



华北电力大学
NORTH CHINA ELECTRIC POWER UNIVERSITY

3 空间域滤波



主要内容

- 空间域滤波概述
 - ✓ 空间域滤波器
 - ✓ 模板卷积
 - ✓ 空间域滤波器分类
- 平滑滤波器
 - ✓ 均值滤波器
 - ✓ 加权均值滤波器
 - ✓ 中值滤波器
 - ✓ 最值滤波器



主要内容

- 锐化滤波器
 - ✓ 梯度滤波器
 - ✓ Roberts滤波器
 - ✓ Prewitt滤波器
 - ✓ Sobel滤波器
 - ✓ 拉普拉斯滤波器



3.1 空间域滤波概述

- 图像滤波是一种用来**改变或增强图像**的技术，通过滤波可以**强化或去除**图像的某种特征，它能够处理图像的操作包括平滑、锐化、边缘增强等。
- 滤波是一种**邻域**操作，在这种操作中，输出图像的像素点的值取决于输入图像某个邻域内的像素点的值。



3.1.1 空间滤波器

- 空间域滤波采用[filter (滤波器)、mask (掩模)、kernel (核)、template (模板)、window (窗)]方法对图像进行滤波，去除图像噪声或增强图像的细节。
- [filter (滤波器)、mask (掩模)、kernel (核)、template (模板)、window (窗)]称为空间域滤波器。



3.1.1 空间滤波器

- 在 $M \times N$ 的图像 f 上，使用 $m \times n$ 的滤波器：

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)$$

$$m = 2a + 1, n = 2b + 1$$

- ✓ $w(s, t)$ 滤波器系数， $f(x, y)$ 是图像灰度值， $g(x, y)$ 滤波后的图像。
- 空间滤波的简化形式：

$$R = w_1 z_1 + w_2 z_2 + \cdots + w_{mn} z_{mn}$$

- ✓ w_i 是滤波器系数， z_i 是与该系数对应的图像灰度值， mn 为滤波器中包含的像素点总数。



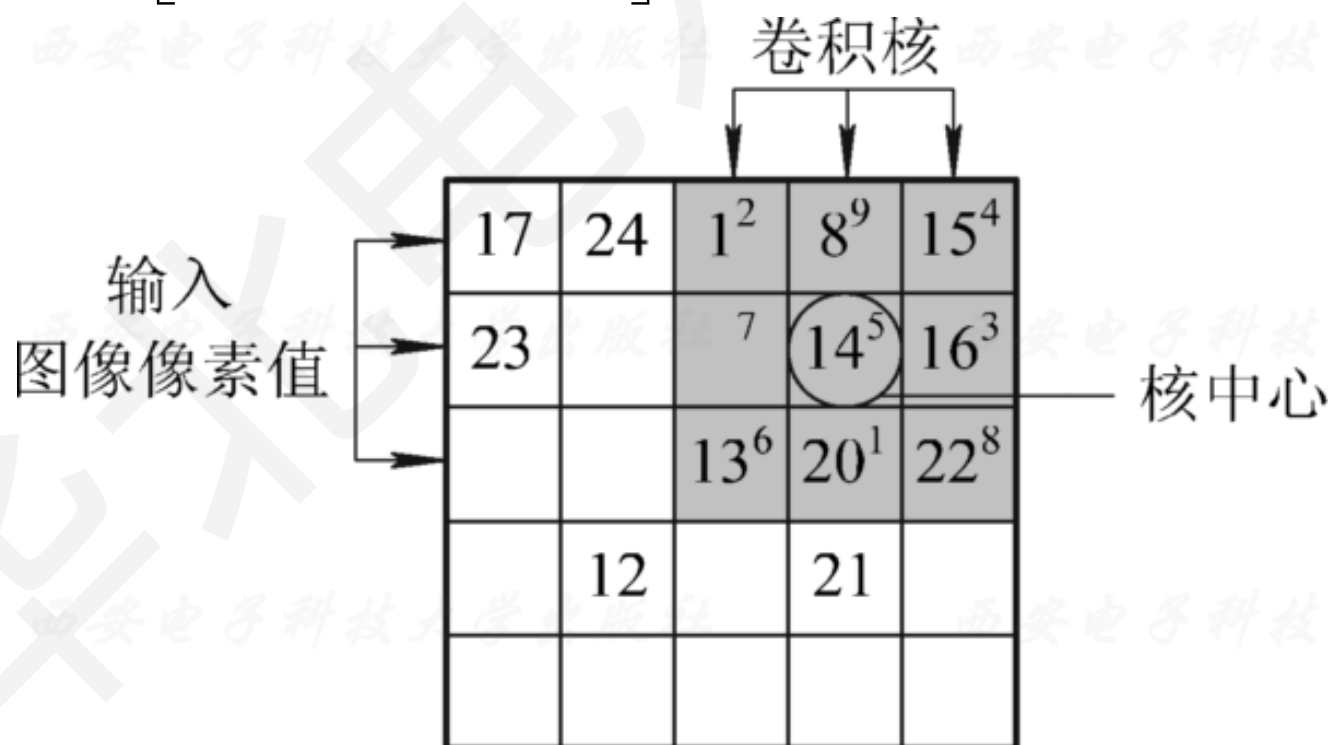
3.1.2 模板卷积

- 卷积是一种邻域操作，操作中输出图像的像素点的值是对应输入像素点邻域内像素点值的加权和，其中权矩阵称为卷积核矩阵，也称为滤波器。卷积核是相关核旋转180度得到的。
- 卷积的计算过程：
 - ✓ 将卷积核绕其中心元素旋转180°
 - ✓ 将旋转后的卷积核平移，使其中心元素与某个像素点重合
 - ✓ 将卷积核元素与输入图像的对应像素相乘
 - ✓ 将以上各项相乘的结果相加，得到卷积结果。



3.1.2 模板卷积

$$A = \begin{bmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{bmatrix} \quad h = \begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix}$$



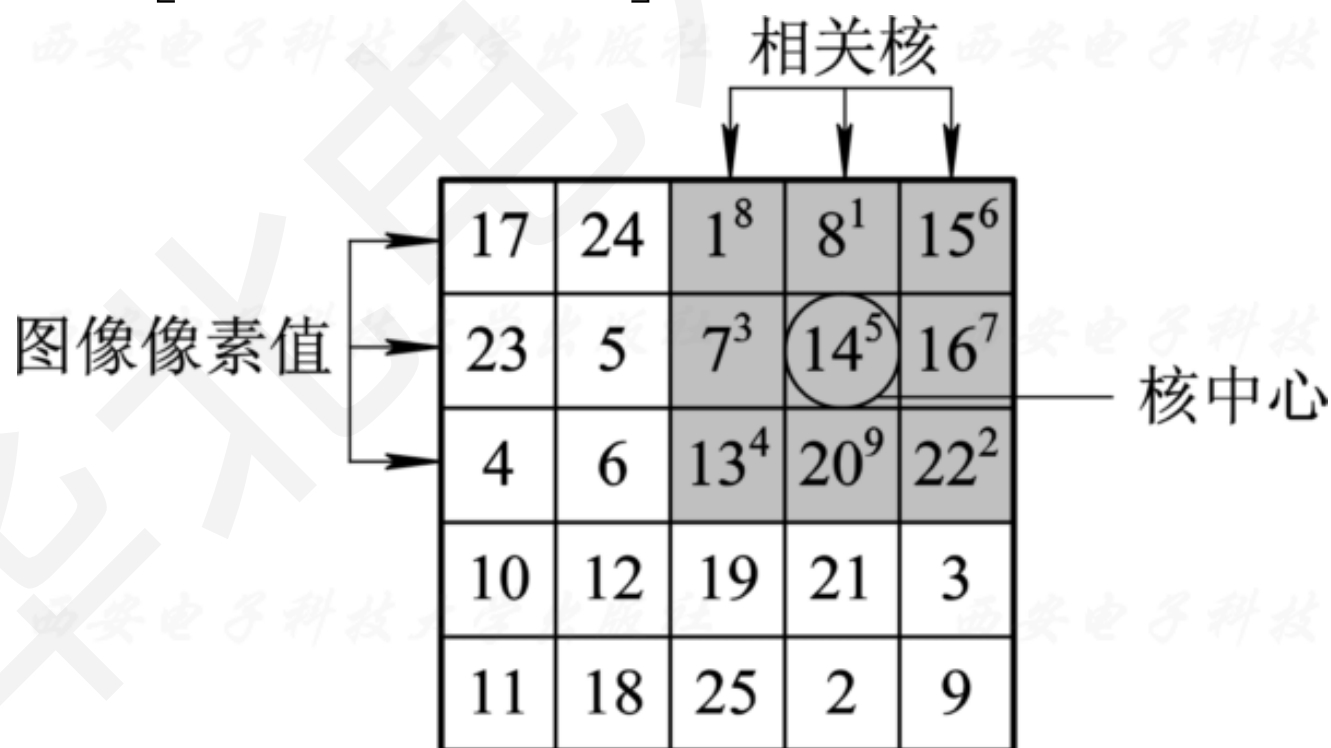


3.1.2 模板卷积

$$R = w_1 z_1 + w_2 z_2 + \cdots + w_{mn} z_{mn}$$

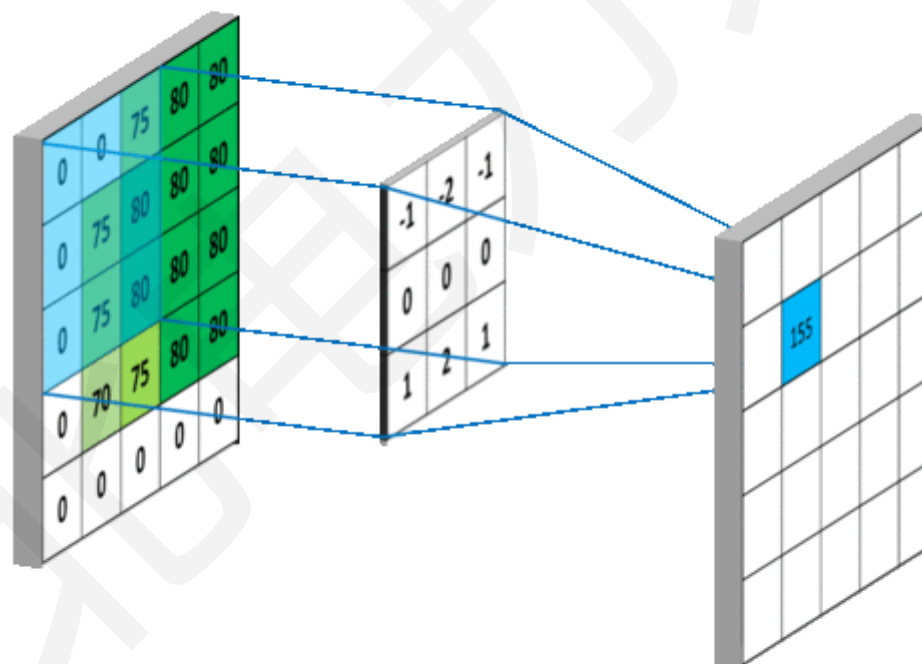
$$A = \begin{bmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{bmatrix}$$

$$h = \begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix}$$





3.1.2 模板卷积





3.1.2 模板卷积-为什么旋转180°

$$I = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad K = \begin{bmatrix} 1 & -2 \\ 2 & 1 \end{bmatrix}$$

做Full卷积, 结果为4x4矩阵

$$(I * K)(0,0) = \sum_{m=0}^1 \sum_{n=0}^2 I(m,n) K(0-m, 0-n)$$

$$= I(0,0)K(0,0) + I(0,1)K(0,-1) + I(0,2)K(0,-2) \\ + I(1,0)K(-1,0) + I(1,1)K(-1,-1) + I(1,2)K(-1,-2) \\ = 1 \times (-1) = -1$$

$$(I * K)(0,1) = \sum_{m=0}^1 \sum_{n=0}^2 I(m,n) K(0-m, 1-n) = -4$$

$$(I * K)(0,2) = \sum_{m=0}^1 \sum_{n=0}^2 I(m,n) K(0-m, 2-n) = -7$$

$$(I * K)(0,3) = \sum_{m=0}^1 \sum_{n=0}^2 I(m,n) K(0-m, 3-n) = -6$$

$$(I * K)(3,3) = \sum_{m=0}^1 \sum_{n=0}^2 I(m,n) K(3-m, 3-n) = 9$$

$$\begin{bmatrix} 1 & 2 \\ -2 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 2 \\ 4 & 5 \\ 7 & 8 \\ 14 & 23 \end{bmatrix} \quad K_{\text{flip}} = \begin{bmatrix} 1 & 2 \\ -2 & -1 \end{bmatrix}$$

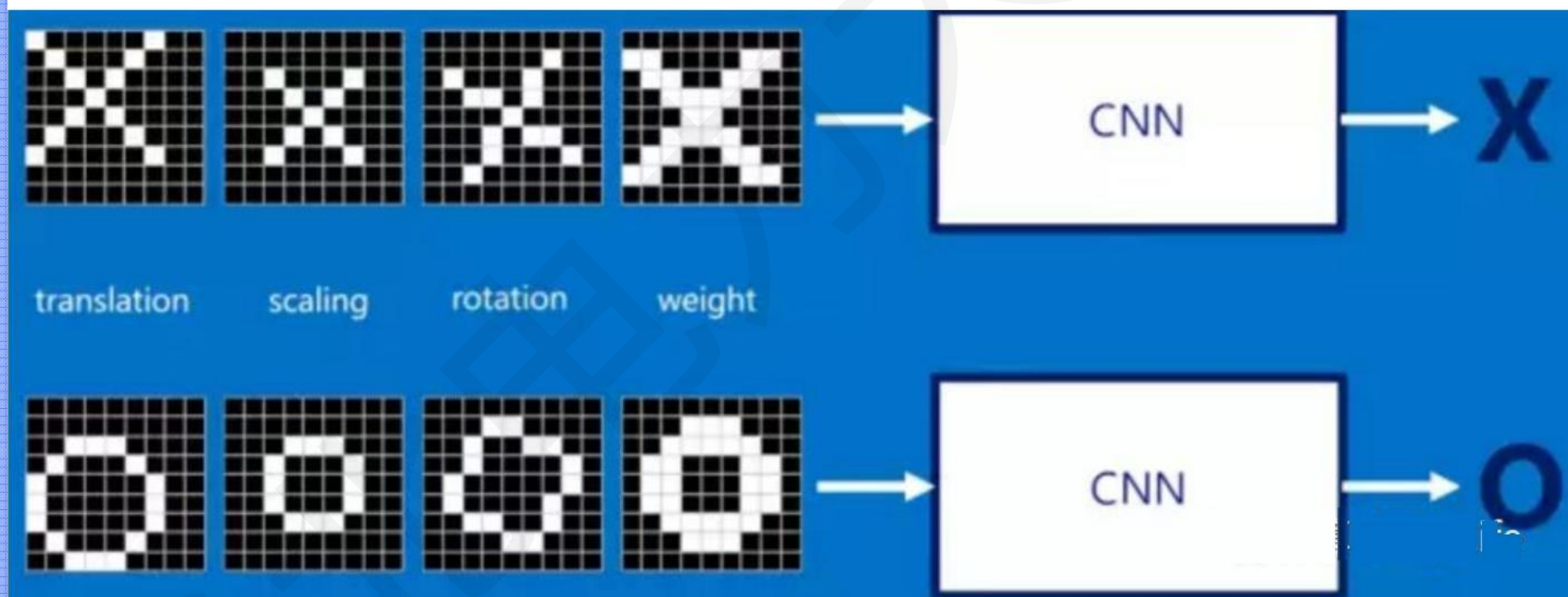
相当于K逆转180°扫描I

$$\Rightarrow \begin{bmatrix} -1 & -4 & -7 & -6 \\ -2 & -8 & -8 & -9 \\ 1 & -8 & -8 & -12 \\ 14 & 23 & 26 & 9 \end{bmatrix}$$

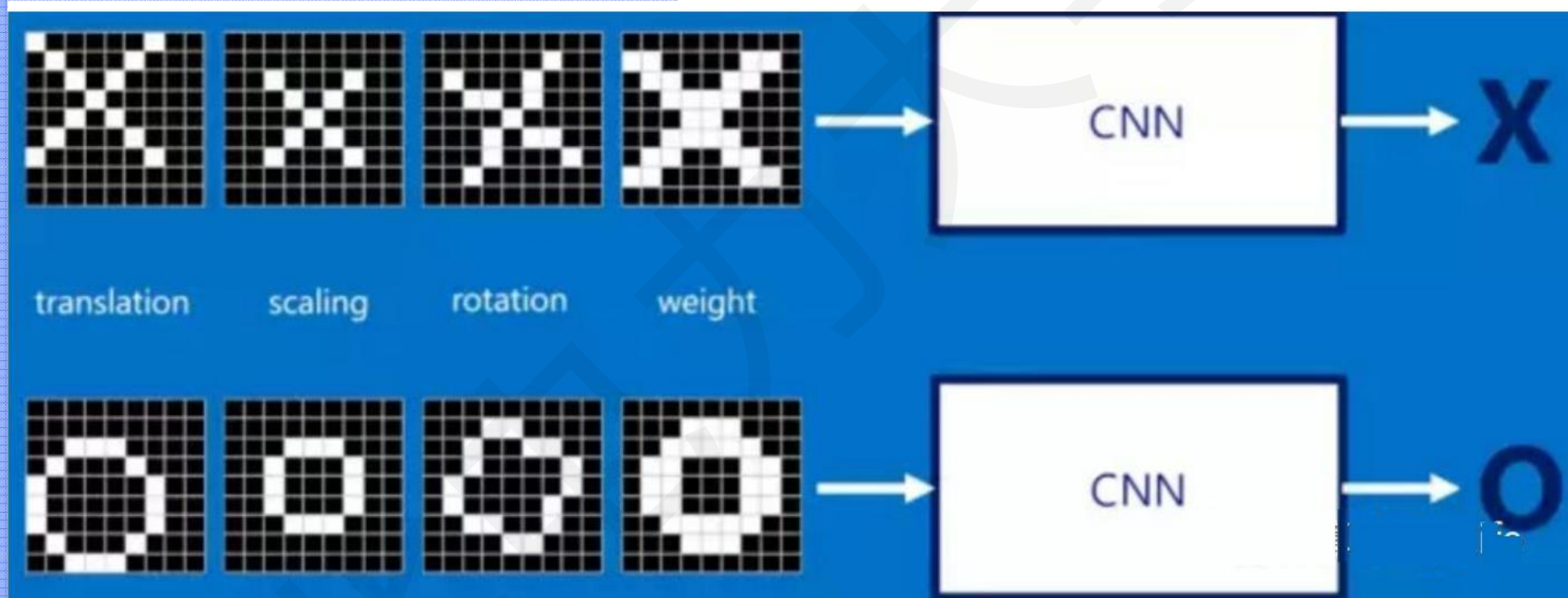
Result.

卷积运算

3.1.2 模板卷积-作用*



3.1.2 模板卷积-作用*

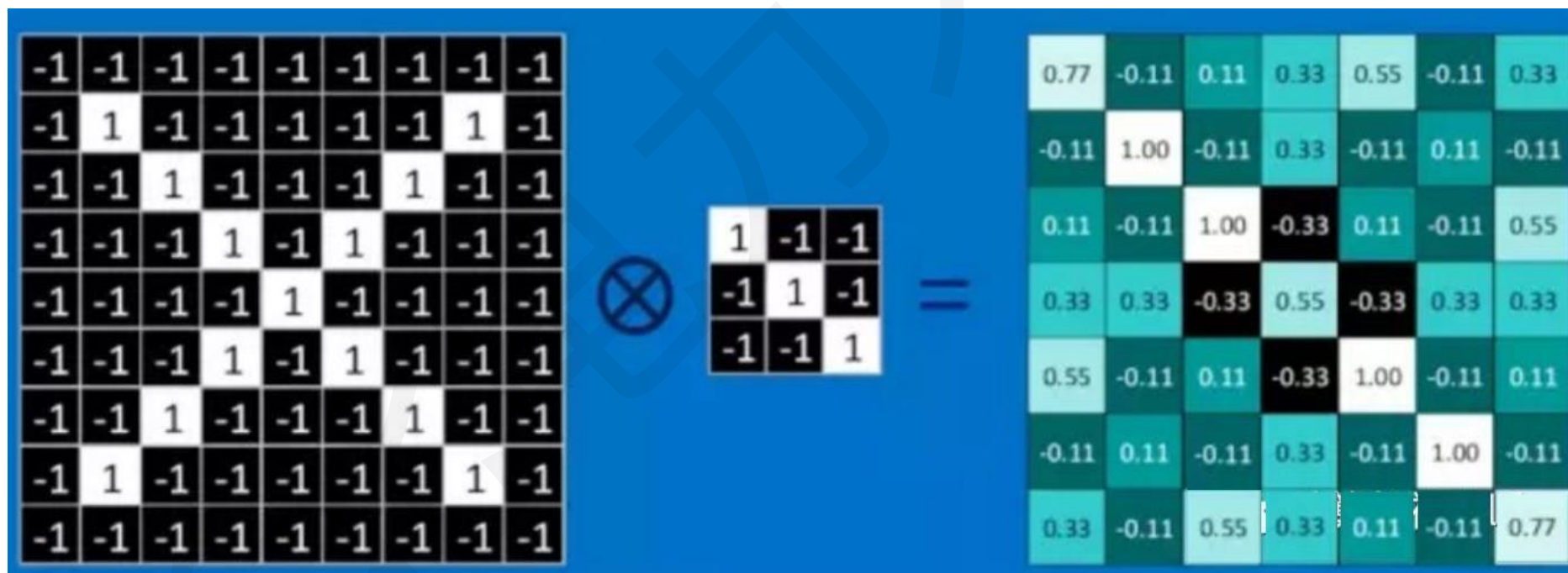


1	-1	-1
-1	1	-1
-1	-1	1

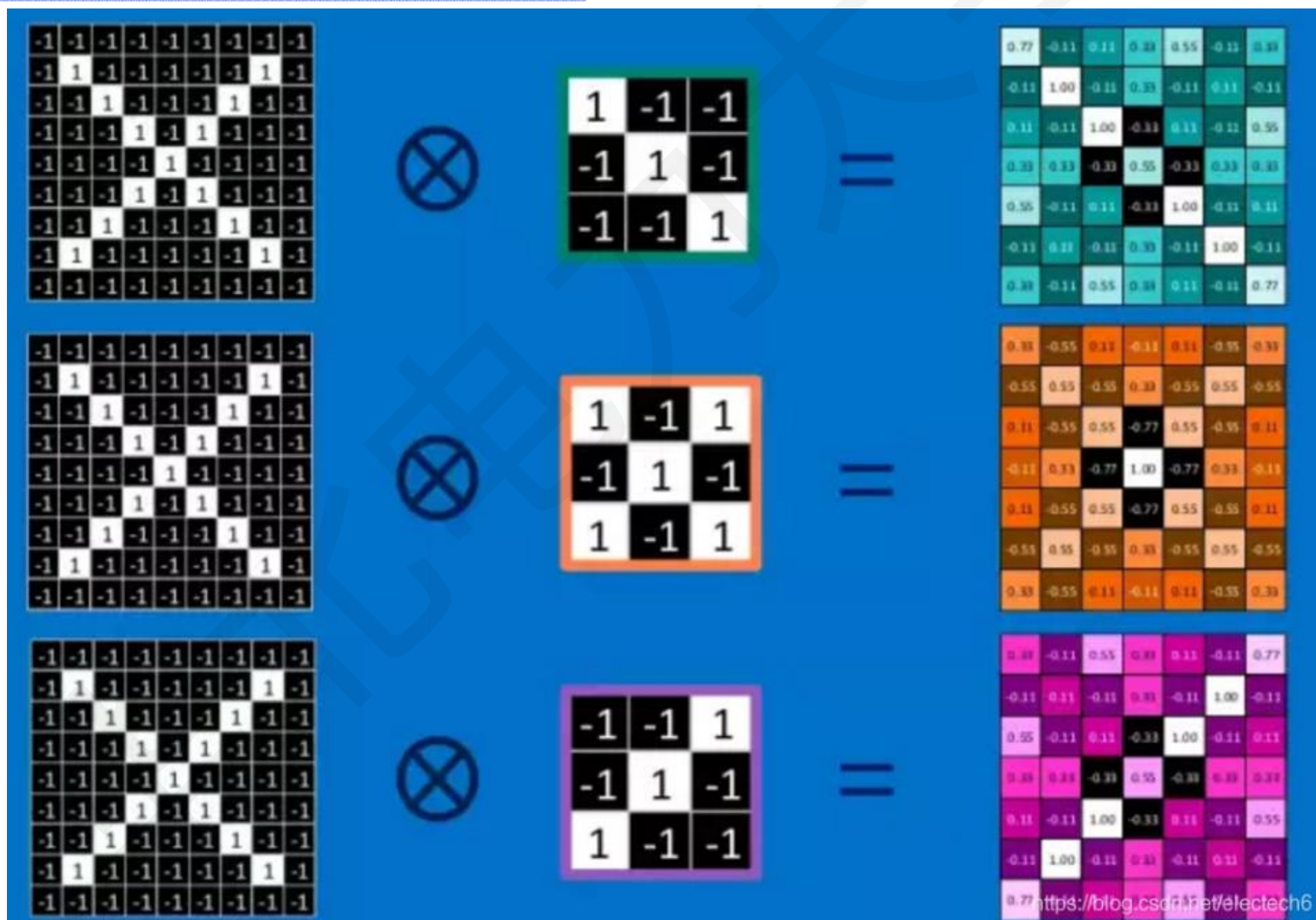
1	-1	1
-1	1	-1
1	-1	1

-1	-1	1
-1	1	-1
1	-1	-1

3.1.2 模板卷积-作用*



3.1.2 模板卷积-作用*





3.1.3 空间域滤波的分类-处理效果

- (1) 平滑滤波器

- ✓ 平滑滤波能减弱或消除图像中的高频率分量，但不影响低频率分量。
- ✓ 因为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，平滑滤波将这些分量滤去可减少局部灰度的起伏，使图像变得比较平滑。
- ✓ 实际中，平滑滤波还可用于消除图像中的噪声（噪声的空间相关性较弱，且对应较高的空间频率），以及在提取较大的目标前去除太小的细节或将目标内的小间断连接起来。



3.1.3 空间域滤波的分类-处理效果

- (2) 锐化滤波器

- ✓ 锐化滤波能减弱或消除图像中的低频率分量，但不影响高频率分量。
- ✓ 因为低频分量对应图像中灰度值缓慢变化的区域，因而与图像的整体特性如整体对比度和平均灰度值等有关。
- ✓ 锐化滤波将这些分量滤去可使图像反差增加，边缘明显。
- ✓ 实际应用中，锐化滤波可用于增强图像中被模糊的细节或景物的边缘。



3.1.3 空间域滤波的分类-数学形态

- (1) 线性滤波器
 - ✓ 线性滤波器由线性系统组成，包括均值滤波器、加权均值滤波器等。
- (2) 非线性滤波器
 - ✓ 非线性滤波器使用模板进行结果像素值的计算，结果值直接取决于像素邻域的值，而不与线性乘积和无关，包括中值滤波，最大最小值滤波器等。



3.2 平滑滤波器

- 线性滤波器
 - ✓ 均值滤波器和加权均值滤波器
- 非线性滤波器
 - ✓ 中值滤波器、最大最小值滤波器

3.2.1 均值滤波器

- 包含在滤波器邻域内像素的平均值
 - ✓ 减小图像灰度的“尖锐”变化，减小噪声。
 - ✓ 由于图像边缘是由图像灰度尖锐变化引起的，所以也存在边缘模糊的问题。

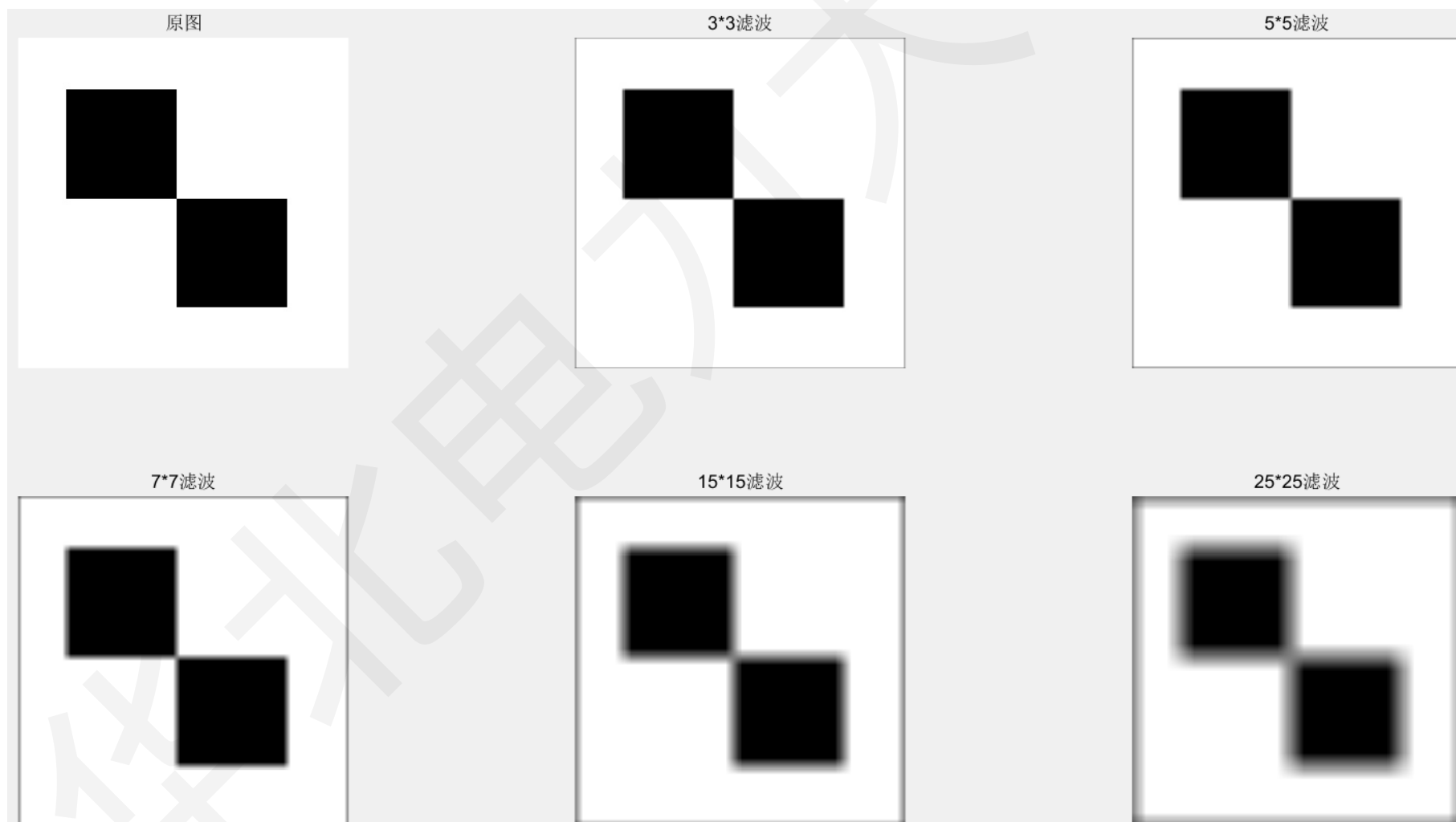
$$\frac{1}{9} \times$$

1	1	1
1	1	1
1	1	1

$$g(x, y) = \frac{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)}{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t)}$$



3.2.1 均值滤波器



黑色框100*100



3.2.1 均值滤波器

- 生成滤波掩模 w 的函数`fspecial`的语法为：

$w = \text{fspecial}(\text{'type'}, \text{parameters})$

- ✓ `'type'` 表示滤波器类型
- ✓ `'parameters'` 进一步定义了指定的滤波器
- ✓ `fspecial('average', [r c])`. 大小为 $r \times c$ 的一个矩形平均滤波器。默认值为 3×3 . 若由一个数来代替 $[r \ c]$, 则表示方形滤波器。



3.2.1 均值滤波器

- 对图像执行滤波操作：

`g = imfilter(f, w, filtering_mode, boundary_options, size_options)`

- ✓ `f`为输入图像，`w`为滤波模板，`g`为滤波后图像。
- ✓ `filtering_mode`用于指定在滤波过程中是使用“相关”还是“卷积”。
- ✓ `boundary_options`用于处理边界充零问题，边界的大小由滤波器的大小确定。
- ✓ `size_options`用于指定生成图像的大小，‘full’输出图像的大小与被扩展图像的大小相同；‘same’输出图像的大小与输入图像的大小相同。



3.2.1 均值滤波器

```
I1 = imread('blackblock.jpg');
I1 = im2double(I1);
F1 = fspecial('average',[3]); %滤波器
I2 = imfilter(I1, F1); %图像滤波
F2 = fspecial('average',[5]); %滤波器
I3 = imfilter(I1, F2); %图像滤波
F3 = fspecial('average',[7]); %滤波器
I4 = imfilter(I1, F3); %图像滤波
F4 = fspecial('average',[15]); %滤波器
I5 = imfilter(I1, F4); %图像滤波
F5 = fspecial('average',[25]); %滤波器
I6 = imfilter(I1, F5); %图像滤波

figure(1);
subplot(2,3,1);imshow(I1);title('原图');
subplot(2,3,2);imshow(I2);title('3*3滤波');
subplot(2,3,3);imshow(I3);title('5*5滤波');
subplot(2,3,4);imshow(I4);title('7*7滤波');
subplot(2,3,5);imshow(I5);title('15*15滤波');
subplot(2,3,6);imshow(I6);title('25*25滤波');
```

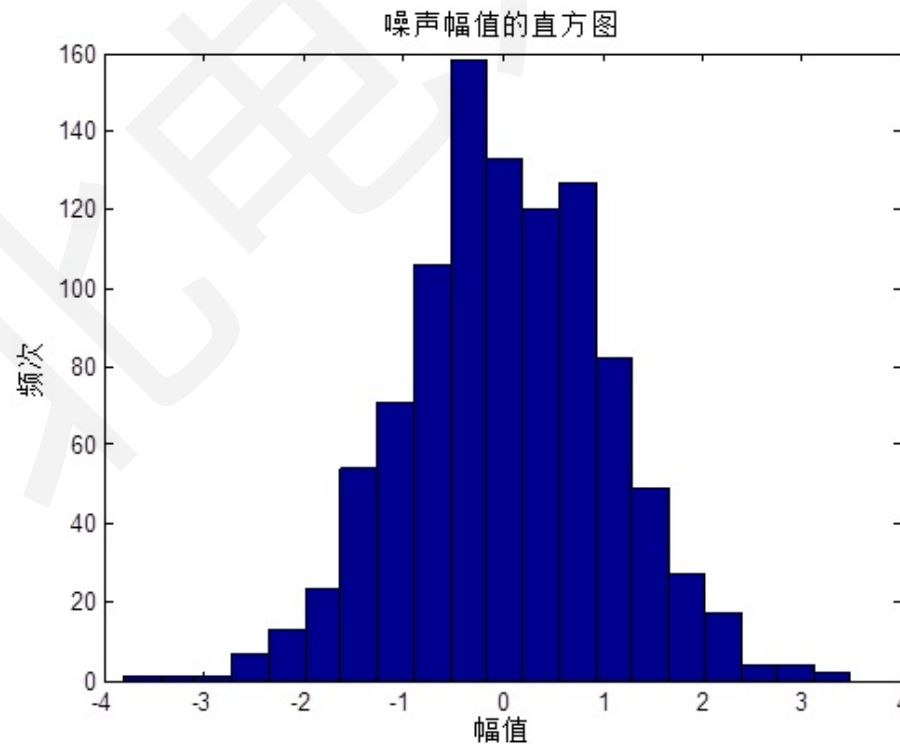

补充 高斯噪声与椒盐噪声

- 高斯噪声，顾名思义是指服从高斯分布（正态分布）的一类噪声，通常是因为不良照明和高温引起的传感器噪声。通常在RGB图像中，显现比较明显。



补充 高斯噪声与椒盐噪声

- 高斯噪声，顾名思义是指服从高斯分布（正态分布）的一类噪声，通常是因为不良照明和高温引起的传感器噪声。通常在RGB图像中，显现比较明显。



补充 高斯噪声与椒盐噪声

- 椒盐噪声，通常是由图像传感器，传输信道，解压处理等产生的黑白相间的亮暗点噪声（椒-黑，盐-白），通常出现在灰度图中。



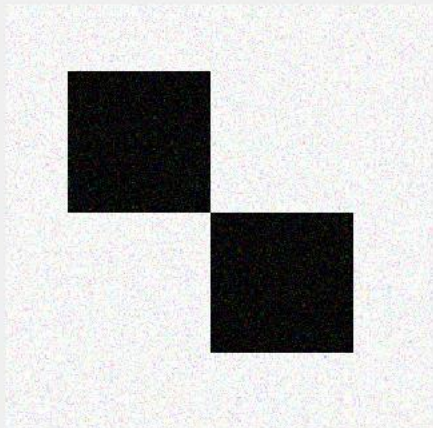


3.2.1 均值滤波器

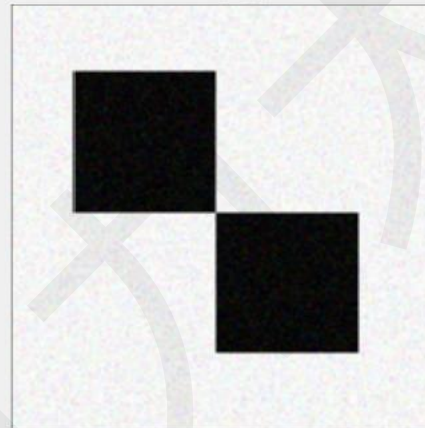
```
I1 = imread('blackblock.jpg');  
I1 = im2double(I1);  
I1 = imnoise(I1, 'gaussian');  
F1 = fspecial('average',[3]); %滤波器  
I2 = imfilter(I1, F1); %图像滤波  
F2 = fspecial('average',[5]); %滤波器  
I3 = imfilter(I1, F2); %图像滤波  
F3 = fspecial('average',[7]); %滤波器  
I4 = imfilter(I1, F3); %图像滤波  
F4 = fspecial('average',[15]); %滤波器  
I5 = imfilter(I1, F4); %图像滤波  
F5 = fspecial('average',[25]); %滤波器  
I6 = imfilter(I1, F5); %图像滤波
```


3.2.1 均值滤波器

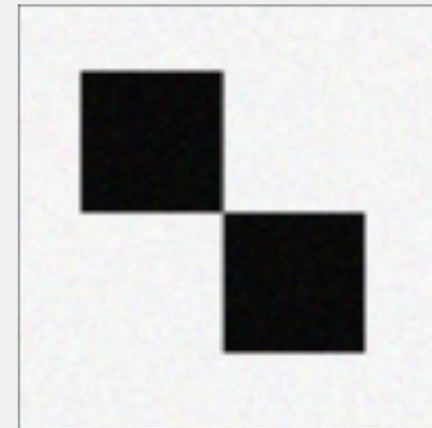
原图



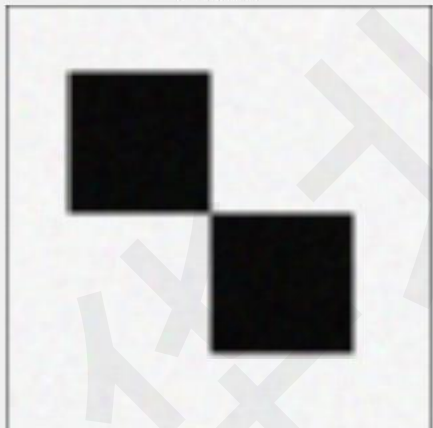
3*3滤波



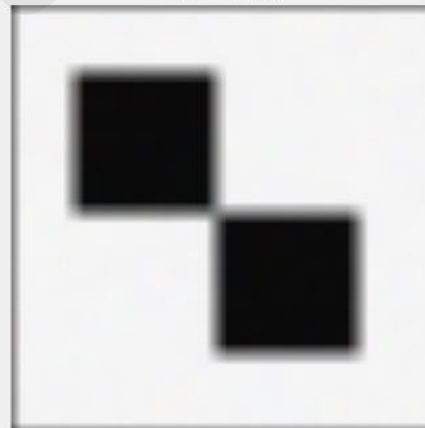
5*5滤波



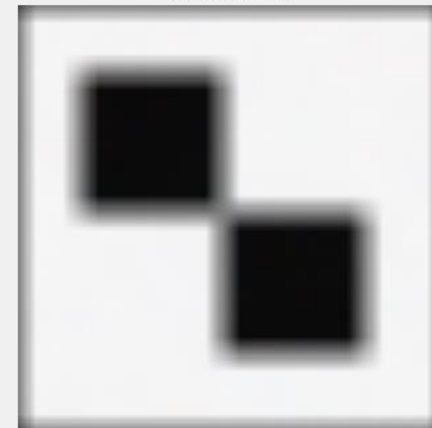
7*7滤波



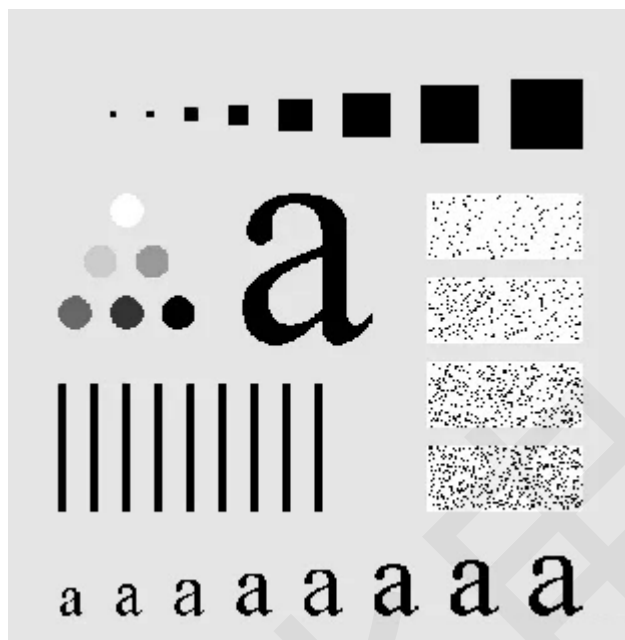
15*15滤波



25*25滤波

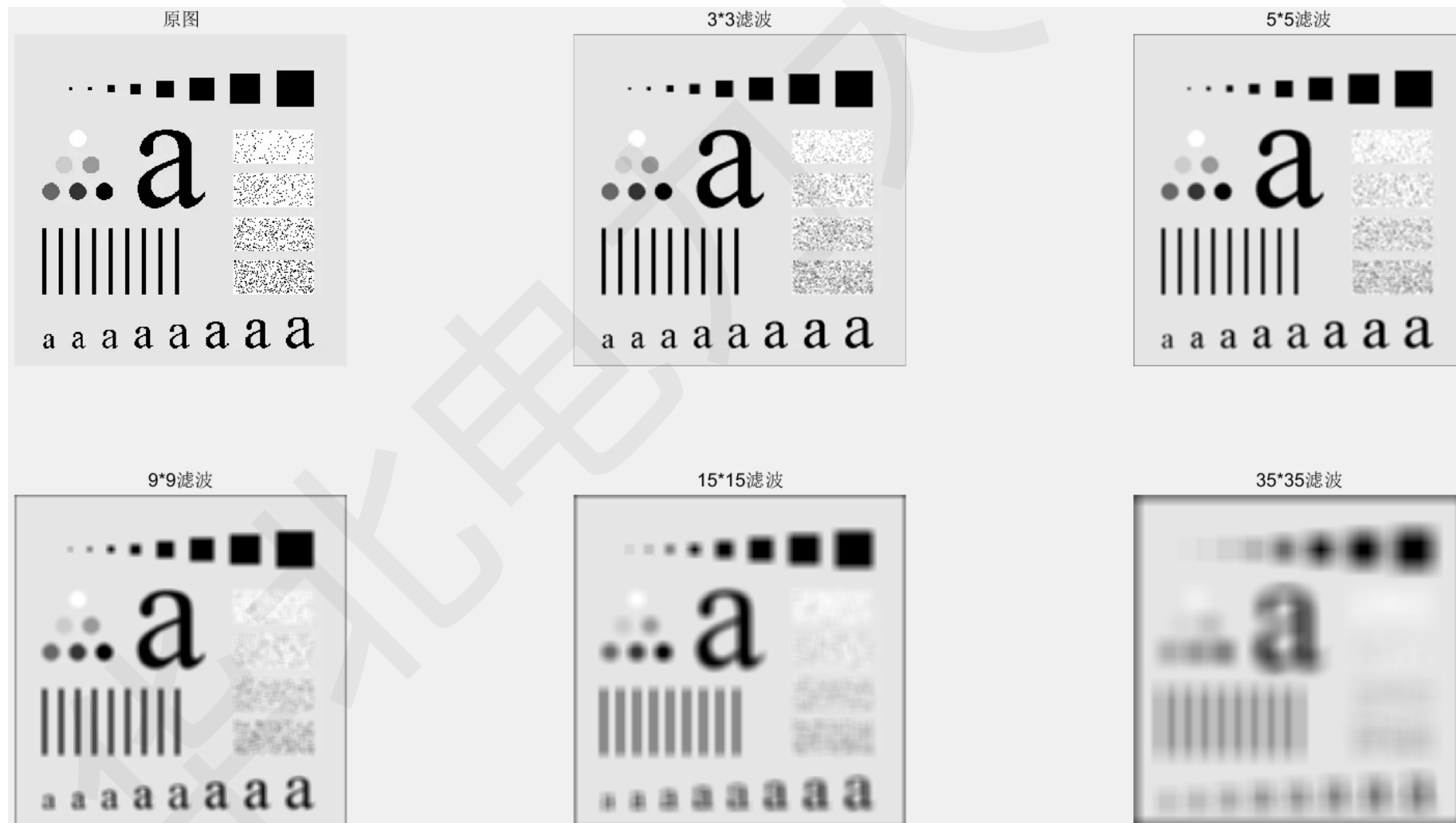


3.2.1 均值滤波器



- ✓ 顶端的黑方块，大小分别为3, 5, 9, 15, 25, 35, 45, 55个像素，边界相隔25个像素。
- ✓ 位于底端的字母在10到24个像素之间，增量为2个像素。
- ✓ 垂直线段5个像素宽，100个像素高，间隔20个像素。
- ✓ 圆的直径25个像素，边缘相隔15个像素，灰度以20%增加。
- ✓ 噪声矩形大小是50*120像素。

3.2.1 均值滤波器





3.2.2 加权均值滤波器

$$\frac{1}{16} \times$$

1	2	1
2	4	2
1	2	1

$$g(x, y) = \frac{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)}{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t)}$$



3.2.2 加权均值滤波器

$$H_1 = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad H_2 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$$H_3 = \frac{1}{8} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad H_4 = \frac{1}{2} \begin{bmatrix} 0 & \frac{1}{4} & 0 \\ \frac{1}{4} & 1 & \frac{1}{4} \\ 0 & \frac{1}{4} & 0 \end{bmatrix}$$

3.2.2 加权均值滤波器

原图



中间2其他1



中间4



中间0其他1



中间1



3*3滤波





3.2.2 加权均值滤波器

```
I1 = imread('lena.jpg');  
I1 = im2double(I1);  
I1 = imnoise(I1, 'salt & pepper'); %椒盐噪声  
  
F1 = ones(3,3);  
F1(2,2) = 2;  
F1 = F1/10;  
I2 = imfilter(I1, F1);
```

3.2.3 中值滤波器

- 用模板区域内像素的中间值，作为结果值

$$R = \text{mid} \{z_k \mid k = 1, 2, \dots, n\}$$

- ✓ 强迫突出的亮点或暗点更象它周围的值，以消除孤立的亮点或暗点





3.2.3 中值滤波器

- 将模板区域内的像素排序，求出中间值，例如：

3x3的模板，第5大的是中值，

5x5的模板，第13大的是中值，

7x7的模板，第25大的是中值，

9x9的模板，第41大的是中值。

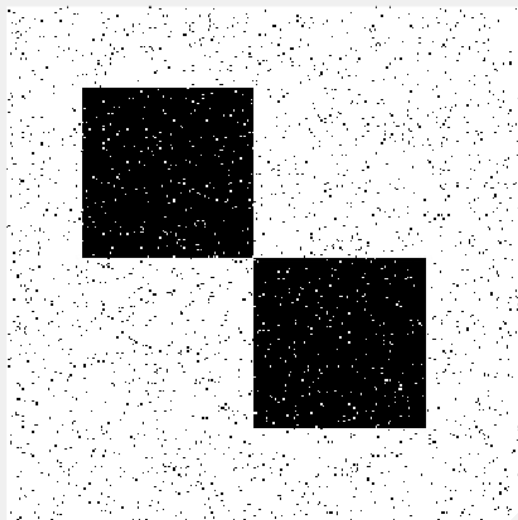


3.2.3 中值滤波器

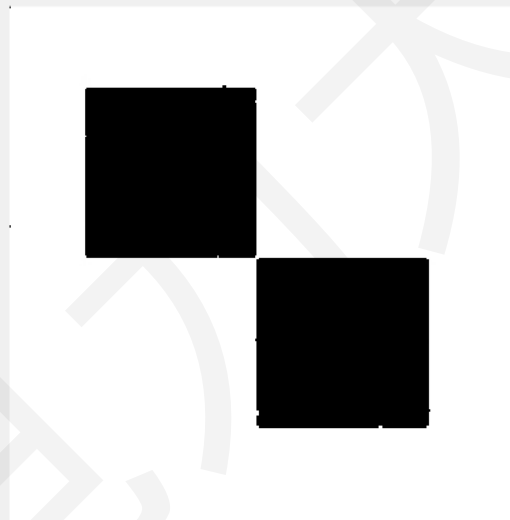
- (1) 在去除噪音的同时，可以比较好地保留边的锐度和图像的细节（优于均值滤波器）
- (2) 能够有效去除脉冲噪声：以黑白点（椒盐噪声）叠加在图像上中。

3.2.3 中值滤波器

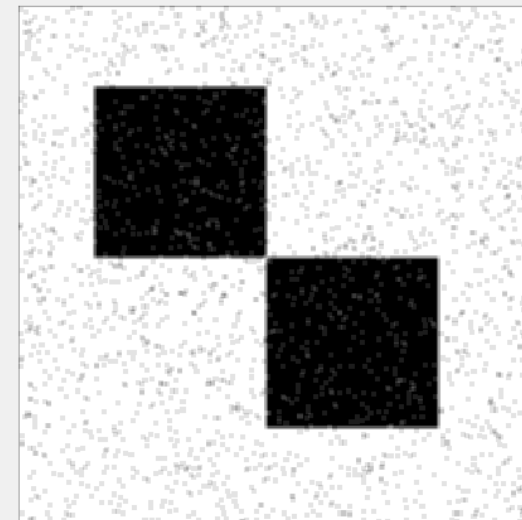
椒盐噪声原图



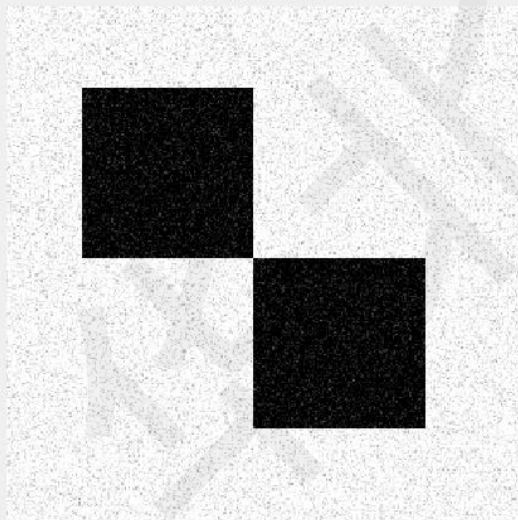
中值滤波



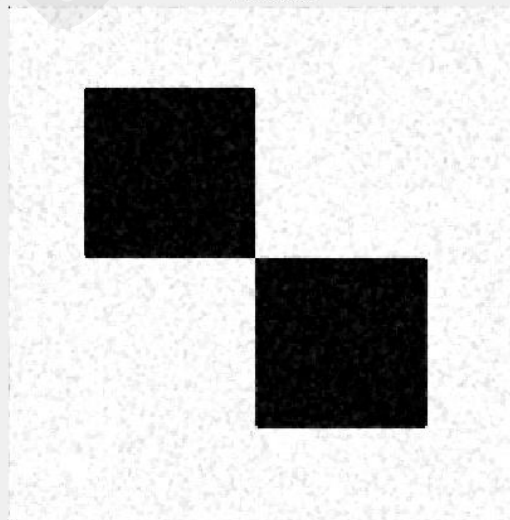
均值滤波



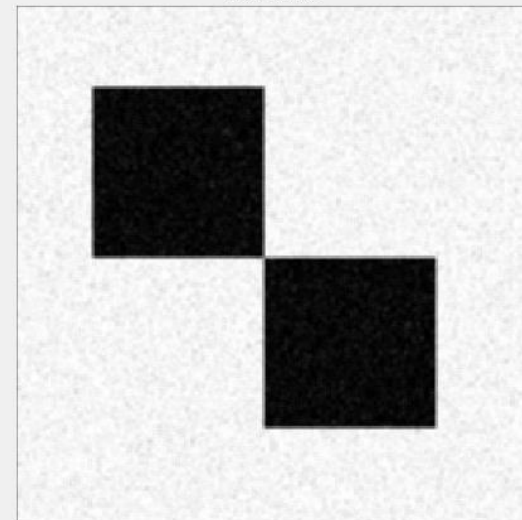
高斯噪声原图



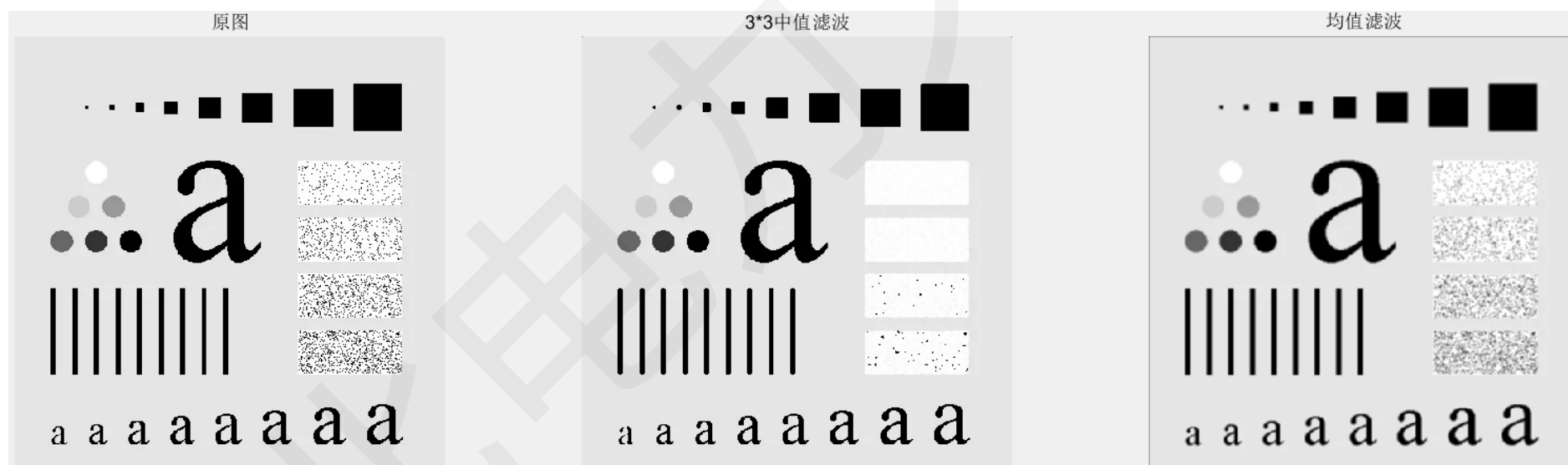
中值滤波



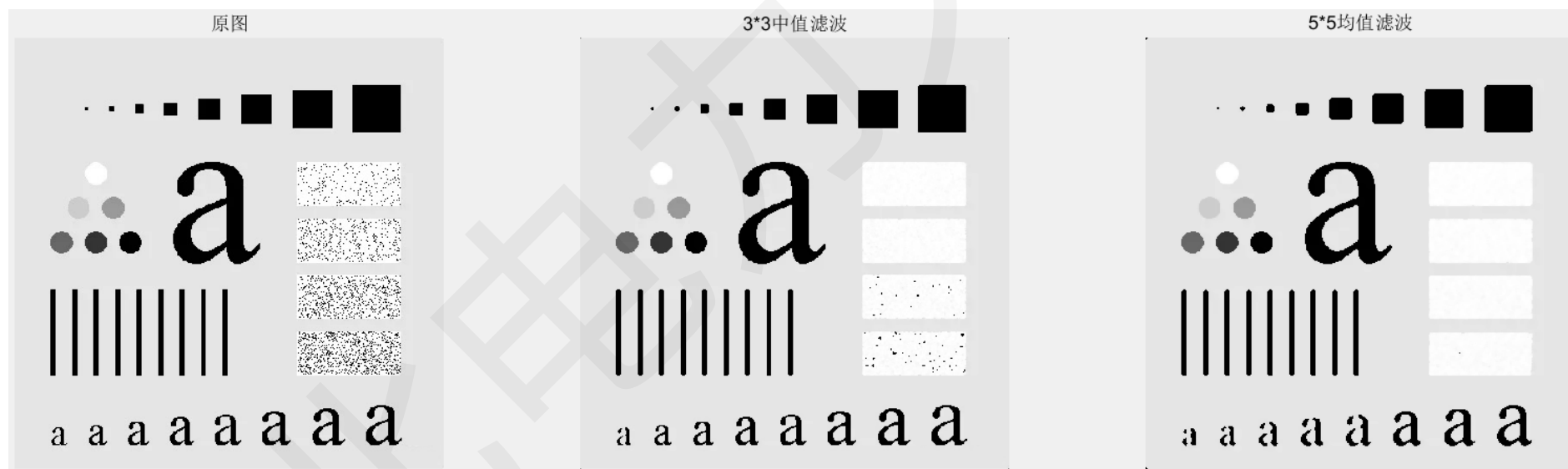
均值滤波



3.2.3 中值滤波器



3.2.3 中值滤波器





3.2.3 中值滤波器

```
I1 = imread('blackblock.jpg');  
I1 = rgb2gray(I1); %必须转为灰度图像  
I1 = im2double(I1);  
I1 = imnoise(I1,'salt & pepper'); %椒盐噪声  
  
I2 = medfilt2(I1, [3,3]); %中值滤波  
  
F1 = fspecial('average',[3]); %均值滤波  
I3 = imfilter(I1, F1);
```



3.2.4 最大值滤波器

- 在滤波窗口内寻找最亮点

$$Y = \text{ordfilt2}(X, \text{order}, \text{domain})$$

- ✓ X为输入图像
- ✓ order为滤波器输出的顺序值
- ✓ domain为滤波窗口。

`I2 = ordfilt2(I1, 9, ones(3,3)); %最大值滤波`



3.2.5 最小值滤波器

- 在滤波窗口内寻找最暗点

$$Y = \text{ordfilt2}(X, \text{order}, \text{domain})$$

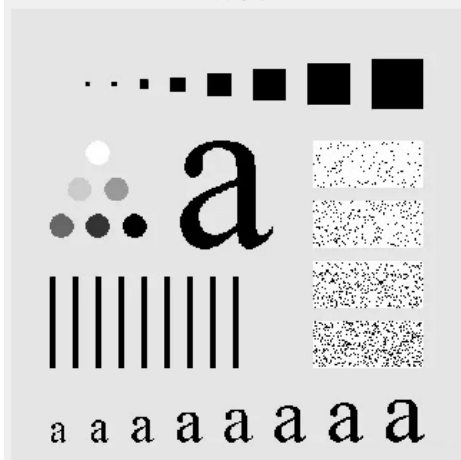
- ✓ X为输入图像
- ✓ order为滤波器输出的顺序值
- ✓ domain为滤波窗口。

`I3 = ordfilt2(I1, 1, ones(3,3)); %最小值滤波`

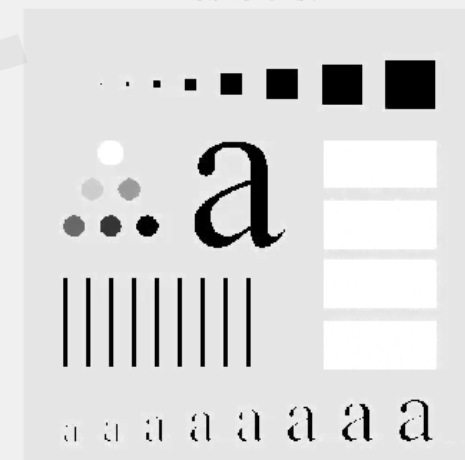


3.2.5 最小值滤波器

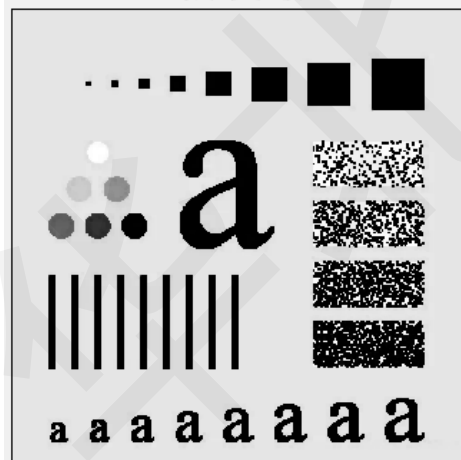
原图



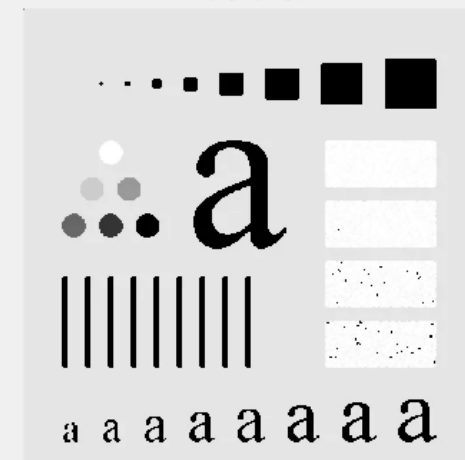
最大值滤波



最小值滤波



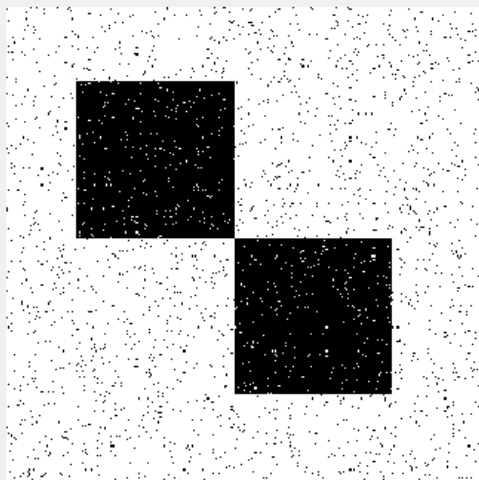
中值滤波





3.2.5 最小值滤波器

原图

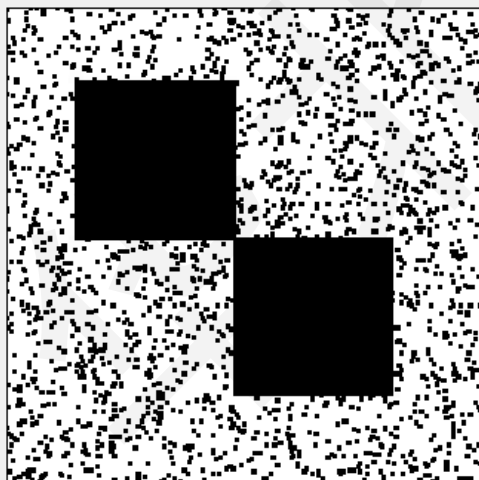


最大值滤波



3*3

最小值滤波

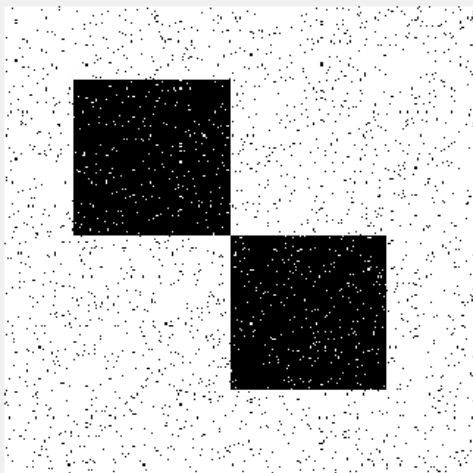


中值滤波



3.2.5 最小值滤波器

原图

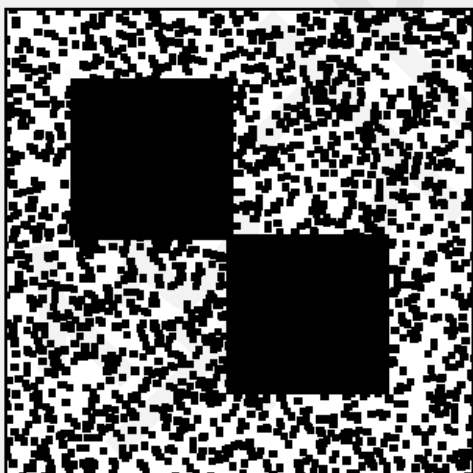


最大值滤波

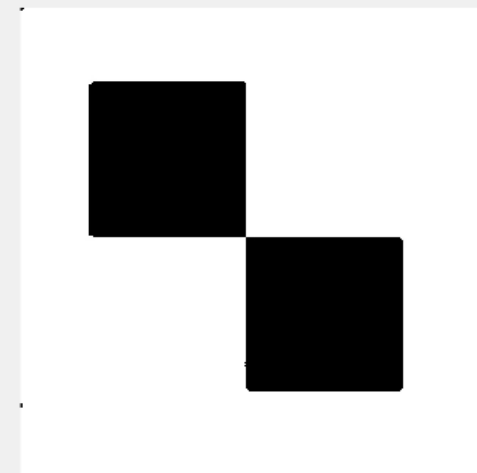


5*5

最小值滤波



中值滤波





3.2.5 最小值滤波器

```
I1 = imread('blackblock.jpg');  
I1 = rgb2gray(I1); %必须转为灰度图像  
I1 = im2double(I1);  
I1 = imnoise(I1,'salt & pepper'); %椒盐噪声  
  
I2 = ordfilt2(I1, 25, ones(5,5)); %最大值滤波  
  
I3 = ordfilt2(I1, 1, ones(5,5)); %最小值滤波  
  
I4 = ordfilt2(I1, 13, ones(5,5)); %中值滤波
```