的半径定义为:  $r = (n - 1) / 2$ 。

基于卷积的空域滤波增强的步骤为:

- (1) 将模板在原始图像中漫游,并将模板中心与原始图像中的某个像素重合;
- (2) 将模板上的各系数与模板下对应像素的灰度值相乘,再将所有乘积相加;
- (3) 将运算结果(模板的输出响应)赋值给变换图像中对应模板中心位置的像素。

图像卷积操作示意图如图 5-1 所示,其中,图 5-1(a)为原始图像的一部分,  $s_i$  为像素的灰度值;图 5-1(b)为一个  $3 \times 3$  的模板,  $k_i$  为模板系数,不同的系数将得到不同的滤波(平滑或锐化)效果;图 5-1(c)为变换图像,将  $k_0$  所在的模板中心位置与原始图像中灰度值为  $s_0$  的像素重合,可得到模板的输出响应  $t = k_0 s_0 + k_1 s_1 + \dots + k_8 s_8$ ,然后将  $t$  赋值给变换图像中对应模板中心位置的像素即完成滤波。

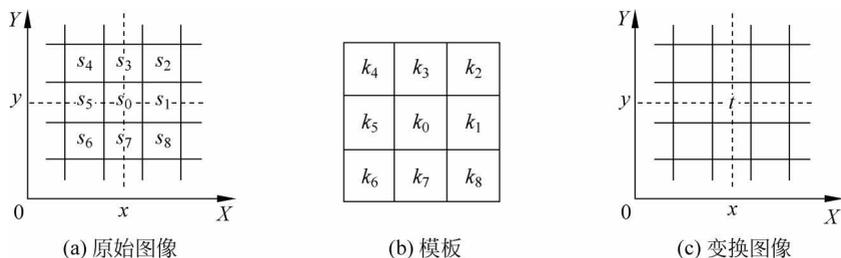


图 5-1 图像卷积操作示意图

按照功能进行分类,空域滤波增强可以分为平滑滤波和锐化滤波。平滑滤波属于低通滤波,即减弱或消除图像中的高频分量(高频分量对应图像中的区域边缘等灰度值具有较大、较快变化的部分),不影响低频分量,可达到减少局部灰度起伏、平滑图像的目的。锐化滤波属于高通滤波,即减弱或消除图像中的低频分量(低频分量对应图像中灰度值变化缓慢

的部分),不影响高频分量,可达到增强局部灰度反差、锐化目标边缘的目的。

按照模板特点进行分类,空域滤波增强可以分为线性滤波和非线性滤波。线性滤波是指对模板涉及的像素值进行线性组合,该方法计算方便,容易并行实现。非线性滤波是指对模板涉及的像素值进行逻辑组合,该方法具有较好的滤波效果。

因此,空域滤波增强技术通常包括线性平滑滤波、非线性平滑滤波、线性锐化滤波和非线性锐化滤波。

## 5.1 卷积原理

卷积来源于信号与系统分析,在物理学、工程科学和数学中经常出现。在信号与系统分析中,对于给定的线性时不变系统,它的零状态响应可以通过系统的冲激响应  $h(t)$  与  $e(t)$  的卷积积分来求得。

$$r(t) = \int_{-\infty}^{+\infty} e(\tau) \times h(t - \tau) d\tau \quad (5-1)$$

### 5.1.1 一维连续卷积

一维连续卷积如下式所示:

$$h(x) = f(x) \otimes g(x) = \int_{-\infty}^{+\infty} f(u)g(x - u) du \quad (5-2)$$

式中,  $f(x)$  和  $g(x)$  为两个一维连续函数。

**例 5.1** 一维连续卷积图解。若函数  $f(x)$  和  $g(x)$  如图 5-2 所示,试求:卷积函数  $h(x)$ 。

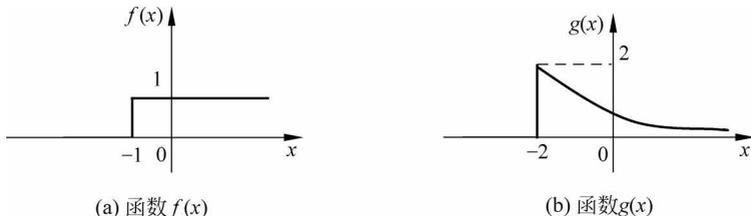


图 5-2 卷积函数

**解:** 卷积过程图解如图 5-3(a)~图 5-3(g)所示,其中,图 5-3(a)为函数  $f(u)$ ,图 5-3(b)为函数  $g(u)$ ,图 5-3(c)为函数  $f(u)$  和  $g(-u)$ ,图 5-3(d)为函数  $f(u)$  和  $g(x-u)$  ( $x_1 > 0$ ),图 5-3(e)为函数  $f(u)$  和  $g(x-u)$  ( $x_2 < 0$ ),图 5-3(f)为函数  $f(u)$  和  $g(x-u)$  ( $x_3 < -3$ ),图 5-3(g)为卷积函数  $h(x)$ 。

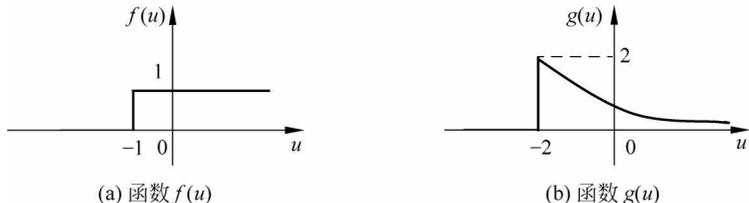


图 5-3 一维连续卷积图解

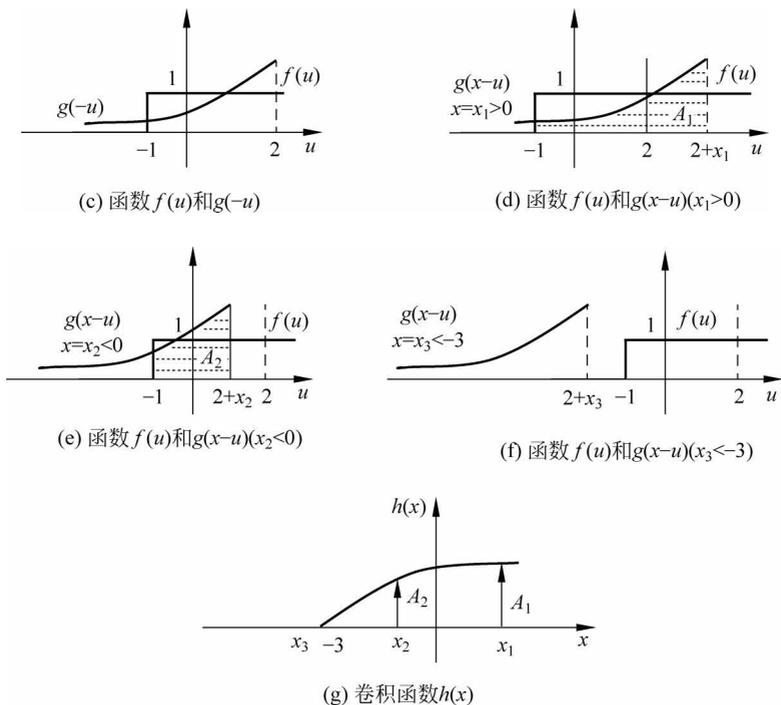


图 5-3 (续)

### 5.1.2 一维离散卷积

一维离散卷积如下式所示：

$$h(i) = f(i) \otimes g(i) = \sum_{u=-\infty}^{\infty} f(u)g(i-u) = \sum_{u=0}^i f(u)g(i-u) \quad (5-3)$$

式中,  $f(i)$  为一维离散函数,  $i \in [0, m-1]$ , 即大小为  $1 \times m$ ;  $g(i)$  为一维离散函数,  $i \in [0, n-1]$ , 即大小为  $1 \times n$ ;  $h(i)$  为卷积结果,  $i \in [0, m+n-1]$ , 即大小为  $m+n-1$ 。

一维原始矩阵与一维模板矩阵相卷积, 卷积过程的步骤为：

(1) 基于卷积定义, 将模板矩阵旋转  $180^\circ$ ;

(2) 将模板矩阵从左到右在原始矩阵中滑动, 计算模板矩阵与原始矩阵交集元素的乘积和, 该和即为卷积结果中对应位置的数值。

**例 5.2** 一维离散卷积示例。

**解：**图 5-4(a) 为一维原始矩阵  $A$ ; 图 5-4(b) 为一维模板矩阵  $B$ ; 图 5-4(c) 为原始矩阵  $A$  与模板矩阵  $B$  的卷积结果; 图 5-4(d)~图 5-4(h) 为卷积过程的步骤。从结果可以看出,

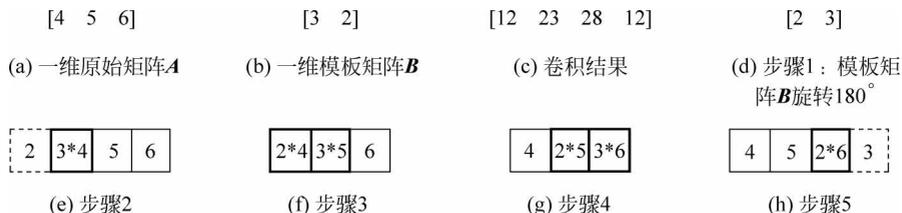


图 5-4 一维离散卷积示例

卷积过程首先将模板矩阵  $\mathbf{B}$  旋转  $180^\circ$ , 然后从左到右在原始矩阵中滑动, 卷积结果中对应位置的数值等于模板矩阵与原始矩阵交集元素的乘积和。

### 5.1.3 二维连续卷积

二维连续卷积如下式所示:

$$h(x, y) = f(x, y) \otimes g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(u, v) g(x-u, y-v) du dv \quad (5-4)$$

式中,  $f(x, y)$  和  $g(x, y)$  为两个二维连续函数。

**例 5.3** 二维连续卷积示例(图 5-5)。

**解:** 二维连续卷积示例如图 5-5 所示, 函数  $g(x, y)$  为平行于  $XY$  平面、值为 1 的平面。对于给定的某个  $u, v$  值, 函数  $f(u, v)$  与  $g(x-u, y-v)$  的卷积结果  $h(x, y)$  为以两函数乘积对应曲面的边缘曲线为准线、母线平行于  $Z$  轴的曲顶柱体的体积。

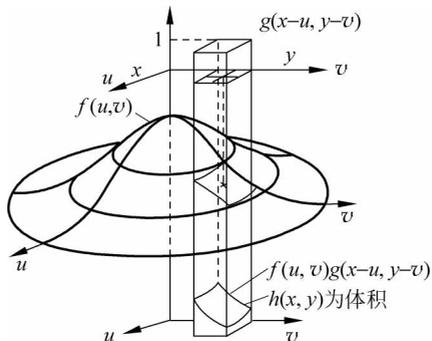


图 5-5 二维连续卷积示例

### 5.1.4 二维离散卷积

二维离散卷积如下式所示:

$$\begin{aligned} h(i, j) &= f(i, j) \otimes g(i, j) = \sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} f(u, v) g(i-u, j-v) \\ &= \sum_{u=0}^i \sum_{v=0}^j f(u, v) g(i-u, j-v) \end{aligned} \quad (5-5)$$

式中,  $f(i, j)$  为二维离散函数,  $i \in [0, m-1], j \in [0, n-1]$ , 即大小为  $m \times n$ ;  $g(i, j)$  为二维离散函数,  $i \in [0, p-1], j \in [0, q-1]$ , 即大小为  $p \times q$ ;  $h(i, j)$  为卷积结果,  $i \in [0, m+p-1], j \in [0, n+q-1]$ , 即大小为  $(m+p-1) \times (n+q-1)$ 。

二维原始矩阵与二维模板矩阵相卷积, 卷积过程的步骤为:

(1) 基于卷积定义, 将模板矩阵旋转  $180^\circ$ ;

(2) 将模板矩阵从上到下、从左到右在原始矩阵中滑动, 计算模板矩阵与原始矩阵交集元素的乘积和, 该和即为卷积结果中对应位置的数值。

**例 5.4** 二维离散卷积示例。

**解:** 图 5-6(a) 为二维原始矩阵  $\mathbf{A}$ ; 图 5-6(b) 为二维模板矩阵  $\mathbf{B}$ ; 图 5-6(c) 为原始矩阵  $\mathbf{A}$  与模板矩阵  $\mathbf{B}$  的卷积结果; 图 5-6(d)~图 5-6(p) 为卷积过程的步骤。从结果可以看出,

卷积过程首先将模板矩阵  $B$  旋转  $180^\circ$ , 然后从上到下、从左到右在原始矩阵中滑动, 卷积结果中对应位置的数值等于模板矩阵与原始矩阵交集元素的乘积和。

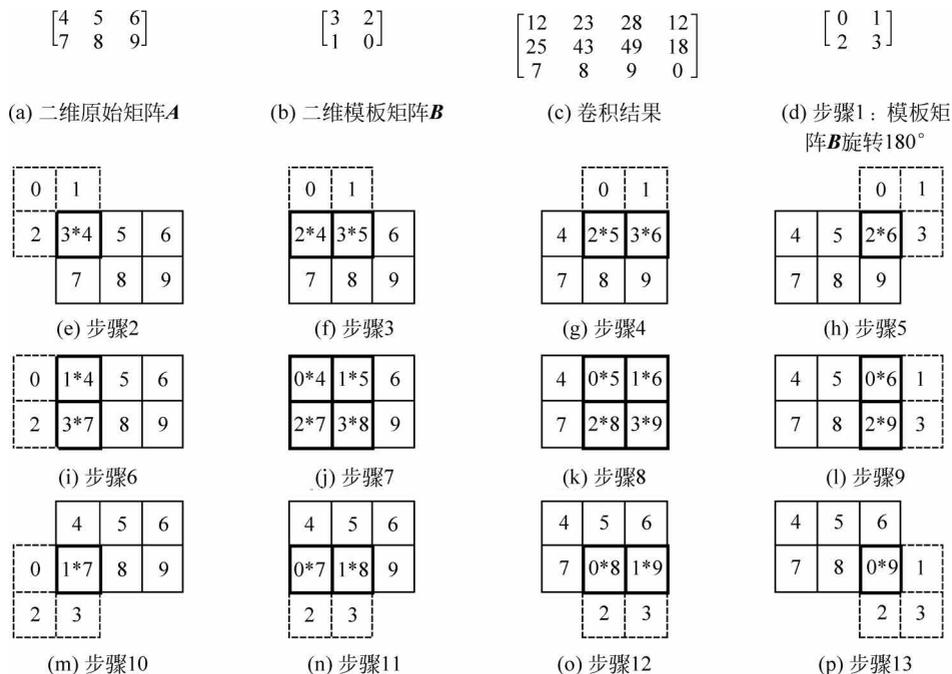


图 5-6 二维离散卷积示例

```
% F5_6.m

disp('步骤 1: 生成 3×3 的矩阵 A 和 B');
A = [4 5 6
     7 8 9]
B = [3 2
     1 0]

rotB = rot90(B,2)

disp('步骤 2: 验证利用卷积函数直接进行卷积');
AconvB = conv2(A,B)
BconvA = conv2(B,A)
```

## 5.2 线性平滑滤波

若原始图像  $f(x, y)$  的大小为  $M \times N$ , 变换图像为  $g(x, y)$ , 则用大小为  $m \times n$  ( $m, n$  一般为奇数) 的模板  $k$  进行线性平滑滤波的原理可由下式给出:

$$g(x, y) = \sum_{i=-a}^a \sum_{j=-b}^b k(i, j) f(x+i, y+j) \quad (5-6)$$

式中,  $x \in [0, M-1], y \in [0, N-1], a = (m-1)/2, b = (n-1)/2$ 。

若仅考虑对某一点  $(x, y)$  进行卷积处理得到的输出响应  $t$ , 则式(5-6)可简单表示为

$$t = k_0 s_0 + k_1 s_1 + \dots + k_{mn-1} s_{mn-1} = \sum_{i=0}^{mn-1} k_i s_i \quad (5-7)$$

式中,  $t$  为变换图像  $g(x, y)$  中对应点  $(x, y)$  的灰度值;  $s$  为原始图像  $f(x, y)$  中某一点  $(x, y)$  的灰度值。

对原始图像  $f(x, y)$  中的每一个点  $(x, y)$  分别进行滤波处理可得到变换图像  $g(x, y)$ 。当  $m=n=3$  时, 线性平滑滤波原理示意图如图 5-7 所示。

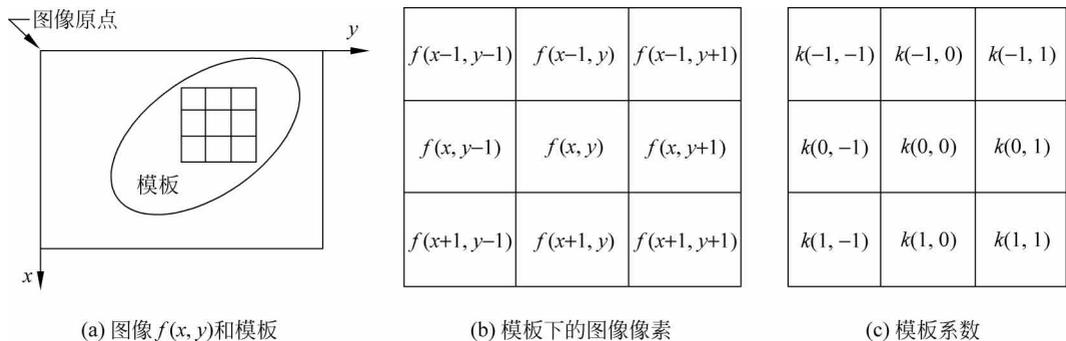


图 5-7 线性平滑滤波原理示意图

### 5.2.1 邻域平均法

设原始图像  $f(x, y)$  的大小为  $M \times N$ , 变换图像为  $g(x, y)$ , 则变换图像每个像素  $(x, y)$  的灰度值由包含该像素的预定邻域内像素灰度值的平均值决定, 如下式所示:

$$g(x, y) = \frac{1}{k} \sum_{(i, j) \in S} f(i, j) \quad (5-8)$$

式中,  $x \in [0, M-1], y \in [0, N-1]$ ;  $S$  为像素  $(x, y)$  的预定邻域;  $k$  为  $S$  内像素点的个数。

邻域平均法的典型邻域模板如图 5-8 所示, 模板的共同特点是系数非负且和为 1。

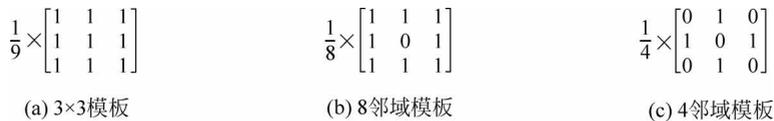


图 5-8 邻域平均法的典型邻域模板

**例 5.5** 邻域平均法原理示例。一张 8 比特灰度图像的矩阵数据如图 5-9(a) 所示, 试采用  $3 \times 3$  模板的邻域平均法对其进行平滑滤波。

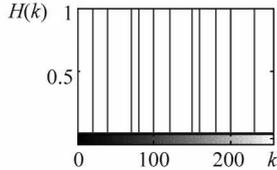
**解:** 基于原始图像的矩阵数据, 得到对应的原始图像如图 5-9(b) 所示, 其灰度统计直方图如图 5-9(c) 所示。采用图 5-9(d) 所示的  $3 \times 3$  模板基于邻域平均法对原始图像进行平滑滤波, 得到的变换图像原始数据如图 5-9(e) 所示, 对该数据进行四舍五入处理得到的变换图像数据如图 5-9(f) 所示, 对应的变换图像如图 5-9(g) 所示, 其灰度统计直方图如图 5-9(h) 所示。

$$\begin{bmatrix} 0 & 20 & 40 & 70 \\ 80 & 100 & 120 & 150 \\ 160 & 180 & 200 & 230 \end{bmatrix}$$

(a) 原始图像数据



(b) 原始图像



(c) 原始图像灰度统计直方图

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

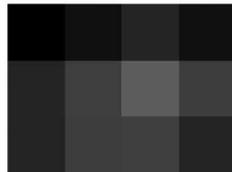
(d) 3×3模板

$$\begin{bmatrix} 22.2222 & 40.0000 & 55.5556 & 42.2222 \\ 60.0000 & 100.0000 & 123.3333 & 90.0000 \\ 57.7778 & 93.3333 & 108.8889 & 77.7778 \end{bmatrix}$$

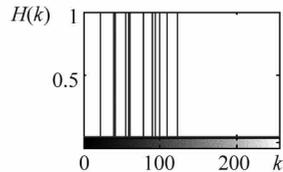
(e) 变换图像原始数据

$$\begin{bmatrix} 22 & 40 & 56 & 42 \\ 60 & 100 & 123 & 90 \\ 58 & 93 & 109 & 78 \end{bmatrix}$$

(f) 变换图像数据



(g) 变换图像



(h) 变换图像灰度统计直方图

图 5-9 邻域平均法的典型邻域模板

```
% F5_9.m

% % -- 设置图片和坐标轴的属性 -----
% fs = 7.5;           % FontSize: 五号:10.5磅,小五号:9磅,六号:7.5磅
% ms = 5;            % MarkerSize: 五号:10.5磅,小五号:9磅,六号:7.5磅
% set(gcf,'Units','centimeters','Position',[25 11 6 4.5]);
%           % 设置图片的位置和大小[left bottom width height],width:height = 4:3
% set(gca,'FontName','宋体','FontSize',fs); % 设置坐标轴(刻度、标签和图例)的字体和字号
% set(gca,'Position',[.12 .11 .83 .77]); % 设置坐标轴所在的矩形区域在图片中的位置
%           % [left bottom width height]

Fxy = [ 0 20 40 70
        80 100 120 150
        160 180 200 230]
uint8Fxy = uint8(Fxy)
subplot(2,2,1), imshow(uint8Fxy), xlabel('(b) 原始图像');

subplot(2,2,2), imhist(uint8Fxy), axis([0,255,0,1]), xlabel('(c) 原始图像灰度统计直方图', 'position',[120, -0.23]);
```

```

text(255, -0.17, '\itk', 'FontName', 'Times New Roman');
text(-70, 1, '\itH{\itk}', 'FontName', 'Times New Roman');

Gxy = filter2(fspecial('average',3), uint8Fxy)
uint8Gxy = uint8(Gxy)
subplot(2,2,3), imshow(uint8Gxy), xlabel('(g) 变换图像');

subplot(2,2,4), imhist(uint8Gxy), axis([0,255,0,1]), xlabel('(h) 变换图像灰度统计直方图',
position',[120, -0.23]);
text(255, -0.17, '\itk', 'FontName', 'Times New Roman');
text(-70, 1, '\itH{\itk}', 'FontName', 'Times New Roman');

```

邻域平均法常用于消除噪声。设：噪声是加性白噪声，均值为 0，方差为  $\sigma^2$ ，且噪声与原始图像  $f(x, y)$  互不相关，则采集图像  $g(x, y)$  可以看作是原始图像  $f(x, y)$  和噪声图像  $e(x, y)$  的叠加，即

$$g(x, y) = f(x, y) + e(x, y) \quad (5-9)$$

采用邻域平均法对采集图像进行处理：

$$g'(x, y) = \frac{1}{k} \sum_{(i,j) \in S} f(i, j) + \frac{1}{k} \sum_{(i,j) \in S} e(i, j) \quad (5-10)$$

处理后残余噪声的均值和方差分别为

$$E\left(\frac{1}{k} \sum_{(i,j) \in S} e(i, j)\right) = 0 \quad (5-11)$$

$$D\left(\frac{1}{k} \sum_{(i,j) \in S} e(i, j)\right) = \frac{1}{k} \sigma^2 \quad (5-12)$$

上式表明：经邻域平均法处理后，残余噪声的均值为 0，方差减小为原来的  $1/k$ ，但原始图像  $f(x, y)$  变为  $\frac{1}{k} \sum_{(i,j) \in S} f(i, j)$ ，这将引起原始图像中目标轮廓模糊或细节特征消失。

#### 例 5.6 邻域平均法实现消除噪声。

**解：**原始图像如图 5-10(a) 所示，原始图像添加噪声密度（即包括噪声值的图像区域的百分比）为 4% 的椒盐噪声后得到的噪声图像如图 (b) 所示。采用  $3 \times 3$  模板、 $5 \times 5$  模板、4 邻域模板和 8 邻域模板分别对噪声图像进行邻域平均法滤波的结果如图 5-10(c)~图 5-10(f) 所示。从结果可以看出，噪声均得到明显消除，其中， $5 \times 5$  模板对噪声的消除效果明显强于  $3 \times 3$  模板，因为离模板中心像素越近的像素对滤波结果的贡献越大， $5 \times 5$  模板采用对某像素点及其周围共 25 个像素点求平均来实现滤波， $3 \times 3$  模板采用对某像素点及其周围共 9 个像素点求平均来实现滤波， $5 \times 5$  模板包含了更多离模板中心像素较远的像素参与滤波，因而平滑和消噪效果更强。

同时可以看出，在消除噪声时，图像轮廓等细节也得到不同程度的消除。一般来说，噪声消除越好，图像细节消除也越强，这正是邻域平均法进行平滑滤波的缺点。

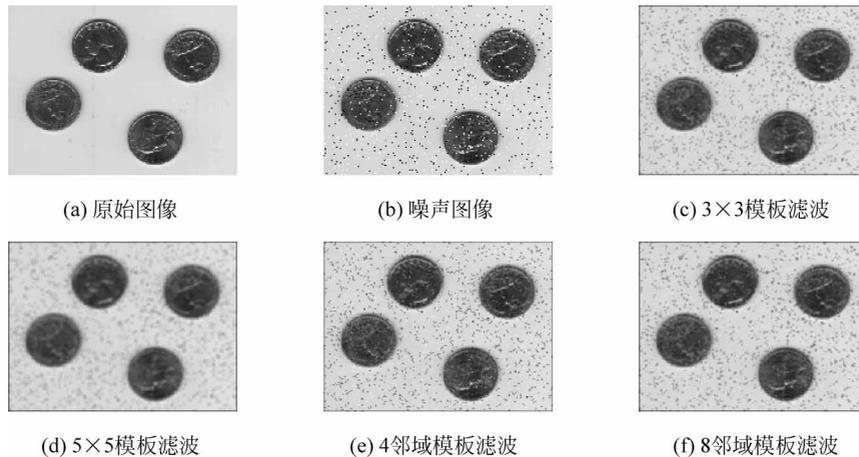


图 5-10 邻域平均法消除噪声示例

```

% F5_10.m

I = imread('eight.tif');
subplot(2,3,1), imshow(I), xlabel('(a) 原始图像');

J = imnoise(I, 'salt & pepper', 0.04);
subplot(2,3,2), imshow(J), xlabel('(b) 噪声图像');

K1 = filter2(fspecial('average',3),J);
subplot(2,3,3), imshow(uint8(K1)), xlabel('(c) 3 * 3 模板均值滤波');

K2 = filter2(fspecial('average',5),J);
subplot(2,3,4), imshow(uint8(K2)), xlabel('(d) 5 * 5 模板均值滤波');

mask1 = [0 1 0
         1 0 1
         0 1 0];
mask1 = (1/4) * mask1;
K3 = filter2(mask1,J);
subplot(2,3,5), imshow(uint8(K3)), xlabel('(e) 4 邻域模板均值滤波');

mask2 = [1 1 1
         1 0 1
         1 1 1];
mask2 = (1/8) * mask2;
K4 = filter2(mask2,J);
subplot(2,3,6), imshow(uint8(K4)), xlabel('(f) 8 邻域模板均值滤波');

```

### 5.2.2 选择平均法

选择平均法以邻域平均法为基础,对灰度相同或相近的像素进行平均,或者按照灰度的特殊程度加权之后再求和,以避免或减轻目标边缘、轮廓等细节特征的模糊。

**例 5.7** 选择平均法原理示例。

**解：**原始图像如图 5-11(a)所示,图像为白色,大小为  $20 \times 20$ ,中间有一个  $8 \times 8$  的黑色方块。原始图像具有图 5-11(b)所示的特征,即目标边缘在某个  $3 \times 3$  邻域内,  $A_1, A_2, \dots, A_5$  灰度值相似,  $B_1, B_2, \dots, B_4$  灰度值相似,表明  $A_i$  与  $B_j$  之间存在水平和垂直边缘。基于这一特征,选择如图 5-11(c)所示的  $3 \times 3$  模板对原始图像进行选择平均法滤波,结果如图 5-11(d)所示。为了比较选择平均法和邻域平均法的滤波效果,采用邻域平均法对原始图像进行滤波,结果如图 5-11(e)所示。从结果可以看出,选择平均法对目标边缘的模糊效果明显优于邻域平均法。

原始图像添加噪声密度(即包括噪声值的图像区域的百分比)为 4% 的椒盐噪声后得到的噪声图像如图 5-11(f)所示,采用选择平均法和邻域平均法对噪声图像进行滤波的结果分别如图 5-11(g)和图 5-11(h)所示。从结果可以看出,选择平均法对目标边缘的模糊效果明显优于邻域平均法。

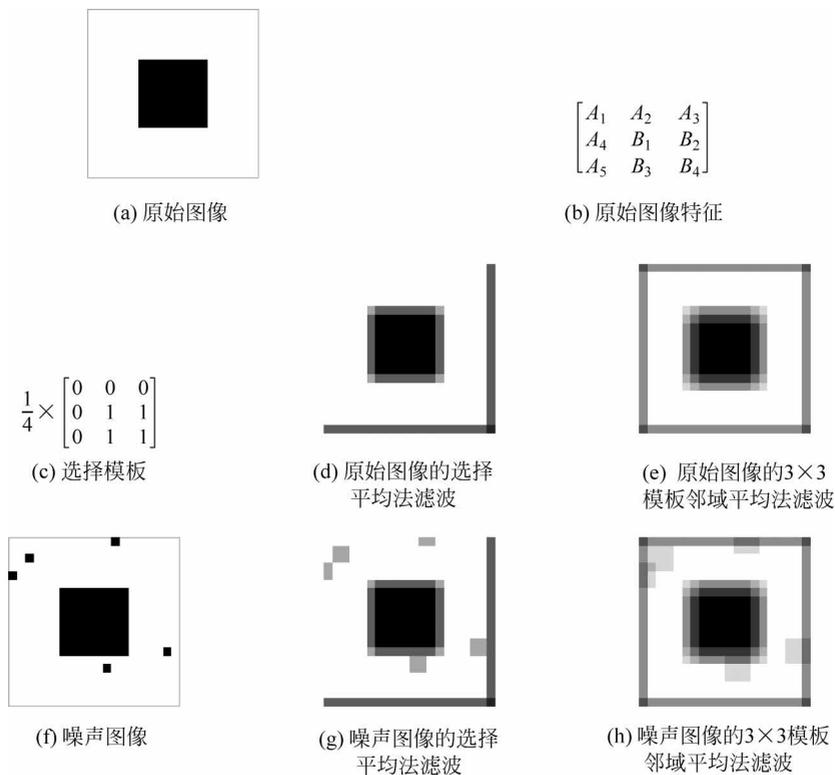


图 5-11 选择平均法原理示例

```
% F5_11.m

I = imread('F5_11a_.bmp');
subplot(2,3,1), imshow(I), xlabel('(a) 原始图像'), ylabel('无噪声滤波过程');

K1 = filter2(fspecial('average',3),I);
```

```

subplot(2,3,2), imshow(uint8(K1)), xlabel('(d) 3 * 3 模板均值滤波');

mask = [0 0 0
        0 1 1
        0 1 1];
mask = (1/4) * mask;
K2 = filter2(mask, I);
subplot(2,3,3), imshow(uint8(K2)), xlabel('(e) 选择模板均值滤波');

J = imnoise(I, 'salt & pepper', 0.04);
subplot(2,3,4), imshow(J), xlabel('(f) 噪声图像'), ylabel('有噪声滤波过程');

K3 = filter2(fspecial('average', 3), J);
subplot(2,3,5), imshow(uint8(K3)), xlabel('(g) 3 * 3 模板均值滤波');

mask = [0 0 0
        0 1 1
        0 1 1];
mask = (1/4) * mask;
K4 = filter2(mask, J);
subplot(2,3,6), imshow(uint8(K4)), xlabel('(h) 选择模板均值滤波');

```

常用的选择平均法包括阈值法和半邻域法。

### 5.2.2.1 阈值法

阈值法的基本思想：当像素灰度值与其邻域内像素灰度值的均值之差较大时，该像素必然是噪声，滤波结果为该像素邻域内像素灰度值的均值；当像素灰度值与其邻域内像素灰度值的均值之差较小时，滤波结果仍为该像素的值。

$$g(x, y) = \begin{cases} \frac{1}{M} \sum_{(i,j) \in S} f(i, j), & \left| f(x, y) - \frac{1}{M} \sum_{(i,j) \in S} f(i, j) \right| > T \\ f(x, y), & \text{其他} \end{cases} \quad (5-13)$$

式中， $S$  为像素  $(x, y)$  的某个邻域； $M$  为  $S$  内像素点的个数； $T$  为非负阈值。

### 5.2.2.2 半邻域法

半邻域法的基本步骤是：

- (1) 设模板为  $3 \times 3$  的矩阵；
- (2) 将模板中心与原始图像中的像素  $P$  重合，对模板覆盖的 8 个像素进行排序，灰度值较小的 3 个像素构成  $A$  组，灰度值较大的 5 个像素构成  $B$  组；
- (3) 设定门限阈值  $T$  并计算  $A$ 、 $B$  两组的平均值  $A_{\text{avg}}$  和  $B_{\text{avg}}$ ；
- (4) 若  $|A_{\text{avg}} - B_{\text{avg}}| \leq T$ ，则认为无边缘通过，采用 8 邻域平均法滤波；反之，则认为有边缘通过，将像素  $P$  与  $B$  组中的 5 个像素共 6 个像素进行平均。

### 5.2.3 加权平均法

加权平均法的基本思想：离模板中心越近的像素对滤波结果的贡献越大，因此，可使接近模板中心的系数大些而远离模板中心的系数小些。

模板系数的确定一般遵循如下原则：

- (1) 模板中心的系数最大；
- (2) 模板周边最小的系数为 1；
- (3) 模板内部的系数根据与模板中心的距离反比地确定，一般成比例地增加，常取 2 的整数次幂，以便于计算机实现。

高斯(Gause)模板是加权平均法的常用模板,如图 5-12 所示,模板系数由采样二维高斯函数得到,模板的特点是系数非负且和为 1。

$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

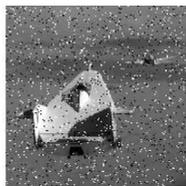
**例 5.8** 高斯模板滤波示例。

**解：**原始图像如图 5-13(a)所示,原始图像添加噪声密度(即包括噪声值的图像区域的百分比)为 6% 的椒盐噪声后得到的噪声图像如图 5-13(b)所示,采用高斯模板进行加权平均法滤波的结果如图 5-13(c)所示。为了比较加权平均法和邻域平均法的滤波效果,采用  $3 \times 3$  模板、4 邻域模板和 8 邻域模板分别对噪声图像进行邻域平均法滤波的结果如图 5-13(d)~图 5-13(f)所示。从结果可以看出,基于高斯模板的加权平均法对噪声的消除优于邻域平均法。

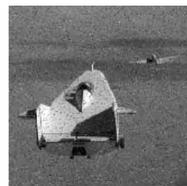
图 5-12 高斯模板



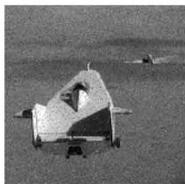
(a) 原始图像



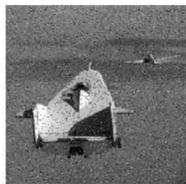
(b) 噪声图像



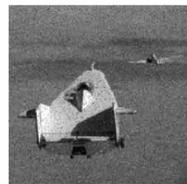
(c) 高斯模板加权平均法滤波



(d)  $3 \times 3$ 模板邻域平均法滤波



(e) 4邻域模板邻域平均法滤波



(f) 8邻域模板邻域平均法滤波

图 5-13 高斯模板滤波示例

```
% F5_13.m

I = imread('liftingbody.png');
subplot(2,3,1), imshow(I), xlabel('(a) 原始图像');

J = imnoise(I, 'salt & pepper', 0.06);
subplot(2,3,2), imshow(J), xlabel('(b) 噪声图像');

mask1 = [1 2 1
         2 4 2
         1 2 1];
mask1 = (1/16) * mask1;
K1 = filter2(mask1, J);
```

```

subplot(2,3,3), imshow(uint8(K1)), xlabel('(c) 高斯模板均值滤波');

K2 = filter2(fspecial('average',3),J);
subplot(2,3,4), imshow(uint8(K2)), xlabel('(d) 3 * 3 模板均值滤波');

mask3 = [0 1 0
         1 0 1
         0 1 0];
mask3 = (1/4) * mask3;
K3 = filter2(mask3,J);
subplot(2,3,5), imshow(uint8(K3)), xlabel('(e) 4 邻域模板均值滤波');

mask4 = [1 1 1
         1 0 1
         1 1 1];
mask4 = (1/8) * mask4;
K4 = filter2(mask4,J);
subplot(2,3,6), imshow(uint8(K4)), xlabel('(f) 8 邻域模板均值滤波');

```

### 5.2.4 Wiener 滤波

Wiener(维纳)滤波器是一款经典的线性降噪滤波器,提出于20世纪40年代。Wiener滤波是一种在平稳条件下采用最小均方误差准则得出的最佳滤波准则,该方法将寻找一个最佳的线性滤波器,使得均方误差最小,其实质是求解维纳-霍夫(Wiener-Hoof)方程。

设原始图像  $f(x, y)$  的大小为  $M \times N$ , 变换图像为  $g(x, y)$ , 则 Wiener 滤波的基本原理如下式所示:

$$g(x, y) = \frac{\sigma^2 - \nu^2}{\sigma^2} (f(x, y) - \mu) + \mu \quad (5-14)$$

式中,  $x \in [0, M-1]$ ,  $y \in [0, N-1]$ ;  $\nu^2$  是整幅图像的方差;  $\mu$  和  $\sigma^2$  分别为像素  $(x, y)$  的邻域内像素的均值和方差; 最后将  $\mu$  加回是为了恢复原始邻域的平均灰度值, 由下式给出:

$$\mu = \frac{1}{mn} \sum_{(i,j) \in S} f(i, j) \quad (5-15)$$

$$\sigma^2 = \frac{1}{mn} \sum_{(i,j) \in S} f^2(i, j) - \mu^2 \quad (5-16)$$

式中,  $S$  为像素  $(x, y)$  的  $m \times n$  邻域。

Wiener 滤波器根据图像的局部方差来调整滤波器的输出, 是一种自适应的滤波器。对于图像中对比度较小的区域, 其局部方差就较小, 而滤波器的效果反而较强, 达到局部增强的目的; 对于图像中对比度较大的区域, 其局部方差就较大, 而滤波器的效果反而较弱, 达到局部平滑的目的。

**例 5.9** Wiener 滤波示例。

**解:** 原始图像如图 5-14(a) 所示, 原始图像添加噪声密度(即包括噪声值的图像区域的百分比)为 4% 的椒盐噪声后得到的噪声图像如图 5-14(b) 所示。采用  $3 \times 3$  模板、 $5 \times 5$  模板、 $7 \times 7$  模板和  $9 \times 9$  模板分别对噪声图像进行 Wiener 滤波的结果如图 5-14(c)~图 5-14(f)

所示。从结果可以看出,噪声均得到明显消除;模板越大,参与滤波的像素越多,噪声消除效果越明显。在消除噪声的同时,图像轮廓等细节也得到不同程度的消除,噪声消除越好,图像细节消除也越强,这也是 Wiener 滤波器进行平滑滤波的缺点。

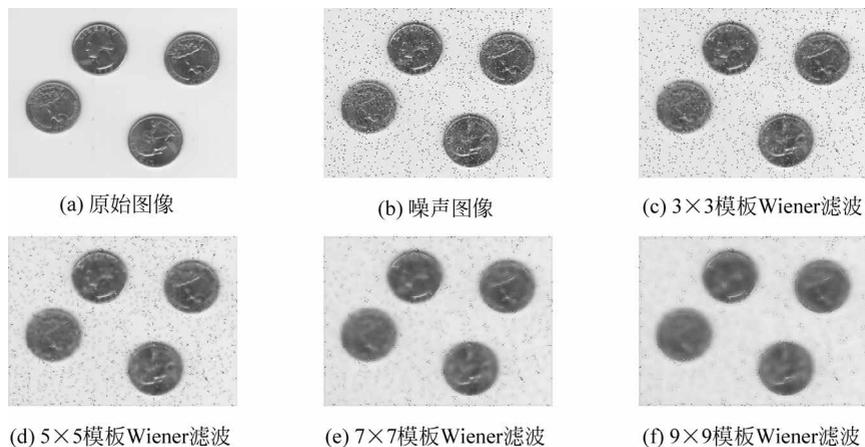


图 5-14 Wiener 滤波示例

```
% F5_14.m

I = imread('eight.tif');
subplot(2,3,1), imshow(I), xlabel('(a) 原始图像');

J = imnoise(I, 'salt & pepper', 0.04);
subplot(2,3,2), imshow(J), xlabel('(b) 噪声图像');

K1 = wiener2(J, [3,3]);
subplot(2,3,3), imshow(uint8(K1)), xlabel('(c) 3 * 3Wiener 滤波');

K2 = wiener2(J, [5,5]);
subplot(2,3,4), imshow(uint8(K2)), xlabel('(d) 5 * 5Wiener 滤波');

K3 = wiener2(J, [7,7]);
subplot(2,3,5), imshow(uint8(K3)), xlabel('(e) 7 * 7Wiener 滤波');

K4 = wiener2(J, [9,9]);
subplot(2,3,6), imshow(uint8(K4)), xlabel('(f) 9 * 9Wiener 滤波');
```

### 5.3 非线性平滑滤波

非线性平滑滤波技术的分类主要包括:

- (1) 基于形状(几何)的分类。该类方法主要基于数学形态学方法进行。
- (2) 基于集合(逻辑)的分类。该类方法采用基于形状形态操作,建立在集合论的基

础上,并可以在有限的数字表达下简化为传统的逻辑表达。

(3) 基于排序(代数)的分类。该类方法被成功地应用于保留图像细节的同时消除脉冲噪声(脉冲噪声也称椒盐噪声,因为这种噪声以黑白点的形式叠加在图像上)。在与分布无关的决策理论中,排序统计已被证明为一种非常有效的方法。

本节主要介绍基于排序(代数)的非线性平滑滤波技术。

### 5.3.1 中值滤波

中值滤波是一种非线性平滑滤波技术,由 Turkey 于 1971 年提出,其基本原理是把数字图像序列中某点的值用该点邻域中各点值的中值代替。

极限像素是指与周围像素灰度值差别较大的像素。噪声属于极限像素,且往往以孤立点的形式存在,对应的像素很少;图像则由像素较多、面积较大的小块组成。中值滤波的特点包括:

(1) 中值滤波对极限像素值的敏感度远不如像素平均值,能克服线性滤波器模糊图像细节的缺点,达到既消除噪声,又保持图像细节的目的,对滤除脉冲干扰和扫描噪声最为有效;

(2) 不适宜细节多,特别是点、线、尖、顶等多的图像。

#### 5.3.1.1 一维中值滤波

一维中值滤波亦称作游程操作,可通过滑动奇数长度的模板来实现。设:一维信号序列 $\{f_i\}$ 由下式给出:

$$f(i) \begin{cases} \neq 0, & 1 \leq i \leq l \\ = 0, & i \leq 0 \parallel i \geq l + 1 \end{cases} \quad (5-17)$$

式中, $l$  为信号长度, $i=1,2,\dots,l$ 。

一维中值滤波的基本原理如下式所示:

$$g_j = \text{median}(f_{j-r}, \dots, f_j, \dots, f_{j+r}) \quad (5-18)$$

式中, $\{f_i\}$ 为一维信号序列,长度为 $l$ , $i=1,2,\dots,l$ ;  $r$ 为滤波模板半径,模板尺寸 $M=2r+1$ ;  $\text{median}()$ 为取中值函数,即对模板覆盖的信号序列按数值大小排序,并取排序后处在中间位置的值。由此可知,信号序列 $\{f_i\}$ 中有一半大于 $g_j$ ,另一半小于 $g_j$ 。

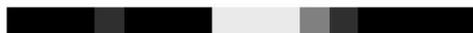
中值滤波可以完全消除孤立脉冲噪声而不对通过的理想边缘产生任何影响。

**例 5.10** 一维中值滤波噪声消除示例。一维 8 比特灰度图像的数据如图 5-15(a)所示,试采用 $1 \times 3$ 模板对其进行中值滤波。

**解:** 基于图 5-15(a)所示的原始图像数据,得到对应的原始图像及其二维示意图如图 5-15(b)~图 5-15(c)所示,图中同时存在边缘和噪声。采用 $1 \times 3$ 模板对其进行中值滤波,得到的滤波图像的二维示意图、图像数据和图像分别如图 5-15(d)~图 5-15(f)所示。从结果可以看出,噪声被完全消除,边缘没有受到任何影响。

[0 0 0 80 0 0 0 240 240 240 160 80 0 0 0 0]

(a) 原始图像数据



(b) 原始图像

图 5-15 一维中值滤波噪声消除示例

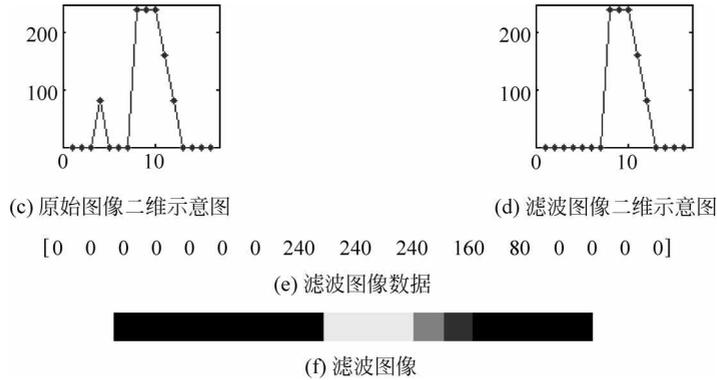


图 5-15 (续)

```

% F5_15.m

I = [0 0 0 80 0 0 0 240 240 240 160 80 0 0 0 0];
uint8I = uint8(I);
subplot(2,2,1), imshow(uint8I), xlabel('(b) 原始图像');
subplot(2,2,2), plot(I, '-'), axis([0 17 0 250]), xlabel('(c) 原始图像二维示意图');

J = medfilt2(uint8I,[1,3]);
subplot(2,2,3), plot(J, '-'), axis([0 17 0 250]), xlabel('(d) 滤波图像');

uint8J = uint8(J);
subplot(2,2,4), imshow(uint8J), xlabel('(f) 滤波图像二维示意图');
    
```

针对接近边缘的脉冲噪声,中值滤波会使边缘发生偏移,这是中值滤波的缺点。

**例 5.11** 一维中值滤波边缘偏移示例。一维 8 比特灰度图像的数据如图 5-16(a)所示,试采用  $1 \times 3$  模板对其进行中值滤波。

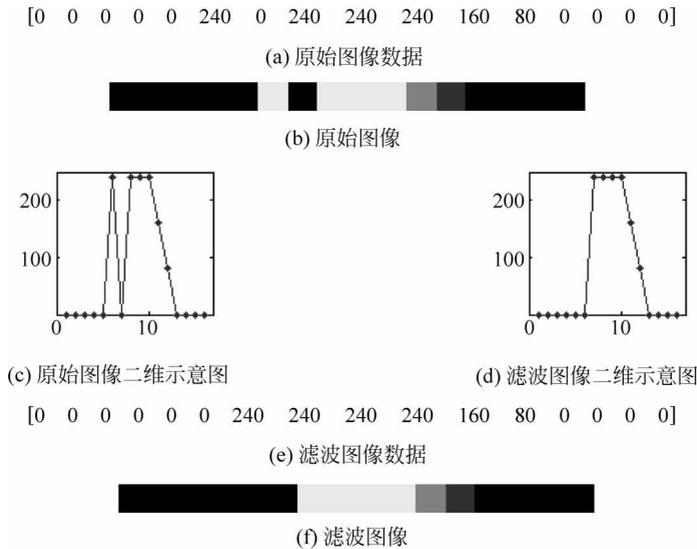


图 5-16 一维中值滤波边缘偏移示例

**解：**基于图 5-16(a) 所示的原始图像数据，得到对应的原始图像及其二维示意图如图 5-16(b)、图 5-16(c) 所示，图中同时存在边缘和噪声，且边缘和噪声相邻。采用  $1 \times 3$  模板对其进行中值滤波，得到的滤波图像的二维示意图、图像数据和图像分别如图 5-16(d)~图 5-16(f) 所示。从结果可以看出，噪声被完全消除，但边缘发生了偏移。

```
% F5_16.m

I = [0 0 0 0 0 240 0 240 240 240 160 80 0 0 0 0];
uint8I = uint8(I);
subplot(2,2,1), imshow(uint8I), xlabel('(b) 原始图像');
subplot(2,2,2), plot(I, '-'), axis([0 17 0 250]), xlabel('(c) 原始图像二维示意图');

J = medfilt2(uint8I, [1,3]);
subplot(2,2,3), plot(J, '-'), axis([0 17 0 250]), xlabel('(d) 滤波图像');

uint8J = uint8(J);
subplot(2,2,4), imshow(uint8J), xlabel('(f) 滤波图像二维示意图');
```

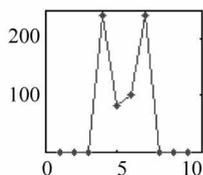
中值滤波的结果与滤波器尺寸和信号长度密切相关。若滤波器半径为  $r$ ，则能够被中值滤波器完全滤去的信号的最大长度依赖于滤波器的长度  $M=2r+1$ 。若  $l \leq r$ ，则输出全部为 0，信号被完全滤除；若  $l > r$ ，则信号按中值滤波方式发生变化。

**例 5.12** 一维中值滤波原理示例。一维 8 比特噪声图像的数据如图 5-17(a) 所示，试采用  $1 \times 3$  模板、 $1 \times 5$  模板、 $1 \times 7$  模板和  $1 \times 9$  模板分别对其进行中值滤波。

**解：**基于图 5-17(a) 所示的噪声图像数据，得到对应的噪声图像及其二维示意图如图 5-17(b)、图 5-17(c) 所示，噪声信号的长度  $l=4$ 。采用  $1 \times 3$  模板(半径  $r=1$ )、 $1 \times 5$  模板(半径  $r=2$ )、 $1 \times 7$  模板(半径  $r=3$ )和  $1 \times 9$  模板(半径  $r=4$ )分别对噪声图像进行中值滤波，得到各自滤波图像的二维示意图、图像数据和和图像分别如图 5-17(d)~图 5-17(f)、图 5-17(g)~图 5-17(i)、图 5-17(j)~图 5-17(l)和图 5-17(m)~图 5-17(o) 所示。从结果可以看出，当噪声信号长度  $l$  大于模板半径  $r$  时，滤波结果按中值滤波方式进行，如图 5-17(d)~图 5-17(l) 所示；当噪声信号长度  $l$  等于模板半径  $r$  时，滤波结果为全 0，噪声信号被完全滤除，如图 5-17(m)~图 5-17(o) 所示。

[0 0 0 240 80 100 240 0 0 0]

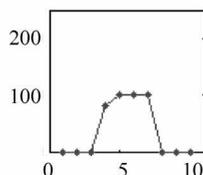
(a) 噪声图像数据( $l=4$ )



(c) 噪声图像二维示意图( $l=4$ )



(b) 噪声图像( $l=r$ )



(d) 滤波图像二维示意图( $r=1$ )

[0 0 0 80 100 100 100 0 0 0]

(e) 滤波图像数据( $r=1$ )



(f) 滤波图像( $r=1$ )

图 5-17 一维中值滤波原理示例

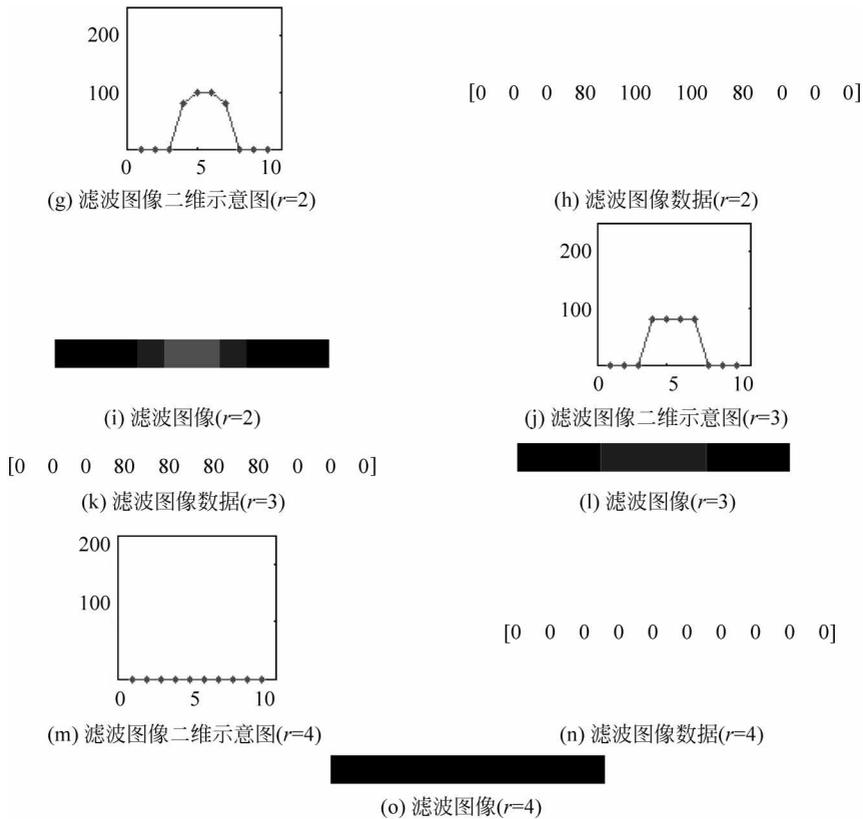


图 5-17 (续)

```

% F5_17.m

I1 = [0 0 0 240 80 100 240 0 0 0]
uint8I1 = uint8(I1);
subplot(3,4,1), imshow(uint8I1), xlabel('(b) 噪声图像(\itl = 4)');
subplot(3,4,2), plot(I1, '-'), axis([0 11 0 250]), xlabel('(c) 噪声图像二维示意图(\itl = 4)');

J1 = medfilt2(uint8I1,[1,3])
subplot(3,4,3), plot(J1, '-'), axis([0 11 0 250]), xlabel('(d) 滤波图像二维示意图(\itr = 1)');

uint8J1 = uint8(J1);
subplot(3,4,4), imshow(uint8J1), xlabel('(f) 滤波图像(\itr = 1)');

J2 = medfilt2(uint8I1,[1,5])
subplot(3,4,5), plot(J2, '-'), axis([0 11 0 250]), xlabel('(g) 滤波图像二维示意图(\itr = 2)');

uint8J2 = uint8(J2);
subplot(3,4,6), imshow(uint8J2), xlabel('(i) 滤波图像(\itr = 2)');

J3 = medfilt2(uint8I1,[1,7])
subplot(3,4,7), plot(J3, '-'), axis([0 11 0 250]), xlabel('(j) 滤波图像二维示意图(\itr = 3)');

uint8J3 = uint8(J3);
subplot(3,4,8), imshow(uint8J3), xlabel('(l) 滤波图像(\itr = 3)');

```

```
J4 = medfilt2(uint8I1,[1,9])
subplot(3,4,9),plot(J4,'.-'),axis([0 11 0 250]),xlabel('(m) 滤波图像二维示意图(\itr = 4)');

uint8J4 = uint8(J4);
subplot(3,4,10),imshow(uint8J4),xlabel('(o) 滤波图像(\itr = 4)');
```

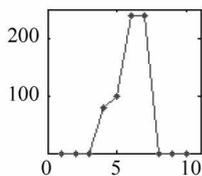
根信号是指不受中值滤波器影响的信号。若信号序列 $\{f_i\} = \{f_1, f_2, \dots, f_{2r+1}\}$ 排序后的结果依次为 $\{f_{(1)}, f_{(2)}, \dots, f_{(2r+1)}\}$ ,则称 $f_{(i)}$ 为信号序列的第 $i$ 阶统计。中值滤波的结果就是图像信号序列的第 $i+1$ 阶统计。例如,若图像信号序列的中值滤波模板尺寸为5,则中值就是第3阶统计,即信号序列排序后的第3个元素。一个信号是尺寸为 $M=2r+1$ 的中值滤波器的根信号的充分条件是:信号局部单调且为 $2r+1$ 阶,即信号的每个长度为 $2r+1$ 的段均为单调。

**例 5.13** 一维中值滤波根信号示例。一维 8 比特噪声图像的数据如图 5-18(a)所示,试采用 $1 \times 3$ 模板、 $1 \times 5$ 模板、 $1 \times 7$ 模板和 $1 \times 9$ 模板分别对其进行中值滤波。

**解:** 基于图 5-18(a)所示的噪声图像数据,得到对应的噪声图像及其二维示意图如图 5-18(b)、图 5-18(c)所示,噪声信号的长度 $l=4$ 且单调递增。采用 $1 \times 3$ 模板(半径 $r=1$ )、 $1 \times 5$ 模板(半径 $r=2$ )、 $1 \times 7$ 模板(半径 $r=3$ )和 $1 \times 9$ 模板(半径 $r=4$ )分别对噪声图像进行中值滤波,得到各自滤波图像的二维示意图、图像数据和图像分别如图 5-18(d)~图 5-18(f)、图 5-18(g)~图 5-18(i)、图 5-18(j)~图 5-18(l)和图 5-18(m)~图 5-18(o)所示。从结果可以看出,噪声信号局部单调且为 $3(2r+1=2 \times 1+1)$ 阶,即信号的每个长度为 3 的段均为单调,因此,该信号是尺寸为 $3(M=2r+1=2 \times 1+1)$ 的中值滤波器的根信号,如图 5-18(d)~图 5-18(f)所示;噪声信号不为 $5(2r+1=2 \times 2+1)$ 阶、 $7(2r+1=2 \times 3+1)$ 阶和 $9(2r+1=2 \times 4+1)$ 阶,因此,该信号不是尺寸为 $5(M=2r+1=2 \times 2+1)$ 、 $7(M=2r+1=2 \times 3+1)$ 和 $9(M=2r+1=2 \times 4+1)$ 的中值滤波器的根信号,如图 5-18(g)~图 5-18(o)所示。

[0 0 0 80 100 240 240 0 0 0]

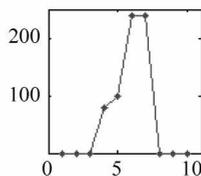
(a) 噪声图像数据( $l=4$ )



(c) 噪声图像二维示意图( $l=4$ )



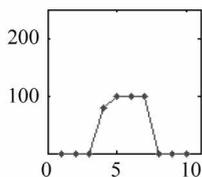
(b) 噪声图像( $l=4$ )



(d) 滤波图像二维示意图( $r=1$ )

[0 0 0 80 100 240 240 0 0 0]

(e) 滤波图像数据( $r=1$ )



(g) 滤波图像二维示意图( $r=2$ )



(f) 滤波图像( $r=1$ )

[0 0 0 80 100 100 100 0 0 0]

(h) 滤波图像数据( $r=2$ )

图 5-18 一维中值滤波根信号示例

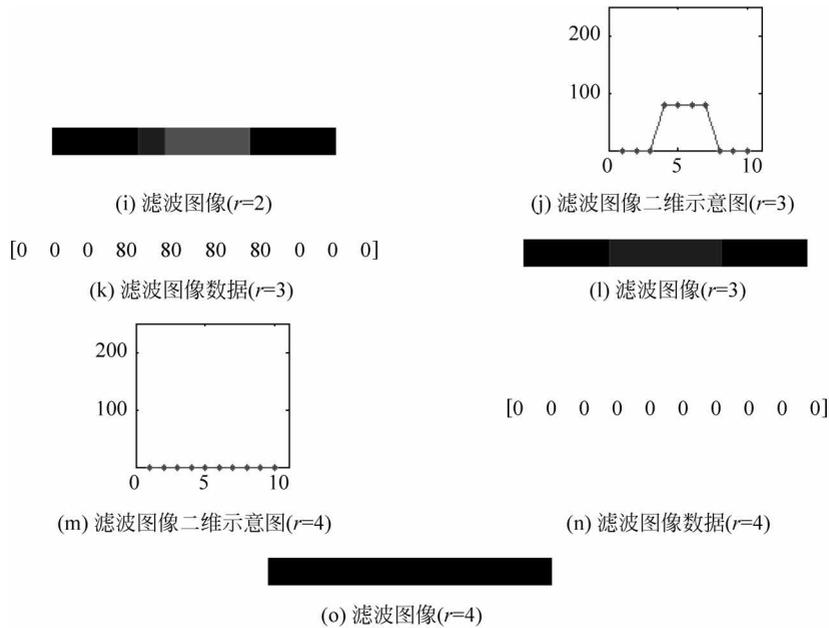


图 5-18 (续)

```

% F5_18.m

I1 = [0 0 0 80 100 240 240 0 0 0]
uint8I1 = uint8(I1);
subplot(3,4,1), imshow(uint8I1), xlabel('(b) 噪声图像(\itl = 4)');
subplot(3,4,2), plot(I1, '-'), axis([0 11 0 250]), xlabel('(c) 噪声图像二维示意图(\itl = 4)');

J1 = medfilt2(uint8I1,[1,3])
subplot(3,4,3), plot(J1, '-'), axis([0 11 0 250]), xlabel('(d) 滤波图像二维示意图(\itr = 1)');

uint8J1 = uint8(J1);
subplot(3,4,4), imshow(uint8J1), xlabel('(f) 滤波图像(\itr = 1)');

J2 = medfilt2(uint8I1,[1,5])
subplot(3,4,5), plot(J2, '-'), axis([0 11 0 250]), xlabel('(g) 滤波图像二维示意图(\itr = 2)');

uint8J2 = uint8(J2);
subplot(3,4,6), imshow(uint8J2), xlabel('(i) 滤波图像(\itr = 2)');

J3 = medfilt2(uint8I1,[1,7])
subplot(3,4,7), plot(J3, '-'), axis([0 11 0 250]), xlabel('(j) 滤波图像二维示意图(\itr = 3)');

uint8J3 = uint8(J3);
subplot(3,4,8), imshow(uint8J3), xlabel('(l) 滤波图像(\itr = 3)');

J4 = medfilt2(uint8I1,[1,9])
subplot(3,4,9), plot(J4, '-'), axis([0 11 0 250]), xlabel('(m) 滤波图像二维示意图(\itr = 4)');

uint8J4 = uint8(J4);
subplot(3,4,10), imshow(uint8J4), xlabel('(o) 滤波图像(\itr = 4)');

```

### 5.3.1.2 二维中值滤波

设：原始图像为  $f(x, y)$ , 变换图像为  $g(x, y)$ , 二维中值滤波的基本原理如下式所示：

$$g_{\text{med}}(x, y) = \text{median}_{(i, j) \in S}(f(i, j)) \quad (5-19)$$

式中,  $S$  为像素  $(x, y)$  的  $m \times n$  邻域;  $\text{median}()$  为取中值函数, 即对模板覆盖的像素按灰度值大小排序, 并取排序后处在中间位置处的值。由此可知, 模板覆盖的像素灰度值有一半大于  $g(x, y)$ , 另一半小于  $g(x, y)$ 。

二维中值滤波的步骤为：

- (1) 将模板在原始图像中漫游, 并将模板中心与原始图像中的某个像素重合;
- (2) 读取模板下对应像素的灰度值, 并按升序或降序排序;
- (3) 读取排序后处在中间位置的值;
- (4) 将该中间值赋值给变换图像中对应模板中心位置的像素。

**注意：**模板尺寸通常选择奇数, 以确保唯一的中间值; 一般地, 首先选择  $3 \times 3$ , 再选择  $5 \times 5$ , 并逐渐增大, 直到滤波效果满意为止。

**例 5.14** 二维中值滤波原理示例。

**解：**原始图像如图 5-19(a) 所示, 原始图像添加噪声密度 (即包括噪声值的图像区域的百分比) 为 2% 的椒盐噪声后得到的噪声图像如图 5-19(b) 所示。采用  $3 \times 3$  模板、 $5 \times 5$  模板、 $7 \times 7$  模板和  $9 \times 9$  模板对噪声图像进行二维中值滤波的结果如图 5-19(c) ~ 图 5-19(f) 所示。从结果可以看出,  $3 \times 3$  模板在几乎消除所有噪声的同时保留了图像细节, 效果最好; 模板尺寸越大, 噪声仍然被消除, 但图像细节消除也越强。

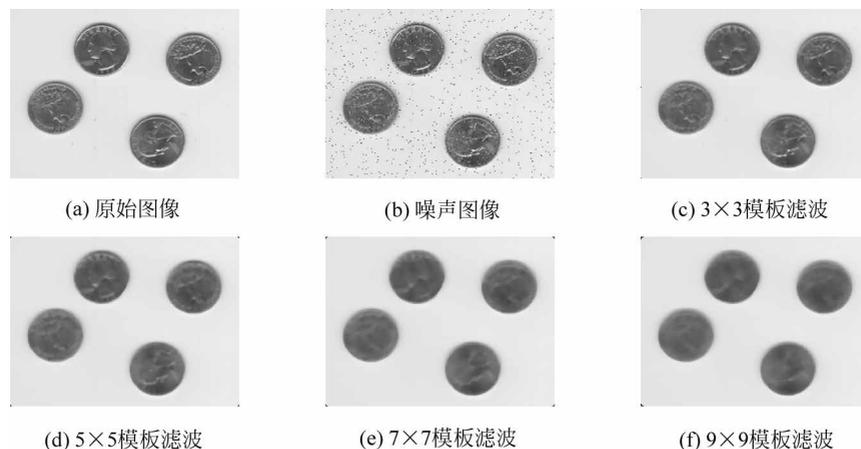


图 5-19 二维中值滤波原理示例

```
% F5_19.m

I = imread('eight.tif');
subplot(2,3,1), imshow(I), xlabel('(a) 原始图像');

J = imnoise(I, 'salt & pepper', 0.02);
subplot(2,3,2), imshow(J), xlabel('(b) 噪声图像');
```

```

K1 = medfilt2(J,[3,3]);
subplot(2,3,3), imshow(uint8(K1)), xlabel('(c) 3 * 3 模板滤波');

K2 = medfilt2(J,[5,5]);
subplot(2,3,4), imshow(uint8(K2)), xlabel('(d) 5 * 5 模板滤波');

K3 = medfilt2(J,[7,7]);
subplot(2,3,5), imshow(uint8(K3)), xlabel('(e) 7 * 7 模板滤波');

K4 = medfilt2(J,[9,9]);
subplot(2,3,6), imshow(uint8(K4)), xlabel('(f) 9 * 9 模板滤波');

```

二维中值滤波的去噪效果主要依赖两个因素：

(1) 模板的空间形状。一般情况下，图像中面积小于模板面积一半的过亮或过暗区域将会被中值滤波器滤除，面积较大的物体几乎会原样保存下来。

(2) 模板中像素的个数及其分布。典型的中值滤波器模板如图 5-20 所示，其中，图 5-20(a) 为对角邻域模板，共 9 个像素；图 5-20(b) 为 4 邻域模板，共 9 个像素；图 5-20(c) 为 16 邻域模板，共 9 个像素；图 5-20(d) 为 4 邻域模板和 16 邻域模板的融合，共 13 个像素；图 5-20(e) 为城区距离模板，共 13 个像素，像素与中心像素的城区距离  $D_4$  小于等于 2；图 5-20(f) 为欧氏距离模板，共 13 个像素，像素与中心像素的欧氏距离  $D_E$  为 2~2.5。

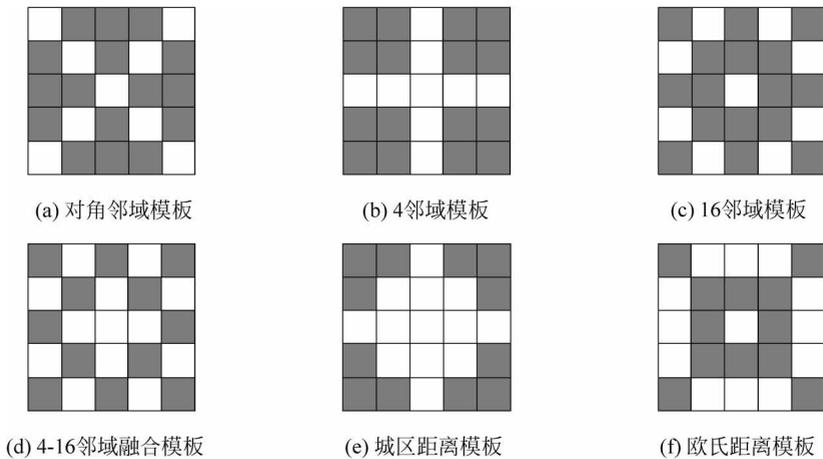


图 5-20 典型中值滤波器模板

模板中像素的分布对滤波结果影响很大。X 形模板仅保留对角线。十字叉模板保留细的水平线和垂直线，但会滤除对角线，由于水平线和垂直线在人类视觉中起重要作用，因此该模板效果较好。方形模板对图像细节最不敏感，会滤除细线并消除边缘上的角点，同时产生令人讨厌的条纹(常数灰度的区域)，这是中值滤波器的缺点。

**例 5.15** 二维中值滤波模板示例。

**解：**原始图像如图 5-21(a) 所示，原始图像添加噪声密度(即包括噪声值的图像区域的百分比)为 3% 的椒盐噪声后得到的噪声图像如图 5-21(b) 所示。采用对角邻域模板、4 邻域模板、16 邻域模板、4-16 邻域融合模板、城区距离模板和欧氏距离模板对噪声图像进行二

维中值滤波的结果如图 5-21(c)~图 5-21(h)所示。从结果可以看出,模板像素个数为 13 的模板(4-16 邻域融合模板、城区距离模板、欧氏距离模板)的去噪效果明显优于模板像素个数为 9 的模板(对角邻域模板、4 邻域模板、16 邻域模板)。

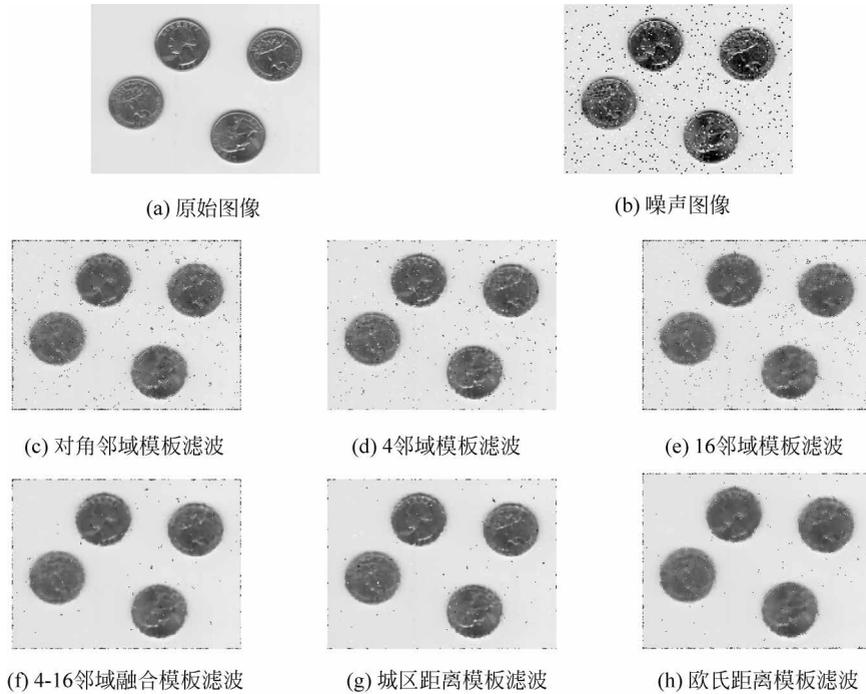


图 5-21 二维中值滤波模板示例

```
% F5_21.m

I = imread('eight.tif');
subplot(3,5,1), imshow(I), xlabel('(a) 原始图像');

J = imnoise(I, 'salt & pepper', 0.3);
subplot(3,5,2), imshow(J), xlabel('(b) 噪声图像');

mask1 = [1 0 0 0 1
         0 1 0 1 0
         0 0 1 0 0
         0 1 0 1 0
         1 0 0 0 1];
subplot(3,5,3), imshow(mask1), xlabel('(c-1) 对角邻域模板');

K1 = ordfilt2(J, 5, mask1);
uint8K1 = uint8(K1);
subplot(3,5,4), imshow(uint8K1), xlabel('(c-2) 对角邻域模板滤波');
```

```
mask2 = [0 0 1 0 0
         0 0 1 0 0
         1 1 1 1 1
         0 0 1 0 0
         0 0 1 0 0];
subplot(3,5,5), imshow(mask2), xlabel('(d-1) 4 邻域模板');

K2 = ordfilt2(J,5,mask2);
uint8K2 = uint8(K2);
subplot(3,5,6), imshow(uint8K2), xlabel('(d-2) 4 邻域模板滤波');

mask3 = [0 1 0 1 0
         1 0 0 0 1
         0 0 1 0 0
         1 0 0 0 1
         0 1 0 1 0];
subplot(3,5,7), imshow(mask3), xlabel('(e-1) 16 邻域模板');

K3 = ordfilt2(J,5,mask3);
uint8K3 = uint8(K3);
subplot(3,5,8), imshow(uint8K3), xlabel('(e-2) 16 邻域模板滤波');

mask4 = [0 1 0 1 0
         1 0 1 0 1
         0 1 1 1 0
         1 0 1 0 1
         0 1 0 1 0];
subplot(3,5,9), imshow(mask4), xlabel('(f-1) 4-16 邻域融合模板');

K4 = ordfilt2(J,7,mask4);
uint8K4 = uint8(K4);
subplot(3,5,10), imshow(uint8K4), xlabel('(f-2) 4-16 邻域融合模板滤波');

mask5 = [0 0 1 0 0
         0 1 1 1 0
         1 1 1 1 1
         0 1 1 1 0
         0 0 1 0 0];
subplot(3,5,11), imshow(mask5), xlabel('(g-1) 城区距离模板');

K5 = ordfilt2(J,7,mask5);
uint8K5 = uint8(K5);
subplot(3,5,12), imshow(uint8K5), xlabel('(g-2) 城区距离模板滤波');

mask6 = [0 1 1 1 0
         1 0 0 0 1
         1 0 1 0 1
         1 0 0 0 1
         0 1 1 1 0];
```

```
subplot(3,5,13),imshow(mask6),xlabel('(h-1) 欧氏距离模板');

K6 = ordfilt2(J,7,mask6);
uint8K6 = uint8(K6);
subplot(3,5,14),imshow(uint8K6),xlabel('(h-2) 欧氏距离模板滤波');
```

### 5.3.2 序统计滤波

序统计滤波是指滤波原理基于模板所覆盖像素的灰度值排序的滤波方法。中值滤波即属于序统计滤波。常用的序统计滤波方法包括：

(1) 中值滤波。指信号序列中某点的值用该点邻域中有序序列位于 50% 位置处的值代替的方法。

(2) 最大值滤波。指信号序列中某点的值用该点邻域中升序序列位于 100% 位置处的值代替的方法，常用来检测图像中的最亮点并可削弱低取值的椒盐噪声。设：原始图像为  $f(x, y)$ ，变换图像为  $g(x, y)$ ，最大值滤波的基本原理如下式所示：

$$g_{\max}(x, y) = \max_{(i, j) \in S} (f(i, j)) \quad (5-20)$$

式中， $S$  为像素  $(x, y)$  的  $m \times n$  邻域； $\max()$  为取最大值函数，即对模板覆盖的像素按灰度值由小到大排序，并取排序后处在最大位置处的值。

(3) 最小值滤波。指信号序列中某点的值用该点邻域中升序序列位于 0% 位置处的值代替的方法，常用来检测图像中的最暗点并可削弱高取值的椒盐噪声。设：原始图像为  $f(x, y)$ ，变换图像为  $g(x, y)$ ，最小值滤波的基本原理如下式所示：

$$g_{\min}(x, y) = \min_{(i, j) \in S} (f(i, j)) \quad (5-21)$$

式中， $S$  为像素  $(x, y)$  的  $m \times n$  邻域； $\min()$  为取最小值函数，即对模板覆盖的像素按灰度值由小到大排序，并取排序后处在最小位置处的值。

(4) 中点滤波。指信号序列中某点的值用该点邻域中有序序列的最大值和最小值的均值代替的方法，该方法结合了排序滤波器和均值滤波器，对多种随机分布的噪声（如高斯噪声、均匀噪声）都比较有效。设：原始图像为  $f(x, y)$ ，变换图像为  $g(x, y)$ ，中点滤波的基本原理如下式所示：

$$g_{\text{mid}}(x, y) = \frac{1}{2} (\max_{(i, j) \in S} (f(i, j)) + \min_{(i, j) \in S} (f(i, j))) = \frac{1}{2} (g_{\max}(x, y) + g_{\min}(x, y)) \quad (5-22)$$

式中， $S$  为像素  $(x, y)$  的  $m \times n$  邻域； $\max()$  为取最大值函数； $\min()$  为取最小值函数。

#### 例 5.16 序统计滤波原理示例。

**解：**一张 8 比特灰度图像的矩阵数据如图 5-22(a) 所示，对应的原始图像如图 5-22(b) 所示。滤波模板数据如图 5-22(c) 所示，滤波模板图像如图 5-22(d) 所示，采用该模板对原始图像进行最小值滤波、第 2 小值滤波、第 3 小值滤波和最大值滤波，得到各自的滤波数据和滤波图像分别如图 5-22(e)、图 5-22(f)、图 5-22(g)、图 5-22(h)，图 5-22(i)、图 5-22(j) 和图 5-22(k)、图 5-22(l) 所示。从结果可以看出，就滤波数据而言，均按滤波方法原理发生变化；就滤波图像而言，最小值滤波将原始图像整体变暗，最大值滤波将原始图像整体变亮，

第 2 小值滤波和第 3 小值滤波的效果介于两者之间。

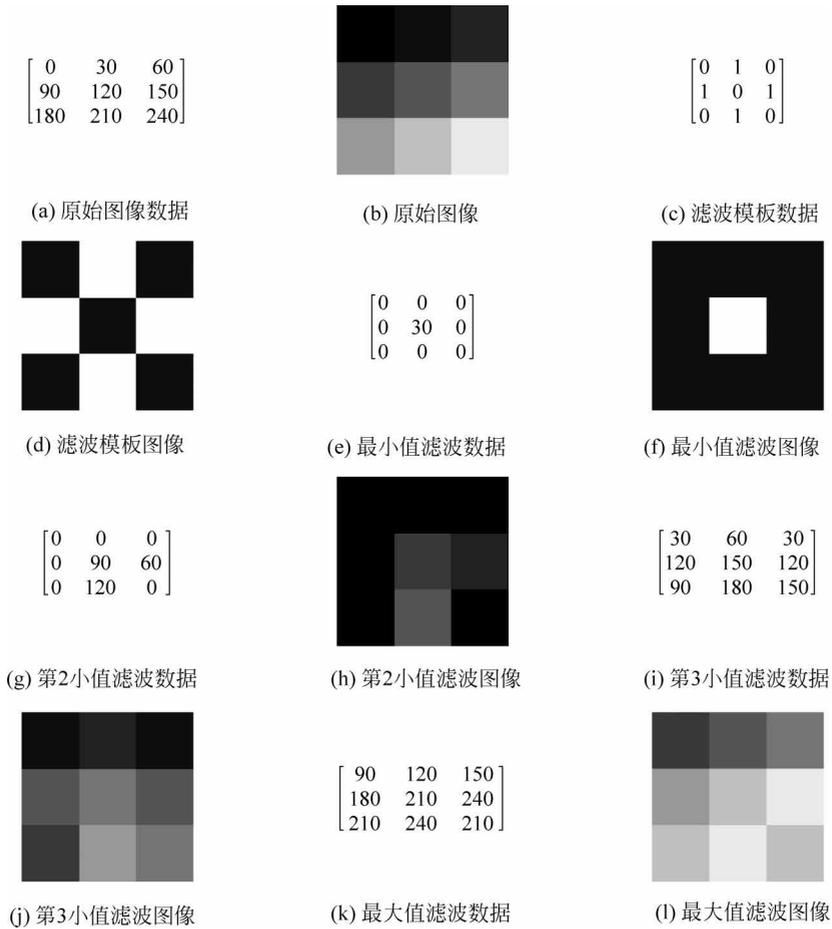


图 5-22 序统计滤波原理示例

```
% F5_22.m

I = [ 0  30  60
      90 120 150
      180 210 240];
uint8I = uint8(I)
subplot(2,3,1), imshow(uint8I), xlabel('(a) 原始图像');

mask = [0  1  0
        1  0  1
        0  1  0];
uint8mask = uint8(mask)
subplot(2,3,2), imshow(~uint8mask), xlabel('(b) 模板图像');

K1 = ordfilt2(I,1,mask);
uint8K1 = uint8(K1)
```

```
subplot(2,3,3),imshow(uint8K1),xlabel('(c) 最小值滤波图像');

K2 = ordfilt2(I,2,mask);
uint8K2 = uint8(K2)
subplot(2,3,4),imshow(uint8K2),xlabel('(d) 第 2 小值滤波图像');

K3 = ordfilt2(I,3,mask);
uint8K3 = uint8(K3)
subplot(2,3,5),imshow(uint8K3),xlabel('(e) 第 3 小值滤波图像');

K4 = ordfilt2(I,4,mask);
uint8K4 = uint8(K4)
subplot(2,3,6),imshow(uint8K4),xlabel('(f) 最大值滤波图像');
```

### 例 5.17 序统计滤波示例。

**解：**原始图像如图 5-23(a)所示,原始图像添加噪声密度(即包括噪声值的图像区域的百分比)为 10%的椒盐噪声后得到的噪声图像如图 5-23(b)所示,滤波模板数据如图 5-23(c)所示,采用该模板对噪声图像进行中值滤波、最小值滤波和最大值滤波,得到的滤波图像分别如图 5-23(d)~图 5-23(f)所示。从结果可以看出,中值滤波的去噪效果最好;最小值滤波不能消除噪声,且将图像整体变暗;最大值滤波不能消除噪声,且将图像整体变亮。

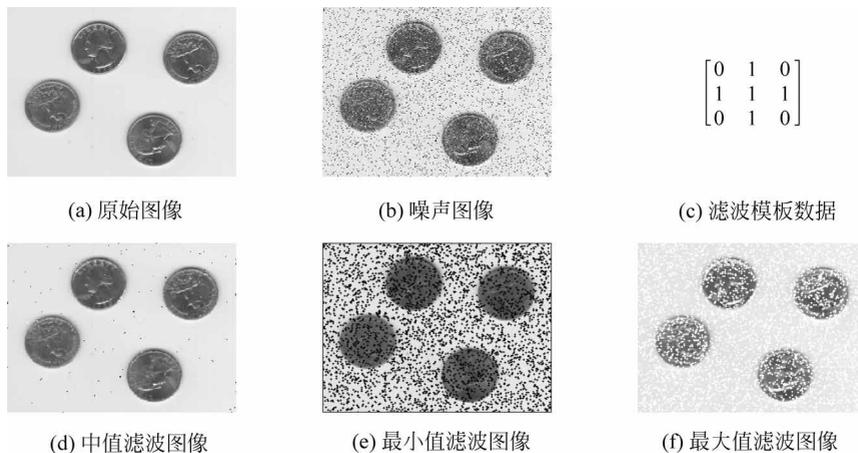


图 5-23 序统计滤波示例

```
% F5_23.m

I = imread('eight.tif');
subplot(2,3,1),imshow(I),xlabel('(a) 原始图像');

J = imnoise(I,'salt & pepper',0.1);
subplot(2,3,2),imshow(J),xlabel('(b) 椒盐噪声图像');
```

```

mask = [0  1  0
        1  1  1
        0  1  0];
subplot(2,3,3), imshow(mask), xlabel('(c) 滤波模板');

K1 = ordfilt2(J,3,mask);
uint8K1 = uint8(K1);
subplot(2,3,4), imshow(uint8K1), xlabel('(d) 中值滤波图像');

K2 = ordfilt2(J,1,mask);
uint8K2 = uint8(K2);
subplot(2,3,5), imshow(uint8K2), xlabel('(e) 最小值滤波图像');

K3 = ordfilt2(J,5,mask);
uint8K3 = uint8(K3);
subplot(2,3,6), imshow(uint8K3), xlabel('(f) 最大值滤波图像');

```

## 5.4 线性锐化滤波

图像锐化的目的是突出图像的边缘信息,加强图像的轮廓特征,常用于物体的边缘检测。从数学的观点来看,图像锐化可通过微分方法实现,以检查图像区域内灰度的变化;图像平滑是积分运算的结果,积分是微分的逆运算,积分将图像灰度值平均,使图像变得模糊;一阶微分和二阶微分必须保证:

- (1) 平坦段(灰度值恒定)微分值为零;
- (2) 灰度阶梯或斜坡的起始点处微分值非零;
- (3) 沿着斜坡面微分值非零。

数字图像是离散数据,在离散空间,微分用差分来实现。一元函数  $f(x)$  的一阶微分的差分定义如式(5-23)所示,一元函数  $f(x)$  的二阶微分的差分定义如式(5-24)所示。

$$\nabla f = \frac{\partial f}{\partial x} = \frac{df}{dx} = \lim_{\Delta x \rightarrow 0} \frac{\Delta f}{\Delta x} = \frac{f(x+1) - f(x)}{(x+1) - x} = f(x+1) - f(x) \quad (5-23)$$

$$\begin{aligned} \nabla^2 f &= \frac{\partial^2 f}{\partial x^2} = \frac{\partial \left( \frac{\partial f}{\partial x} \right)}{\partial x} = \frac{[f(x+1) - f(x)] - [f(x-1+1) - f(x-1)]}{x - (x-1)} \\ &= f(x+1) + f(x-1) - 2f(x) \end{aligned} \quad (5-24)$$

**例 5.18** 微分方法示例。一维图像的二维示意图如图 5-24(a)所示,试求图像的一阶微分和二阶微分结果。

**解:** 基于图 5-24(a)所示的一维图像的二维示意图,得到的图像数据及其一阶微分和二阶微分的结果如图 5-24(b)所示。从结果可以看出微分方法的如下特点。一般来说,二阶微分比一阶微分效果好。

- (1) 一阶微分通常会产生较宽的边缘,二阶微分则细得多。沿着整个灰度值斜坡,一阶

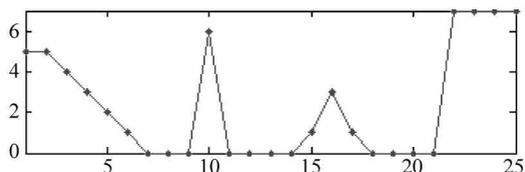
微分值都不为零,经二阶微分后,非零值仅出现在斜坡的起始处和终点处,图像边缘类似这种过渡。

(2) 二阶微分对细节有较强的响应,如细线和孤立点。细线可看作细节。在孤立点及其周围点上,二阶微分比一阶微分的响应要强得多。

(3) 一阶微分一般对灰度阶梯有较强的响应。

(4) 二阶微分一般对灰度阶梯变化产生双响应(双边缘效果)。二阶微分值有一个从正到负的过渡,该现象表现为双线。

(5) 二阶微分在灰度值变化相似时,对点的响应比对线强,对线的响应比对阶梯强。



(a) 一维图像的二维示意图

图像数据	[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]
一阶微分	0	-1	-1	-1	-1	-1	0	0	6	-6	0	0	0	1	2	-2	-1	0	0	0	7	0	0	0	0
二阶微分																									
		-1	0	0	0	0	1	0	6	-12	6	0	0	1	1	-4	1	1	0	0	7	-7	0	0	0

(b) 图像数据、一阶微分、二阶微分结果

图 5-24 微分方法示例

```
% F5_24a.m
```

```
I1 = [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]
```

```
uint8I1 = uint8(I1);
```

```
subplot(1,2,1), imshow(uint8I1), xlabel('(a) 噪声图像(l = 4)');
```

```
subplot(1,2,2), plot(I1, '-'), axis([1 25 0 7]), xlabel('(b) 噪声图像的灰度值序列');
```

Laplacian(拉普拉斯)算子是基于二阶微分的线性锐化增强方法。二元函数  $f(x, y)$  的拉普拉斯变换定义为如求 5-25 所示。

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (5-25)$$

水平( $y$ )方向二阶偏微分的离散形式定义为如式(5-26)所示,垂直( $x$ )方向二阶偏微分的离散形式定义为如式(5-27)所示。因此,水平垂直方向的拉普拉斯算子的离散形式(线性地)定义为如式(5-28)所示。

$$\frac{\partial^2 f}{\partial y^2} = f(x, y + 1) + f(x, y - 1) - 2f(x, y) \quad (5-26)$$

$$\frac{\partial^2 f}{\partial x^2} = f(x + 1, y) + f(x - 1, y) - 2f(x, y) \quad (5-27)$$

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y) \quad (5-28)$$

对角方向左上右下的二阶偏微分的离散形式定义为如式(5-29)所示, 对角方向左下右上的二阶偏微分的离散形式定义为如式(5-30)所示。因此, 水平垂直对角方向的拉普拉斯算子的离散形式(线性地)定义为如式(5-31)所示。

$$f(x+1, y+1) + f(x-1, y-1) - 2f(x, y) \quad (5-29)$$

$$f(x+1, y-1) + f(x-1, y+1) - 2f(x, y) \quad (5-30)$$

$$\nabla^2 f = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) + f(x+1, y+1) + f(x-1, y-1) + f(x+1, y-1) + f(x-1, y+1) - 8f(x, y) \quad (5-31)$$

拉普拉斯算子的模板如图 5-25 所示, 模板的共同特点是系数之和为 0。其中, 图 5-25(a) 为水平垂直模板, 模板中心系数为负且系数之和为 0, 具有  $90^\circ$  旋转的各向同性结果; 图 5-25(b) 为图 5-25(a) 的变形, 仍为水平垂直模板, 模板中心系数为正且系数之和保持为 0; 图 5-25(c) 为水平垂直对角模板, 模板中心系数为负且系数之和为 0, 具有  $45^\circ$  旋转的各向同性结果; 图 5-25(d) 为图 5-25(c) 的变形, 仍为水平垂直对角模板, 模板中心系数为正且系数之和保持为 0。

$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$
(a) 负中心系数的 水平垂直模板	(b) 正中心系数的 水平垂直模板	(c) 负中心系数的 水平垂直对角模板	(d) 正中心系数的 水平垂直对角模板

图 5-25 拉普拉斯算子模板

拉普拉斯算子强调图像中灰度的突变以及降低灰度慢变化区域的亮度, 结果是产生一幅把边线和突变点叠加到暗背景中的图像。因此, 通常将原始图像和拉普拉斯图像叠加在一起, 既保持了拉普拉斯锐化处理的效果, 又恢复了原始图像的背景信息。图像叠加方法如下式所示:

$$g(x, y) = \begin{cases} f(x, y) - \nabla^2 f(x, y), & \text{模板中心系数为负} \\ f(x, y) + \nabla^2 f(x, y), & \text{模板中心系数为正} \end{cases} \quad (5-32)$$

#### 例 5.19 拉普拉斯算子示例。

**解:** 原始图像如图 5-26(a) 所示, 负中心系数和正中心系数的水平垂直模板滤波图像分别如图 5-26(b) 和图 5-26(d) 所示, 负中心系数和正中心系数的水平垂直对角模板滤波图像分别如图 5-26(f) 和图 5-26(h) 所示。从结果可以看出, 拉普拉斯滤波得到暗背景下的边缘检测结果, 同时, 水平垂直对角模板滤波图像的边缘检测效果优于水平垂直模板滤波图像。为了恢复原始图像的背景信息, 同时保持锐化处理的结果, 将原始图像和滤波图像叠加, 得到负中心系数和正中心系数的水平垂直模板叠加图像分别如图 5-26(c) 和图 5-26(e) 所示, 得到负中心系数和正中心系数的水平垂直对角模板叠加图像分别如图 5-26(g) 和图 5-26(i) 所示。从结果可以看出, 滤波图像既保持了锐化处理的结果, 又恢复了原始图像的背景信息, 具有良好的视觉效果。



图 5-26 拉普拉斯算子示例

```

% F5_26.m

I = imread('lena.bmp');
subplot(3,3,1), imshow(I), xlabel('(a) 原始图像');

mask1 = fspecial('laplacian',0)
LB1 = filter2(mask1,I);
subplot(3,3,2), imshow(uint8(LB1)), text(-14,285,'(b) 负中心系数的水平垂直模板滤波图像');
subplot(3,3,3), imshow(uint8(double(I) - LB1)), xlabel('(c) 负中心系数的水平垂直模板减法叠加图像');

mask2 = - mask1
LB2 = filter2(mask2,I);
subplot(3,3,4), imshow(uint8(LB2)), text(-14,285,'(d) 正中心系数的水平垂直模板滤波图像');
subplot(3,3,5), imshow(uint8(double(I) + LB2)), xlabel('(e) 正中心系数的水平垂直模板加法叠加图像');

```

```

mask3 = [1   1   1
         1  -8   1
         1   1   1]
LB3 = filter2(mask3,I);
subplot(3,3,6), imshow(uint8(LB3)), text(-28,285,'(f) 负中心系数的水平垂直对角模板滤波图像');
subplot(3,3,7), imshow(uint8(double(I) - LB3)), xlabel('(g) 负中心系数的水平垂直对角模板减法叠加图像');

mask4 = - mask3
LB4 = filter2(mask4,I);
subplot(3,3,8), imshow(uint8(LB4)), text(-28,285,'(h) 正中心系数的水平垂直对角模板滤波图像');
subplot(3,3,9), imshow(uint8(double(I) + LB4)), xlabel('(i) 正中心系数的水平垂直对角模板加法叠加图像');

```

## 5.5 非线性锐化滤波

非线性锐化滤波技术基于非线性的微分方法。常用的非线性锐化滤波方法包括梯度法、Prewitt算子、Sobel算子、Log算子、高通滤波、掩模法、Canny算子、Krisch算子等。

### 5.5.1 梯度法

梯度法是基于—阶微分的非线性锐化滤波方法。二元函数  $f(x, y)$  在坐标  $(x, y)$  的梯度定义为一个二维列向量：

$$\text{grad}(f(x, y)) = \nabla f = \left( \frac{\partial f}{\partial x} \quad \frac{\partial f}{\partial y} \right)^T = (G_x \quad G_y)^T = \begin{pmatrix} G_x \\ G_y \end{pmatrix} \quad (5-33)$$

式中,  $\text{grad}(x)$  为梯度函数。

梯度是一个向量, 指向函数最大变化率的方向, 梯度的方向角定义为

$$\alpha(f(x, y)) = \arctan\left(\frac{G_y}{G_x}\right) = \arctan\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right) \quad (5-34)$$

式中,  $\alpha(x)$  为方向角函数。

梯度的模用  $G(f(x, y))$  来表示, 有三种常用的非线性的定义方式:

(1) 范数为 2 的欧氏距离定义为

$$G(f(x, y)) = |\nabla f_2| = (G_x^2 + G_y^2)^{1/2} \quad (5-35)$$

(2) 范数为 1 的城区距离定义为

$$G(f(x, y)) = |\nabla f_1| = |G_x| + |G_y| \quad (5-36)$$

(3) 范数为  $\infty$  的棋盘距离定义为

$$G(f(x, y)) = |\nabla f_\infty| = \max(|G_x|, |G_y|) \quad (5-37)$$

梯度的模的微分操作仍用差分来实现, 包括两种差分方法。

## 1) 水平垂直差分法

水平垂直差分法原理如图 5-27 所示。其中,图 5-27(a)为差分示意图,垂直方向的差分  $G_x = f(x, y) - f(x+1, y)$ ,水平方向的差分  $G_y = f(x, y) - f(x, y+1)$ ;图 5-27(b)为水平差分模板和垂直差分模板。

$$\begin{array}{ccc} \left[ \begin{array}{cc} f(x, y) & - & f(x, y+1) \\ | & & \\ f(x+1, y) & & f(x+1, y+1) \end{array} \right] & & \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 1 & 0 \\ -1 & 0 \end{bmatrix} \\ \text{(a) 差分示意图} & & \text{(b) 水平、垂直差分模板} \end{array}$$

图 5-27 水平垂直差分法原理

基于梯度模的三种定义方式,得到水平垂直差分法梯度模的三种计算方法。基于范数为 2 的欧氏距离的梯度模如式(5-38)所示,基于范数为 1 的城区距离的梯度模如式(5-39)所示,基于范数为  $\infty$  的棋盘距离的梯度模如式(5-40)所示。

$$\begin{aligned} G(f(x, y)) &= |\nabla f_2| = (G_x^2 + G_y^2)^{1/2} \\ &= ((f(x, y) - f(x+1, y))^2 + (f(x, y) - f(x, y+1))^2)^{1/2} \end{aligned} \quad (5-38)$$

$$\begin{aligned} G(f(x, y)) &= |\nabla f_1| = |G_x| + |G_y| \\ &= |f(x, y) - f(x+1, y)| + |f(x, y) - f(x, y+1)| \end{aligned} \quad (5-39)$$

$$\begin{aligned} G(f(x, y)) &= |\nabla f_\infty| = \max(|G_x|, |G_y|) \\ &= \max(|f(x, y) - f(x+1, y)|, |f(x, y) - f(x, y+1)|) \end{aligned} \quad (5-40)$$

## 2) 罗伯特(Robert)差分法

罗伯特差分法原理如图 5-28 所示。其中,图 5-28(a)为差分示意图,左上右下方向的差分  $G_x = f(x, y) - f(x+1, y+1)$ ,左下右上方向的差分  $G_y = f(x+1, y) - f(x, y+1)$ ;图 5-28(b)为水平差分模板和垂直差分模板。

$$\begin{array}{ccc} \left[ \begin{array}{cc} f(x, y) & \times & f(x, y+1) \\ & & \\ f(x+1, y) & & f(x+1, y+1) \end{array} \right] & & \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \\ \text{(a) 差分示意图} & & \text{(b) 水平、垂直差分模板} \end{array}$$

图 5-28 罗伯特差分法原理

基于梯度模的三种定义方式,得到罗伯特差分法梯度模的三种计算方法。基于范数为 2 的欧氏距离的梯度模如式(5-41)所示,基于范数为 1 的城区距离的梯度模如式(5-42)所示,基于范数为  $\infty$  的棋盘距离的梯度模如式(5-43)所示。

$$\begin{aligned} G(f(x, y)) &= |\nabla f_2| = (G_x^2 + G_y^2)^{1/2} \\ &= ((f(x, y) - f(x+1, y+1))^2 + (f(x+1, y) - f(x, y+1))^2)^{1/2} \end{aligned} \quad (5-41)$$

$$\begin{aligned} G(f(x, y)) &= |\nabla f_1| = |G_x| + |G_y| \\ &= |f(x, y) - f(x+1, y+1)| + |f(x+1, y) - f(x, y+1)| \end{aligned} \quad (5-42)$$

$$\begin{aligned} G(f(x, y)) &= |\nabla f_\infty| = \max(|G_x|, |G_y|) \\ &= \max(|f(x, y) - f(x+1, y+1)|, |f(x+1, y) - f(x, y+1)|) \end{aligned} \quad (5-43)$$

由上述公式可知,梯度的模与相邻像素的灰度差成正比。边缘区域梯度值较大,平滑区域梯度值较小,常数灰度值区域梯度值为0。注意:图像的最后一行和最后一列不能计算梯度,在实际应用中,常由倒数第2行和倒数第2列的梯度值近似代替。

梯度法的基本原理如式(5-44)所示。该方法的优点是简单,坐标 $(x, y)$ 处的值等于该点的梯度;缺点是图像中的平滑区域由于梯度值较小而成为暗区。

$$g(x, y) = G(f(x, y)) \quad (5-44)$$

式中, $f(x, y)$ 为原始图像; $g(x, y)$ 为变换图像。

梯度法的结果是产生一幅把边线和突变点叠加到暗背景中的图像。为了保持锐化处理的效果,同时恢复原始图像的背景信息,梯度法采用的计算思路如图5-29所示。

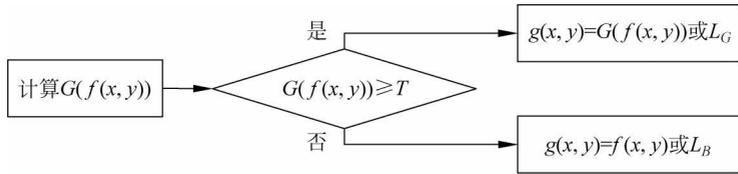


图 5-29 梯度法的计算思路

基于该计算思路,梯度的计算包括四种情况:第一种情况如式(5-45)所示,通过合理地选择阈值 $T$ ,既没有破坏平滑区域的灰度值,又能够有效地突出图像的边缘;第二种情况如式(5-46)所示,设置背景为灰度值常量 $L_B$ ,仅关注边缘灰度值的变化,不受背景的影响;第三种情况如式(5-47)所示,将明显的边缘用灰度值常量 $L_G$ 来表征;第四种情况如式(5-48)所示,给出一幅二值图像,仅关注边缘的位置。

$$g(x, y) = \begin{cases} G(f(x, y)), & G(f(x, y)) \geq T \\ f(x, y), & \text{其他} \end{cases} \quad (5-45)$$

$$g(x, y) = \begin{cases} G(f(x, y)), & G(f(x, y)) \geq T \\ L_B, & \text{其他} \end{cases} \quad (5-46)$$

$$g(x, y) = \begin{cases} L_G, & G(f(x, y)) \geq T \\ f(x, y), & \text{其他} \end{cases} \quad (5-47)$$

$$g(x, y) = \begin{cases} L_G, & G(f(x, y)) \geq T \\ L_B, & \text{其他} \end{cases} \quad (5-48)$$

### 例 5.20 梯度法示例。

**解:**原始图像如图5-30(a)所示,梯度法滤波图像如图5-30(b)所示,第一、二、三、四种情况滤波图像分别如图5-30(c)~图5-30(f)所示。从结果可以看出,梯度法滤波图像整体较暗,边缘信息叠加到暗背景中;第一种情况滤波图像在保持平滑区域灰度值的同时又突出了图像的边缘;第二种情况滤波图像保持背景灰度值为常量,仅关注边缘灰度值的变化;第三种情况滤波图像保持了平滑区域灰度值,将边缘用固定的灰度值来表征;第四种情况滤波图像是一幅二值图像,突出了边缘的位置。



(a) 原始图像



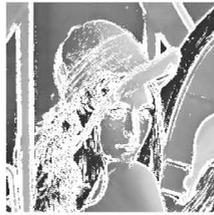
(b) 梯度法滤波图像



(c) 第一种情况滤波图像



(d) 第二种情况滤波图像



(e) 第三种情况滤波图像



(f) 第四种情况滤波图像

图 5-30 梯度法示例

```

% F5_30.m

I = imread('lena.bmp');
subplot(2,3,1), imshow(I), xlabel('(a) 原始图像');

I = double(I);
[Gx,Gy] = gradient(I);
G = sqrt(Gx.*Gx + Gy.*Gy);

J1 = G;
subplot(2,3,2), imshow(uint8(J1)), xlabel('(b) 梯度法滤波图像');

J2 = I;
K = find(G >= 7);
J2(K) = G(K);
subplot(2,3,3), imshow(uint8(J2)), xlabel('(c) 第一种情况滤波图像');

J3 = I;
K = find(G >= 7);
J3(K) = 255;
subplot(2,3,4), imshow(uint8(J3)), xlabel('(d) 第二种情况滤波图像');

J4 = G;
K = find(G <= 7);
J4(K) = 255;
subplot(2,3,5), imshow(uint8(J4)), xlabel('(e) 第三种情况滤波图像');

J5 = I;
K = find(G <= 7);

```

```
J5(K) = 0;
Q = find(G >= 7);
J5(Q) = 255;
subplot(2,3,6), imshow(uint8(J5)), xlabel('(f) 第四种情况滤波图像');
```

### 5.5.2 Prewitt 算子

Prewitt 算子是 1970 年由 Prewitt 提出的边缘检测算子。Prewitt 算子的水平模板和垂直模板如图 5-31 所示,模板的共同特点是系数之和为 0。从模板特征可以看出,水平模板对水平边缘不敏感,不适合水平边缘较多的图像;垂直模板对垂直边缘不敏感,不适合垂直边缘较多的图像。

Prewitt 算子差分示意图如图 5-32 所示。基于 Prewitt 算子模板,得到水平 Prewitt 算子的定义如式(5-49)所示,垂直 Prewitt 算子的定义如式(5-50)所示。

$$P_H = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad (a) \text{ 水平模板}$$

$$P_V = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad (b) \text{ 垂直模板}$$

$$\begin{bmatrix} f(x-1, y-1) & f(x-1, y) & f(x-1, y+1) \\ f(x, y-1) & f(x, y) & f(x, y+1) \\ f(x+1, y-1) & f(x+1, y) & f(x+1, y+1) \end{bmatrix}$$

图 5-31 Prewitt 算子模板

图 5-32 Prewitt 算子差分示意图

$$P_H = (f(x-1, y-1) + f(x, y-1) + f(x+1, y-1)) - (f(x-1, y+1) + f(x, y+1) + f(x+1, y+1)) \quad (5-49)$$

$$P_V = (f(x-1, y-1) + f(x-1, y) + f(x-1, y+1)) - (f(x+1, y-1) + f(x+1, y) + f(x+1, y+1)) \quad (5-50)$$

**例 5.21** Prewitt 算子原理示例。

**解:** 原始图像如图 5-33(a)所示,水平模板滤波图像如图 5-33(b)所示,垂直模板滤波图像如图 5-33(c)所示。从结果可以看出,水平模板对水平边缘不敏感,漏检了水平边缘;垂直模板对垂直边缘不敏感,漏检了垂直边缘。

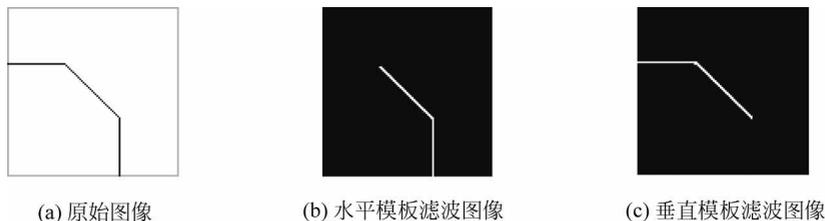


图 5-33 Prewitt 算子原理示例

```
% F5_33.m

im = imread('F5_33.bmp');
subplot(1,3,1), imshow(im), xlabel('(a) 原始图像');

dim = size(size(im));
```

```

if ~ (dim(2) == 2)
    I = rgb2gray(im);
end

MaskPrewittV = fspecial('prewitt')
MaskPrewittH = MaskPrewittV'

KB1 = filter2(MaskPrewittH, I);
subplot(1,3,2), imshow(uint8(KB1)), xlabel('(b) 水平模板滤波图像');

KB2 = filter2(MaskPrewittV, I);
subplot(1,3,3), imshow(uint8(KB2)), xlabel('(c) 垂直模板滤波图像');

```

Prewitt 算子滤波的结果是产生一幅把边线和突变点叠加到暗背景中的图像。为了保持锐化处理的效果,同时恢复原始图像的背景信息,通常将原始图像和 Prewitt 算子滤波图像叠加在一起,加法叠加方法如式(5-51)所示,减法叠加方法如式(5-52)所示。

$$g(x, y) = f(x, y) + \text{Prewitt}(f(x, y)) \quad (5-51)$$

$$g(x, y) = f(x, y) - \text{Prewitt}(f(x, y)) \quad (5-52)$$

式中,原始图像为  $f(x, y)$ ; 变换图像为  $g(x, y)$ ; Prewitt() 为 Prewitt 算子滤波函数。

#### 例 5.22 Prewitt 算子示例。

**解:** 原始图像如图 5-34(a) 所示,水平模板滤波图像及其加法、减法叠加图像分别如图 5-34(b)~图 5-34(d) 所示,垂直模板滤波图像及其加法、减法叠加图像分别如图 5-34(e)~图 5-34(g) 所示。从结果可以看出,就边缘检测而言,水平模板滤波对水平边缘不敏感,忽略较多水平边缘特征,检测出大量垂直边缘信息;垂直模板滤波对垂直边缘不敏感,忽略较多垂直边缘特征,检测出大量水平边缘信息;由于原始图像垂直边缘信息显著,因此,水平模板滤波优于垂直模板滤波。就图像叠加而言,加法叠加方法和减法叠加方法均保持了锐化处理的效果,同时恢复了原始图像的背景信息,具有良好的视觉效果。



(a) 原始图像



(b) 水平模板滤波图像



(c) 水平模板加法叠加图像



(d) 水平模板减法叠加图像

图 5-34 Prewitt 算子示例



(e) 垂直模板滤波图像



(f) 垂直模板加法叠加图像



(g) 垂直模板减法叠加图像

图 5-34 (续)

```

% F5_34.m

I = imread('lena.bmp');
subplot(3,3,2), imshow(I), xlabel('(a) 原始图像');

MaskPrewittV = fspecial('prewitt')
MaskPrewittH = MaskPrewittV'

KB1 = filter2(MaskPrewittH, I);
subplot(3,3,4), imshow(uint8(KB1)), xlabel('(b) 水平模板滤波图像');
subplot(3,3,5), imshow(uint8(double(I) + KB1)), xlabel('(c) 水平模板加法叠加图像');
subplot(3,3,6), imshow(uint8(double(I) - KB1)), xlabel('(d) 水平模板减法叠加图像');

KB2 = filter2(MaskPrewittV, I);
subplot(3,3,7), imshow(uint8(KB2)), xlabel('(e) 垂直模板滤波图像');
subplot(3,3,8), imshow(uint8(double(I) + KB2)), xlabel('(f) 垂直模板加法叠加图像');
subplot(3,3,9), imshow(uint8(double(I) - KB2)), xlabel('(g) 垂直模板减法叠加图像');

```

### 5.5.3 Sobel 算子

Sobel 算子的水平模板和垂直模板如图 5-35 所示,权重为 2 是为了突出中心点的作用,模板系数和为 0 确保了灰度恒定区域的响应为 0。从模板特征可以看出,水平模板对水平边缘不敏感,不适合水平边缘较多的图像;垂直模板对垂直边缘不敏感,不适合垂直边缘较多的图像。

Sobel 算子差分示意图如图 5-36 所示。基于 Sobel 算子模板,得到水平 Sobel 算子的定义如式(5-53)所示,垂直 Sobel 算子的定义如式(5-54)所示。

$$S_H = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad (a) \text{ 水平模板}$$

$$S_V = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (b) \text{ 垂直模板}$$

$$\begin{bmatrix} f(x-1, y-1) & f(x-1, y) & f(x-1, y+1) \\ f(x, y-1) & f(x, y) & f(x, y+1) \\ f(x+1, y-1) & f(x+1, y) & f(x+1, y+1) \end{bmatrix}$$

图 5-35 Sobel 算子模板

图 5-36 Sobel 算子差分示意图

$$S_H = (f(x-1, y-1) + 2f(x, y-1) + f(x+1, y-1)) - (f(x-1, y+1) + 2f(x, y+1) + f(x+1, y+1)) \quad (5-53)$$

$$S_V = (f(x-1, y-1) + 2f(x-1, y) + f(x-1, y+1)) - (f(x+1, y-1) + 2f(x+1, y) + f(x+1, y+1)) \quad (5-54)$$

**例 5.23** Sobel 算子原理示例。

**解：**原始图像如图 5-37(a)所示,水平模板滤波图像如图 5-37(b)所示,垂直模板滤波图像如图 5-37(c)所示。从结果可以看出,水平模板对水平边缘不敏感,漏检了水平边缘;垂直模板对垂直边缘不敏感,漏检了垂直边缘。

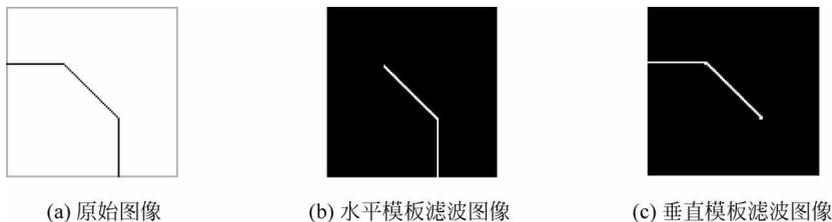


图 5-37 Sobel 算子原理示例

```
% F5_37.m

im = imread('F5_37.bmp');
subplot(1,3,1), imshow(im), xlabel('(a) 原始图像');

dim = size(size(im));
if ~(dim(2) == 2)
    I = rgb2gray(im);
end

MaskPrewittV = fspecial('sobel');
MaskPrewittH = MaskPrewittV';

KB1 = filter2(MaskPrewittH, I);
subplot(1,3,2), imshow(uint8(KB1)), xlabel('(b) 水平模板滤波图像');

KB2 = filter2(MaskPrewittV, I);
subplot(1,3,3), imshow(uint8(KB2)), xlabel('(c) 垂直模板滤波图像');
```

Sobel 算子滤波的结果是产生一幅把边线和突变点叠加到暗背景中的图像。为了保持锐化处理的效果,同时恢复原始图像的背景信息,通常将原始图像和 Sobel 算子滤波图像叠加在一起,加法叠加方法如式(5-55)所示,减法叠加方法如式(5-56)所示。

$$g(x, y) = f(x, y) + \text{Sobel}(f(x, y)) \quad (5-55)$$

$$g(x, y) = f(x, y) - \text{Sobel}(f(x, y)) \quad (5-56)$$

式中,原始图像为  $f(x, y)$ ; 变换图像为  $g(x, y)$ ;  $\text{Sobel}()$  为 Sobel 算子滤波函数。

**例 5.24** Sobel 算子示例。

**解：**原始图像如图 5-38(a)所示,水平模板滤波图像及其加法、减法叠加图像分别如图 5-38(b)~图 5-38(d)所示,垂直模板滤波图像及其加法、减法叠加图像分别如图 5-38(e)~图 5-38(g)所示。从结果可以看出,就边缘检测而言,水平模板滤波对水平边缘不敏感,忽略较多水平边缘特征,检测出大量垂直边缘信息;垂直模板滤波对垂直边缘不敏感,忽略较多垂直边缘特征,检测出大量水平边缘信息。就图像叠加而言,加法叠加方法和减法叠加方

法均保持了锐化处理的效果,同时恢复了原始图像的背景信息,具有良好的视觉效果。

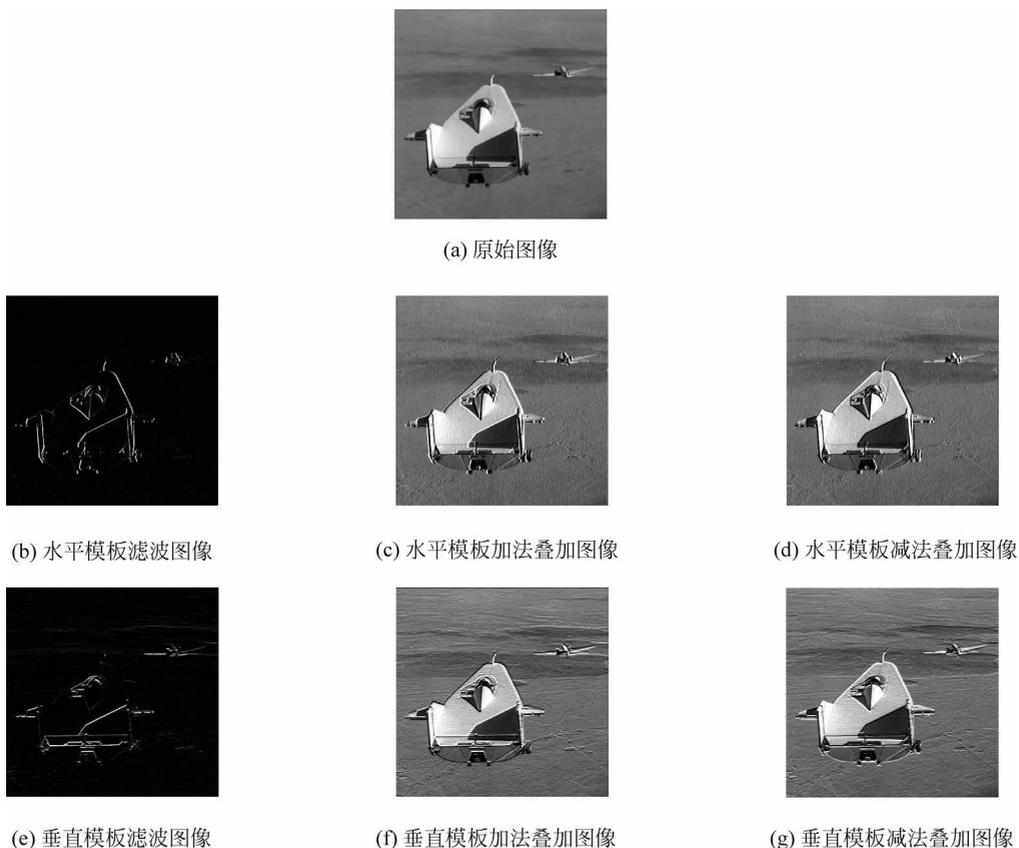


图 5-38 Sobel 算子示例

```
% F5_38.m

I = imread('liftingbody.png');
subplot(3,3,2), imshow(I), xlabel('(a) 原始图像');

MaskSobelV = fspecial('sobel')
MaskSobelH = MaskSobelV'

KB1 = filter2(MaskSobelH, I);
subplot(3,3,4), imshow(uint8(KB1)), xlabel('(b) 水平模板滤波图像');
subplot(3,3,5), imshow(uint8(double(I) + KB1)), xlabel('(c) 水平模板加法叠加图像');
subplot(3,3,6), imshow(uint8(double(I) - KB1)), xlabel('(d) 水平模板减法叠加图像');

KB2 = filter2(MaskSobelV, I);
subplot(3,3,7), imshow(uint8(KB2)), xlabel('(e) 垂直模板滤波图像');
subplot(3,3,8), imshow(uint8(double(I) + KB2)), xlabel('(f) 垂直模板加法叠加图像');
subplot(3,3,9), imshow(uint8(double(I) - KB2)), xlabel('(g) 垂直模板减法叠加图像');
```

### 5.5.4 Log 算子

Log(Laplacian of Gaussian)算子即高斯-拉普拉斯算子,由 Marr 和 Hildreth 提出,被誉为最佳的边缘检测器之一。Log 算子利用高斯滤波器对图像进行平滑,利用拉普拉斯算子对图像进行锐化。Log 算子的原理如下所述。

(1) 二维高斯滤波器的响应函数如下式所示:

$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (5-57)$$

(2) 设:原始图像为  $f(x, y)$ ,由线性系统中卷积与微分的可交换性,可知

$$\nabla^2(g(x, y) * f(x, y)) = (\nabla^2 g(x, y)) * f(x, y) \quad (5-58)$$

(3) 图像的高斯平滑滤波与拉普拉斯微分运算可结合成一个卷积算子,如下式所示:

$$\begin{aligned} \nabla^2 g(x, y) &= \frac{\partial^2 g(x, y)}{\partial x^2} + \frac{\partial^2 g(x, y)}{\partial y^2} = \frac{1}{2\pi\sigma^4} \left( \frac{x^2}{\sigma^2} - 1 \right) e^{-\frac{x^2+y^2}{2\sigma^2}} + \frac{1}{2\pi\sigma^4} \left( \frac{y^2}{\sigma^2} - 1 \right) e^{-\frac{x^2+y^2}{2\sigma^2}} \\ &= \frac{1}{2\pi\sigma^4} \left( \frac{x^2+y^2}{\sigma^2} - 2 \right) e^{-\frac{x^2+y^2}{2\sigma^2}} = A^2 \left( \frac{x^2}{\sigma^2} - 1 \right) e^{-\frac{x^2}{2\sigma^2}} e^{-\frac{y^2}{2\sigma^2}} + A^2 \left( \frac{y^2}{\sigma^2} - 1 \right) e^{-\frac{x^2}{2\sigma^2}} e^{-\frac{y^2}{2\sigma^2}} \\ &= K_1(x)K_2(y) + K_1(y)K_2(x) \end{aligned} \quad (5-59)$$

式中,  $A = \frac{1}{\sqrt{2\pi\sigma^2}}$ ;  $K_1(x) = A \left( \frac{x^2}{\sigma^2} - 1 \right) e^{-\frac{x^2}{2\sigma^2}}$ ;  $K_2(x) = A e^{-\frac{y^2}{2\sigma^2}}$ 。

(4) 用上述算子卷积图像,通过判断符号的变化确定出零交叉点的位置,该位置就是边缘点,这是 Log 算法的基本思想。

(5) 利用  $\nabla^2 g$  的可分解性,对图像的二维卷积可简化为两个一维卷积,如下式所示:

$$\begin{aligned} (\nabla^2 g(x, y)) * f(x, y) &= \sum_{i=-W}^W \sum_{j=-W}^W f(x-j, y-i) (K_1(i)K_2(j) + K_1(j)K_2(i)) \\ &= \sum_{j=-W}^W \left( \left( \sum_{i=-W}^W I(x-j, y-i) K_2(i) \right) K_1(j) + \left( \sum_{i=-W}^W I(x-j, y-i) K_1(i) \right) K_2(j) \right) \\ &= \sum_{j=-W}^W (C(x-j, y) K_1(j) + D(x-j, y) K_2(j)) \end{aligned} \quad (5-60)$$

式中,  $C(x-j, y) = \sum_{i=-W}^W I(x-j, y-i) K_2(i)$ ;  $D(x-j, y) = \sum_{i=-W}^W I(x-j, y-i) K_1(i)$ 。

Log 算子滤波的结果是产生一幅把边线和突变点叠加到暗背景中的图像。为了保持锐化处理的效果,同时恢复原始图像的背景信息,通常将原始图像和 Log 算子滤波图像叠加在一起,加法叠加方法如式(5-61)所示,减法叠加方法如式(5-62)所示。

$$g(x, y) = f(x, y) + \text{Log}(f(x, y)) \quad (5-61)$$

$$g(x, y) = f(x, y) - \text{Log}(f(x, y)) \quad (5-62)$$

式中,原始图像为  $f(x, y)$ ; 变换图像为  $g(x, y)$ ;  $\text{Log}()$  为 Log 算子滤波函数。

**例 5.25** Log 算子示例。

**解:** 原始图像如图 5-39(a)所示,滤波图像如图 5-39(b)所示,加法和减法叠加图像分别如图 5-39(c)和图 5-39(d)所示。从结果可以看出,就边缘检测而言,Log 算子能有效检测出

水平边缘和垂直边缘,具有良好的边缘检测效果。就图像叠加而言,加法叠加方法和减法叠加方法均恢复了原始图像的背景信息,具有良好的视觉效果;同时,加法叠加方法实现了图像平滑的效果,减法叠加方法实现了图像锐化的效果。

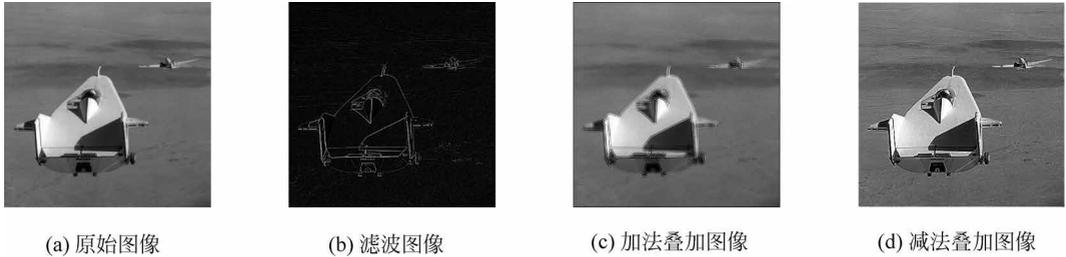


图 5-39 Log 算子示例

```
% F5_39.m

I = imread('liftingbody.png');
subplot(2,2,1), imshow(I), xlabel('(a) 原始图像');

MaskLog = fspecial('log');
KB = filter2(MaskLog, I);
subplot(2,2,2), imshow(uint8(KB)), xlabel('(b) 滤波图像');
subplot(2,2,3), imshow(uint8(double(I) + KB)), xlabel('(c) 加法叠加图像');
subplot(2,2,4), imshow(uint8(double(I) - KB)), xlabel('(d) 减法叠加图像');
```

### 5.5.5 高通滤波

高通滤波是指让高频分量顺利通过,对低频分量充分抑制。由于图像边缘与图像频谱中的高频分量相对应,因此,高通滤波使图像边缘变得清晰,实现了图像锐化。

空间域高通滤波建立在离散卷积的基础上,如下式所示:

$$g(x, y) = \sum_{(i, j) \in S} f(i, j) H(x - i + 1, y - j + 1) \quad (5-63)$$

式中,  $f(x, y)$  为原始图像;  $g(x, y)$  为变换图像;  $S$  为像素  $(x, y)$  的预定邻域;  $H(x - i + 1, y - j + 1)$  为冲击响应矩阵,常用的实现模板如图 5-40 所示,模板的共同特点是系数有正有负且和为 1。

$$H_1 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

(a) 模板1

$$H_2 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

(b) 模板2

$$H_3 = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

(c) 模板3

图 5-40 高通滤波实现模板

由于高通滤波模板系数之和为 1,因此,高通滤波的结果是产生一幅把边线和突变点叠加到亮背景中的图像。

**例 5.26** 高通滤波示例。

**解:** 原始图像如图 5-41(a)所示,模板 1 滤波图像如图 5-41(b)所示,模板 2 滤波图像如

图 5-41(c)所示,模板 3 滤波图像如图 5-41(d)所示。从结果可以看出,高通滤波产生一幅把边缘信息叠加到亮背景中的图像,且有效实现了图像锐化。

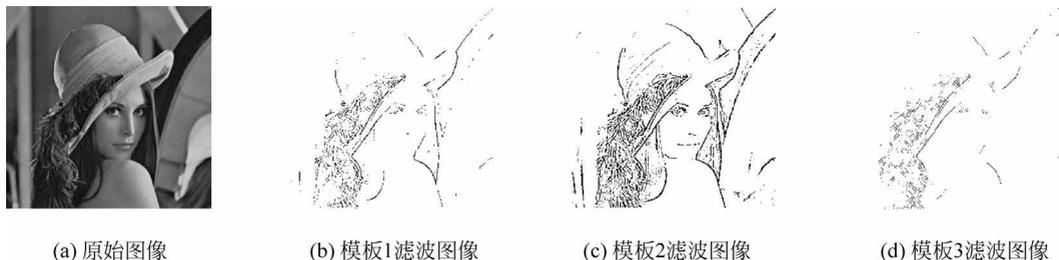


图 5-41 高通滤波示例

```
% F5_41.m

I = imread('lena.bmp');
subplot(2,2,1), imshow(I), xlabel('(a) 原始图像');

mask1 = [ 0  -1  0
         -1  5  -1
          0  -1  0]
KB1 = filter2(mask1,I);
subplot(2,2,2), imshow(KB1), xlabel('(b) 模板 1 滤波图像');

mask2 = [-1  -1  -1
         -1   9  -1
         -1  -1  -1]
KB2 = filter2(mask2,I);
subplot(2,2,3), imshow(KB2), xlabel('(c) 模板 2 滤波图像');

mask3 = [ 1  -2  1
         -2  5  -2
          1  -2  1]
KB3 = filter2(mask3,I);
subplot(2,2,4), imshow(KB3), xlabel('(d) 模板 3 滤波图像');
```

### 5.5.6 掩模法

掩模法是指用原始图像减去低通(平滑)图像得到高通(锐化)图像,这样能够恢复高频滤波时丢失的部分低频成分,也称为非锐化掩模法,如下式所示:

$$g(x,y) = kf(x,y) - \text{Lowpass}(f(x,y)) \quad (5-64)$$

式中, $f(x,y)$ 为原始图像; $g(x,y)$ 为变换图像; $k$ 为放大系数; $\text{Lowpass}()$ 为低通滤波函数。

掩模法的结果是产生一幅把边线和突变点叠加到暗背景中的图像。为了使锐化效果可视化,可通过提高暗背景的灰度值,采用如下式所示方法突出对比度:

$$g(x,y) = \begin{cases} g(x,y), & g(x,y) \geq T \\ 255, & g(x,y) < T \end{cases} \quad (5-65)$$

**例 5.27** 掩模法示例。

**解：**原始图像如图 5-42(a)所示,  $3 \times 3$  模板低通滤波图像如图 5-42(b)所示, 掩模法滤波图像如图 5-42(c)所示, 突出对比度图像如图 5-42(d)所示。从结果可以看出, 掩模法有效实现了图像锐化。



图 5-42 掩模法示例

```
% F5_42.m

I = imread('lena.bmp');
subplot(2,2,1), imshow(I), xlabel('(a) 原始图像');

mask = ones(3)/9
J = filter2(mask,I);
subplot(2,2,2), imshow(uint8(J)), xlabel('(b) 3 x 3 模板低通滤波图像');

K = double(I) - J;
subplot(2,2,3), imshow(uint8(K)), xlabel('(c) 掩模法滤波图像');

Q = find(K < 5);
K(Q) = 255;
subplot(2,2,4), imshow(uint8(K)), xlabel('(d) 突出对比度图像');
```

## 习题

- 5-1 试简述空域滤波增强的步骤。
- 5-2 将  $M$  幅图像相加求平均可以达到消除噪声的效果, 用一个  $n \times n$  的模板进行平滑滤波也可以达到消除噪声的效果。试比较这两种方法的区别和联系。
- 5-3 试简述加权平均法的基本思想及其模板系数的确定原则。
- 5-4 试简述中值滤波的基本原理及特点。
- 5-5 试简述中值滤波的根信号的概念及特点。
- 5-6 试简述二维中值滤波的基本原理及步骤。

5-7 试简述最小值滤波、最大值滤波和中点滤波的概念。

5-8 若原始图像数据如图 5-43(a)所示,试求:

$$\begin{bmatrix} 10 & 40 & 70 \\ 100 & 130 & 160 \\ 190 & 220 & 250 \end{bmatrix}$$

(a) 原始图像数据

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

(b) 滤波模板1

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

(c) 滤波模板2

图 5-43 习题 5-8 图

(1) 若采用图 5-43(b)所示的滤波模板对原始图像进行滤波,试求:最小值滤波、中值滤波、最大值滤波、中点滤波的结果。

(2) 若采用图 5-43(c)所示的滤波模板对原始图像进行滤波,试求:最小值滤波、中值滤波、最大值滤波、中点滤波的结果。

5-9 若一维图像数据如图 5-44 所示,试求其一阶微分和二阶微分的结果。

$$[10 \quad 30 \quad 30 \quad 50 \quad 100 \quad 200 \quad 200 \quad 200 \quad 150 \quad 90]$$

图 5-44 一维图像数据