

01.09.2020

Digital Image Processing (CSE/ECE 478)

Lecture-7: Spatial Filtering

Ravi Kiran



Center for Visual Information Technology (CVIT), IIIT Hyderabad

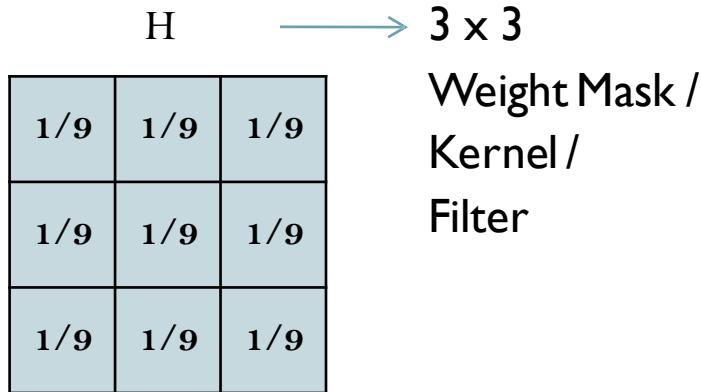
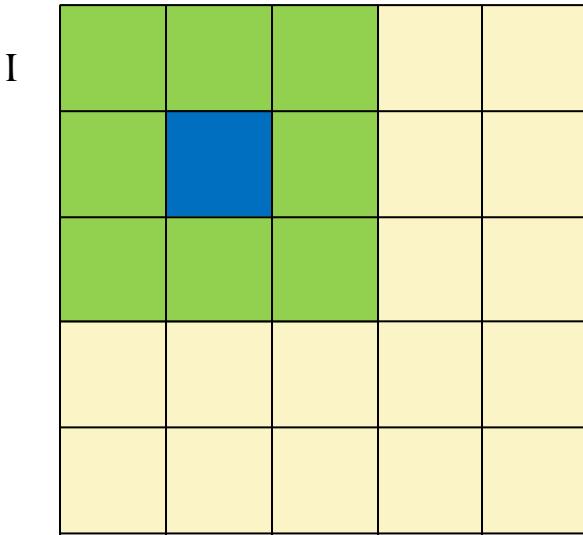


Announcements

- Mini Quiz – 2 today (hopefully !)

Mean/Average Filter

Note: Coefficients sum to 1



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

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

Effect of Mask Size

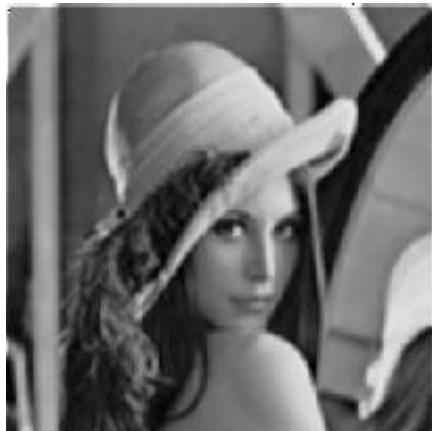
Original Image



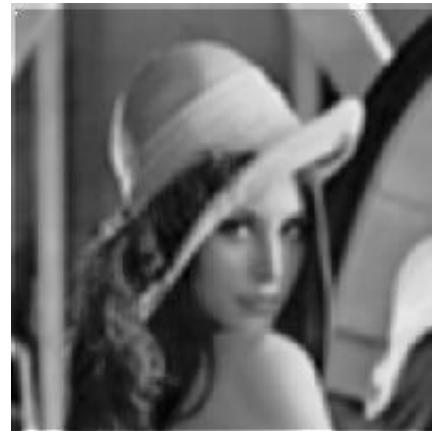
[3x3]



[5x5]



[7x7]



Repeated Averaging Using Same Filter



Before



After



After repeated
averaging

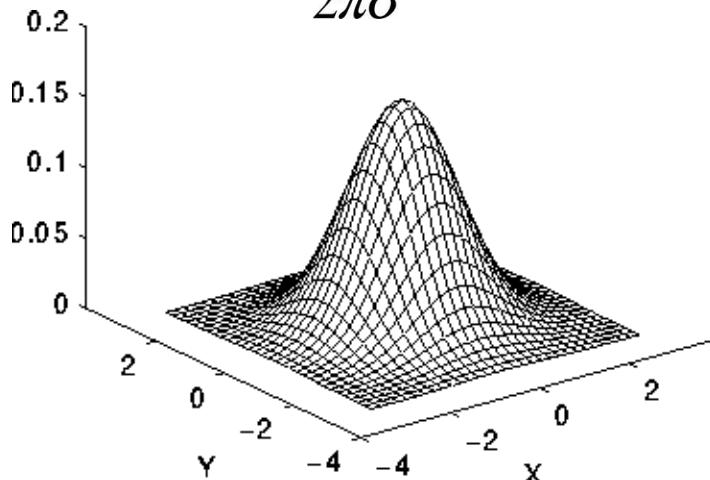
>

NOTE: Can get the effect of larger filters by smoothing repeatedly with smaller filters

Gaussian Smoothing

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

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



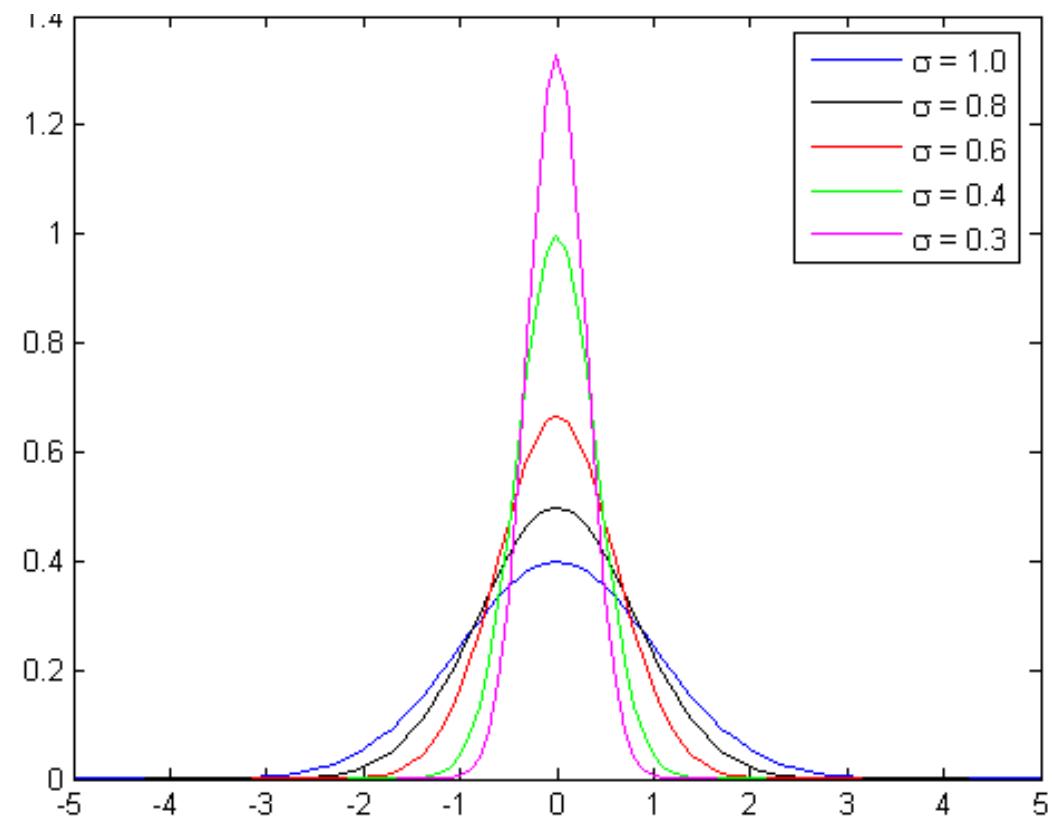
$$\frac{1}{265}$$

256

1	4	6	4	1
4	16	26	16	4
6	26	43	26	6
4	16	26	16	4
1	4	6	4	1

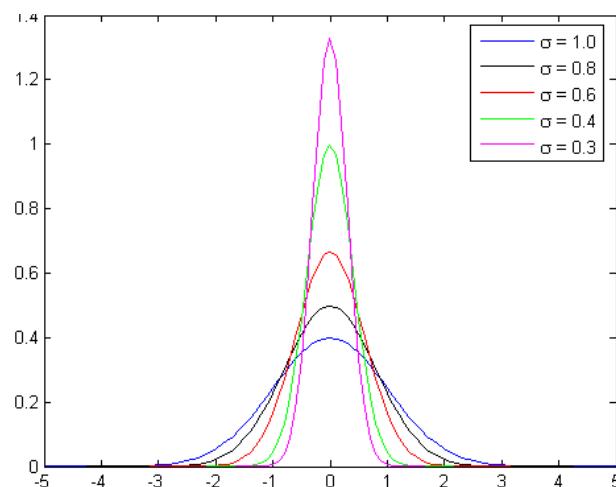
5×5 Gaussian filter, $\sigma=1$

How are Gaussian filter coefficients obtained ?

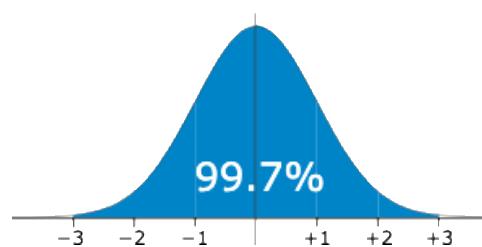
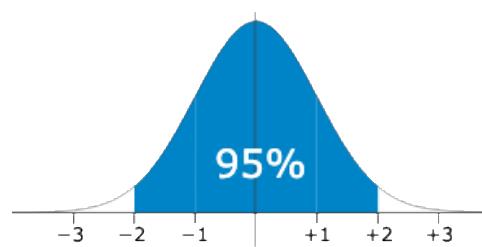
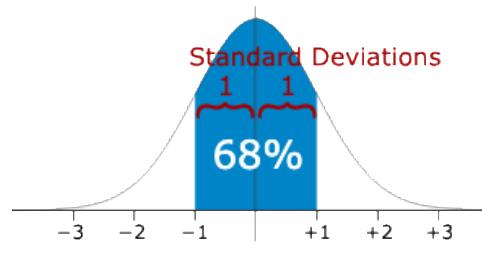


$$g_{\sigma}(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

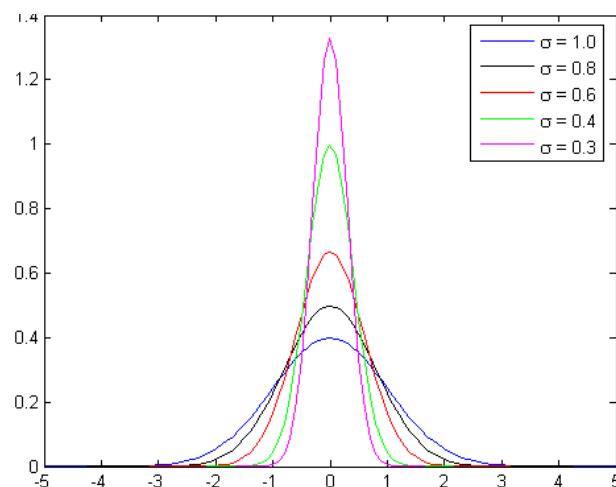
How are Gaussian filter coefficients obtained ?



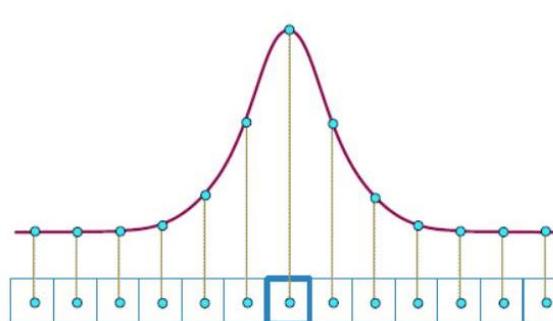
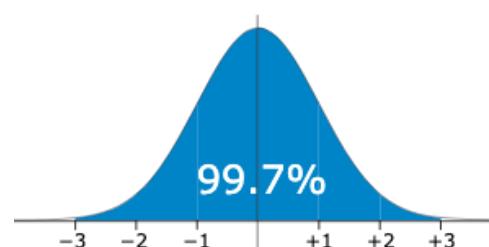
$$g_{\sigma}(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$



How are Gaussian filter coefficients obtained ?



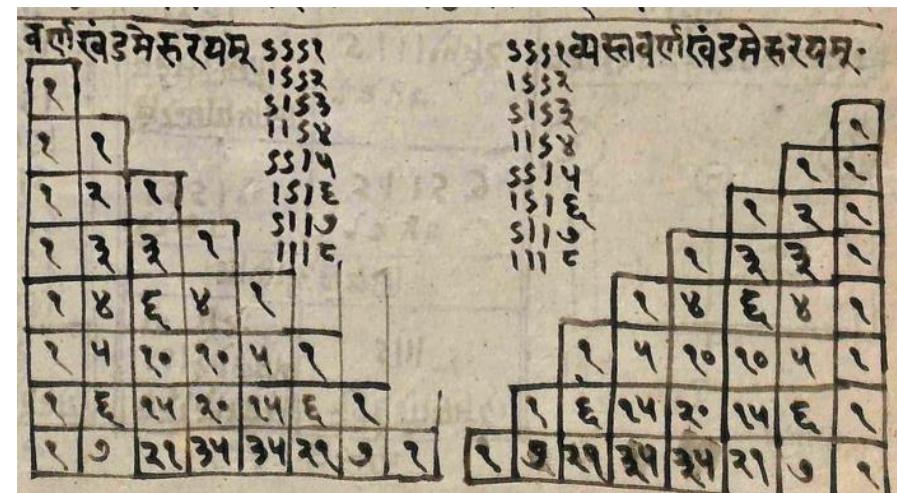
$$g_{\sigma}(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$



How are Gaussian filter coefficients obtained ?

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

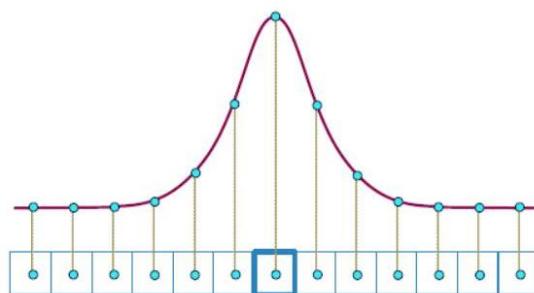


How are Gaussian filter coefficients obtained ?

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

E.g. $s = 7 \times 7$

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

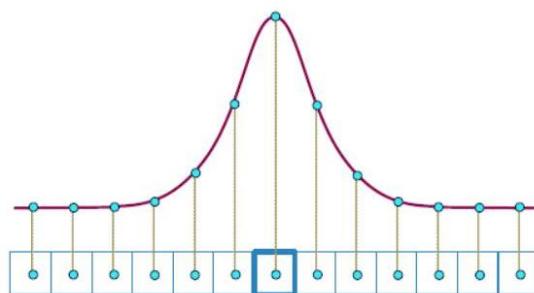


How are Gaussian filter coefficients obtained ?

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

$S = 7 \times 7$

Index N	Coefficients	Sum = 2^N	
0	1	1	1/64
1	1 1	2	1
2	1 2 1	4	6
3	1 3 3 1	8	15
4	1 4 6 4 1	16	20
5	1 5 10 10 5 1	32	15
6	1 6 15 20 15 6 1	64	6
7	1 7 21 35 35 21 7 1	128	1
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	



How are Gaussian filter coefficients obtained ?

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

$S = 7 \times 7$

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

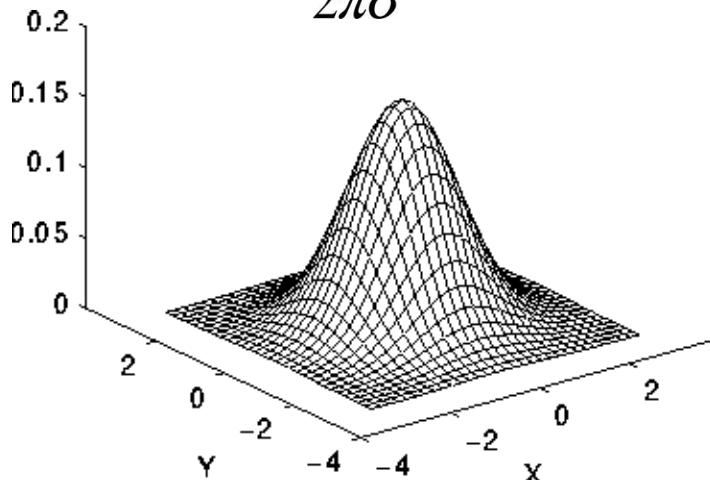
$$\begin{bmatrix} 1 & 6 & 15 & 20 & 15 & 6 & 1 \\ 6 & 36 & 90 & 120 & 90 & 36 & 6 \\ 15 & 90 & 225 & 300 & 225 & 90 & 15 \\ 20 & 120 & 300 & 400 & 300 & 120 & 20 \\ 15 & 90 & 225 & 300 & 225 & 90 & 15 \\ 6 & 36 & 90 & 120 & 90 & 36 & 6 \\ 1 & 6 & 15 & 20 & 15 & 6 & 1 \end{bmatrix}$$

1/4096

Gaussian Smoothing

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

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



$$\frac{1}{256}$$

1	4	6	4	1
4	16	26	16	4
6	26	43	26	6
4	16	26	16	4
1	4	6	4	1

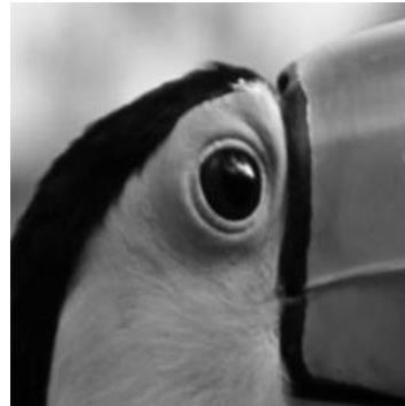
5×5 Gaussian filter, $\sigma=1$

Gaussian Smoothing – Effect of sigma

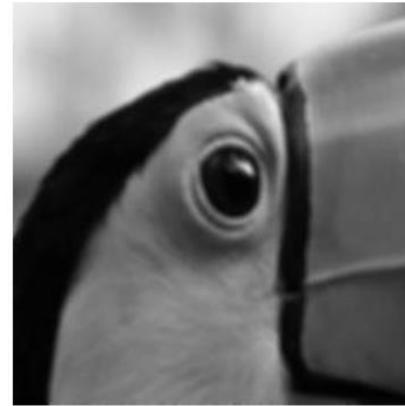
$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\{-(x^2 + y^2)/2\sigma^2\}$$



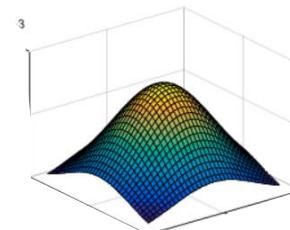
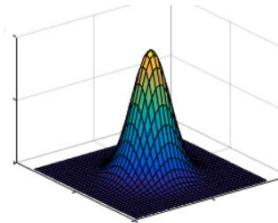
Original Image
(Sigma 0)



Gaussian Blur
(Sigma 0.7)



Gaussian Blur
(Sigma 2.8)



Edge detection

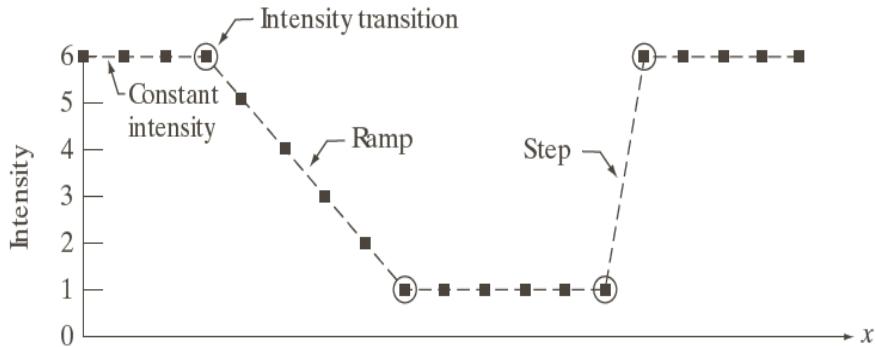
- **Goal:** Identify sudden changes (discontinuities) in an image
 - Intuitively, most semantic and shape information from the image can be encoded in the edges
 - More compact than pixels
- **Ideal:** artist's line drawing (but artist is also using object-level knowledge)

Essentially what area V1 does in our visual cortex.



► First Derivative (Digital approximation)

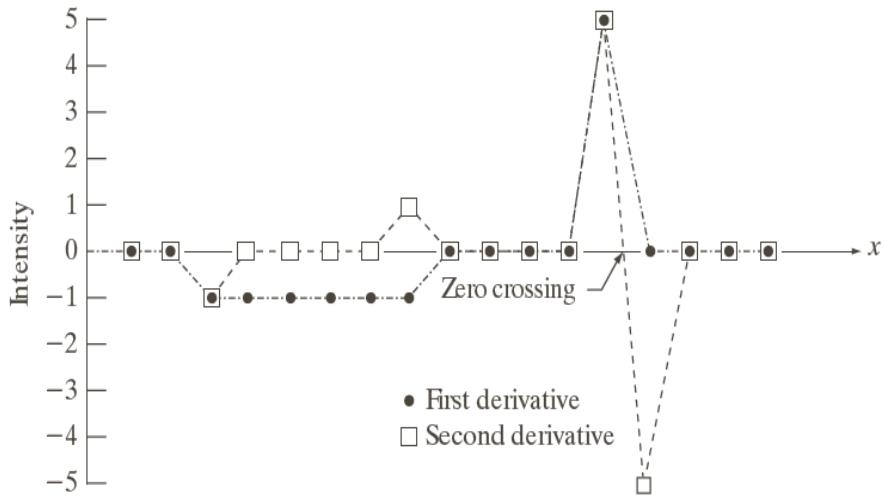
$$\frac{\partial f(x, y)}{\partial x} \sim f[x + 1, y] - f[x, y]$$



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	0	5	0	0	0	0	
2nd derivative	0	0	-1	0	0	0	0	1	0	0	0	0	0	5	-5	0	0	0	

► Second Derivative (Digital Approximation)

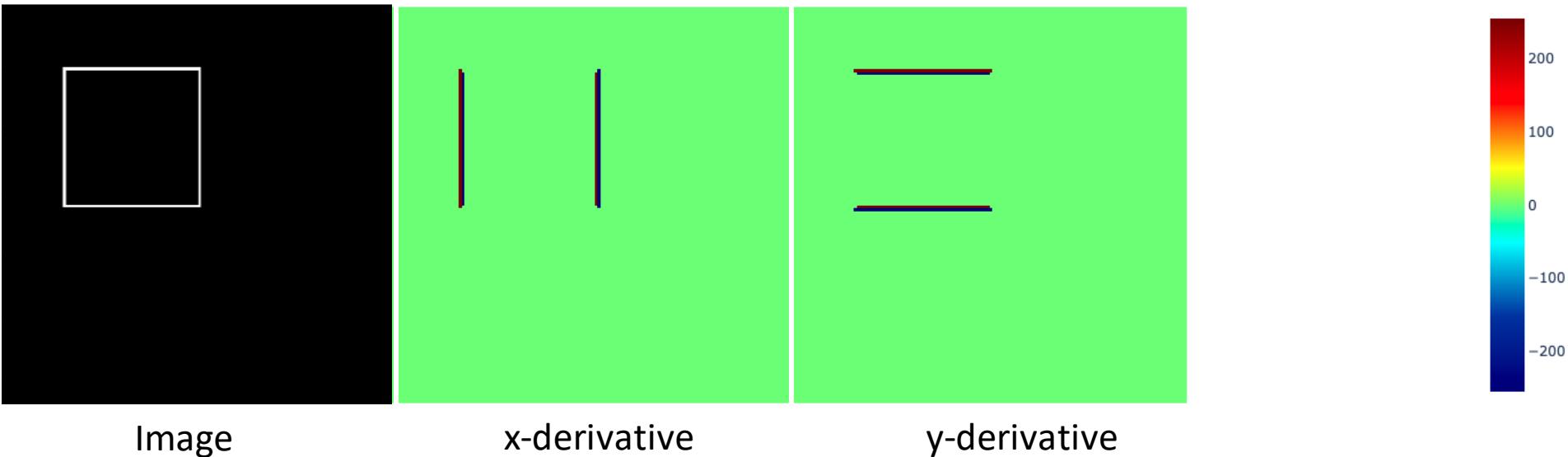
$$\frac{\partial^2 f(x, y)}{\partial x^2} \sim (f[x + 1, y] - f[x, y]) - (f[x, y] - f[x - 1, y])$$



$$\frac{f(x+h,y) - f(x-h,y)}{2h} \rightarrow \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline\end{array}$$

$$\frac{f(x,y+h) - f(x,y-h)}{2h} \rightarrow \begin{array}{|c|}\hline -1 \\ \hline 0 \\ \hline 1 \\ \hline\end{array}$$

Image Gradient and Edges





Dr. Prewitt

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

Prewitt Edge Filter

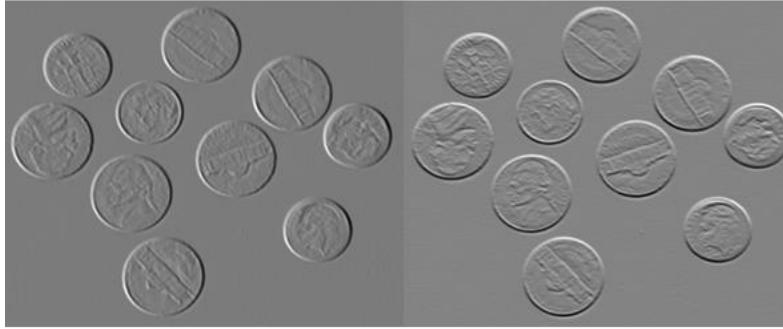
$$\begin{array}{|c|c|c|} \hline -1 & 0 & +1 \\ \hline -1 & 0 & +1 \\ \hline -1 & 0 & +1 \\ \hline \end{array}$$

G_x

$$\begin{array}{|c|c|c|} \hline +1 & +1 & +1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$$

G_y

Edge is perpendicular to gradient



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

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

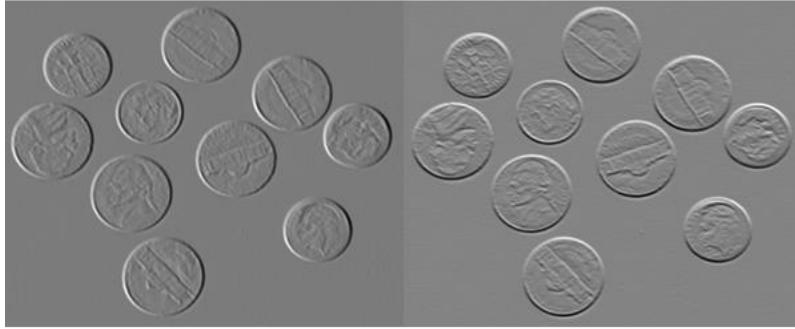
-1	0	+1
-1	0	+1
-1	0	+1

G_x

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

G_y

Gradient Magnitude and Orientation



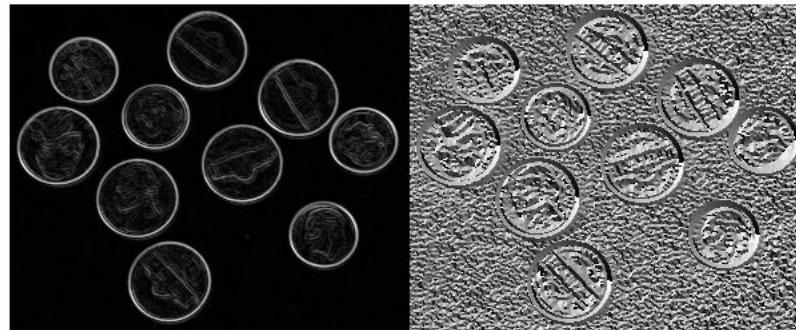
$$\nabla f = \left[\frac{\partial f}{\partial x}, 0 \right]$$

A small grayscale image showing a horizontal gradient from light to dark. A red arrow points horizontally to the right, indicating the direction of the gradient.

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

A small grayscale image showing a vertical gradient from light to dark. A red arrow points vertically downwards, indicating the direction of the gradient.

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



A small grayscale image showing a diagonal gradient from top-left to bottom-right. A red arrow points diagonally upwards and to the right, indicating the direction of the gradient.

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

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

2-D Laplacian Filter

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

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

0	1	0
1	-4	1
0	1	0

Edge Masks – Sobel , Laplacian

Original



Laplacian



0	-1	0
-1	4	-1
0	-1	0

Note: Coefficients sum to 0

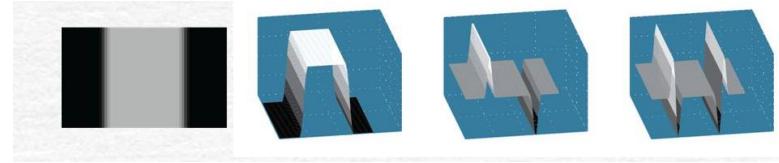
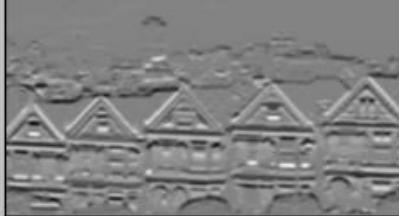
Sobel X

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



Sobel Y

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



Edge Masks – Sobel , Laplacian

Original



Laplacian



0	-1	0
-1	4	-1
0	-1	0

Sobel X



Sobel Y



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

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

Image Sharpening

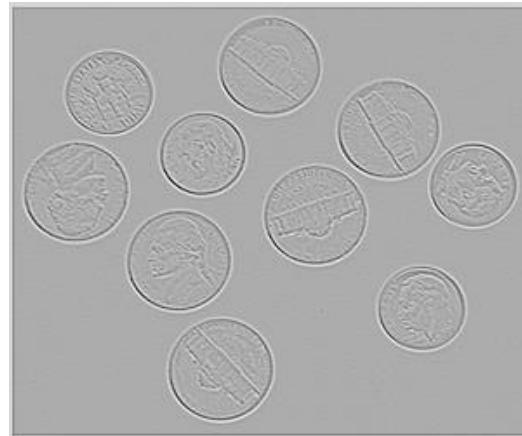
$$\nabla^2 I(u, v)$$

$$I(u, v)$$



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

(For visualization)



$$I'(u, v)$$



Sharpening (Unsharp Masking)

$$I(u, v)$$



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



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

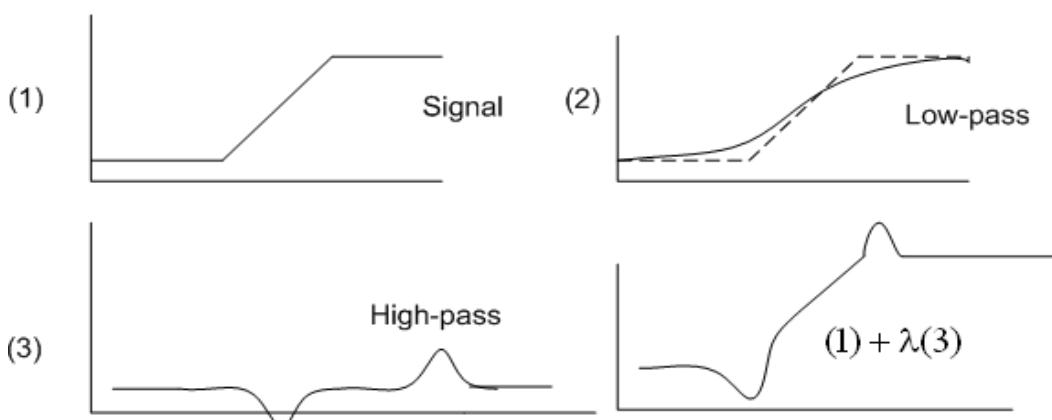
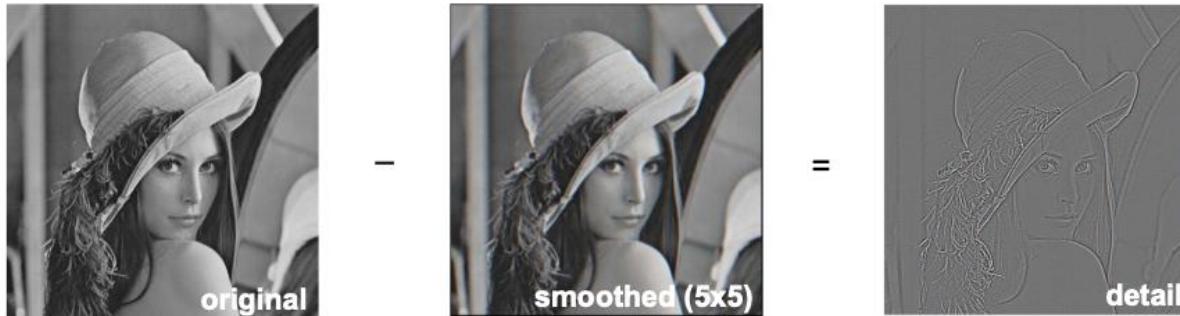


Image Courtesy: NASA

Highboost Filtering

- What does blurring take away?



- Let's add it back:



Unsharp Masking vs Highboost Filtering



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

$a \leftarrow$ Average of intensities in a $M \times M$ 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

$$x \begin{pmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{pmatrix} = \begin{pmatrix} & & & \\ & 98 & & \\ & & & \\ & & & \\ & & & \\ & & & \end{pmatrix}$$

Image Padding

Outside pixels are assumed to be 0.

The diagram illustrates image padding where outside pixels are assumed to be 0. A 3x3 kernel is applied to a 5x5 input image. The input image has values: Row 1: 17, 24, 1³, 8⁵, 15⁷; Row 2: 23, 5, 7⁴, 14⁹, 16²; Row 3: 4, 6, 13, 20, 22; Row 4: 10, 12, 19, 21, 3; Row 5: 11, 18, 25, 2, 9. The kernel has values: Row 1: 0⁸, 0¹, 0⁶; Row 2: 1⁸, 8¹, 15⁶. The center of the kernel is highlighted with a circle around the value 8⁵. Arrows point from the kernel's center to the corresponding input pixel 8⁵ and the output value 15⁷. A callout points to the input pixel 8⁵ with the text "Center of kernel".

17	24	1 ³	8 ⁵	15 ⁷
23	5	7 ⁴	14 ⁹	16 ²
4	6	13	20	22
10	12	19	21	3
11	18	25	2	9

zero

These pixel values are replicated from boundary pixels.

The diagram illustrates image padding where pixel values are replicated from boundary pixels. A 3x3 kernel is applied to a 5x5 input image. The input image has values: Row 1: 17, 24, 1³, 8⁵, 15⁷; Row 2: 23, 5, 7⁴, 14⁹, 16²; Row 3: 4, 6, 13, 20, 22; Row 4: 10, 12, 19, 21, 3; Row 5: 11, 18, 25, 2, 9. The kernel has values: Row 1: 1⁸, 8¹, 15⁶; Row 2: 1⁸, 8¹, 15⁶. The center of the kernel is highlighted with a circle around the value 8⁵. Arrows point from the kernel's center to the corresponding input pixel 8⁵ and the output value 15⁷. A callout points to the input pixel 8⁵ with the text "Center of kernel".

17	24	1 ³	8 ⁵	15 ⁷
23	5	7 ⁴	14 ⁹	16 ²
4	6	13	20	22
10	12	19	21	3
11	18	25	2	9

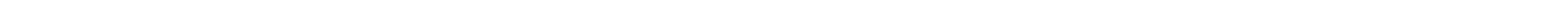
replicate

References

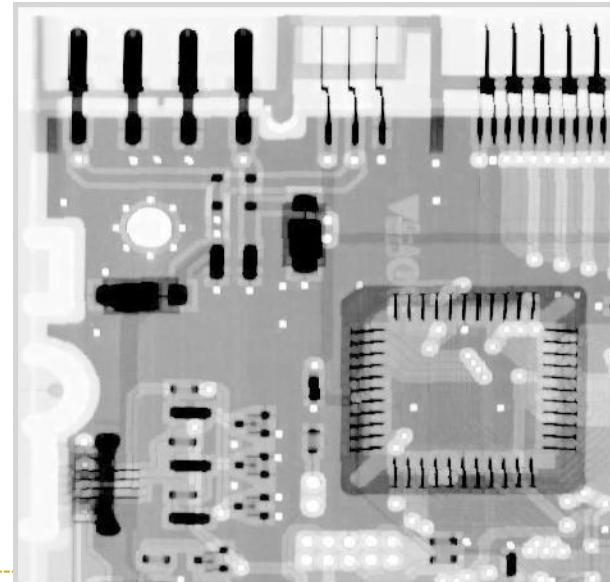
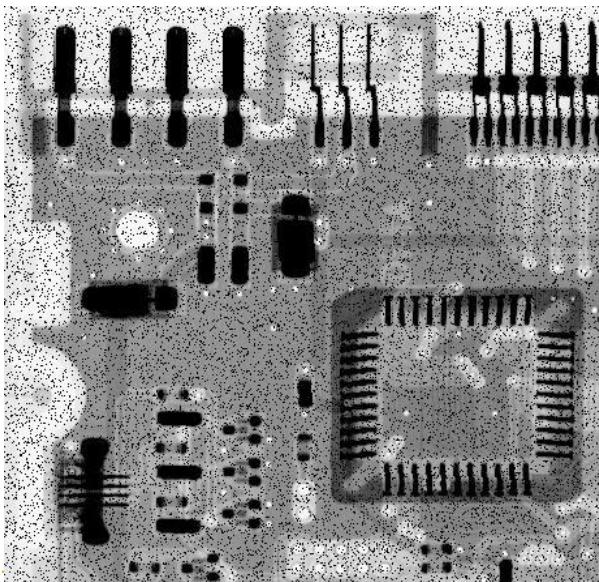
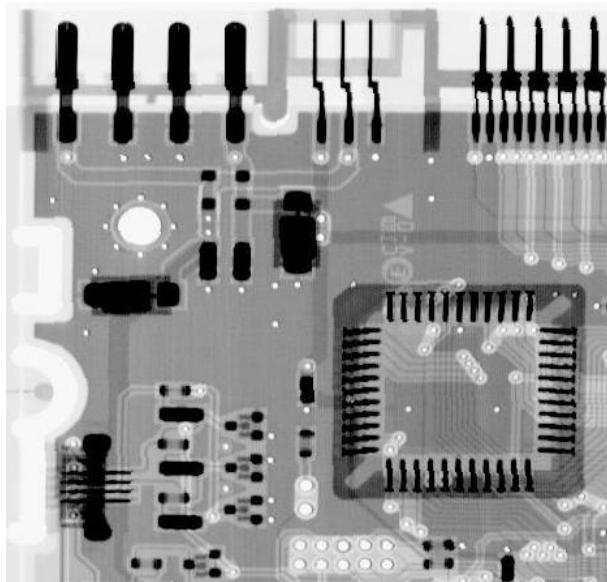
- ▶ **GW Chapter – 3.4.1, 3.5.1, 3.6**

Spatial Domain Filtering - Approaches

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

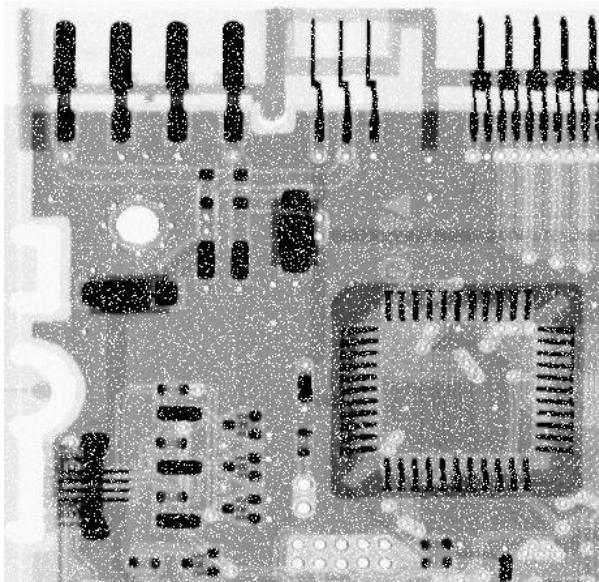
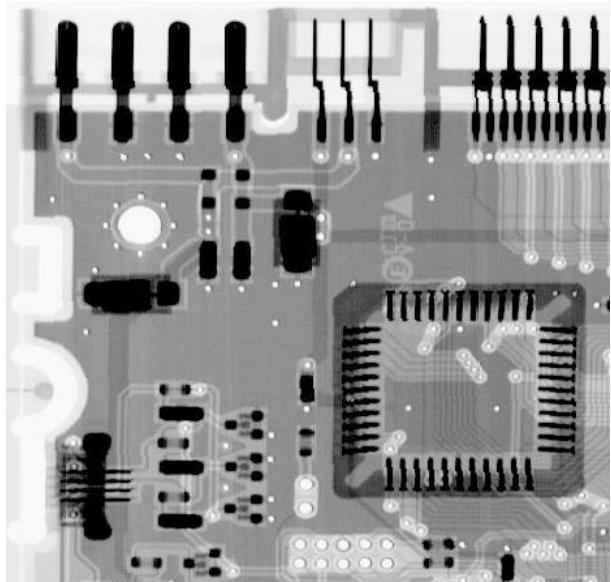


Non-linear Spatial Filters (max)

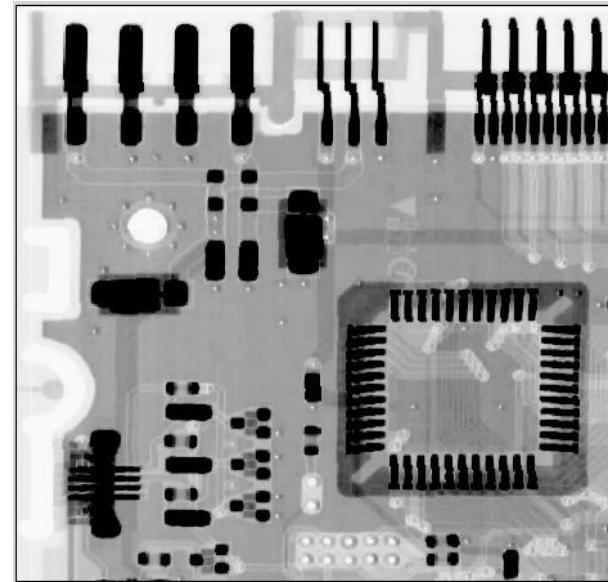


Non-linear Spatial Filters (min)

salt noise

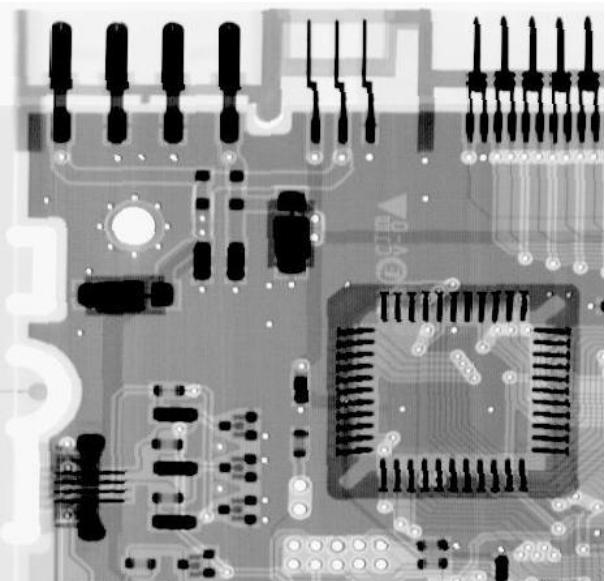


After applying min filter

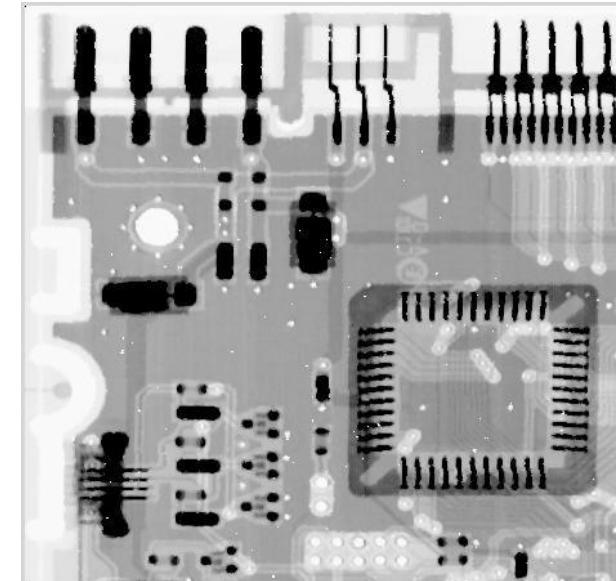


Non-linear Spatial Filters (median)

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

