Lecture 4
Intensity transformation & Spatial Filtering (2)

Dr. Xiran Cai

Email: caixr@shanghaitech.edu.cn

Office: 3-438 SIST

Tel: 20684431

ShanghaiTech University

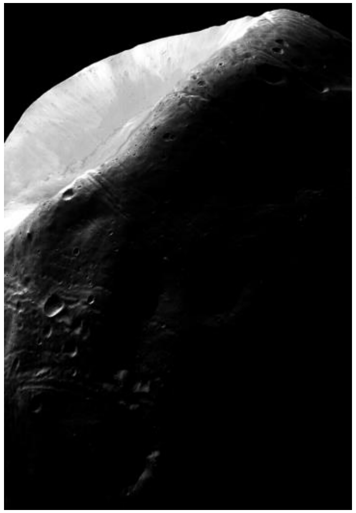
上海科技大学
ShanghaiTech University

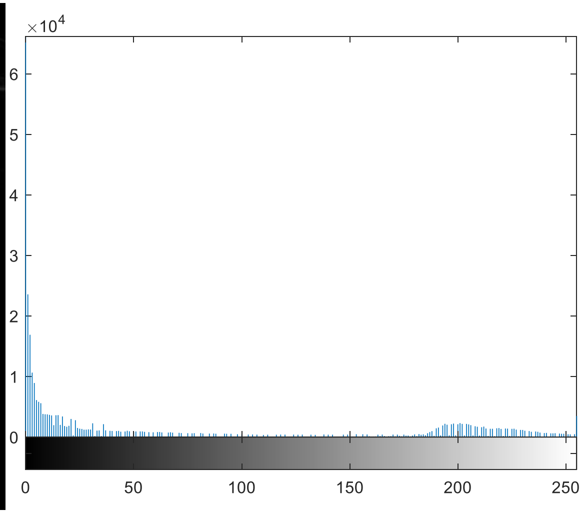
Intensity transform (2)

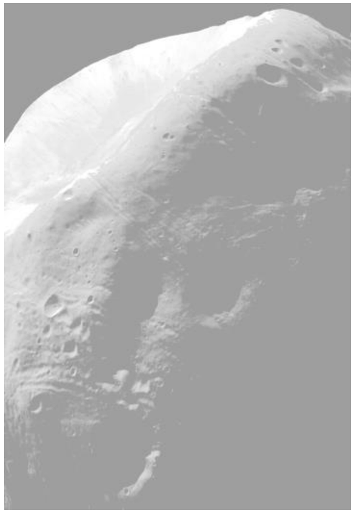
- Adaptive Histogram Equalization (AHE)
- Contrast Limited Adaptive Histogram Equalization (CLAHE)

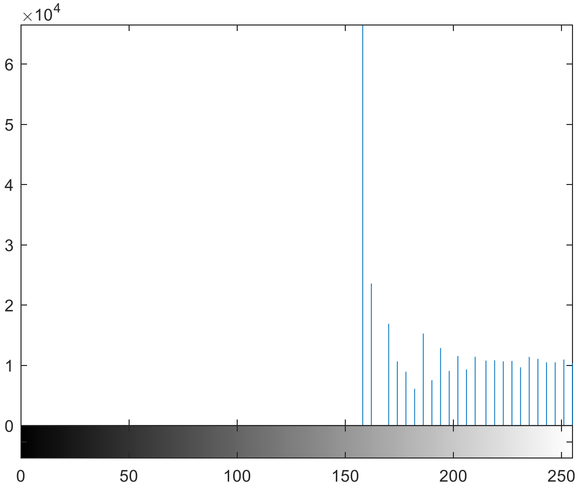
Key problem of HE

$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw = (L-1)\sum_{j=0}^{k} p_r(r_{\mathrm{j}}) = (L-1)\sum_{j=0}^{k}\frac{n_j}{MN} = \frac{L-1}{MN}\sum_{j=0}^{k} n_j$$

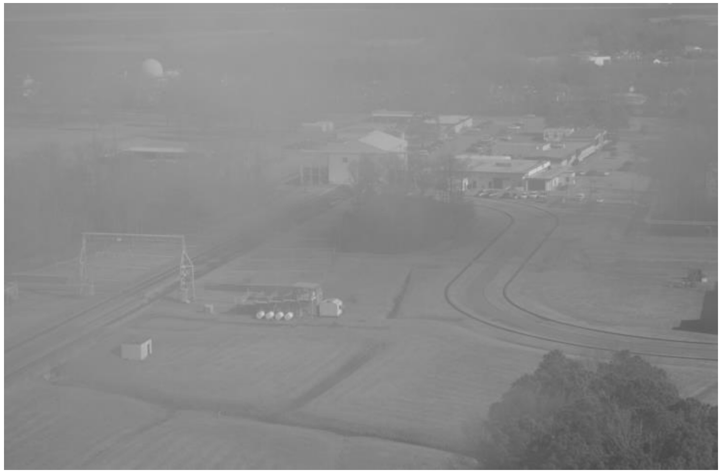


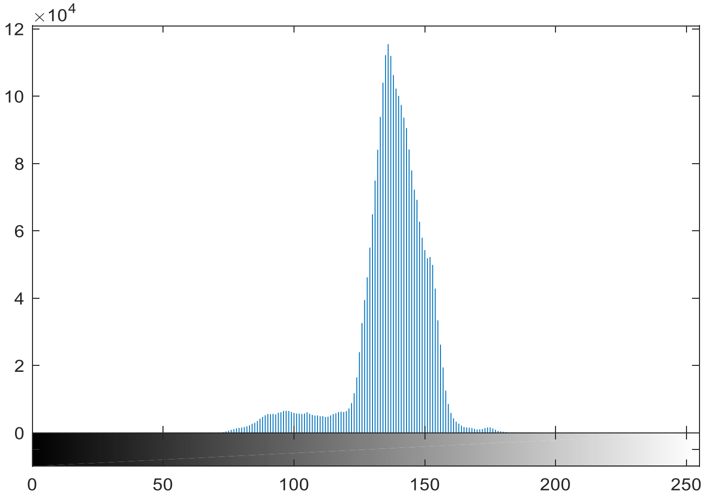


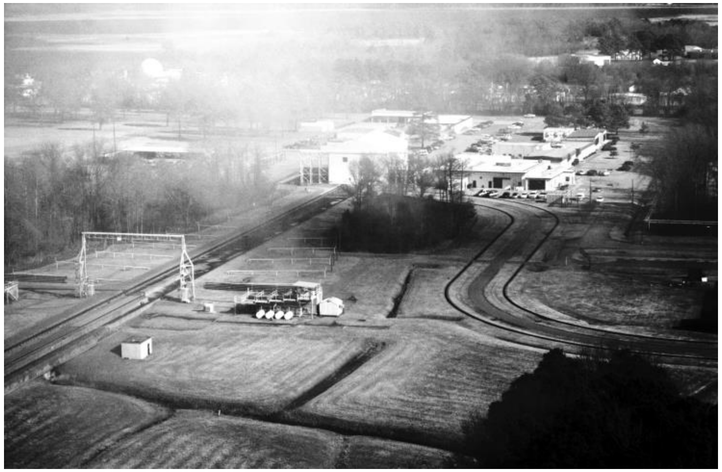


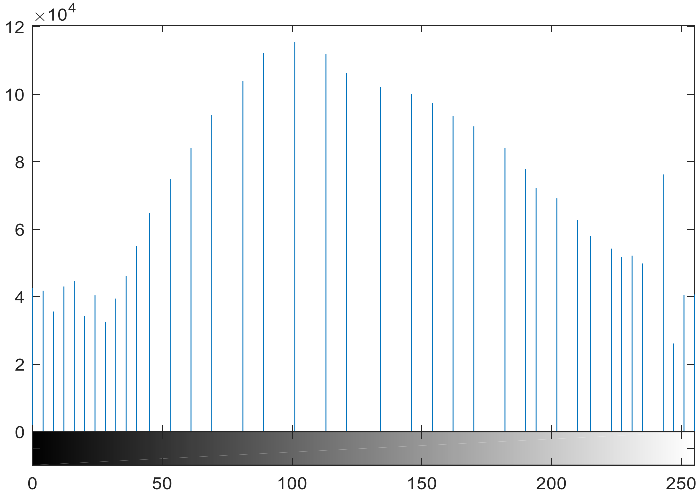


Key problem of HE







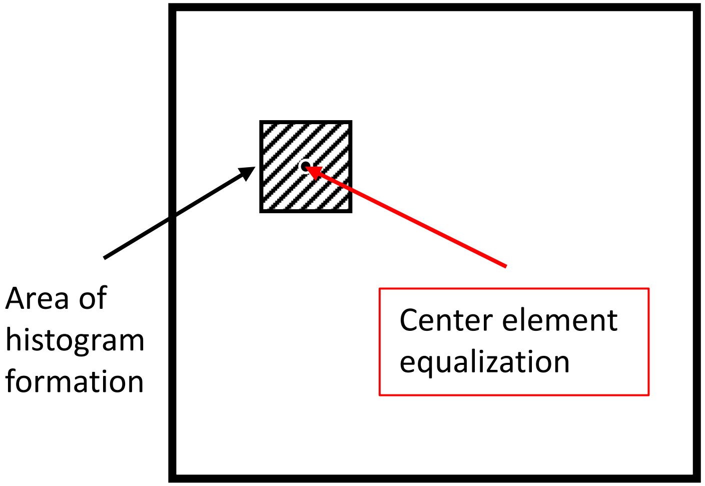


Adaptive Histogram Equalization

- Traverse every pixel with a $W * W$ patch, process histogram equalization within each patch and update the center pixel.
- Advantage: better uniform distributed histogram.
- Disadvantage： high complexity

$O(W * W + L)$ within each patch

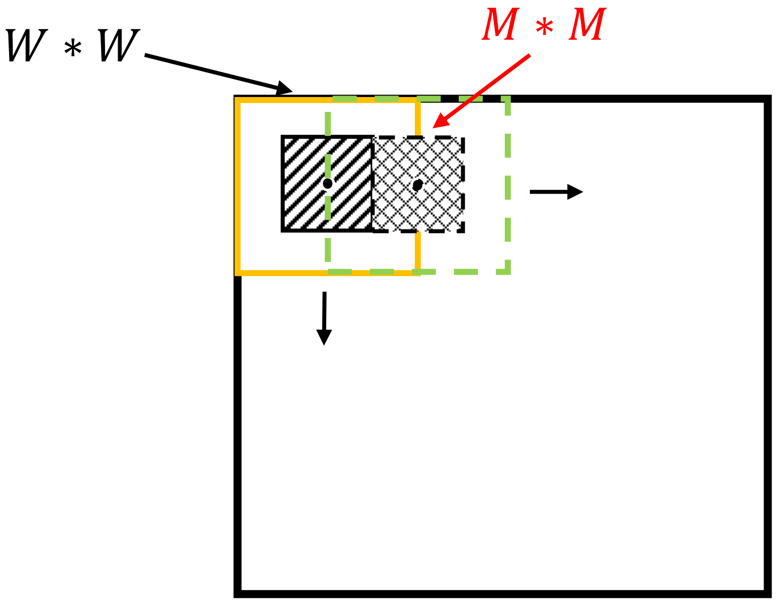
$O(M * N * (W * W + L))$ for whole image



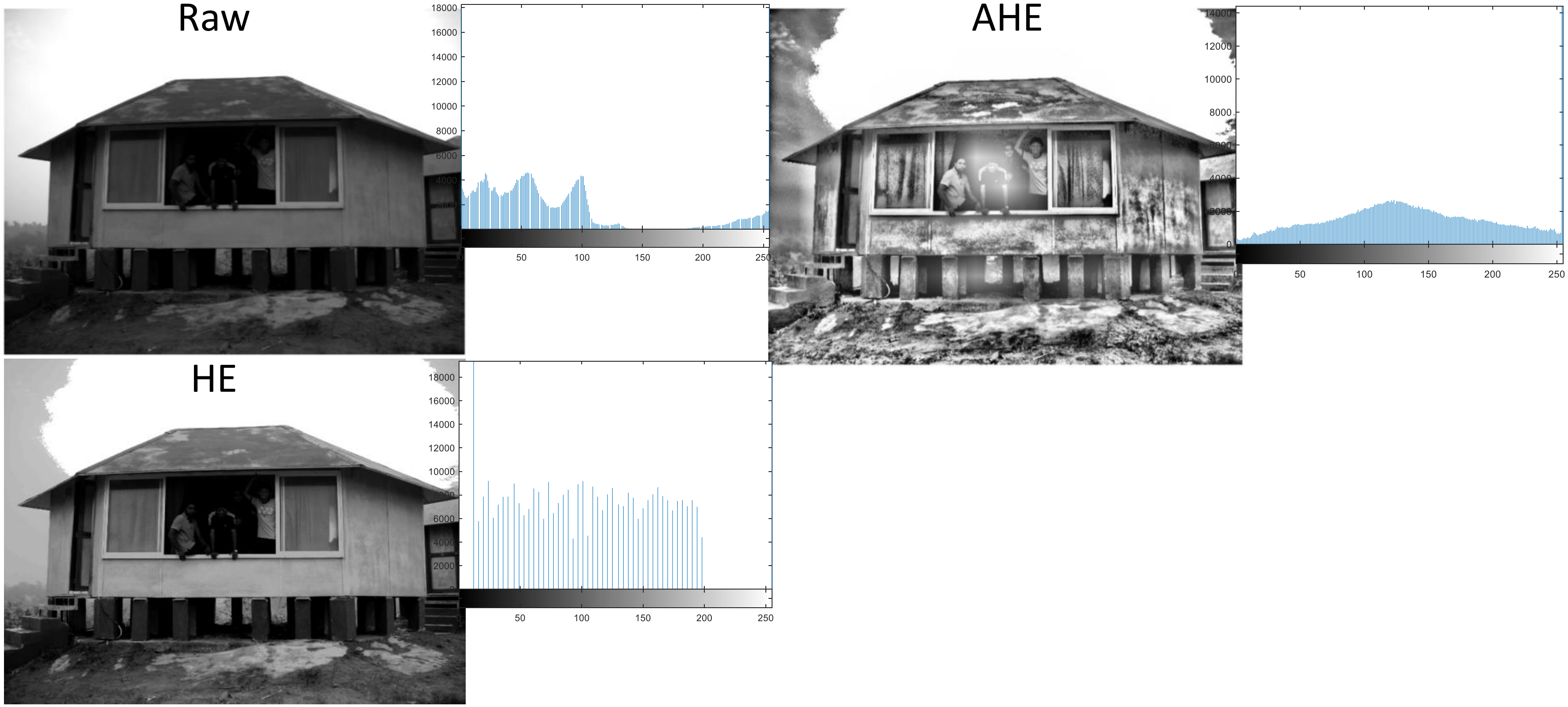
Adaptive Histogram Equalization

- For faster processing AHE, it is proposed to update a center patch of size $M * M$ instead of just the center pixel in each HE in each within the $W * W$ patch HE.

- Pixels near the image boundary have to be treated specially, This can be solved by extending the image by mirroring pixel lines and columns with respect to the image boundary.



Effect of AHE

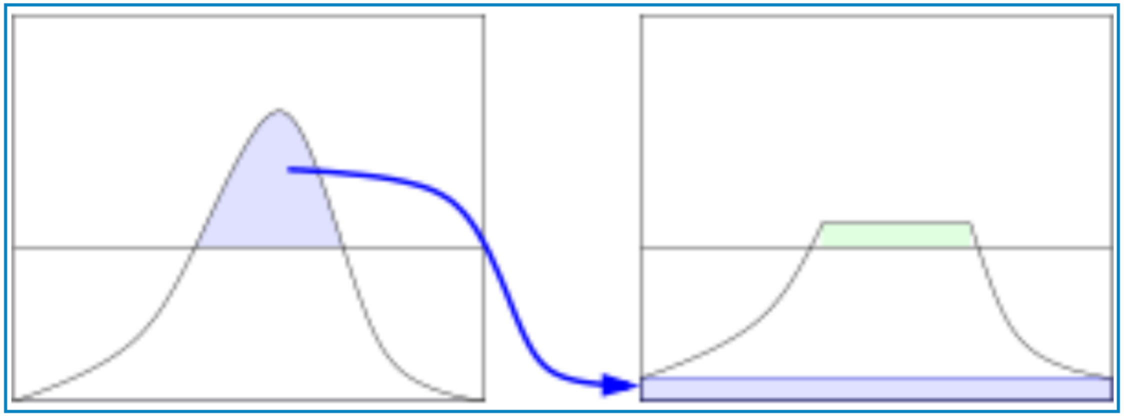


Contrast Limited Adaptive Histogram Equalization (CLAHE)

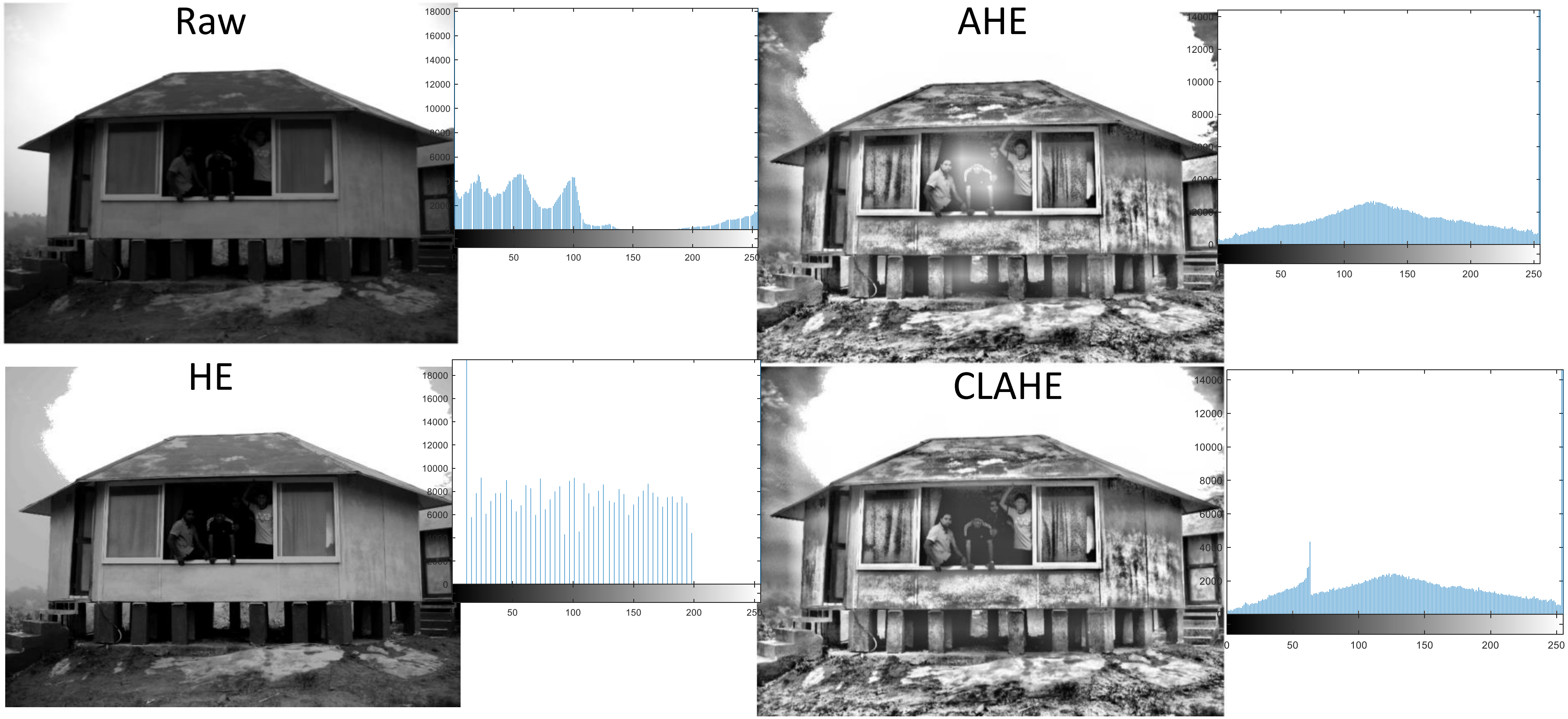
- ❑ CLAHE differs from naive AHE in its contrast limiting.
- ❑ CLAHE was developed to prevent the over amplification of noise that AHE can give rise to.
- ❑ This feature can also be applied to global histogram equalization, giving rise to contrast limited histogram equalization.

CLAHE

- CLAHE limits the amplification by clipping the histogram at a predefined value before computing the CDF.
- This limits the slope of the CDF and therefore of the transformation function.
- The so-called clip limit depends on the normalization of the histogram and thereby on the size of the neighborhood region.

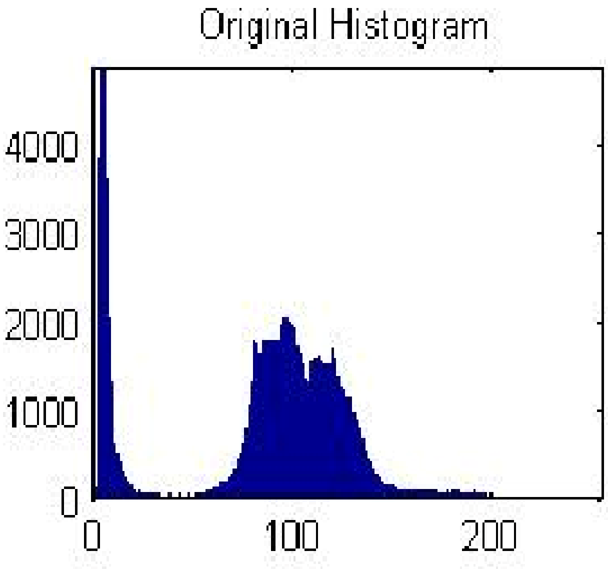


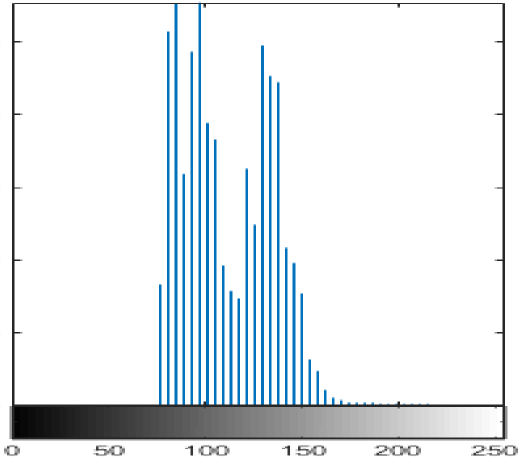
CLAHE

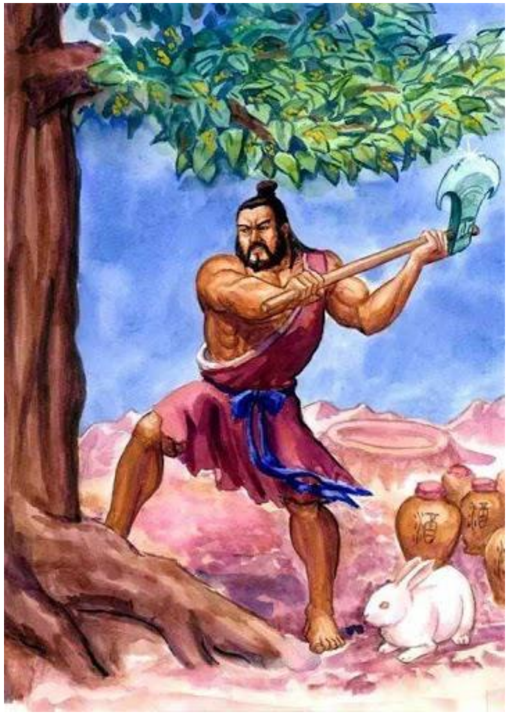

Raw
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
50
100
150
200
250
AHE
14000
12000
10000
8000
6000
4000
2000
0
50
100
150
200
250
HE
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
50
100
150
200
250
CLAHE
14000
12000
10000
8000
6000
4000
2000
0
50
100
150
200
250

Take home message

- ❑ **Key idea: AHE&CLAHE was developed to prevent the over amplification of noise.**



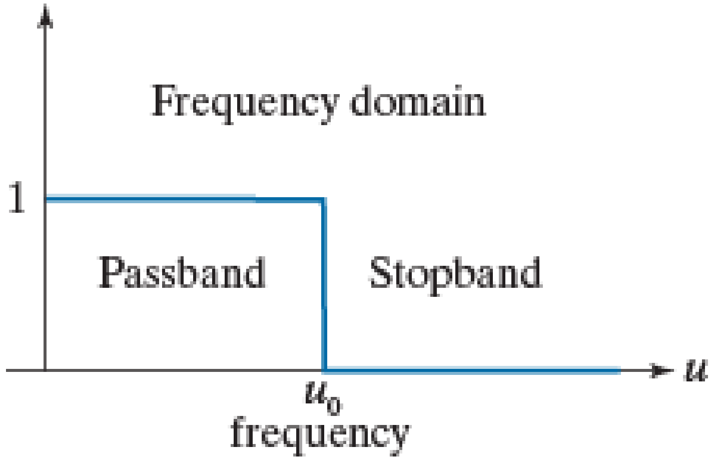


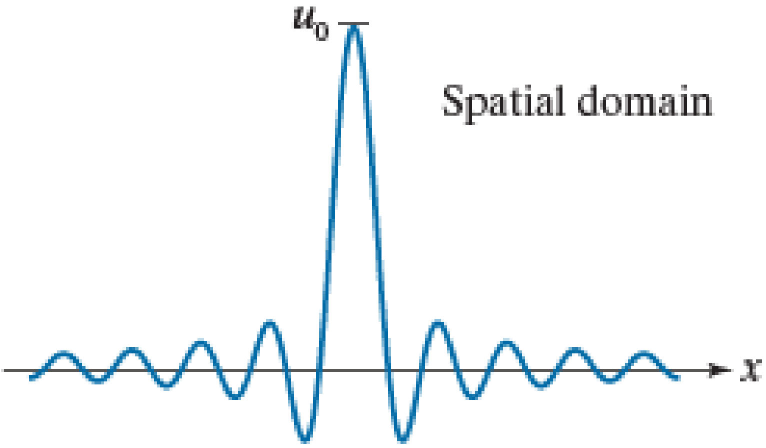


Spatial filtering (2)

- ❑ Some other perspectives on spatial filtering
- ❑ Sobel Filter
- ❑ Unsharpen Filter (非锐化掩蔽)
- ❑ LoG Filter
  - ➢- useful for finding edges
  - ➢- also useful for finding blobs

Filtering in frequency domain and spatial domain





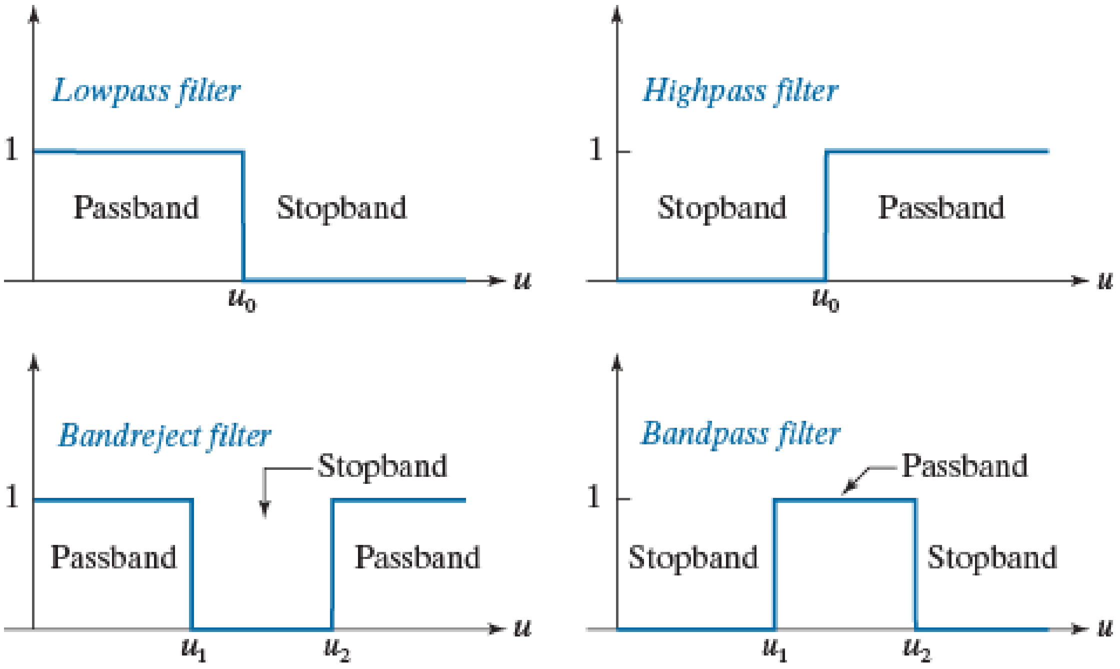
Ideal 1D LP filter

Ideal 1D LP filter kernel in spatial domain

Q: Is ideal filter really ideal for image processing?

Filtering in frequency domain and spatial domain

- 4 types of filters

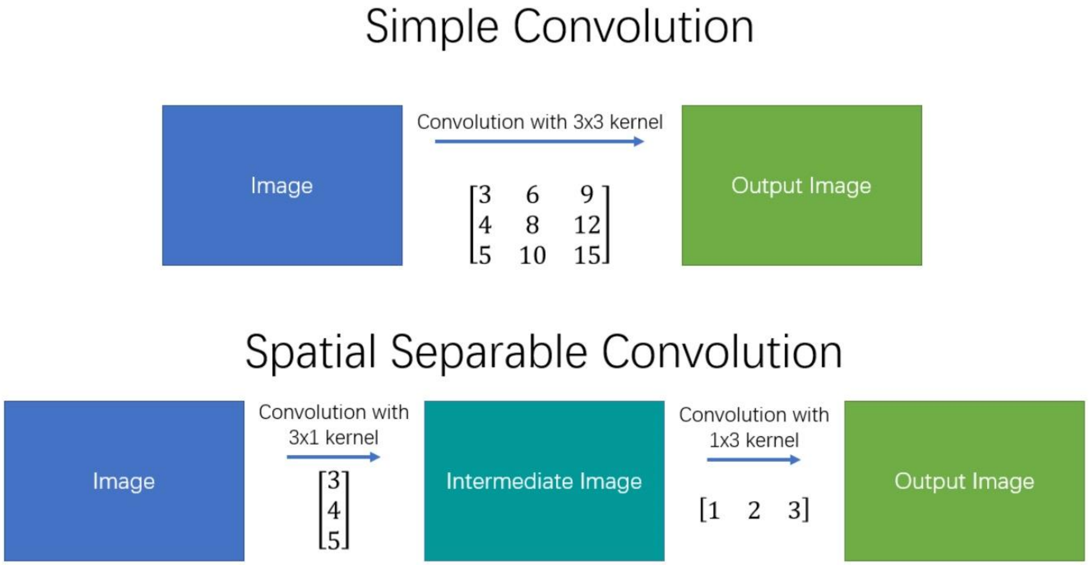


Separable filter kernels

- Example: $\begin{bmatrix} 3 & 6 & 9 \\ 4 & 8 & 12 \\ 5 & 10 & 15 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$

$$\boldsymbol{w} = \boldsymbol{a}\boldsymbol{b}^T$$

$$\boldsymbol{w} \star \boldsymbol{f} = (\boldsymbol{w}_1 \star \boldsymbol{w}_2) \star \boldsymbol{f} = (\boldsymbol{w}_2 \star \boldsymbol{w}_1) \star \boldsymbol{f} = \boldsymbol{w}_2 \star (\boldsymbol{w}_1 \star \boldsymbol{f}) = (\boldsymbol{w}_1 \star \boldsymbol{f}) \star \boldsymbol{w}_2$$



Simple Convolution

Image $\xrightarrow{\text{Convolution with 3x3 kernel}}$ Output Image

$\begin{bmatrix} 3 & 6 & 9 \\ 4 & 8 & 12 \\ 5 & 10 & 15 \end{bmatrix}$

Spatial Separable Convolution

Image $\xrightarrow{\text{Convolution with 3x1 kernel}}$ Intermediate Image $\xrightarrow{\text{Convolution with 1x3 kernel}}$ Output Image

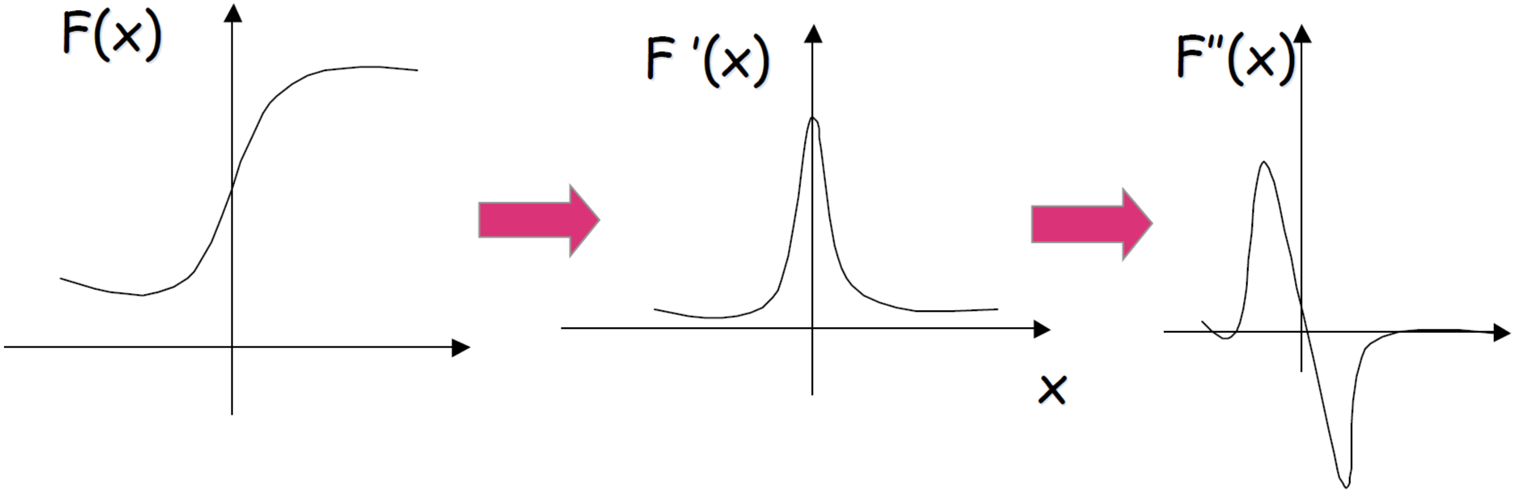
$\begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix}$ $\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$

Computational advantage:

$$\boldsymbol{C} = \frac{\boldsymbol{MNmn}}{\boldsymbol{MN(m+n)}} = \frac{\boldsymbol{mn}}{\boldsymbol{(m+n)}}$$

Recall: First & Second-Derivative filters

- Sharp changes in gray level of the input image corresponds to “peaks or valleys” of the first-derivative of the input signal.
- Peaks or valleys of the first derivative of the input signal, correspond to “zero-crossings” of the second-derivative of the input signal.



Laplacian(拉普拉斯算子)

For an image function $f(x, y)$,

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

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

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(拉普拉斯算子)

For an image function $f(x,y)$,

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

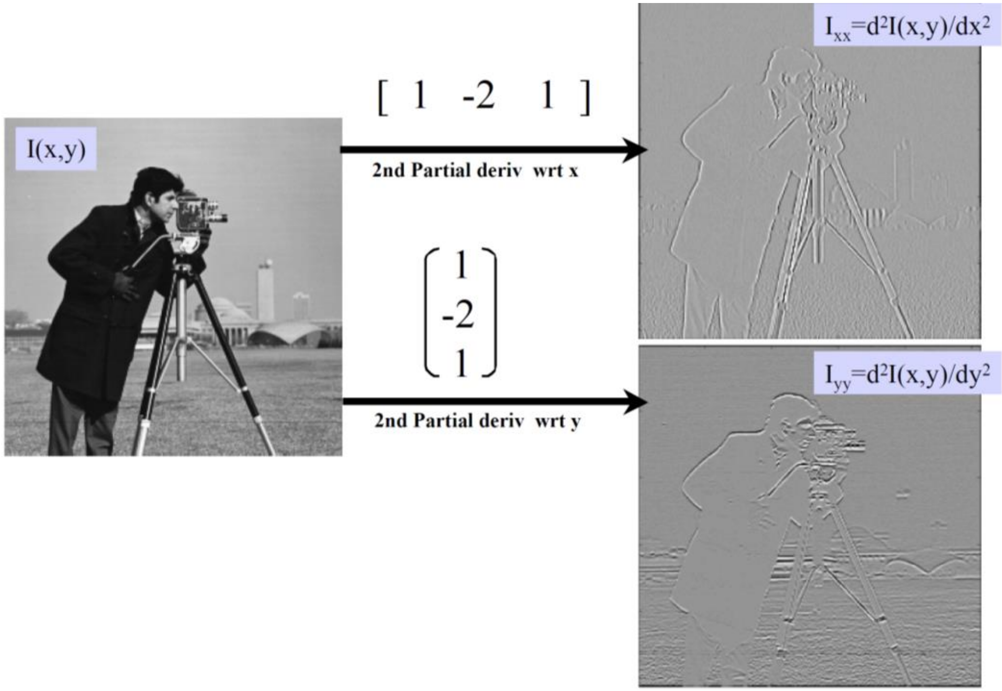
Y direction： $\frac{\partial^2 f}{\partial y^2} = f(x,y+1) + f(x,y-1) - 2f(x,y)$

I_{xx}

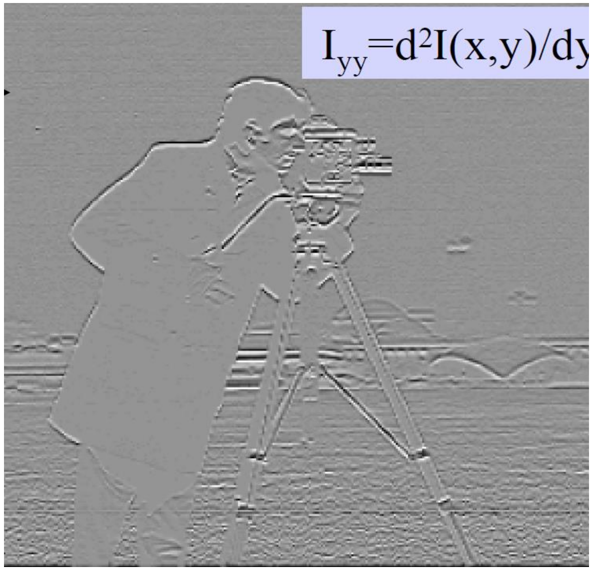
1	-2	1

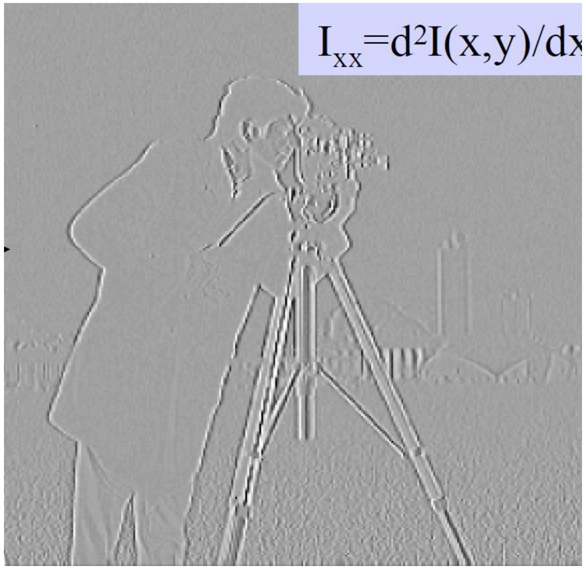
I_{yy}

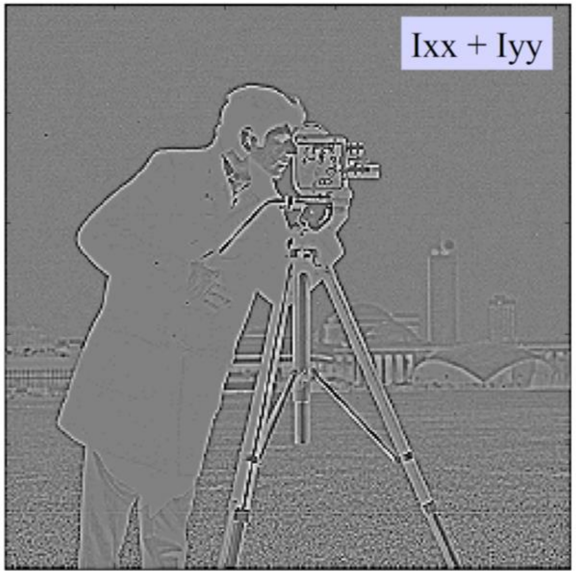
1
-2
1



Laplacian







Gradient(梯度)

The first-order derivative of $f(x, y)$: $\boldsymbol{\nabla f} \equiv \mathbf{grad}(\boldsymbol{f}) \equiv \begin{cases} \boldsymbol{g_x} \\ \boldsymbol{g_y} \end{cases} = \begin{cases} \dfrac{\boldsymbol{\partial f}}{\boldsymbol{\partial x}} \\ \dfrac{\boldsymbol{\partial f}}{\boldsymbol{\partial y}} \end{cases}$

The amplitude： $\boldsymbol{M}(\boldsymbol{x}, \boldsymbol{y}) = \mathbf{mag}(\boldsymbol{\nabla f}) = \sqrt{\boldsymbol{g_x^2} + \boldsymbol{g_y^2}}$

$$\boldsymbol{M}(\boldsymbol{x}, \boldsymbol{y}) \approx |\boldsymbol{g_x}| + |\boldsymbol{g_y}|$$

Gradient(梯度)

- **Roberts cross-gradient operator（罗伯特交叉梯度算子）**

$$M(x,y) \approx |g_x| + |g_y|$$

$$= |z_9 - z_5| + |z_8 - z_6|$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

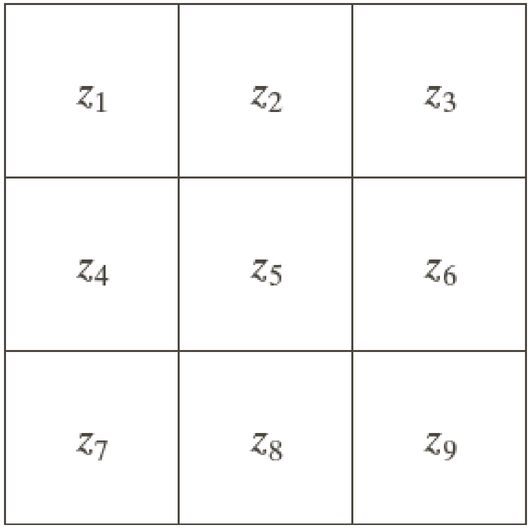
−1	0
0	1

0	−1
1	0

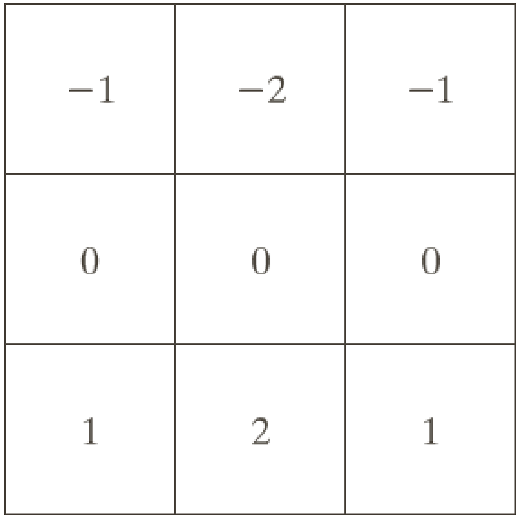
Gradient(梯度)

- **Sobel operator （Sobel算子）**

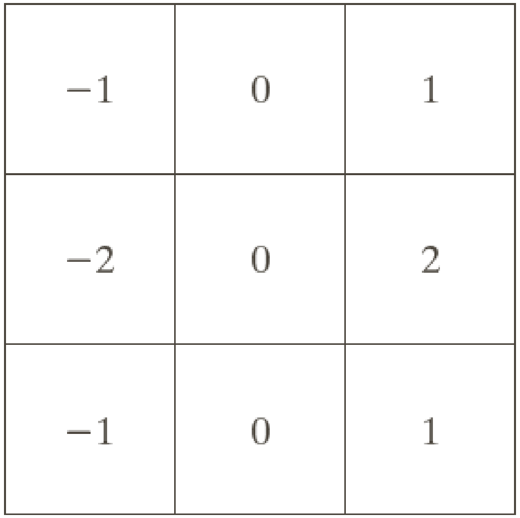
$$M(x,y) = |(z_7+2z_8+z_9) - (z_1+2z_2+z_3)| + |(z_3+2z_6+z_9) - (z_1+2z_4+z_7)|$$



z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9



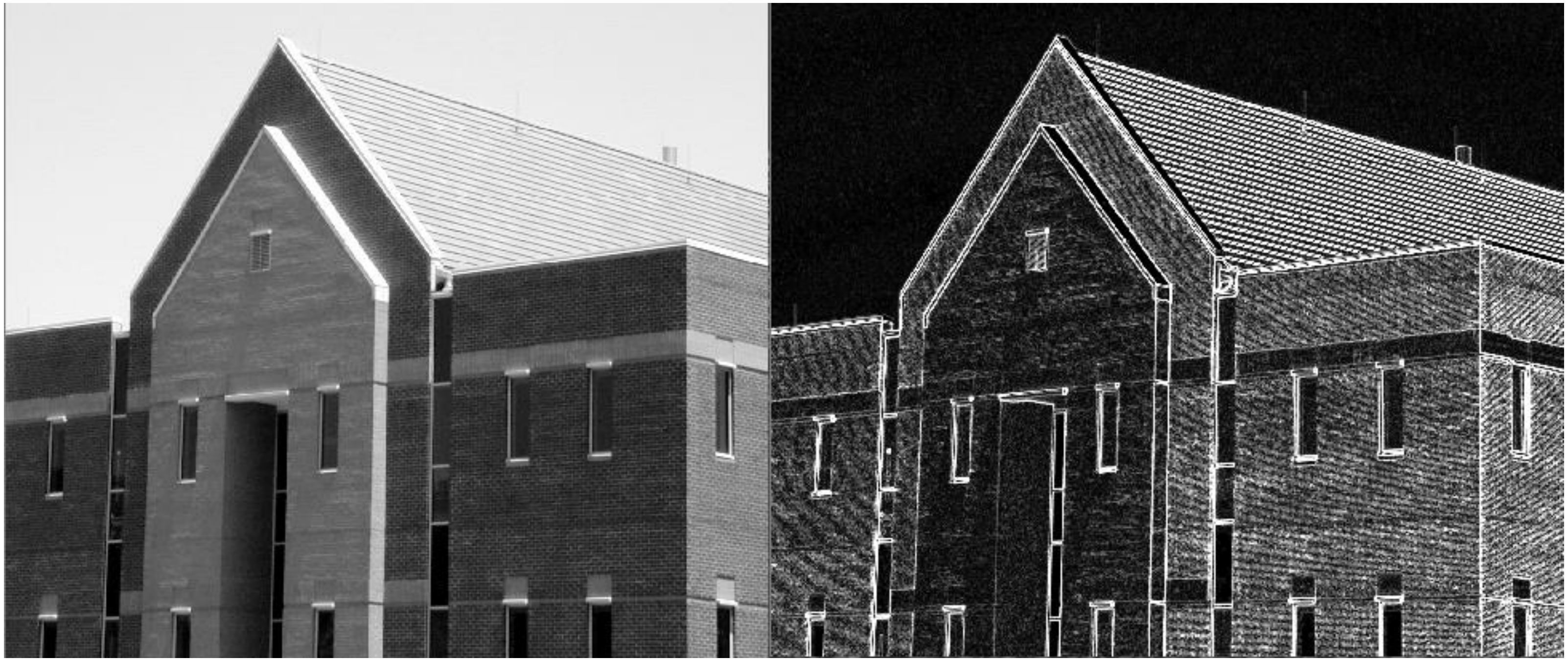
−1	−2	−1
0	0	0
1	2	1



−1	0	1
−2	0	2
−1	0	1

Q: How to really understand Sobel operator? What are the functions?

Sobel operator



More Sobel operators

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

2	−1	−1
−1	2	−1
−1	−1	2

+45°

−1	2	−1
−1	2	−1
−1	2	−1

Vertical

−1	−1	2
−1	2	−1
2	−1	−1

−45°

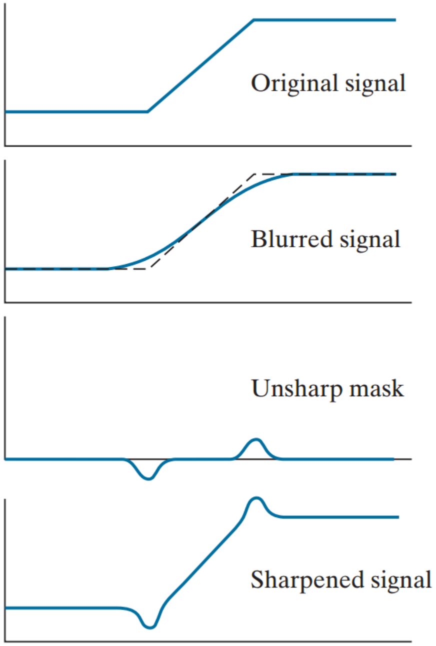
The Notes about the Laplacian

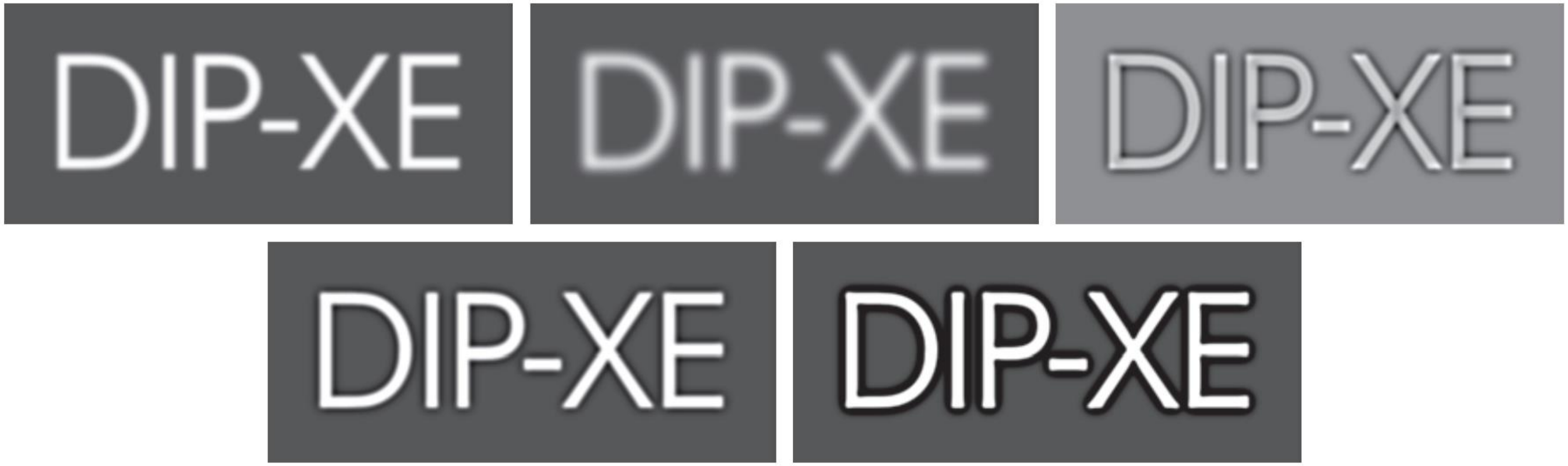
- $\nabla^2 I(x, y)$ is a SCALAR
  - ↑ Can be found using a SINGLE mask
  - ↓ Orientation information is lost
- $\nabla^2 I(x, y)$ is the sum of SECOND-order derivatives
  - But taking derivatives increases noise.
  - Very noise sensitive!
- It is always combined with a smoothing operation.

Unsharpen Mask(非锐化掩蔽)

$$g_{\text{mask}}(x,y) = f(x,y) - \overline{f(x,y)}$$

$$g(x,y) = f(x,y) + k * g_{\text{mask}}(x,y)$$





Laplacian of Gaussian (LoG) Filter

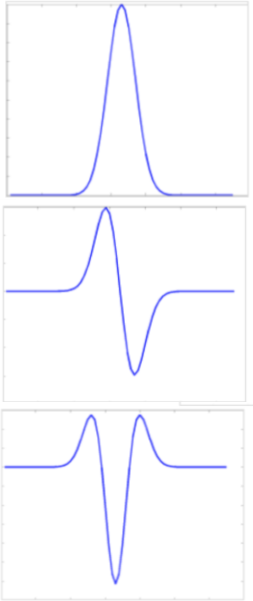
- **First smooth (Gaussian filter),**
- **Then, find zero-crossings (Laplacian filter):**

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

$$G(x,y) = e^{-\frac{x^2+\mathrm{y}^2}{2\sigma^2}}$$

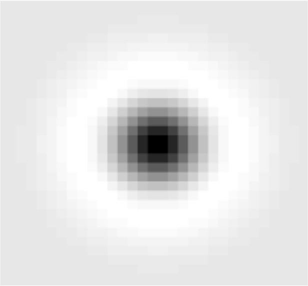
$$G'(x,y) = -\frac{1}{2\sigma^2}2(x+y)e^{-\frac{x^2+\mathrm{y}^2}{2\sigma^2}} = -\frac{x+y}{\sigma^2}e^{-\frac{x^2+\mathrm{y}^2}{2\sigma^2}}$$

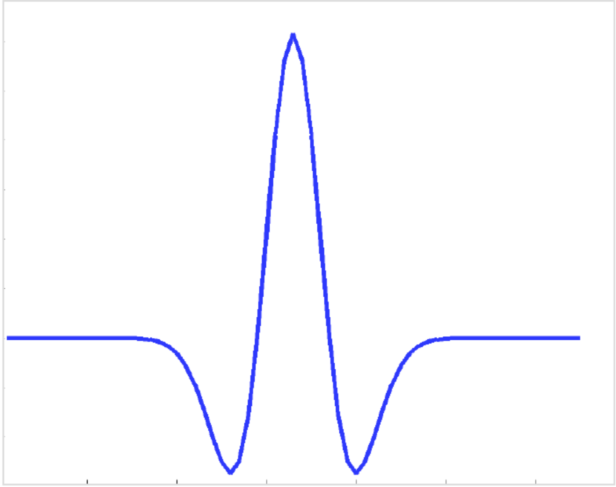
$$G''(x,y) = -\frac{1}{\pi\sigma^4}\,(1-\frac{x^2+\mathrm{y}^2}{2\sigma^2})\;e^{-\frac{x^2+\mathrm{y}^2}{2\sigma^2}}$$



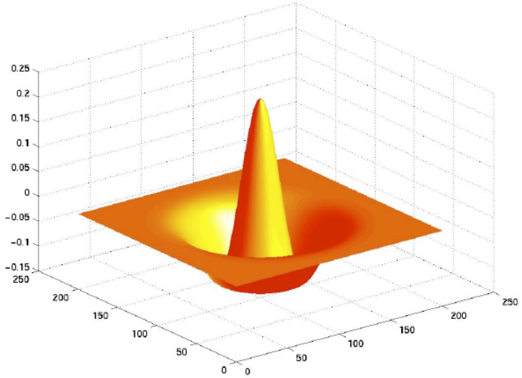
Second derivative of a Gaussian

$$G''(x,y) = -\frac{1}{\pi\sigma^4}\left(1-\frac{x^2+y^2}{2\sigma^2}\right)e^{-\frac{x^2+y^2}{2\sigma^2}}$$



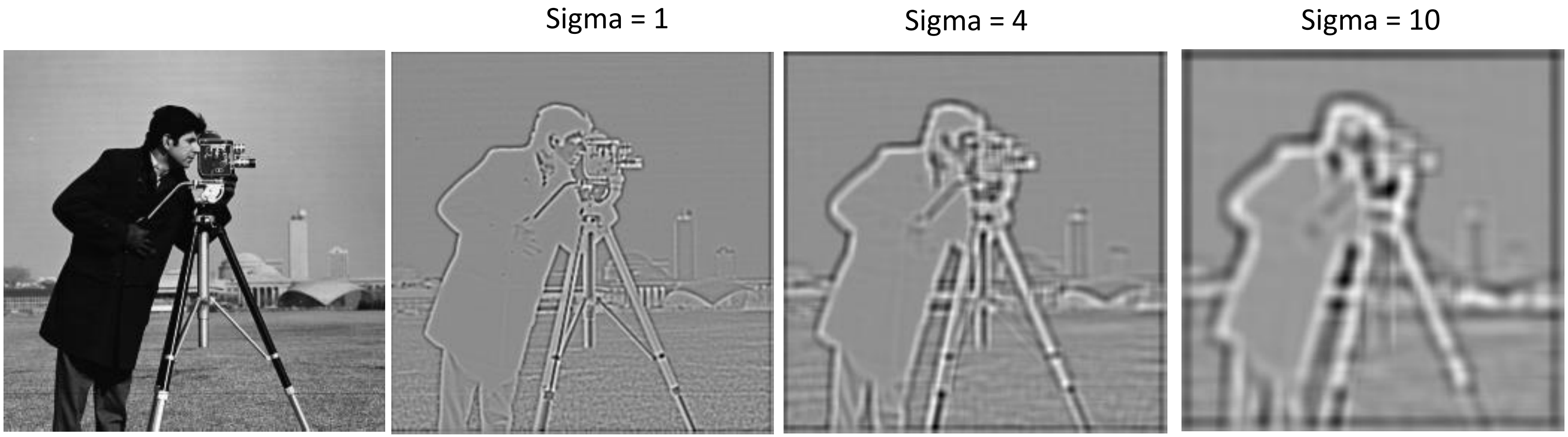


2D analog



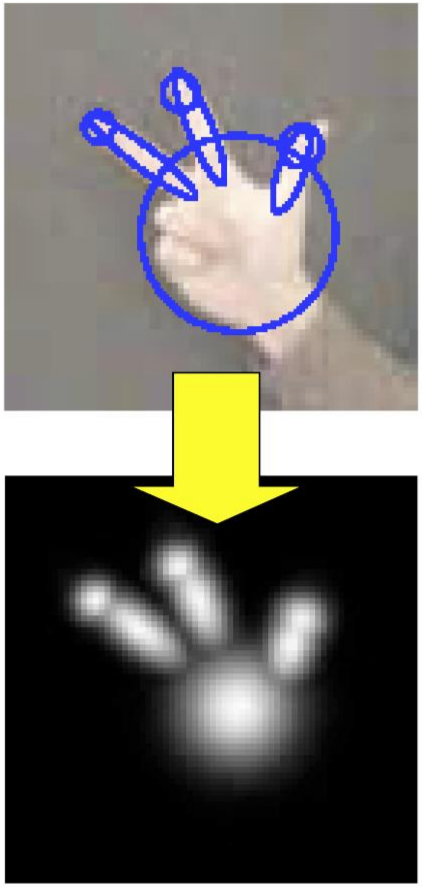
LoG "Mexican Hat"

Effect of LoG Filter



Band-Pass Filter (suppresses both high and low frequencies)

Application of LoG Filter



Gesture recognition for
the ultimate couch potato

Matlab practice: spatial filtering

```
w  =  fspecial(‘type’, parameters)
g  =  imfilter(f, w, ’replicate’)
```

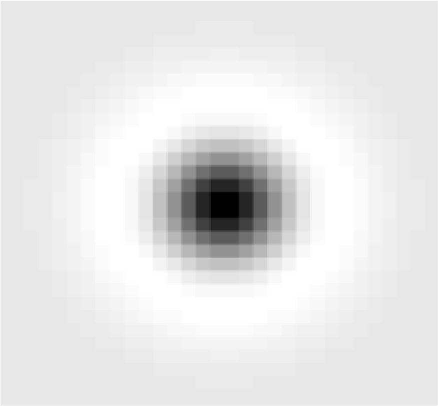
- □ See some examples.

- □ Then practice by yourself…

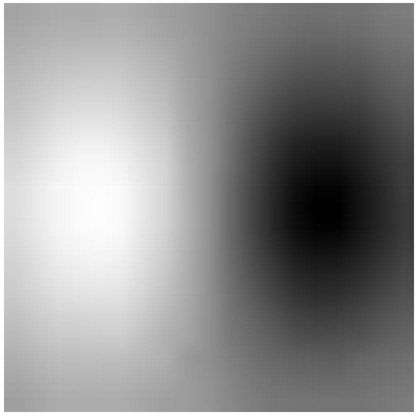
Take home message

- **Key idea: Cross correlation with a filter can be viewed as comparing a little “picture” of what you want to find against all local regions in the image.**

LoG

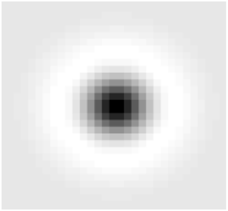


Derivative of Gaussian



Take home message

*



=