Lecture 7:

Image Enhancement in Spatial Domain

Basic image filtering/masking operations

Spatial filtering for Smoothing

- For blurring/noise reduction;
- Blurring is usually used in preprocessing steps,

  e.g., to remove small details from an image prior to object extraction,

  or to bridge small gaps in lines or curves
- Equivalent to Low-pass spatial filtering in frequency domain because smaller (high frequency) details are removed based on neighborhood averaging (averaging filters)

Implementation: The simplest form of the spatial filter for averaging is a square mask (assume $m \times m$ mask) with the same coefficients $1/m^2$ to preserve the gray levels (averaging).

Applications: Reduce noise; smooth false contours

Side effect: Edge blurring

Smoothing filters

Consider the output pixel is positioned at the center

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Box filter all coefficients are equal

Weighted average give more (less) weight to pixels near (away from) the output location

Consider mask size:

$$m \times n$$

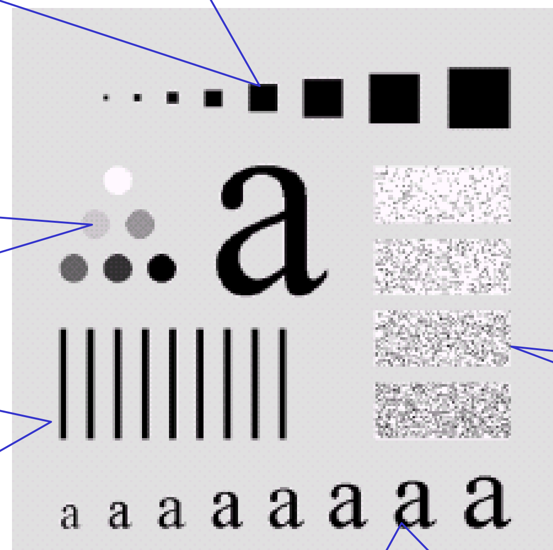
$$w_i = \frac{1}{mn}, i = 1, \ldots, mn$$

Spatial filtering for Smoothing (example)

Squares of sizes: 3, 5, 9, 15, 25, 35, 45, and 55 pixels; borders are 25 pixels apart

Circles: Diameter 25 pixels, separation 15 pixels

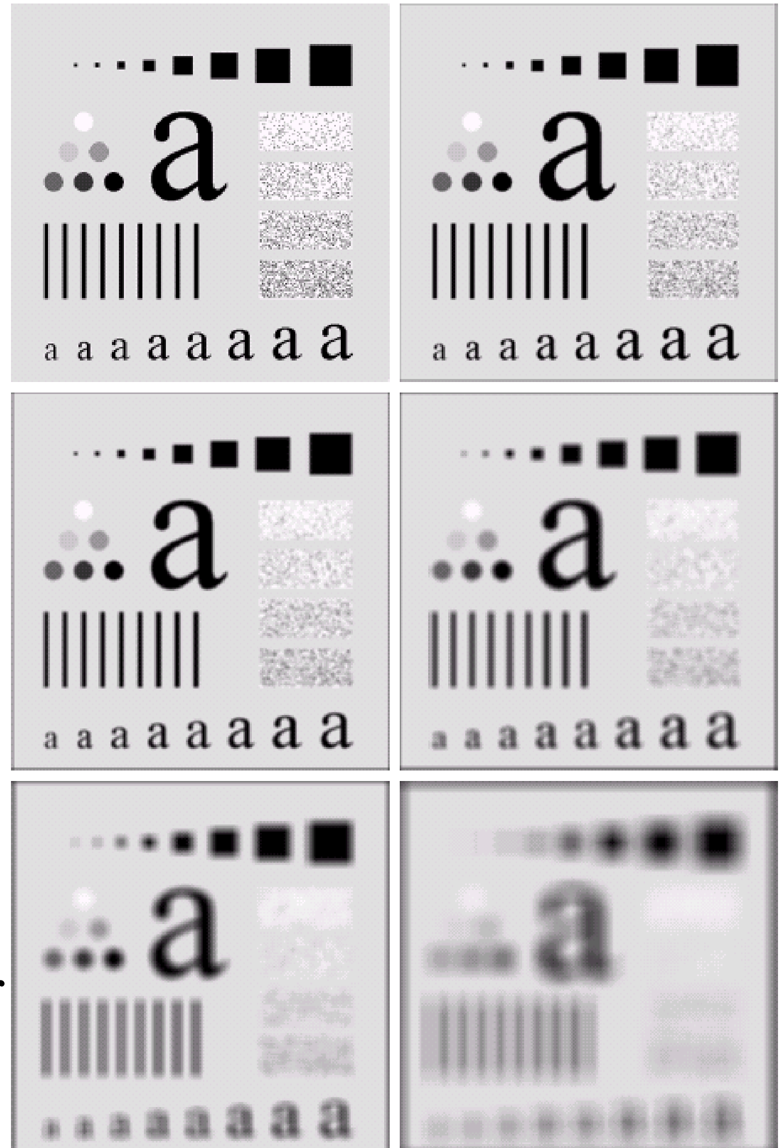
Vertical bars: 5 pixels wide, 100 pixels high, separation 20 pixels



Noisy rectangles: 50x120 pixels

Letter sizes: range from 10 to 24 points, in increments of 2 points

Spatial filtering for Smoothing (example)



Original image
size: 500 x 500

Smoothed by
3 x 3 box filter

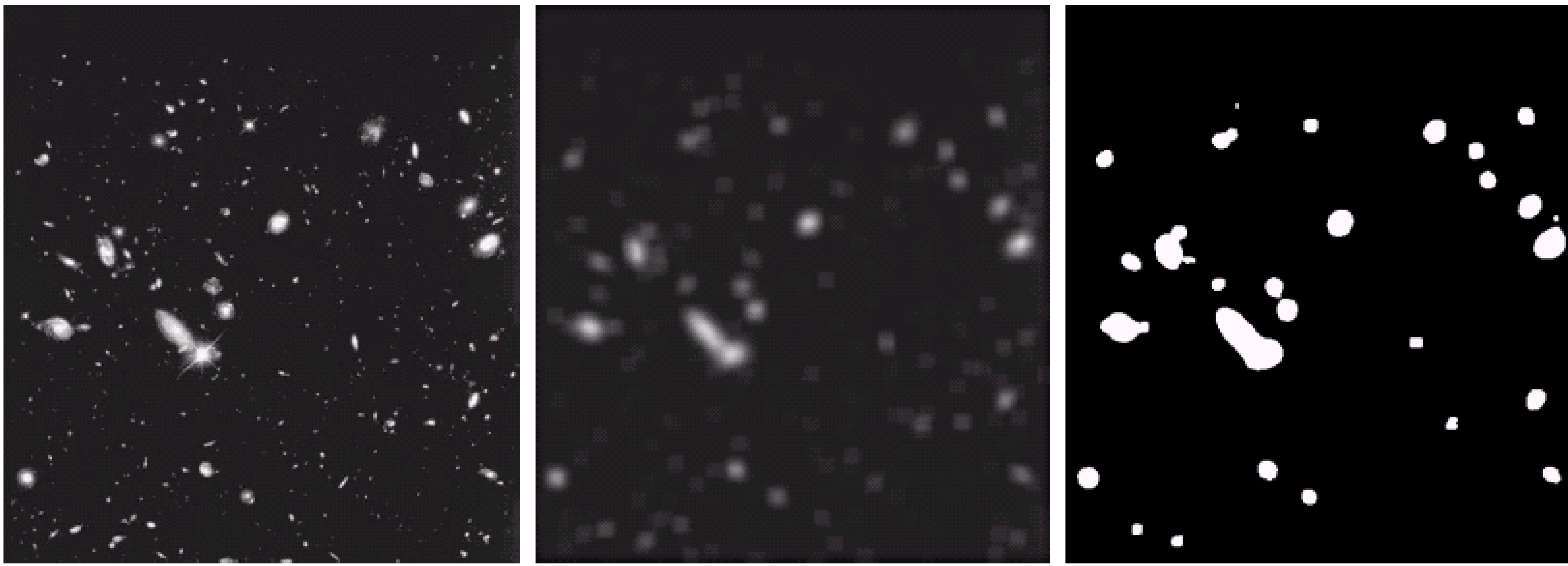
Smoothed by
5 x 5 box filter

Smoothed by
9 x 9 box filter

Smoothed by
15 x 15 box filter

Smoothed by
35 x 35 box filter

Spatial filtering for Smoothing (example)



a b c

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.)

Order-statistics filtering

- Nonlinear spatial filters
- Output is based on order of gray levels in the masked area (sub-image)
- Examples: Median filtering, Max & Min filtering

Median filtering

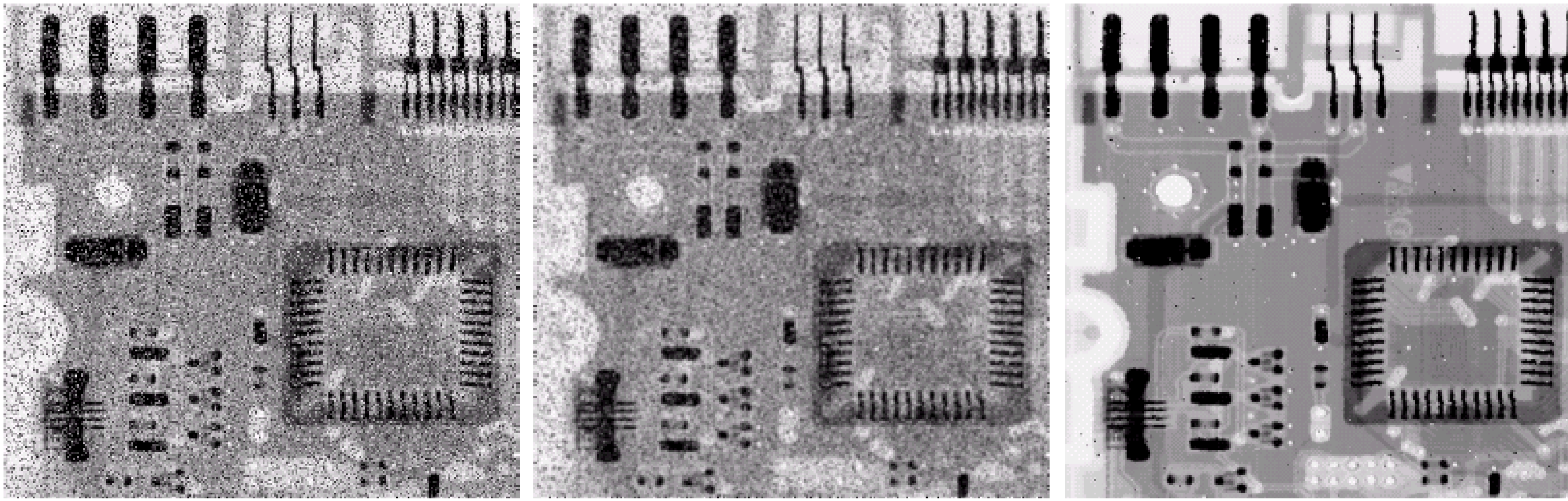
- Assigns the mid value of all the gray levels in the mask to the center of mask;
- Particularly effective when
  - *the noise pattern consists of strong, spiky components (impulse noise, salt-and-pepper*
  - *edges are to be preserved*
  - *Force points with distinct gray levels to be more like their neighbors*

Median Filtering

10	20	20
20	15	20
20	25	100

Output = ? **20**

Median Filtering (example)



a b c

FIGURE 3.37 (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.)

Spatial filtering for image sharpening

Background: to highlight fine detail in an image or to enhance blurred detail

Applications: electronic printing, medical imaging, industrial inspection, autonomous target detection (smart weapons)......

Foundation:

- Blurring/smoothing is performed by spatial averaging (equivalent to integration)
- Sharpening is performed by noting only the gray level changes in the image that is the differentiation

Spatial filtering for image sharpening

Operation of Image Differentiation

- Enhance edges and discontinuities (magnitude of output gray level >>0)
- De-emphasize areas with slowly varying gray-level values (output gray level: 0)

Mathematical Basis of Filtering for Image Sharpening

- First-order and second-order derivatives
- Approximation in discrete-space domain
- Implementation by mask filtering

First and second order derivatives

$$\boxed{\frac{\partial f}{\partial x} = f(x+1) - f(x)}$$

$f(x)$	$f(x+1)$

Position for the output pixel

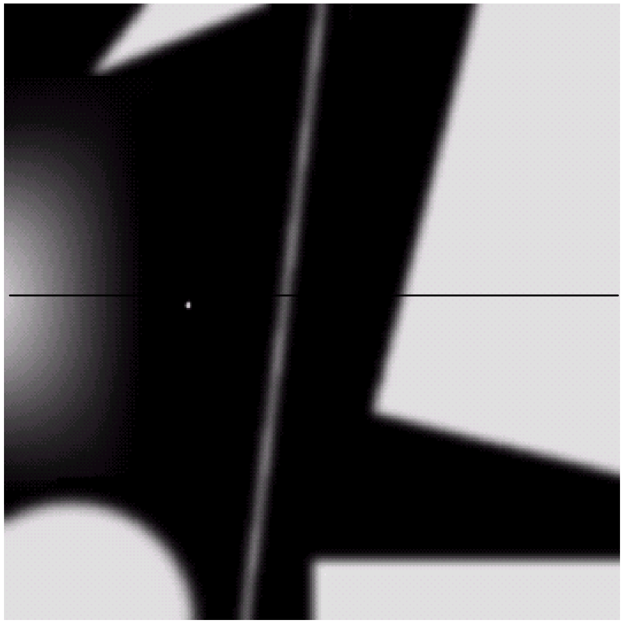
$$\frac{\partial^2 f}{\partial x^2} = f'(x+1) - f'(x)$$
$$= [f(x+2) - f(x+1)] - [f(x+1) - f(x)]$$
$$= f(x+2) - 2f(x+1) + f(x)$$

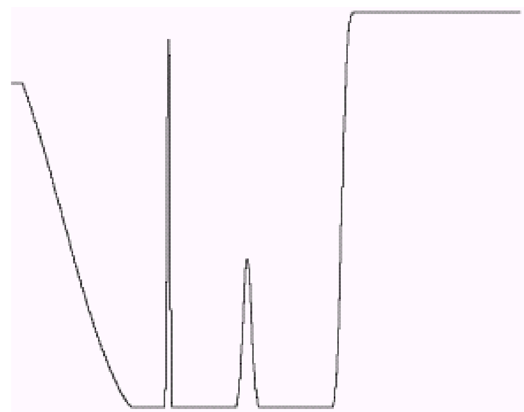
$f(x)$	$f(x+1)$	$f(x+2)$

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

$f(x-1)$	$f(x)$	$f(x+1)$

Example for discrete derivatives





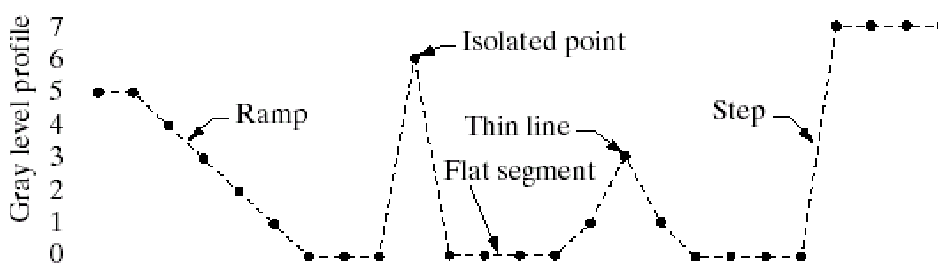


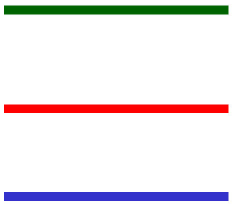
Image strip	5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	7	•	•
First Derivative		−1	−1	−1	−1	−1	0	0	6	−6	0	0	0	1	2	−2	−1	0	0	0	7	0	0	0			
Second Derivative		−1	0	0	0	0	1	0	6	−12	6	0	0	1	1	−4	1	1	0	0	7	−7	0	0			

Various situations encountered for derivatives

$$f' = \frac{\partial f}{\partial x} \qquad f'' = \frac{\partial^2 f}{\partial x^2}$$

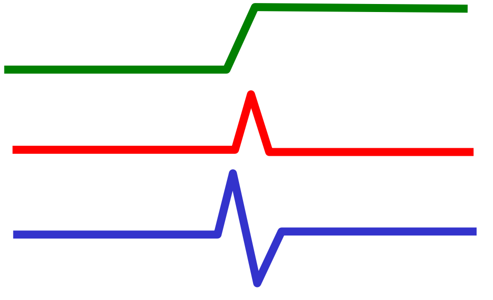
- Flat segment→ $(f')=0$; $(f'')=0$

f	0	0	0	0	0
f′		0	0	0	0
f″		0	0	0	



- Step→ (f'):{0,+,0}; (f''):{0,+,–,0}

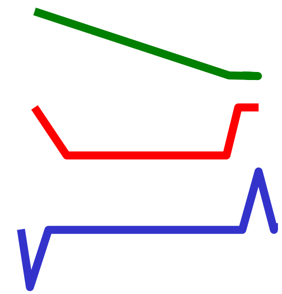
f	0	0	0	7	7	7	7	
f′		0	0	7	0	0	0	0
f″		0	7	–7	0	0	0	



Various situations encountered for derivatives

- Ramp→ $(f')\approx$constant; $(f'')=0$

f	5	4	3	2	1	0	0	
f'	0	−1	−1	−1	−1	−1	0	0
f''	−1	0	0	0	0	1	0	

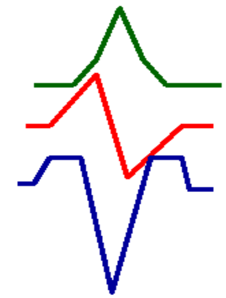


- *Ramps or steps* in the 1D profile normally characterize the edges in an image
- f'' is nonzero at the onset and end of the ramp: produce thin (double) edges
- f' is nonzero along the entire ramp produce thick edges

Various situations encountered for derivatives

- Thin lines

f	0	0	1	3	1	0	0	
f'	0	0	1	2	–2	–1	0	0
f''	0	1	1	–4	1	1	0	



- Isolated point

f	0	0	0	6	0	0	0	
f'		0	0	6	–6	0	0	0
f''		0	6	–12	6	0	0	

f'' responses much stronger than f' around the point

f'' enhances fine detail (including noise) much more than f'

Comparison between f'' and f'

- f' generally produce thicker edges in an image
- f'' have a stronger response to file detail
- f' generally have a stronger response to a gray-level step
- f'' produces a double response at step changes in gray level
- f'' responses given similar changes in gray-level values line > point > step
- For image enhancement, f'' is generally better suited than f'
- Major application of f' is for edge extraction;
  f' used together with f'' results in impressive enhancement effect

Laplacian for image enhancement

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

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

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

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

y

$f(x-1, y-1)$	$f(x-1, y)$	$f(x-1, y+1)$
$f(x, y-1)$	$f(x, y)$	$f(x, y+1)$
$f(x+1, y-1)$	$f(x+1, y)$	$f(x+1, y+1)$

x

0	1	0
1	–4	1
0	1	0

Laplacian for image enhancement

0	1	0
1	–4	1
0	1	0

Isotropic for rotations in increments of 90°

+

1	0	1
0	–4	0
1	0	1

$\Rightarrow$

1	1	1
1	–8	1
1	1	1

Isotropic for rotations in increments of 45°

Laplacian for image enhancement

(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.

Laplacian for image enhancement

To obtain the enhanced image

$$g(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y), w_5 < 0 \\ f(x,y) + \nabla^2 f(x,y), w_5 > 0 \end{cases}$$

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

In this way, background tonality can be perfectly preserved

while details are enhanced

–1	–1	–1
–1	8	–1
–1	–1	–1

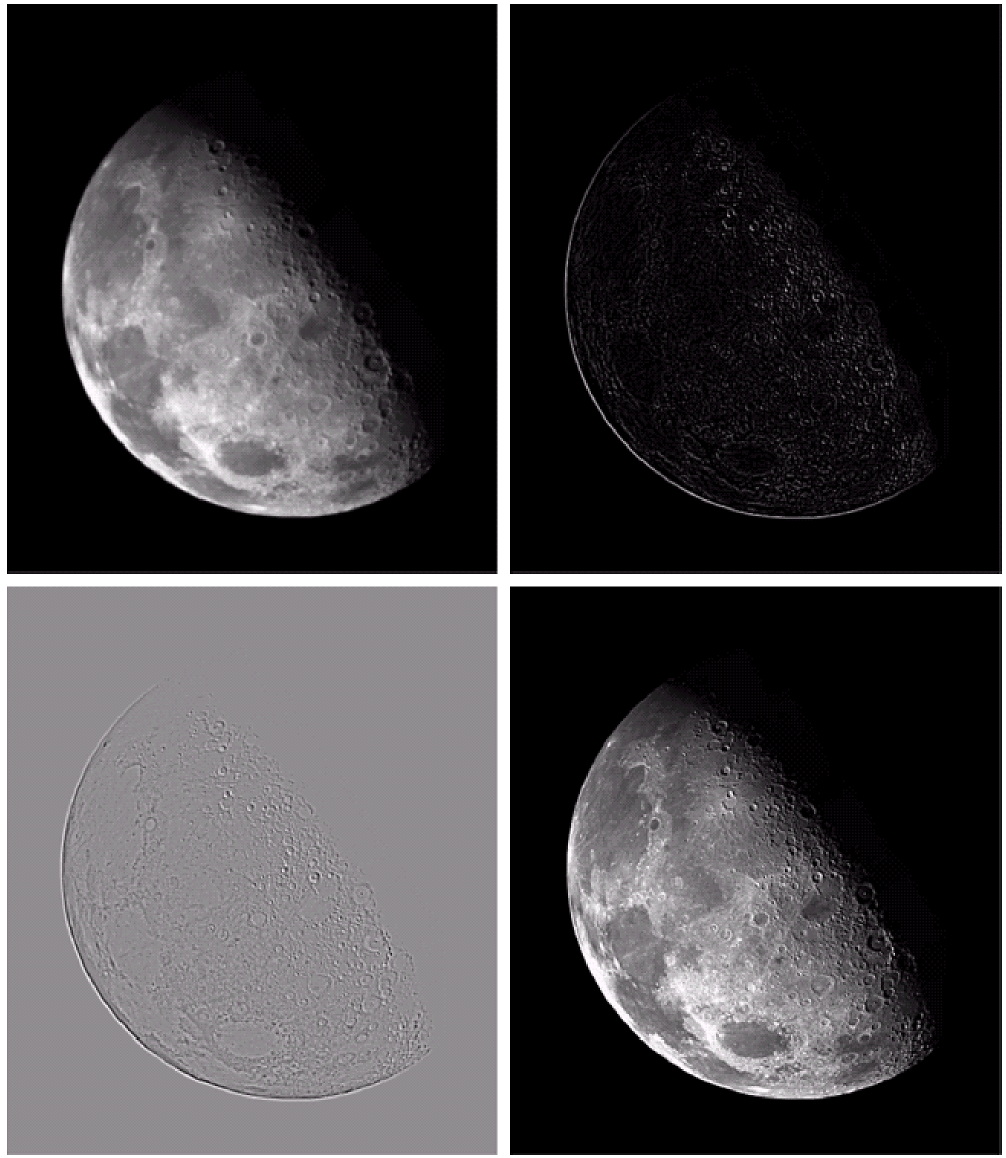
+

0	0	0
0	1	0
0	0	0

→

–1	–1	–1
–1	9	–1
–1	–1	–1

Laplacian for image enhancement (example)



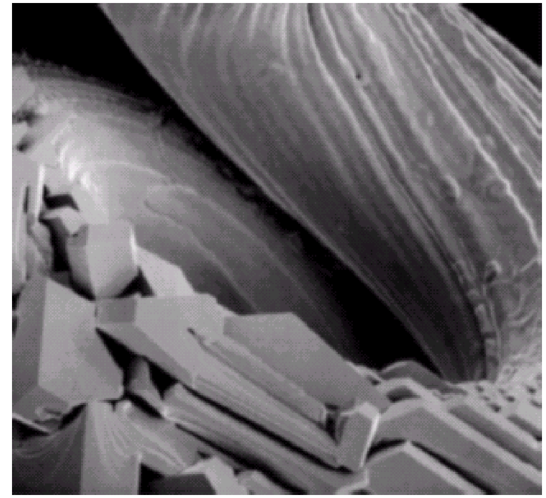
a	b
c	d

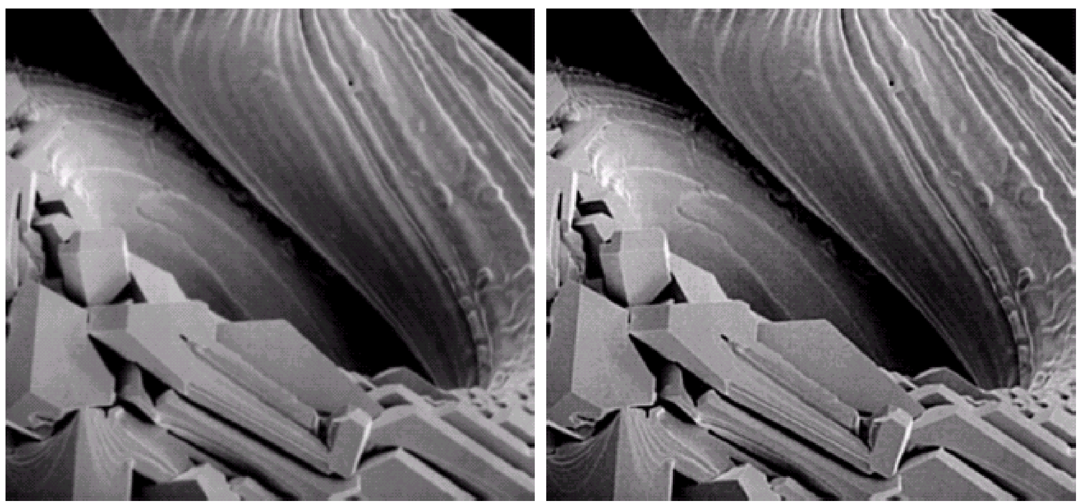
FIGURE 3.40 (a) Image of the North Pole of the moon. (b) Laplacian-filtered image. (c) Laplacian image scaled for display purposes. (d) Image enhanced by using Eq. (3.7-5). (Original image courtesy of NASA.)

Laplacian for image enhancement (example)

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.)

Image sharpening based on unsharp masking

Subtract the blurred image from the original image

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

$f_s(x,y) \Rightarrow$ sharpened image

$\bar{f}(x,y) \Rightarrow$ blurred image of $f(x,y)$

Generalization of *unsharp masking* ☞ high-boost filtering

$$f_{hb}(x,y) = Af(x,y) - \bar{f}(x,y) \qquad A \geq 1$$

High boost filtering

$$f_{hb}(x,y) = Af(x,y) - \bar{f}(x,y),\ A > 1$$

$$= (A-1)f(x,y) + f(x,y) - \bar{f}(x,y)$$

$$= (A-1)f(x,y) + f_s(x,y)$$

$$\xrightarrow[\text{to } f_s(x,y)]{\text{apply Laplacian mask}} = (A-1)f(x,y) + \left[f(x,y) \pm \nabla^2 f(x.y)\right]$$

$$= Af(x,y) \pm \nabla^2 f(x,y)$$

Principal application:

Boost filtering is used when input image is darker than desired, high-boost filter makes the image lighter and more natural

High boost filtering masks

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & -1 & 0 \\ -1 & A+4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} -1 & -1 & -1 \\ -1 & \mathrm{A}+8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$A \geq 1$

Gradient Operators

2D first derivative (∇f) in image processing are implemented using the magnitude of the gradient

The gradient is generally given by

$$\nabla \mathrm{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

The magnitude is given by $\nabla f = \qquad \nabla$

$$= \left[G_x^2 + G_y^2\right]^{1/2} = \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2}$$

Magnitude is approximated as $\nabla f \approx \left|G_x\right| + \left|G_y\right|.$

Gradient Operators

- For a sub-image given in the figure:

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

The gradient is approximated by

$$\nabla f \approx |z_9 - z_5| + |z_8 - z_6|$$

This equation can be represented in following masks (Robert cross gradient operators)

−1	0
0	1

0	−1
1	0

Gradient Operators

Normally the smallest mask used is of size 3 x 3

Based on the concept of approximating the gradient several spatial masks have been proposed:

Prewitt operator

−1	−1	−1
0	0	0
1	1	1

Extract horizontal edges

−1	0	1
−1	0	1
−1	0	1

Extract vertical edges

Pixel arrangement

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Prewitt operators (equation):

$$\nabla f \approx \left|(z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)\right| + \left|(z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)\right|$$

Gradient Operators

Sobel operators

$$\begin{array}{c} \xrightarrow{\quad y \quad} \\ x \downarrow \begin{array}{|c|c|c|} \hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array} \end{array} \qquad \begin{array}{c} \xrightarrow{\quad y \quad} \\ x \downarrow \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array} \end{array}$$

Extract horizontal edges

Extract vertical edges

Sobel operators (equation):

$$\nabla f \approx \left|\left(z_7 + 2z_8 + z_9\right) - \left(z_1 + 2z_2 + z_3\right)\right| + \left|\left(z_3 + 2z_6 + z_9\right) - \left(z_1 + 2z_4 + z_7\right)\right|$$

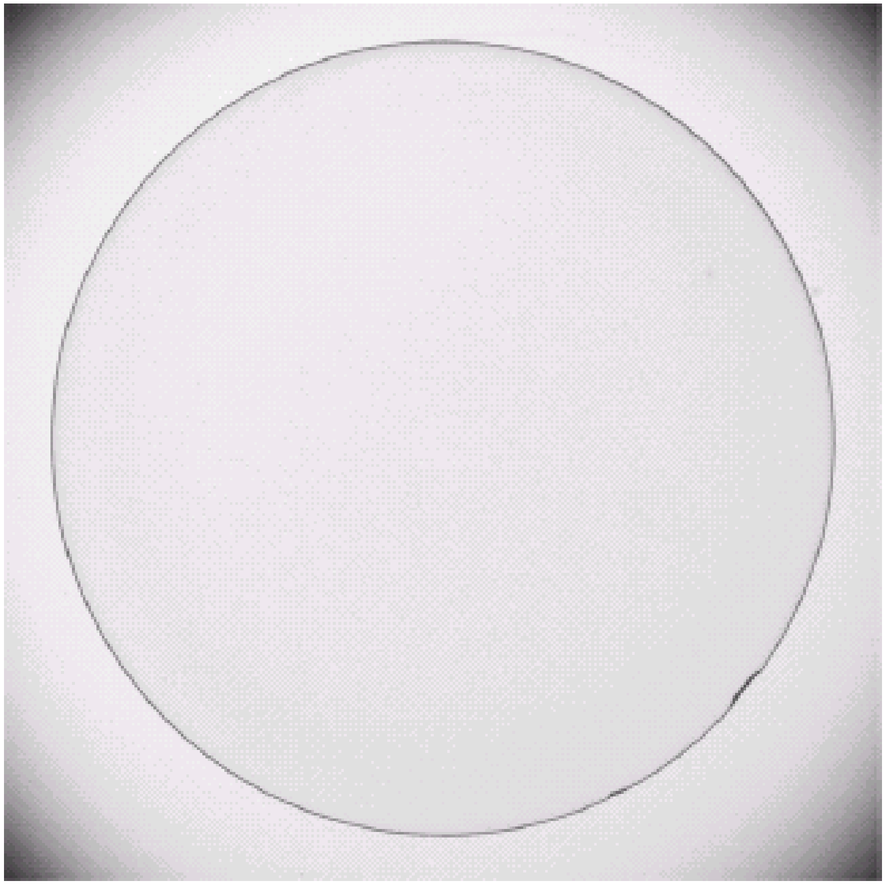
Emphasize more the current position (y)

Emphasize more the current position (x)

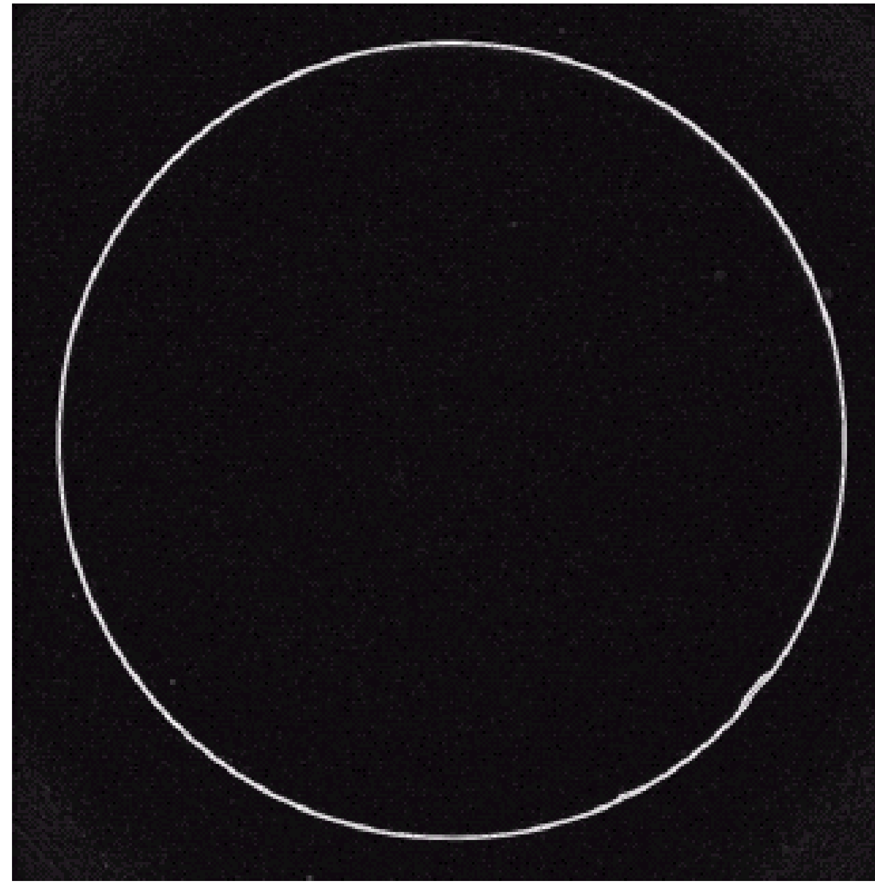
Pixel arrangement

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Gradient Processing (example)



Optical image of contact lens
(note defects at 4 and 5 o' clock



Sobel gradient