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)

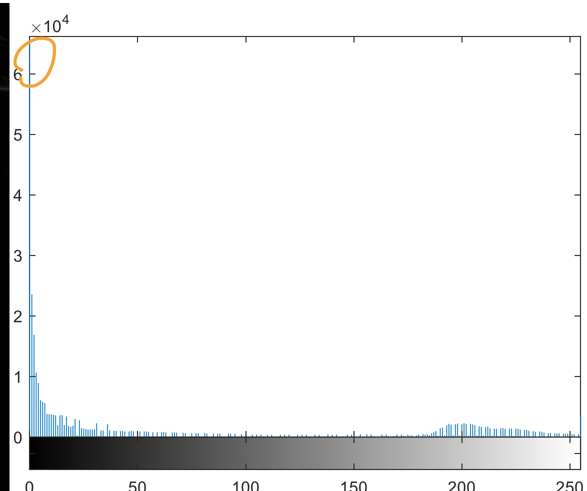
Key problem of HE

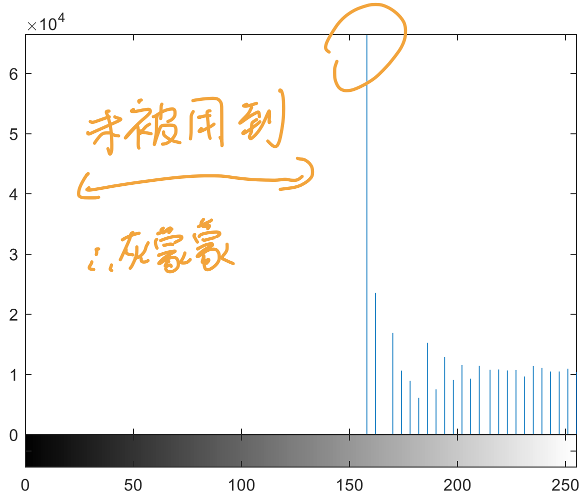
$r=0 \to s=?$

$k=0$ $\quad 255 \cdot \underbrace{p_r(r_0)}_{0.5} = 128$

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

prob: 0.5



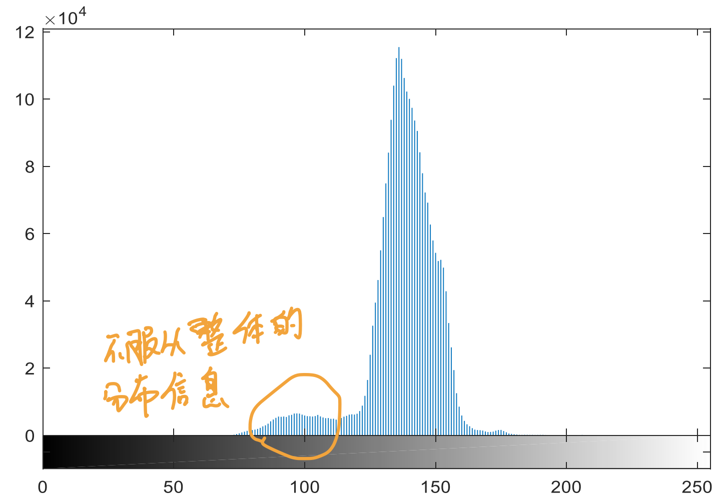


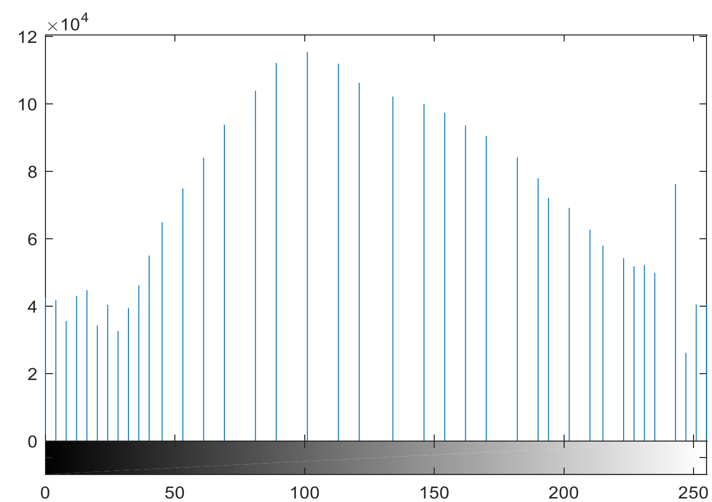
未被用到

∴灰蒙蒙

直方图均衡化

Key problem of HE





Adaptive Histogram Equalization

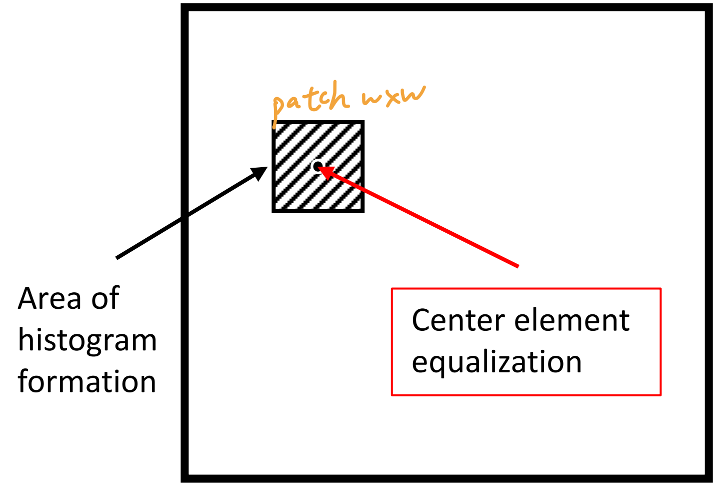
- Traverse every pixel with a $W * W$ patch, process histogram equalization within each patch and update the center pixel.

取一小块，以局部直调
分别均衡化

- Advantage: better uniform distributed histogram.
- Disadvantage： high complexity

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

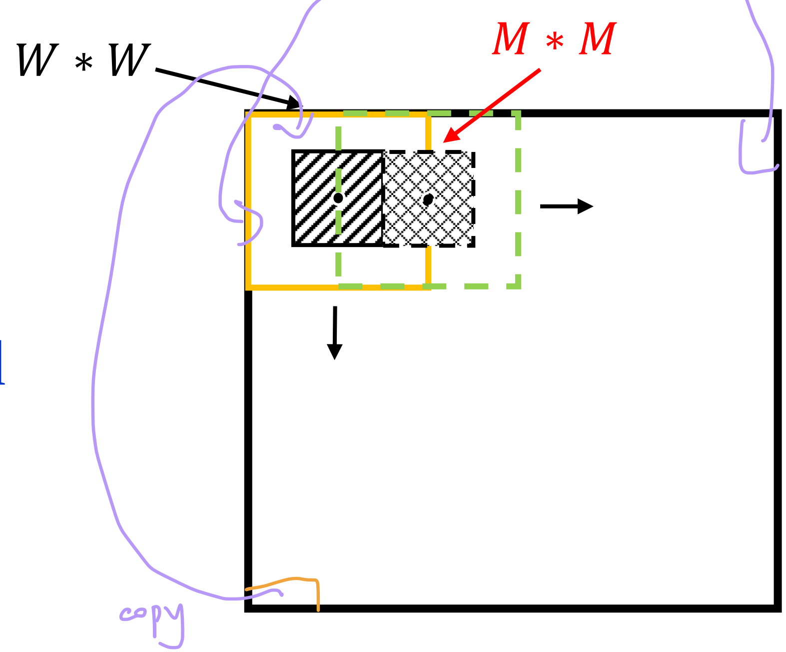
$O(M * N * (W * W + L))$ for whole image



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.

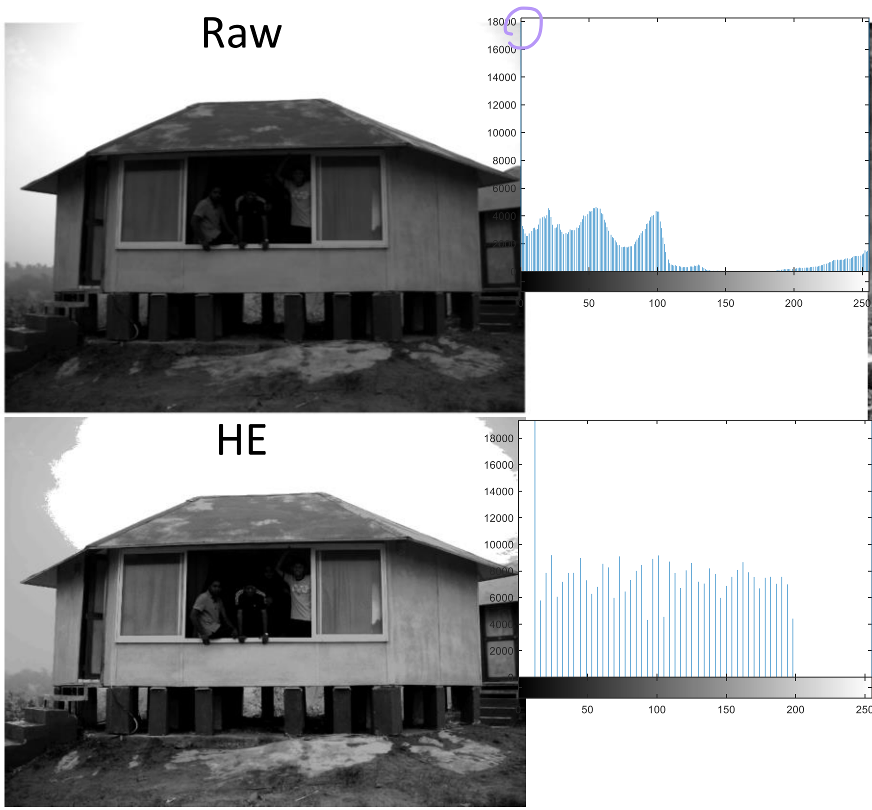
更快, may 出现不连续

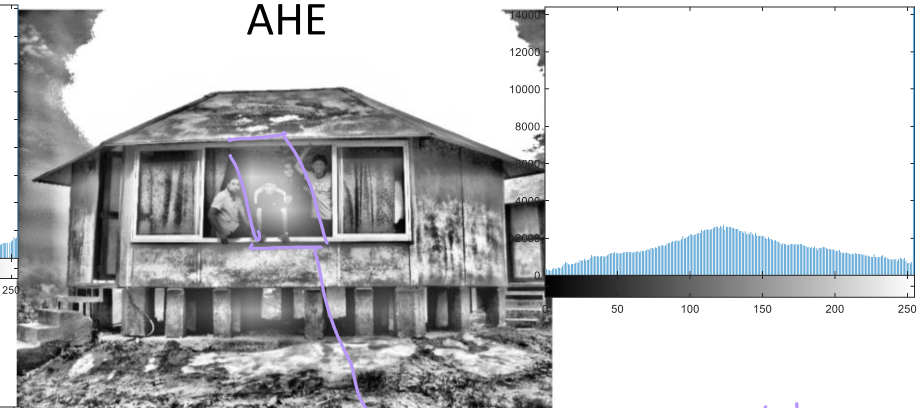


- Pixels near the image boundary have to be treated specially, This can be solved by extending the image by mirroring pixel lines and columns with respect to the image boundary.

上海科技大学 ShanghaiTech University

Effect of AHE





过亮，改进

对比度不太强

Contrast Limited Adaptive Histogram Equalization (CLAHE) 对对比度限制

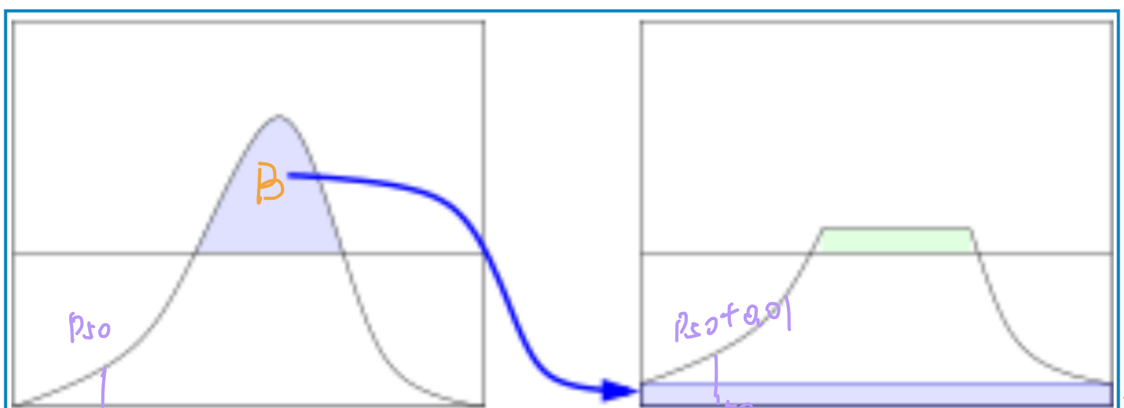
- CLAHE differs from naive AHE in its contrast limiting.
- CLAHE was developed to prevent the over amplification of noise that AHE can give rise to.
- This feature can also be applied to global histogram equalization, giving rise to contrast limited histogram equalization.

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.

B: $I_p > a$

a

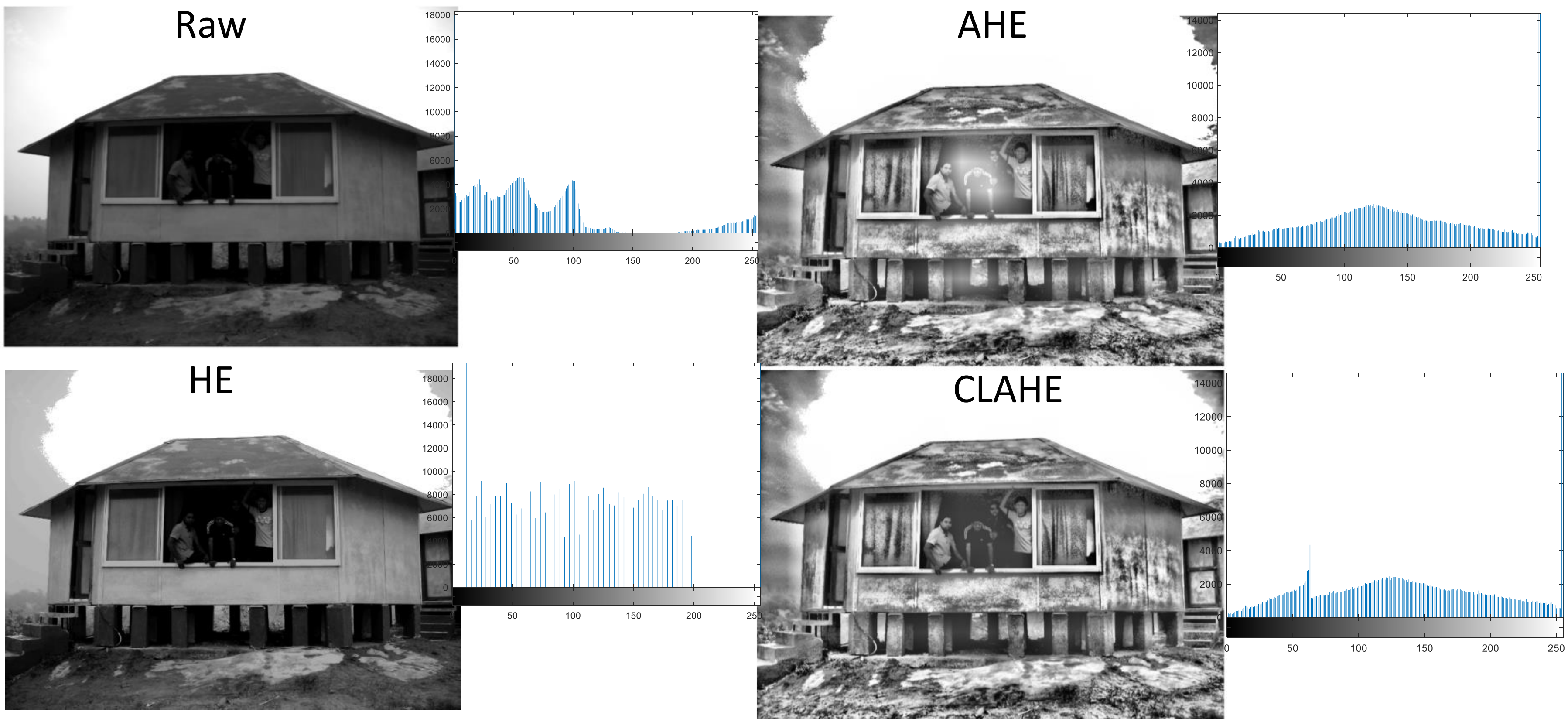


→ 均衡化

$\frac{S_B\ (\text{area } B)}{256}$

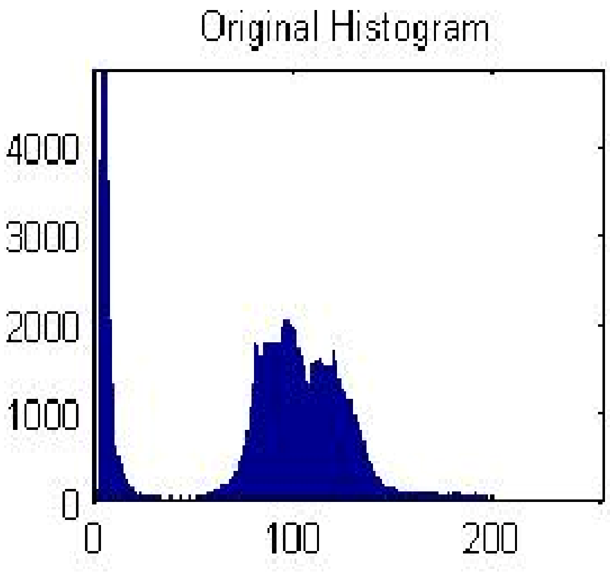
拿下来均匀加在其它prob

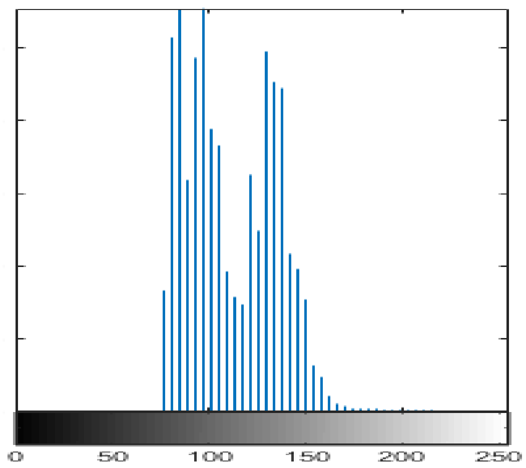
CLAHE

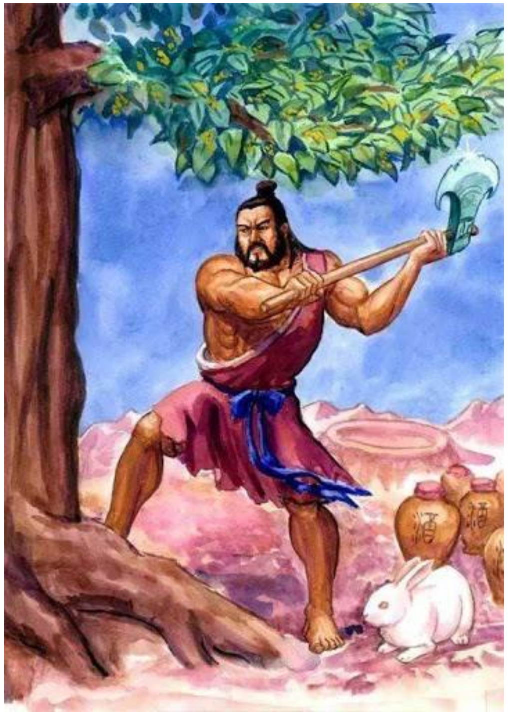

Raw
18000
16000
14000
12000
10000
50
100
150
200
250
AHE
14000
12000
10000
8000
50
100
150
200
250
HE
18000
16000
14000
12000
10000
8000
6000
0
50
100
150
200
250
CLAHE
14000
12000
10000
8000
6000
4000
2000
0
50
100
150
200
250

Take home message

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





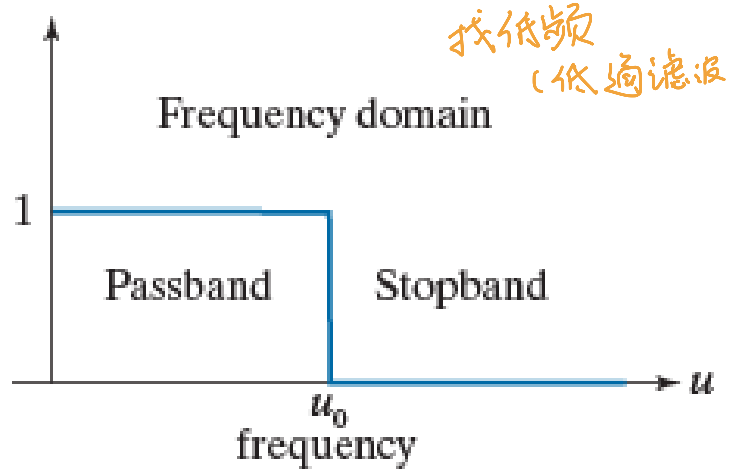


Spatial filtering (2)

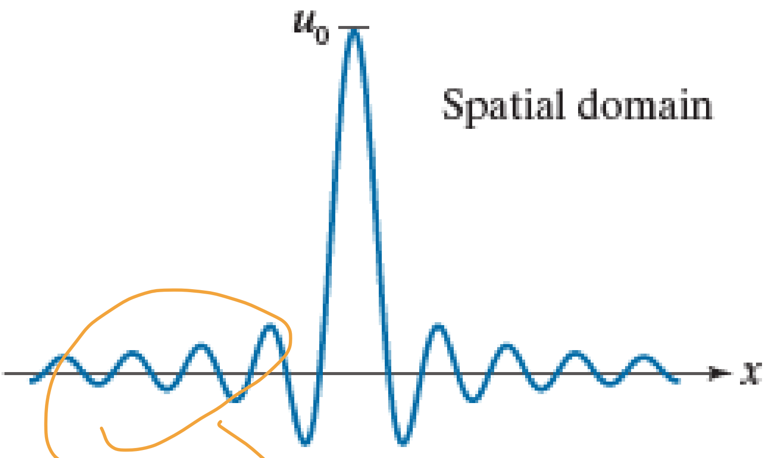
- Some other perspectives on spatial filtering
- Sobel Filter
- Unsharpen Filter (非锐化掩蔽)
- LoG Filter
  - - useful for finding edges
  - - also useful for finding blobs

Filtering in frequency domain and spatial domain

找低频
(低通滤波)



Ideal 1D LP filter



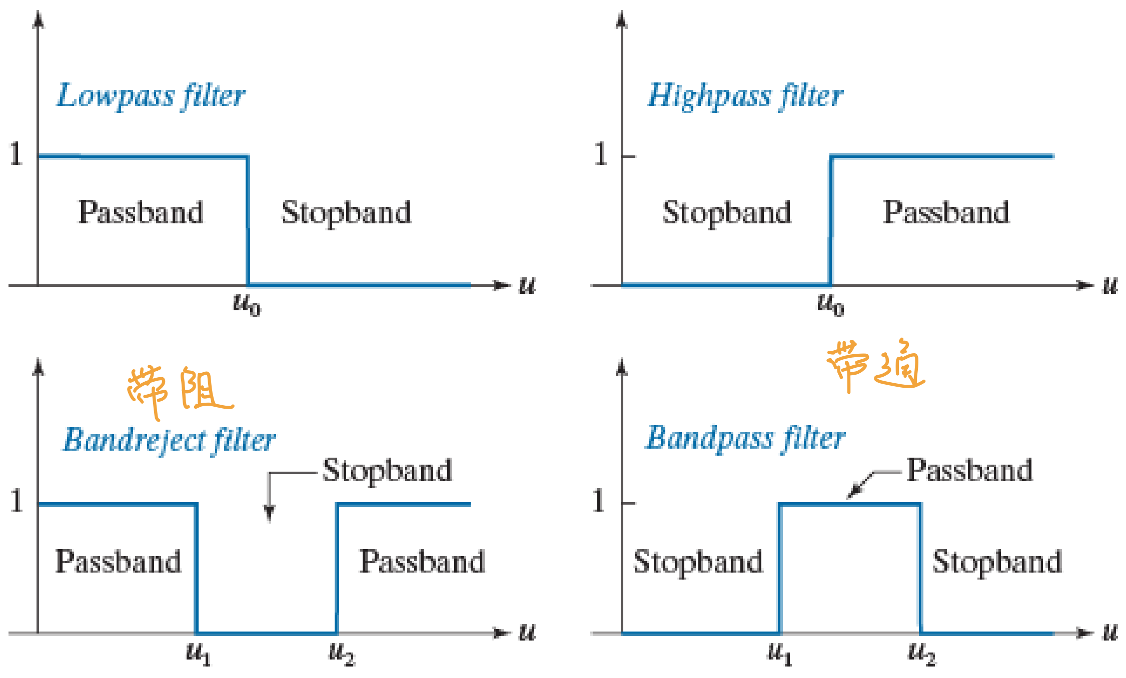
Ideal 1D LP filter kernel in spatial domain

Q: Is ideal filter really ideal for image processing?

不会直接用理想的 → 伪影，波纹

Filtering in frequency domain and spatial domain

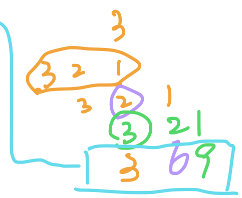
- 4 types of filters



Separable filter kernels

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

a当图像，kernel翻转

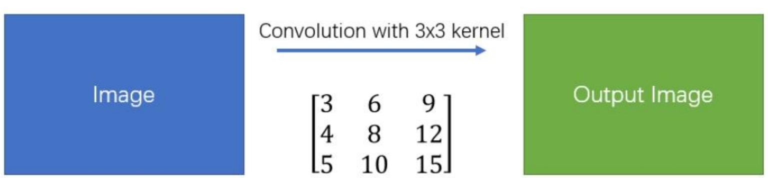


$$w = ab^T = a * b^T$$

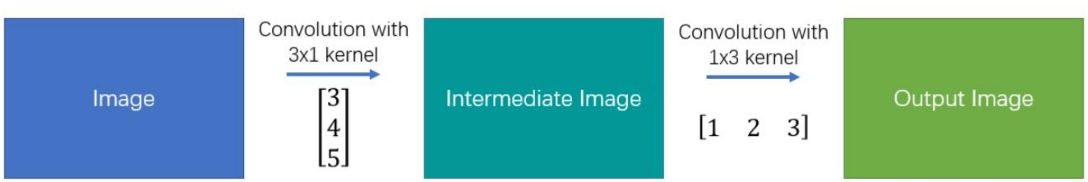
$$w \star f = (w_1 \star w_2) \star f = (w_2 \star w_1) \star f = w_2 \star (w_1 \star f) = (w_1 \star f) \star w_2$$

交换率

Simple Convolution



Spatial Separable Convolution



picture: MxN

kernel size: mxn

Computational advantage:

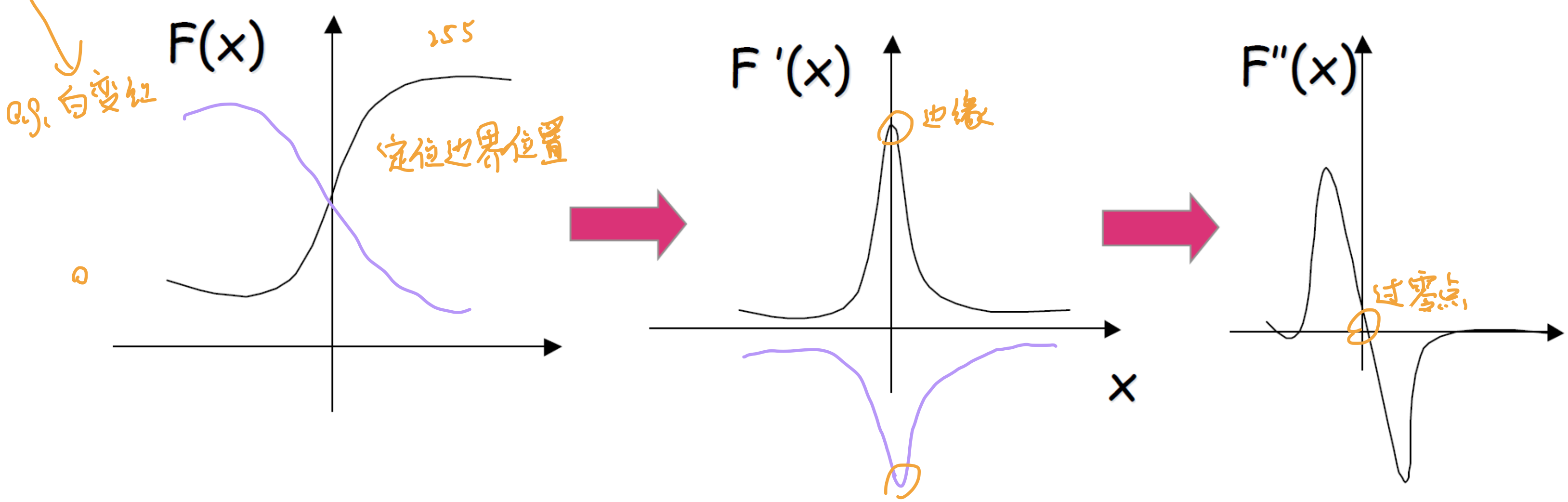
simple conv

$$C = \frac{MNmn}{\underbrace{MN(m+n)}_{\text{spatial separable}}} = \frac{mn}{(m+n)}$$

Recall: First & Second-Derivative filters

see

- 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

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

加入对角线差导

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)

$$\begin{matrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{matrix} \neq \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \times \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$$

因为相加，不是连续卷

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

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

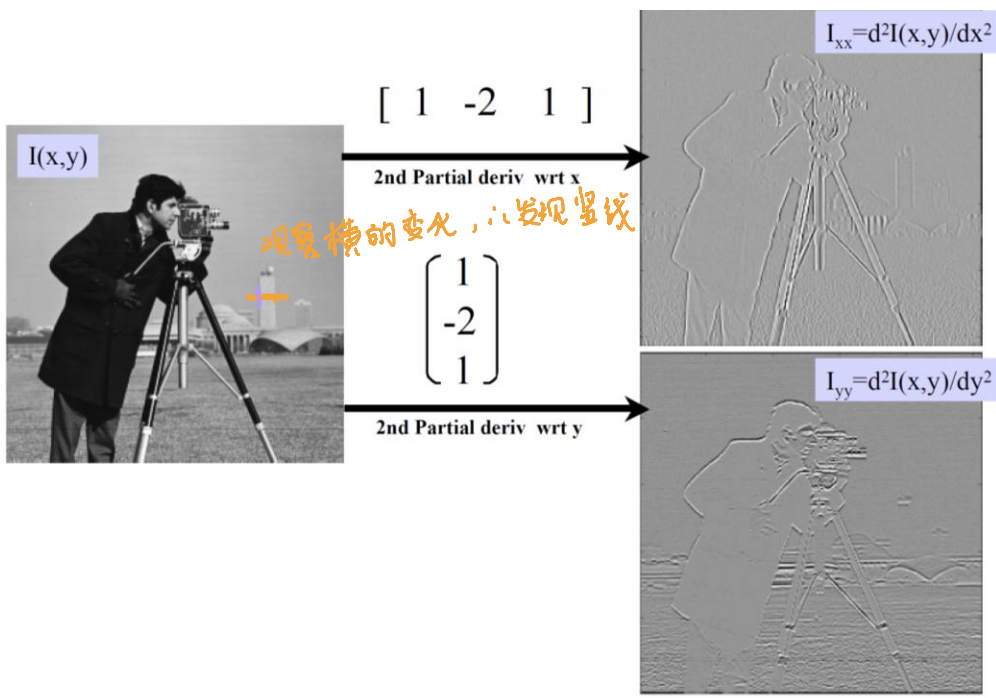
Y direction: $\dfrac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$

I_{xx}

1	-2	1

I_{yy}

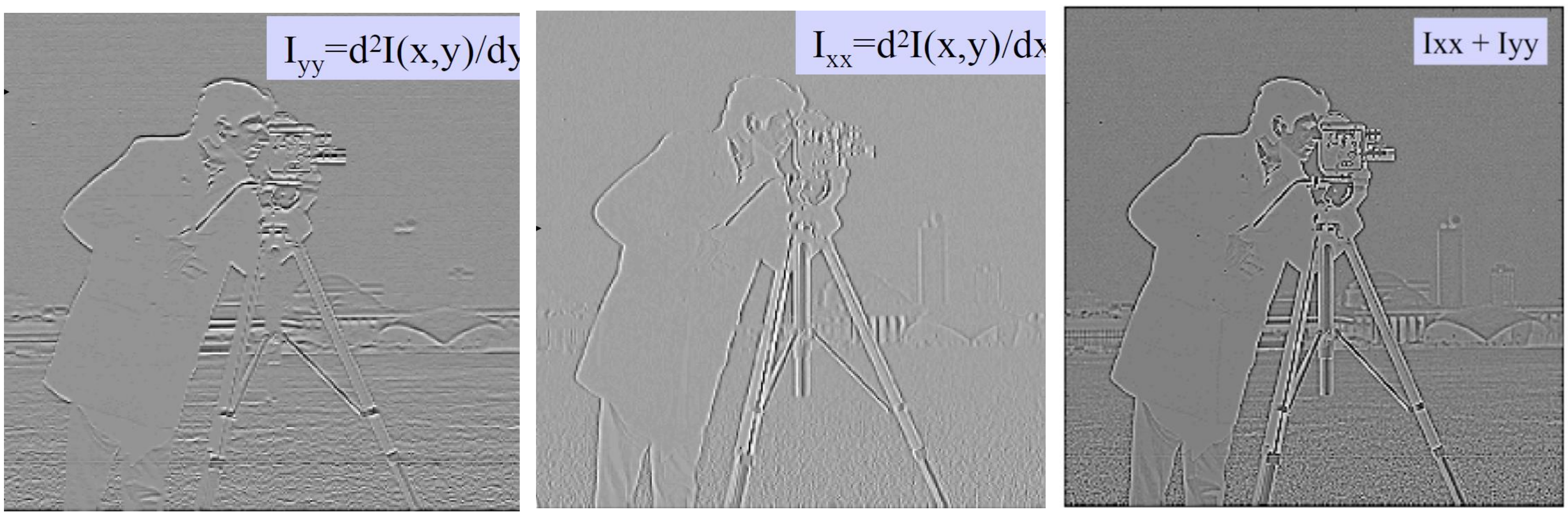
1
-2
1



观察横的变化，∴发现竖线

Laplacian

y x



Gradient(梯度)

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

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

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

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

Gradient(梯度)

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

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

$$= |z_9 - z_5| + |z_8 - z_6|$$ 对角线

偏移半个像素点

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）**

提取两个方向上差异

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

平滑 → 抑制噪声

求差分对噪声敏感

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

一阶导

x方向

y方向

−1	−2	−1
0	0	0
1	2	1

平滑

−1	0	1
−2	0	2
−1	0	1

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

Sobel operator

laplace 比 sobel 更多噪声

The Notes about the Laplacian

- $\nabla^2 I(x, y)$ is a SCALAR 标量
  - ↑ 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.

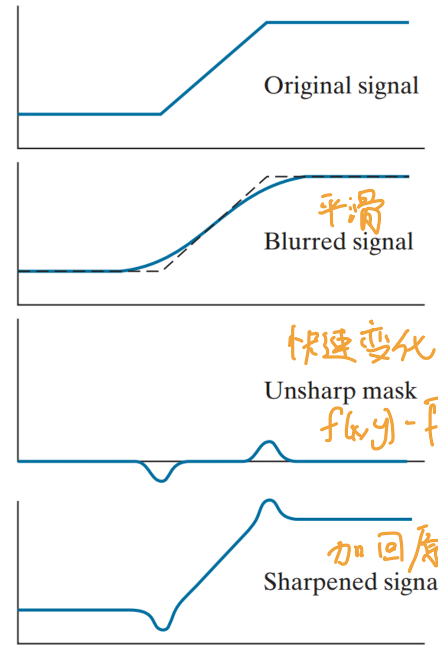
Unsharpen Mask(非锐化掩蔽)

想锐化高频信息

原　　低频部分

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

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



Laplacian of Gaussian (LoG) Filter

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

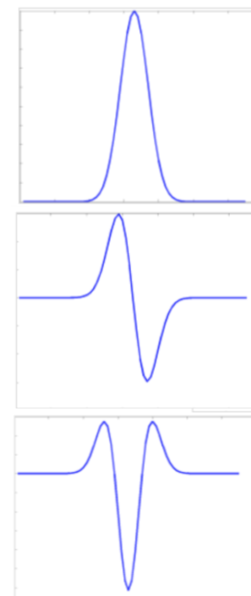
$$\nabla^2\,(G(x,y))$$

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

可调参数σ

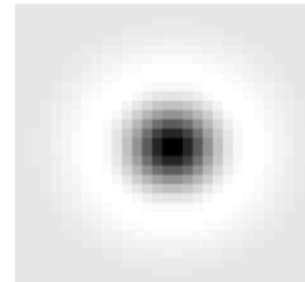
$$G'(x,y) = -\frac{1}{2\sigma^2}2(x+y)e^{-\frac{x^2+y^2}{2\sigma^2}} = -\frac{x+y}{\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}$$

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

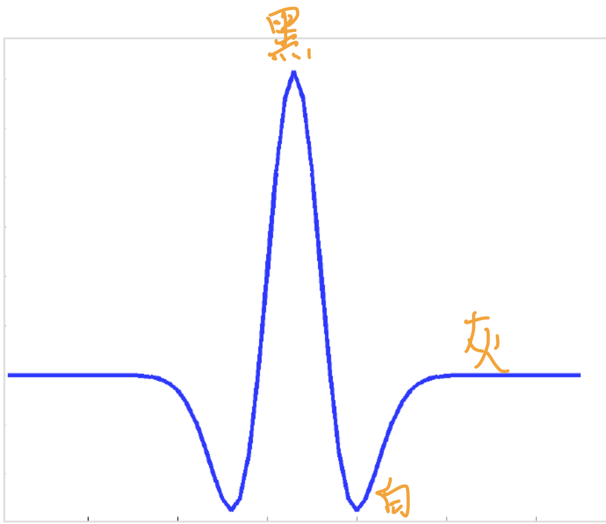


Second derivative of a Gaussian

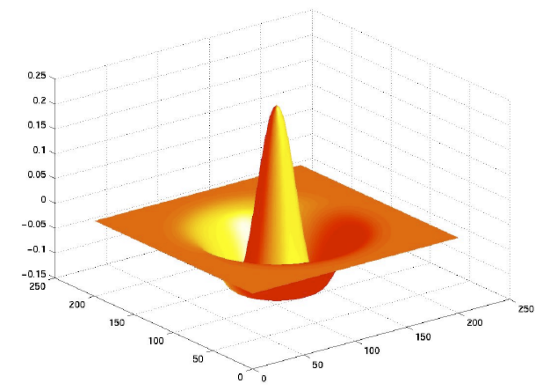
$$G''(x,y) = -\frac{1}{\pi\sigma^4}\left(1-\frac{x^2+y^2}{2\sigma^2}\right)e^{-\frac{x^2+y^2}{2\sigma^2}}$$



取反



2D analog



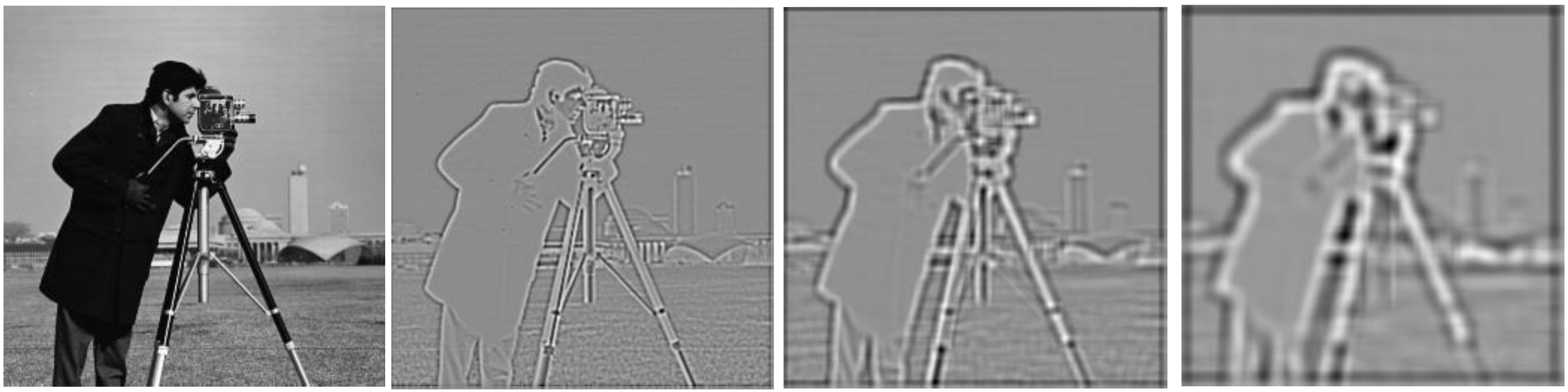
LoG "Mexican Hat"

频域也 gaussian

Effect of LoG Filter

低尺度
低频成分 提取边缘

Sigma = 1 Sigma = 4 Sigma = 10

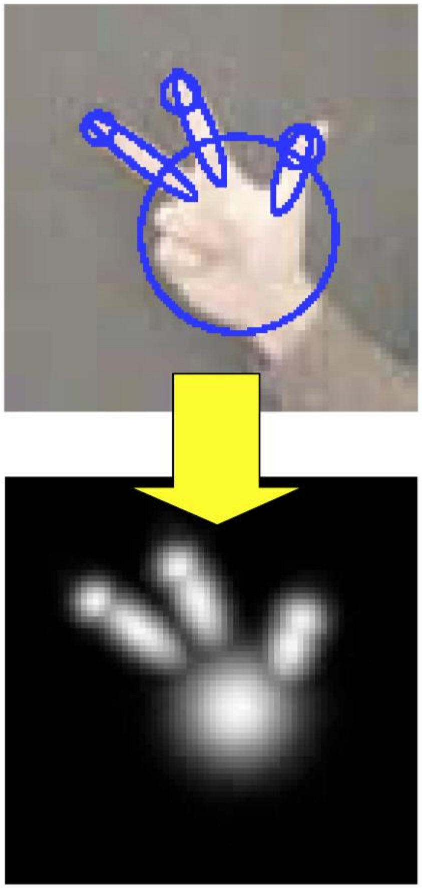


Band-Pass Filter (suppresses both high and low frequencies)

本质上带通滤波器

Application of LoG Filter

提取不同尺度下轮廓



Gesture recognition for
沙发上吃薯片
the ultimate couch potato
懒人不想拿摇控器

Matlab practice: spatial filtering

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

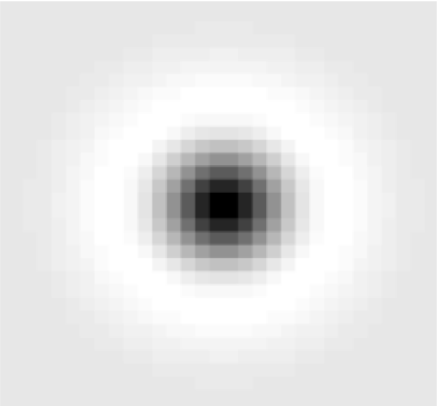
- See some examples.

- Then practice by yourself…

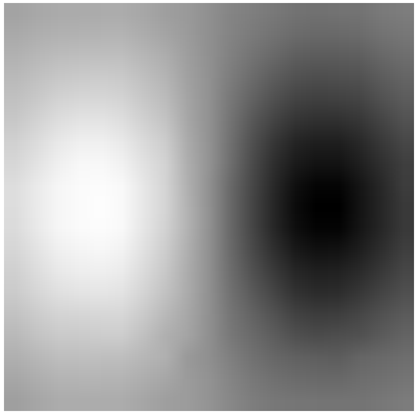
Take home message

- 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



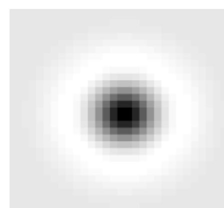
Derivative of Gaussian



Take home message

找图像与kernel相似部分

*



=