Lecture 7-1： Spatial Filtering

Yuyao Zhang, Xiran Cai PhD

zhangyy8@shanghaitech.edu.cn caixr@shanghaitech.edu.cn

SIST Building 2 302-F/302-C

Course piazza link： piazza.com/shanghaitech.edu.cn/spring2021/cs270spring2021

上海科技大学 ShanghaiTech University

Outline

- **Spatial filtering definition**
- **Smoothing（平滑）**
  - Linear filter
  - Non-linear filter
- **Sharpening（锐化）**
  - Spatial differentiation
  - Laplace filter

Spatial Filtering

- **A Spatial filter is directly applied on the image**
- **A Spatial filter is also called spatial masks（掩模）、kernels（核）、templates（模板）、windows（窗口）**
- **A Spatial filter consists of**

  1) neighborhood 2) a predefined operation

- **A Spatial filter can be linear and nonlinear**
  - Linear spatial filter corresponds to spectral filter in frequency domain
  - Nonlinear spatial filter cannot be accomplished in frequency domain

Time-domain Convolution

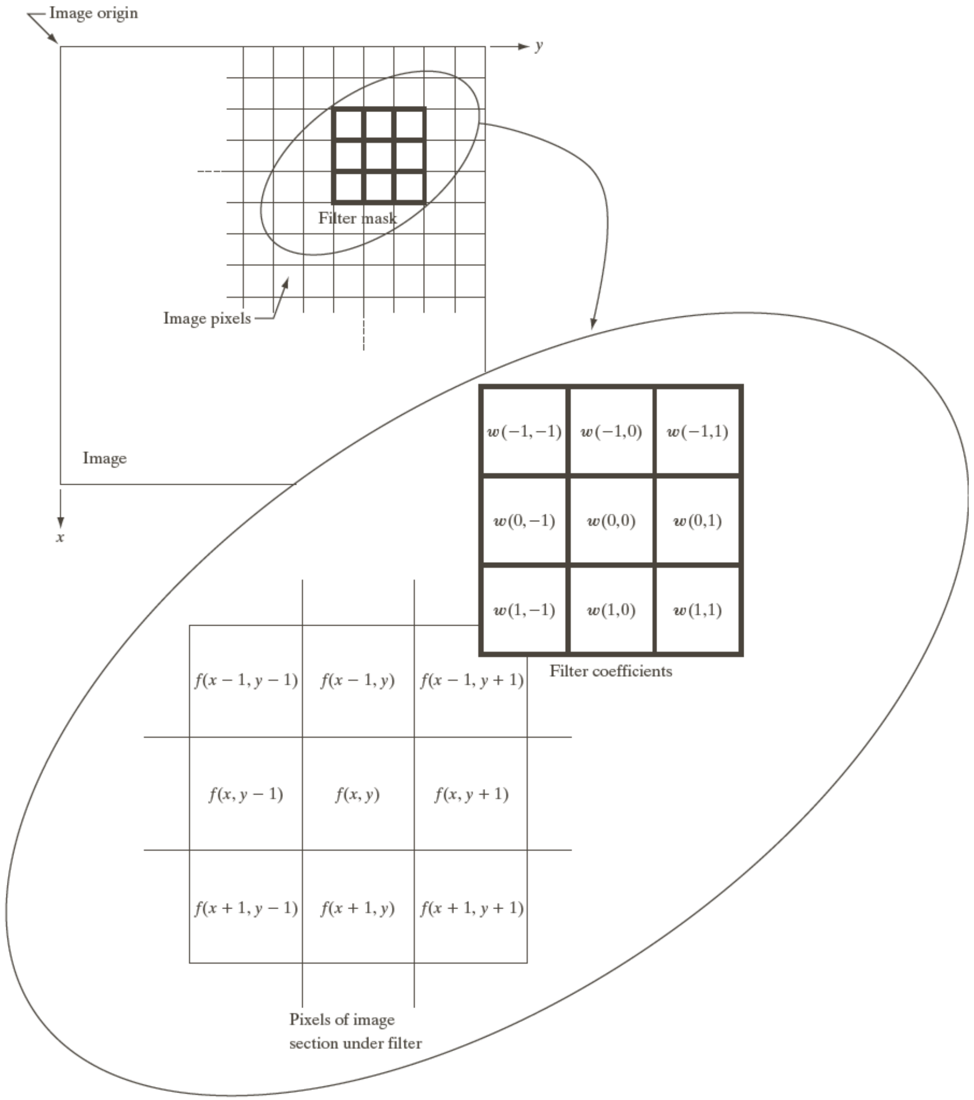
- Convolution of two signals x(t) and h(t), denoted by $x(t) * h(t)$, is defined by

$$y(t) = x(t) * h(t) = \int_{-\infty}^{+\infty} x(\tau)h(t-\tau)d\tau$$

- For discrete-time

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{+\infty} x[k]h[n-k]$$

Spatial Filter



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

- $f(x,y)$: input image
- $g(x,y)$: output filtered image
- $w(s,t)$: $m \times n$ spatial filter, where $m = 2a+1, n = 2b+1$

Spatial Filter

- For discrete-time convolution

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{+\infty} x[k]h[n-k]$$

- Spatial filter

$w(-1,-1)$	$w(-1,0)$	$w(-1,1)$
$w(0,-1)$	$w(0,0)$	$w(0,1)$
$w(1,-1)$	$w(1,0)$	$w(1,1)$

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

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

Correlation and Convolution (2D)

Origin $f(x, y)$

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

$w(x, y)$

1	2	3
4	5	6
7	8	9

(a)

Padded f

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(b)

Initial position for w

1	2	3	0	0	0	0	0	0
4	5	6	0	0	0	0	0	0
7	8	9	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(c)

Full correlation result

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	9	8	7	0	0	0
0	0	0	6	5	4	0	0	0
0	0	0	3	2	1	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(d)

Cropped correlation result

0	0	0	0	0
0	9	8	7	0
0	6	5	4	0
0	3	2	1	0
0	0	0	0	0

(e)

Rotated w

9	8	7	0	0	0	0	0	0
6	5	4	0	0	0	0	0	0
3	2	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(f)

Full convolution result

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	1	2	3	0	0	0
0	0	0	4	5	6	0	0	0
0	0	0	7	8	9	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(g)

Cropped convolution result

0	0	0	0	0
0	1	2	3	0
0	4	5	6	0
0	7	8	9	0
0	0	0	0	0

(h)

Equations

Correlation

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

Convolution

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

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Spatial Filter Masks

- **Linear Spatial Filter (线性滤波器)**
  - $R = \frac{1}{9}\sum_{k=1}^{9} z_k$
  - $h(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$
- **Nonlinear Spatial Filter（非线性滤波器）**
  - Max filter (最大值滤波)
  - Median filter （中值滤波）

Smooth Filters (平滑滤波器)

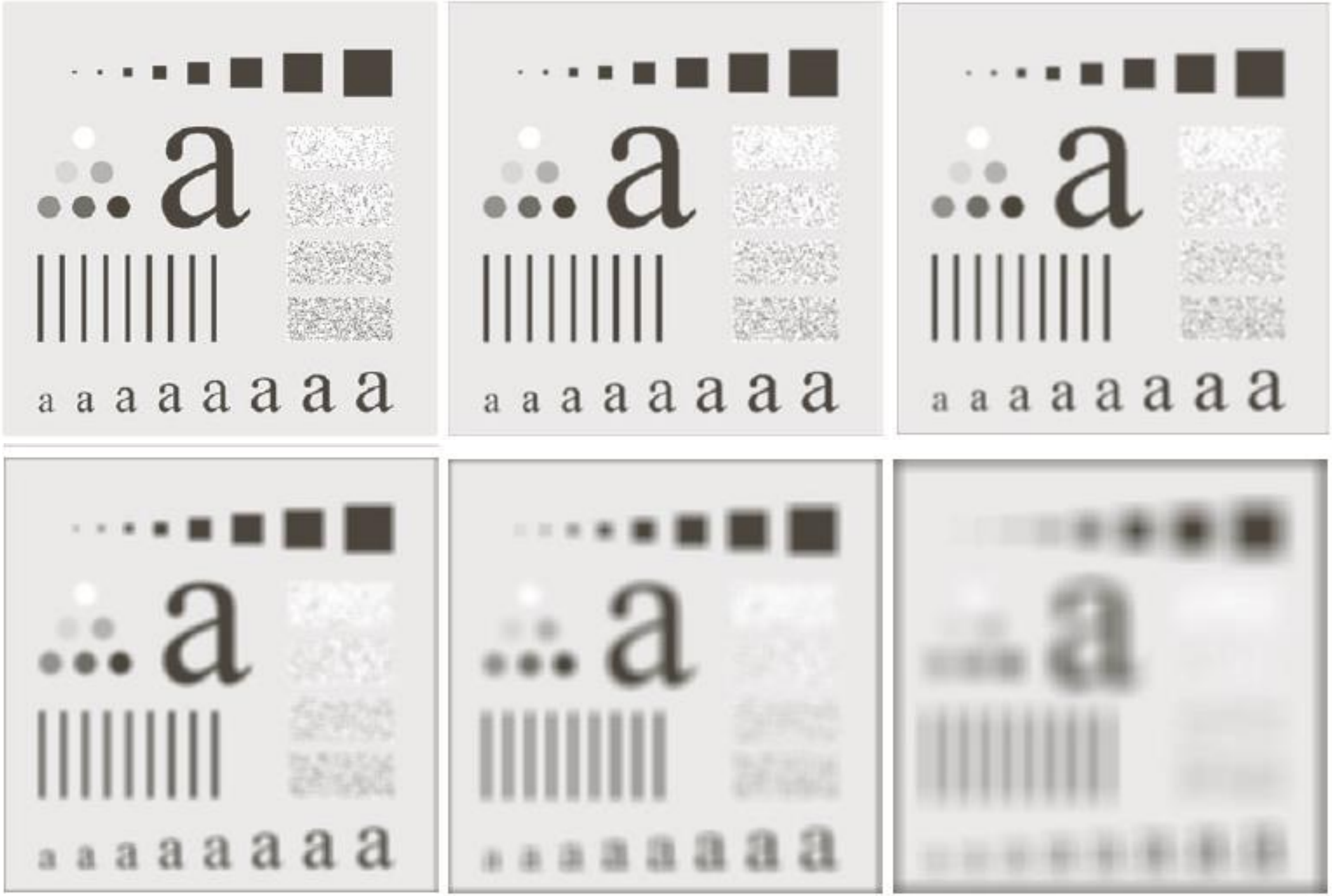
- Blurring – for preprocessing tasks
- Noise deduction
  - Linear filter : average filtering – lowpass filter in frequency domain
  - Nonlinear filter

Smooth Filters (平滑滤波器)

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

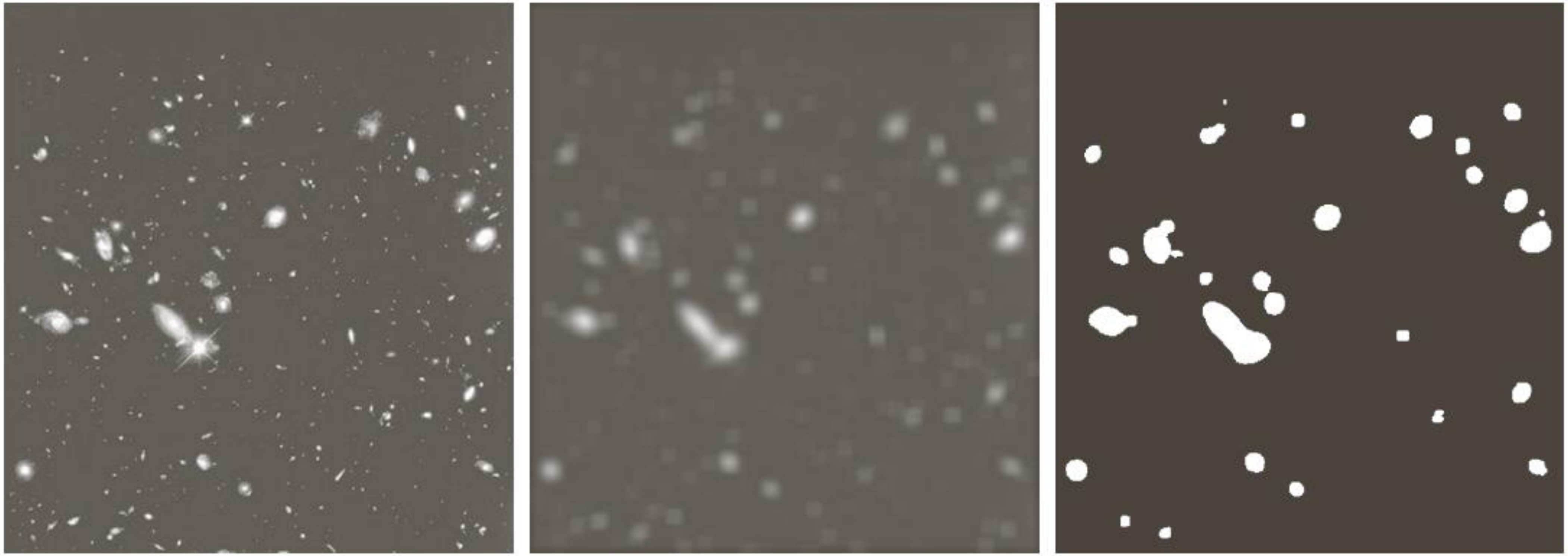
$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Filter size



上海科技大学
ShanghaiTech University

Smooth Filter and Thresholding(阈值处理)



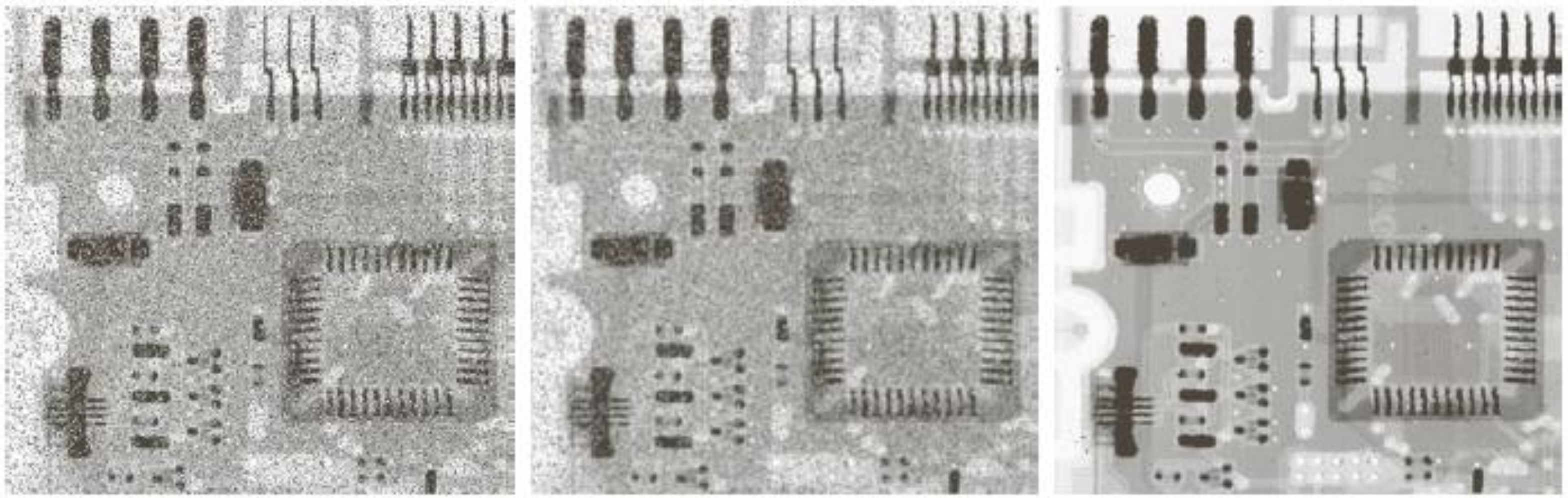
Nonlinear Smooth Filters

- **Order-statistic filter (统计排序滤波器)**

Ex：median filter （中值滤波器）

$g(x,y) \ = \ median\{m \times n$ pixel neighbouring around $I(x,y)\}$

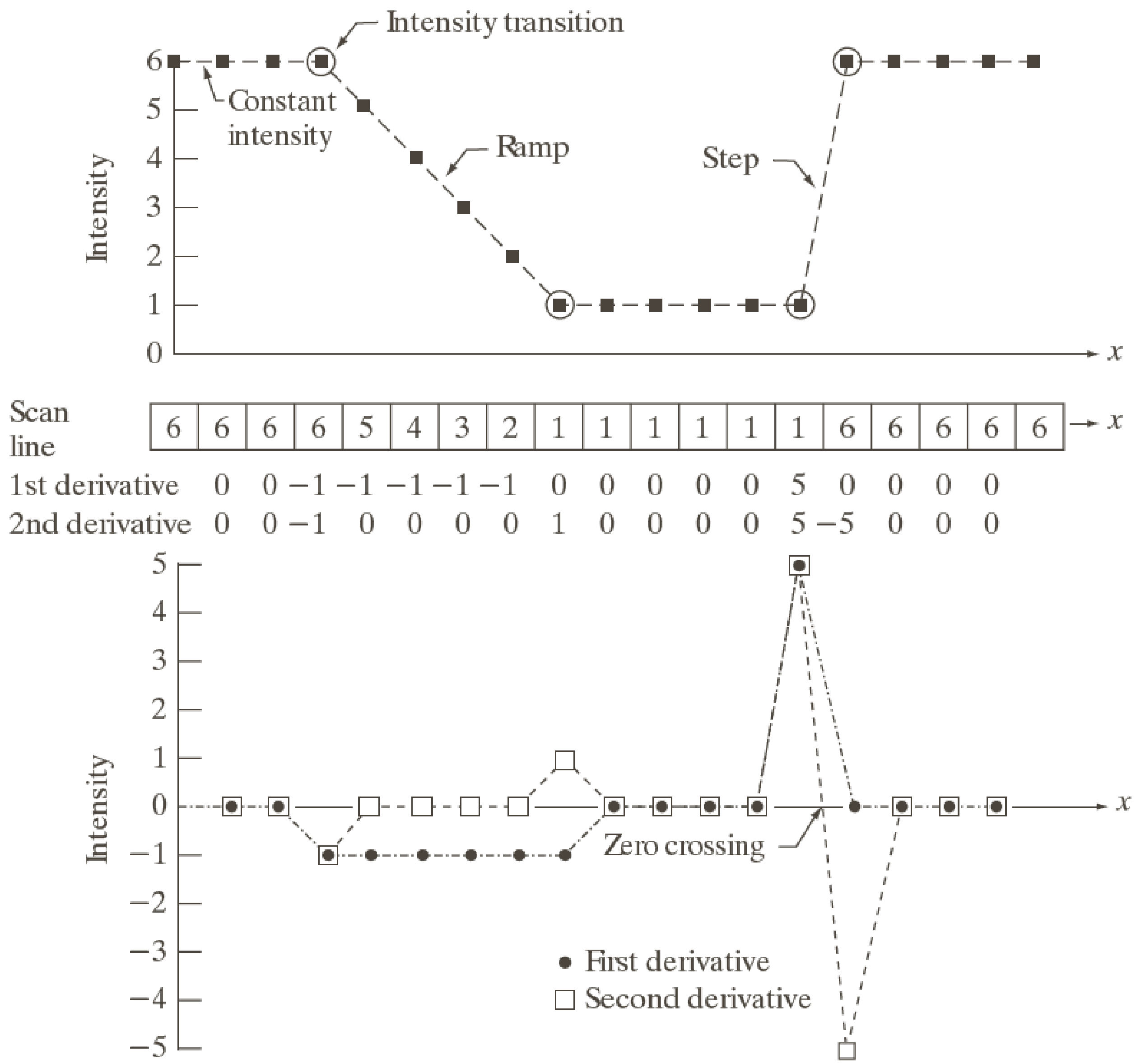
50	48	46	42
52	0	50	48
46	47	255	40
51	48	46	42



Sharpening Filter

- ➢ **Spatial differentiation（空间微分）**
- ➢ **Sharpening filter**
  - **Laplacian filtering（拉普拉斯算子）**

Derivative



Intensity transition
Constant intensity
Ramp
Step
Intensity
x
Scan line
6 6 6 6 5 4 3 2 1 1 1 1 1 1 6 6 6 6 6
1st derivative 0 0 −1 −1 −1 −1 −1 0 0 0 0 0 5 0 0 0 0
2nd derivative 0 0 −1 0 0 0 0 1 0 0 0 0 5 −5 0 0 0
Zero crossing
First derivative
Second derivative

Sharpening Filter

1. Zero in area of constant intensity

2. Nonzero at the onset of intensity step or ramp

3. (1) Nonzero along intensity ramp – 1^{st} order derivative

   (2) Zero along intensity ramp with constant slope – 2^{nd} order derivative

Sharpening Filter

- To highlight transitions in intensity
- Accomplished by spatial differentiation
  - First-order derivative: $\frac{\partial f}{\partial x} = f(x+1) - f(x)$
  - Second-order derivative: $\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$

Laplacian(拉普拉斯算子)

For an image function $\boldsymbol{f(x,y)}$,

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

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

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

$$= f(x,y+1) + f(x,y-1) + f(x+1,y) + f(x-1,y) - 4f(x,y)$$

Laplacian Filter Masks

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

Laplacian Filter Masks

$$\boldsymbol{g(x,y) = f(x,y) + c\nabla^2 f(x,y)}, \qquad \textbf{where } \boldsymbol{c = \pm 1}$$

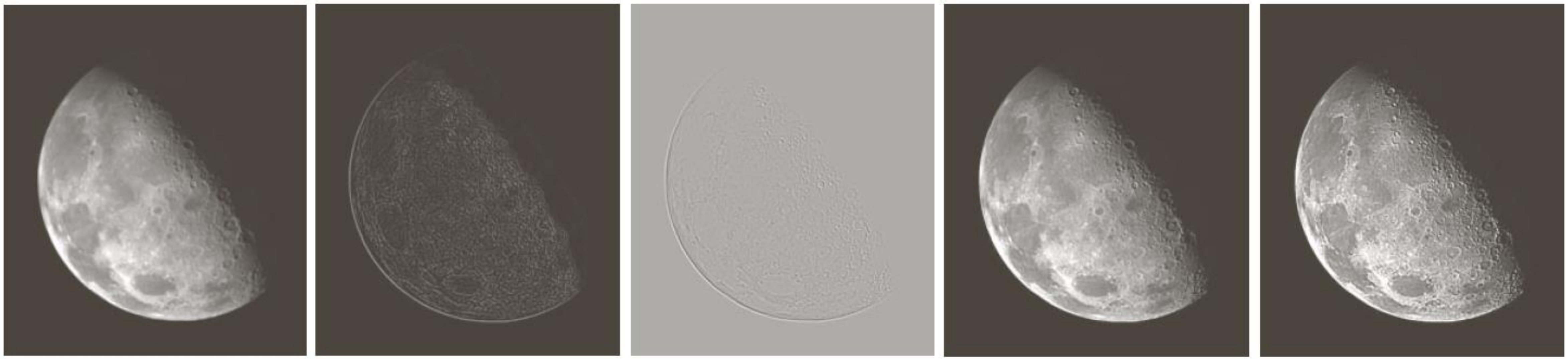
	1	
1	-4	1
	1	

$$\boldsymbol{c = -1}$$

	-1	
-1	4	-1
	-1	

$$\boldsymbol{c = 1}$$

Image Sharpening with Laplacian



Implementations in matlab

- **Low pass filter example:**

```
>> LP = 1/9 *[1,1,1;1,1,1;1,1,1];
>> im3 = imfilter(com,LP);
>>figure; imshow(im3,[]);
```

- **Sharpening filter example:**

```
>> f3 = [-1,-1,-1; -1,8,-1;-1,-1,-1];
>> J1 = imfilter(im,f1);
>>figure; imshow(J1,[]);
```

- **Median filter example:**

```
>> J2 = medfilt2(im,[3 3]);
>> J4 = medfilt2(im,[6 6]);
>> J3 = medfilt2(im,[11 1]);
>> J5 = medfilt2(im,[1 11]);
```

Take home message

- For image processing, the spatial domain processing is similar to 1-D signal processing in time domain.
- The spatial filter we discussed in this lecture is actually the calculation of correlation between the image and the filter. When using a diagnose symmetric filter, it is equivalent to convolution.
- Common spatial filters include smoothing filter and sharpening filter.