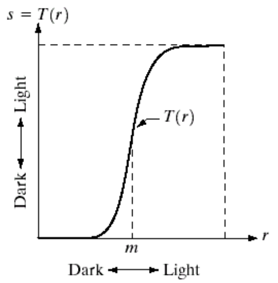
Digital Image Processing
Chapter 3:
Image Enhancement in the Spatial Domain

Types of Image Enhancement in the Spatial Domain

- Single pixel methods
  - Gray level transformations
    Example
    - Historgram equalization
    - Contrast stretching
  - Arithmetic/logic operations
    Examples
    - Image subtraction
    - Image averaging
- Multiple pixel methods
  Examples
  Spatial filtering
  - Smoothing filters
  - Sharpening filters

Gray Level Transformation

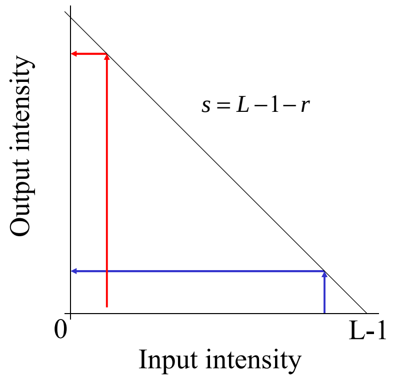
$$s = T(r)$$

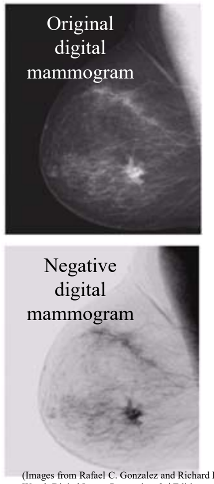


Contrast enhancement

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Image Negative



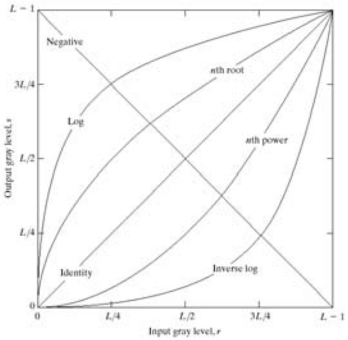


(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Log Transformations

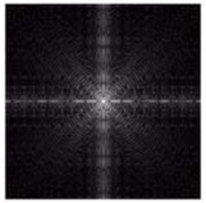
Application

$$s = c\log(r+1)$$



Fourier spectrum

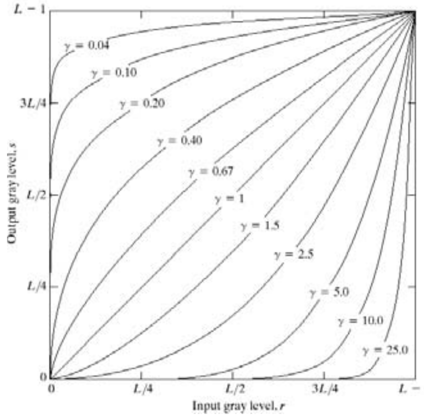
Log Tr. of Fourier spectrum



(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

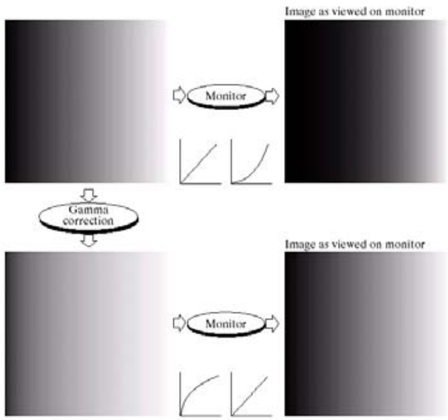
Power-Law Transformations

$$s = cr^{\gamma}$$



(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Power-Law Transformations : Gamma Correction Application



Gamma correction

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Power-Law Transformations : Gamma Correction Application

a b
c d

FIGURE 3.8 (a) Magnetic resonance (MR) image of a fractured human spine. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 0.6, 0.4,$ and 0.3, respectively. (Original image for this example courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

Gamma Correction

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Power-Law Transformations : Gamma Correction Application

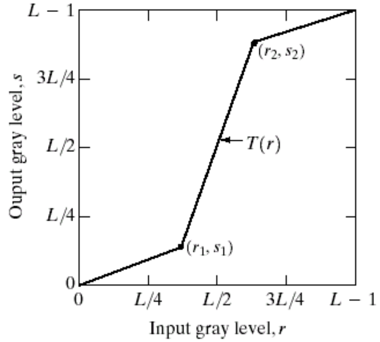
FIGURE 3.9 (a) Aerial image. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0, 4.0,$ and 5.0, respectively. (Original image for this example courtesy of NASA.)

Gamma Correction

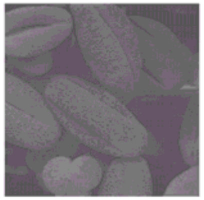
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Contrast Stretching

Contrast



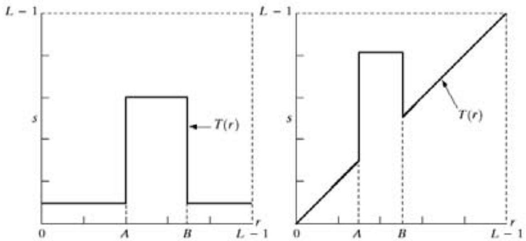
Before contrast enhancement

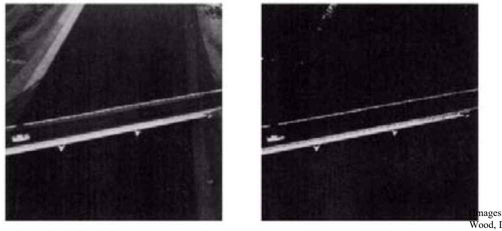


After

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

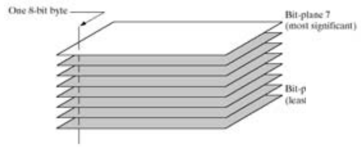
Gray Level Slicing

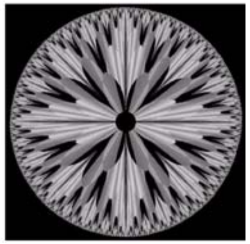


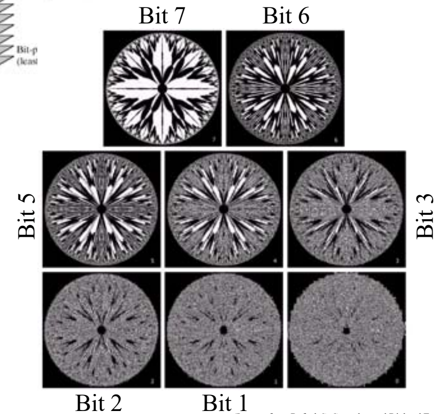


(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Bit-plane Slicing





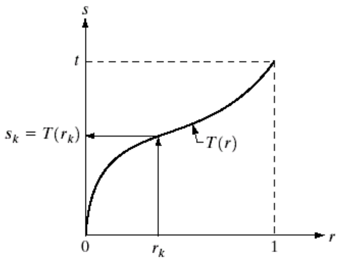


(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Monotonically Increasing Function

Function T

$$s = T(r)$$



Histogram processing function

1. Monotonically increasing function

2. $0 \le T(r) \le 1 \quad \text{for} \quad 0 \le r \le 1$

Transformation of Random Variables

Random variables $p_s(s)$, $p_r(r)$ in a transformation $s = T(r)$

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$$

Probability Density Function

- In probability theory, a **probability density function (pdf)**, is a function that describes the relative likelihood for this random variable to take on a given value.
- The probability of the random variable falling within a particular range of values is given by the integral of this variable's density over that range

A random variable X has density f_X if

$$\Pr[a \le X \le b] = \int_a^b f_X(x)\,dx.$$

The cumulative distribution of X :

$$F_X(x) = \int_{-\infty}^{x} f_X(u)\,du,$$

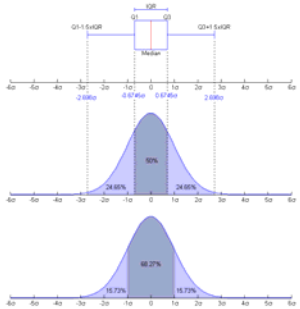
And if f_X is continuous at X

$$f_X(x) = \frac{d}{dx}F_X(x).$$

Probability Density Function

Any function f that describes the probability density in terms of the input variable x is a probability density function if and only if it is non-negative and the area under the graph is 1:

$$f(x) \ge 0\,\forall x \wedge \int_{-\infty}^{\infty} f(x)\,dx = 1$$



a probability density function (pdf) of a Normal N(0,1,σ^2) Population

Histogram Equalization

$$s = T(r) = \int_0^r p_r(w)dw$$

a *uniform* probability density function.

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| = p_r(r)\cdot\left|\frac{1}{\frac{ds}{dr}}\right|$$

$$= p_r(r)\cdot\left|\frac{1}{\frac{d\left(\int_0^r p_r(w)dw\right)}{dr}}\right| = p_r(r)\cdot\left|\frac{1}{p_r(r)}\right| = 1$$

Probability density function of uniform distribution

a density function that is constant, making it the simplest kind of density function.

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{for } a \le x \le b, \\ 0 & \text{for } x < a \text{ or } x > b, \end{cases}$$

The equation for the standard uniform distribution is

$$f(x) = 1 \quad \text{for } 0 \le x \le 1$$

Histogram Equalization

Continuous PDF

Histogram Digital Image

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j)$$

$$= \sum_{j=0}^{k} \frac{n_j}{N}$$

n_j = the number of pixels with intensity = j

N = the number of total pixels

Histogram Equalization Example

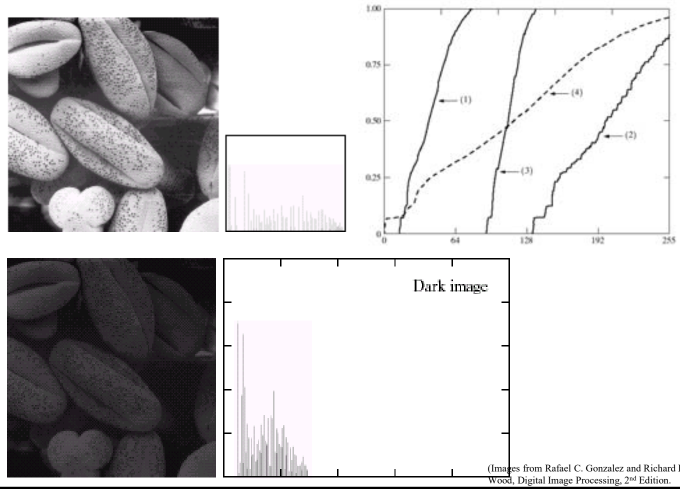
Intensity	# pixels
0	20
1	5
2	25
3	10
4	15
5	5
6	10
7	10
Total	100

Accumulative Sum of P_r
20/100 = 0.2
(20+5)/100 = 0.25
(20+5+25)/100 = 0.5
(20+5+25+10)/100 = 0.6
(20+5+25+10+15)/100 = 0.75
(20+5+25+10+15+5)/100 = 0.8
(20+5+25+10+15+5+10)/100 = 0.9
(20+5+25+10+15+5+10+10)/100 = 1.0
1.0

Histogram Equalization Example (cont.)

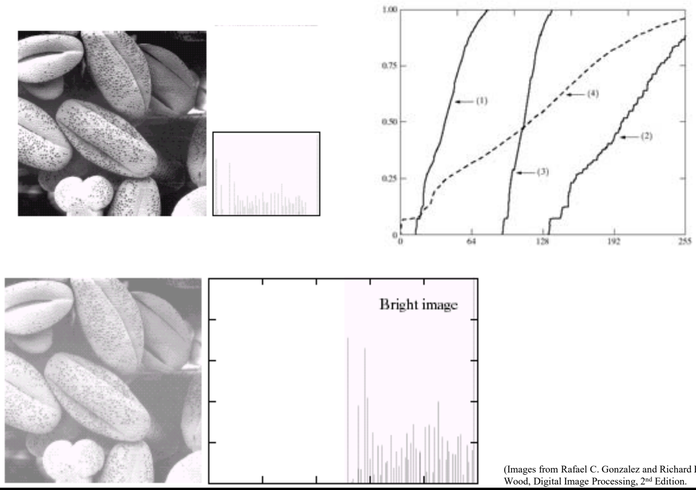
Intensity (r)	No. of Pixels (n_j)	Acc Sum of P_r	Output value	Quantized Output (s)
0	20	0.2	0.2x7 = 1.4	1
1	5	0.25	0.25*7 = 1.75	2
2	25	0.5	0.5*7 = 3.5	3
3	10	0.6	0.6*7 = 4.2	4
4	15	0.75	0.75*7 = 5.25	5
5	5	0.8	0.8*7 = 5.6	6
6	10	0.9	0.9*7 = 6.3	6
7	10	1.0	1.0x7 = 7	7
Total	100			

Histogram Equalization



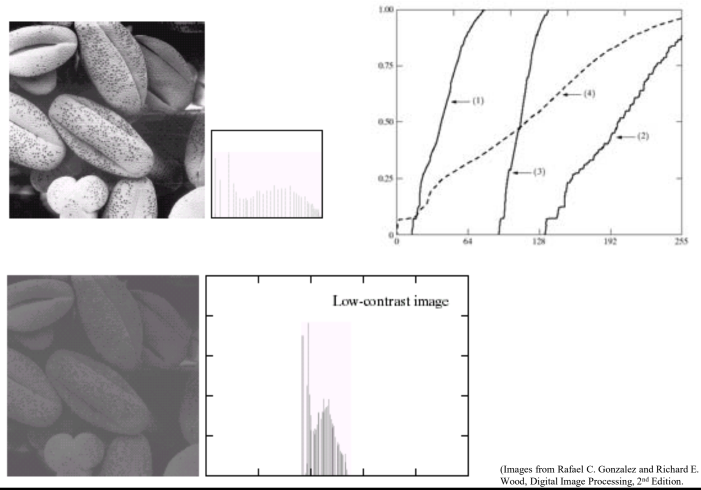
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Histogram Equalization (cont.)



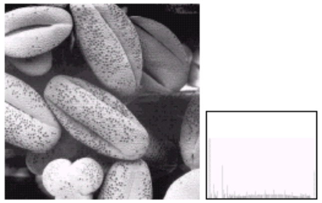
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

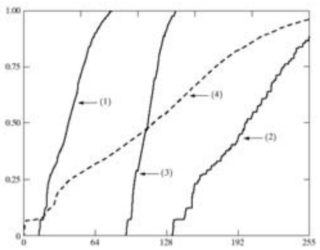
Histogram Equalization (cont.)

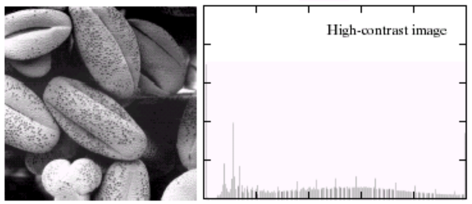


(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Histogram Equalization (cont.)

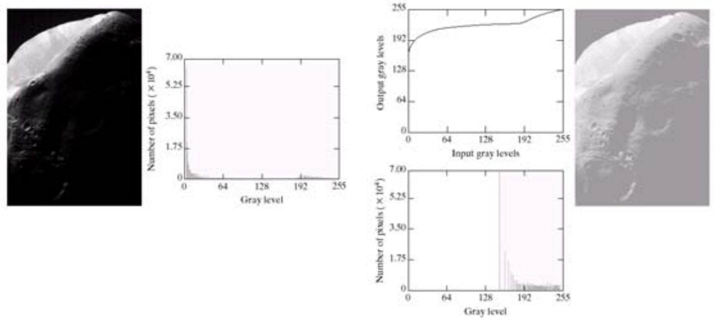






(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Histogram Equalization (cont.)



Histogram Eq.

Histogram equalization Low Contrast

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Histogram Matching : Algorithm

Histogram equalization

$$s = T(r) = \int_0^r p_r(w)dw$$

$p_s(s) = 1$

Histogram

User output image PDF $p_z(z)$

$p_z(z)$

$$v = G(z) = \int_0^z p_z(u)du$$

$p_v(v) = 1$

$p_s(s) = p_v(v) = 1$, s=v

r → T( ) → s → G^{-1}( ) → z

Histogram Matching : definition

There are applications in which attempting to base enhancement on a uniform histogram is not the best approach.

In particular, it is useful sometimes to be able to specify the shape of the histogram that we wish the processed image to have.

The method used to generate a processed image that has a specified histogram is called *histogram matching* or *histogram specification*.

Histogram Matching : Algorithm

Histogram

Histogram equalization

$$s = T(r) = \int_0^r p_r(w)dw$$

$p_s(s) = 1$

User output image PDF $p_z(z)$

$p_z(z)$

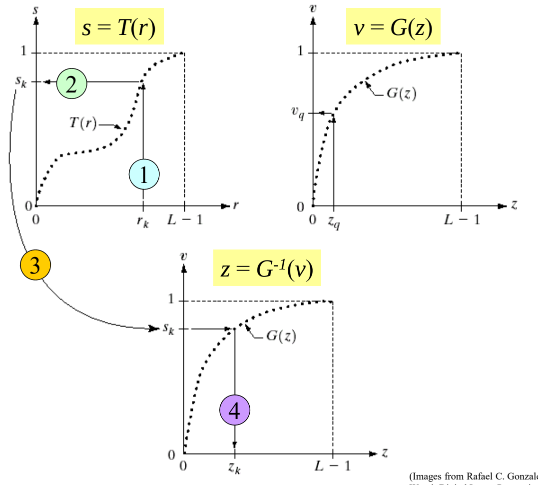
$$v = G(z) = \int_0^z p_z(u)du$$

$p_v(v) = 1$

$p_s(s) = p_v(v) = 1$, s=v

r → T( ) → s → G^{-1}( ) → z

Histogram Matching : Algorithm (cont.)



(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Histogram Matching Example

Histogram
input image

Intensity (s)	# pixels
0	20
1	5
2	25
3	10
4	15
5	5
6	10
7	10
Total	100

Histogram
output image

Intensity (z)	# pixels
0	5
1	10
2	15
3	20
4	20
5	15
6	10
7	5
Total	100

User

Histogram Matching Example (cont.)

1. Histogram Equalization

r	(n_j)	ΣP_r	s
0	20	0.2	1
1	5	0.25	2
2	25	0.5	3
3	10	0.6	4
4	15	0.75	5
5	5	0.8	6
6	10	0.9	6
7	10	1.0	7

$s_k = T(r_k)$

z	(n_j)	ΣP_z	v
0	5	0.05	0
1	10	0.15	1
2	15	0.3	2
3	20	0.5	4
4	20	0.7	5
5	15	0.85	6
6	10	0.95	7
7	5	1.0	7

$v_k = G(z_k)$

Histogram Matching Example (cont.)

2. Map

r → s

s → v

v → z

r	s
0	1
1	2
2	3
3	4
4	5
5	6
6	6
7	7

v	z
0	0
1	1
2	2
4	3
5	4
6	5
7	6
7	7

$s_k = T(r_k)$

$z_k = G^{-1}(v_k)$

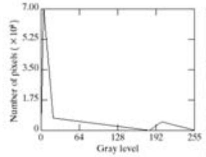
r	z
0	1
1	2
2	2
3	3
4	4
5	5
6	5
7	6

Actual Output Histogram

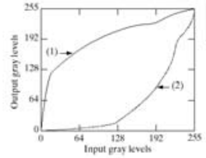
z	# Pixels
0	0
1	20
2	30
3	10
4	15
5	15
6	10
7	0

Histogram Matching Example (cont.)

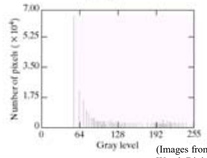
Desired histogram



Transfer function

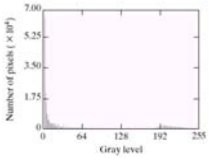


Actual histogram

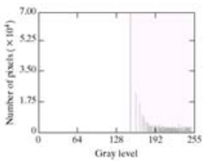


(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

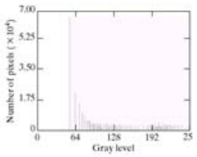
Histogram Matching Example (cont.)



Original image



After histogram equalization

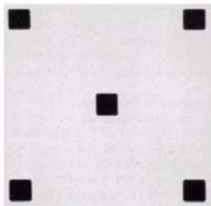


After histogram matching

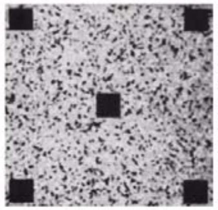
Local Enhancement : Local Histogram Equalization

Concept: Perform histogram equalization in a small neighborhood

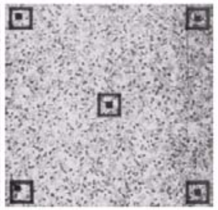
Orignal image



After Hist Eq.



After Local Hist Eq.
In 7x7 neighborhood



(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Local Enhancement – low contrast

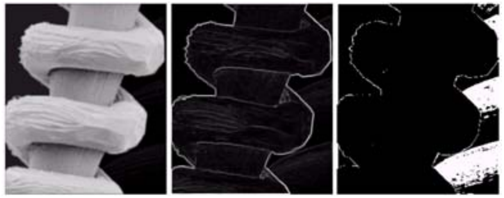
Local enhancement: compare local average & global mean, to Global standard deviation

SEM (scanning electron microscope) image of a tungsten filament

$$m_{S_{xy}} = \sum_{(s,t)\in S_{xy}} r_{s,t}\, p(r_{s,t}) \qquad \sigma^2_{S_{xy}} = \sum_{(s,t)\in S_{xy}} \left[r_{s,t} - m_{S_{xy}}\right]^2 p(r_{s,t})$$

$$g(x,y) = \begin{cases} E \cdot f(x,y) & \text{when } m_{s_{xy}} \le k_0 M_G \text{ and } k_1 D_G \le \sigma_{s_{xy}} \le k_2 D_G \\ f(x,y) & \text{otherwise} \end{cases}$$

Original image | Local Variance image | Multiplication factor



a b c

FIGURE 3.25 (a) Image formed from all local means obtained from Fig. 3.24 using Eq. (3.3-21). (b) Image formed from all local standard deviations obtained from Fig. 3.24 using Eq. (3.3-22). (c) Image formed from all multiplication constants used to produce the enhanced image shown in Fig. 3.26.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

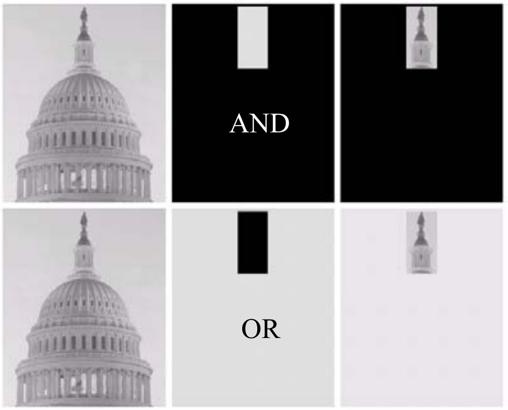
Local Enhancement

FIGURE 3.26 Enhanced SEM image. Compare with Fig. 3.24. Note in particular the enhanced area on the right side of the image.

Output image

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Logic Operations



Application:

Original image | Image mask | ROI: Region of Interest

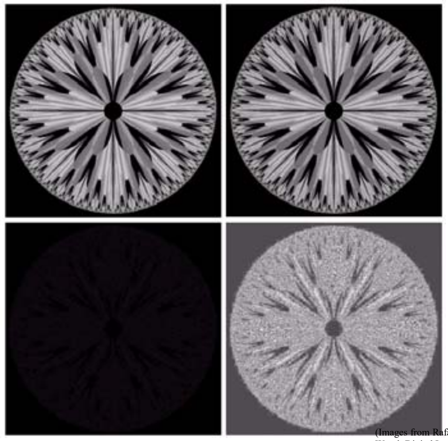
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Arithmetic Operation: Subtraction

Application: Error measurement

a b
c d

FIGURE 3.28 (a) Original fractal image. (b) Result of setting the four lower-order bit planes to zero. (c) Difference between (a) and (b). (d) Histogram-equalized difference image. (Original image courtesy of Ms. Melissa D. Binde, Swarthmore College, Swarthmore, PA).

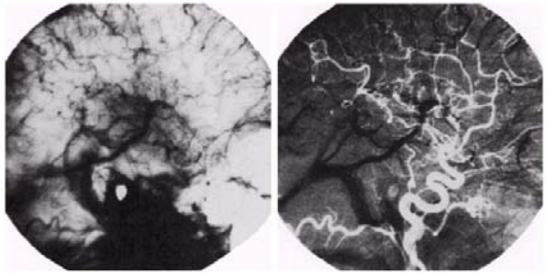


Error image

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Arithmetic Operation: Subtraction (cont.)

Application: Mask mode radiography in angiography work

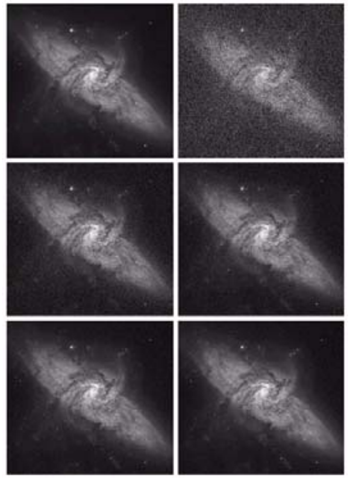


a b

FIGURE 3.29 Enhancement by image subtraction. (a) Mask image. (b) An image (taken after injection of a contrast medium into the bloodstream) with mask subtracted out.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Arithmetic Operation: Image Averaging



Application : Noise reduction

Degraded image

$$g(x,y) = f(x,y) + \eta(x,y)$$

(noise)

Image averaging

$$\bar{g}(x,y) = \frac{1}{K}\sum_{i=1}^{K} g_i(x,y)$$

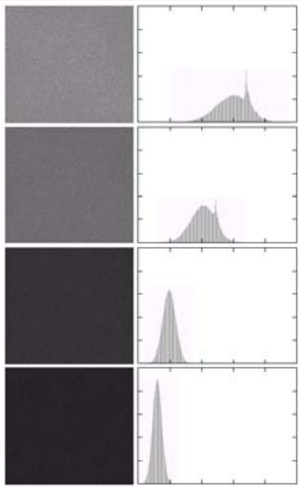
Average Variance noise

$$\sigma_{\bar{g}(x,y)} = \frac{1}{\sqrt{K}}\sigma_{\eta(x,y)}$$

a b
c d
e f

FIGURE 3.30 (a) Image of Galaxy Pair NGC 3314. (b) Image corrupted by additive Gaussian noise with zero mean and a standard deviation of 64 gray levels. (c)–(f) Results of averaging $K = 8, 16, 64$, and 128 noisy images. (Original image courtesy of NASA.)

Arithmetic Operation: Image Averaging (cont.)



a b

FIGURE 3.31 (a) From top to bottom: Difference images between Fig. 3.30(a) and the four images in Figs. 3.30(c) through (f), respectively. (b) Corresponding histograms.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Basics of Spatial Filtering

Sometimes we need to manipulate values obtained from neighboring pixels

Example: How can we compute an average value of pixels in a 3x3 region center at a pixel *z*?

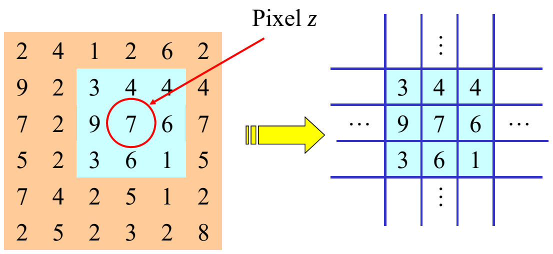
Pixel z

2	4	1	2	6	2
9	2	3	4	4	4
7	2	9	7	6	7
5	2	3	6	1	5
7	4	2	5	1	2
2	5	2	3	2	8

Image

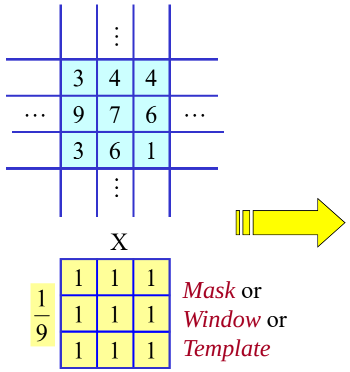
Basics of Spatial Filtering (cont.)

Step 1. Selected only needed pixels



Basics of Spatial Filtering (cont.)

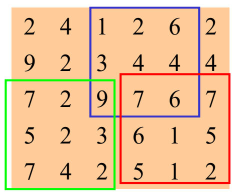
Step 2. Multiply every pixel by 1/9 and then sum up the values



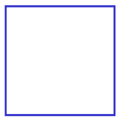
$$y = \frac{1}{9}\cdot 3 + \frac{1}{9}\cdot 4 + \frac{1}{9}\cdot 4 + \frac{1}{9}\cdot 9 + \frac{1}{9}\cdot 7 + \frac{1}{9}\cdot 6 + \frac{1}{9}\cdot 3 + \frac{1}{9}\cdot 6 + \frac{1}{9}\cdot 1$$

Basics of Spatial Filtering (cont.)

Question: How to compute the 3x3 average values at every pixels?



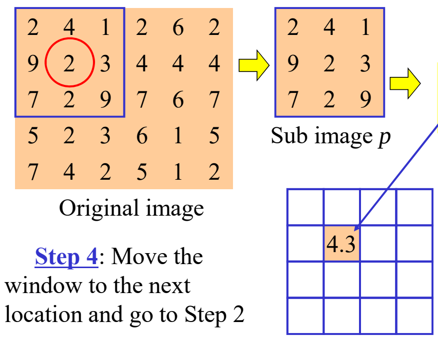
Solution: Imagine that we have a 3x3 window that can be placed everywhere on the image



Masking Window

Basics of Spatial Filtering (cont.)

Step 1: Move the window to the first location where we want to compute the average value and then select only pixels inside the window.



Step 2: Compute the average value

$$y = \sum_{i=1}^{3}\sum_{j=1}^{3}\frac{1}{9}\cdot p(i,j)$$

Step 3: Place the result at the pixel in the output image

Step 4: Move the window to the next location and go to Step 2

Basics of Spatial Filtering (cont.)

The 3x3 averaging method is one example of the ***mask operation*** or ***Spatial filtering***.

- The mask operation has the corresponding *mask* (sometimes called *window* or *template*).
- The mask contains coefficients to be multiplied with pixel values.

w(1,1)	w(2,1)	w(3,1)
w(1,2)	w(2,2)	w(3,2)
w(3,1)	w(3,2)	w(3,3)

Mask coefficients

Example : moving averaging

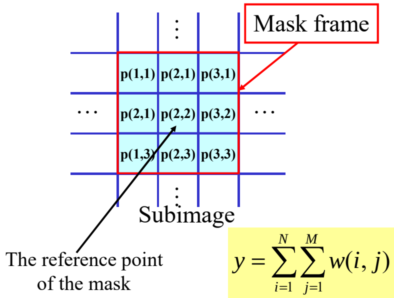
$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

The mask of the 3x3 moving average filter has all coefficients = 1/9

Basics of Spatial Filtering (cont.)

The mask operation at each point is performed by:

1. Move the reference point (center) of *mask* to the location to be computed
2. Compute ***sum of products*** between mask coefficients and pixels in subimage under the mask.



w(1,1)	w(2,1)	w(3,1)
w(1,2)	w(2,2)	w(3,2)
w(3,1)	w(3,2)	w(3,3)

Mask coefficients

$$y = \sum_{i=1}^{N}\sum_{j=1}^{M} w(i,j)\cdot p(i,j)$$

Basics of Spatial Filtering (cont.)

The spatial filtering on the whole image is given by:

1. Move the *mask* over the image at each location.
2. Compute sum of products between the mask coefficeints and pixels inside subimage under the mask.
3. Store the results at the corresponding pixels of the output image.
4. Move the mask to the next location and go to step 2 until all pixel locations have been used.

Examples of Spatial Filtering Masks

Examples of the masks

Sobel operators

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

to compute $\frac{\partial P}{\partial x}$

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

to compute $\frac{\partial P}{\partial y}$

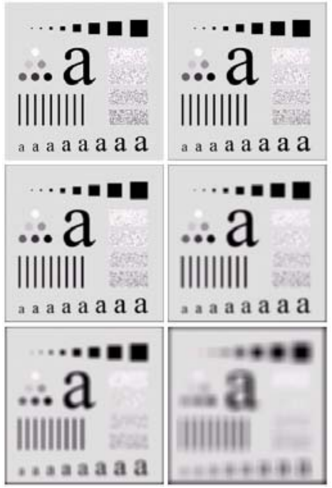
3x3 moving average filter

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

3x3 sharpening filter

$$\frac{1}{9}\begin{bmatrix}-1 & -1 & -1\\ -1 & 8 & -1\\ -1 & -1 & -1\end{bmatrix}$$

Smoothing Linear Filter : Moving Average



Application : noise reduction and image smoothing

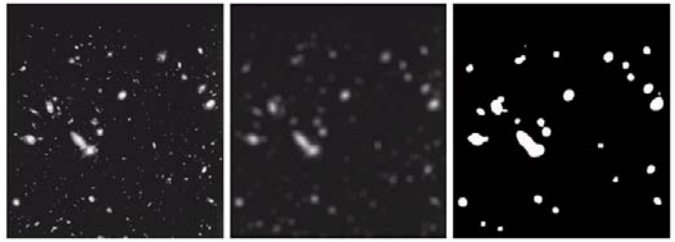
Disadvantage: lose sharp details

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

a b
c d
e f

FIGURE 3.35 (a) Original image, of size 500 × 500 pixels (b)–(f) Results of smoothing with square averaging filter masks of sizes n = 3, 5, 9, 15, and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. The letters at the bottom range in size from 10 to 24 points, in increments of 2 points; the large letter at the top is 60 points. The vertical bars are 5 pixels wide and 100 pixels high; their separation is 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart; their gray levels range from 0% to 100% black in increments of 20%. The background of the image is 10% black. The noisy rectangles are of size 50 × 120 pixels.

Smoothing Linear Filter (cont.)

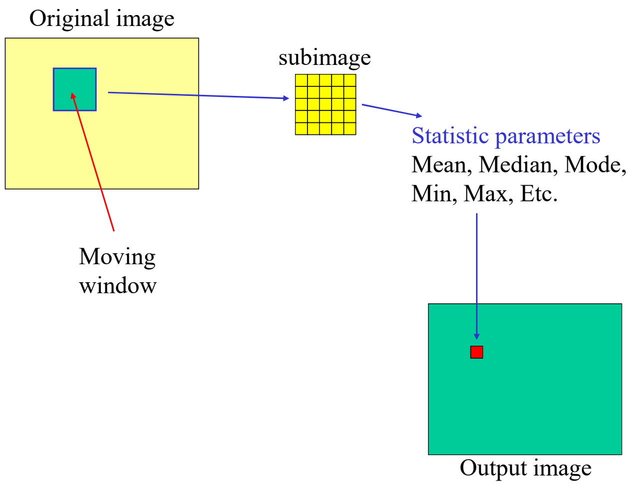


a b c

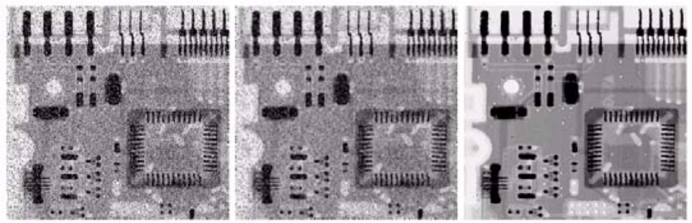
FIGURE 3.36 (a) Image from the Hubble Space Telescope. (b) Image processed by a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Order-Statistic Filters



Order-Statistic Filters: Median Filter



a b c

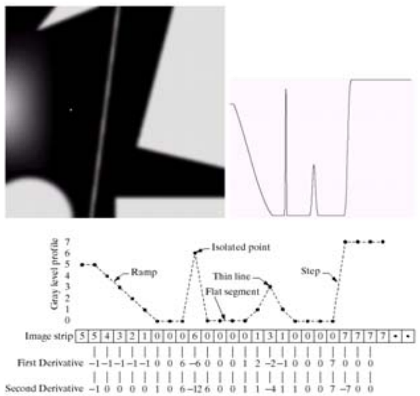
FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3 × 3 averaging mask. (c) Noise reduction with a 3 × 3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Sharpening Spatial Filters

There are intensity discontinuities near object edges in an image

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).



(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

First and Second Derivative (Laplacian)

$$\frac{\partial f}{\partial x} = f(x+1) - f(x).$$ The first order derivative

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

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}.$$ Laplacian : an isotropic derivative operator

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

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

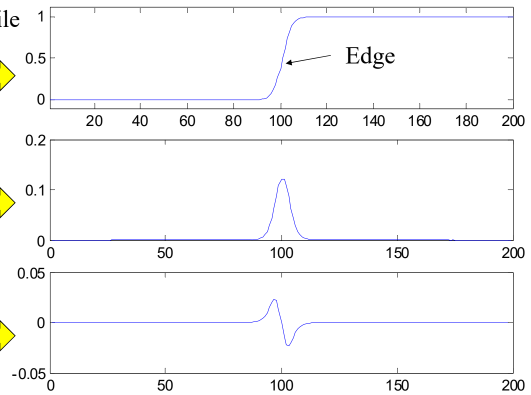
$$\nabla^2 f = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] - 4f(x, y).$$

Laplacian Sharpening : How it works

Intensity profile $p(x)$

1st derivative $\frac{dp}{dx}$

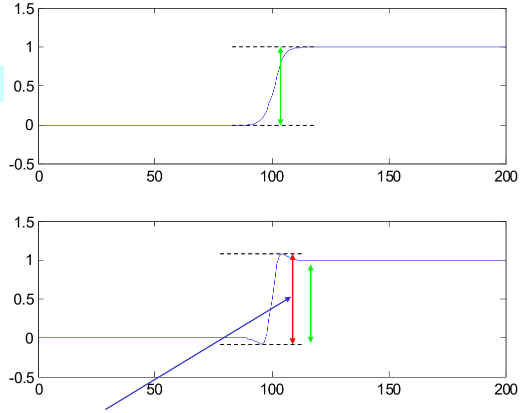
2nd derivative $\frac{d^2 p}{dx^2}$



Laplacian Sharpening : How it works (cont.)

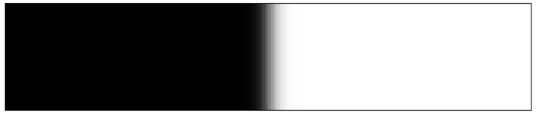
$p(x)$

$p(x) - 10\frac{d^2 p}{dx^2}$



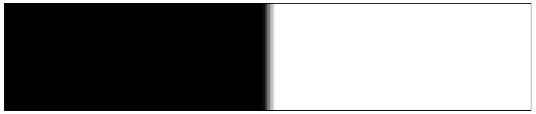
Laplacian sharpening results in larger intensity discontinuity near the edge.

Laplacian Sharpening : How it works (cont.)



Before sharpening

$p(x)$



After sharpening

$$p(x) - 10\frac{d^2 p}{dx^2}$$

Laplacian Masks

Used for estimating image Laplacian $\nabla^2 P = \frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2}$

-1	-1	-1
-1	8	-1
-1	-1	-1

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

The center of the mask is positive

or

1	1	1
1	-8	1
1	1	1

0	1	0
1	-4	1
0	1	0

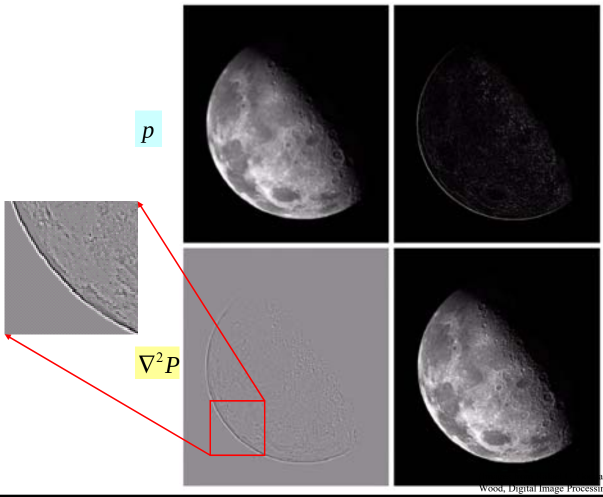
The center of the mask is negative

Isotropic : rotation invacriant

Application: Enhance edge, line, point

Disadvantage: Enhance noise

Laplacian Sharpening Example



(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Laplacian Sharpening (cont.)

Mask for

$P - \nabla^2 P$

0	-1	0
-1	5	-1
0	-1	0

or

-1	-1	-1
-1	9	-1
-1	-1	-1

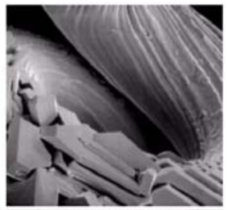
Mask for

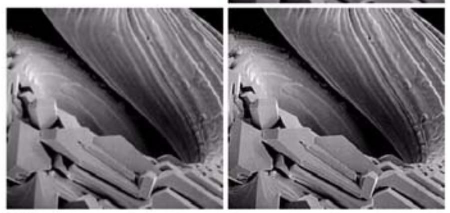
$\nabla^2 P$

1	1	1
1	-8	1
1	1	1

or

0	1	0
1	-4	1
0	1	0





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.)

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Unsharp Masking and High-Boost Filtering

Subtract a blurred version from the original : *unsharp masking*

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

A further generalization: *unsharphigh-boost filtering*

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

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

$$f_{hb}(x, y) = (A - 1)f(x, y) + f_s(x, y)$$

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

Unsharp Masking and High-Boost Filtering

-1	-1	-1
-1	A+8	-1
-1	-1	-1

0	-1	0
-1	A+4	-1
0	-1	0

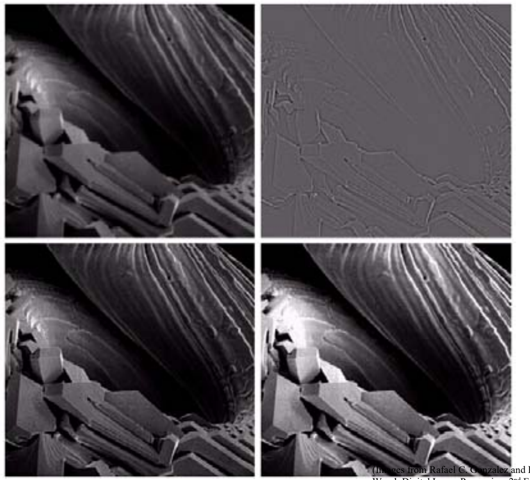
Equation:

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

Unsharp Masking and High-Boost Filtering (cont.)

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 $A = 1.7$.



(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

First Order Derivative – the Gradient

Graient of f at (x,y)

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

Magnitude of the gradient

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

Approximation

$$\nabla f \approx |G_x| + |G_y|.$$

First Order Derivative – the Gradient

Intensity profile

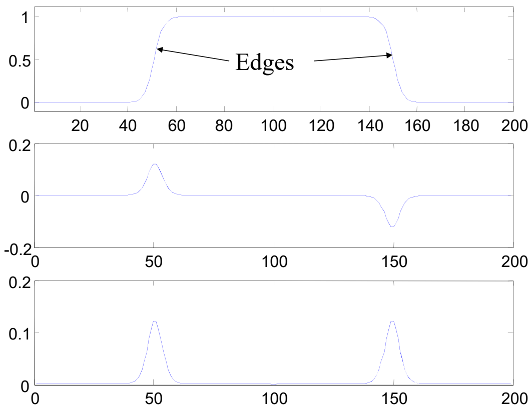
$f(x)$

1st derivative

$\frac{df}{dx}$

2nd derivative

$\left|\frac{df}{dx}\right|$



Use of Gradient for Edge Enhancement

A 3*3 region of an image (the z's are gray-level values) and

masks used to compute the gradient at point labeled z5.

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

−1	0
0	1

0	−1
1	0

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

First Order Partial Derivative:

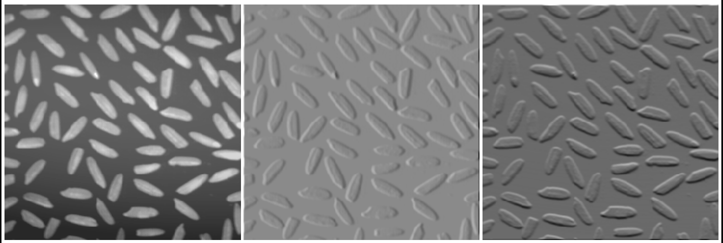
Sobel operators

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

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

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

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

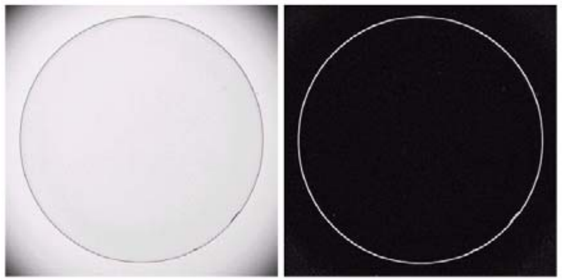


P $\qquad \frac{\partial P}{\partial x}$ $\qquad \frac{\partial P}{\partial y}$

First Order Partial Derivative: Image Gradient

Gradient magnitude

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



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.)

A gradient image emphasizes edges

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

First Order Partial Derivative: Image Gradient

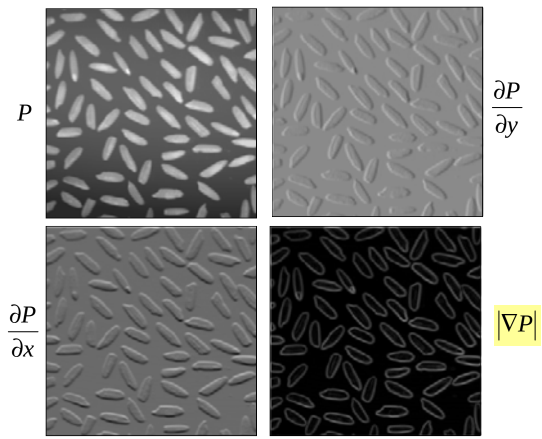


Image Enhancement in the Spatial Domain : Mix things up !

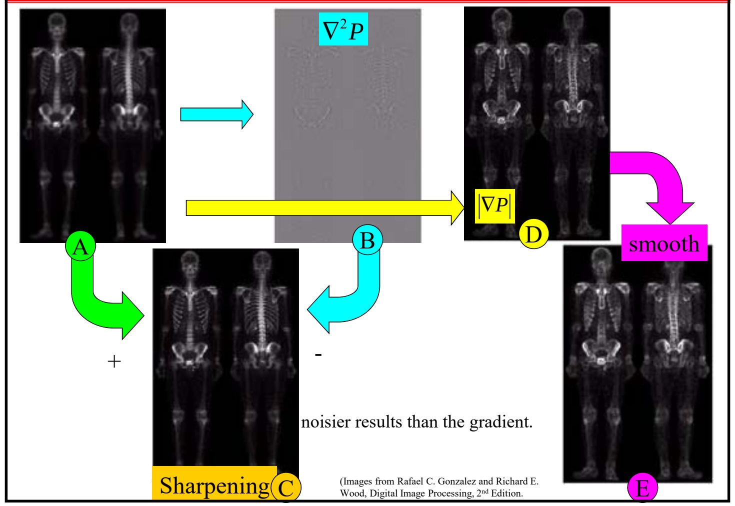
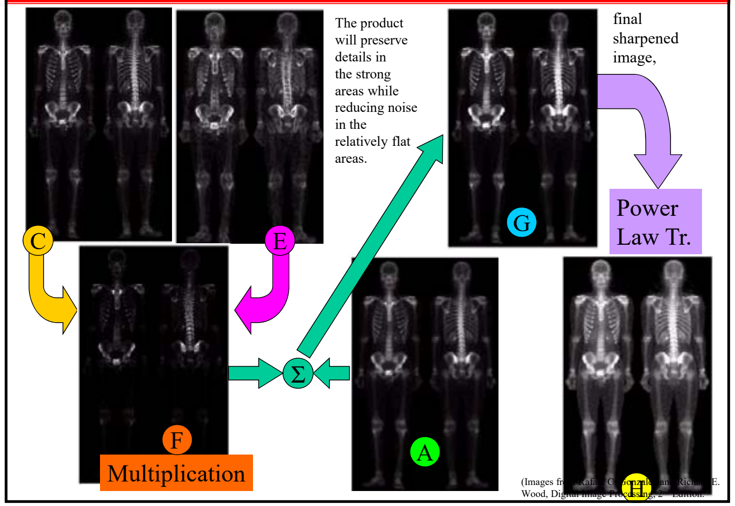


Image Enhancement in the Spatial Domain : Mix things up !



[Challenges] blurred text : Can these be enhanced?

Image Enhancement of blurred text