01.09.2020

Digital Image Processing (CSE/ECE 478)

Lecture-7: Spatial Filtering

Ravi Kiran

Center for Visual Information Technology (CVIT), IIIT Hyderabad

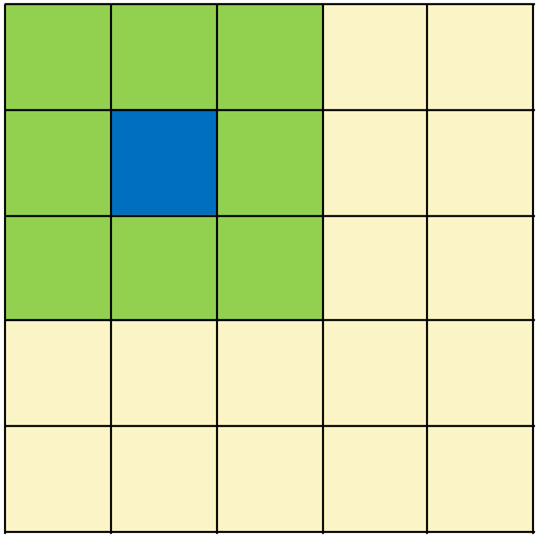
Announcements

- Mini Quiz – 2 today (hopefully !)

Mean/Average Filter

Note: Coefficients sum to 1 N

I



H → 3 x 3

Weight Mask / Kernel / Filter

1/9 1/N	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

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

3 x 3

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

Effect of Mask Size

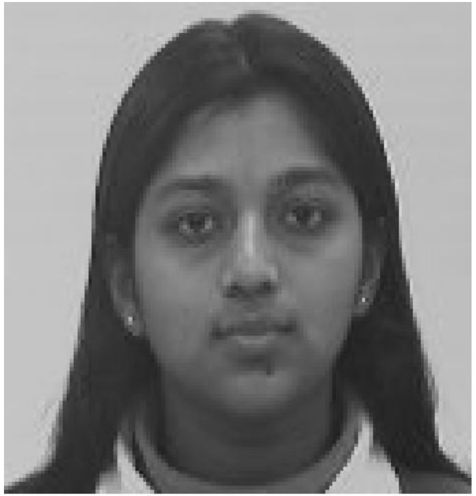
Original Image

[3x3]

[5x5]

[7x7]

Repeated Averaging Using Same Filter



Before

After

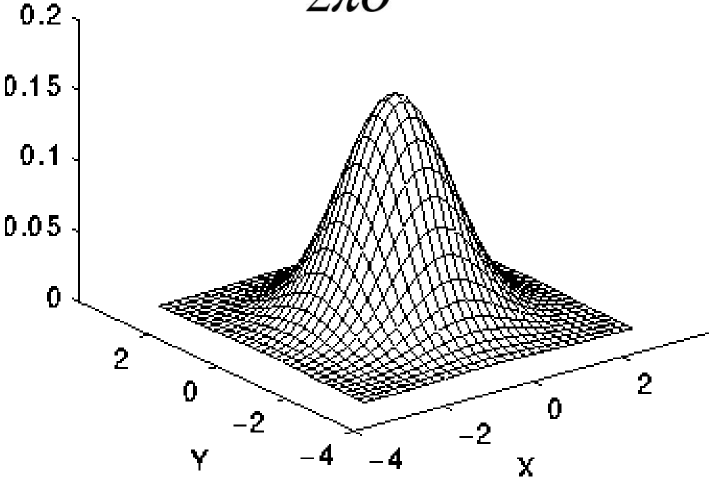
After repeated averaging

NOTE: Can get the <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

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

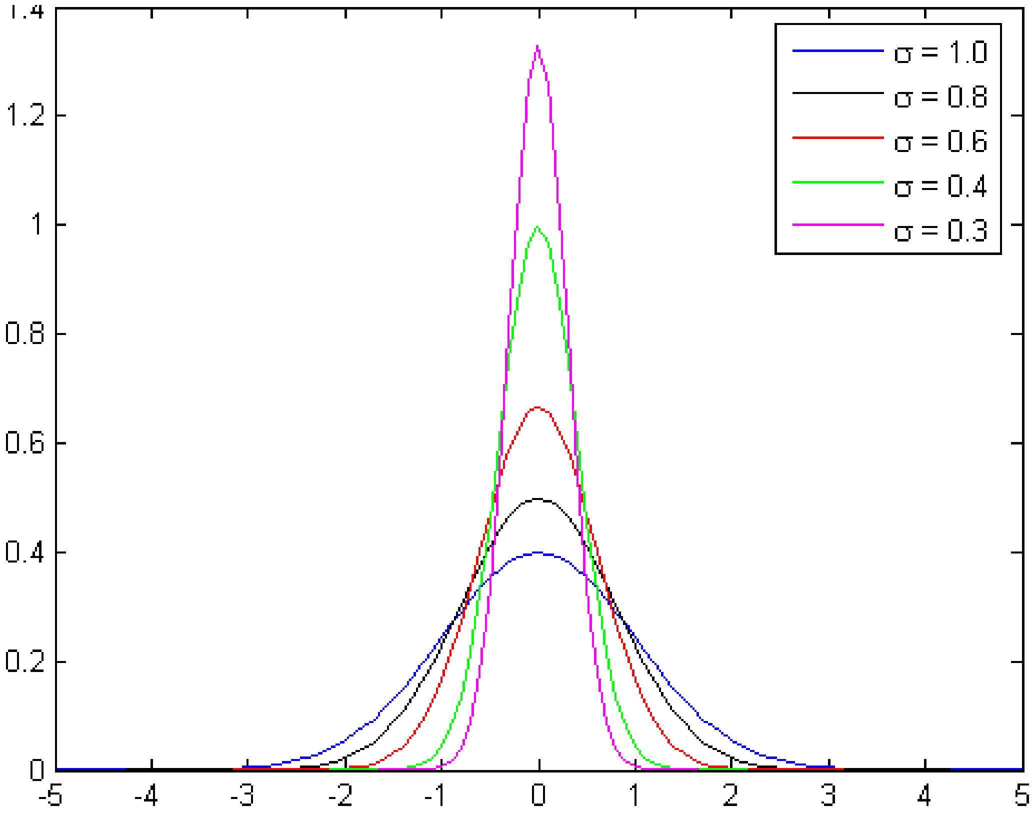


$\frac{1}{\cancel{265}}$ 256

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, σ=1

How are Gaussian filter coefficients obtained ?

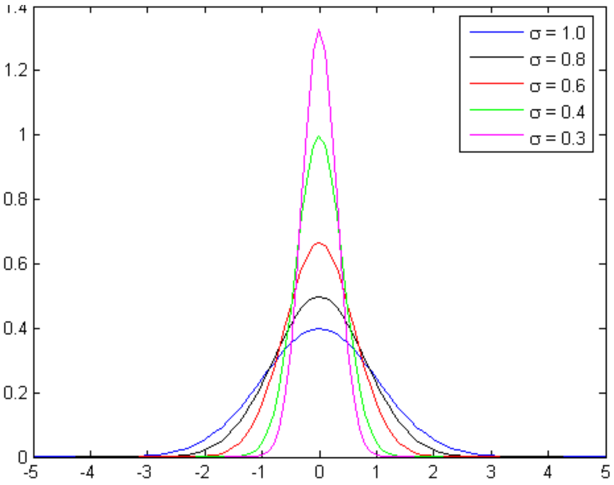


pdf

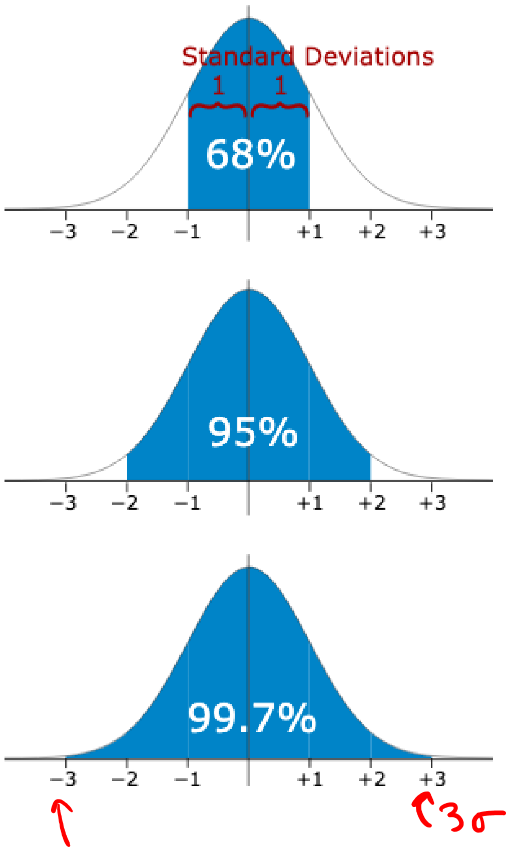
$$g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{x^2}{2\sigma^2}\right)$$

$$\int_{-\infty}^{\infty} \underbrace{g_\sigma(x)\,dx} = 1$$

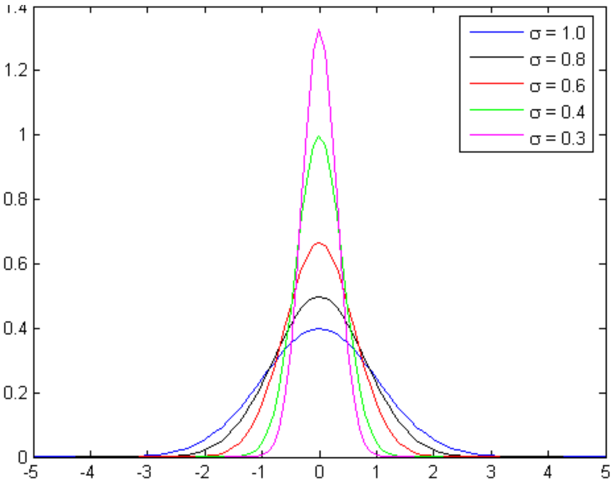
How are Gaussian filter coefficients obtained ?



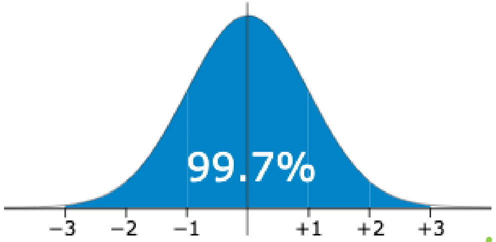
$$g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$



How are Gaussian filter coefficients obtained ?

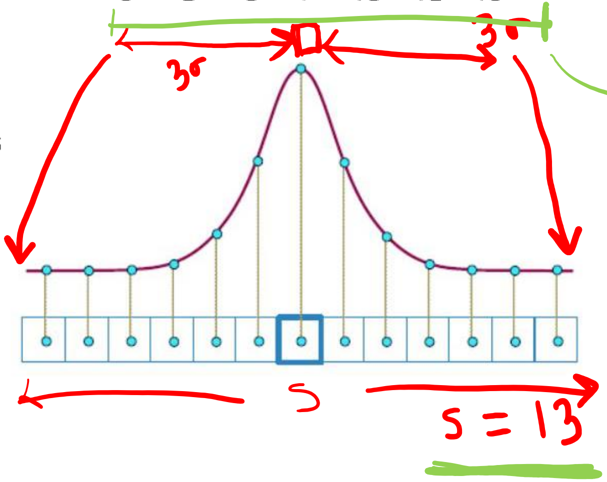


$$g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{x^2}{2\sigma^2}\right)$$

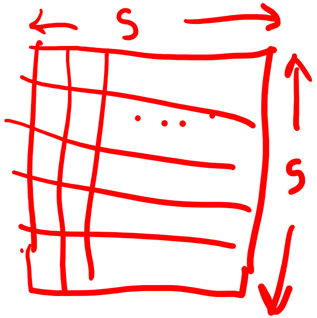


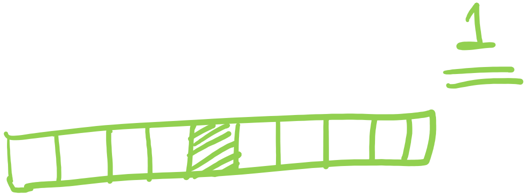
$s = \text{round}(7\sigma)$

$s = \text{round}(6\sigma)$



s





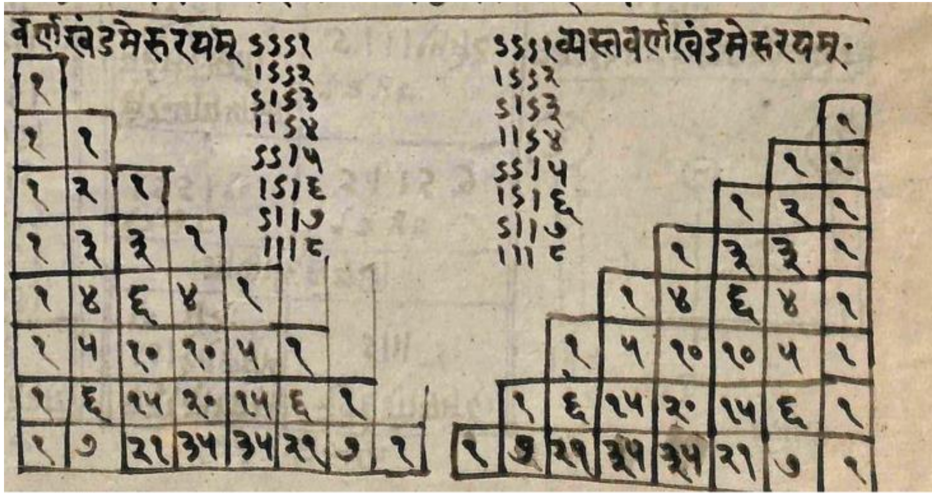
How are Gaussian filter coefficients obtained ?

$$\rightarrow \sum_{k=0}^{N} {}^{N}C_{k} = 2^{N}$$

$$\Rightarrow \sum_{k} \frac{{}^{N}C_{k}}{2^{N}} = \underline{1}$$

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

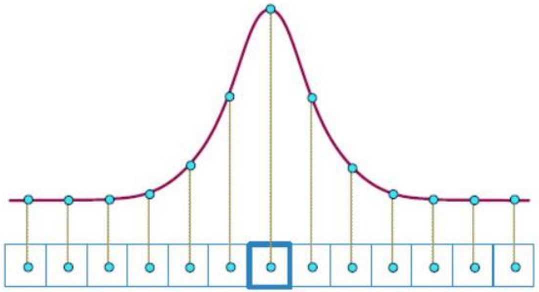


How are Gaussian filter coefficients obtained ?

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

E.g. s = 7 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



How are Gaussian filter coefficients obtained ?

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

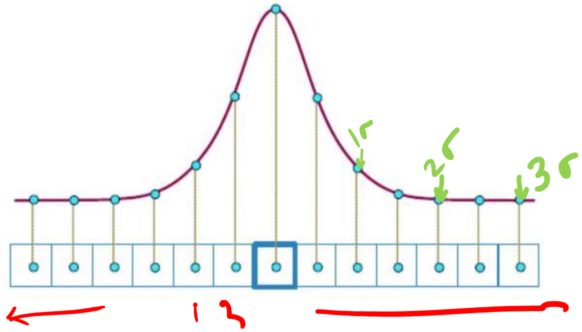
$s = 13 \times 13$

s = 7 x 7

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

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

7



How are Gaussian filter coefficients obtained ?

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

σ

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

s = 7 x 7

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

7x7

$[\ v\ ]$

$\frac{1}{64} \begin{bmatrix} v \end{bmatrix} \frac{1}{64}$

$\frac{1}{64} v v^T \frac{1}{64}$

7x1 1x7

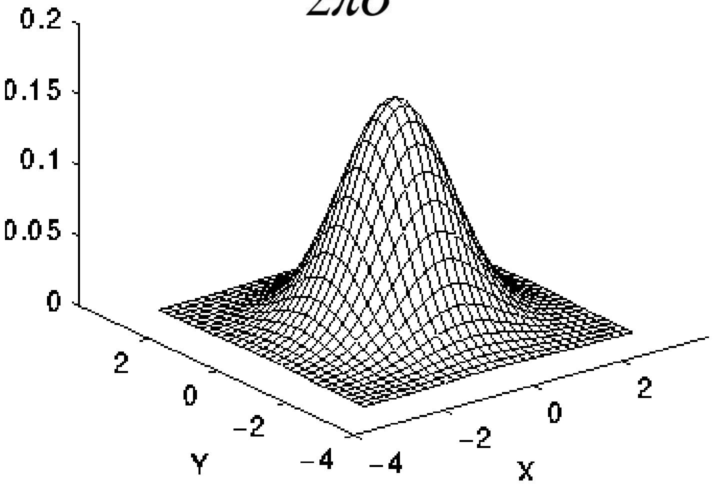
$N = s - 1$

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

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, σ=1

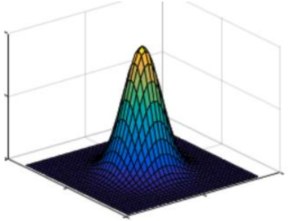
Gaussian Smoothing – Effect of sigma

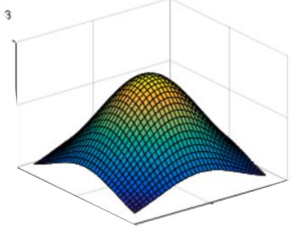
$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\{-(x^2 + y^2) / 2\sigma^2\}$$

Original Image
(Sigma 0)

Gaussian Blur
(Sigma 0.7)

Gaussian Blur
(Sigma 2.8)





Edge detection

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)

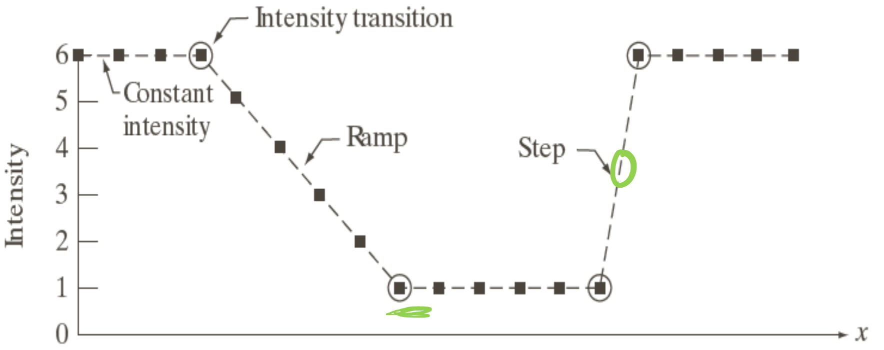
$$\frac{\partial f(x,y)}{\partial x} \sim f[x+1,y] - f[x,y]$$

-1	1

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



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	5	0	0	0	0	
2nd derivative		0	0	−1	0	0	0	0	1	0	0	0	0	5	−5	0	0	0	

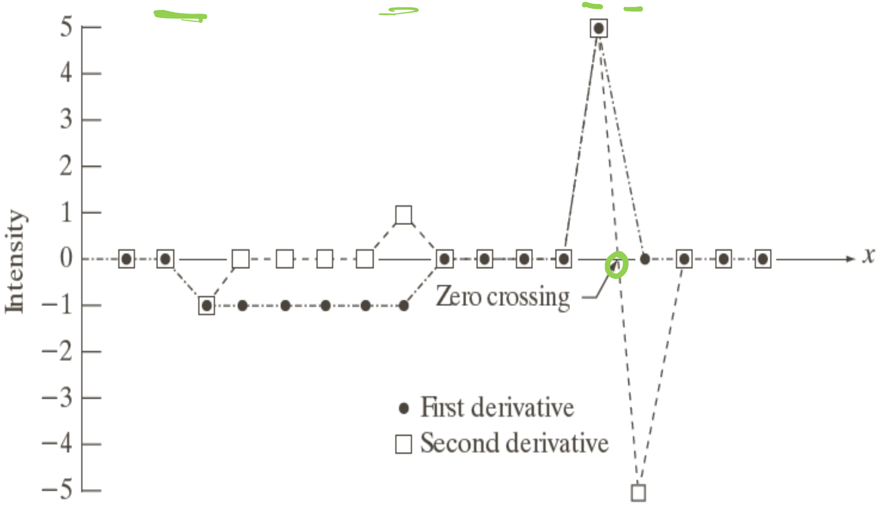


Image Gradient and Edges

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

x-derivative

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

y-derivative

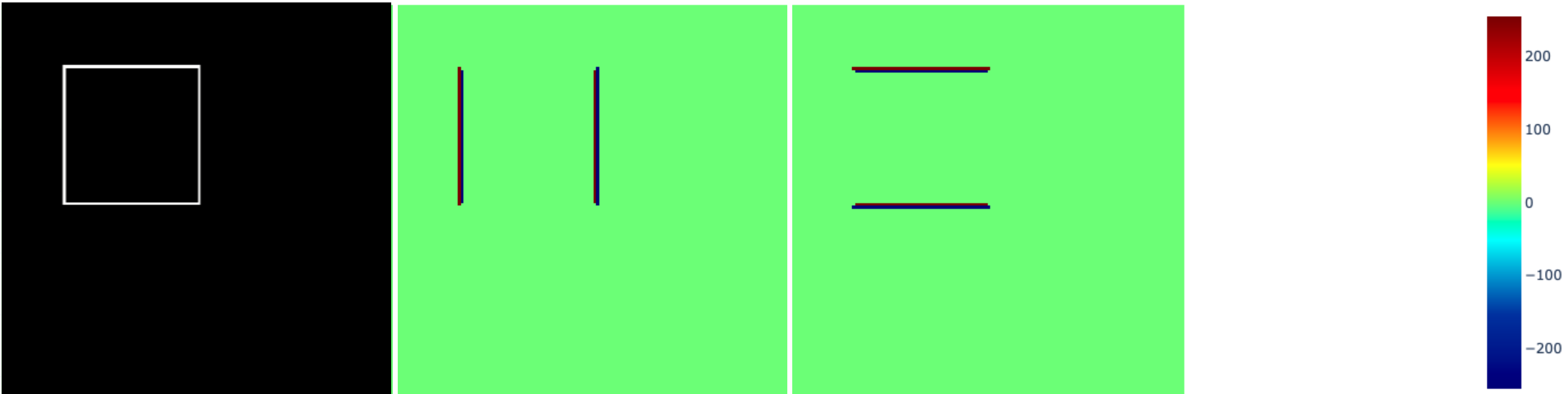


Image | x-derivative | y-derivative

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

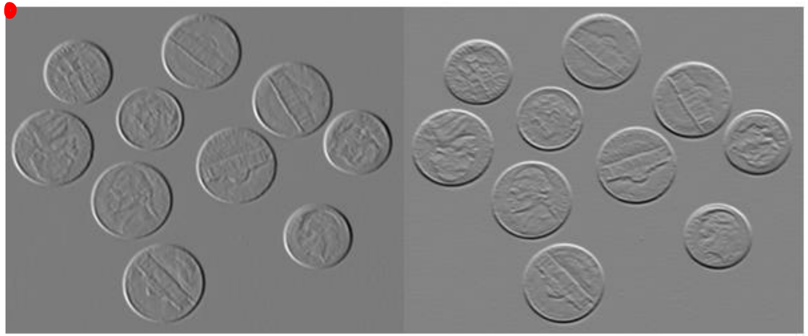
G_x

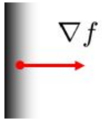
+1	+1	+1
0	0	0
-1	-1	-1

G_y

-1	0	1
-1	0	1
-1	0	1

Edge is perpendicular to gradient





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

$[x, 0]$

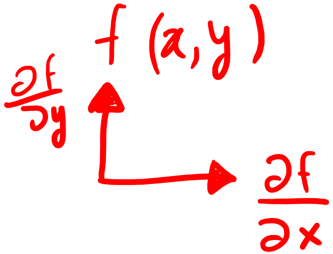
$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$

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

$\mathbf{G_x}$

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

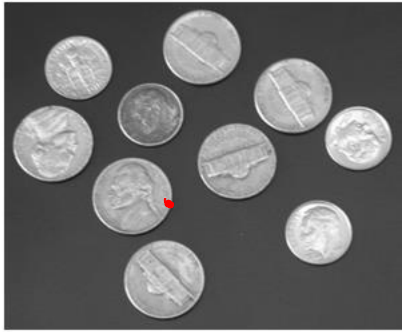
$\mathbf{G_y}$



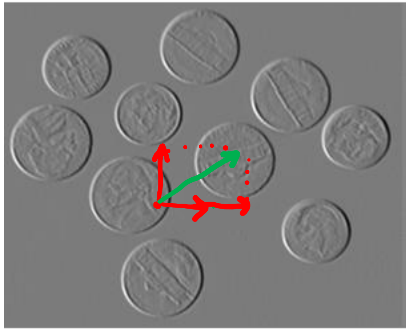
$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x-h)}{2h}$$

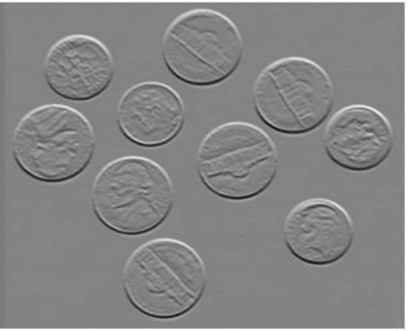
$$= \frac{\boxed{1} f(x+h) + \boxed{0} f(x) \boxed{-1} f(x-h)}{2h}$$

Gradient Magnitude and Orientation

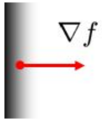


I





$\frac{\partial f}{\partial x}$



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

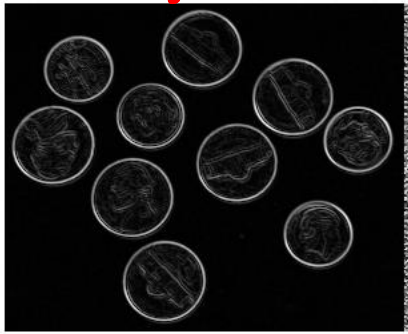
$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$

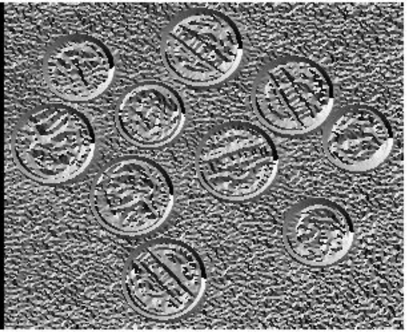
$\frac{\partial f}{\partial y}$

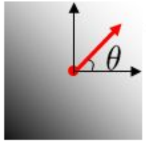
magnitude

orientation

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







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

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

2-D Laplacian Filter

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

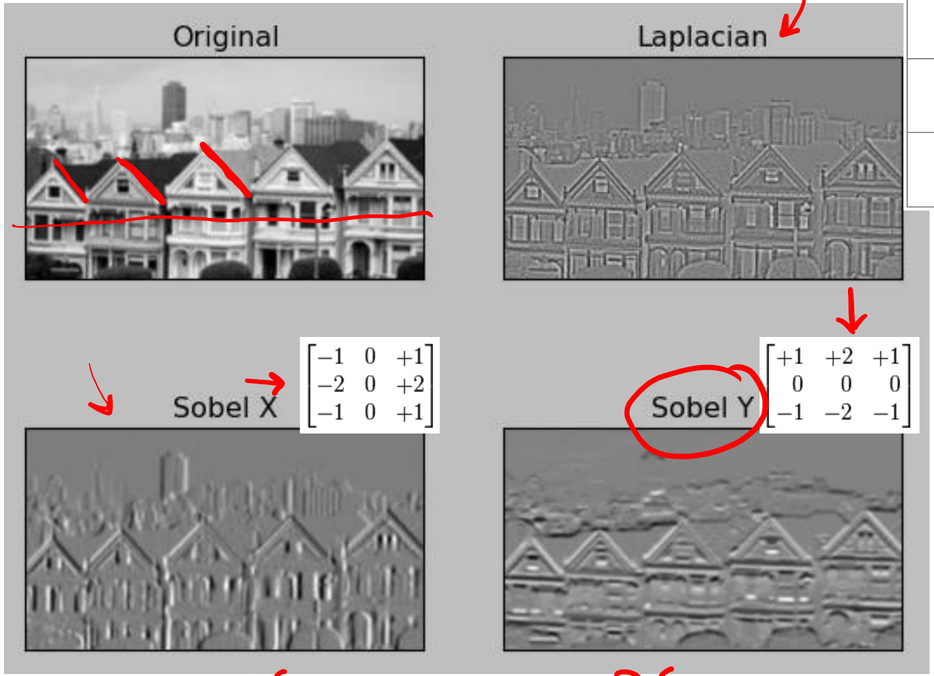
$$\nabla^2 f = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

Laplacian filter

0	1	0
1	-4	1
0	1	0

0

Edge Masks – Sobel , Laplacian



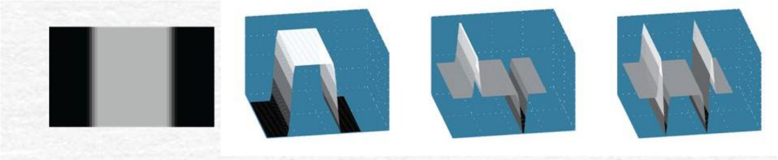
Sobel X: $\begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$

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

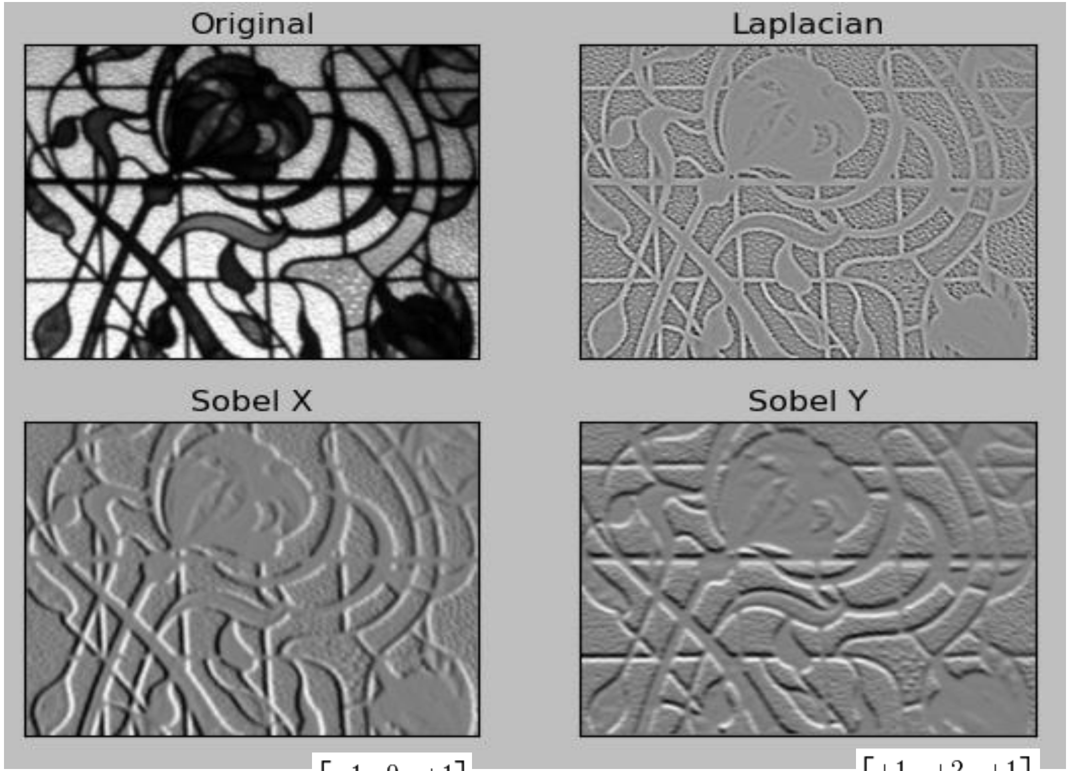
$\frac{\partial f}{\partial x}$ $\frac{\partial f}{\partial y}$

0	−1	0
−1	4	−1
0	−1	0

Note: Coefficients sum to 0



Edge Masks – Sobel , Laplacian



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

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

0	−1	0
−1	4	−1
0	−1	0

$-1 \quad 0 \quad 1$

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

Image Sharpening

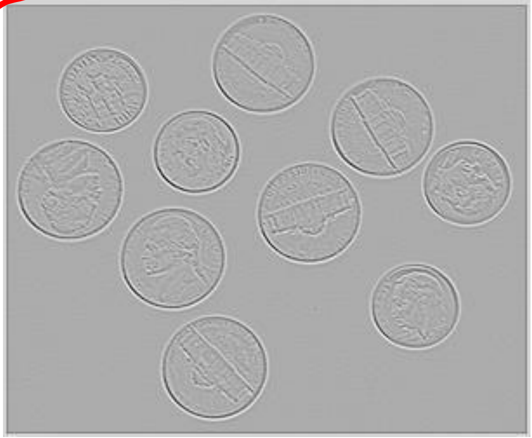
$I(u,v)$

+

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

=

$\boxed{I'(u,v)}$



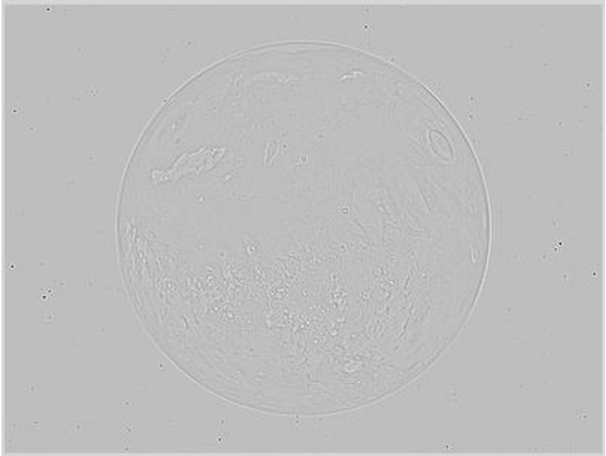
$\nabla^2 I(u,v) + 128$

(For visualization)

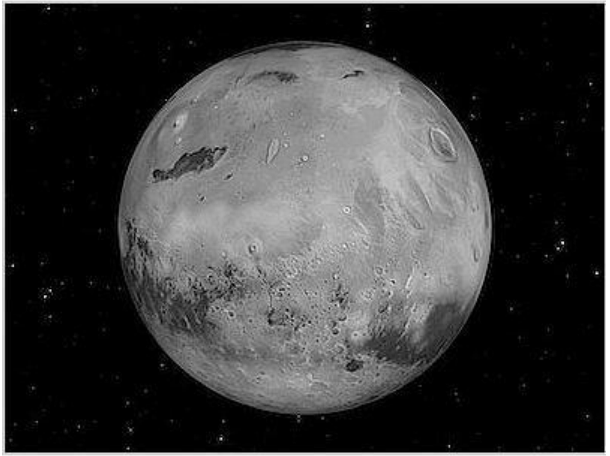
Sharpening (Unsharp Masking)

$I(u,v)$

$\nabla \ I(u,v)$



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



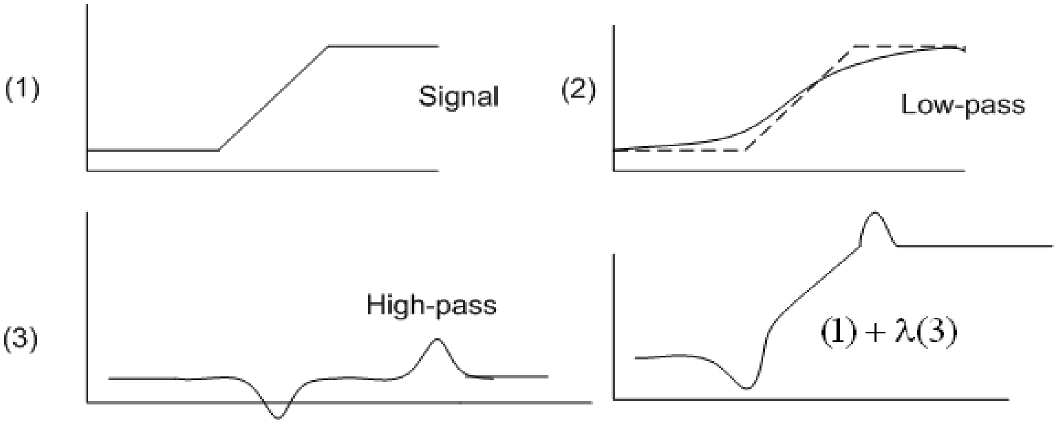


Image Courtesy: NASA

Highboost Filtering

- What does blurring take away?

–

=

- Let’s add it back:

+ a

=

Unsharp Masking vs Highboost Filtering

∇^2

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)
```

valid

0 0

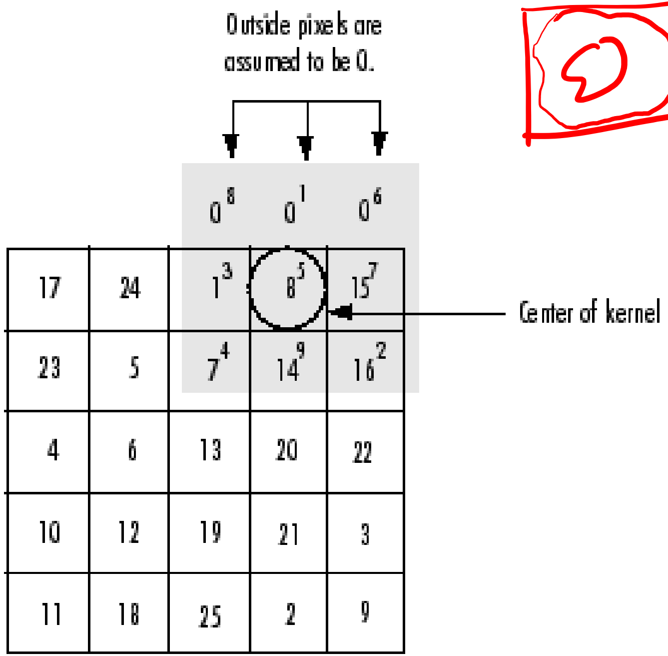
$$\begin{bmatrix}120&190&140&150&200\\17&21&30&8&27\\89&123&150&73&56\\10&178&140&150&18\\190&14&76&69&87\end{bmatrix} \times \begin{bmatrix}1/9&1/9&1/9\\1/9&1/9&1/9\\1/9&1/9&1/9\end{bmatrix} = \begin{bmatrix} & & & & \\ &98& & & \\ & & & & \\ & & & & \\ & & & & \end{bmatrix}$$

3x3

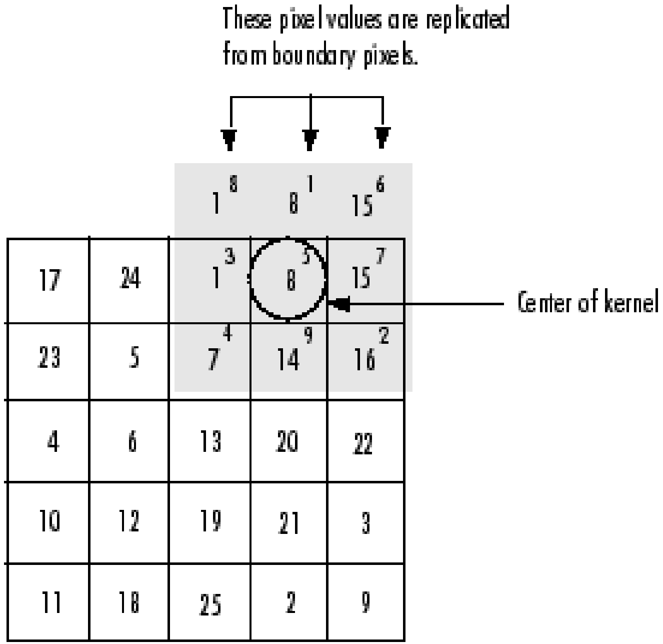
5x5

5x5

Image Padding



zero



replicate

References

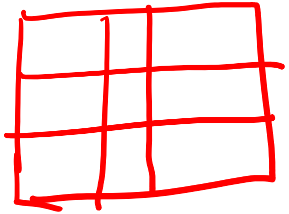
- GW Chapter – 3.4.1, 3.5.1,3.6

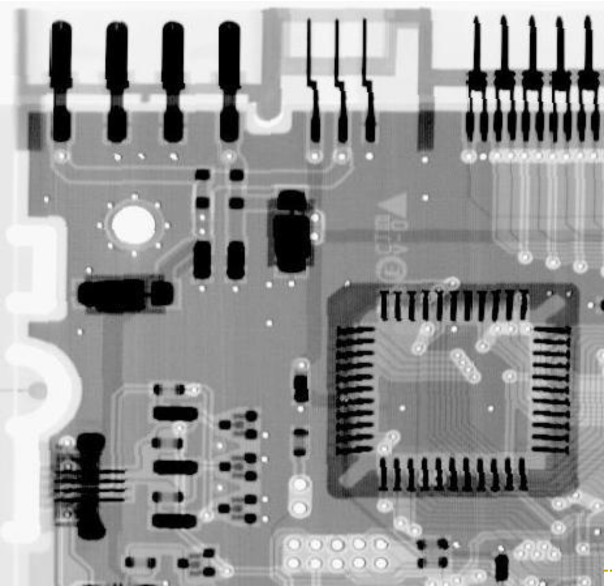
Spatial Domain Filtering - Approaches

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

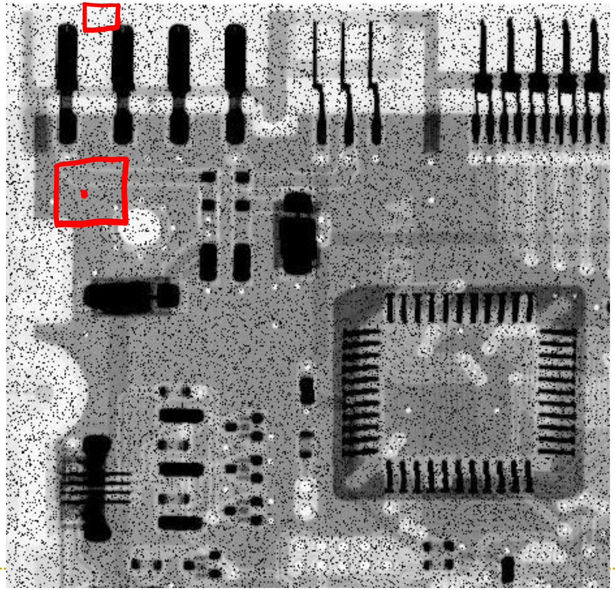
Non-linear Spatial Filters (max)

3×3

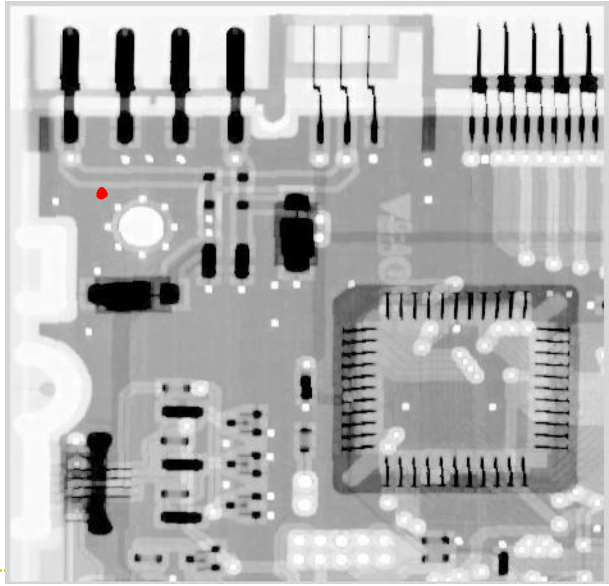




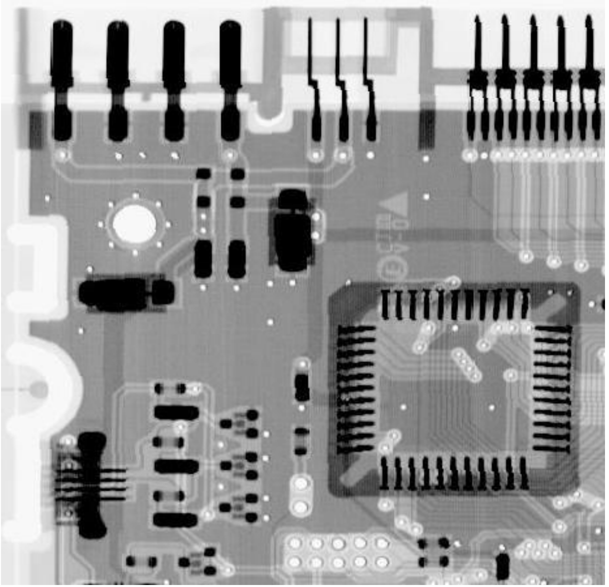
pepper noise



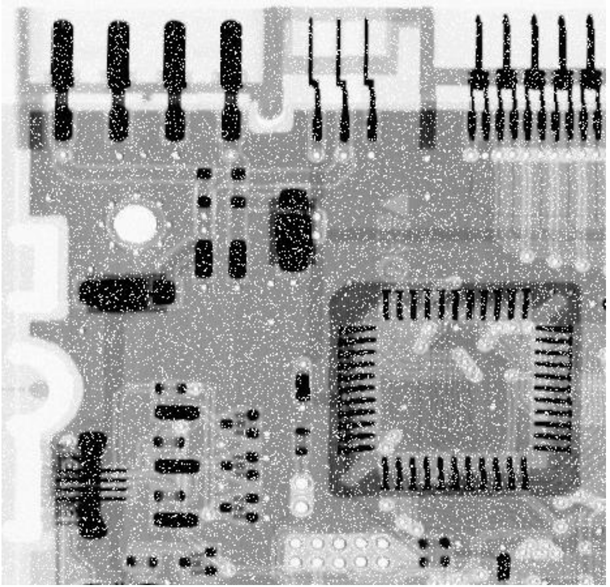
After applying max filter



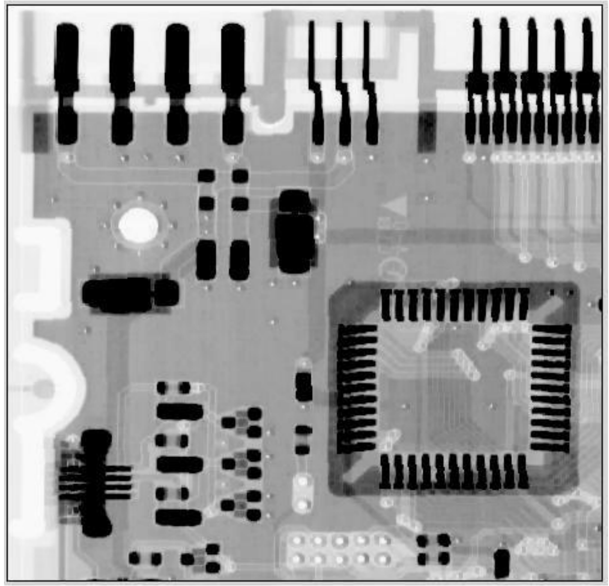
Non-linear Spatial Filters (min)



salt noise



After applying min filter

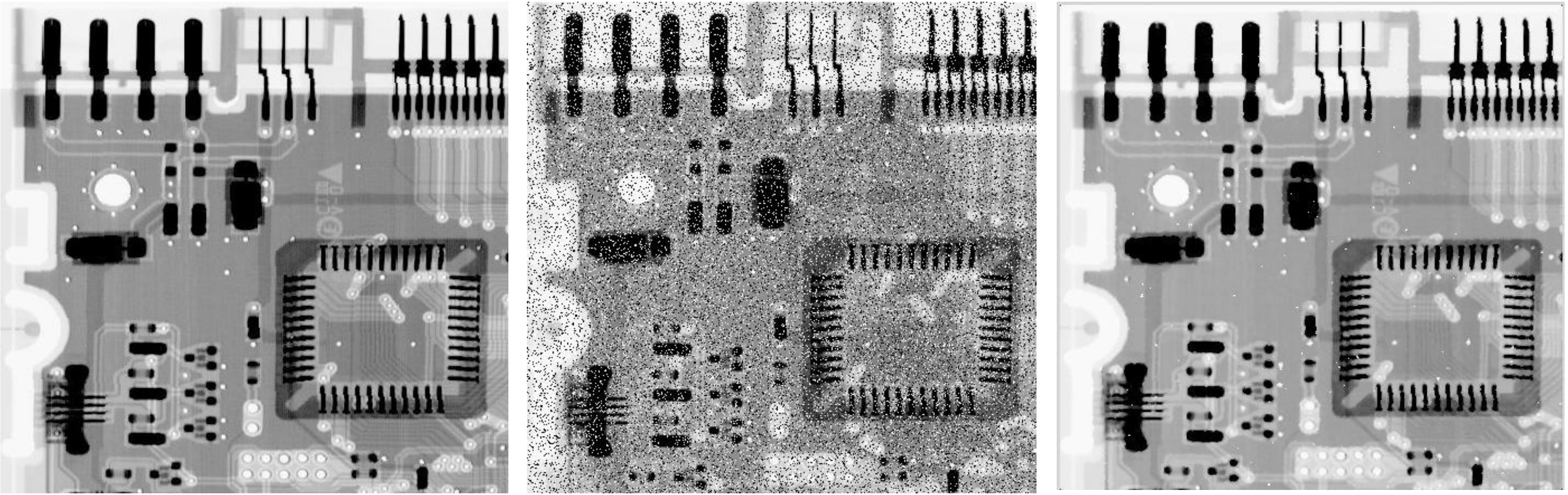


Non-linear Spatial Filters (median)

$0\ 0\ 0\ 0\ \boxed{0}\ 0 \cdots 0\ 2\ 1\ 0$

salt & pepper noise

After applying median filter



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

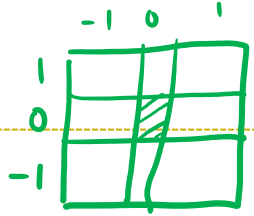
Other Spatial Filters

- Geometric mean
- Harmonic mean
- Contra harmonic mean
- Mid Point filter 25 50 75
- Alpha trimmed mean filter
- …..

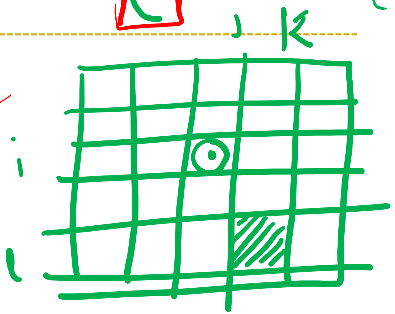
Bilateral Filtering (Edge preserving smoothing)

Original image taken from cs.cityu.edu.hk

Linear Spatial Filter

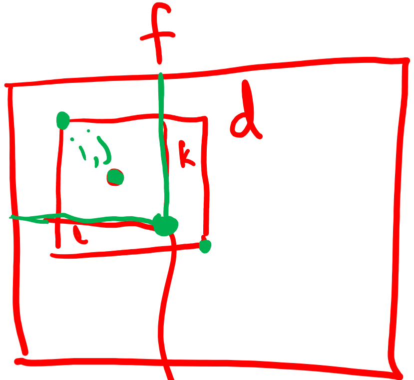


$d(i,j,k,l) = e^{-\left(\frac{(i-k)^2+(j-l)^2}{2\sigma_d^2}\right)}$

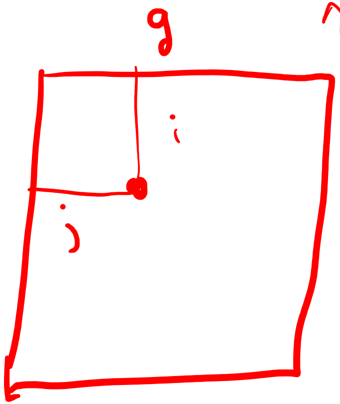


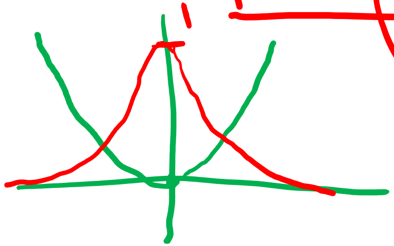
$d(i,j,k,l)$

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



$f(k,l)$





$$g(i,j) = \frac{\sum_{k}\sum_{l} f(k,l)\, d(i,j,k,l)}{\sum_{k,l} d(i,j,k,l)}$$

References

- GW Chapter – 3.4