CMPUT 206: FILTERING BASICS

Nilanjan Ray, Computing Science, University of Alberta

Contents

- In this lecture we will look at spatial filtering techniques:
  - Basics of Spatial Filtering
  - Convolution
  - Smoothing Spatial Filters
  - Sharpening Spatial Filters
  - Combining Spatial Enhancement Methods
  - Basic Non-linear Spatial Filters

Basics of Spatial Filtering

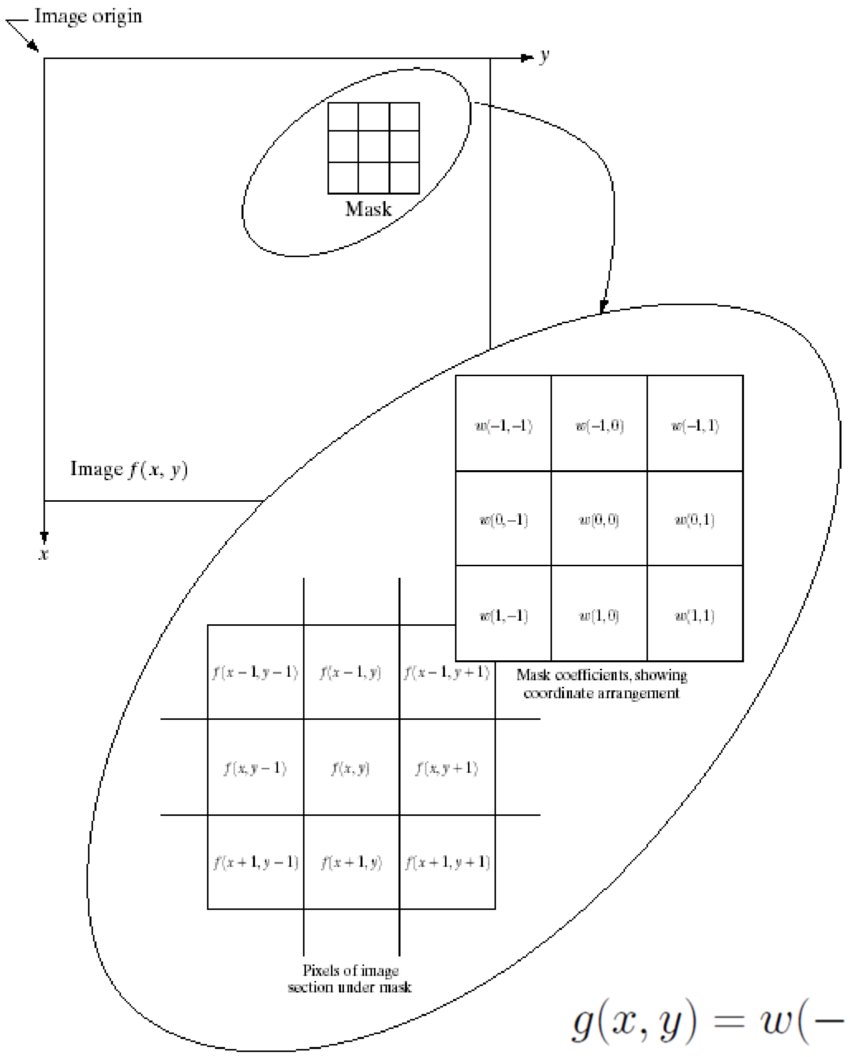


FIGURE 3.32 The mechanics of spatial filtering. The magnified drawing shows a 3×3 mask and the image section directly under it; the image section is shown displaced out from under the mask for ease of readability.

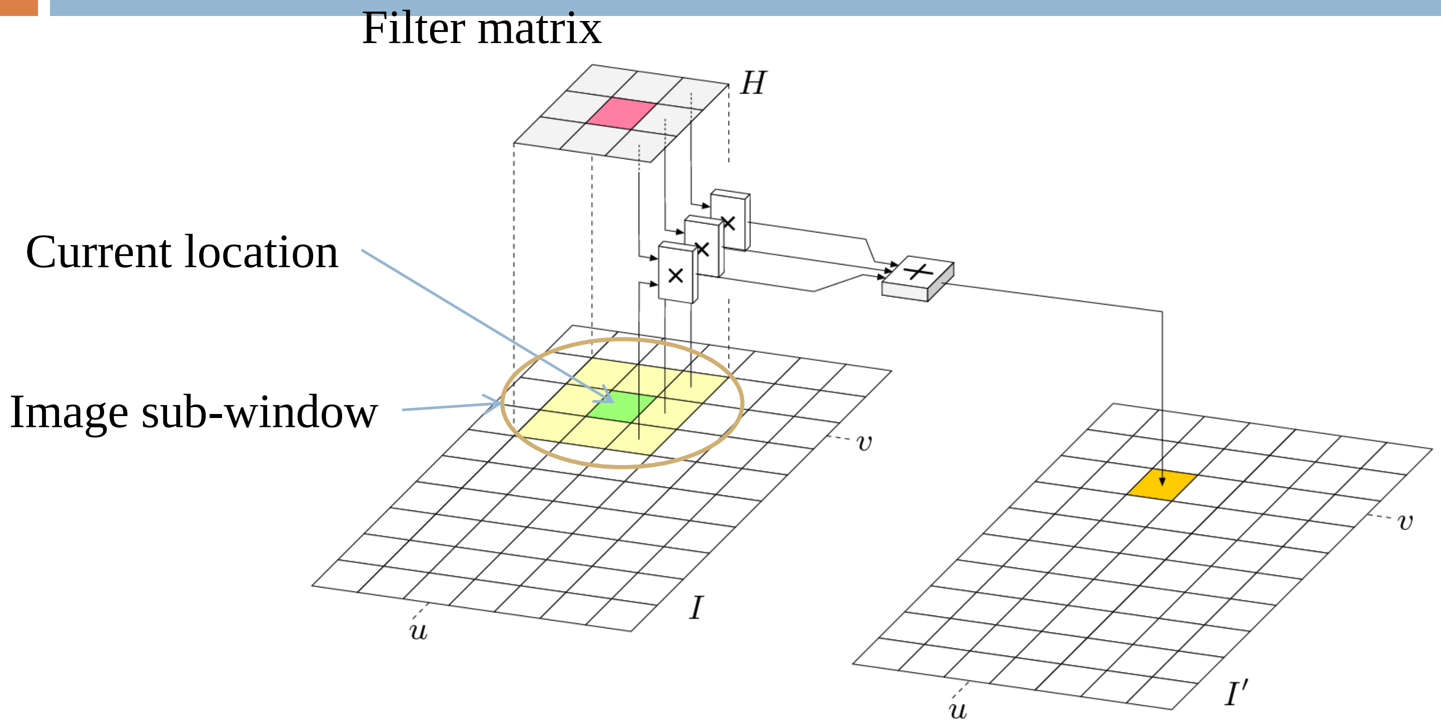
The name “filter” has been borrowed from the signal processing community

It refers to accepting / rejecting frequency components

We will learn about frequency later

$$g(x,y) = w(-1,-1)f(x-1,y-1) + \cdots + w(1,1)f(x+1,y+1)$$

Linear filters: how to compute?

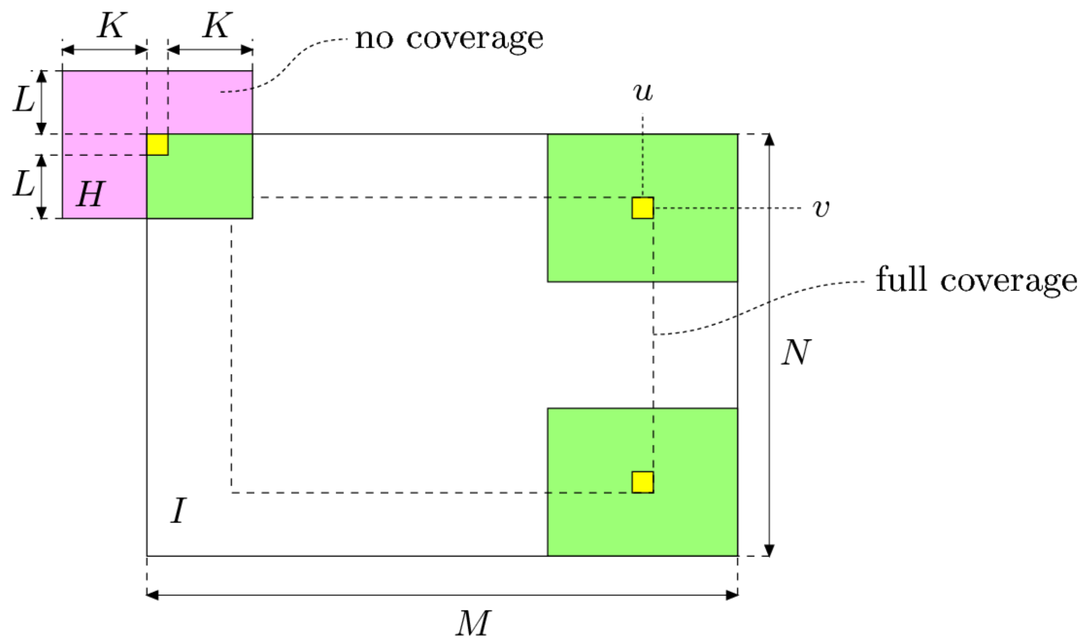


Do until all pixel locations are covered:

Step 1: At current location point-wise multiply and add image sub-window and filter matrix

Step 2: Shift current location, and go to step 1

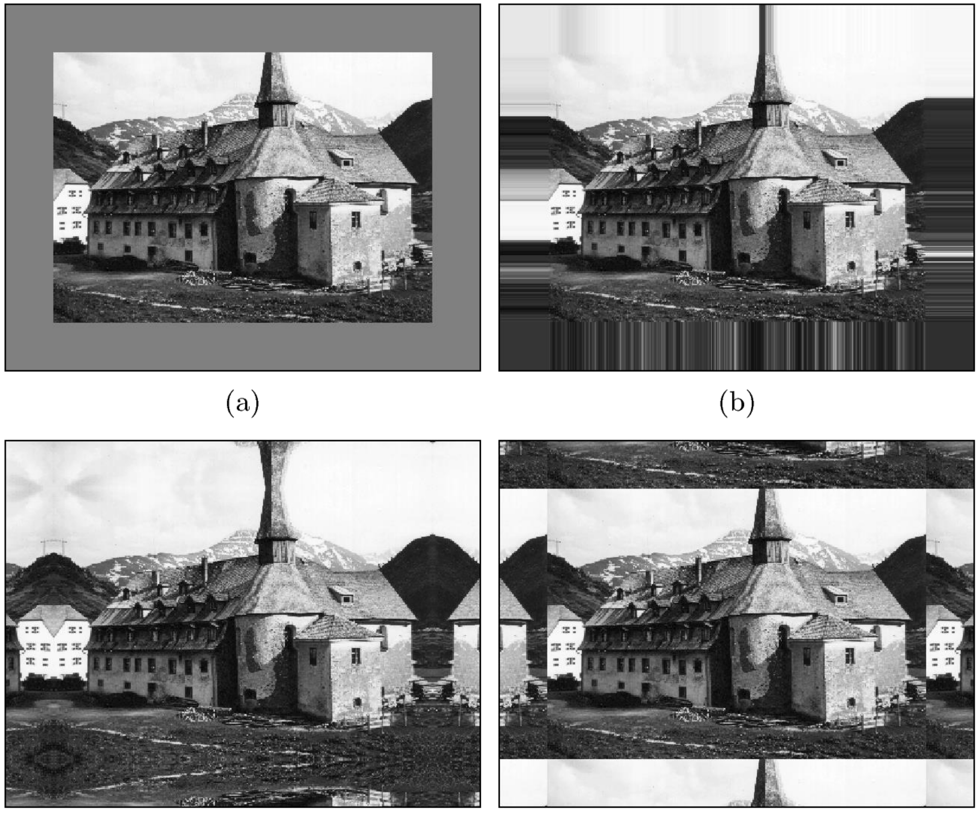
Linear filters: image borders



Why does image borders pose problems in filtering? How about point operations?

Let’s discuss some strategies to handle image borders in linear filtering.

Handing image borders


(a) (b) (c) (d)

Methods for extending the image to facilitate filtering along the borders. The assumption is that the (nonexisting) pixels outside the original image are either set to some constant value (a), take on the value of the closest border pixel (b), are mirrored at the image boundaries (c), or repeat periodically along the coordinate axes.

Linear filters: filter size

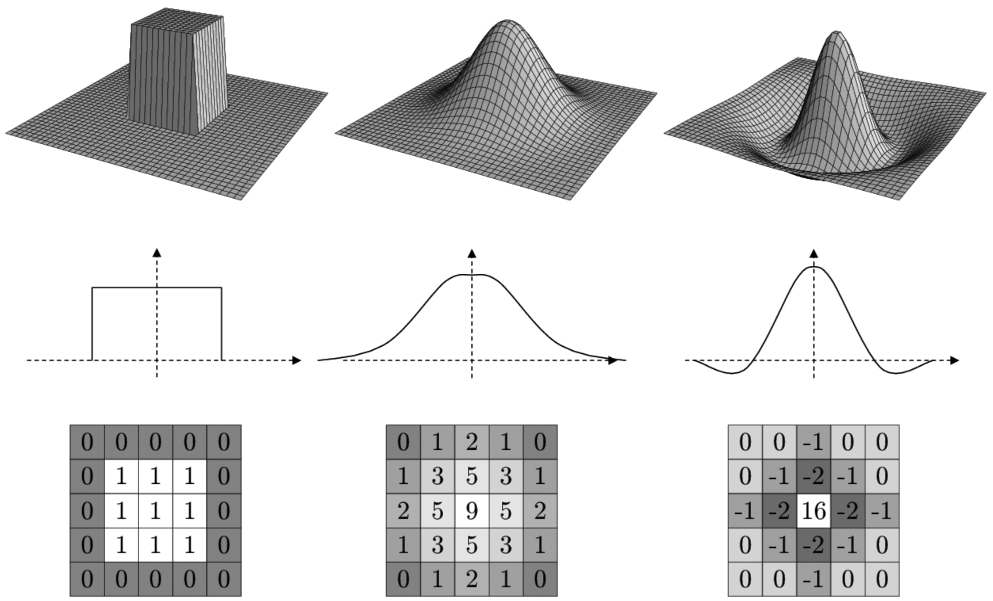
- Typically a filter matrix is of size $(2K+1)$-by-$(2K+1)$, *i.e.*, odd matrix size– why?
- Computational complexity increases with filter size
- For really large filter matrix, a trick called frequency domain processing is employed– this computationally very attractive

Types of linear filters

- Essentially of two types
  - Smoothing filters, typically used in
    - Noise reduction
    - Segmentation
  - Difference filters, typically used in
    - Edge detection
    - Image sharpening

Types of linear filters in pictures

- Often visualization of filter matrix reveals the nature of a linear filter



0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

(a)

0	1	2	1	0
1	3	5	3	1
2	5	9	5	2
1	3	5	3	1
0	1	2	1	0

(b)

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

(c)

(a) Box filter, (b) Gaussian filter, (c) Laplace filter: Can you guess their types?

Convolution

- Convolution in 1D

$$g(x) = (w * f)(x) = \sum_{s=-\infty}^{\infty} w(s) f(x-s) = \sum_{s=-\infty}^{\infty} w(x-s) f(s)$$

- Convolution in 2D

$$g(x,y) = (w * f)(x,y) = \sum_{s=-\infty}^{\infty} \sum_{t=-\infty}^{\infty} w(s,t) f(x-s, y-t) = \sum_{s=-\infty}^{\infty} \sum_{t=-\infty}^{\infty} w(x-s, y-t) f(s,t)$$

- Convolution = Flip, shift, multiply and add

- IMPORTANT: Spatial Linear Filter = Convolution

Convolution in 1D

- Let’s examine a demo page: http://bmia.bmt.tue.nl/education/courses/fev/course/notebooks/Convolution.html

Correlation

- Correlation in 1D

$$g(x) = (w \otimes f)(x) = \sum_{s=-\infty}^{\infty} w(s) f(x+s)$$

- Correlation in 2D

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

Correlations and convolutions give same results when at least one of *f* and *w* is symmetric.

Convolution as a matrix multiplication

$$
\begin{aligned}
y &= h * x \\
&= \begin{bmatrix}
h_1 & 0 & \dots & 0 & 0 \\
h_2 & h_1 & \dots & \vdots & \vdots \\
h_3 & h_2 & \dots & 0 & 0 \\
\vdots & h_3 & \dots & h_1 & 0 \\
h_{m-1} & \vdots & \dots & h_2 & h_1 \\
h_m & h_{m-1} & \vdots & \vdots & h_2 \\
0 & h_m & \dots & h_{m-2} & \vdots \\
0 & 0 & \dots & h_{m-1} & h_{m-2} \\
\vdots & \vdots & \vdots & h_m & h_{m-1} \\
0 & 0 & 0 & \dots & h_m
\end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} \\
y^T &= \begin{bmatrix} h_1 & h_2 & h_3 & \dots & h_{m-1} & h_m \end{bmatrix}
\begin{bmatrix}
x_1 & x_2 & x_3 & \dots & x_n & 0 & 0 & 0 & \dots & 0 \\
0 & x_1 & x_2 & x_3 & \dots & x_n & 0 & 0 & \dots & 0 \\
0 & 0 & x_1 & x_2 & x_3 & \dots & x_n & 0 & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \dots & 0 \\
0 & \dots & 0 & 0 & x_1 & \dots & x_{n-2} & x_{n-1} & x_n & \vdots \\
0 & \dots & 0 & 0 & 0 & x_1 & \dots & x_{n-2} & x_{n-1} & x_n
\end{bmatrix}
\end{aligned}
$$

This concept may not be useful for “implementation,” but is often helpful for “theorem-proof”

Some properties of linear filters / convolutions

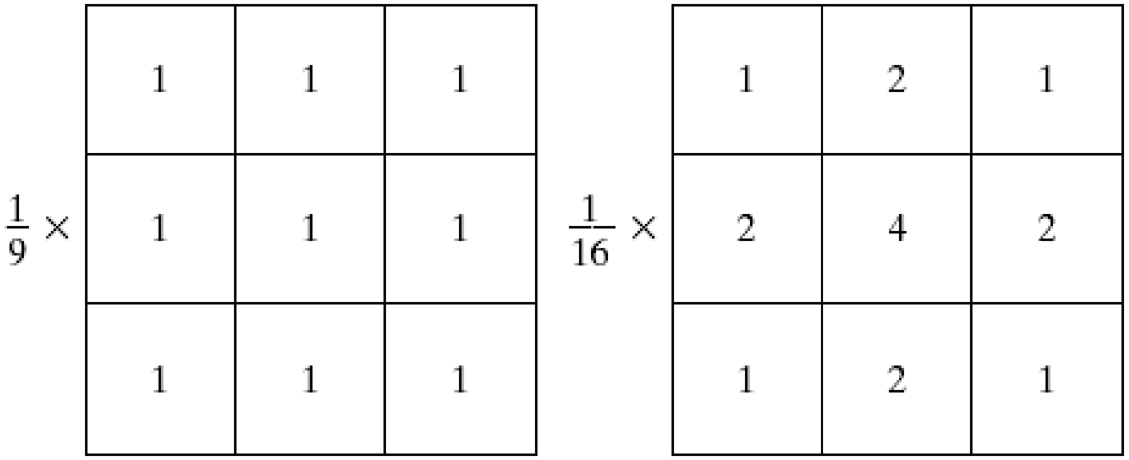
- Commutative $I * H = H * I$ "*" expresses linear filtering

- Linear
$$(s \cdot I) * H \quad = \quad I * (s \cdot H) \quad = \quad s \cdot (I * H)$$
$$(I_1 + I_2) * H \quad = \quad (I_1 * H) + (I_2 * H)$$

- Associative $A * (B * C) = (A * B) * C$

Smoothing Spatial Filters

- Used for blurring and for noise reduction
- Smoothing Linear Filters:
- Output is simply the average of the pixels contained in the neighbourhood of the filter mask.

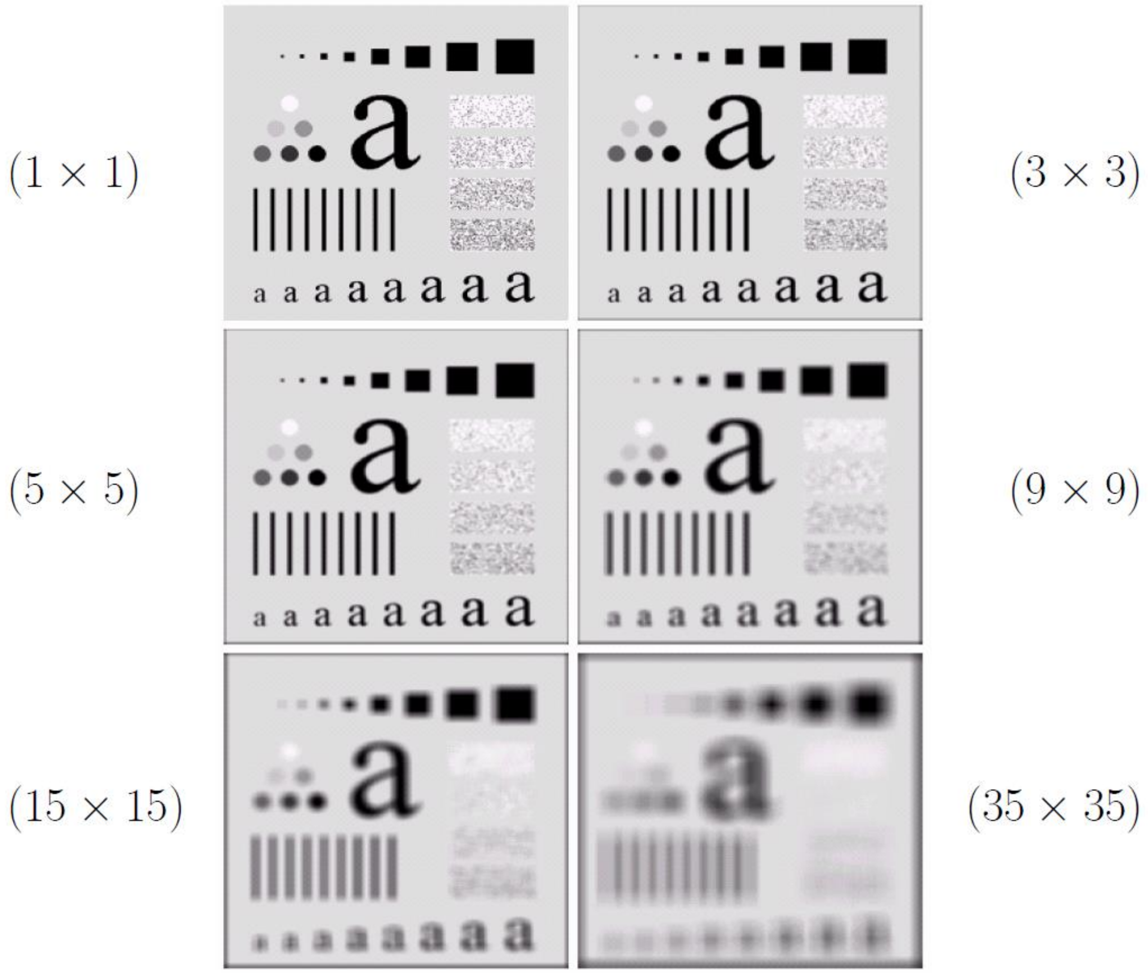


$$\frac{1}{9} \times \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad \frac{1}{16} \times \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

a b

FIGURE 3.34 Two 3 × 3 smoothing (averaging) filter masks. The constant multipli er in front of each mask is equal to the sum of the values of its coefficients, as is required to compute an average.

Smoothing Spatial Filters



Examples of Smoothing Filters…

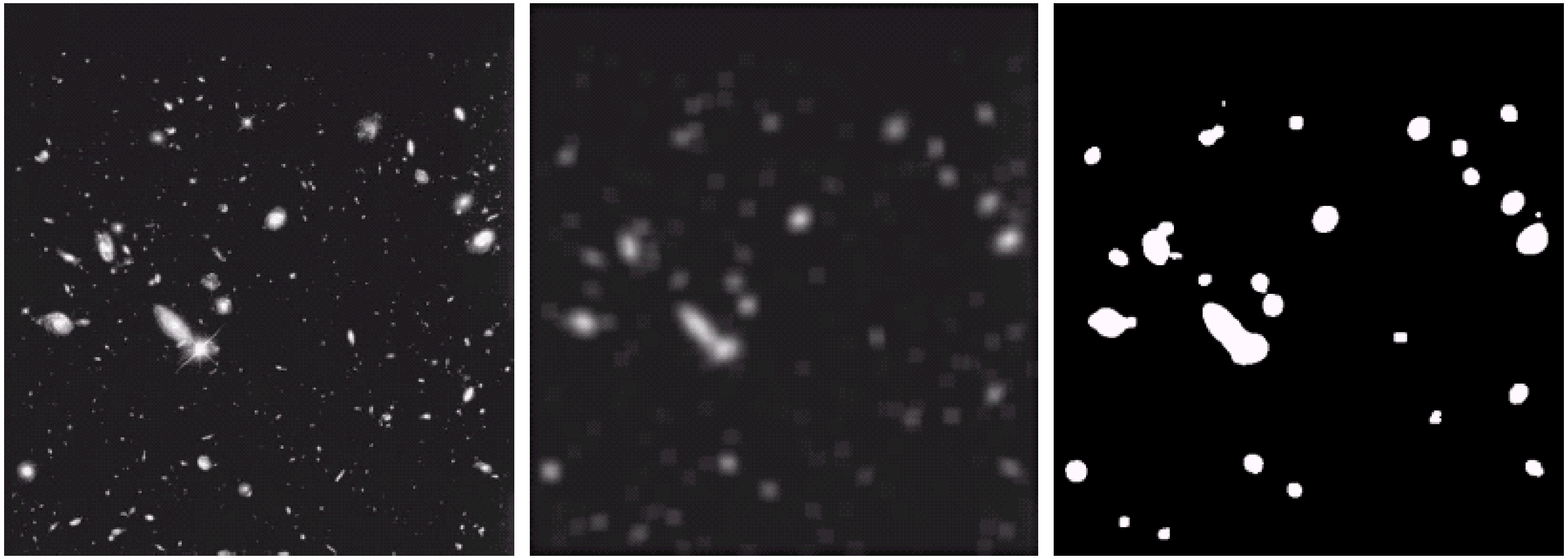


FIGURE 3.36 (a) Image from the Hubble Space Telescope. (b) Image processed by a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

Sharpening Spatial Filters

- Purpose: highlight fine detail
- Smoothing: pixel averaging = integration
- Sharpening: differentiation = enhances edges and deemphasizes

Foundation

- We will talk about some sharpening filters that are based on one and two dimensional derivatives
- Derivatives of digital functions are defined in terms of differences
- Requirements for definitions:

First-order derivative
(1)Zero in flat segments
(2) Nonzero at onset of step or ramp
(3) Nonzero along ramp

Second-order derivative
(1)Zero in flat segments
(2)Nonzero at onset of step or ramp
(3) Zero along ramp of constant slope

Discrete Differentiation

First-order derivative: $$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

Second-order derivative: $$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

Makes sense intuitively, but
where are these formulas coming from?

Examples of discrete derivatives

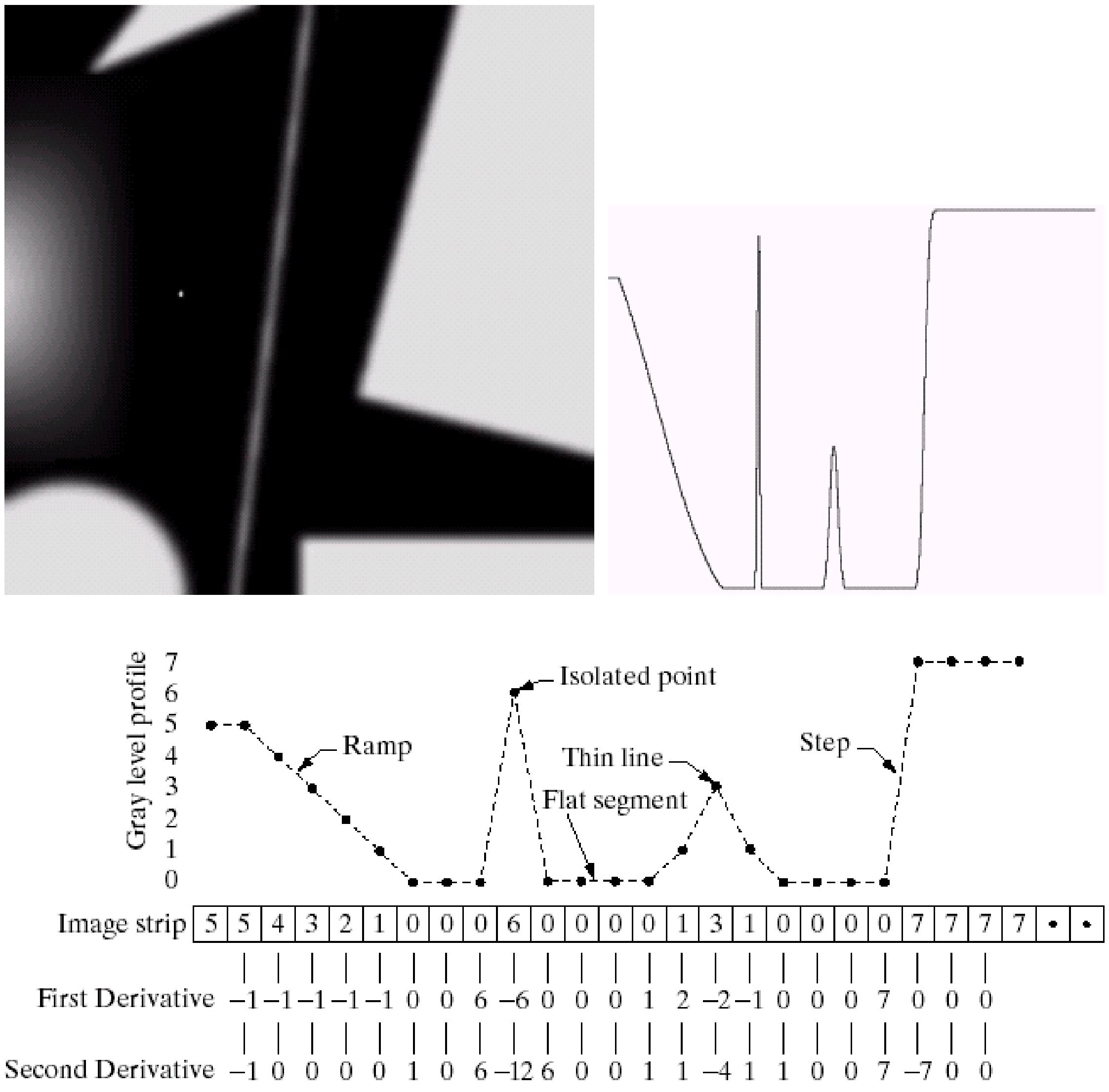


Image strip: 5 5 4 3 2 1 0 0 0 6 0 0 0 0 1 3 1 0 0 0 0 7 7 7 7 • •

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

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).

Properties of discrete derivatives

- Comparing the response we can say
  - 1st-order derivatives produce thicker edges
  - 2nd-order derivatives: stronger response to fine detail
  - 1st-order derivatives: stronger response to step
  - 2nd-order derivatives: double response at step changes
  - 2nd-order derivatives: stronger response to line than step

Use of Second Derivatives - The Laplacian

- We are interested in Isotropic filters which is rotation invariant
- Development of the method:
- Leplacian derivative operator

Continuous form:

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

Laplacian – discrete form

Discrete form: x-direction

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

Discrete form: y-direction

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

Can be derived from continuous forms

Laplacian – discrete form…

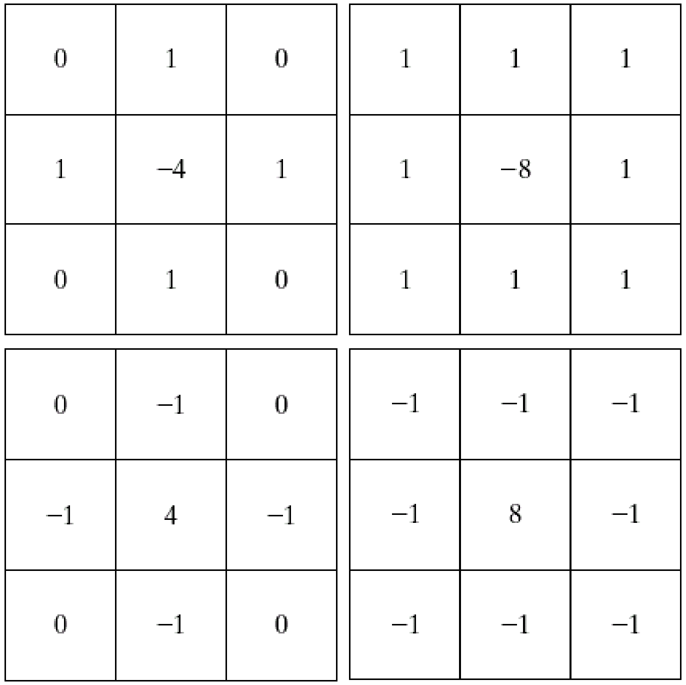
- Discrete form: 2-D Laplacian - sum of the x and y components

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

Laplacian is one of the most important filters used in image sharpening

2D Discrete Laplacian: Implementation

- Implementation



(a)

0	1	0
1	−4	1
0	1	0

(b)

1	1	1
1	−8	1
1	1	1

(c)

0	−1	0
−1	4	−1
0	−1	0

(d)

−1	−1	−1
−1	8	−1
−1	−1	−1

a b
c d

FIGURE 3.39 (a) Filter mask used to implement the digital Laplacian, as defined in Eq. (3.7-4). (b) Mask used to implement an extension of this equation that includes the diagonal neighbors. (c) and (d) Two other implementations of the Laplacian.

(a) and (c): Isotropic results for increments of 90 degree
(b) and (d): Isotropic results for increments of 45 degree

Image Enhancement with Laplacian

- The basic way of using Laplacian for image enhancement:

$$g(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y), & \text{if the center coefficient of the Laplacian mask is negative} \\ f(x,y) + \nabla^2 f(x,y), & \text{if the center coefficient of the Laplacian mask is positive} \end{cases}$$

Image Enhancement Example

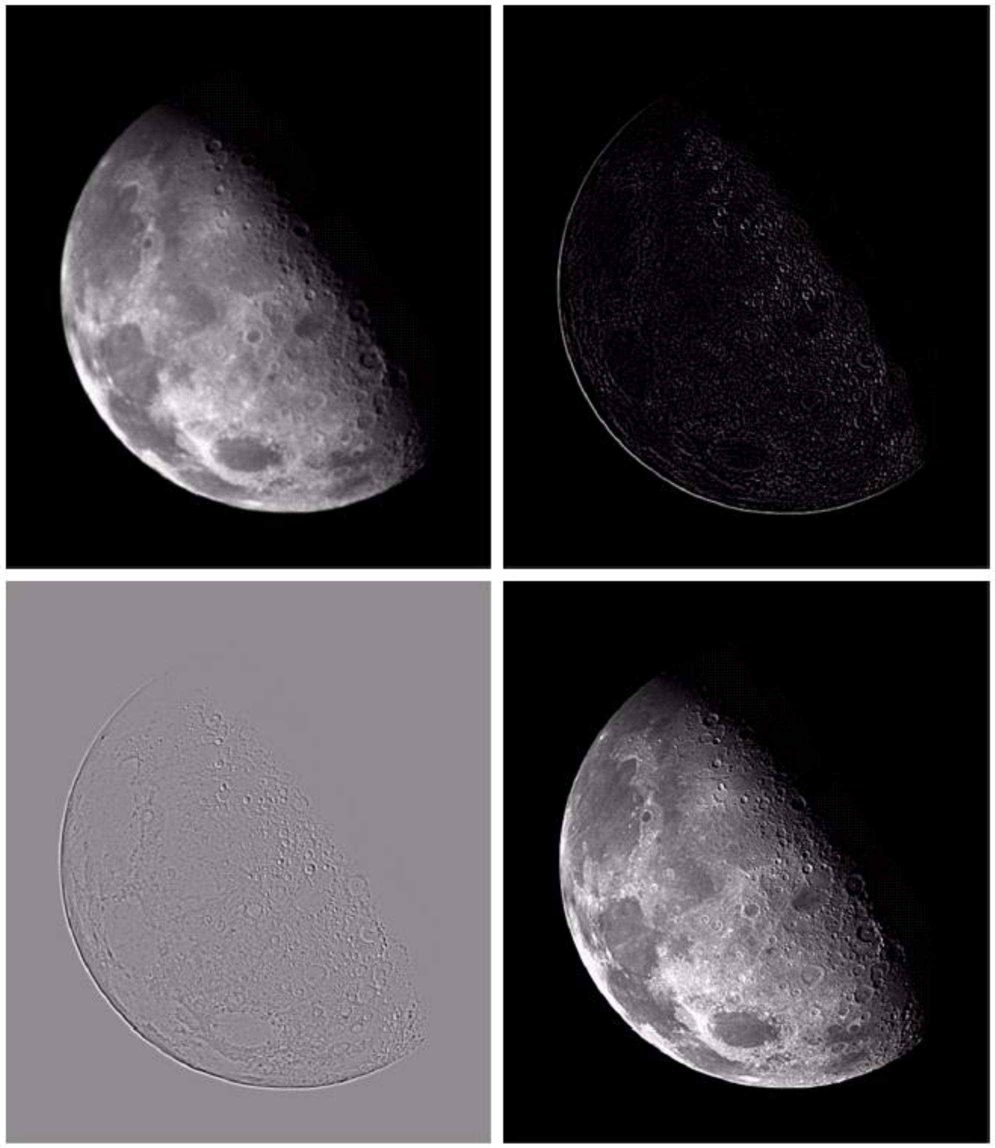
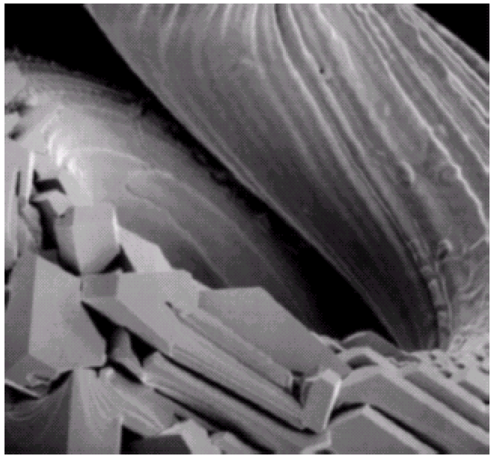


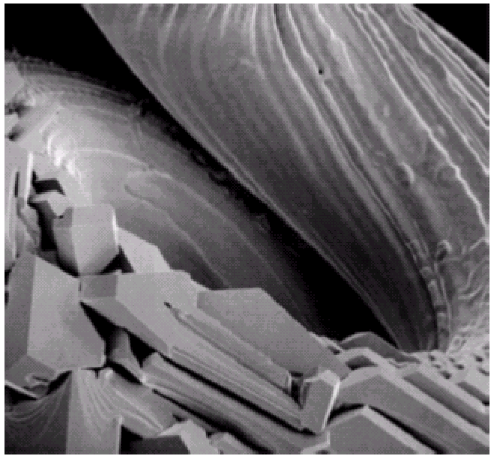
FIGURE 3.40 (a) Image of the North Pole of the moon. (b) Laplacian-filtered image. (c) Laplacian image scaled for display purposes. (d) Image enhanced by using Eq. (3.7-5). (Original image courtesy of NASA.)

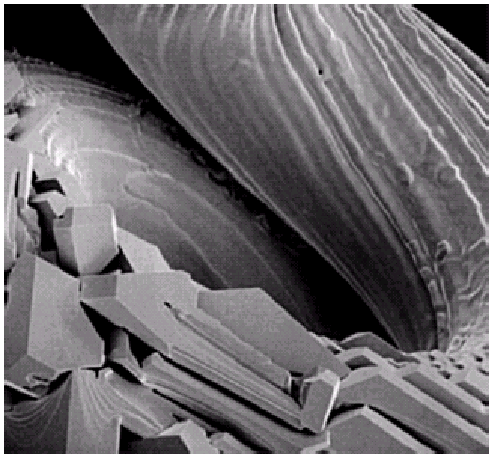
Image Enhancement Filter

0	−1	0
−1	5	−1
0	−1	0

−1	−1	−1
−1	9	−1
−1	−1	−1







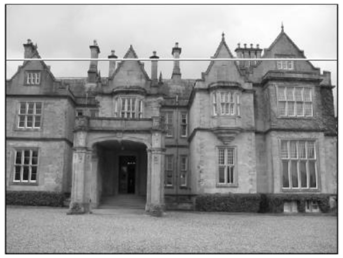
a b c
d e

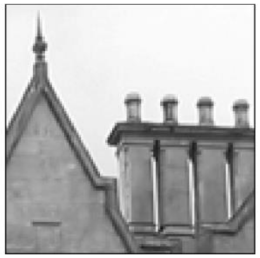
FIGURE 3.41 (a) Composite Laplacian mask. (b) A second composite mask. (c) Scanning electron microscope image. (d) and (e) Results of filtering with the masks in (a) and (b), respectively. Note how much sharper (e) is than (d). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

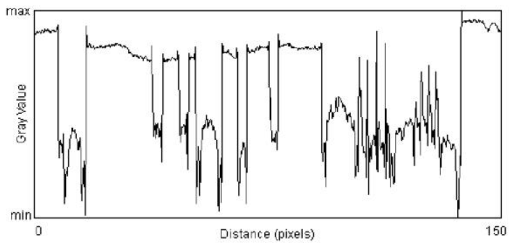
Unsharp Masking

- Unsharp masking:
  - Blur the image
  - Subtract the blurred image from the original; call this resulting image as mask
  - Multiply the mask image with a positive constant
  - Add the multiplied mask image to the original image

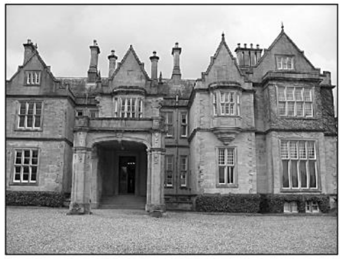
Unsharp masking: Example

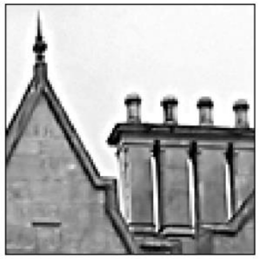

(a) Original

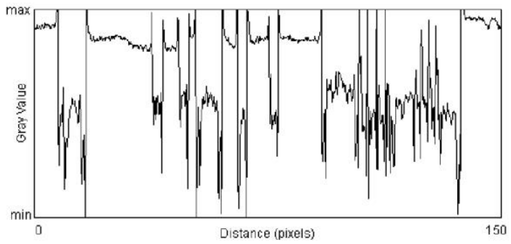

(b)



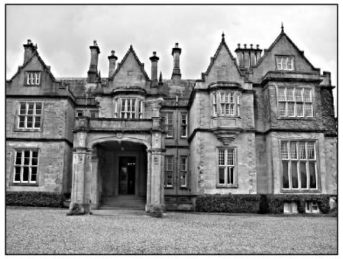
(c)

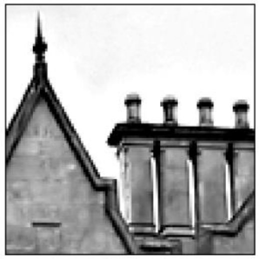

(d) $\sigma = 2.5$

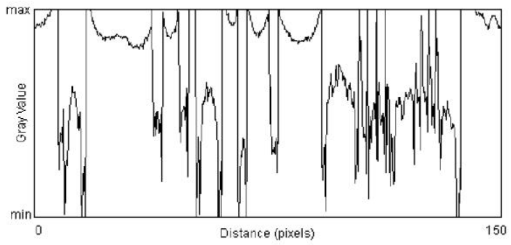

(e)



(f)


(g) $\sigma = 10.0$


(h)

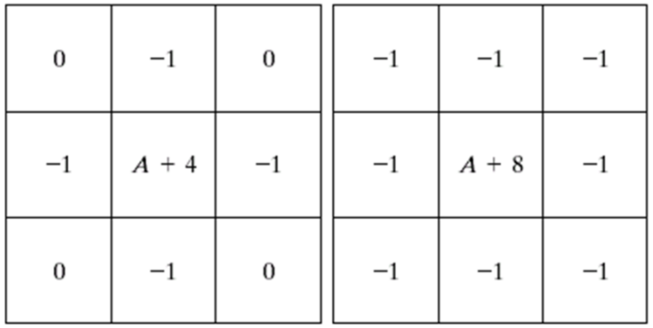


(i)

High-boost Filtering

- High-boost filtering

$$f_{hb}(x,y) = \begin{cases} Af(x,y) - \nabla^2 f(x,y), & \text{if the center coefficient of the Laplacian mask is negative} \\ Af(x,y) + \nabla^2 f(x,y), & \text{if the center coefficient of the Laplacian mask is positive} \end{cases}$$



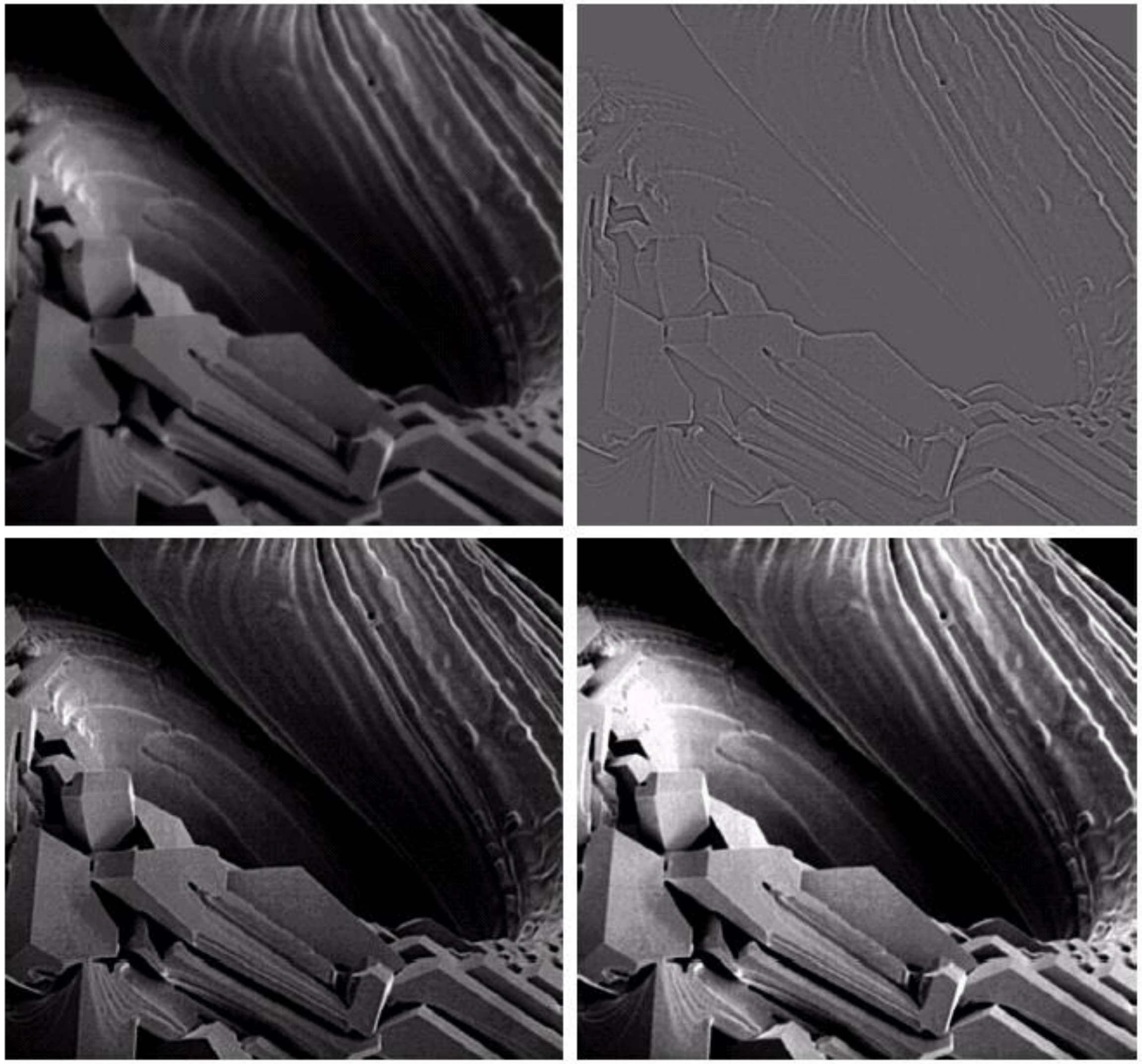
a

0	−1	0
−1	$A+4$	−1
0	−1	0

b

−1	−1	−1
−1	$A+8$	−1
−1	−1	−1

FIGURE 3.42 The high-boost filtering technique can be implemented with either one of these masks, with $A \geq 1$.



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 $A = 1.7$.

Image Gradient

- Use of First Derivatives - The Gradient

Continuous form: Gradient

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

Gradient Magnitude

Magnitude of gradient

$$\begin{aligned}\nabla f &= \mathrm{mag}(\nabla \mathbf{f}) \\ &= [G_x^2 + G_y^2]^{1/2} \\ &= \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2}\end{aligned}$$

Approximation: $\nabla f \approx |G_x| + |G_y|$

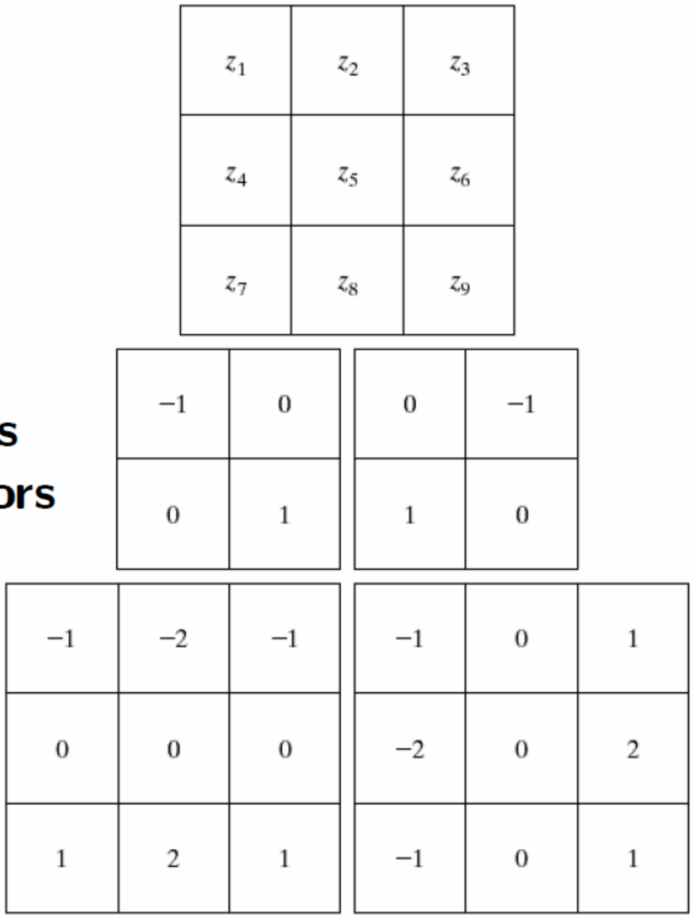
Gradient / Edge Operators

Discrete form: Roberts: $\nabla f \approx |z_9 - z_5| + |z_8 - z_6|$

Sobel:

$$\begin{aligned}\nabla f \;\approx\; & |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| \\ & + |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|\end{aligned}$$

Gradient / Edge Filters



a
b c
d e

FIGURE 3.44 A 3 × 3 region of an image (the z's are gray-level values) and masks used to compute the gradient at point labeled z_5. All masks coefficients sum to zero, as expected of a derivative operator.

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Roberts operators

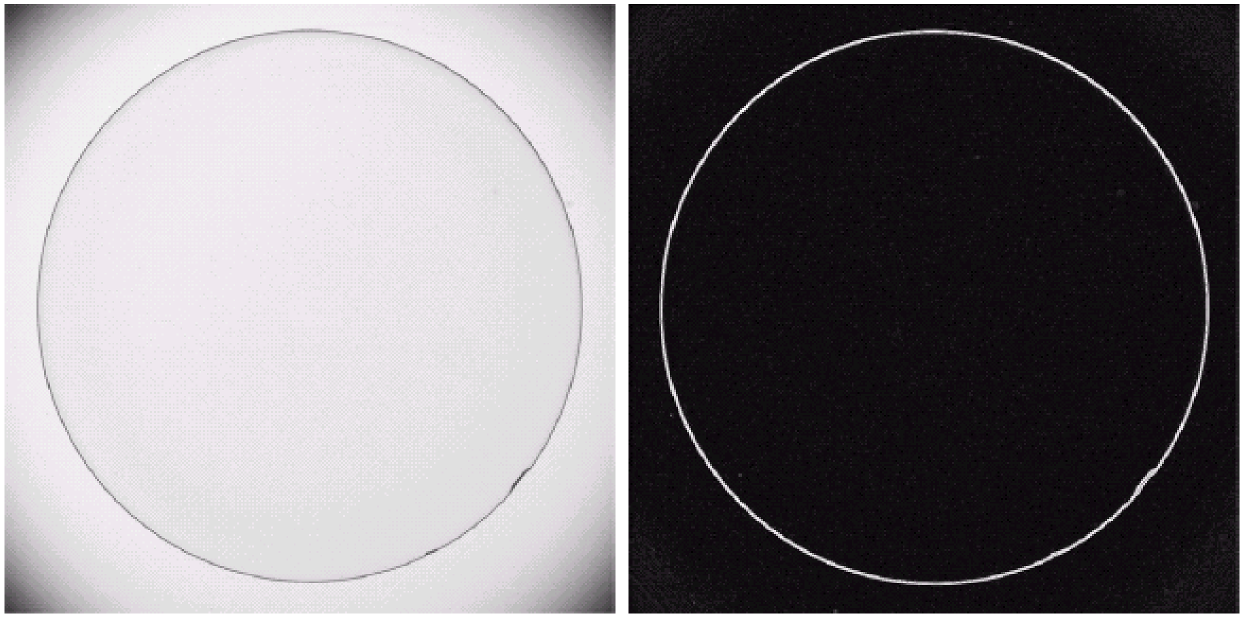
−1	0
0	1

0	−1
1	0

Sobel operators

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1



a b

FIGURE 3.45 Optical image of contact lens (note defects on the boundary at 4 and 5 o'clock). (b) Sobel gradient. (Original image courtesy of Mr. Pete Sites, Perceptics Corporation.)

Combining Spatial Enhancement Methods

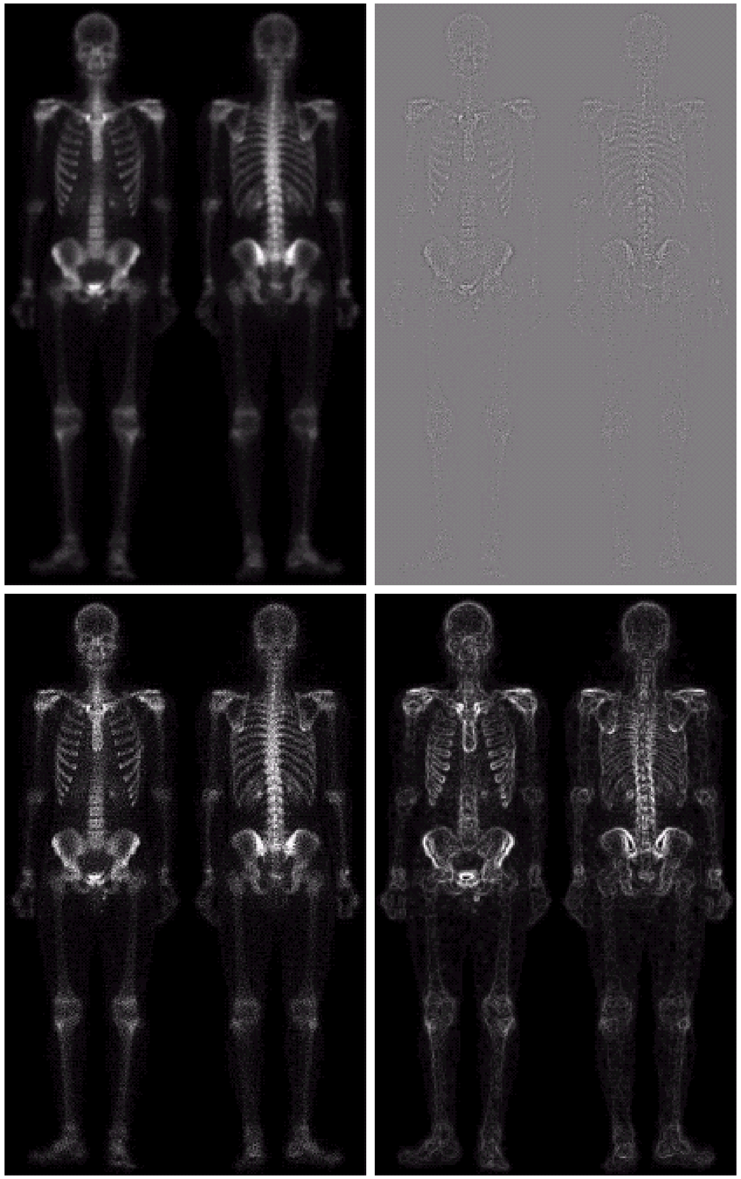
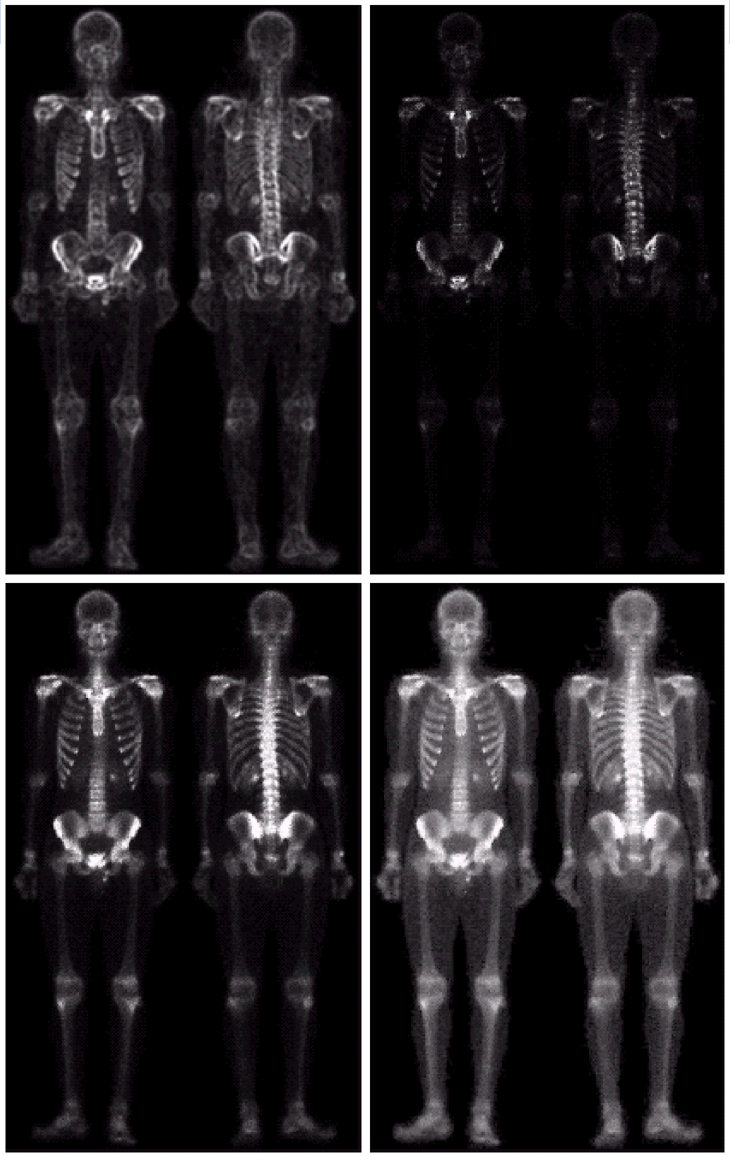


FIGURE 3.46 (a) Image of whole body bone scan. (b) Laplacian of (a). (c) Sharpened image obtained by adding (a) and (b). (d) Sobel of (a).

Combining Spatial Enhancement Methods

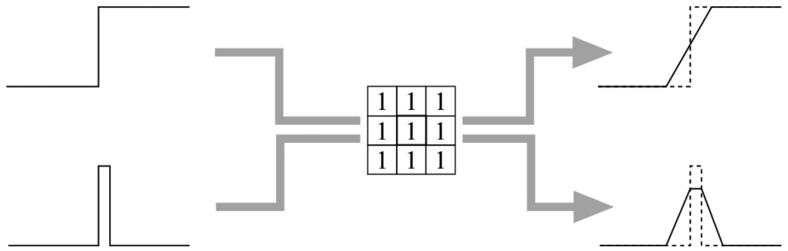


e	f
g	h

FIGURE 3.46 *(Continued)* (e) Sobel image smoothed with a 5 × 5 averaging filter. (f) Mask image formed by the product of (c) and (e). (g) Sharpened image obtained by the sum of (a) and (f). (h) Final result obtained by applying a power-law transformation to (g). Compare (g) and (h) with (a). (Original image courtesy of G.E. Medical Systems.)

Nonlinear Spatial Filters

- One important use of linear filter is noise reduction; however, linear (smoothing) filters have a significant drawback – along with the noise, it also smoothes out important structures, edges, etc. in the image:



- We now want to examine if some nonlinear filters can offer a better solution here

What is a nonlinear filter?

- It's a filter– resulting pixel value is a function of the pixel values in a corresponding neighborhood, i.e., sub-window around the pixel location
- It does not maintain linearity property

Median filter

- Median filter replaces every image pixel by the median of the pixels in the corresponding image sub-window *R*:

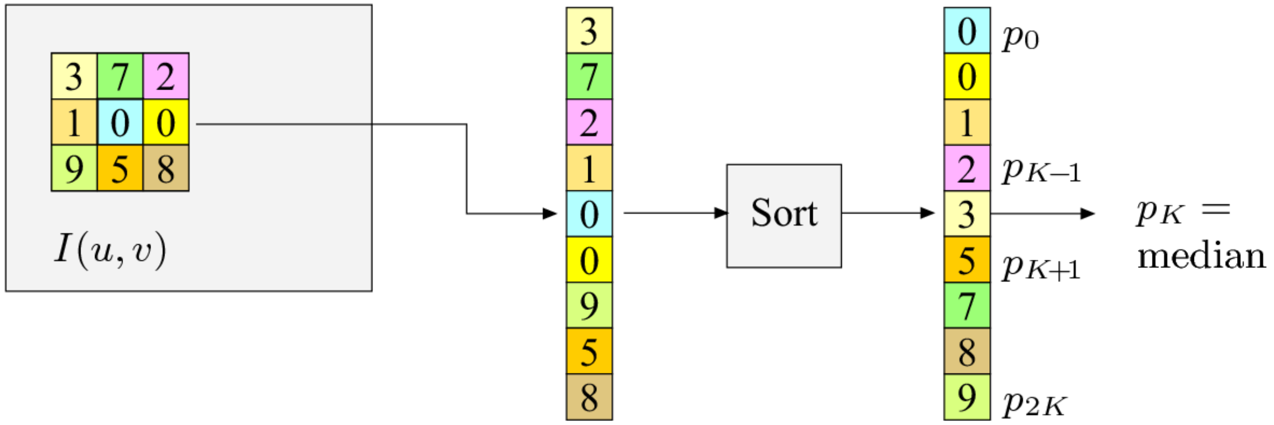
$$I'(u,v) \leftarrow \text{median}\,\{I(u+i, v+j) \mid (i,j) \in R\}$$

- The median of 2*K*+1 pixel values p_i is defined as:

$$\text{median}\,(p_0, p_1, \ldots, p_K, \ldots, p_{2K}) \triangleq p_K$$

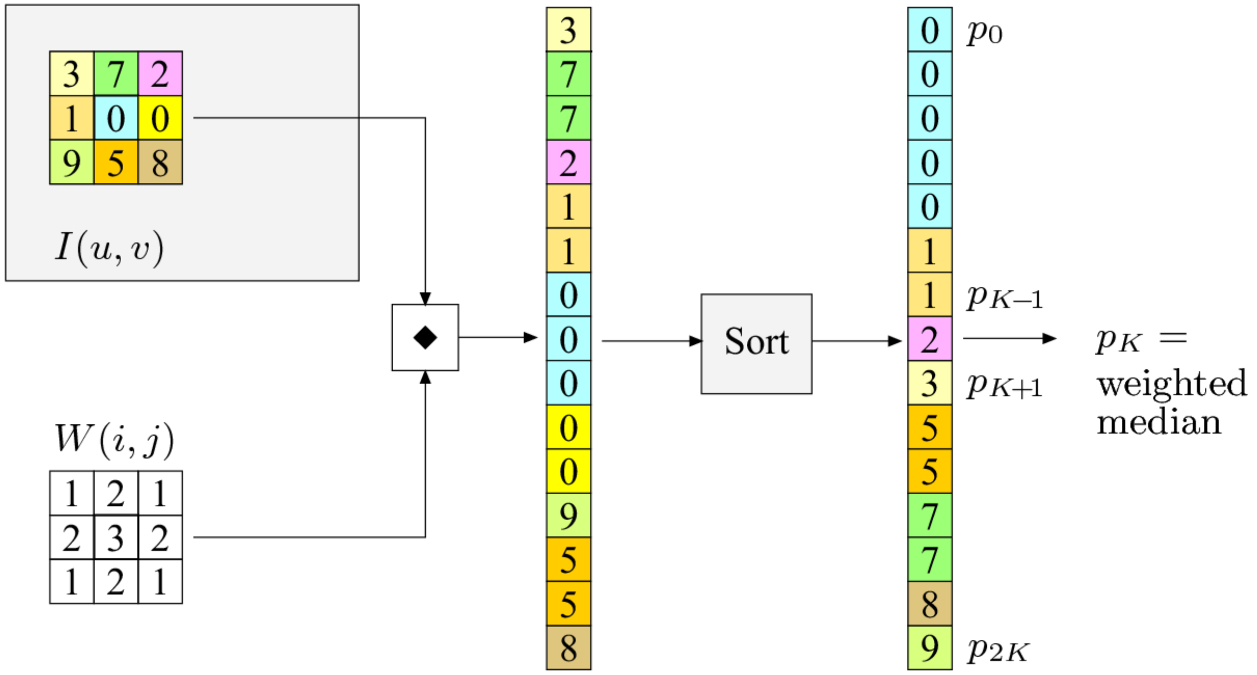
where p_i are sorted, *i*.e., $p_0 \le p_1 \le \ldots \le p_{2K}$

Median filter: pictorially



Computation of a 3x3 median filter. The 9 pixel values extract
from the 3x3 image sub-window are arranged a sorted vector.
The resulting center value is taken.

Weighted median filter

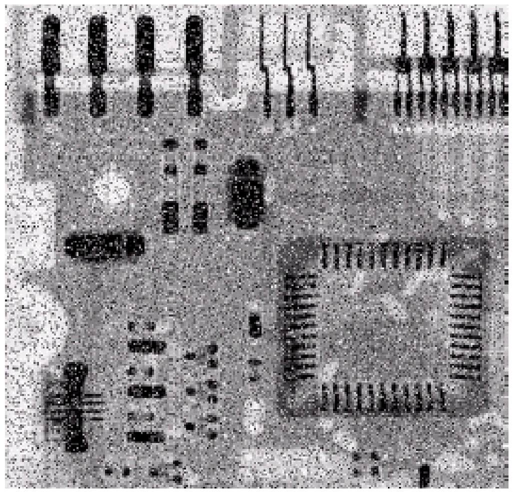


Each pixel value is inserted into the extended pixel vector multi
as specified by the weight matrix.

Mimics voting

Averaging Filter Vs. Median Filter Example

Images taken from Gonzalez & Woods, Digital Image Processing (2002)


Original Image With Noise

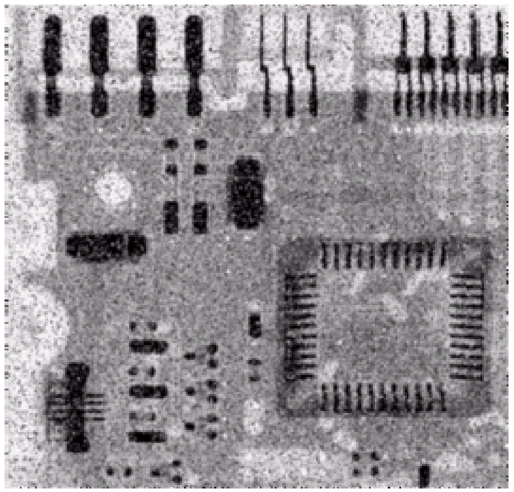


Image After Averaging Filter

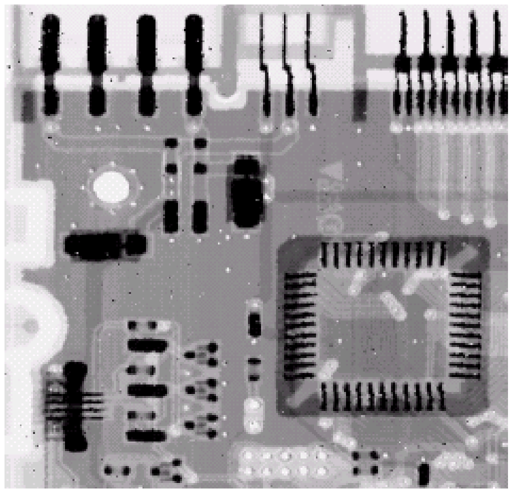
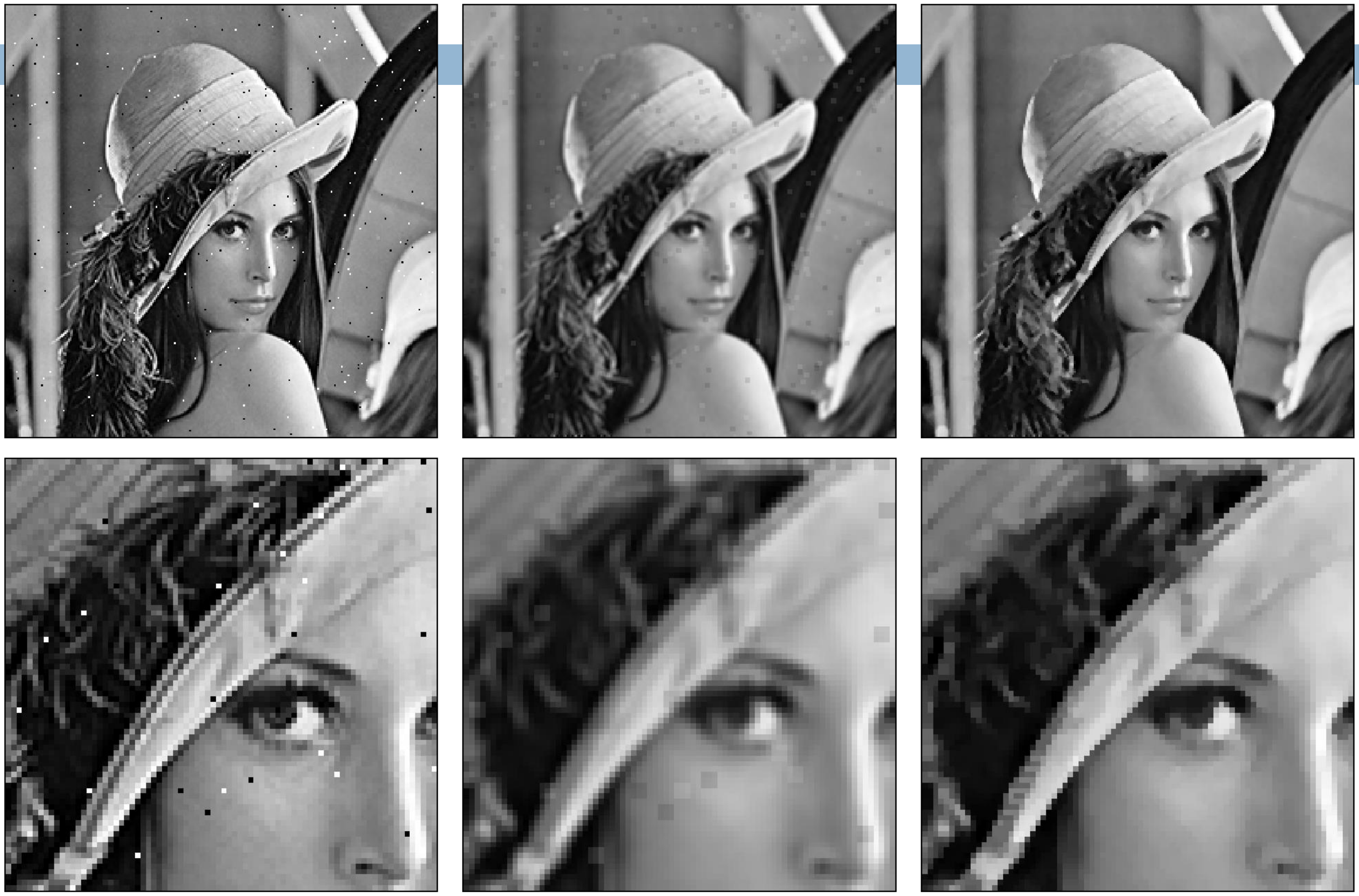


Image After Median Filter

- Filtering is often used to remove noise from images
- Often a median filter works better than an averaging filter

Some illustrations



(a) Lena image with ‘salt and pepper’ noise; (b) linear 3x3 box filter; (c) median filter result