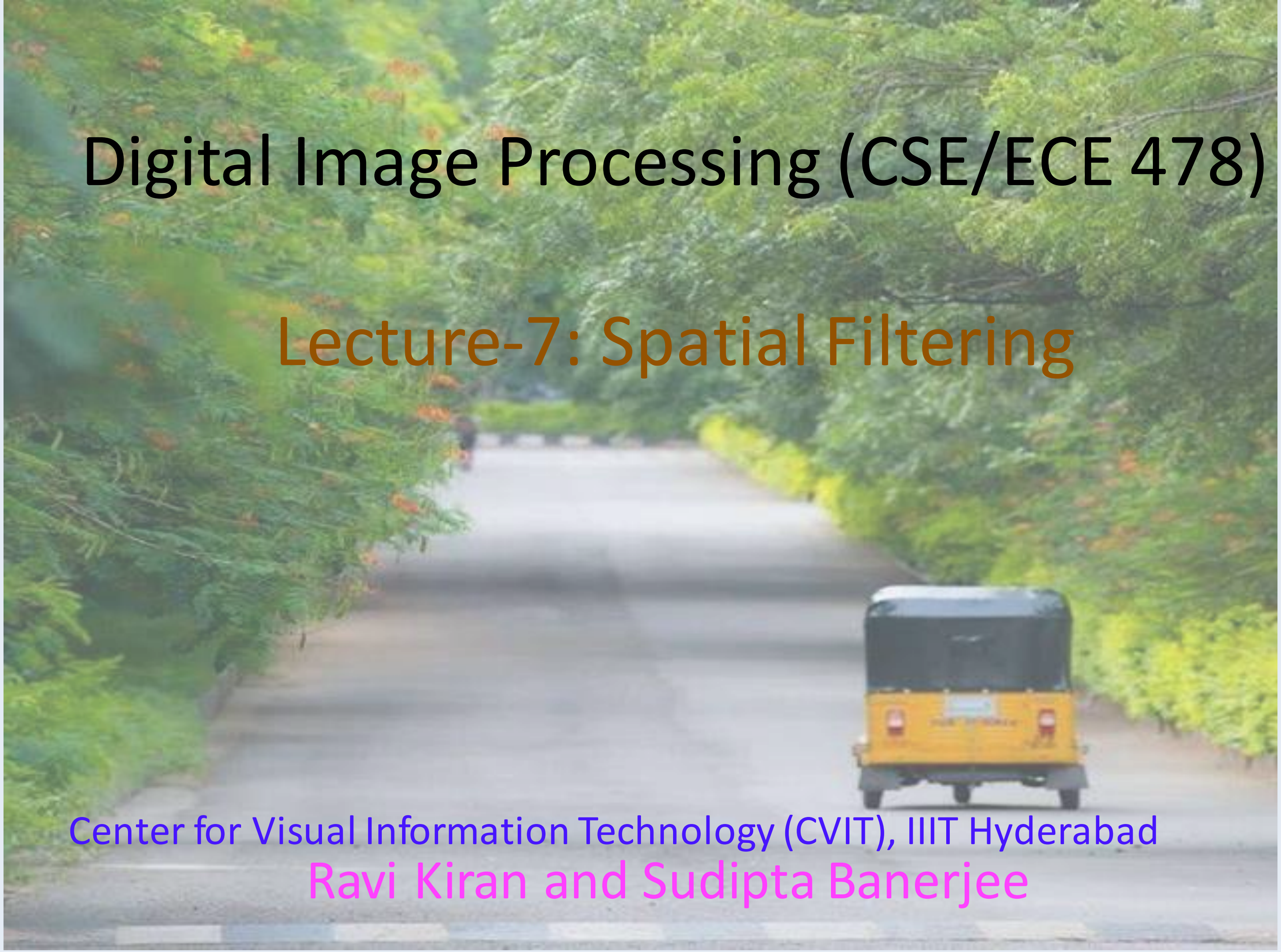
14.09.2021



Digital Image Processing (CSE/ECE 478)

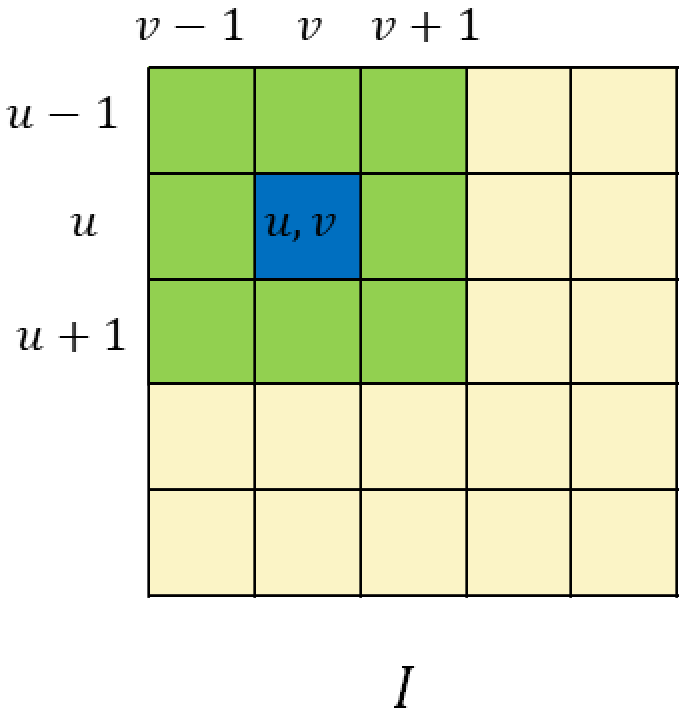
Lecture-7: Spatial Filtering

Center for Visual Information Technology (CVIT), IIIT Hyderabad
Ravi Kiran and Sudipta Banerjee

Announcements

- Mini Quiz – 2 today

Mean/Average Filter



Note: Coefficients sum to 1

H → 3 x 3 Weight Mask / Kernel / Filter

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

$$I'(u,v) \leftarrow \frac{1}{9} \cdot \sum_{j=-1}^{1} \sum_{i=-1}^{1} I(u+i, v+j)$$

$$I'(u,v) \leftarrow \sum_{j=-1}^{1} \sum_{i=-1}^{1} I(u+i, v+j)$$

- $H(i,j)$

Effect of Mask Size

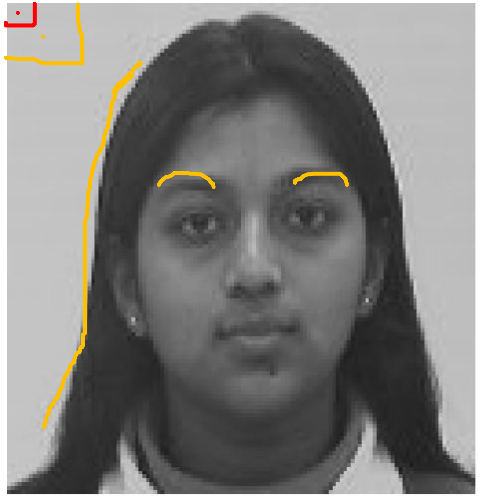
Original Image

[3x3]

[5x5]

[7x7]

Repeated Averaging Using Same Filter



Before

After

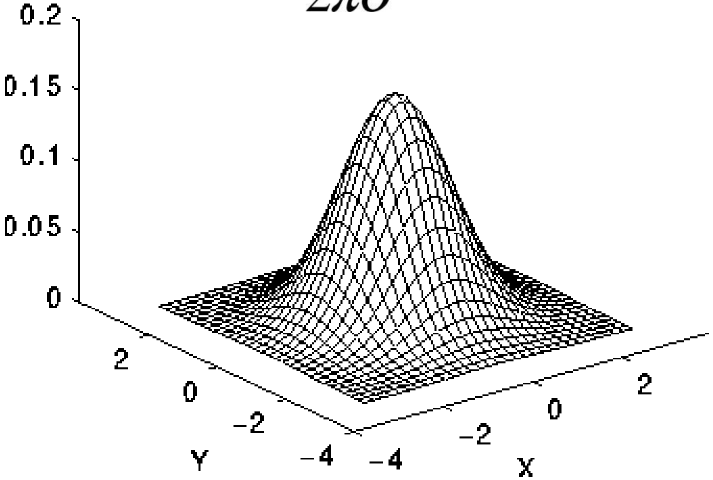
After repeated averaging

NOTE: Can get the <u>effect</u> of larger filters by smoothing repeatedly with smaller filters

Gaussian Smoothing

- Mask weights are samples of a zero-mean 2-D Gaussian

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\{-(x^2 + y^2)/2\sigma^2\}$$

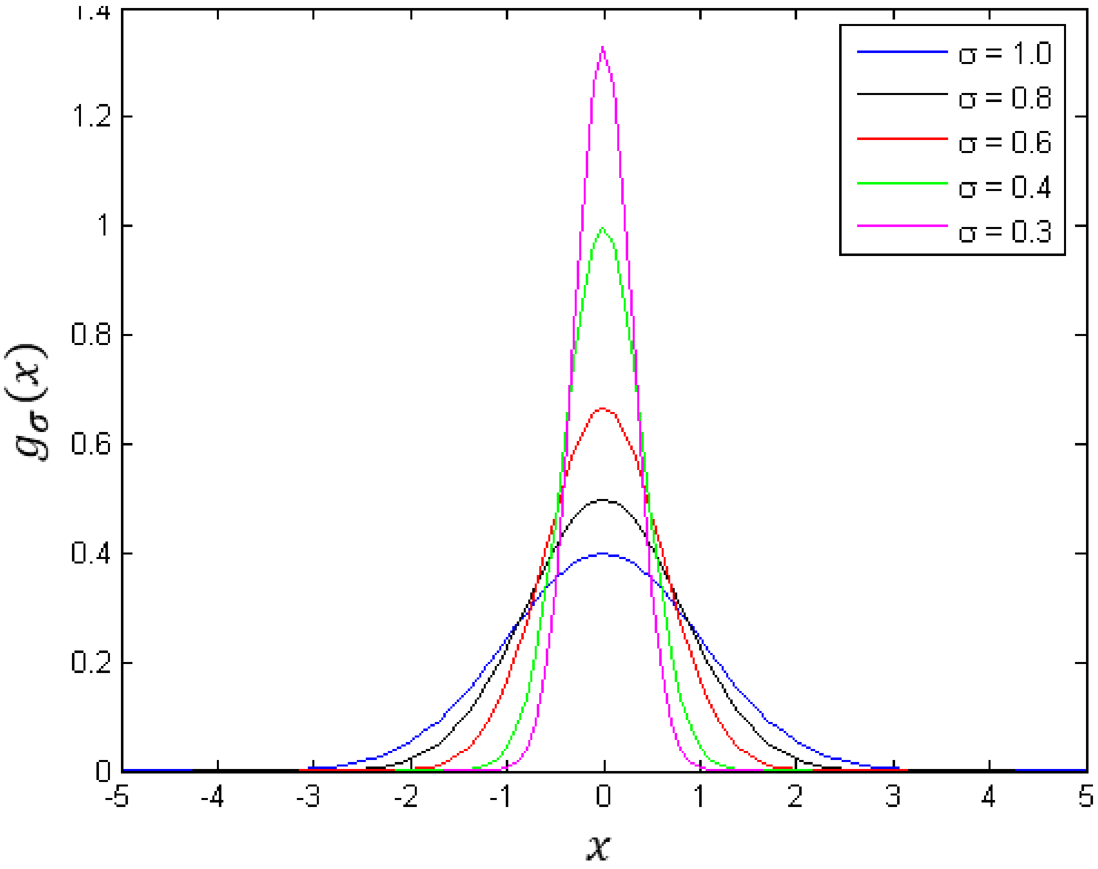


$\frac{1}{273}$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

5×5 Gaussian filter, σ=1

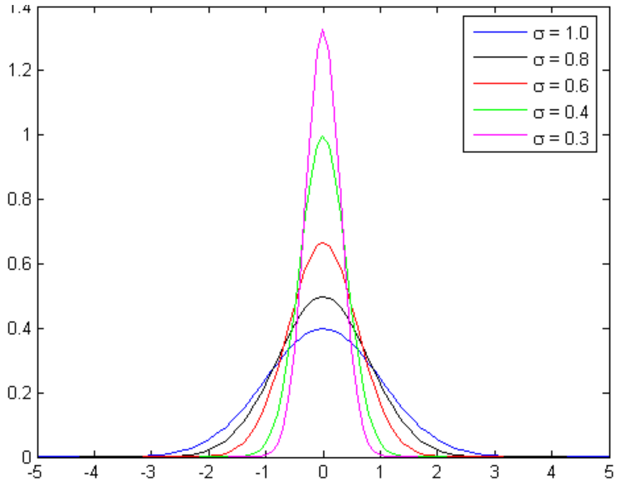
How are Gaussian filter coefficients obtained ?



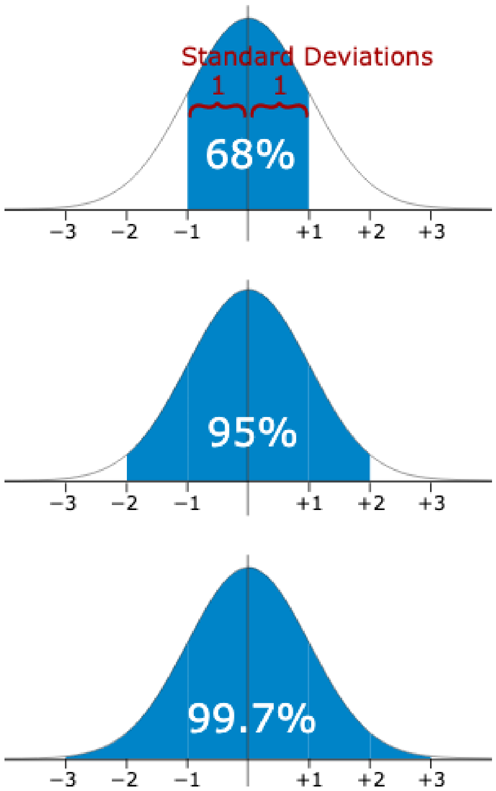
$$g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

Note: This is a PDF NOT Probability

How are Gaussian filter coefficients obtained ?

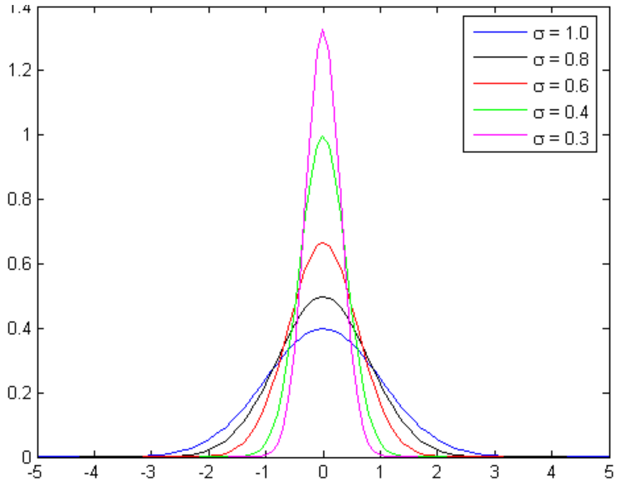


$$g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

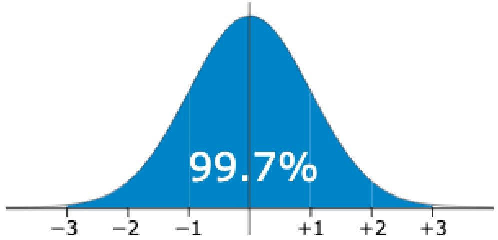


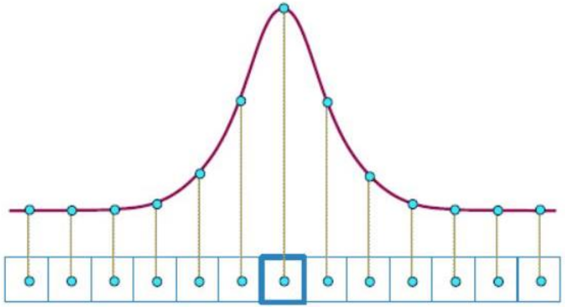
How are Gaussian filter coefficients obtained ?

$P(X \leq x)$ $P(X = x)$



$$g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$





Let s be the size of the filter, i.e., sxs

Heuristic:

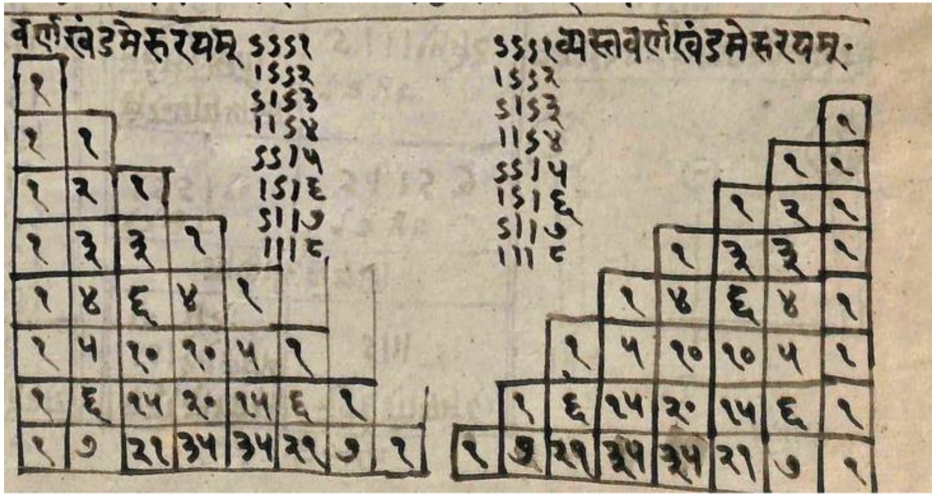
s = round(σ)

How are Gaussian filter coefficients obtained ?

$$\sum_{k=0}^{N} {}^{N}C_k = 2^N$$

Index N	Coefficients	Sum = 2^N
0	1	1
1	1 1	2
2	1 2 1	4
3	1 3 3 1	8
4	1 4 6 4 1	16
5	1 5 10 10 5 1	32
6	1 6 15 20 15 6 1	64
7	1 7 21 35 35 21 7 1	128
8	1 8 28 56 70 56 28 8 1	256
9	1 9 36 84 126 126 84 36 9 1	512
10	1 10 45 120 210 252 210 120 45 10 1	1024
11	1 11 55 165 330 462 462 330 165 55 11 1	2048
12	1 12 66 220 495 792 924 792 495 220 66 12 1	4096

Meru Prastaara, derived from Pingala's formulae (2 BCE), Manuscript from Raghunath Temple Library, Jammu



How are Gaussian filter coefficients obtained ?

- Use the row *n* of Pascal's Triangle as a one-dimensional, n-point approximation of a Gaussian filter.

						1						
					1		1					
				1		2		1				
			1		3		3		1			
		1		4		6		4		1		
	1		5		10		10		5		1	
1		6		15		20		15		6		1

Image Courtesy: http://ece.uprm.edu/~mtoledo/web/6088/F2018/Slides/6_Filtering.pdf

How are Gaussian filter coefficients obtained ?

$$g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right) \qquad \int_{-\infty}^{\infty} g_\sigma(x)dx = 1$$

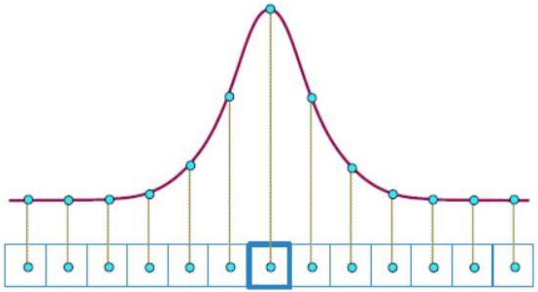
s = 7 x 7

$$\sum_{k=0}^{N} {}^{N}C_k = 2^N$$

Index N	Coefficients	Sum = 2^N
0	1	1
1	1 1	2
2	1 2 1	4
3	1 3 3 1	8
4	1 4 6 4 1	16
5	1 5 10 10 5 1	32
6	1 6 15 20 15 6 1	64
7	1 7 21 35 35 21 7 1	128
8	1 8 28 56 70 56 28 8 1	256
9	1 9 36 84 126 126 84 36 9 1	512
10	1 10 45 120 210 252 210 120 45 10 1	1024
11	1 11 55 165 330 462 462 330 165 55 11 1	2048
12	1 12 66 220 495 792 924 792 495 220 66 12 1	4096

1/64 1 6 15 20 15 6 1

$$\frac{\sum_{k=0}^{N} {}^{N}C_k}{2^N} = 1$$



Points to note:

- N=s-1
- Require the kernel size such that it extends most of the Gaussian area
- Heuristic: For sigma=1, use 5xsigma to cover 98.76% of the area

How are Gaussian filter coefficients obtained ?

$$g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{x^2}{2\sigma^2}\right)$$

s = 7 x 7

Index N	Coefficients	Sum = 2^N
0	1	1
1	1 1	2
2	1 2 1	4
3	1 3 3 1	8
4	1 4 6 4 1	16
5	1 5 10 10 5 1	32
6	1 6 15 20 15 6 1	64
7	1 7 21 35 35 21 7 1	128
8	1 8 28 56 70 56 28 8 1	256
9	1 9 36 84 126 126 84 36 9 1	512
10	1 10 45 120 210 252 210 120 45 10 1	1024
11	1 11 55 165 330 462 462 330 165 55 11 1	2048
12	1 12 66 220 495 792 924 792 495 220 66 12 1	4096

$$\begin{bmatrix} 1 & 6 & 15 & 20 & 15 & 6 & 1 \\ 6 & 36 & 90 & 120 & 90 & 36 & 6 \\ 15 & 90 & 225 & 300 & 225 & 90 & 15 \\ 20 & 120 & 300 & 400 & 300 & 120 & 20 \\ 15 & 90 & 225 & 300 & 225 & 90 & 15 \\ 6 & 36 & 90 & 120 & 90 & 36 & 6 \\ 1 & 6 & 15 & 20 & 15 & 6 & 1 \end{bmatrix} 1/4096$$

$$\frac{1}{\sqrt{2\pi}\sigma} = \frac{{}^{N}C_{N/2}}{2^N}$$

$$\frac{1}{64}\begin{bmatrix} 1 \\ 6 \\ 15 \\ 20 \\ 15 \\ 6 \\ 1 \end{bmatrix} \times \frac{1}{64}\begin{bmatrix} 1 & 6 & 15 & 20 & 15 & 6 & 1 \end{bmatrix}$$

What happens if other sampling values are used?

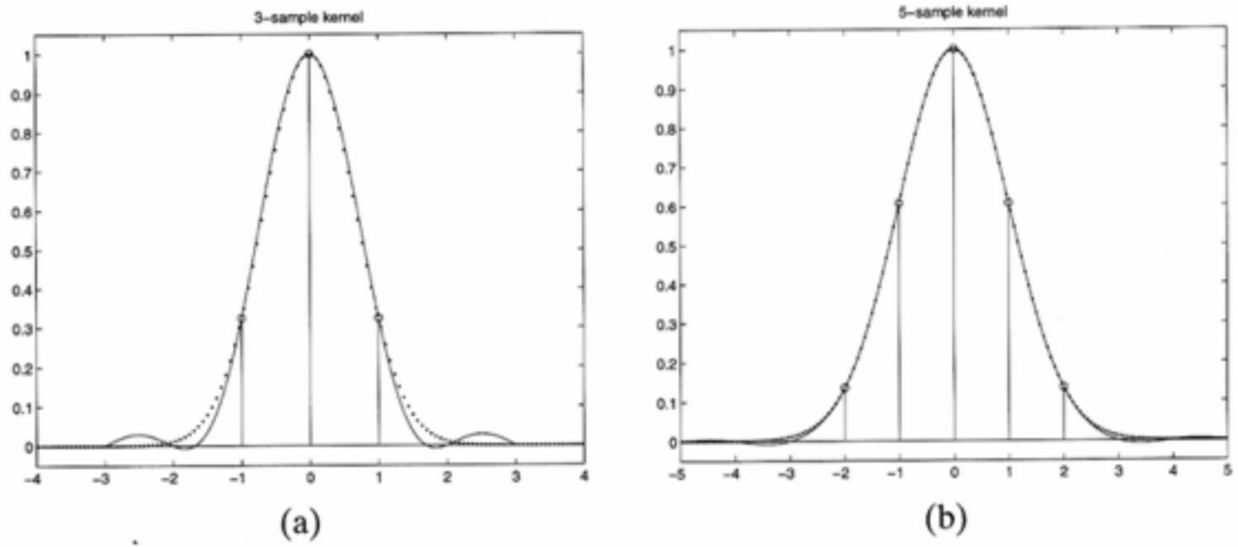
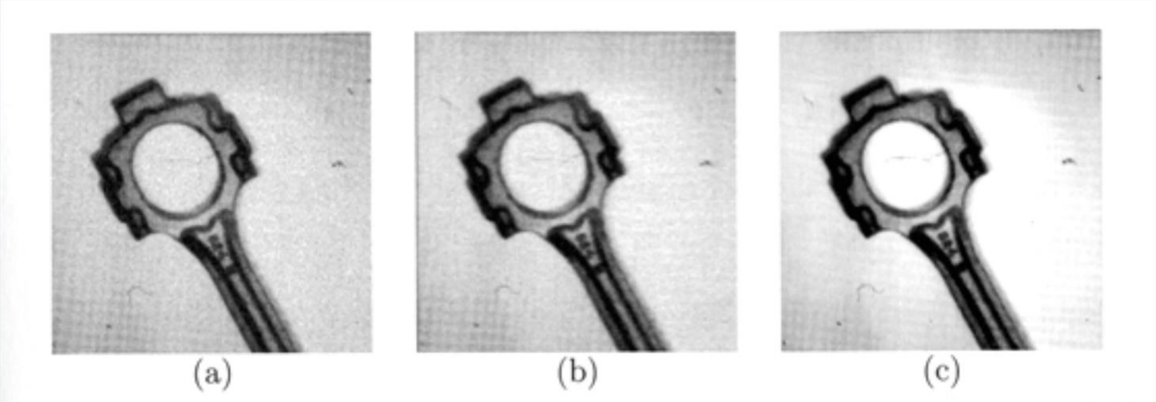


Figure 3.6 Continuous Gaussian kernels (dotted), sampled real kernels, and continuous kernels reconstructed from samples (solid), for $\sigma = 0.6$ ($w = 3$) (a) and $\sigma = 1$ ($w = 5$) (b) respectively.

Image Courtesy: http://ece.uprm.edu/~mtoledo/web/6088/F2018/Slides/6_Filtering.pdf

Example of Gaussian smoothing



Using the fifth row of Pascal's triangle as a Gaussian filter. (a) original; (b) After smoothing in the horizontal dir. (c) After smoothing in the vertical direction

Image Courtesy: http://ece.uprm.edu/~mtoledo/web/6088/F2018/Slides/6_Filtering.pdf

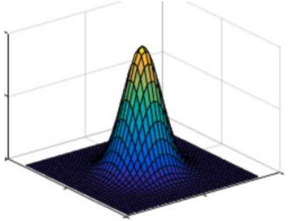
Gaussian Smoothing – Effect of sigma

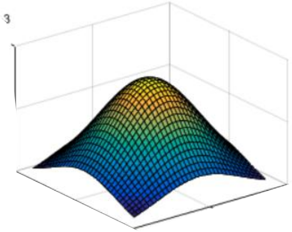
$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\{-(x^2 + y^2)/2\sigma^2\}$$

Original Image
(Sigma 0)

Gaussian Blur
(Sigma 0.7)

Gaussian Blur
(Sigma 2.8)





Edge detection

Essentially what area V1 does in our visual cortex.

- **Goal:** Identify sudden changes (discontinuities) in an image
  - Intuitively, most semantic and shape information from the image can be encoded in the edges
  - More compact than pixels
- **Ideal:** artist's line drawing (but artist is also using object-level knowledge)

Source: D. Lowe

First Derivative (Digital approximation)

$$\frac{\partial f(x,y)}{\partial x} \sim f[x+1,y] - f[x,y]$$

−1	1
$f[x,y]$	$f[x+1,y]$

Second Derivative (Digital Approximation)

$$\frac{\partial^2 f(x,y)}{\partial x^2} \sim (f[x+1,y] - f[x,y]) - (f[x,y] - f[x-1,y])$$

1	−2	1
$f[x-1,y]$	$f[x,y]$	$f[x+1,y]$

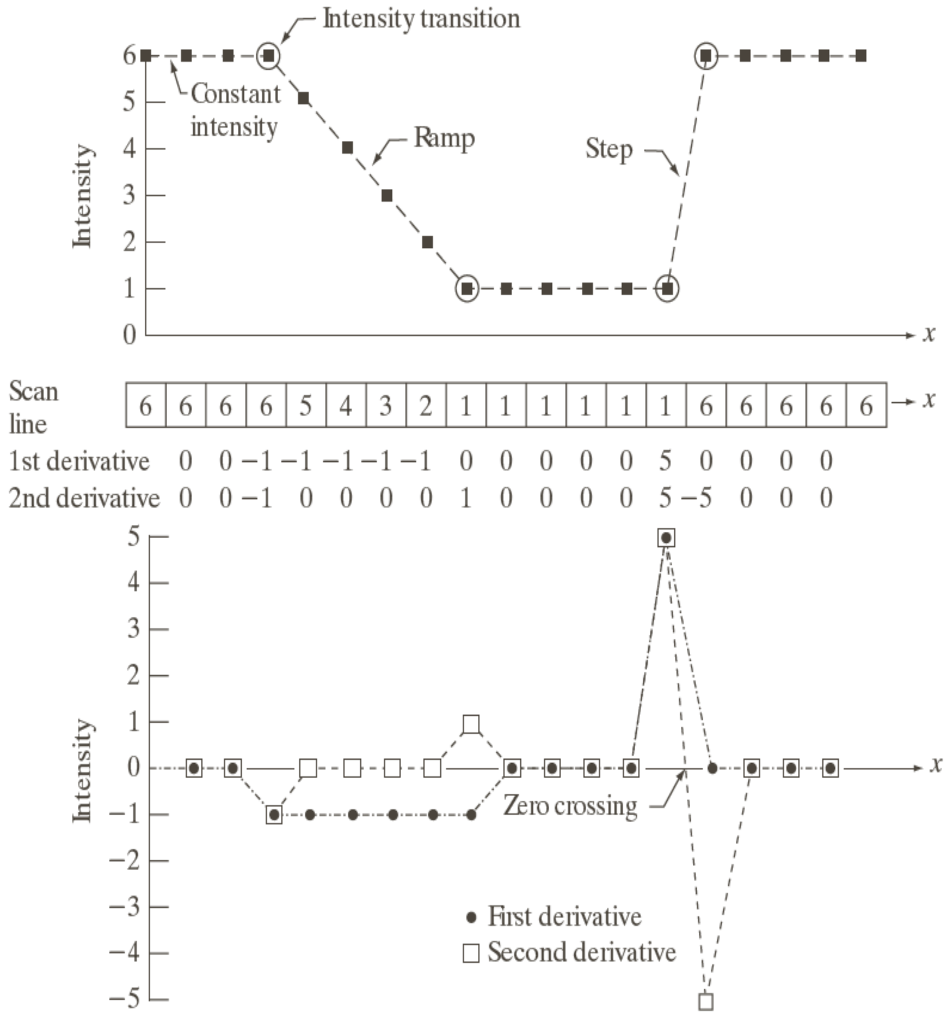


Image Gradient and Edges

$$\frac{f(x+h,y)-f(x-h,y)}{2h} \Rightarrow \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

x-derivative

$$\frac{f(x,y+h)-f(x,y-h)}{2h} \Rightarrow \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

y-derivative

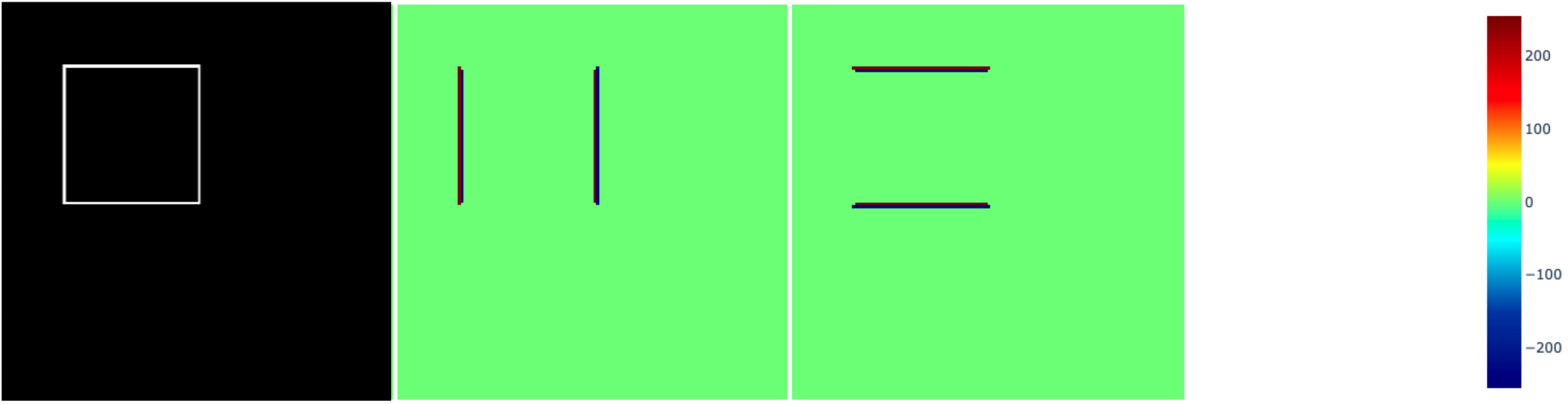


Image x-derivative y-derivative

Prewitt Edge Filter

Dr. Prewitt

https://nihrecord.nih.gov/sites/recordNIH/files/pdf/1984/NIH-Record-1984-03-13.pdf

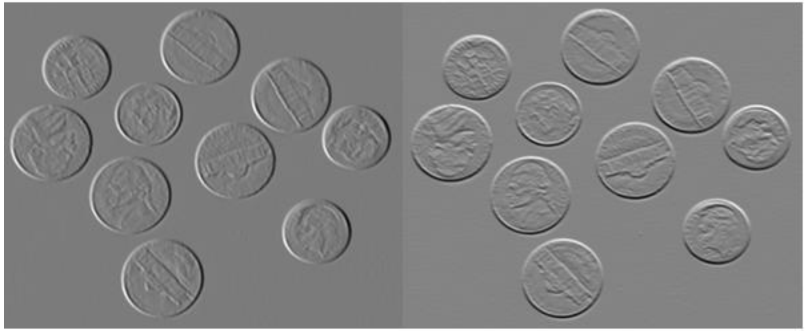
-1	0	+1
-1	0	+1
-1	0	+1

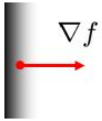
G_x

+1	+1	+1
0	0	0
-1	-1	-1

G_y

Edge is perpendicular to gradient





$\nabla f = \left[\frac{\partial f}{\partial x}, 0\right]$

$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$

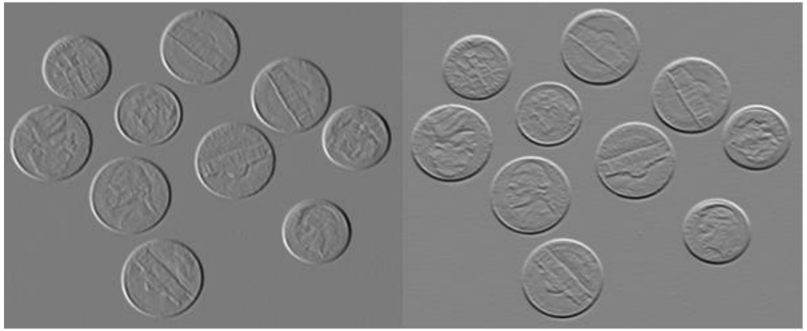
-1	0	+1
-1	0	+1
-1	0	+1

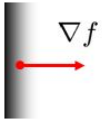
G_x

+1	+1	+1
0	0	0
-1	-1	-1

G_y

Gradient Magnitude and Orientation



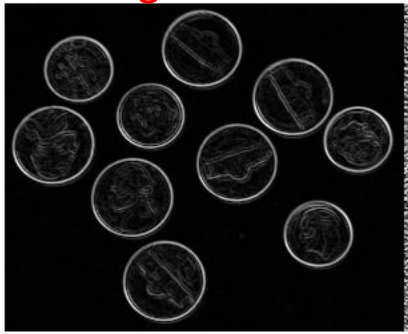


$$\nabla f = \left[\frac{\partial f}{\partial x}, 0\right]$$

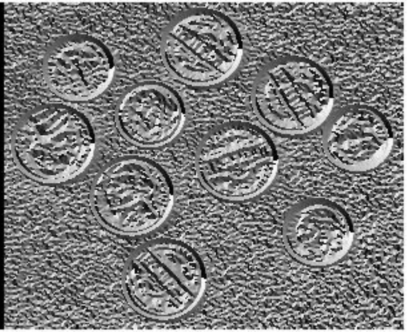
$$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$$

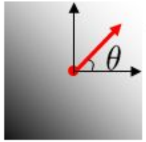
$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Magnitude



Orientation





$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$

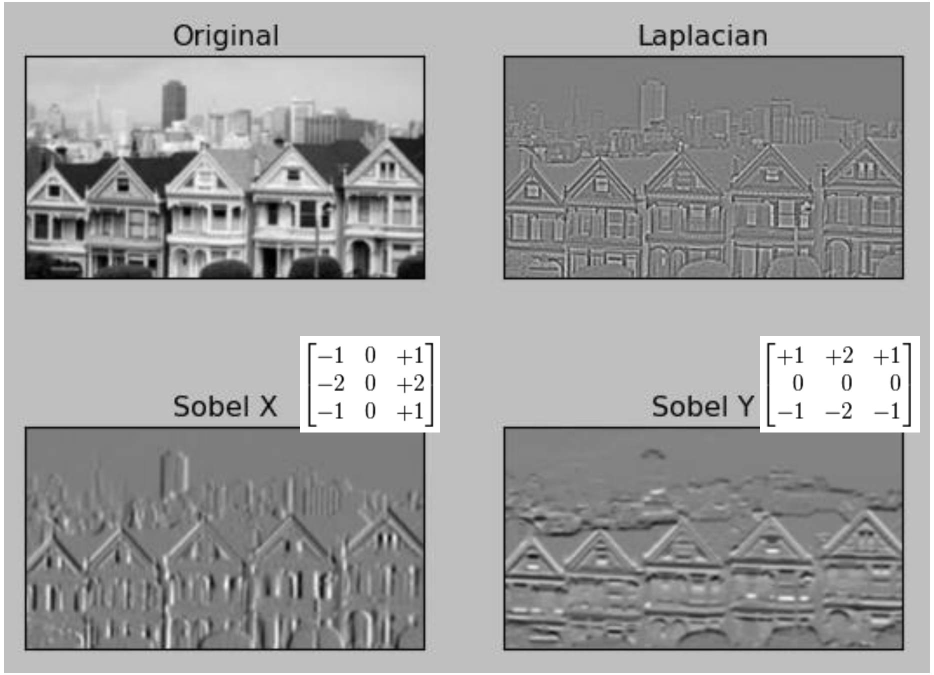
2-D Laplacian Filter

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

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

0	1	0
1	-4	1
0	1	0

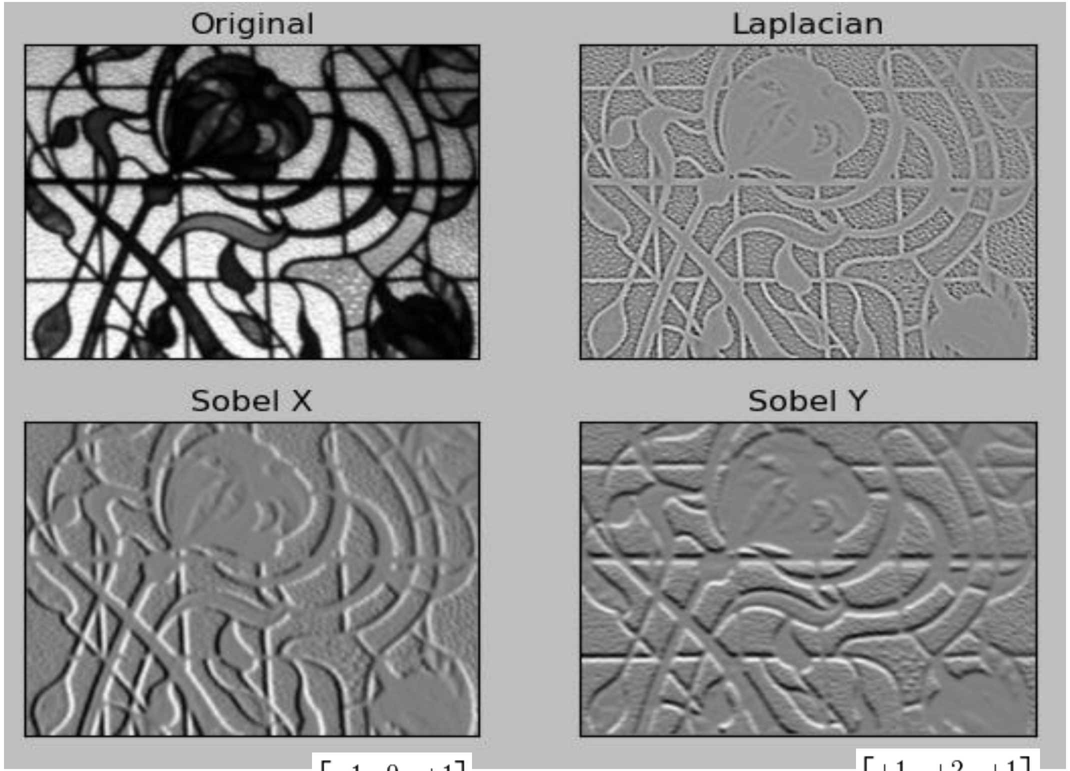
Edge Masks – Sobel , Laplacian



0	−1	0
−1	4	−1
0	−1	0

Note: Coefficients sum to 0

Edge Masks – Sobel , Laplacian



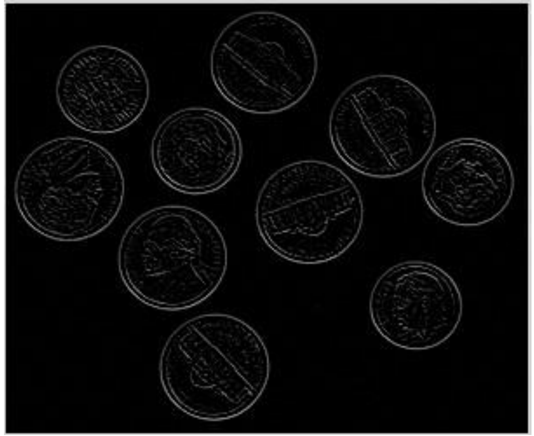
0	−1	0
−1	4	−1
0	−1	0

$$\begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

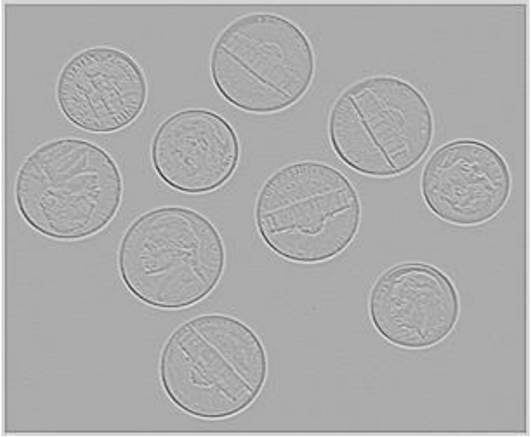
$$\begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Image Sharpening Using Laplacian

$I(u,v)$



$\nabla^2 I(u,v)$



$\nabla^2 I(u,v) + 128$

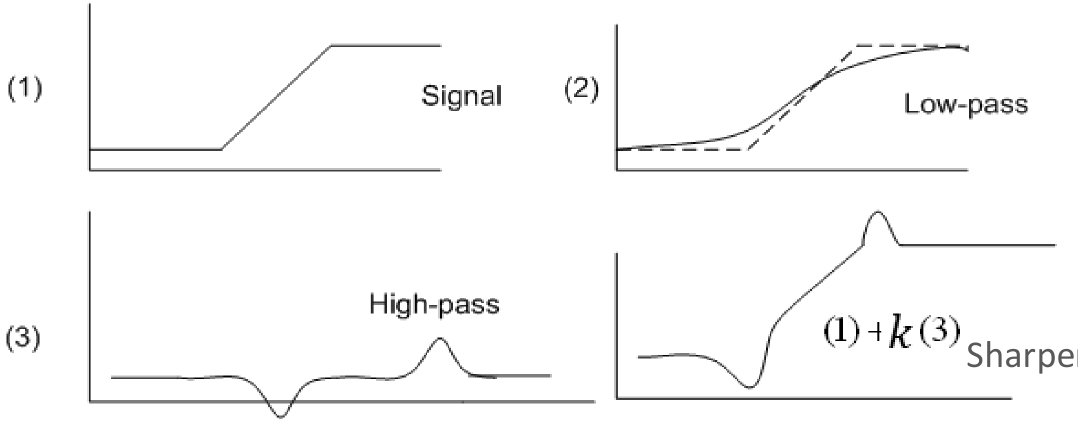
(For visualization)

$I'(u,v) = I(u,v) + \nabla^2 I(u,v)$

Image Sharpening (Unsharp Masking)

1. Blur the original image
2. Subtract the original image from the blurred image (outputs mask)
3. Add the mask back to the original image

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



Sharpened signal

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

Unsharp masking

- What does blurring take away?

$-$

$=$

Difference between original and smoothed images

- Let's add it back:

$+ \; k$

$=$

Add original and weighted detail images

Unsharp Masking vs Highboost Filtering

$k \geq 0 \ (k = 1)$

$k > 1$

Unsharp Masking / Highboost Filtering as Spatial Filters

A=1

w = 9A-1

-1	-1	-1
-1	w	-1
-1	-1	-1

A=2

w = 17

-1	-1	-1
-1	17	-1
-1	-1	-1

- If **A=1**, we get unsharp masking. $I'(u,v) = I(u,v) + \nabla^2 I(u,v)$
- If **A>1**, original image is added back to detail image (highboost filtering).

Corner cases, Padding

```
M = 3
For each valid location [x,y] in S
        a ← Average of intensities in a M x M neighborhood centered on [x,y]
        D[x,y] = round(a)
```

120	190	140	150	200
17	21	30	8	27
89	123	150	73	56
10	178	140	150	18
190	14	76	69	87

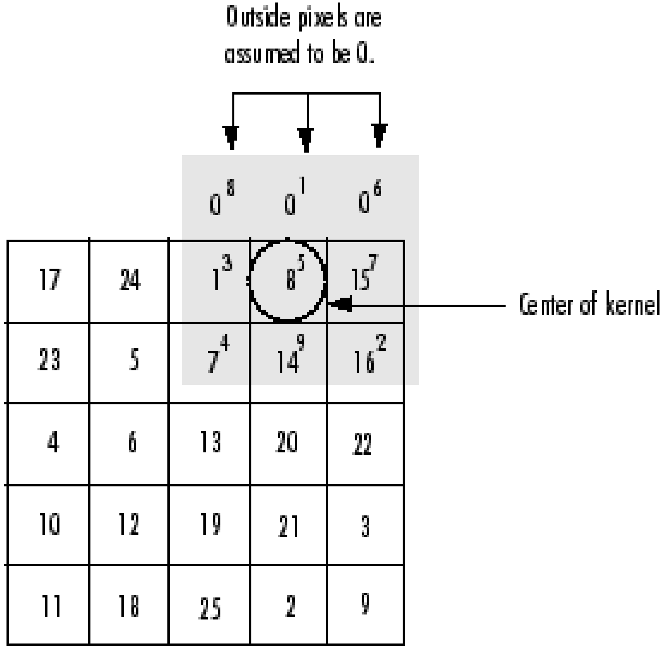
x

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

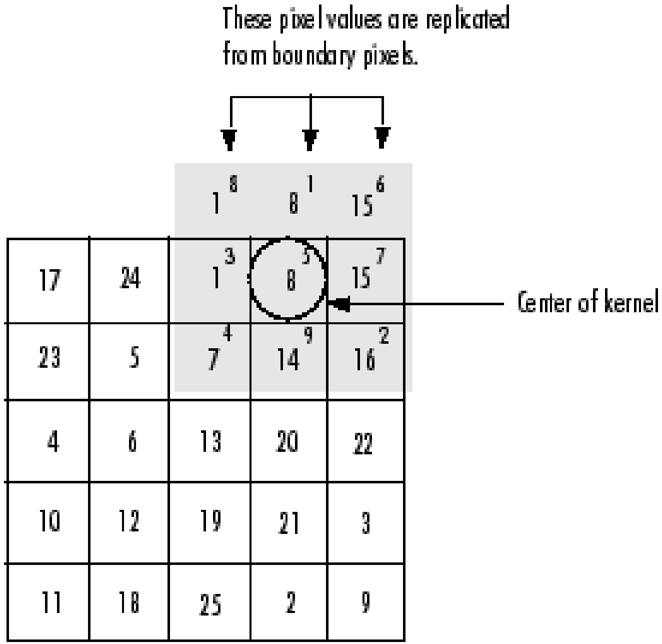
=

	98			

Image Padding



zero



replicate

Edge Detection-Biometrics

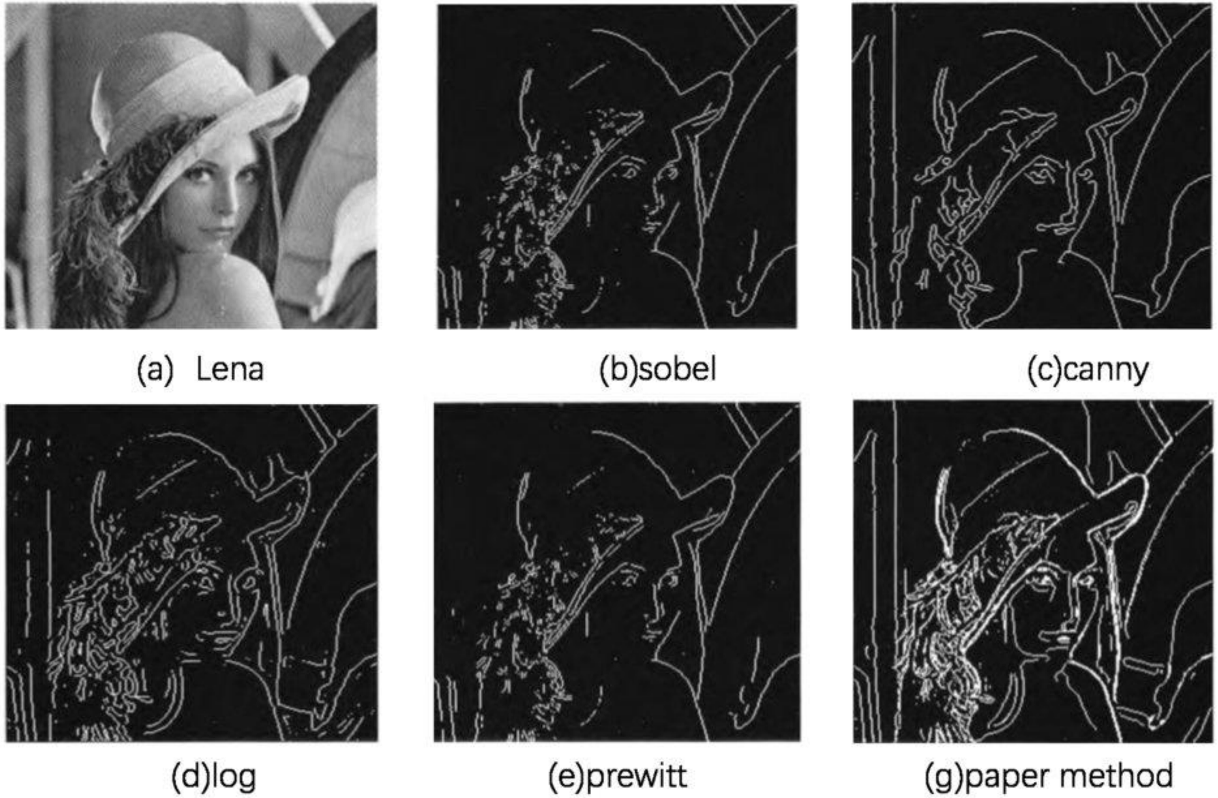


Image Courtesy: Hua et al., “An edge detection method with boundary reserved based on non-subsampled contourlet transform for remote sensing imagery,” Optics and Photonics for Information Processing XI, SPIE, 2017

Edge Detection-Biometrics

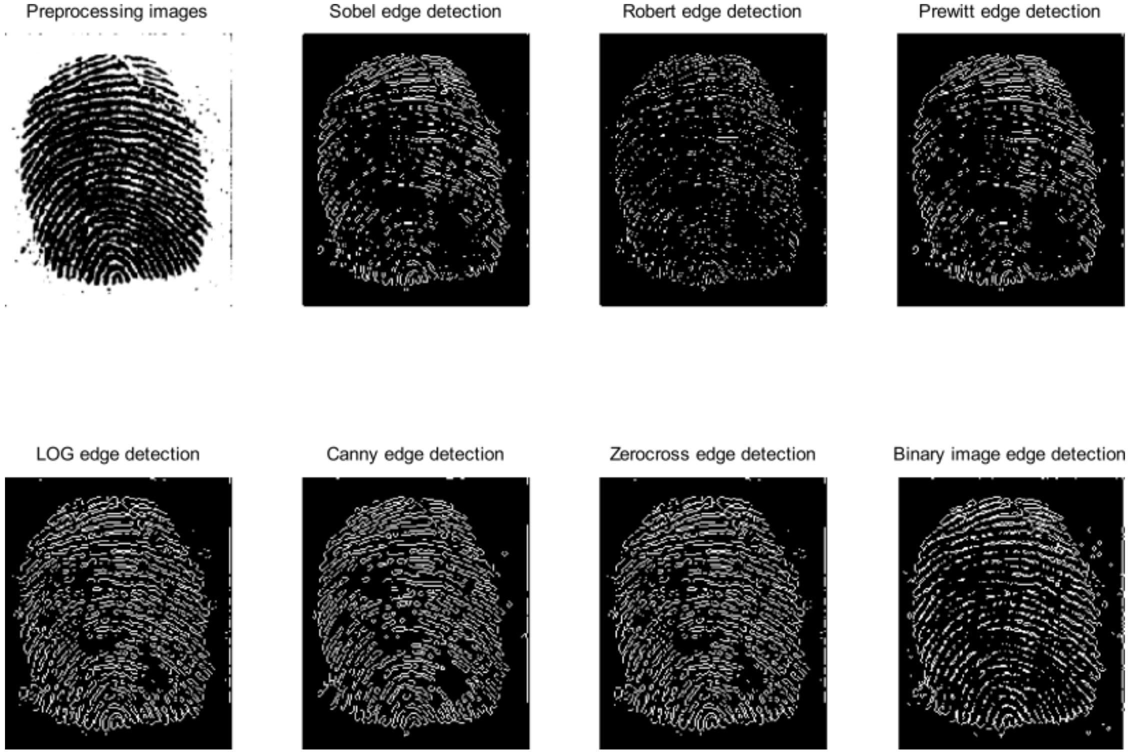


Image Courtesy: Cui et al., "The research of edge detection algorithm for fingerprint images," World Automation Congress, 2008

Edge Detection-Biometrics

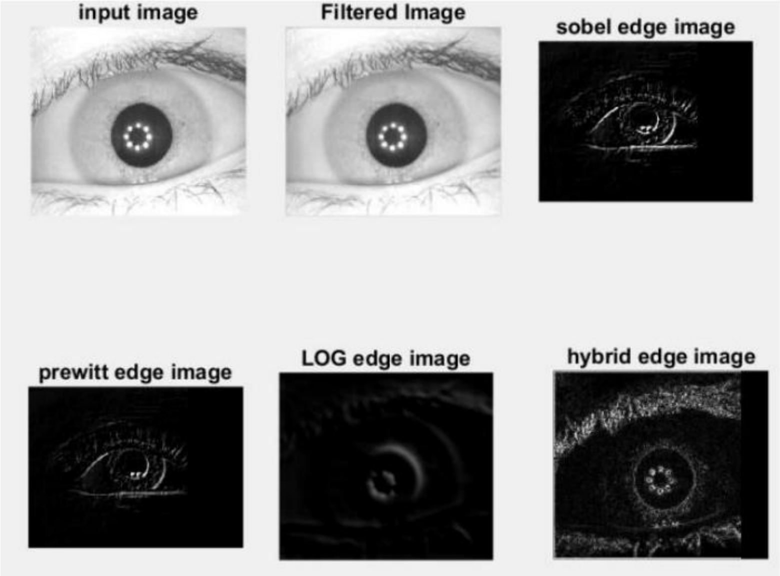


Image Courtesy: Kaur et al., “Comparison of edge detection techniques for iris recognition,” Intl. Journal of Computer Applications, 2016

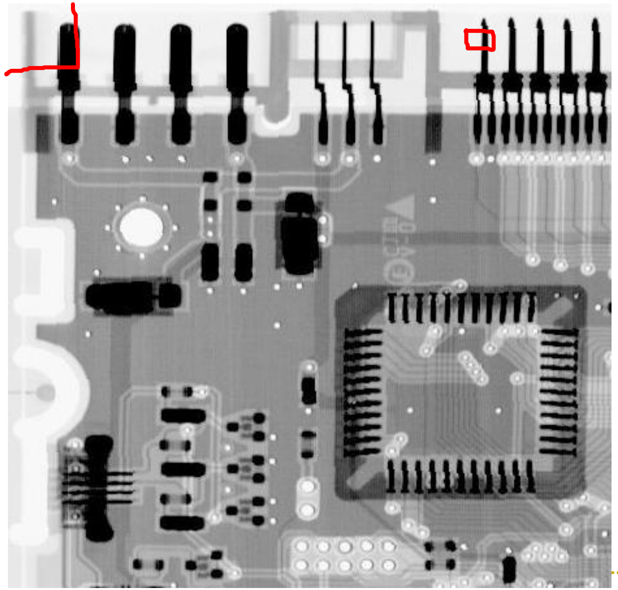
References

- GW Chapter – 3.4.1, 3.5.1,3.6

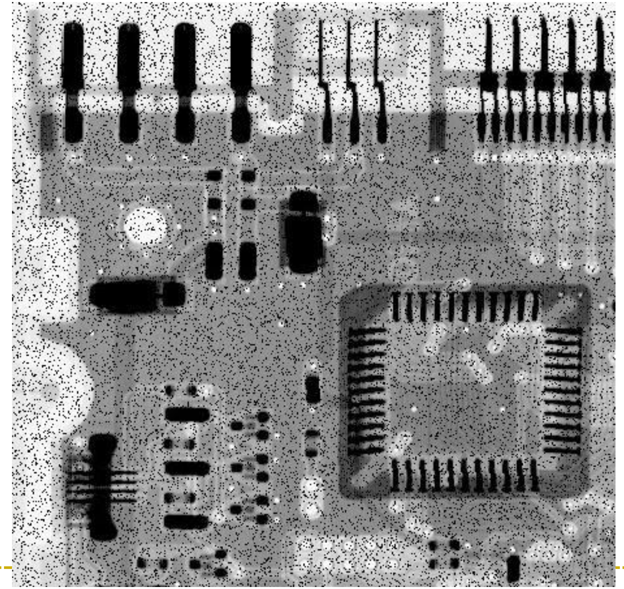
Spatial Domain Filtering - Approaches

- Linear (Average, Gaussian, Prewitt, Sobel, Laplacian)
- Non-linear

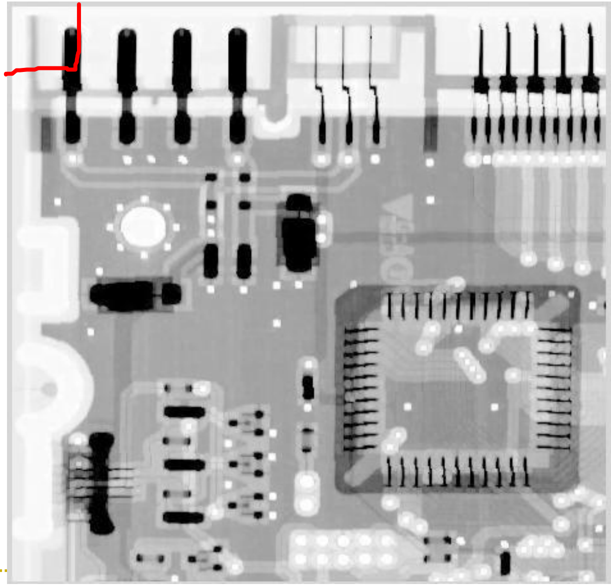
Non-linear Spatial Filters (max)



pepper noise



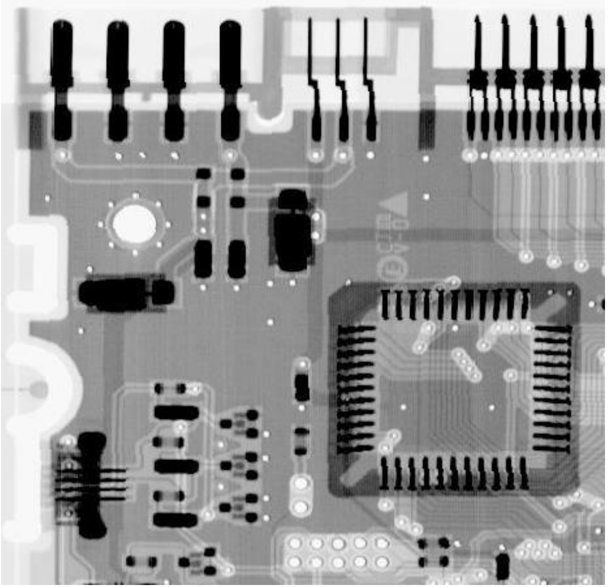
After applying max filter

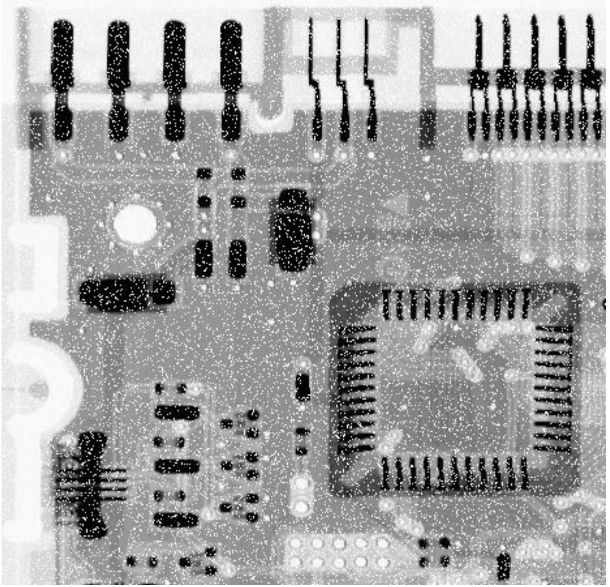


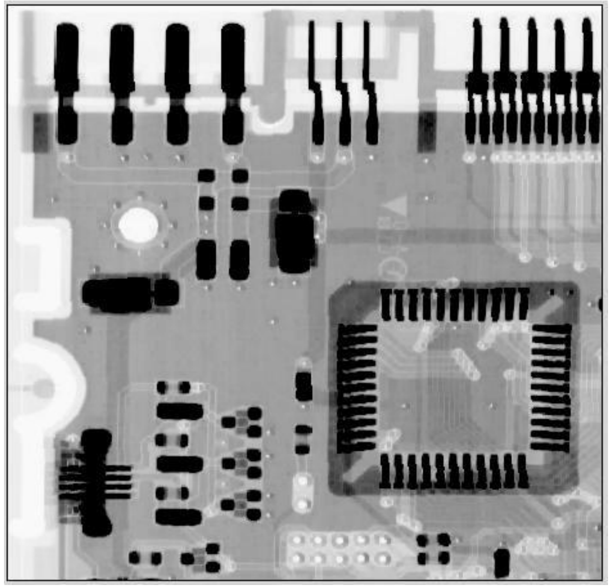
Non-linear Spatial Filters (min)

salt noise

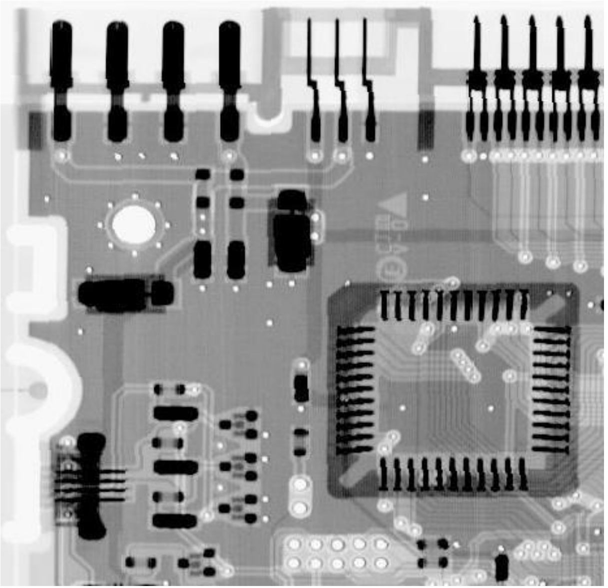
After applying min filter



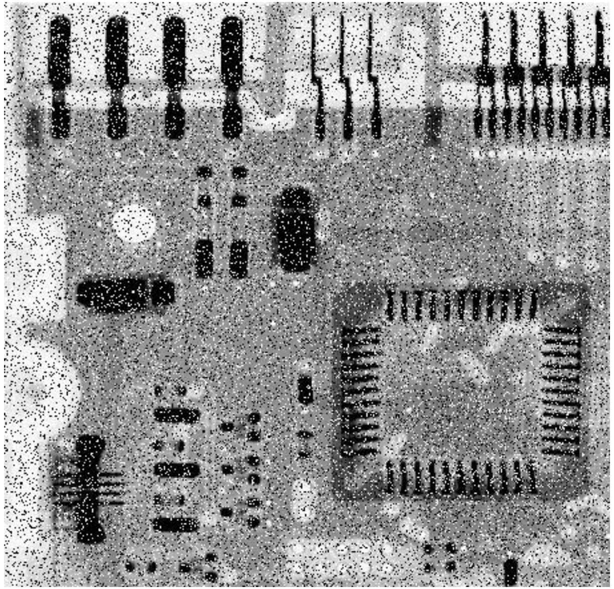




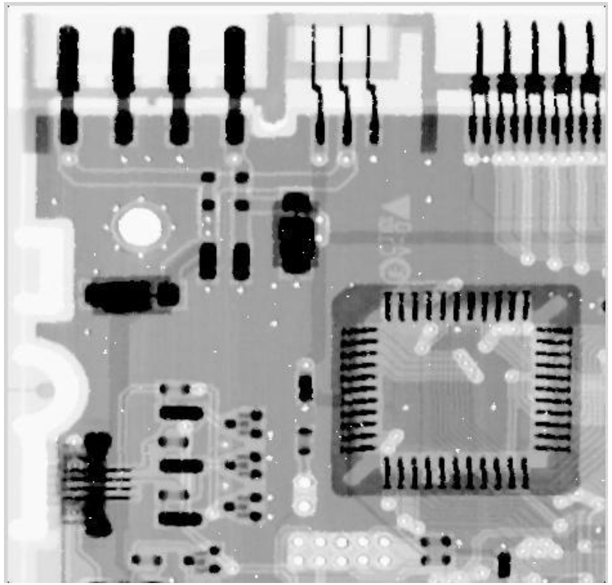
Non-linear Spatial Filters (median)



salt & pepper noise



After applying median filter



max, min, median → also known as rank / order statistic filters

Other Spatial Filters

- Geometric mean
- Harmonic mean
- Contra harmonic mean
- Mid Point filter
- Alpha trimmed mean filter
- …..

Bilateral Filtering (Edge preserving smoothing)

Original image taken from cs.cityu.edu.hk

References

- GW Chapter – 3.4