Lecture 4
Image Enhancement II

Instructor: Yiming Xiao

Email: yiming.xiao@concordia.ca

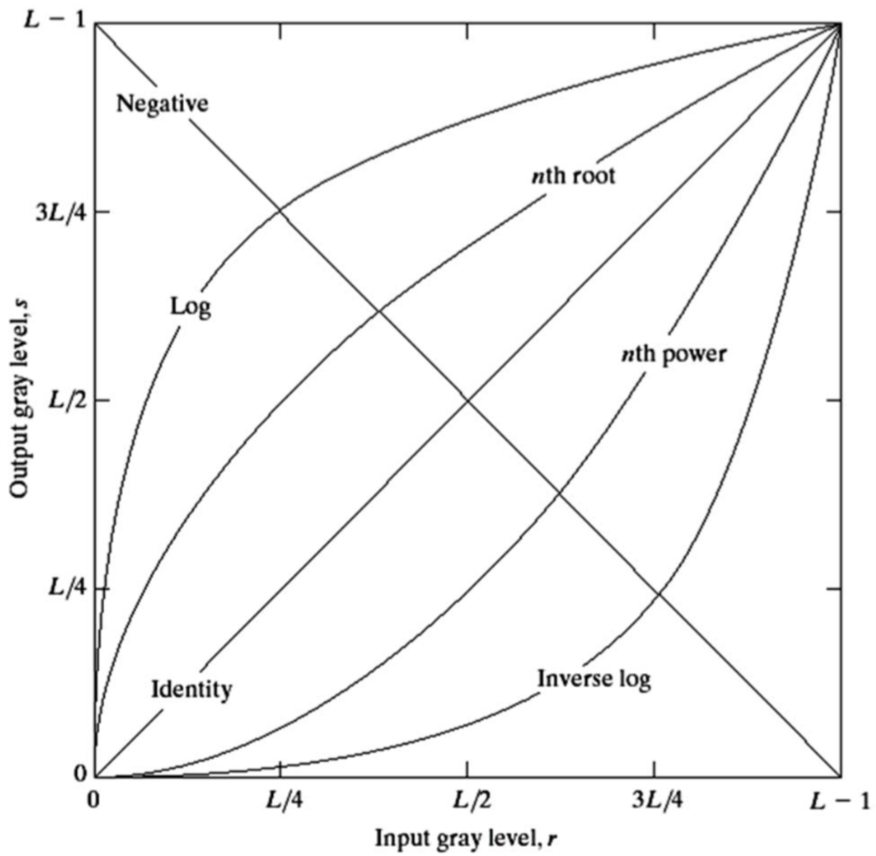
Department of Computer Science and Software Engineering
Concordia University

Slides modified from materials provided by Dr. Tien D Bui

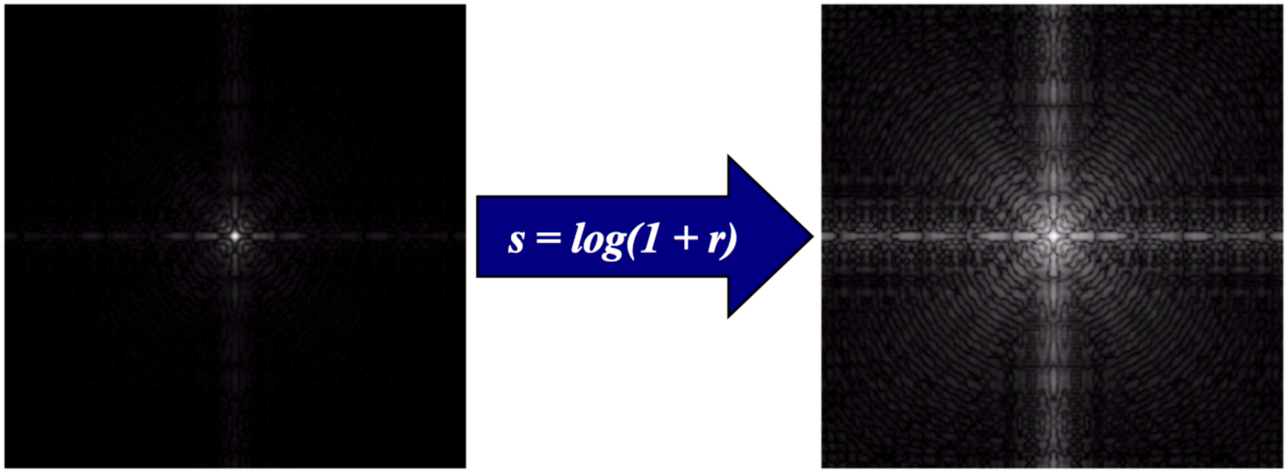
Point-wise operation - something left from Lec 3

- Linear
  - Negative/Identity
- Logarithmic
  - Log/Inverse log
- Power law
  - n^{th} power/n^{th} root

Gamma correction



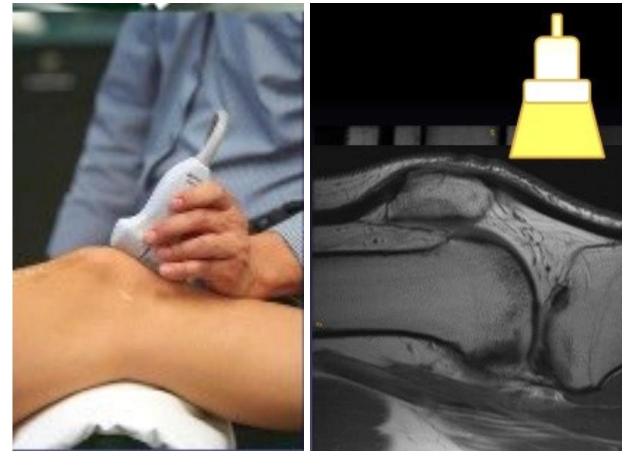
Point-wise operation — Log transforms



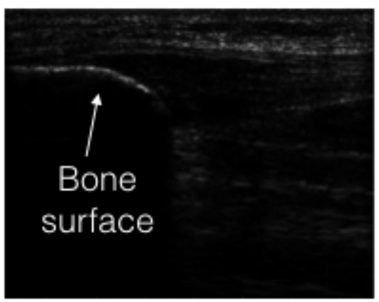
New pixel value $\longrightarrow s = c * log(1 + r) \longleftarrow$ Old pixel value

- Maps wide range of input levels to a narrower output range
- Inverse log, or exponential transform does the opposite
- Log transform of a Fourier transform result of an image shown above

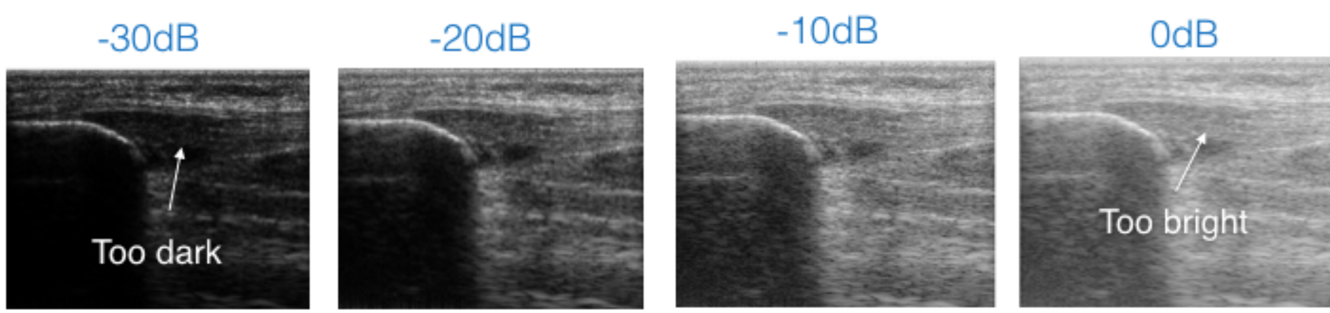
- Original signal from ultrasound imaging has a huge dynamic range
- The digital display fails to display the full details but only the bright regions
- Log transform is used to compress the intensity range to fit the display
- Problem of adjusting the brightness level

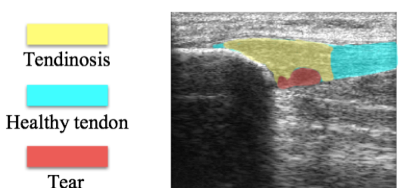


Imaging the patellar tendon injury



Raw US signal





Log-compression US at different gains

Xiao et al., Ultrasound in Medicine and Biology, 2018

Spatial Filtering

- Filter
  - In general, a tool to suppress or emphasize some aspect of signal
  - Accepting or rejecting certain frequency components
  - E.g., lowpass filter, highpass filter, bandpass filter
  - Lowpass filter: smoothing or blurring image
  - Highpass filter: emphasize edges or details (texture and noise)
- Spatial Filter
  - One of principal tools for a broad spectrum of applications
  - A spatial filter has a window (square,rectangle or a diamond shape) and predefined operations (add, multiply etc.).
  - New pixel value is created at the center pixel of the window.
  - If operations are linear, we have a linear spatial filter.

$$f(x) \longrightarrow \boxed{\begin{array}{c}\text{System}\\ H\end{array}} \longrightarrow g(x)$$

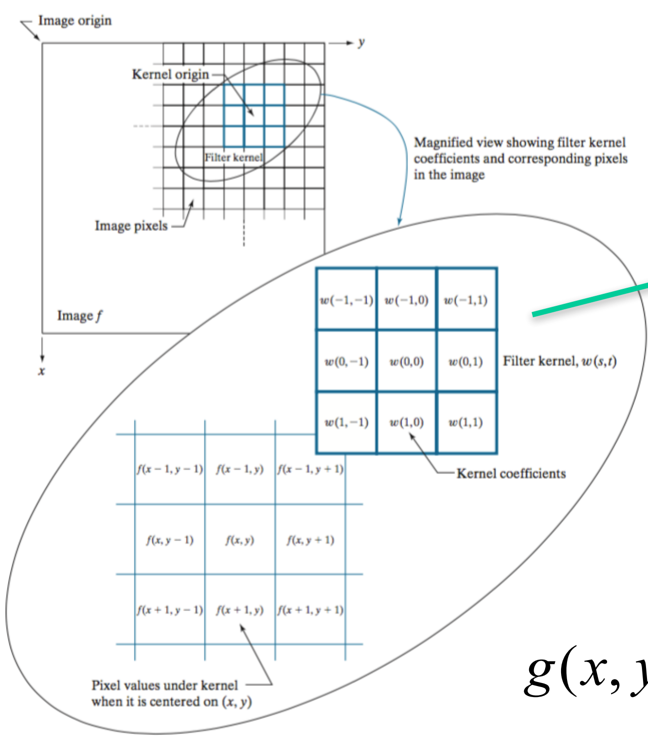
Filter function

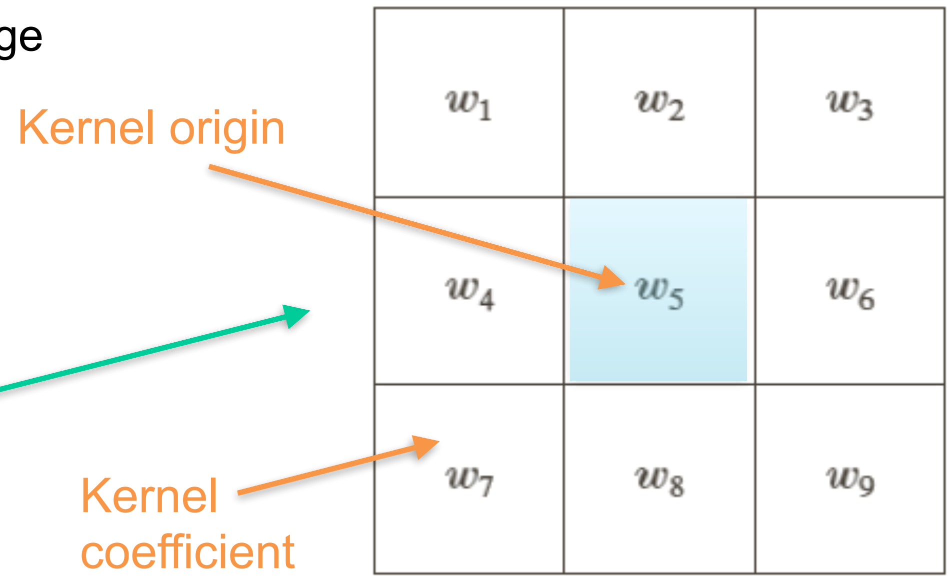
https://medium.com/udacity-pytorch-challengers/style-transfer-using-deep-nural-network-and-pytorch-3fae1c2dd73e

Spatial Filtering

Use a window or mask to alter pixel values according to local operation

Local linear operations on an image

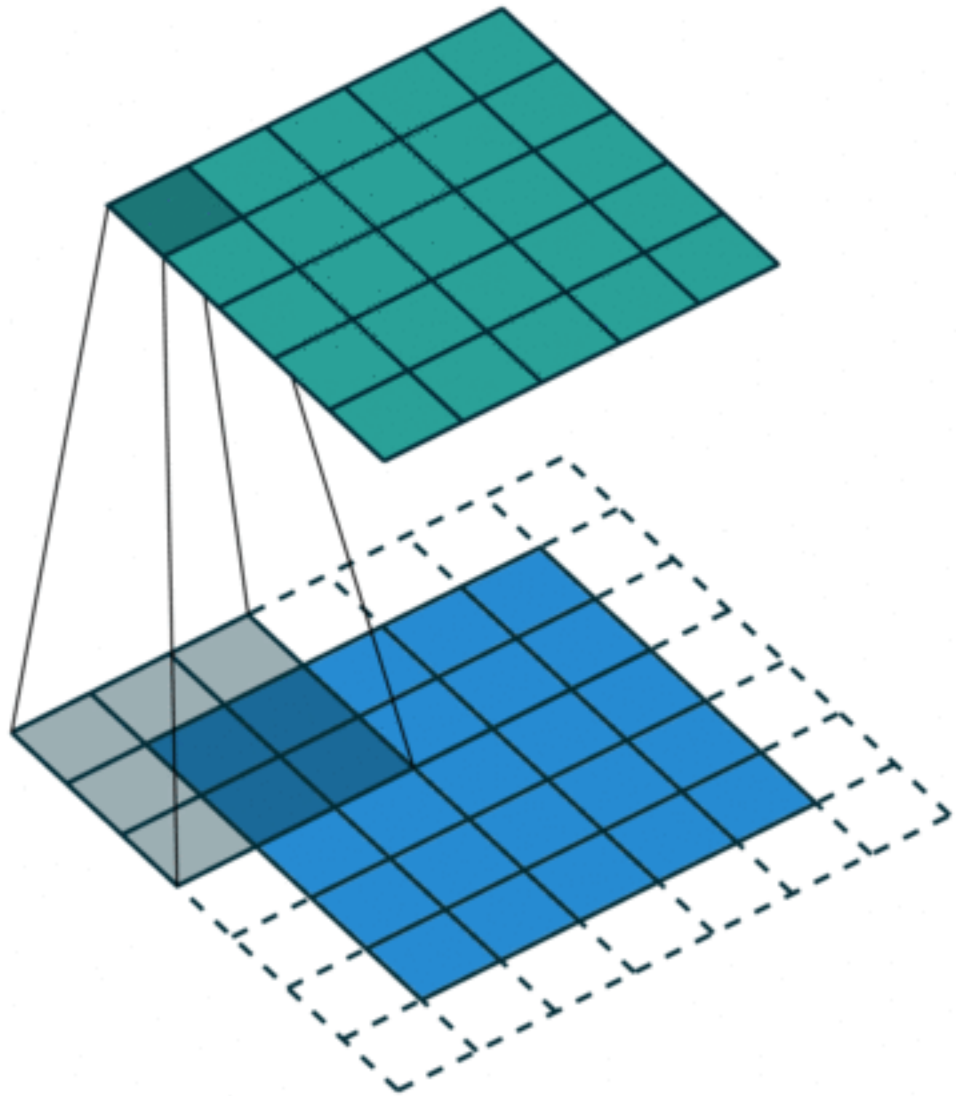




Input: f(x,y), Output: g(x,y)

$$
\begin{aligned}
g(x,y) = {} & w_1 f(x-1, y-1) + w_2 f(x-1, y) + \cdots \\
& w_8 f(x+1, y-1) + w_9 f(x+1, y+1)
\end{aligned}
$$

FIGURE 3.28 The mechanics of linear spatial filtering using a 3 × 3 filter mask. The form chosen to denote the coordinates of the filter mask coefficients simplifies writing expressions for linear filtering.



Spatial Filtering

- Window size
  - *m* x n
  - *m* and *n* are odd. *m* = 2*a* +1. *n* = 2*b* +1.
  - Smallest is 3x3. Then 5x5, 7x7, 9x9,

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

- Vector representation
  - $\mathbf{w} = (w_1, w_2, \cdots, w_{m\times n})^T$ Kernel coefficients
  - $\mathbf{z} = (z_1, z_2, \cdots, z_{m\times n})^T$ Pixel values
  - $\mathrm{g}(\mathrm{x},\mathrm{y}) = \mathbf{w}^T\mathbf{z}$

Spatial Filtering: Smoothing

Example: averaging mask

$$\frac{1}{9}\times\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array}$$

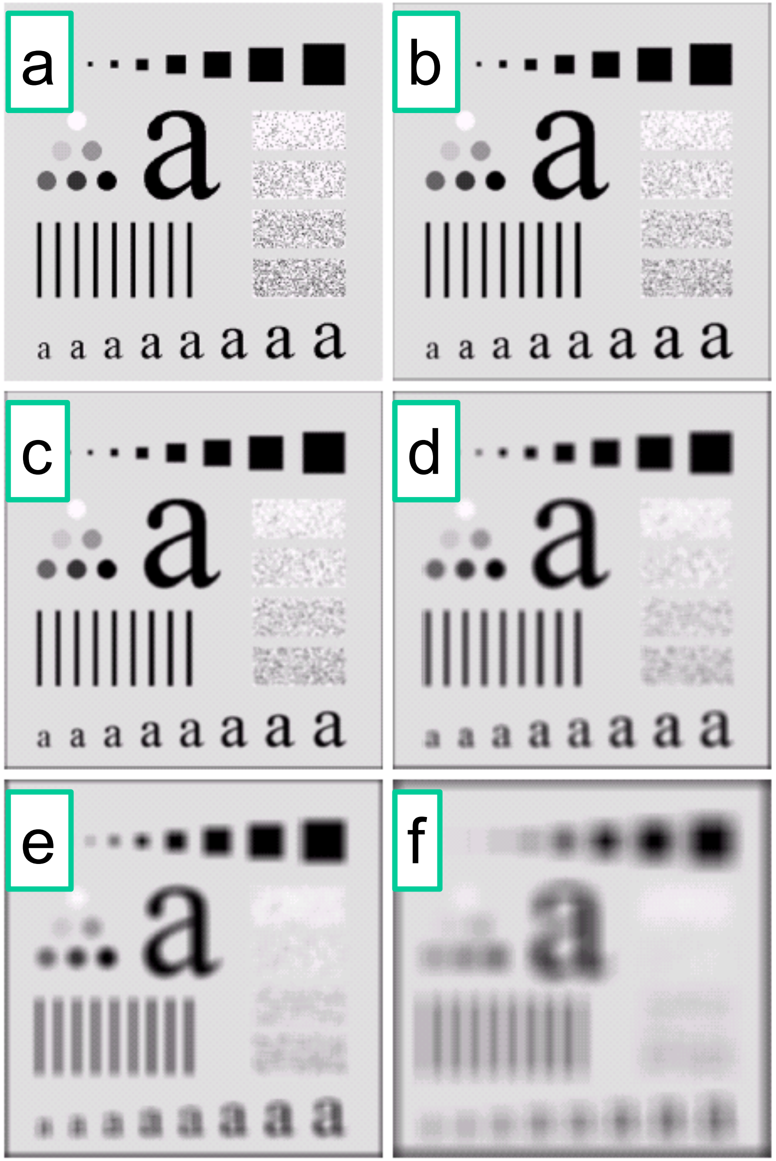
box filter

$$\frac{1}{16}\times\begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline\end{array}$$

Weighted filter

$$g(x,y)=\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)\,f(x+s,y+t)$$

Spatial Filtering: Smoothing

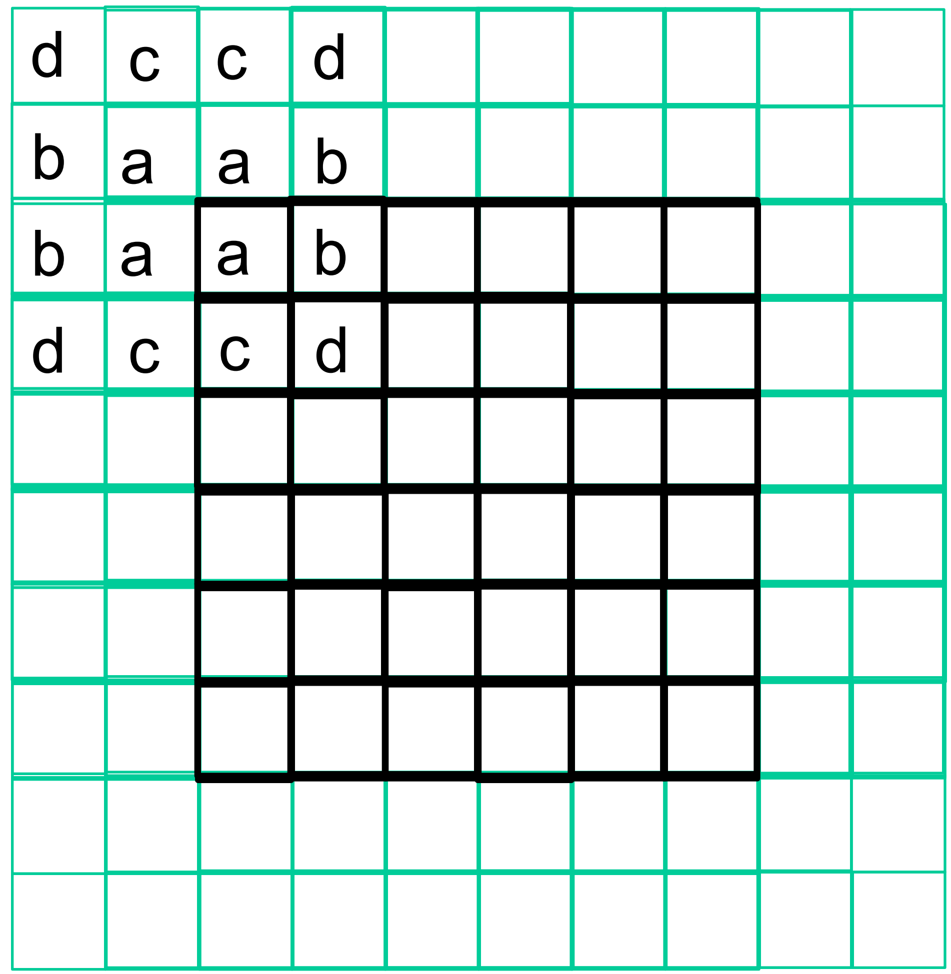


- a). original image 500x500 pixel
- b).~ f). results of smoothing with square averaging filter masks of size n = 3, 5, 9, 15 and 35, respectively.
- Note:
  - ✓ Big mask is used to eliminate small objects from an image.
  - ✓ The size of the mask establishes the relative size of the object to be blended with the background

How to Deal with Boundaries?

Example: 5x5 Mask – How to fill in “border”?

- Zeros (“OK”)
- Replication (Better)
- Reflection (“Best”)



Procedure:

– Replicate row-wise

– Replicate column-wise

– Apply filtering

– Remove borders

Linear Filtering (example)

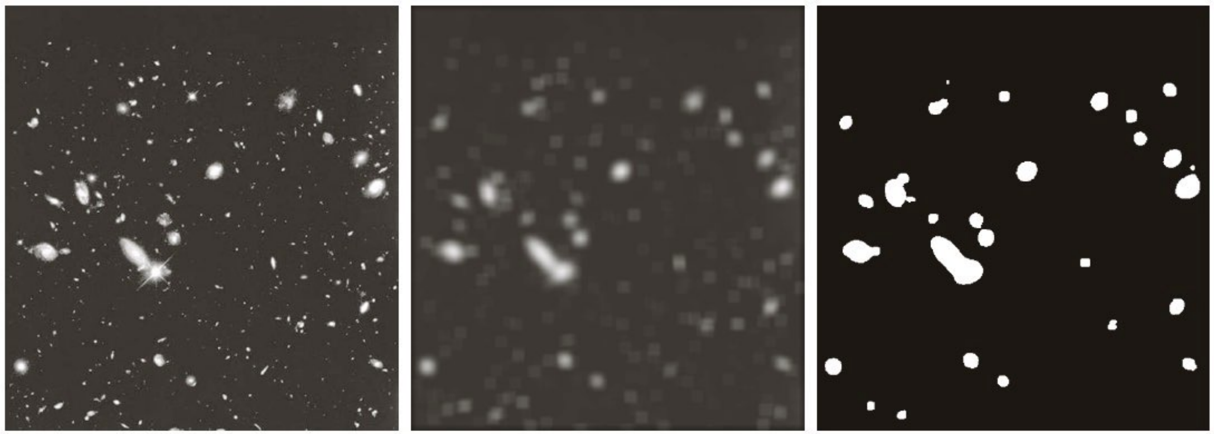


FIGURE 3.34 (a) Image of size 528 × 485 pixels from the Hubble Space Telescope. (b) Image filtered with a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

Image from Hubble telescope: (a) contains small objects that could be noise, we want to extract larger and brighter objects; (b) blurred with 15 by 15 averaging mask, we see some small objects are blended to background; (c) thresholding of (b) to highlight larger and brighter objects.

Order-statistics Filters

- Linear filters of the type we have seen will smooth the image and reduce certain kinds of noise.
- Nonlinear smoothing filters can also be considered
  - Instead of computing a weighted average over the masked area, perform an operation on the sorted list of pixels in the area.
- Order statistic filters are useful for removing certain "impulsive" types of noise.

a*F(A)+b*F(B) != F(a*A+b*B)

Order-statistics Filters

Linear Filter

$$\underbrace{\begin{bmatrix} f_1 & f_2 & \cdots & f_8 & f_9 \end{bmatrix} \quad \begin{bmatrix} w_1 & w_2 & \cdots & w_8 & w_9 \end{bmatrix}}_{\text{Inner product}}$$

$$g = \sum_i w_i f_i = \underline{w}^T \underline{f}$$

Nonlinear (OS) Filter

$$\underbrace{\begin{bmatrix} f_1 & f_2 & \cdots & f_8 & f_9 \end{bmatrix}}_{\text{sort}}$$

$$\begin{bmatrix} z_1 & z_2 & \cdots & z_8 & z_9 \end{bmatrix}$$

re-order from low to high values

$$\underbrace{\begin{bmatrix} w_1 & w_2 & \cdots & w_8 & w_9 \end{bmatrix}}_{\text{Inner product}}$$

$$g = \sum_i w_i z_i = \underline{w}^T \underline{z}$$

Nonlinear operator

Examples of Order-statistics Filters

Nonlinear (OS) Filter

$$\underbrace{\begin{bmatrix} f_1 & f_2 & \cdots & f_8 & f_9 \end{bmatrix}}_{\text{sort}}$$

$$\begin{bmatrix} z_1 & z_2 & \cdots & z_8 & z_9 \end{bmatrix}$$

re-order from low to high values

$$\underbrace{\begin{bmatrix} w_1 & w_2 & \cdots & w_8 & w_9 \end{bmatrix}}_{\text{Inner product}}$$

$$g = \sum_i w_i z_i = \underline{w}^T \underline{z}$$

Median Filter

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

Min Filter

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

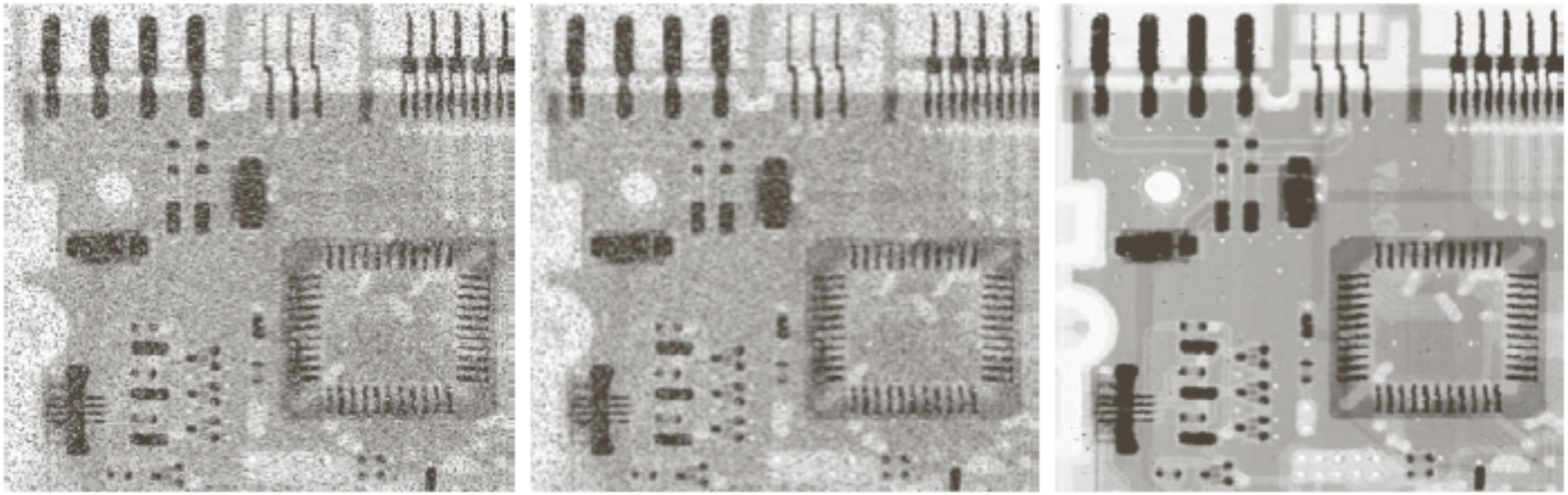
Max Filter

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

Trimmed Mean Filter

$$\begin{bmatrix} 0 & 0 & 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & 0 & 0 & 0 \end{bmatrix}$$

Examples of Order-statistics Filters



a b c

FIGURE 3.35 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3 × 3 averaging mask. (c) Noise reduction with a 3 × 3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

(a) X-ray image with salt-and-pepper noise; (b) noise reduction using a 3 by 3 averaging mask; noise reduction using a 3 by 3 median filter

Remember to check out what "salt and pepper " noise is

More Linear Operations: Sharpening Filters

- Sharpening filters use masks that typically have + and – numbers in them.

- They are useful for highlighting or enhancing details and high-frequency information (e.g. edges)

- They can (and often are) based on derivative-type operations in the image (whereas smoothing operations were based on “integral” type operations)
- Recall:

$$\frac{\partial f}{\partial \mathrm{x}} \approx \frac{f(\mathrm{x}+h)-f(\mathrm{x})}{h} \approx f(x+1)-f(x)$$

Derivative-type Filters

$$\frac{\partial f}{\partial \mathrm{x}} \approx f(x+1, y) - f(x, y)$$

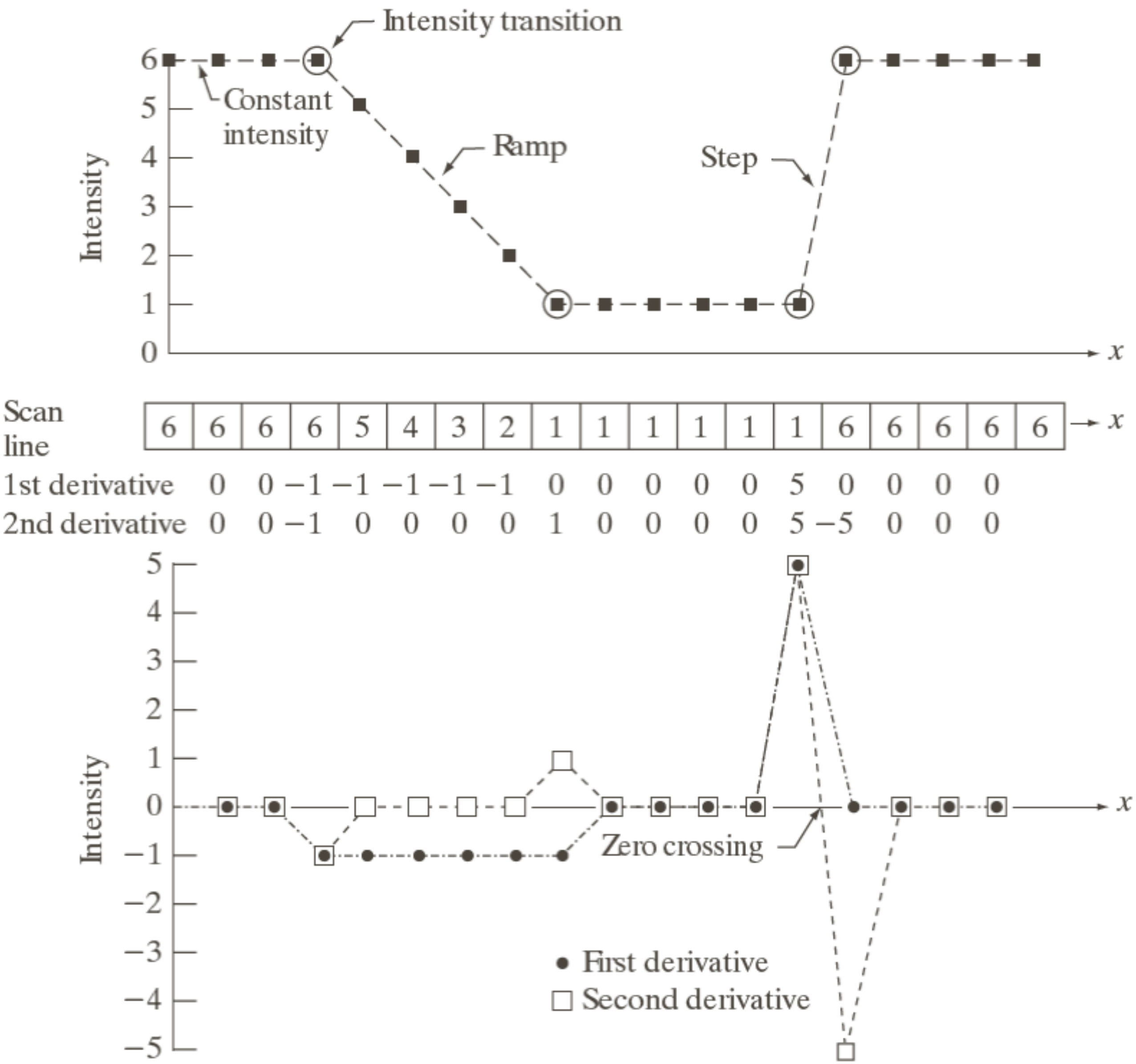
$$\frac{\partial f}{\partial \mathrm{y}} \approx f(x, y+1) - f(x, y)$$

$$\frac{\partial^2 f}{\partial \mathrm{x}^2} \approx f(x+1, y) - 2f(x, y) + f(x-1, y)$$

$$\frac{\partial^2 f}{\partial \mathrm{y}^2} \approx f(x, y+1) - 2f(x, y) + f(x, y-1)$$

Laplacian: $\nabla^2 f = \dfrac{\partial^2 f}{\partial \mathrm{x}^2} + \dfrac{\partial^2 f}{\partial \mathrm{y}^2} \Rightarrow \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} + \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

Derivative-type Filters



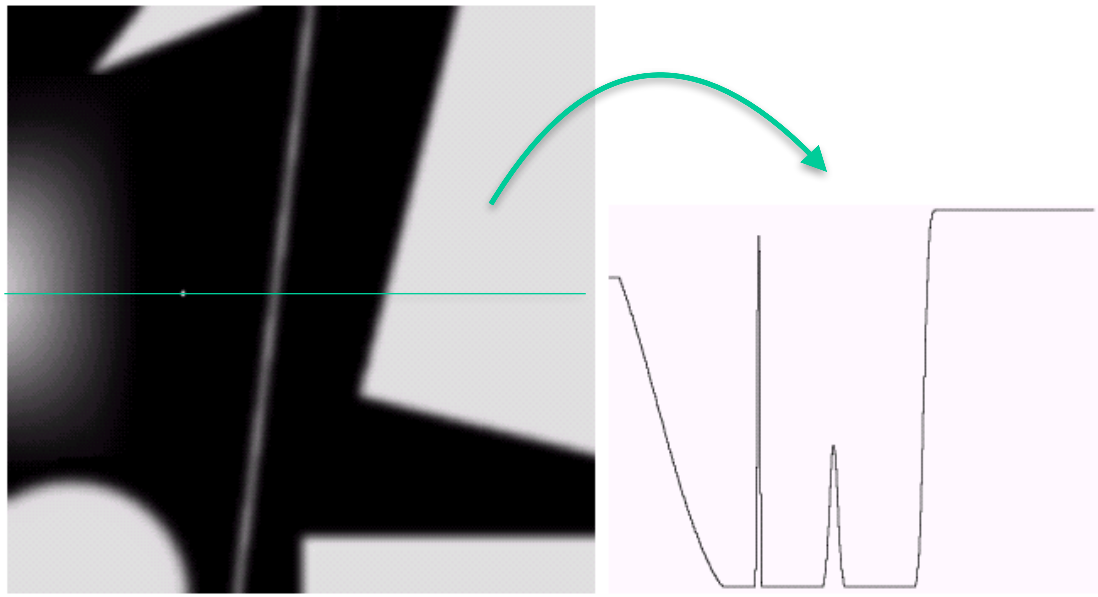
a
b
c

FIGURE 3.36 Illustration of the first and second derivatives of a 1-D digital function representing a section of a horizontal intensity profile from an image. In (a) and (c) data points are joined by dashed lines as a visualization aid.

Derivative-type Filters

a b
c

FIGURE 3.38
(a) A simple image. (b) 1-D horizontal gray-level profile along the center of the image and including the isolated noise point.
(c) Simplified profile (the points are joined by dashed lines to simplify interpretation).



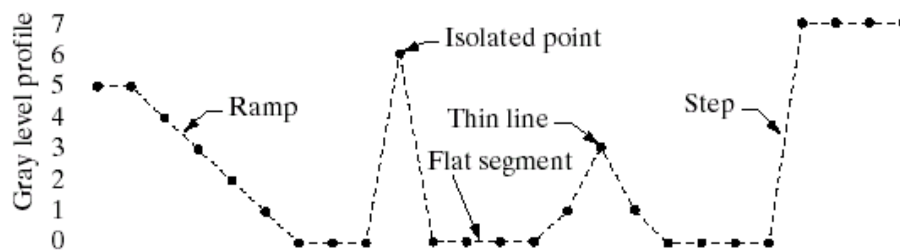


Image strip	5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	7	7	7	7	•	•
First Derivative		−1	−1	−1	−1	−1	0	0	6	−6	0	0	0	1	2	−2	−1	0	0	0	7	0	0	0		
Second Derivative		−1	0	0	0	0	1	0	6	−12	6	0	0	1	1	−4	1	1	0	0	7	−7	0	0		

Second Derivative for Sharpening

Variations of the Laplacian Filter

Laplacian: $\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \Rightarrow \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} + \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

This filter gives the derivative at the mid-point

Same response in row/column directions

Consider: $\begin{bmatrix} 1 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 \\ 0 & -2 & 0 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & -4 & 0 \\ 1 & 0 & 1 \end{bmatrix}$

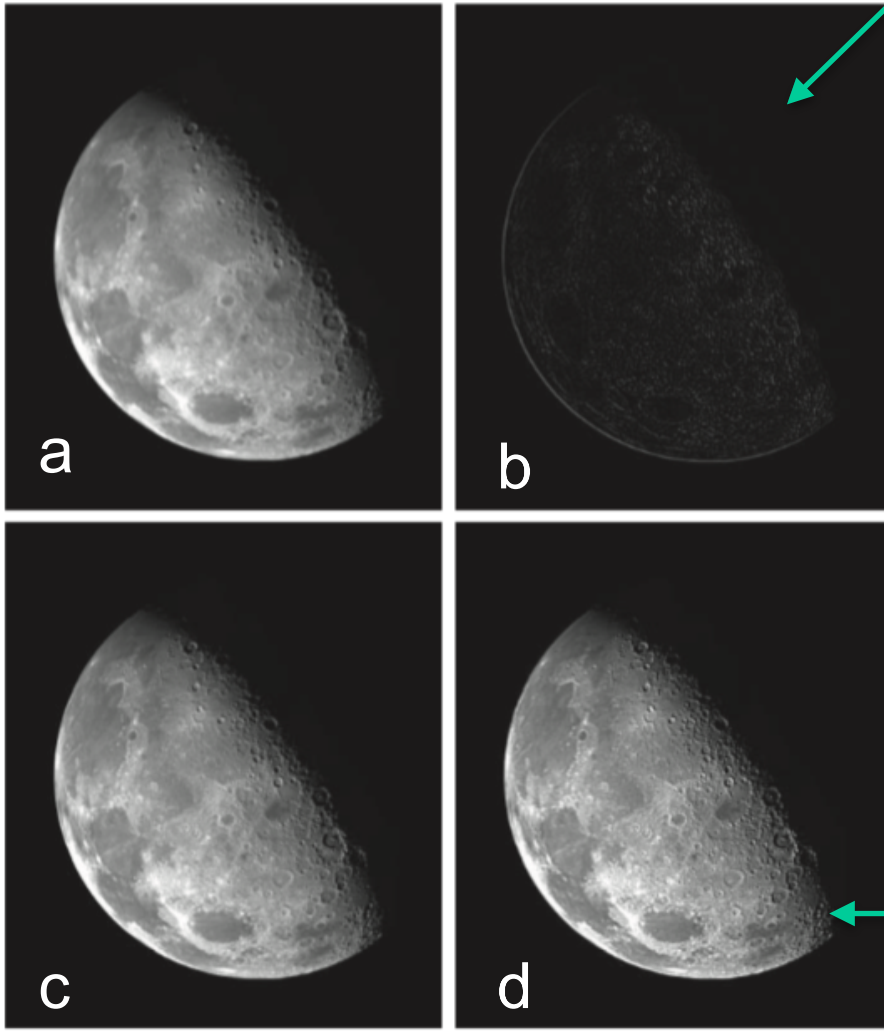
Same response in diagonal directions

Together: $\begin{bmatrix} 1 & 0 & 1 \\ 0 & -4 & 0 \\ 1 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

"Isotropic" filter

To sharpen the image: add/substract the Laplacian image from the original

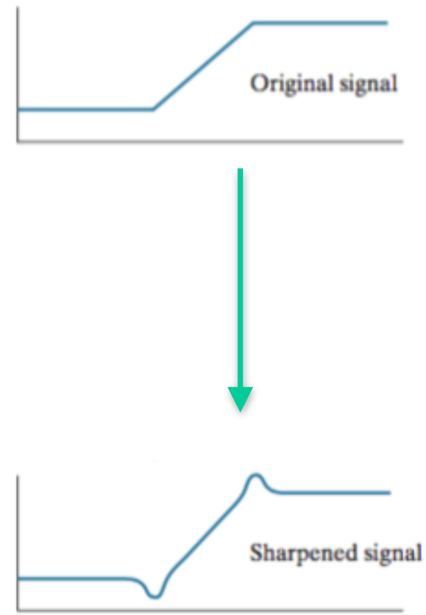
$$g(x,y) = f(x,y) + c\nabla^2 f(x,y)$$



a b
c d

FIGURE 3.46 (a) Blurred image of the North Pole of the moon. (b) Laplacian image obtained using the kernel in Fig. 3.45(a). (c) Image sharpened using Eq. (3-54) with $c = -1$. (d) Image sharpened using the same procedure, but with the kernel in Fig. 3.45(b). (Original image courtesy of NASA.)

0	1	0
1	−4	1
0	1	0



1	1	1
1	−8	1
1	1	1

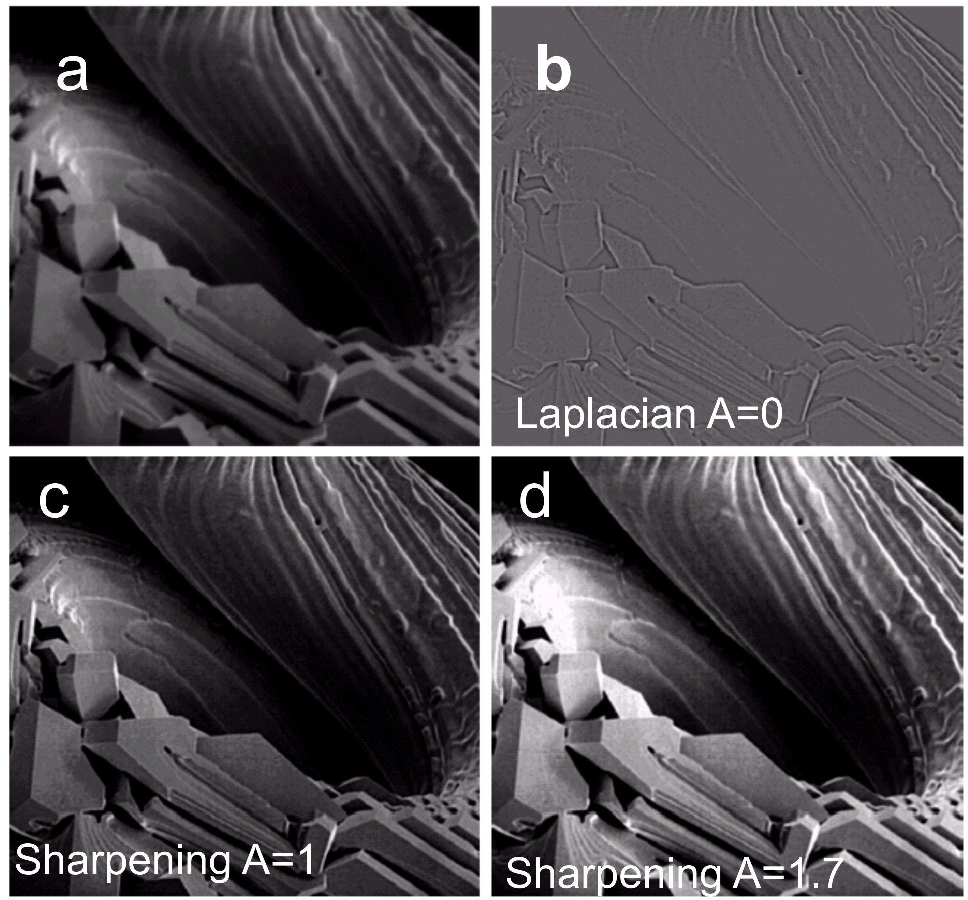
$$g(x,y) = f(x,y) + c\left[\nabla^2 f(x,y)\right] \qquad (3\text{-}54)$$

Sharpening Using the Laplacian Filter

0	−1	0
−1	$A + 4$	−1
0	−1	0

−1	−1	−1
−1	$A + 8$	−1
−1	−1	−1

This Figure is from G&W 2nd Edition



a b
c d

FIGURE 3.43 (a) Same as Fig. 3.41(c), but darker. (a) Laplacian of (a) computed with the mask in Fig. 3.42(b) using $A = 0$. (c) Laplacian enhanced image using the mask in Fig. 3.42(b) with $A = 1$. (d) Same as (c), but using $A = 1.7$.

Using First Derivative for Image Sharpening The Gradient

(a)

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

(b)

−1	0
0	1

(c)

0	−1
1	0

(d)

−1	−2	−1
0	0	0
1	2	1

(e)

−1	0	1
−2	0	2
−1	0	1

Sobel operators

a
b c
d e

FIGURE 3.41 A 3 × 3 region of an image (the zs are intensity values). (b)–(c) Roberts cross gradient operators. (d)–(e) Sobel operators. All the mask coefficients sum to zero, as expected of a derivative operator.

$$\nabla f \equiv \text{grad}(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

$$M(x, y) = \|\nabla f\| = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2}$$

Magnitude of the gradient

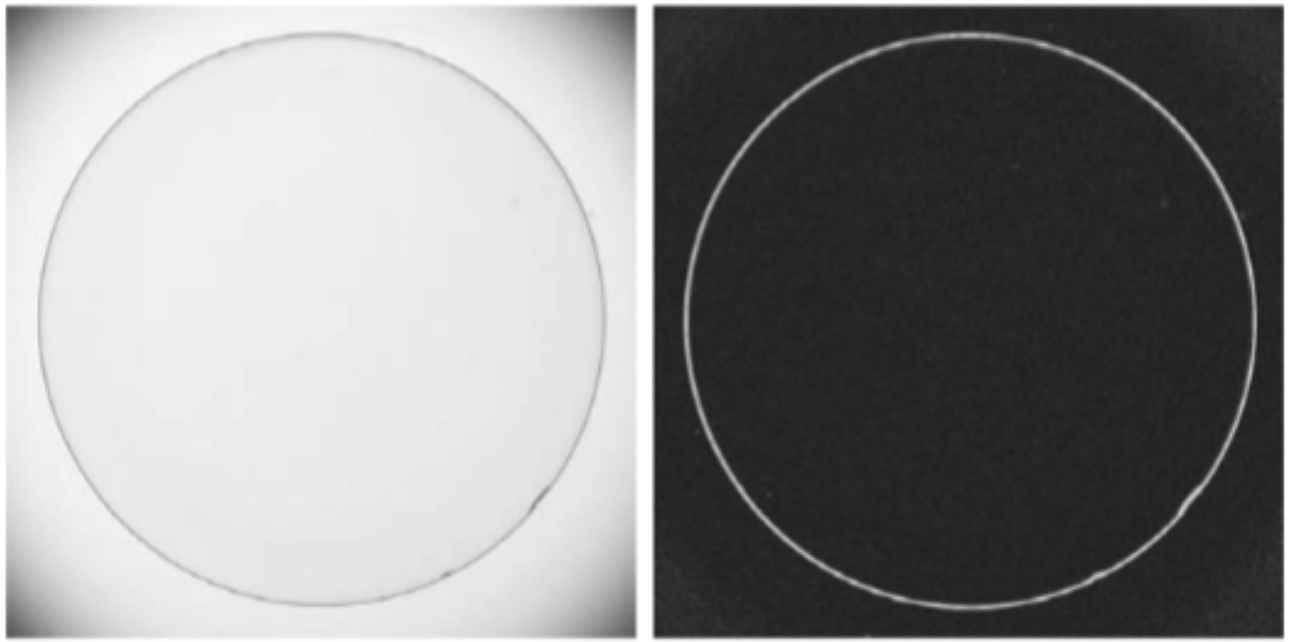
This is what we called "the gradient image"

Try these filters with your own images in MATLAB

Example of Sobel Gradient

a b

FIGURE 3.51
(a) Image of a contact lens (note defects on the boundary at 4 and 5 o'clock).
(b) Sobel gradient.
(Original image courtesy of Perceptics Corporation.)



original

Sobel gradient

−1	0	1
−2	0	2
−1	0	1

Sobel filter, x-direction

Sobel filter, y-direction

−1	−2	−1
0	0	0
1	2	1

$$M(x,y) = \|\nabla f\| = \mathrm{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2}$$

Further reading: *https://www.sciencedirect.com/topics/computer-science/sobel-filter*

Unsharp masking & highboost filtering

- Unsharp mask: Subtracting an unsharp (smoothed) version of an image from the original

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

Original image *blurred image*

- Image enhancement: Add the mask back to the original with a coefficient k

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

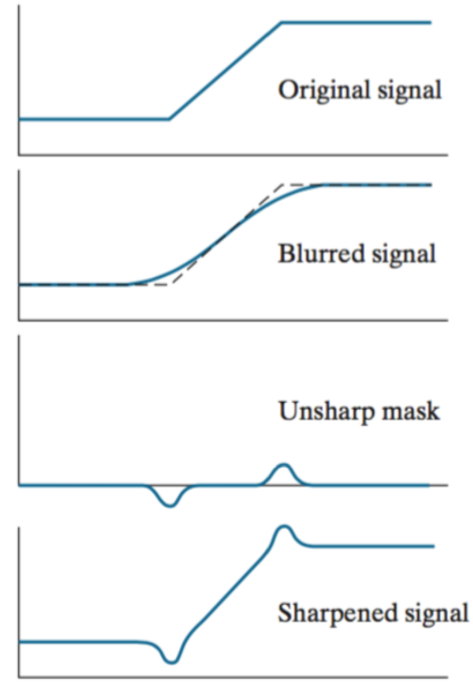
image sharpening

★ k= 1, unsharp masking
★ k>1, the process is referred to as ***High-boost filtering***
★ k<1 de-emphasize the contribution of the unsharp mask

Unsharp masking & highboost filtering

Original

Blurred image



Unsharp mask

Unsharp image k=1

Highboost filter k=4.5

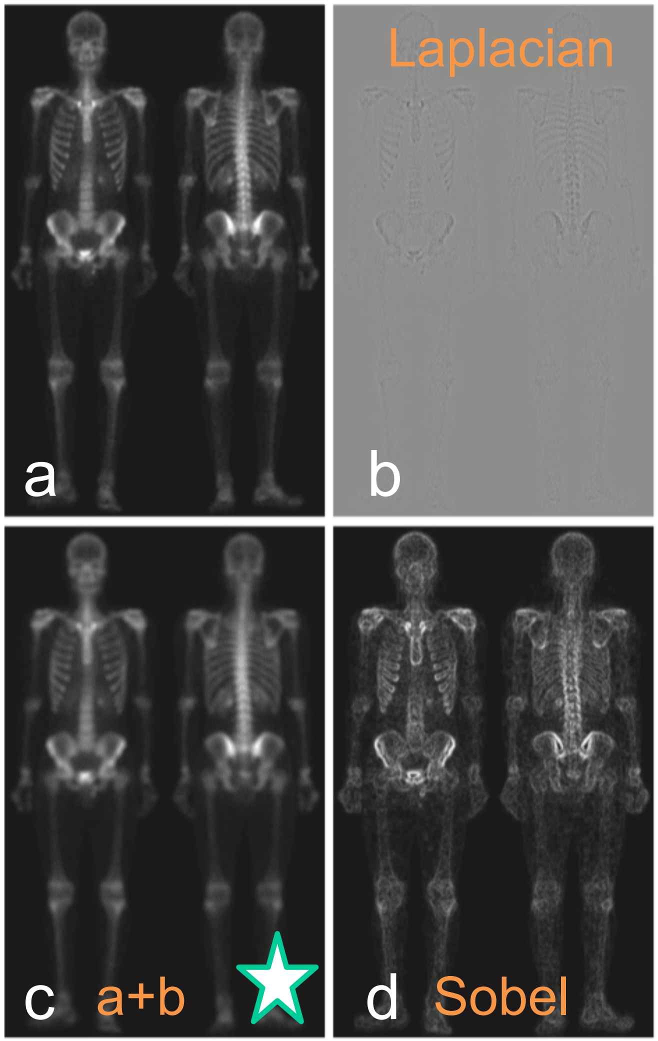
Intensity profiles

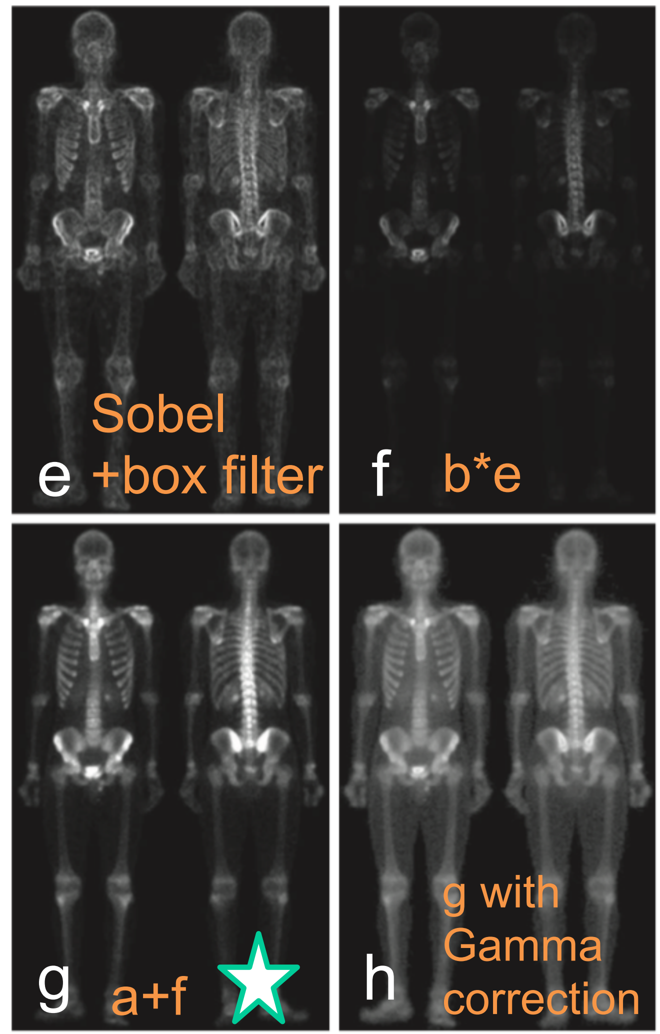
$$g(x,y) = f(x,y) + k\,g_{\text{mask}}(x,y)$$

Example Applications

a b
c d

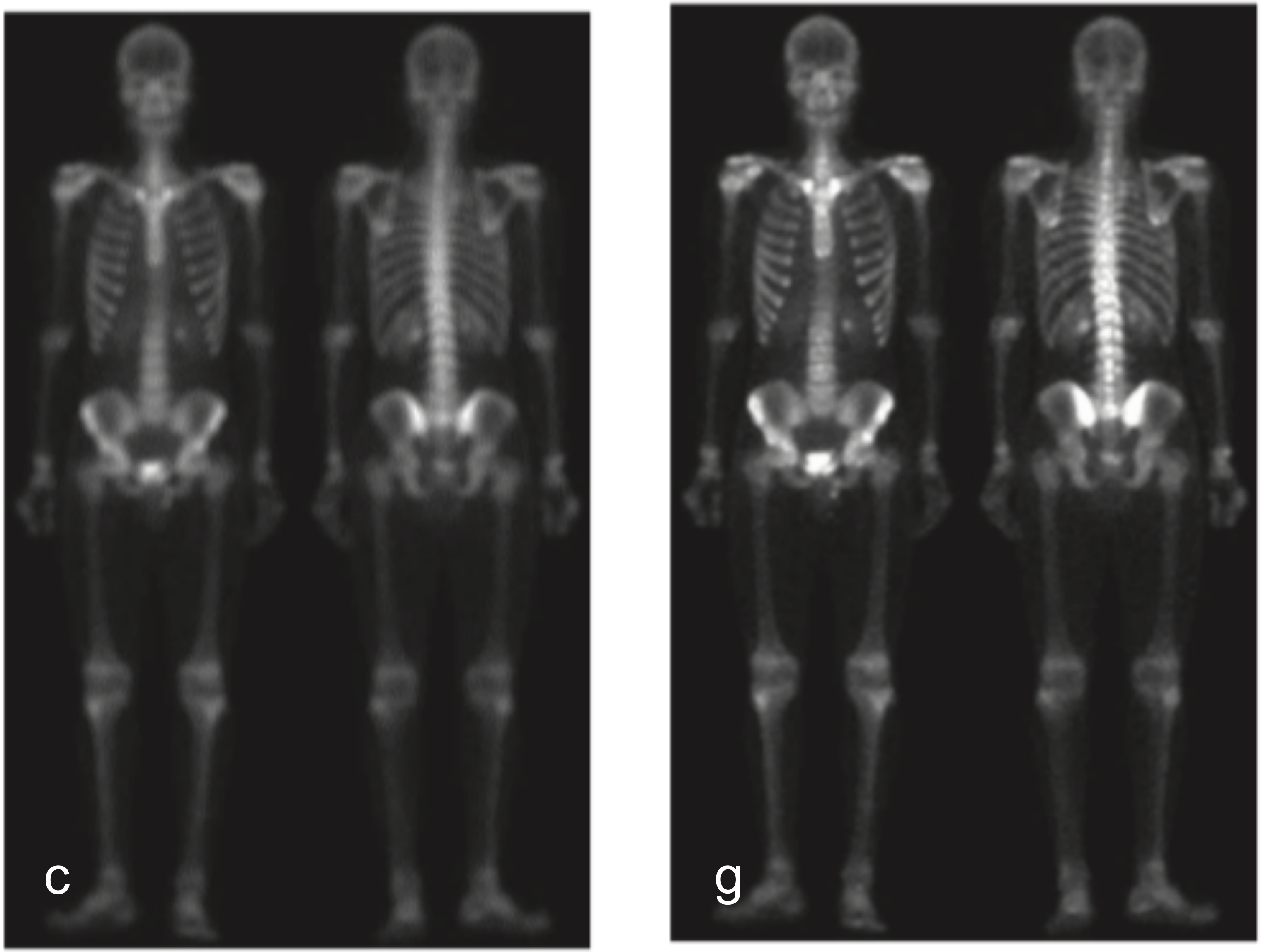
FIGURE 3.57
(a) Image of whole body bone scan. (b) Laplacian of (a). (c) Sharpened image obtained by adding (a) and (b). (d) Sobel gradient of image (a). (Original image courtesy of G.E. Medical Systems.)





e f
g h

FIGURE 3.57
(Continued)
(e) Sobel image smoothed with a 5×5 box filter. (f) Mask image formed by the product of (b) and (e). (g) Sharpened image obtained by the adding images (a) and (f). (h) Final result obtained by applying a power-law transformation to (g). Compare images (g) and (h) with (a). (Original image courtesy of G.E. Medical Systems.)


c
g

Summary

- How spatial filters work
- Linear averaging filters
- Order-statistics filters
- Derivative-type of filters - image sharpening
- Unsharp mask and Sobel gradient

Textbook reading

- Chap 3: P153-155, 165-166, 174-188

Food-for-thought questions

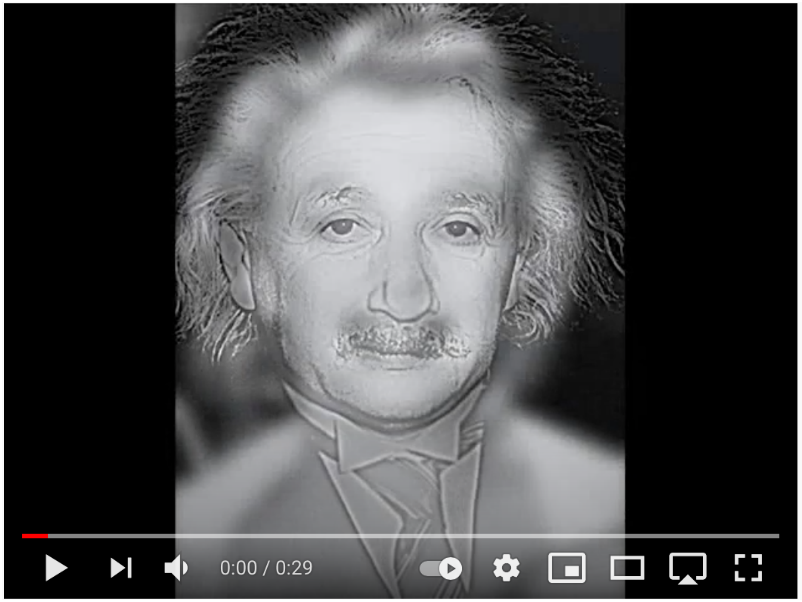
- What are effects of low-pass and high-pass filters?
- What types of noise is best dealt with by the median filter?
- Can the sum of all elements in an averaging filter >1? How do we choose the size of an average filter?
- What are the properties of the 1st order derivative filters?
- What are the properties of the 2nd order derivative filters?
- When there is a sharpening step in the image, how will results of 1st and 2nd order derivative filter differ?
- Sobel filter vs Laplacian filter: isotropic?
- Why a high-boost filter can create halo effects? and will the results from high-boost filter have negative values?

High frequency vs. low frequency

a b c

FIGURE 3.42 (a) Image shaded by a shading pattern oriented in the $-45°$ direction. (b) Estimate of the shading patterns obtained using lowpass filtering. (c) Result of dividing (a) by (b). (See Section 9.8 for a morphological approach to shading correction).

High frequency vs. low frequency



https://www.youtube.com/watch?v=x2yVBIGyQaY