IMAGE PROCESSING

Le Thanh Ha, Ph.D
Assoc. Prof. at University of Engineering and Technology,
Vietnam National University
lt; ; 0983 692 592

Content

3. Intensity Transformation & Spatial Filtering
    - Some basic transformations
    - Histogram processing
    - Spatial filtering (Smoothing, Sharpening, Edge detection)
    - …

Digital Image

- A continuous image $a(x,y)$ is sampled and quantized to form a digital image presented by a matrix with N rows and M column. One element in the matrix is called “Pixel”.
- The value of a pixel , called gray level, is assigned with an integer I[m,n] $\{m=0,1,2,...,M-1\}$ and $\{n=0,1,2,...,N-1\}$

But the camera sees this:

194	210	201	212	199	213	215	195	178	158	182	209
180	189	190	221	209	205	191	167	147	115	129	163
114	126	140	188	176	165	152	140	170	106	78	88
87	103	115	154	143	142	149	153	173	101	57	57
102	112	106	131	122	138	152	147	128	84	58	66
94	95	79	104	105	124	129	113	107	87	69	67
68	71	69	98	89	92	98	95	89	88	76	67
41	56	68	99	63	45	60	82	58	76	74	65
20	41	69	75	56	41	51	73	55	70	63	44
50	50	57	69	75	75	73	74	53	68	59	37
72	59	53	66	84	92	84	74	57	72	63	42
67	61	58	65	75	78	76	73	59	75	69	50

Values frequently used

Parameter	*Symbol*	*Values frequently used*
row	N	256,512,525,625,1024,1035
Column	M	256,512,768,1024,1320
Gray level	L	2,64,256,1024,4096,16384

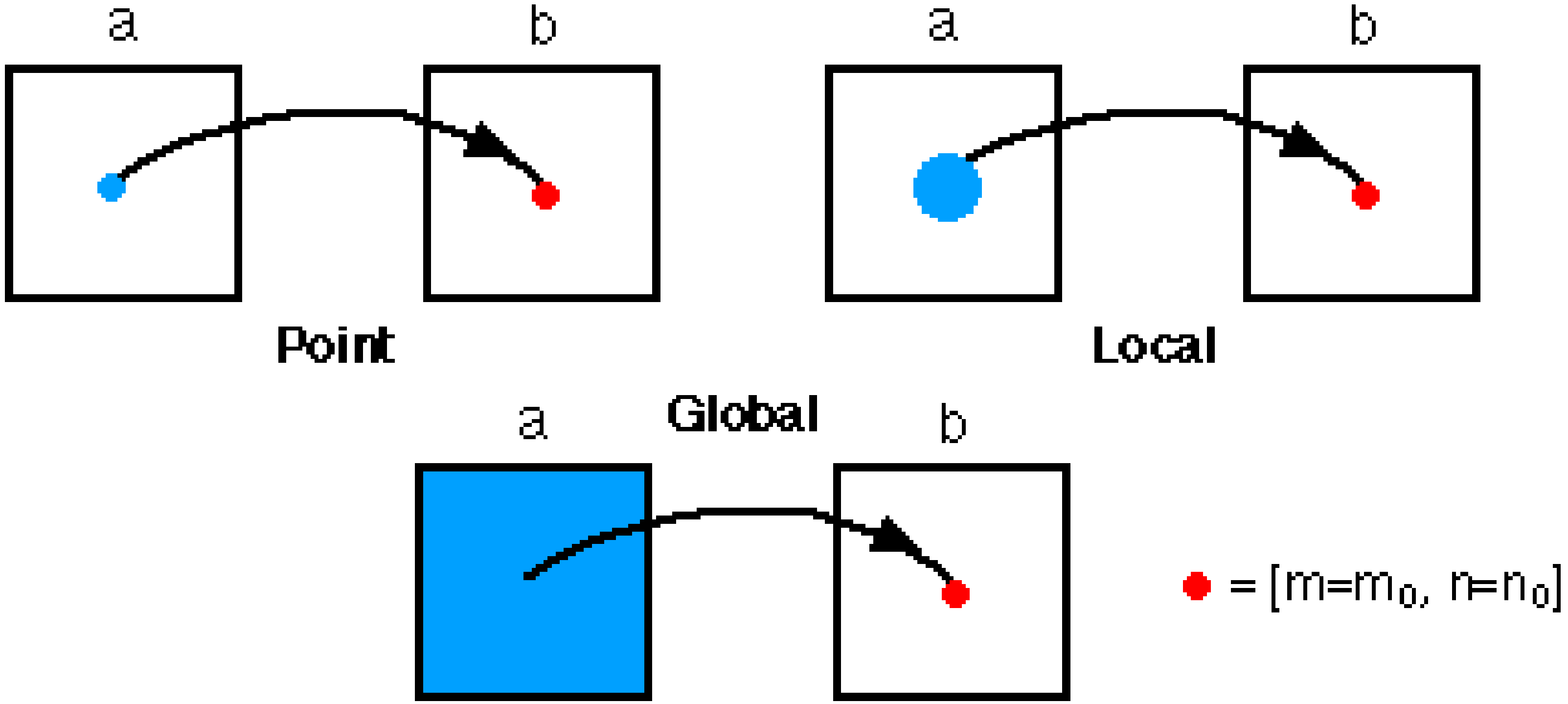
Image operator types

- Image operator takes images as inputs and also output images.
- Operators for Images are classified based on the affecting region of the operators. There are three types:
  - Point
  - Local
  - Global

Image operator types

Operator	**Describe**	**Complexity**
Point	The value of output pixel at a specific location depends only on that of pixel at that location.	*Constant*
Local	The value of output pixel at a specific location depends on the values of the pixel's neighbors	P^2
Global	The value of output every pixel depends on all pixels in the image.	N^2

Các thao tác trên ảnh (...)



-Image negatives
-Power-law (gamma) transformations
- Contrast stretching
- Intensity-level slicing
-Histogram processing

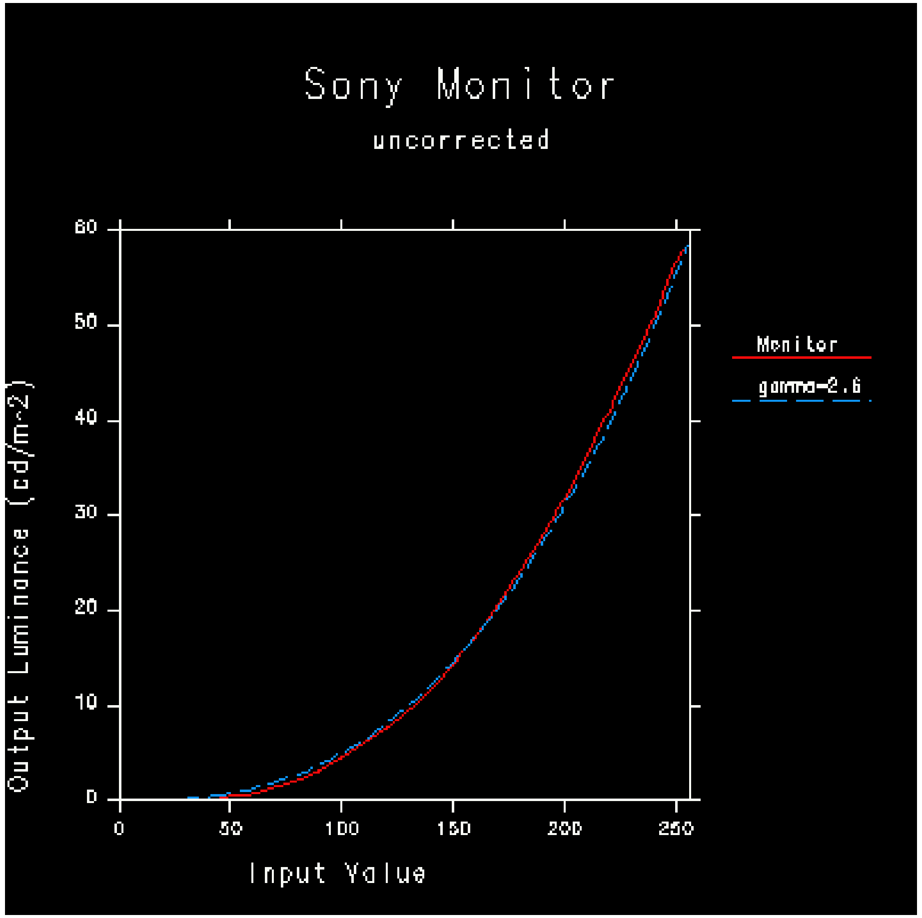
INTENSITY TRANSFORMATION (POINT OPERATORS)

Image negatives

- $I(x,y) = 255 - I(x,y)$

Power-law (gamma) transformations

- A CRT device has an intensity-to-voltage response that is a power function.
  - The displayed images are darker than they actually are.

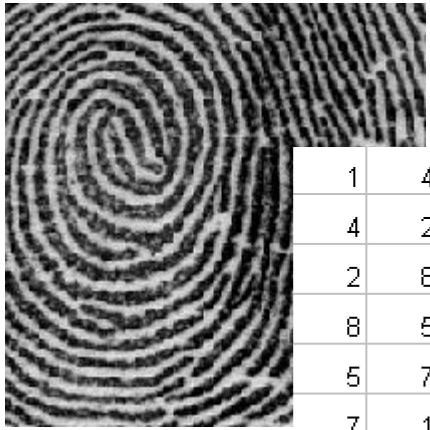


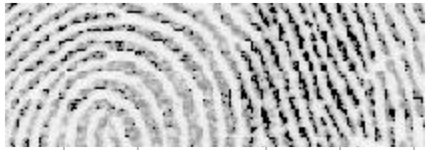
Power-law (gamma) transformations

- Need a gamma correction for these CRT devices:

  $I(x,y) \leftarrow I(x,y)^{\gamma}$

  Example with γ=2





1	4	2	8	5	7				1	16	4	64	25	49
4	2	8	5	7	1				16	4	64	25	49	1
2	8	5	7	1	4		gamma=2		4	64	25	49	1	16
8	5	7	1	4	2				64	25	49	1	16	4
5	7	1	4	2	8				25	49	1	16	4	64
7	1	4	2	8	5				49	1	16	4	64	25
1	4	2	8	5	7				1	16	4	64	25	49

Contrast stretching

- Low-contrast images can result from:
  - Poor illumination
  - Small sensitive range in imaging sensor
  - …

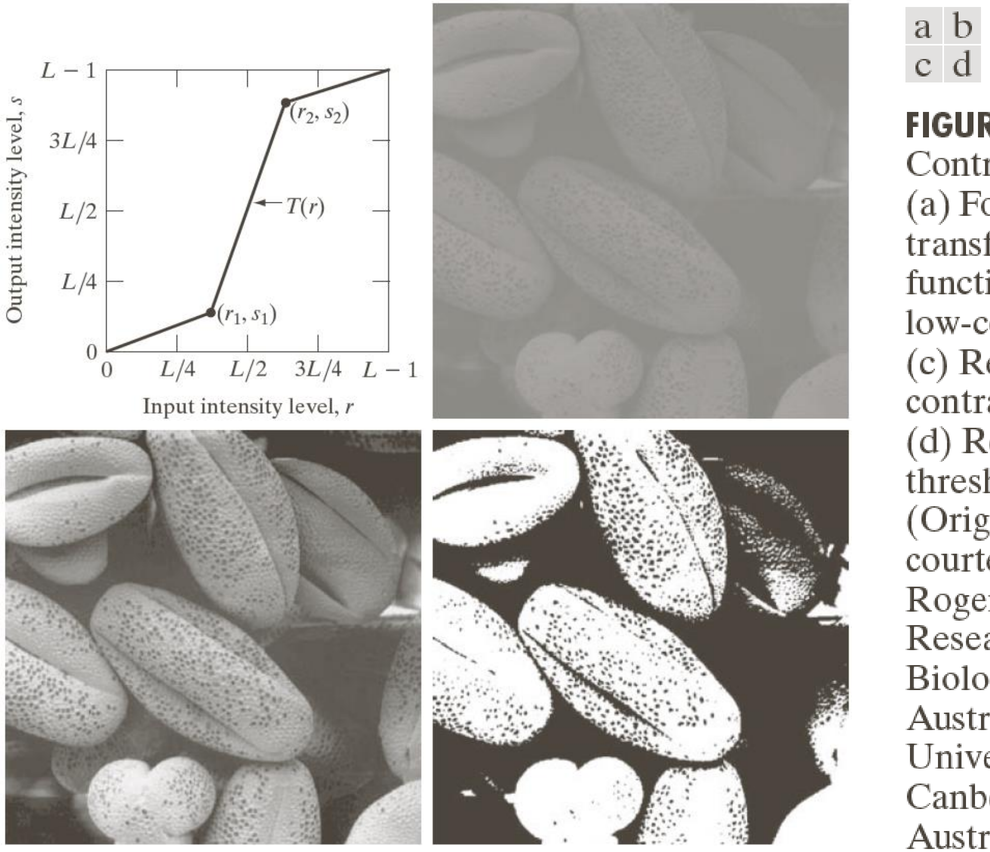
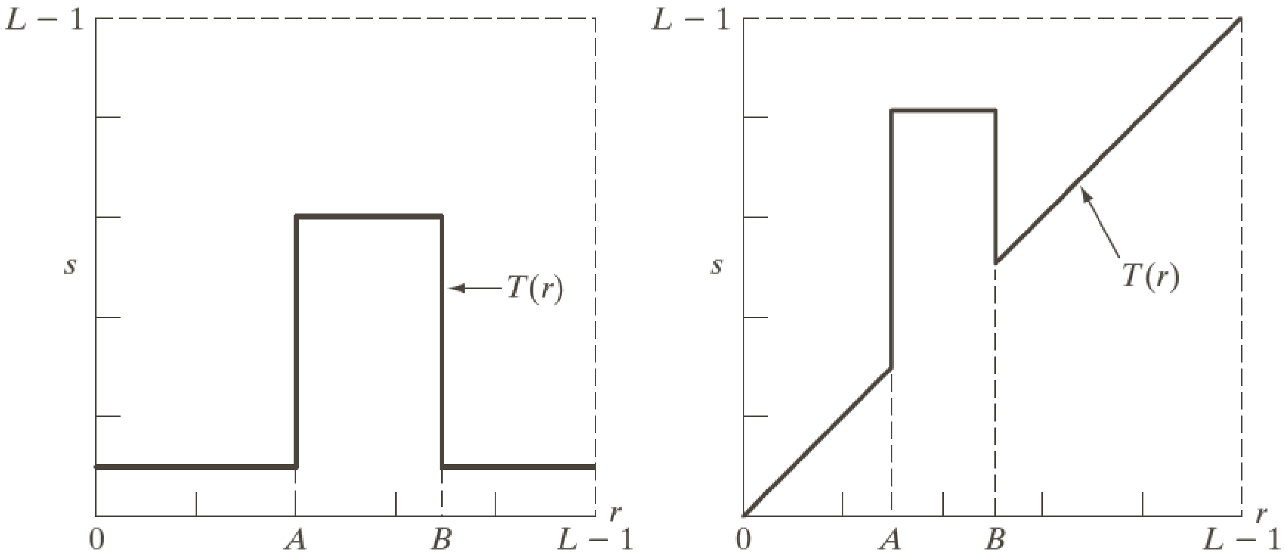


FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

Intensity-level slicing

a b

FIGURE 3.11 (a) This transformation highlights intensity range [A, B] and reduces all other intensities to a lower level. (b) This transformation highlights range [A, B] and preserves all other intensity levels.



- Highligting a interest range of intensities for specific applications:
  - Enhancing features
  - Two types of highligting transformation functions

HMI
HUMAN MACHINE INTERACTION LABORATORY

Histogram processing

- Simply, histogram shows the count number of each gray level.
- In maths, histogram is the probability mass function of intensity random variable.
- Histogram applications:
  – Easily to be implemented in hardware
  – How contrast an image is.
  – Useful for applications involved with image segmentation and compression.
  – …

Histogram processing

$$\begin{bmatrix} 1 & 2 & 1 & 2 & 1 & 1 \\ 1 & 3 & 3 & 2 & 3 & 1 \\ 1 & 5 & 6 & 5 & 5 & 1 \\ 2 & 5 & 7 & 7 & 4 & 2 \\ 1 & 3 & 3 & 2 & 3 & 1 \\ 1 & 2 & 1 & 1 & 1 & 2 \end{bmatrix}$$

- H(i) is the number of pixels with gray level i.
  - H(1) = 14
  - H(2) = 8
  - H(3) = 6
  - H(4) = 1
  - H(5) = 4
  - H(6) = 1
  - H(7) = 2

Histogram processing

- Dark image: large numbers at low intensity range.
- Light image: large numbers at high intensity range.
- Low-contrast image: histogram is condensed.
- High-contrast image: histogram is look like a uniform distribution.

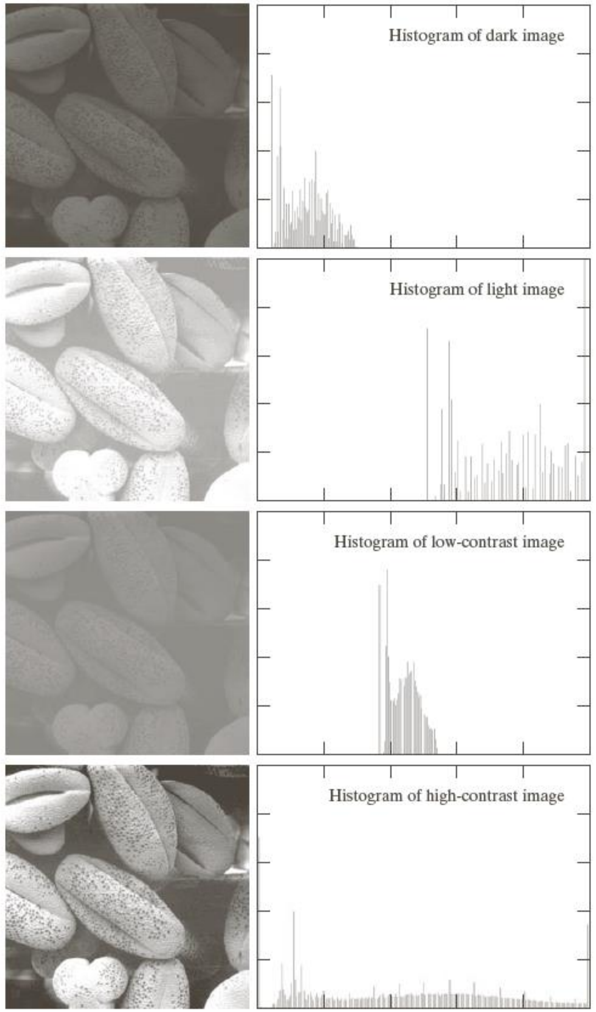


FIGURE 3.16 Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms.

Histogram Equalization

- Is used to increase the contrast of a given image by linearizing its histogram.

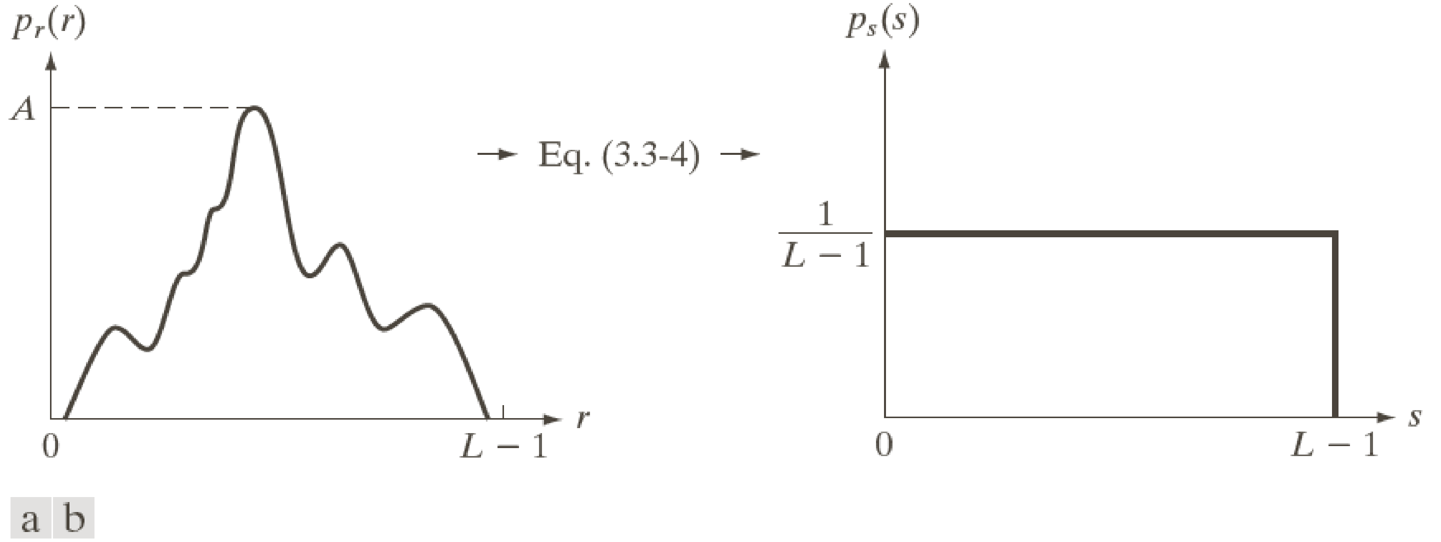


FIGURE 3.18 (a) An arbitrary PDF. (b) Result of applying the transformation in Eq. (3.3-4) to all intensity levels, r. The resulting intensities, s, have a uniform PDF, independently of the form of the PDF of the r's.

Histogram Equalization

- Pseudocode
  - Input I[M,N]:image, L:maximum gray level.
  - Output I_{he}[M,N]: histogram equalized image.
  - Procedure:

```
1.  Calculate histogram p_I for image I.
2.  T[0] = p_I[0]
3.  For k=1 to L
        T[k] = T[k-1] + p_I[k]
    End for

4.  For r=0 to L
        S[r] = round(T[r]*L)
    End for

5.  For y=0 to M-1
    For x=0 to N-1
        r = I[y,x]
        I_he[y,x] = S[r]
    End for
    End for
```

$$I = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 2 & 2 \\ 0 & 0 & 0 & 2 \\ 2 & 2 & 3 & 3 \end{bmatrix}$$

Histogram Equalization

- Equalize histogram of following image, (L=7):

$$I = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 2 & 2 \\ 0 & 0 & 0 & 2 \\ 2 & 2 & 3 & 3 \end{bmatrix}$$

Histogram Equalization

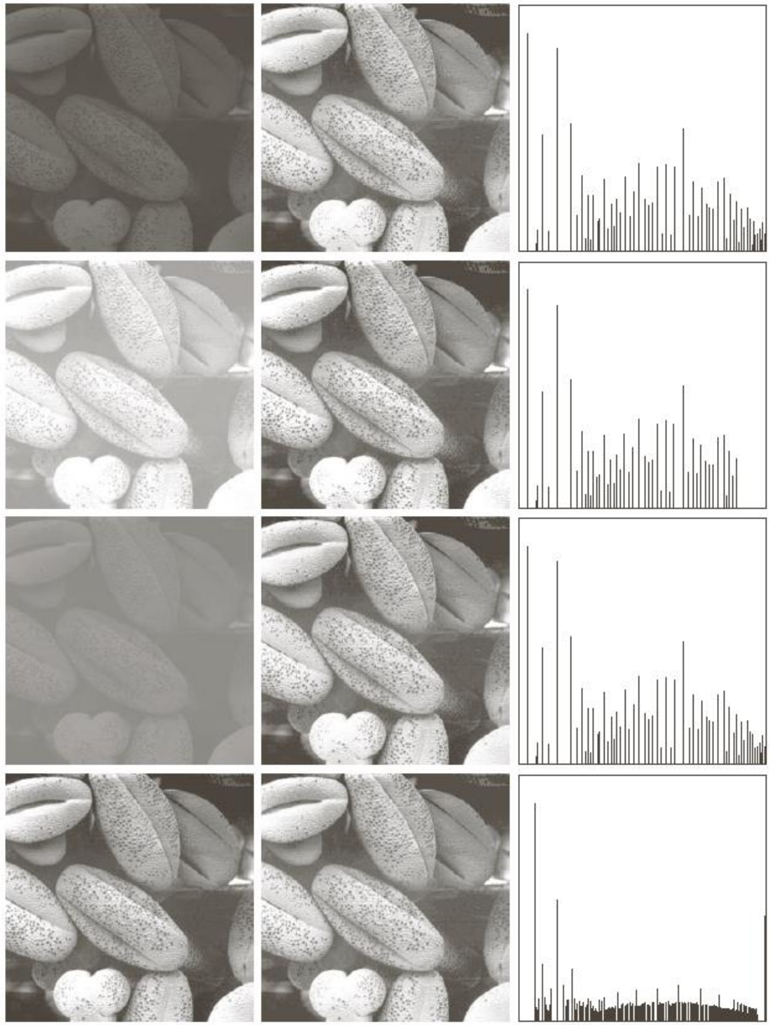


FIGURE 3.20 Left column: images from Fig. 3.16. Center column: corresponding histogram-equalized images. Right column: histograms of the images in the center column.

-Convolution

-Convolution for Image Processing

-Average filtering

-Median filtering

SPATIAL FILTERING

Convolution

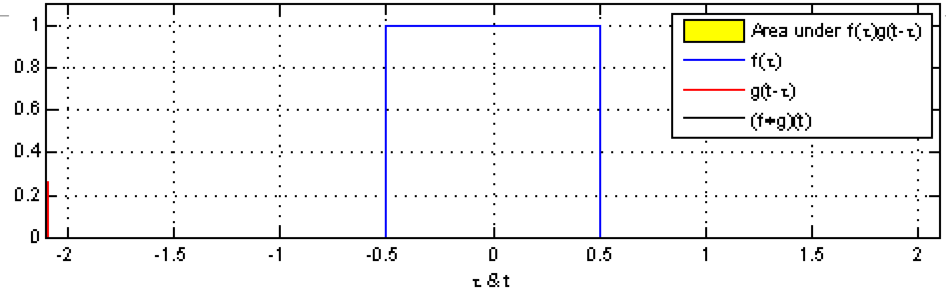
- Convolution of two continous signals:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$$

- Convolution of two discrete signals:

$$(f * g)[n] = \sum_{i=-\infty}^{\infty} f[i]g[n-i]$$

- Example:



Convolution

- Convolution of 2D continuous signals:

$$(f*g)(x,y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\tau_x,\tau_y)g(x-\tau_x,y-\tau_y)d\tau_x d\tau_y$$

- Convolution of 2D discrete signals:

$$(f*g)[m,n]=\sum_{j=-\infty}^{\infty}\sum_{i=-\infty}^{\infty} f[m,n]g[m-i,n-j]$$

Properties of convolution

- Commulative:

  $c = a*b = b*a$

- Associativity:

  $d = a*(b*c) = (a*b)*c$

- Distributivity

  $d = a*(b+c) = a*b + a*c$

Convolution for Image Processing

- Discrete convolution for Image Processing:

$$f[x,y] = I[x,y] * h[x,y]$$

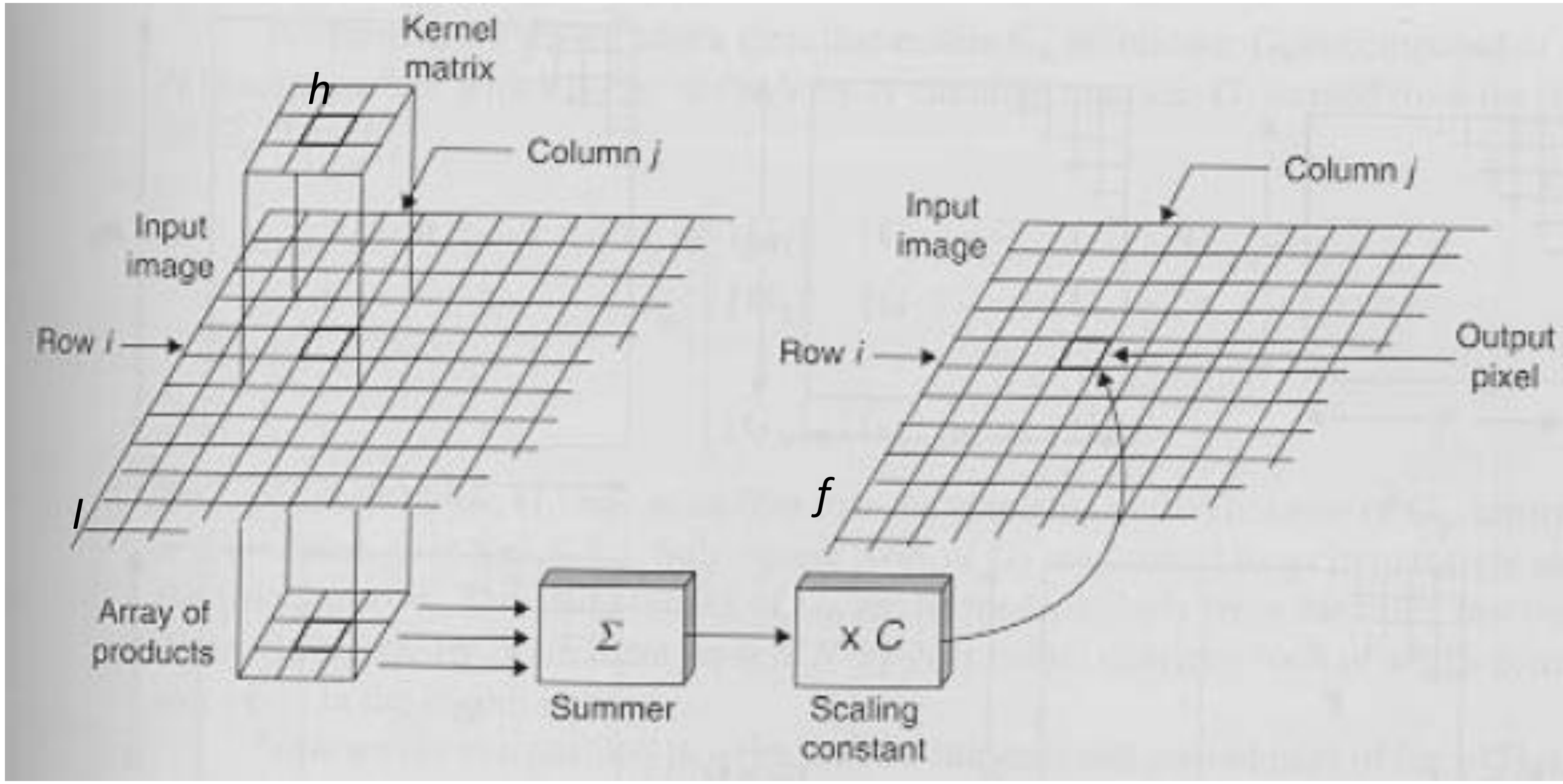
$$= \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} I[x,y]h[x-i, y-j]$$

$$= \sum_{j=1}^{M} \sum_{i=1}^{N} I[x,y]h[x-i, y-j]$$

- Image border problem

Convolution for Image Processing

- Matrix manipulation:



Convolution for Image Processing

- Filtering

$$f[x, y] = I[x, y] * h[x, y]$$

  - *I* is an Image, *h* is a mask
  - A pixel called anchor for the mask (topleft, or center)

Convolution for Image Processing

- Mask h: 3x3 – An average filter used to smoothen image
- Anchor point is (1,1) in the mask.

1/8	1/8	1/8
1/8	0	1/8
1/8	1/8	1/8

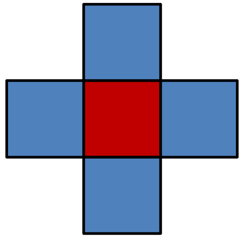
Convolution for Image Processing

Border problem

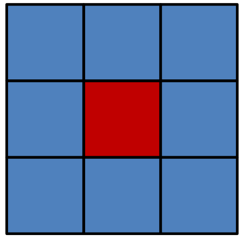
- It happens when calculating convolution at border points where some points of mask is placed outside of the image.
- It can be solved by padding the missing pixel by zero value or copy vale from the nearest available pixels.

Mask shape

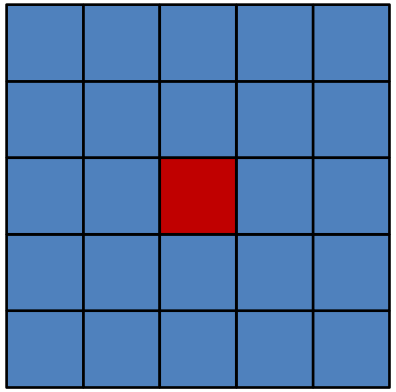
- The value of output pixel at a specific location depends on the values of the pixel's neighbors



4-neighbor

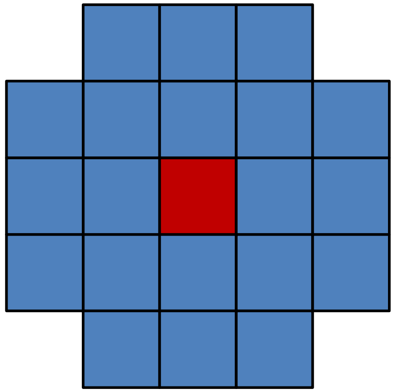


8-neighbor
Mask size 3x3

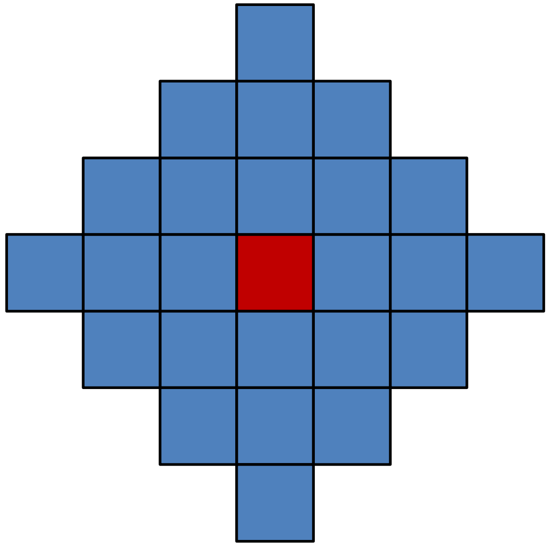


Mask size 5x5

Mask shape



Round with radius of 3



Lozenge shape

Local Image Processing methods

- Smoothing and noise removal
  - Average filtering
  - Median filtering
  - Probability filtering
- Edge detection and highboost filtering
  - Gradient methods
  - Laplace method
  - Highboost filtering

Average Filter

- Random noise typically consists of sharp transitions in gray levels

- The most obvious application of smoothing is noise reduction

Average Filter

- Average filter is to replace the value of every pixel by the average of the gray levels in the neighborhood defined by the filter mask.

- Beside noise, edges are charaterized by sharp transitions $\rightarrow$ average filter also blurs edges.

Average Filter

- Frequently used masks:

$$h_1 = \begin{bmatrix} 1/4 & 1/4 \\ 1/4 & 1/4 \end{bmatrix} \qquad h_2 = \begin{bmatrix} 0 & 1/6 & 0 \\ 1/6 & 1/3 & 1/6 \\ 0 & 1/6 & 0 \end{bmatrix}$$

$$h_3 = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} \qquad h_4 = \frac{1}{10} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Average Filter

Original Image

Mask: h_1

Mask: h_2

Mask: h_3

Mask: h_4

Average Filter

- The effects of smoothing as a function of filter size
  - The larger is the size, the more significant blurring observed.

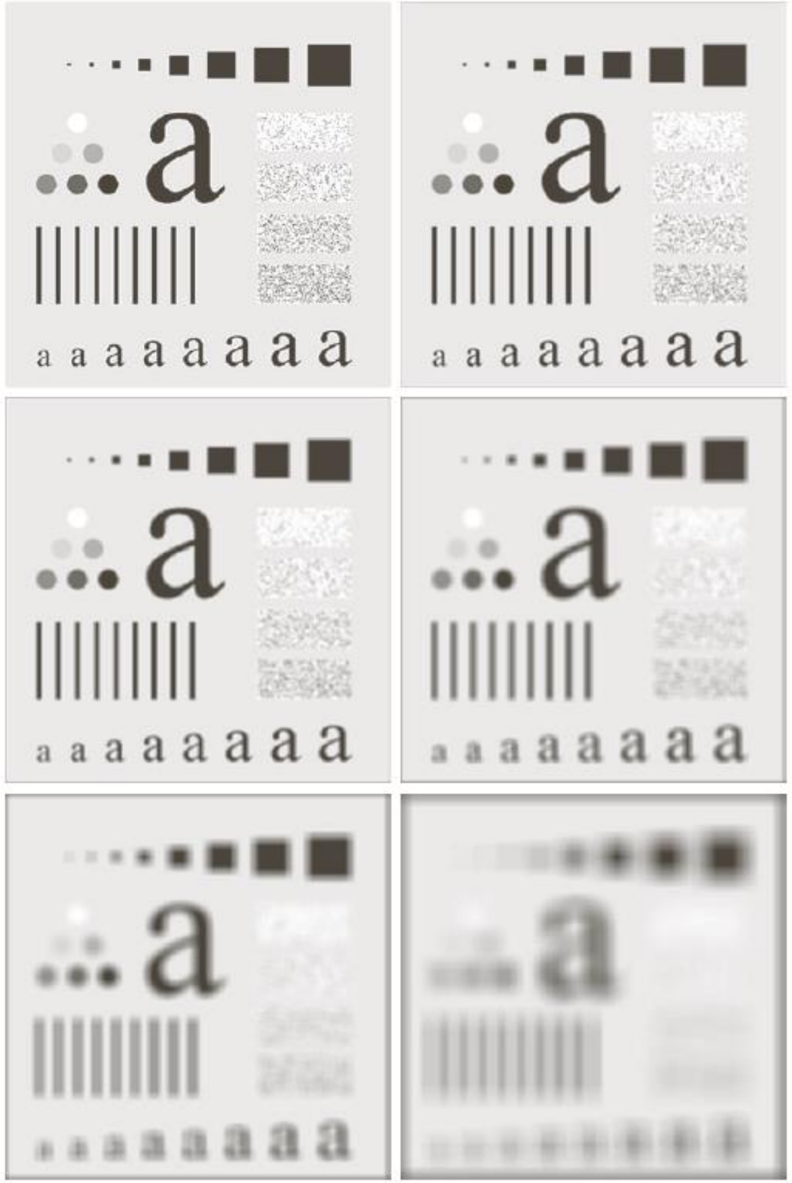


FIGURE 3.33 (a) Original image, of size 500 × 500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $m = 3, 5, 9, 15$, and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. 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 intensity 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.

a b
c d
e f

Median Filter

- For each p pixel in the image:
  - Collect all neighbor pixels which are in the mask.
  - Sort these pixel in the order of pixel’s value.
  - Chose the pixel at the center of the sorted pixel array.
  - Assign the value of chosen pixel to p.

- Median Filter are often used for noise removal.

Median Filter

- Median Filter is different from Average Filter.
- A = {5 4 1 6 2 7 2 6 2 10 2 3}
- The average value is 4.667
- The median value:
  - Sort array A = [1 2 2 2 2 3 4 5 6 6 7 10]
  - Choose the the center value A = [1 2 2 2 2 3 4 5 6 6 7 10]
  - The median value is 3

Median Filter

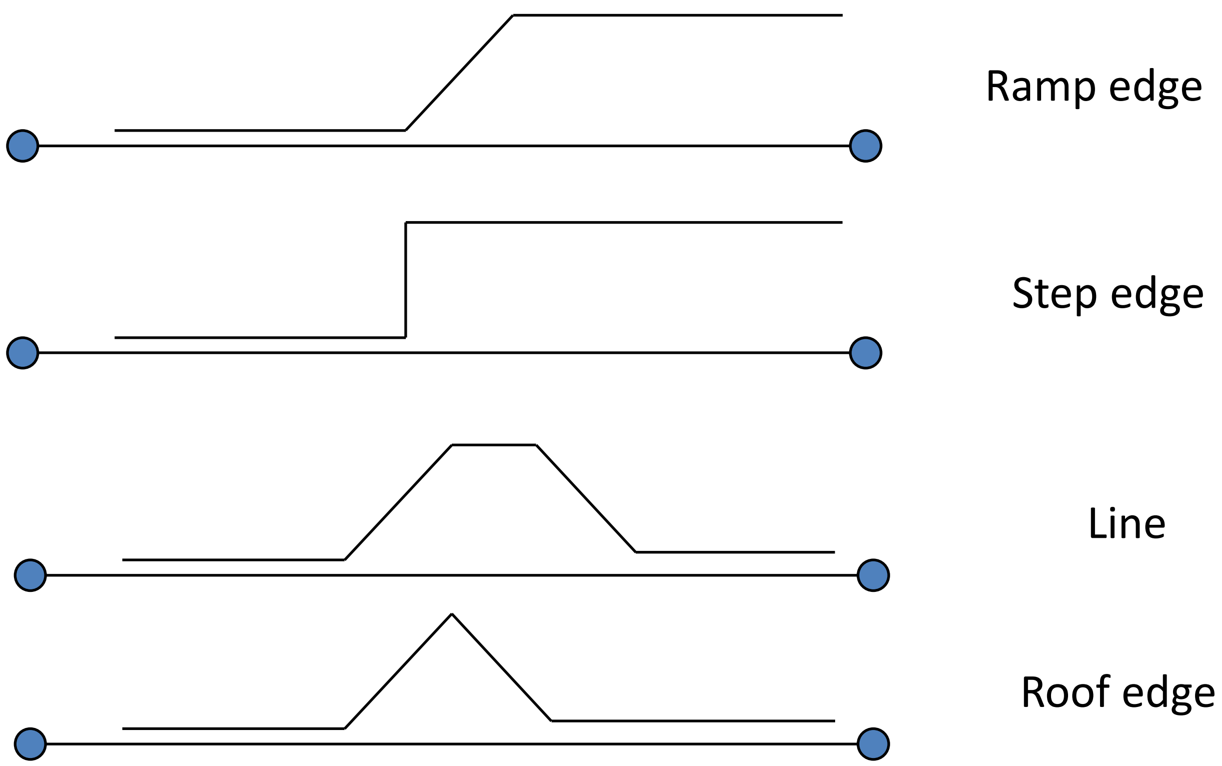
Orignal Image

Noised images

Noise removed

Edge detection - Gradient

- Edges in 1 dimesion:



Gradient

- Gradient of a function indicates how the function strongest increases.
  - For 1-dimension function: $f(x) = x^2$

$$Grad(x) = \frac{\partial f(x)}{\partial(x)} = 2x$$

  - Grad(2)=4 indicates the the increasing direction of the function is to the right.
  - Grad(-1)=-2 indicates the increasing direction of the function is to the left.

Edge detection - Gradient

- Gradient of a 2-dimension function is calculated as follows:

$$Grad(x,y) = \frac{\partial f(x,y)}{\partial x}\vec{i} + \frac{\partial f(x,y)}{\partial y}\vec{j}$$

- The gradient is approximated as follows (first-order derivative) :

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

Edge detection - Gradient

- The magnitude of gradient indicates the strong of edges:

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

- Edge dection procedure:
  – Calculate column gradient
  – Calculate row gradient
  – Calculate final gradient by the above function

Edge detection - Gradient

- Pixel Difference masks

Column mask

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

Row mask

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

5	2	7	2	6	3
3	2	8	3	5	2
7	8	2	4	7	3
4	2	6	2	5	2

Edge Detection – Gradient

Original Image

Column edges

Row edges

Final edges

Edge detection - Gradient

- Roberts masks calculate gradient from two diagonals

Column

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

Row

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

Edge detection - Gradient

Original Image

Column edges

Row edges

Final edges

Edge detection - Gradient

- Prewitt masks

Column	Row
$\frac{1}{3}\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$	$\frac{1}{3}\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$

Edge detection - Gradient

Original Image

Row edges

Column edges

Final edges

Edge detection - Gradient

- Sobel masks

$$\begin{array}{cc} \text{Column} & \text{Row} \\ \frac{1}{4}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} & \frac{1}{4}\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \end{array}$$

Edge detection - Gradient

Column

Original Image

Final edges

Row

10/5/2022
54

Edge detection - compass

- Image is convoluted with 8 masks
- Each mask is sensitive to a geometric direction of edge.
- The mask with highest value is chosen.

Edge detection - compass

- T is a template mask
- Let T_0=T; T_i is obtained by rotating T with an angle of $i*\pi/4$.
- $A(x,y)=\max\{|T_i*I(x,y)|: i=0,1,2,\ldots,7\}$

$$\Pr ewitt=\begin{bmatrix}1 & 1 & -1\\ 1 & -2 & -1\\ 1 & 1 & -1\end{bmatrix}; Kirsh=\begin{bmatrix}5 & -3 & -3\\ 5 & 0 & -3\\ 5 & -3 & -3\end{bmatrix};$$

$$\begin{matrix}Robinson\\ bac3\end{matrix}=\begin{bmatrix}1 & 0 & -1\\ 1 & 0 & -1\\ 1 & 0 & -1\end{bmatrix}; \begin{matrix}Robinson\\ bac5\end{matrix}=\begin{bmatrix}1 & 0 & -1\\ 2 & 0 & -2\\ 1 & 0 & -1\end{bmatrix}$$

Edge detection - compass

Example: Kirsh mask

$$H_0 = \begin{pmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{pmatrix}; H_1 = \begin{pmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix}; H_2 = \begin{pmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix}; H_3 = \begin{pmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{pmatrix};$$

$$H_4 = \begin{pmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{pmatrix}; H_5 = \begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{pmatrix}; H_6 = \begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{pmatrix}; H_7 = \begin{pmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{pmatrix};$$

$$|G| = max(|G_i| : i = 1 \ to \ n)$$

Edge detection – Laplace

- Laplace edge in the continuous domain is defined as follows:

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

Edge detection - Laplace

- In discrete domain, Laplace edge is approximated by the second order of dirivative:

$$G(x,y) = [f(x,y) - f(x,y-1] - [f(x,y+1) - f(x,y)] \\ + [f(x,y) - f(x+1,y] - [f(x-1,y) - f(x,y)]$$

$$= f(x,y) * H(x,y)$$

Edge detection - Laplace

- Mask H

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

$$= \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

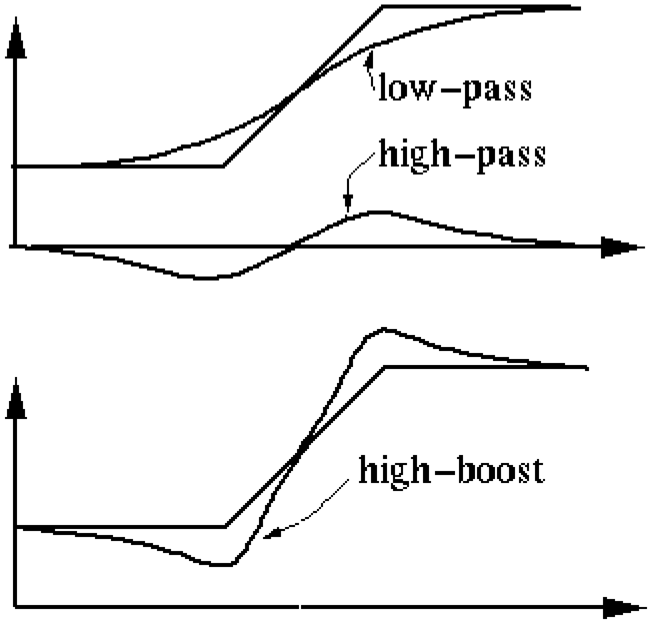
Edge detection - Laplace

Original image I

|I*H|

Highboost filtering

- Highboost the details (high frequencies) of an image without defect the image form (low frequencies)



Highboost filtering

- Overall method:

$$I_{highboost} = c \cdot I_{orginal} + I_{highpass}$$
$$= \left(c \cdot \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + H \right) * I_{original}$$

- Using Laplace mask

$$I_{highboost} = \left(c \cdot \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \right) * I_{original} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & c+4 & -1 \\ 0 & -1 & 0 \end{bmatrix} * I_{original}$$

Highboost filtering

Original Image

c=0.5

c=1