CS270: Digital Image Processing

Dr. Xavier LAGORCE

Email: xavier@shanghaitech.edu.cn

Office: SIST 1D-301B

ShanghaiTech University

Lecture 4
Intensity transformation & Spatial Filtering II

Dr. Xavier LAGORCE
Email: xavier@shanghaitech.edu.cn
Office: 1D-301.B, SIST
ShanghaiTech University

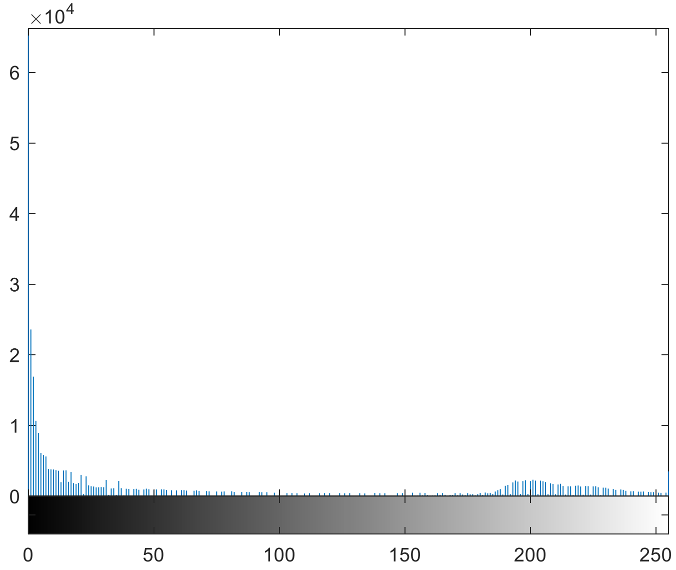
上海科技大学
ShanghaiTech University

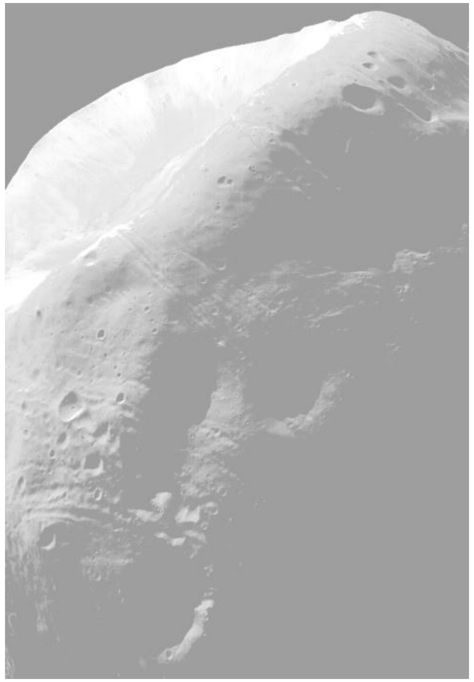
Intensity transforms (2)

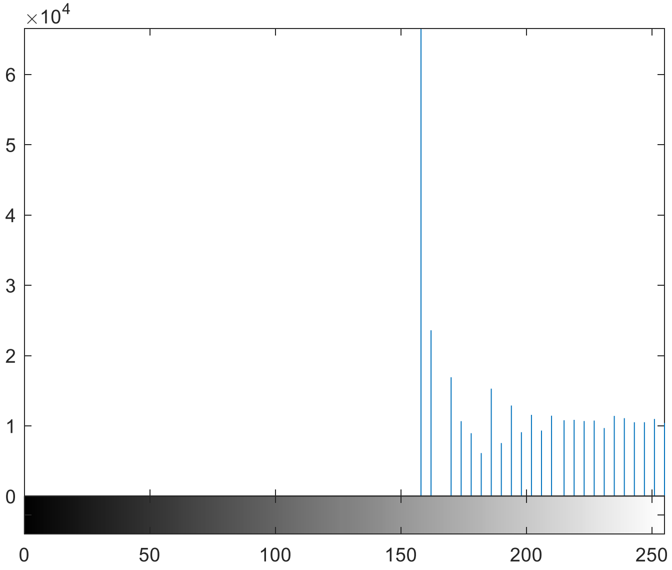
- Adaptive Histogram Equalization (AHE)
- Contrast Limited Adaptive Histogram Equalization (CLAHE)

Key problem of Histogram Equalization (HE)

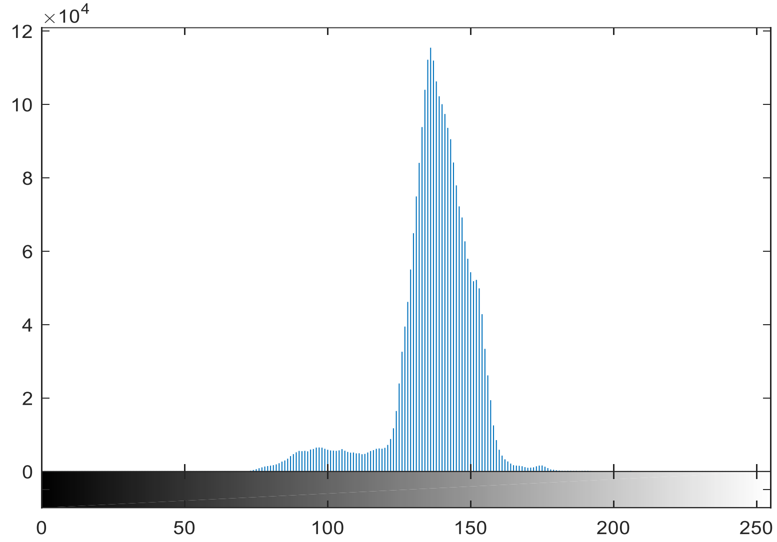
$$s = T(r) = (L-1)\int_0^r p_r(w)dw = (L-1)\sum_{j=0}^{k} p_r(r_j) = (L-1)\sum_{j=0}^{k}\frac{n_j}{MN} = \frac{L-1}{MN}\sum_{j=0}^{k} n_j$$

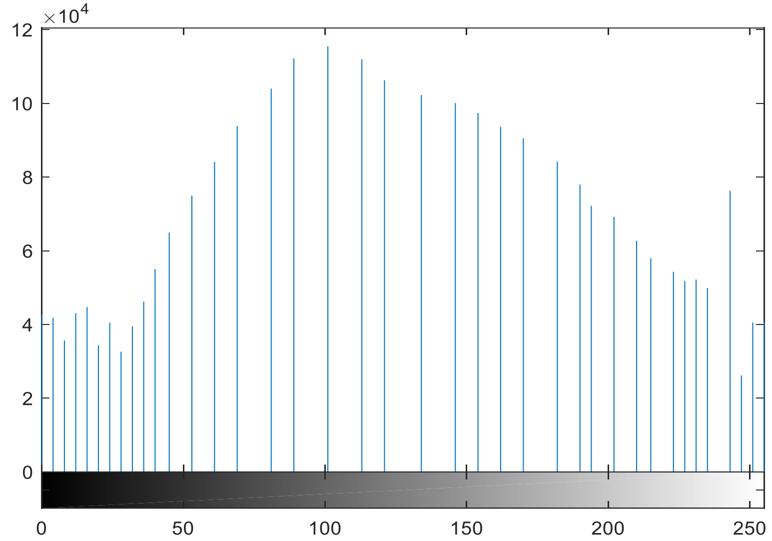






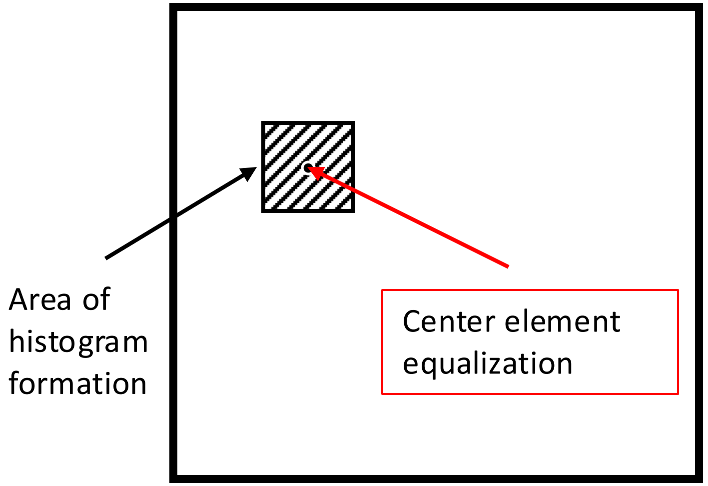
Key problem of Histogram Equalization (HE)

Adaptive Histogram Equalization (AHE)

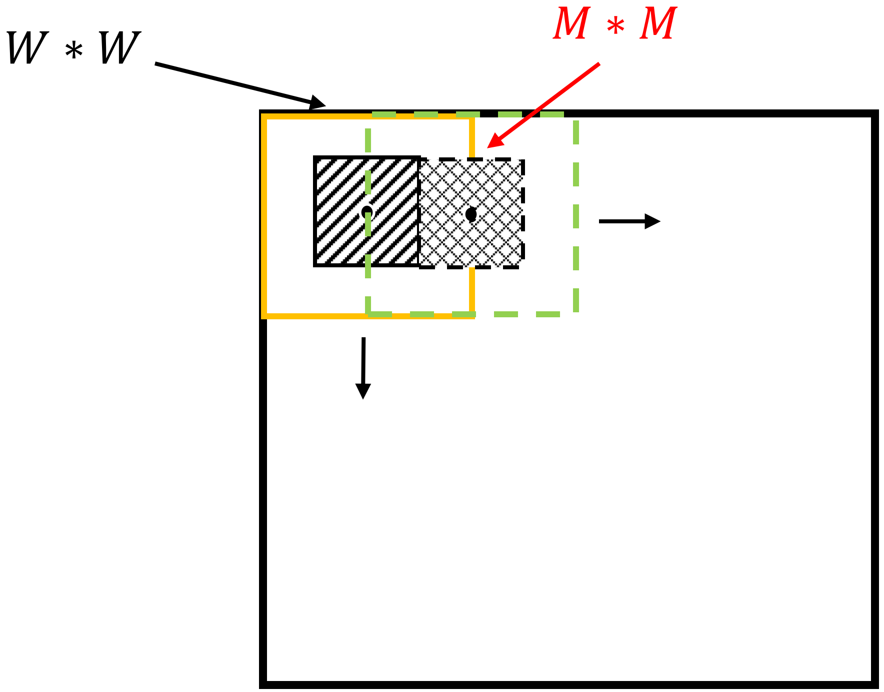
- Traverse every pixel with a $W * W$ patch, process histogram equalization within each patch and update the center pixel.
- Advantage: better uniform distributed histogram.
- Disadvantage: high complexity
- $O(W * W + L)$ within each patch
- $O(M * N * (W * W + L))$ for whole image



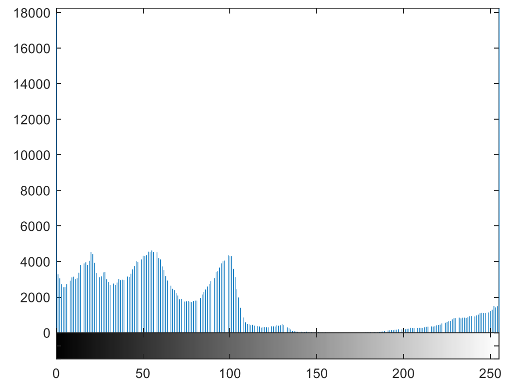
Adaptive Histogram Equalization (AHE)

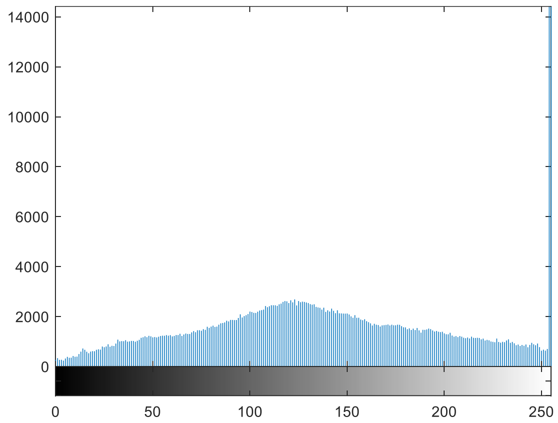
- For faster processing AHE, it is proposed to update a center patch of size $M * M$ instead of just the center pixel in each HE within the $W * W$ patch HE.

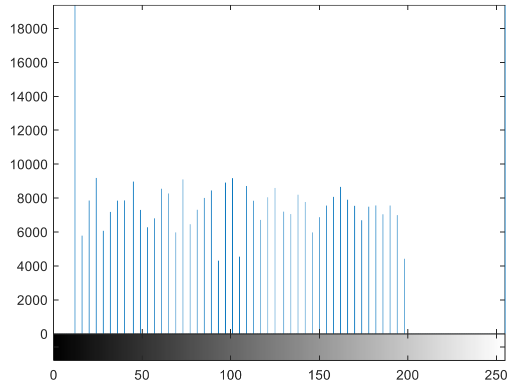
- Pixels near the image boundary have to be treated specially, This can be solved by extending the image by mirroring pixel lines and columns with respect to the image boundary.



Effect of AHE







- AHE has a tendency to overamplify noise in homogeneous areas
- Better at improving local contrast and edge information

Contrast Limited Adaptive Histogram Equalization (CLAHE)

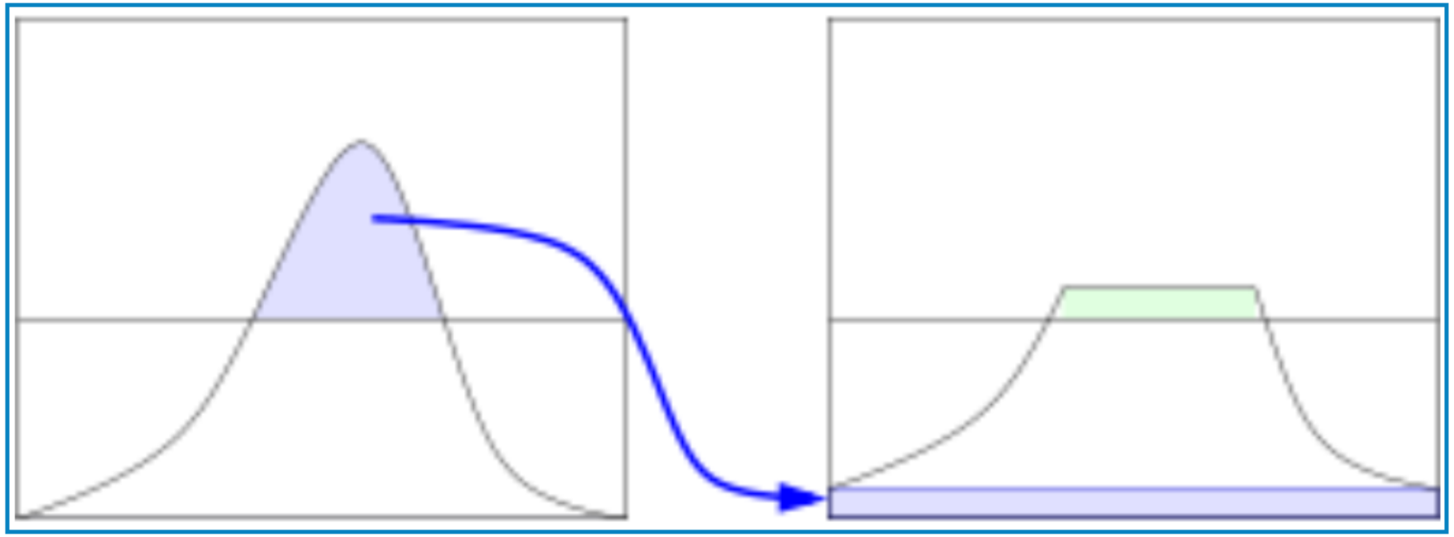
上海科技大学
ShanghaiTech University

- CLAHE differs from naive AHE because it limits local contrast.
- 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.

立志成才 报国裕民

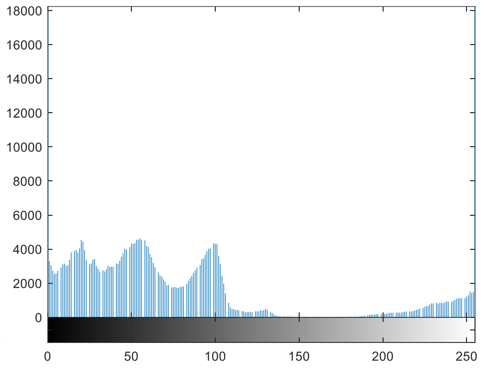
Contrast Limited Adaptive Histogram Equalization (CLAHE)

- 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.

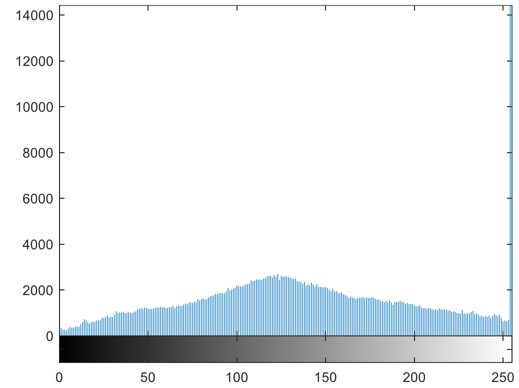


CLAHE

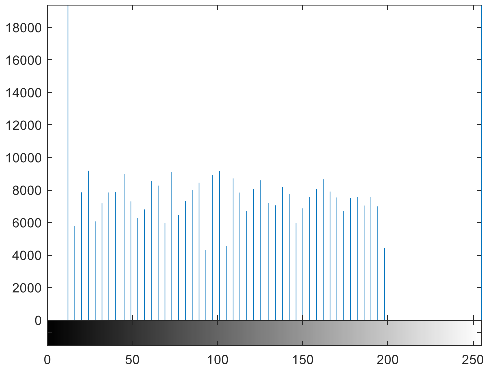

Raw


18000
16000
14000
12000
10000
8000
6000
4000
2000
0
0
50
100
150
200
250

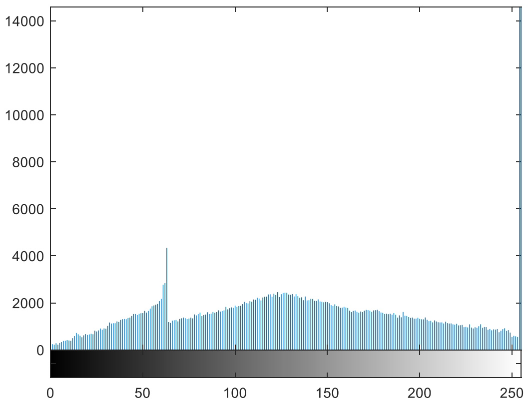

AHE


14000
12000
10000
8000
6000
4000
2000
0
0
50
100
150
200
250


HE

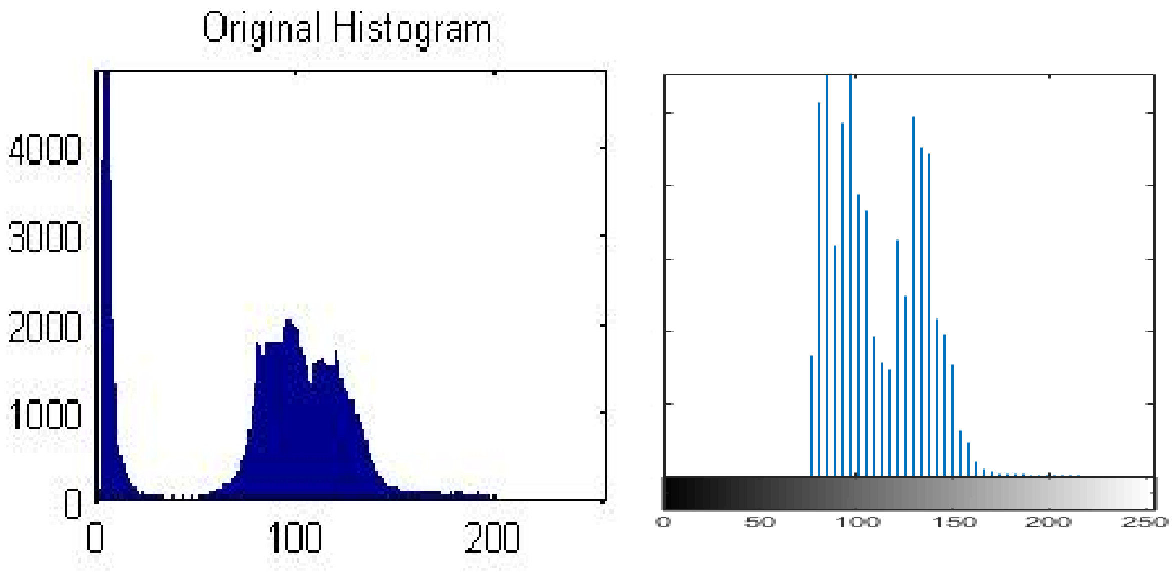

18000
16000
14000
12000
10000
8000
6000
4000
2000
0
0
50
100
150
200
250


CLAHE


14000
12000
10000
8000
6000
4000
2000
0
0
50
100
150
200
250

Take home message

- Key idea: AHE&CLAHE were developed to prevent the over amplification of noise.
  - AHE: Fragments the bins to spread them
  - CLAHE: cuts-off the head of the histogram and distributes it



Spatial filtering (2)

上海科技大学 ShanghaiTech University

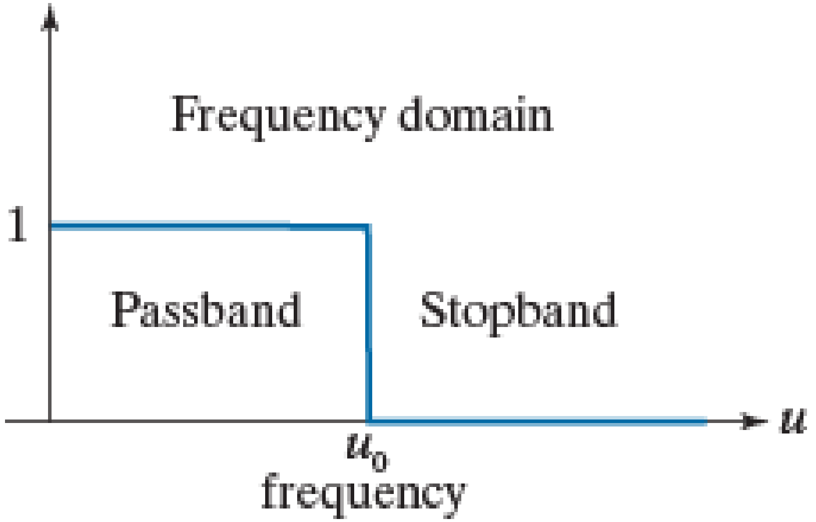
- Some other perspectives on spatial filtering

- Sobel Filter
- Unsharp Masking (非锐化掩蔽)
- LoG Filter (Laplacian of Gaussian)
  - useful for finding edges
  - also useful for finding blobs

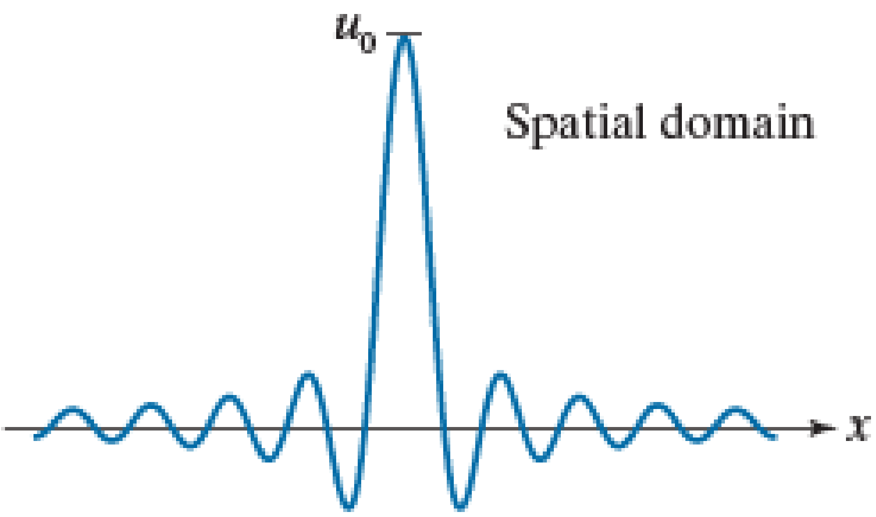
立志成才报国裕民

Filtering in frequency domain and spatial domain

上海科技大学
ShanghaiTech University



Ideal 1D Low-Pass filter



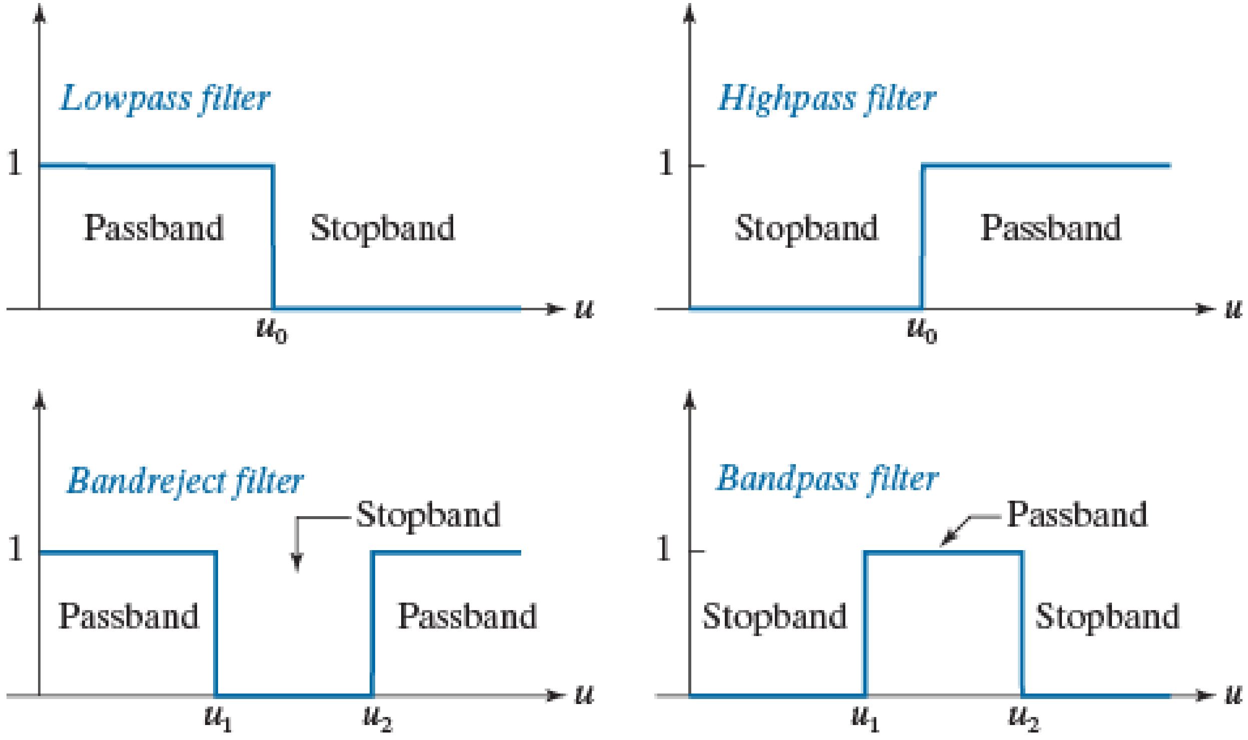
Ideal 1D Low-Pass filter kernel in spatial domain

Q: Is ideal filter really ideal for image processing?

立志成才报国裕民

Filtering in frequency domain and spatial domain

- 4 types of filters



Filter type	Spatial kernel in terms of lowpass kernel, *lp*
Lowpass	$lp(x,y)$
Highpass	$hp(x,y) = \delta(x,y) - lp(x,y)$
Bandreject	$br(x,y) = lp_1(x,y) + hp_2(x,y)$ $= lp_1(x,y) + [\delta(x,y) - lp_2(x,y)]$
Bandpass	$bp(x,y) = \delta(x,y) - br(x,y)$ $= \delta(x,y) - [lp_1(x,y) + [\delta(x,y) - lp_2(x,y)]]$

Unsharp Mask (非锐化掩蔽)

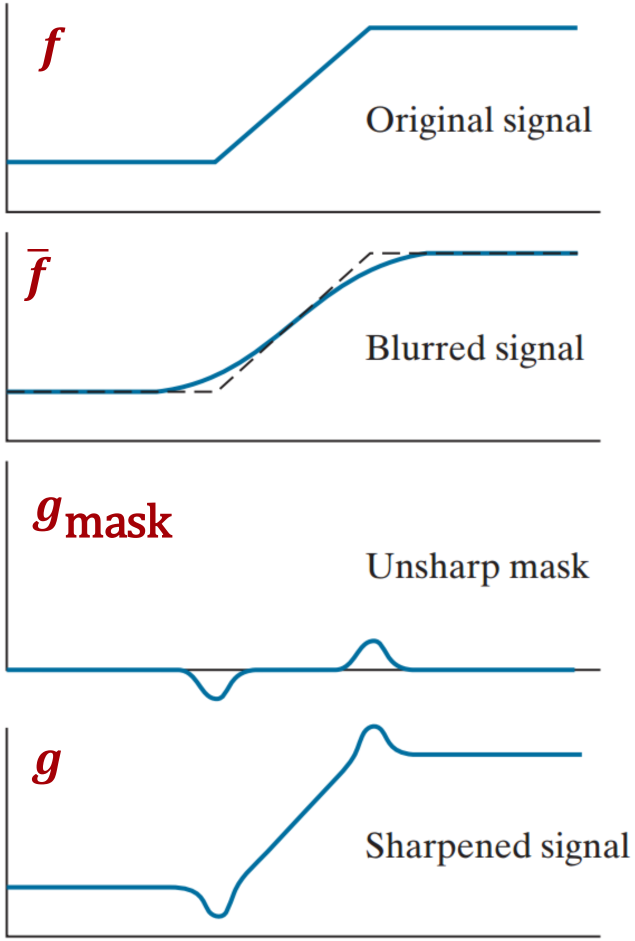
上海科技大学
ShanghaiTech University

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

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

f $\bar{f}$ g_{mask}

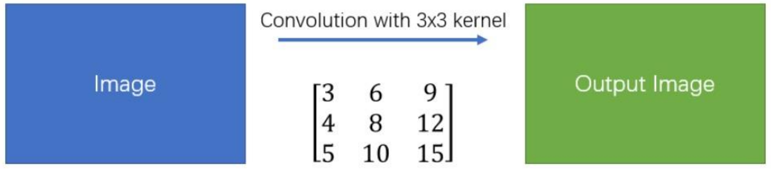
$g, k = 1$ $g, k = 4, 5$



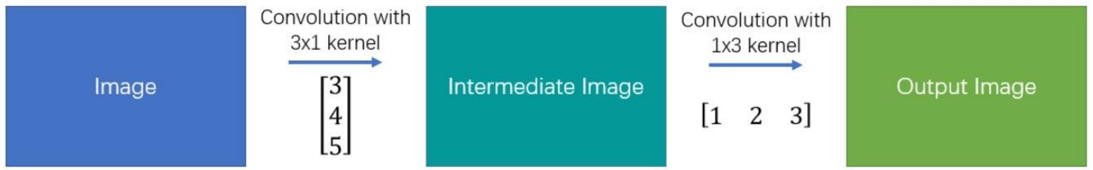
Separable filter kernels

- Example: $\begin{bmatrix} 3 & 6 & 9 \\ 4 & 8 & 12 \\ 5 & 10 & 15 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$
- $\boldsymbol{w} = \boldsymbol{ab}^T$: outer product of 2 vectors   equivalent to convolution !
- If $\boldsymbol{w} = \boldsymbol{w}_1 \star \boldsymbol{w}_2$, then
  - $\boldsymbol{w} \star \boldsymbol{f} = (\boldsymbol{w}_1 \star \boldsymbol{w}_2) \star \boldsymbol{f} = (\boldsymbol{w}_2 \star \boldsymbol{w}_1) \star \boldsymbol{f} = \boldsymbol{w}_2 \star (\boldsymbol{w}_1 \star \boldsymbol{f}) = (\boldsymbol{w}_1 \star \boldsymbol{f}) \star \boldsymbol{w}_2$

Simple Convolution



Spatial Separable Convolution

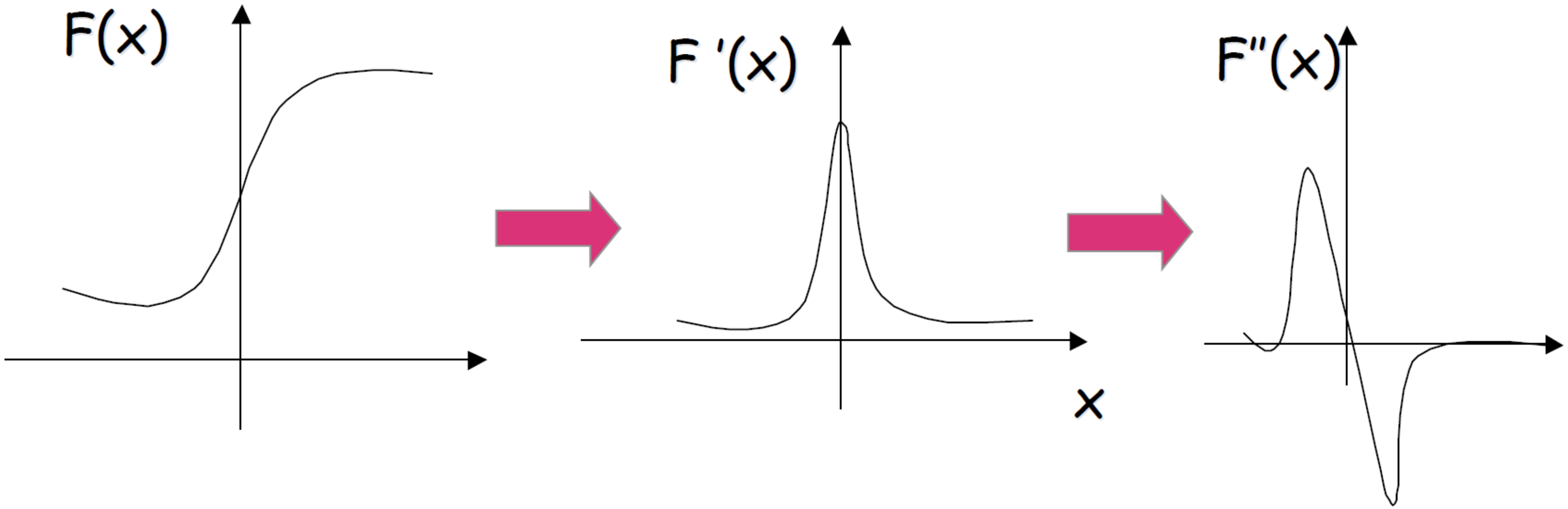


Computational advantage:

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

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

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

Discrete Laplacian

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

Laplacian Filter Masks

- Laplacian as a filter kernel

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

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

$$\begin{aligned}\nabla^2 f(x,y) = \; & f(x,y+1) + f(x,y-1) + f(x+1,y) + f(x-1,y) \\ & + f(x+1,y+1) + f(x+1,y-1) + f(x-1,y-1) + f(x-1,y+1) \\ & - 8f(x,y)\end{aligned}$$

Laplacian (拉普拉斯算子)

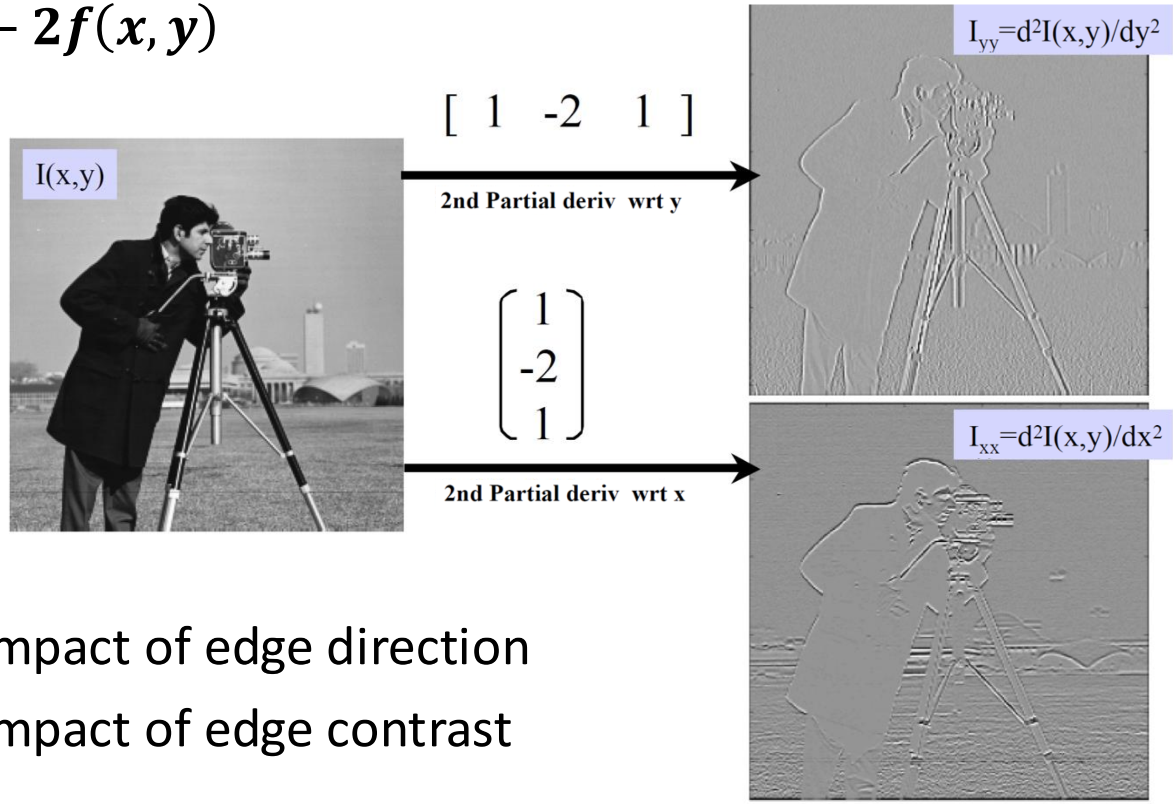
- For an image function $f(x,y)$,
  - X direction： $\frac{\partial 2f}{\partial x2} = f(x+1,y) + f(x-1,y) - 2f(x,y)$
  - Y direction： $\frac{\partial 2f}{\partial y2} = f(x,y+1) + f(x,y-1) - 2f(x,y)$

I_{yy}

1	-2	1

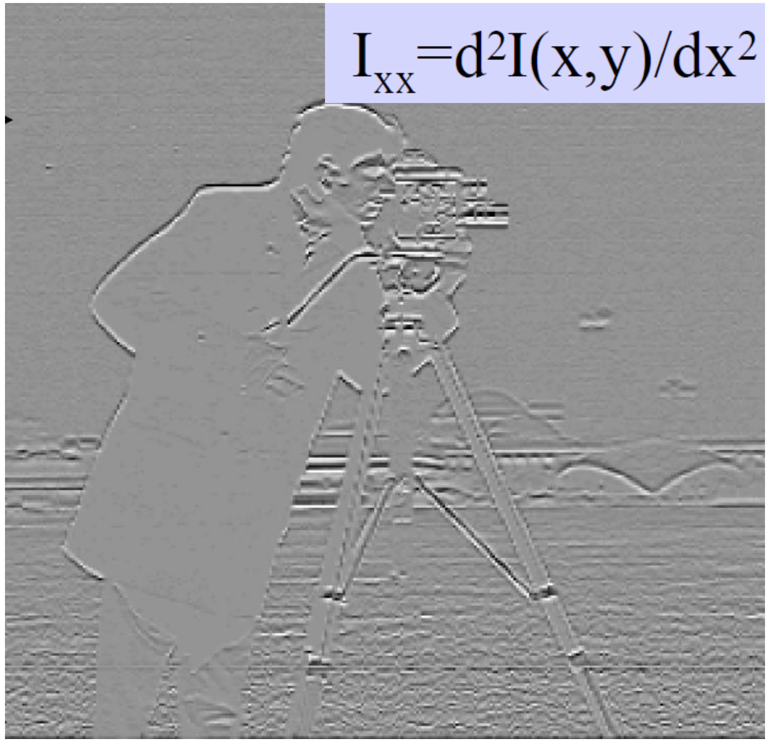
1
-2
1

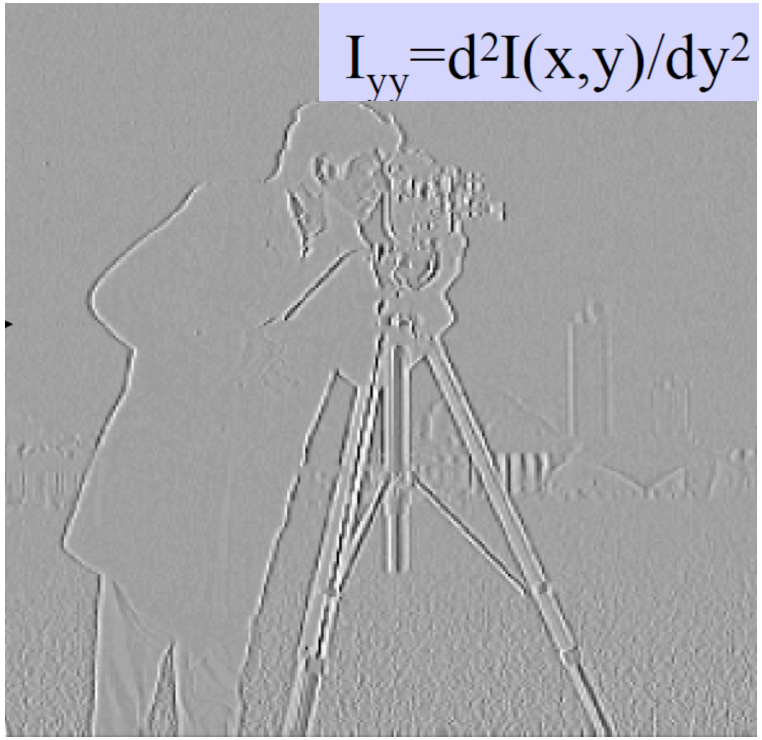
I_{xx}



1. Impact of edge direction
2. Impact of edge contrast

Laplacian





Gradient (梯度)

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

points in the direction of the greatest rate of change of f at location (x, y)

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

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

To help computation complexity !
But keeps the main properties of the amplitude

Gradient (梯度)

- Simplest possibility
  - $g_x = (z_8 - z_5)$ and $g_y = (z_6 - z_5)$

- Roberts cross-gradient operator（罗伯特交叉梯度算子）
  - Uses cross differences
  - $\boldsymbol{M(x,y) \approx |g_x| + |g_y|}$
  - $\boldsymbol{= |z_9 - z_5| + |z_8 - z_6|}$

-1	0
0	1

0	-1
1	0

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- But the kernel does not have an odd size, we prefer at least a 3x3 kernel…

Gradient (梯度)

- Sobel operator （Sobel算子）
- $M(x,y) = |\underbrace{(z_7+2z_8+z_9)-(z_1+2z_2+z_3)}_{g_x}| + |\underbrace{(z_3+2z_6+z_9)-(z_1+2z_4+z_7)}_{g_y}|$

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

g_x

−1	0	1
−2	0	2
−1	0	1

g_y

Averaging

Partial derivative

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

Sobel operator

g_x

−1	−2	−1
0	0	0
1	2	1

Horizontal edges

g_y

−1	0	1
−2	0	2
−1	0	1

Vertical edges

Sobel operator

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.

Laplacian of Gaussian (LoG) Filter

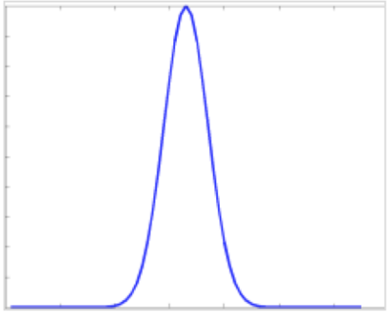
- **First smooth (Gaussian filter),**
- **Then, find zero-crossings (Laplacian filter):**

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

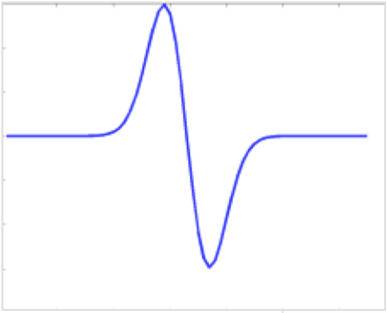
$$G(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

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

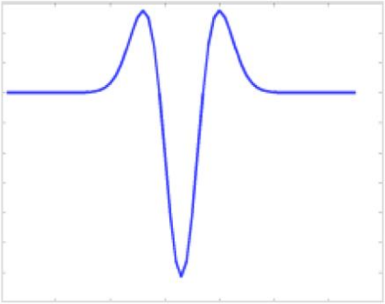
$$G''(x,y) = -\frac{1}{\pi\sigma^4}(1-\frac{x^2+y^2}{2\sigma^2})\ e^{-\frac{x^2+y^2}{2\sigma^2}}$$



$G(x,y)$



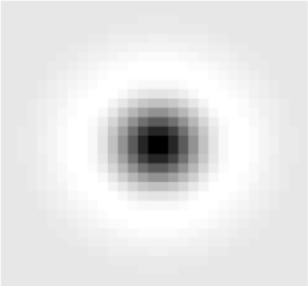
$G'(x,y)$

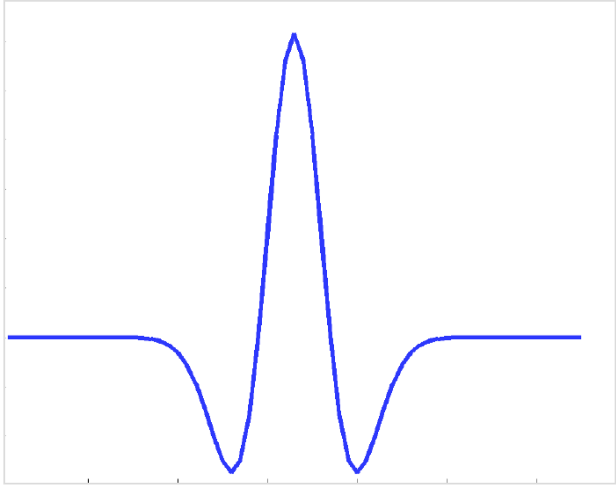


$G''(x,y)$

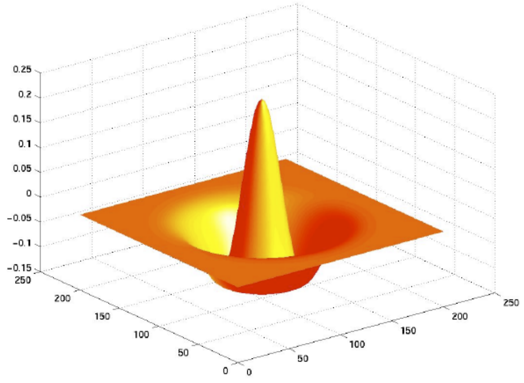
Second derivative of a Gaussian

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





2D analog



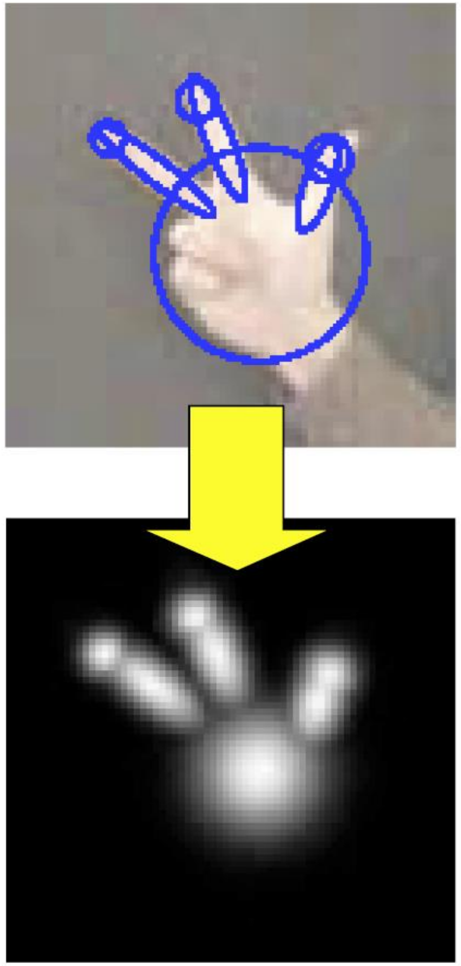
LoG "Mexican Hat"

Effect of LoG Filter

Sigma = 1 | Sigma = 4 | Sigma = 10

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

Application of LoG Filter



Gesture recognition for
the ultimate couch potato

Matlab practice: spatial filtering

上海科技大学
ShanghaiTech University

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

- See some examples.

- Then practice by yourself…

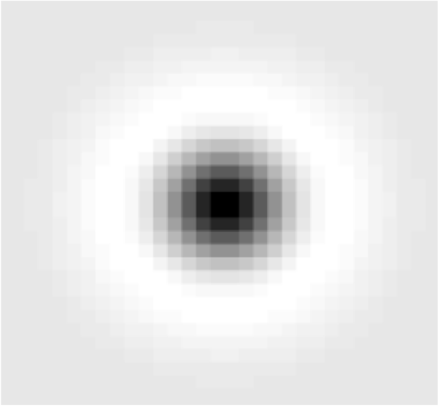
立志成才报国裕民

Take home message

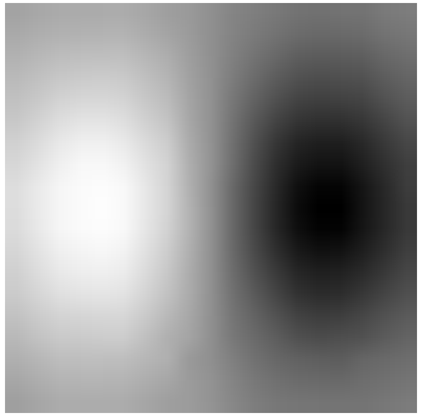
上海科技大学
ShanghaiTech University

- 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



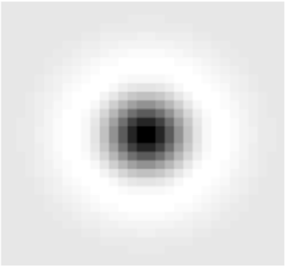
Derivative of Gaussian



立志成才报国裕民

Take home message

*



=