Chapter 2

Fundamentals of Digital Image and Spatial Domain Enhancement

2.1 Digital image Representation, Elements of digital image processing systems, sampling and quantization, basic relationships between pixels, mathematical operations on images.

2.2 Spatial domain enhancement techniques: Point processing, Neighborhood processing, spatial domain filtering, zooming.

2.3 Spatial enhancement: Global processing: Histogram Equalization.

Self-Learning Topic: Histogram specification

What is Digital Image Processing?

Digital image processing focuses on two major tasks

- Improvement of pictorial information for human interpretation
- Processing of image data for storage, transmission and representation for autonomous machine perception

Some argument about where image processing ends and fields such as image analysis and computer vision start

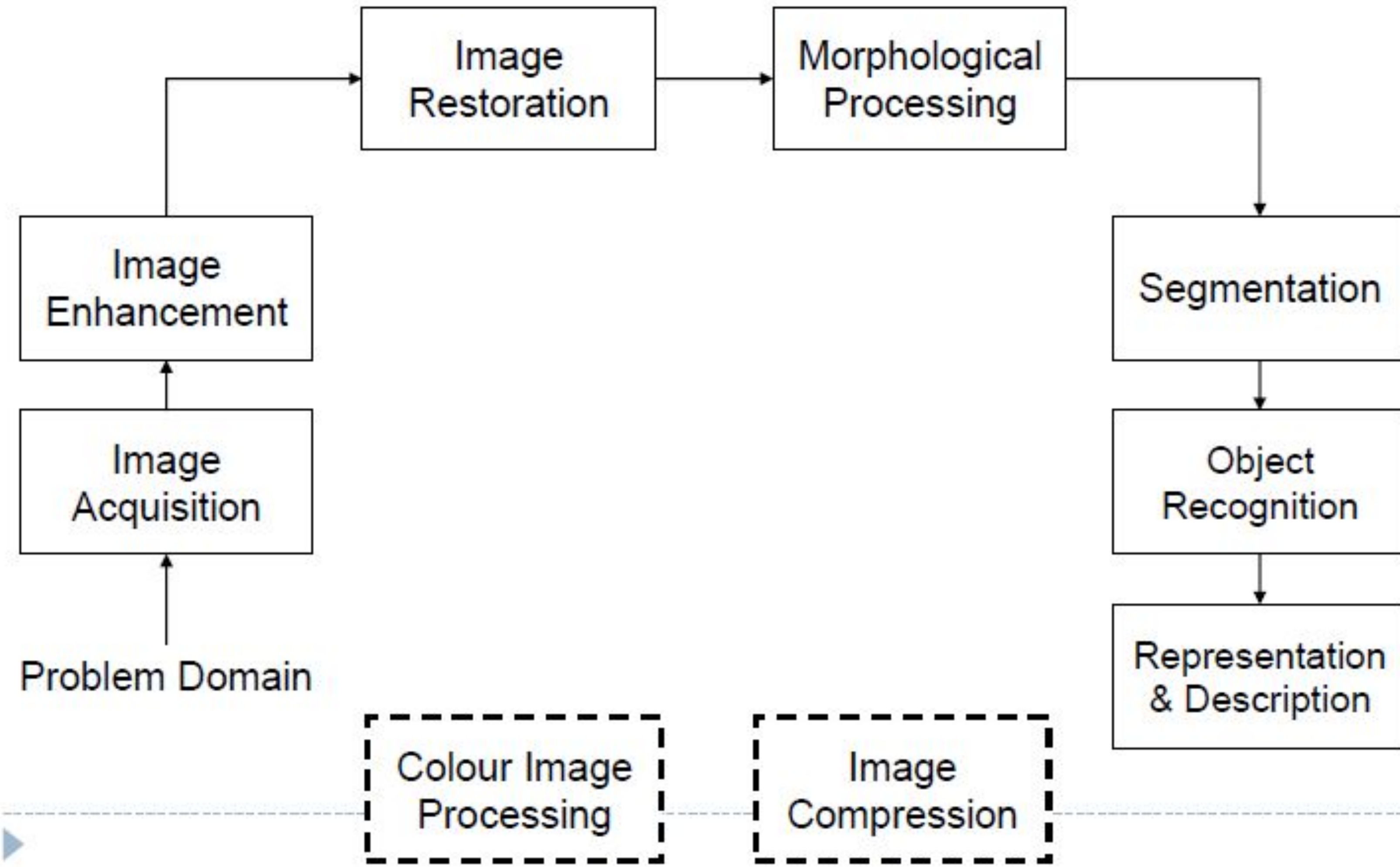
Light

- Light
  - Particles known as photons
  - Act as ‘waves’
- Two fundamental properties
  - Amplitude
  - Wavelength
    - Frequency is the inverse of wavelength
    - Relationship between wavelength (lambda) and frequency (f)

$$\lambda = c / f$$

Where c = speed of light = 299,792,458 m / s

Key Stages in Digital Image Processing



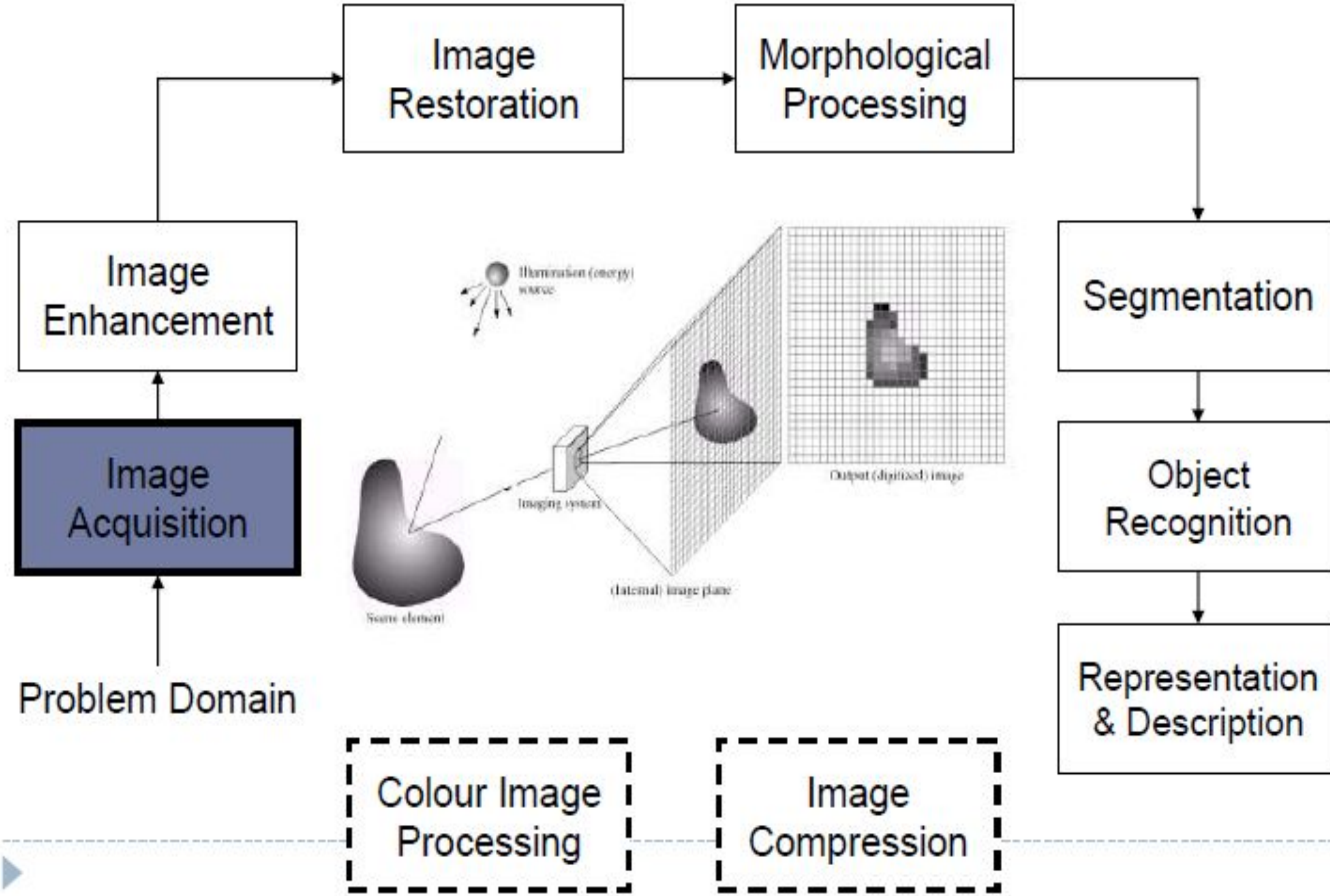


Image Restoration
Morphological Processing
Image Enhancement
Segmentation
Image Acquisition
Object Recognition
Problem Domain
Representation & Description
Colour Image Processing
Image Compression

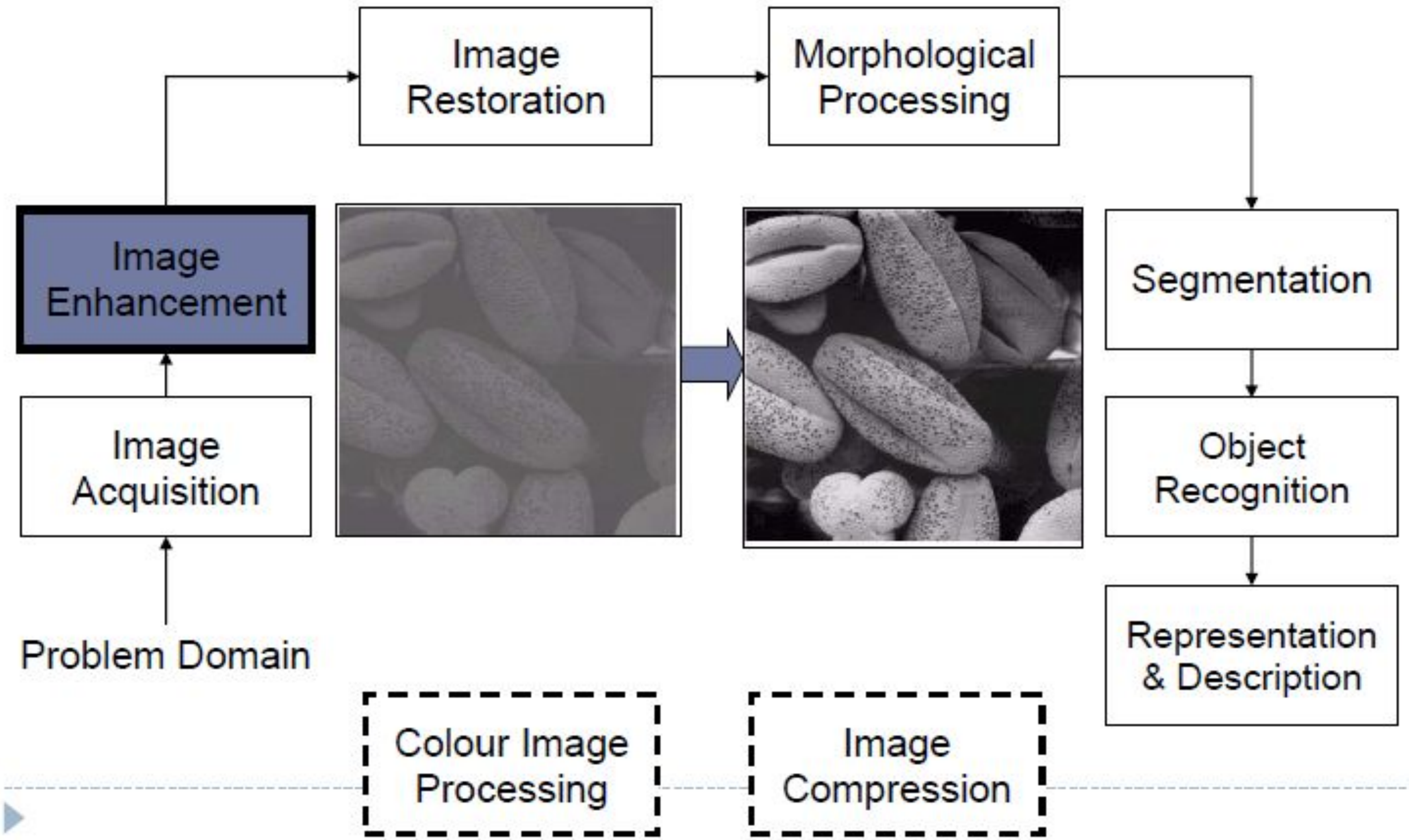
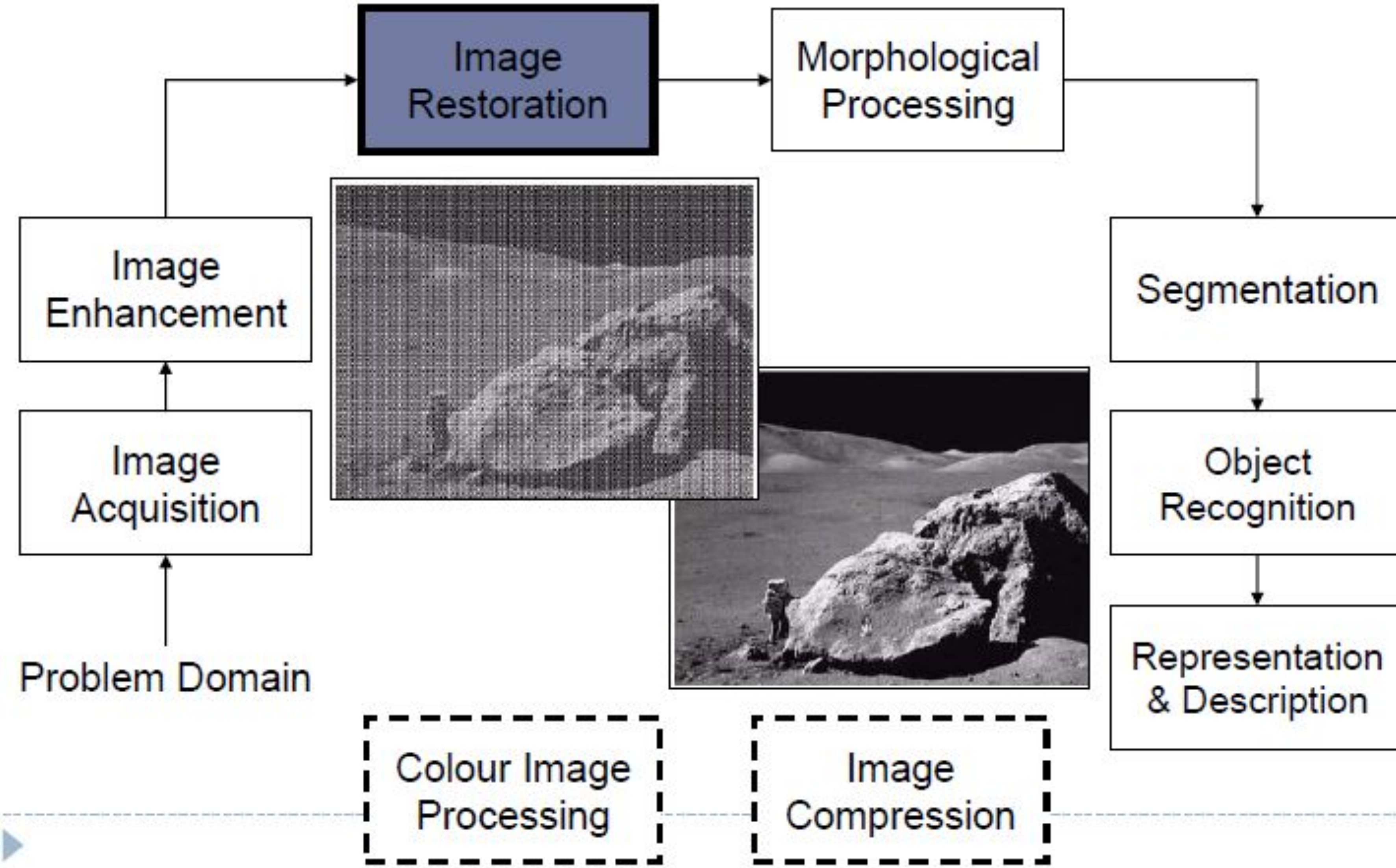
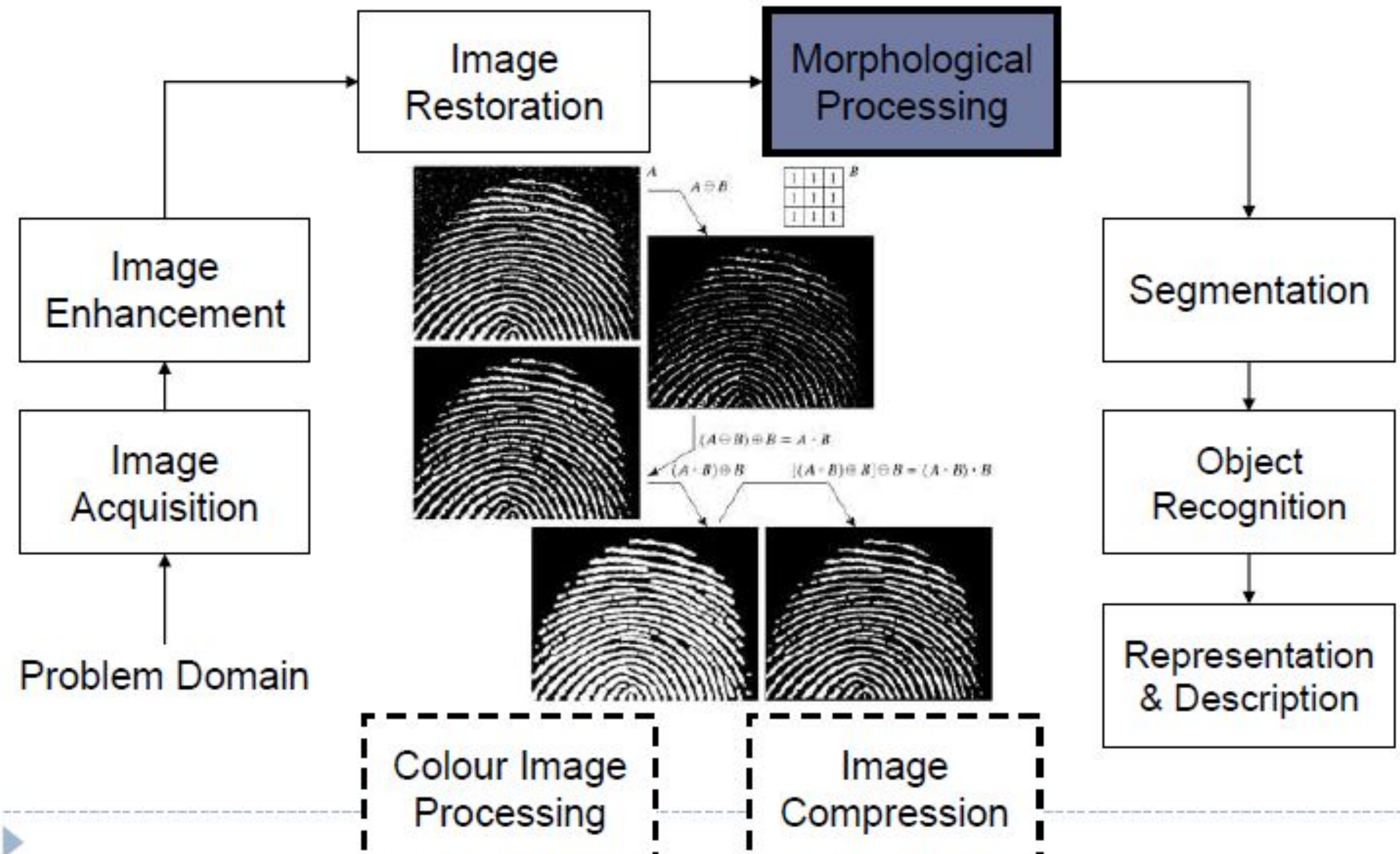


Image Restoration
Morphological Processing
Image Enhancement
Segmentation
Image Acquisition
Object Recognition
Representation & Description
Problem Domain
Colour Image Processing
Image Compression

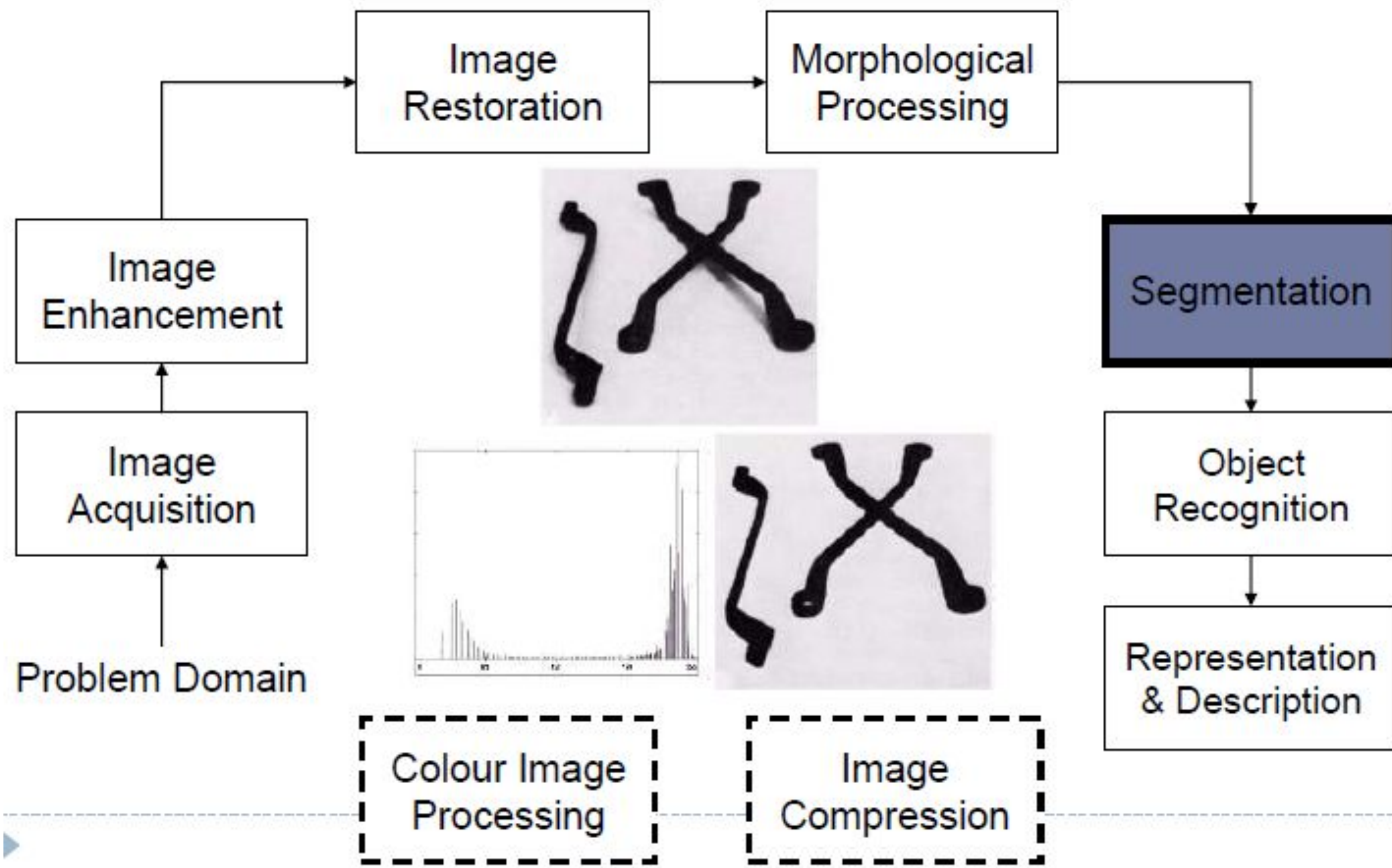
Image Restoration



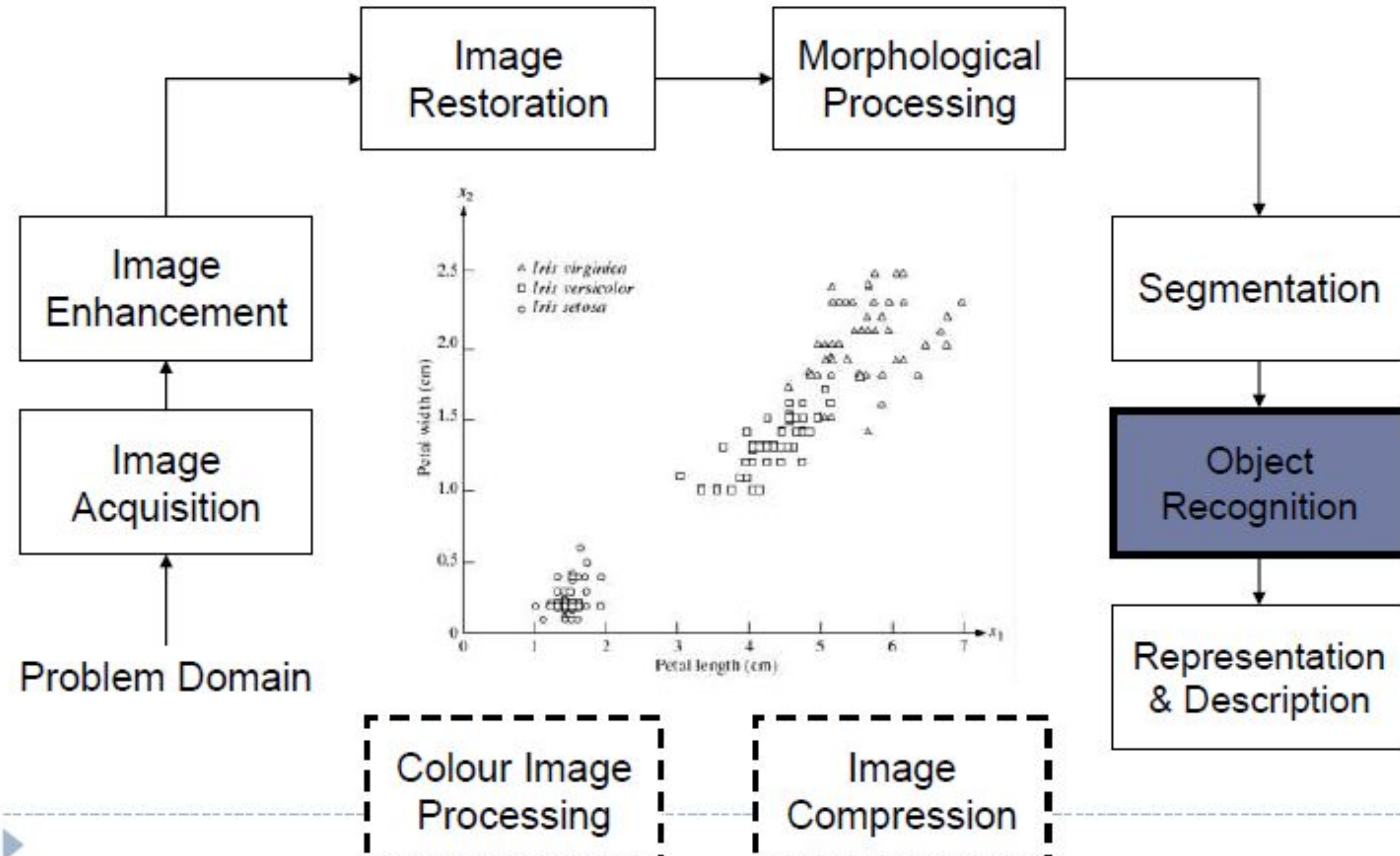
Key Stages in Digital Image Processing: Morphological Processing



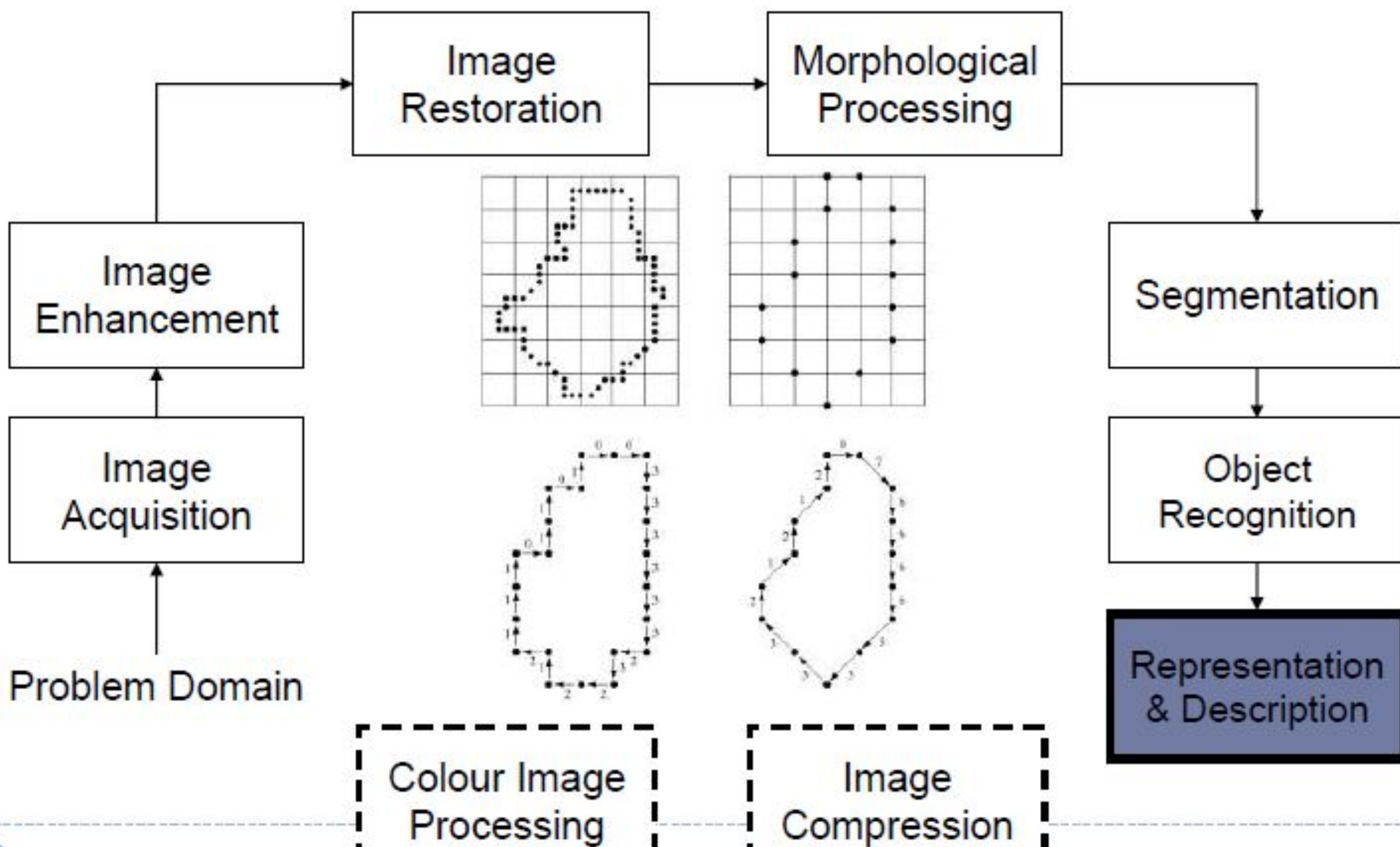
Key Stages in Digital Image Processing: Segmentation



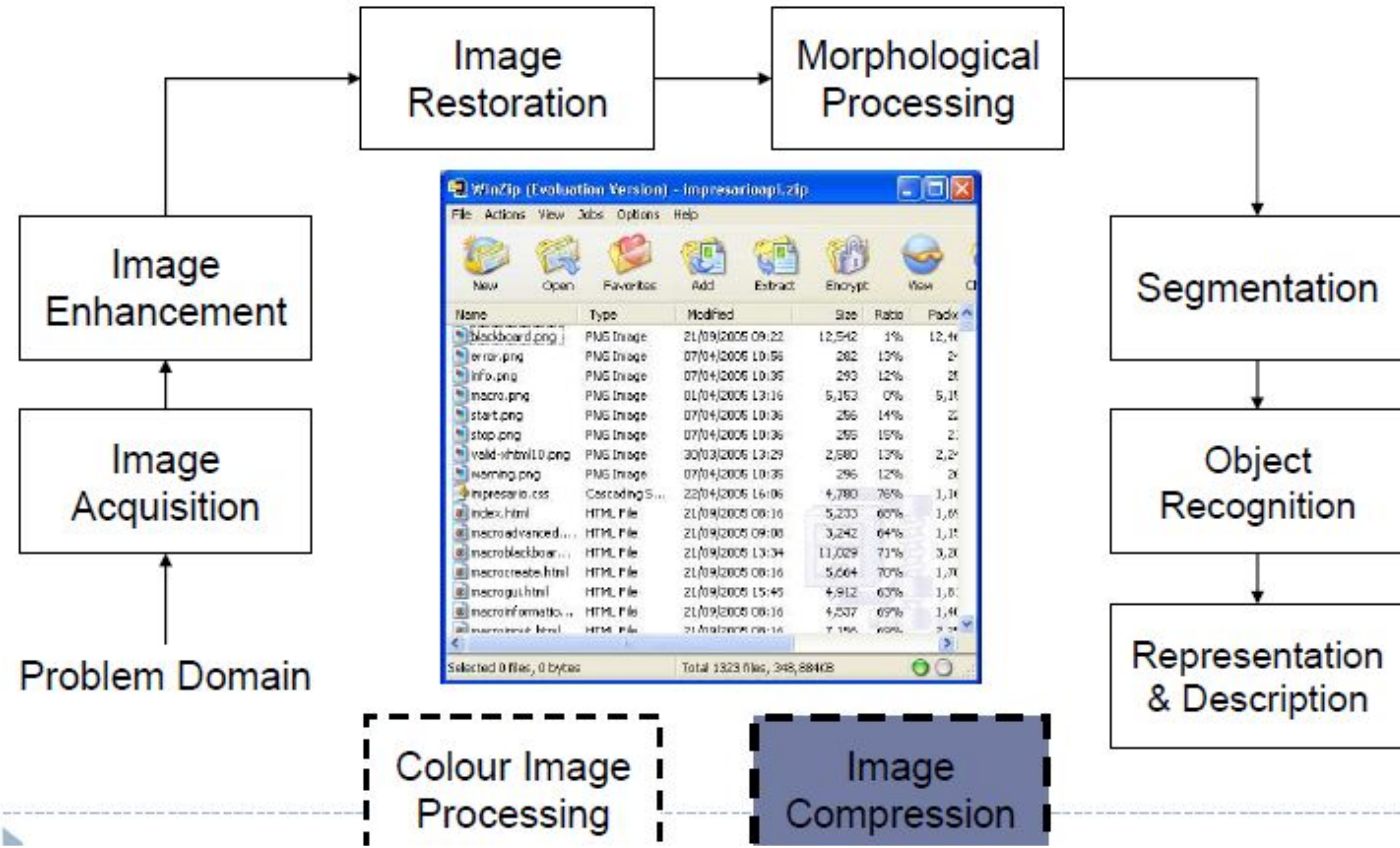
Key Stages in Digital Image Processing: Object Recognition



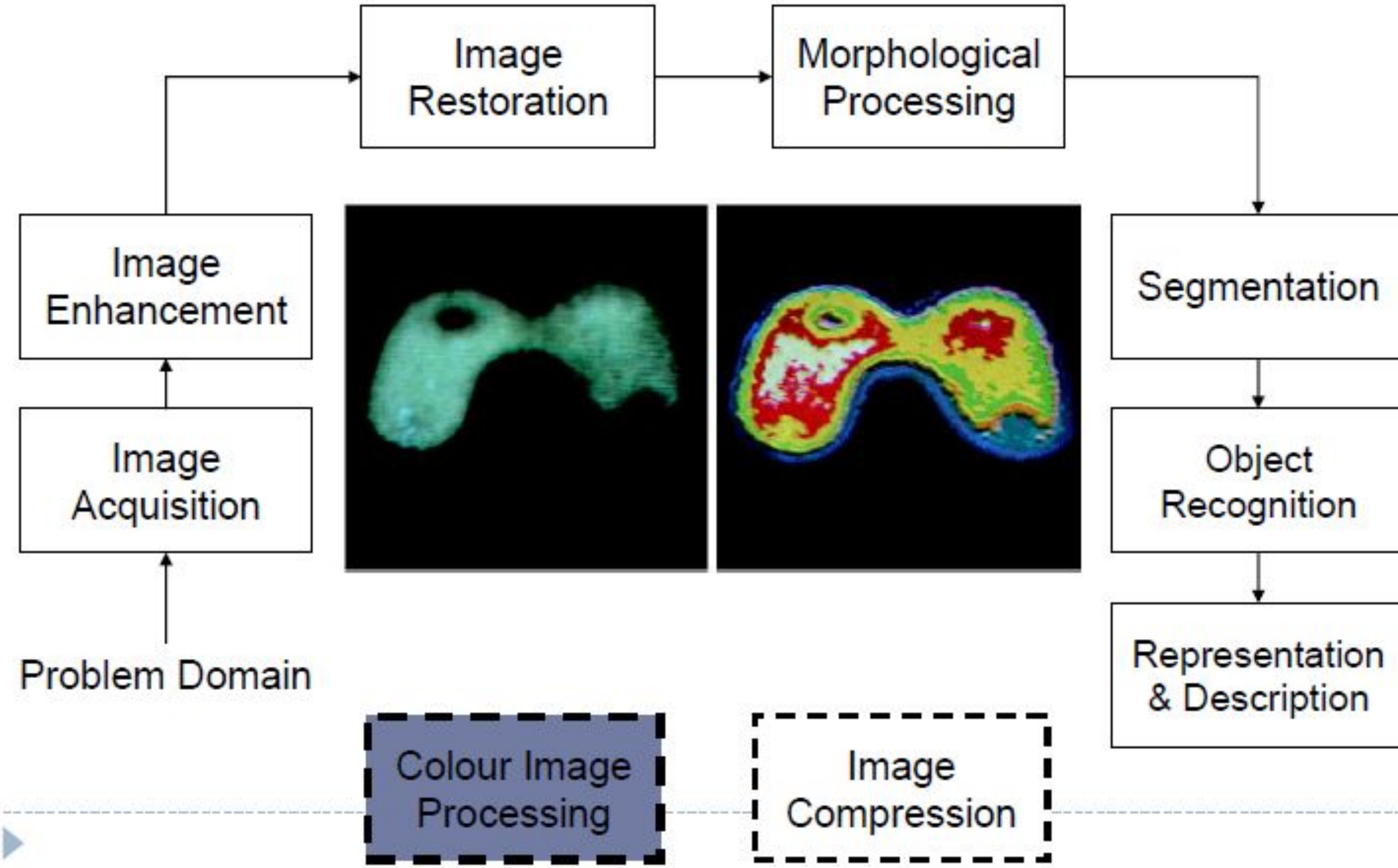
Key Stages in Digital Image Processing: Representation & Description



Key Stages in Digital Image Processing: Image Compression



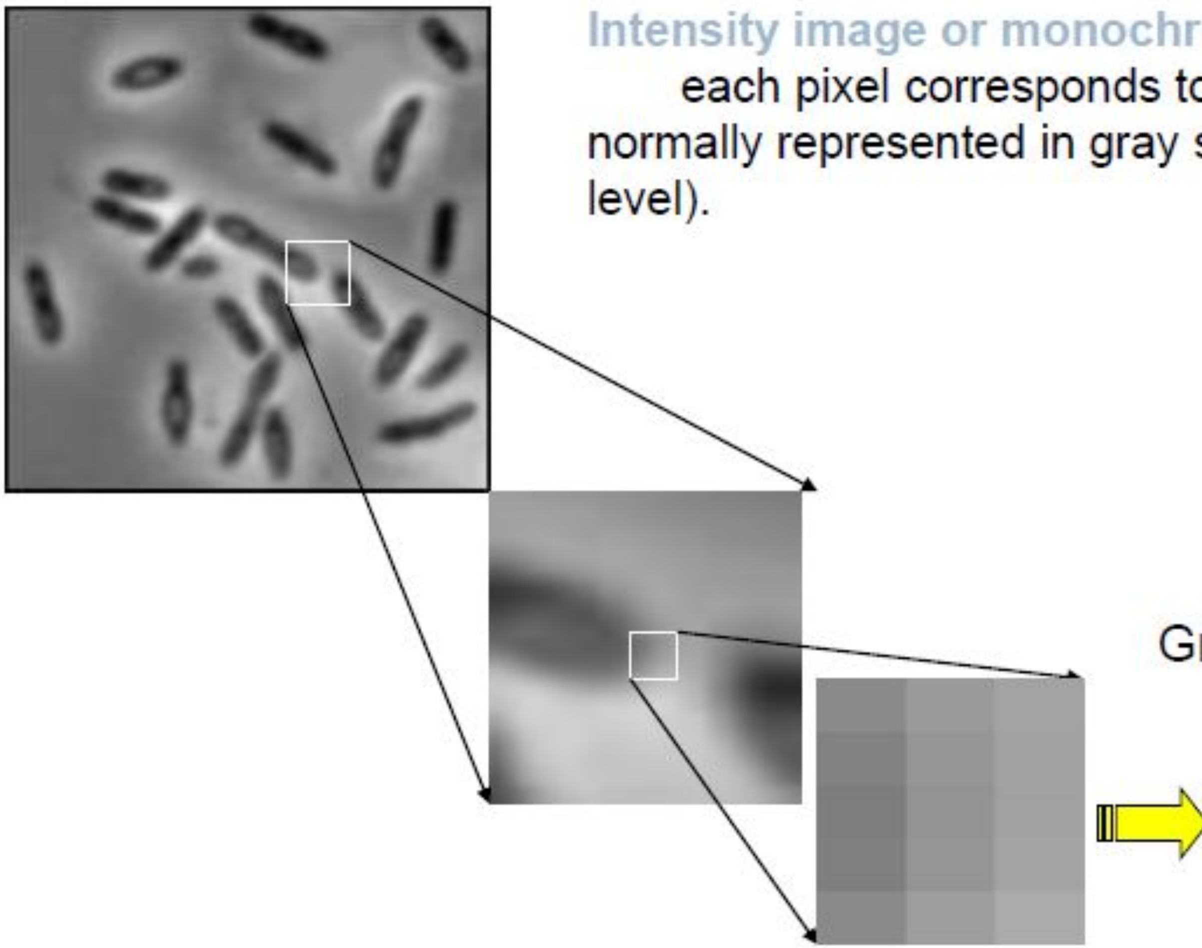
Key Stages in Digital Image Processing: Colour Image Processing



Digital Image Types : Intensity Image

Intensity image or monochrome image

each pixel corresponds to light intensity normally represented in gray scale (gray level).

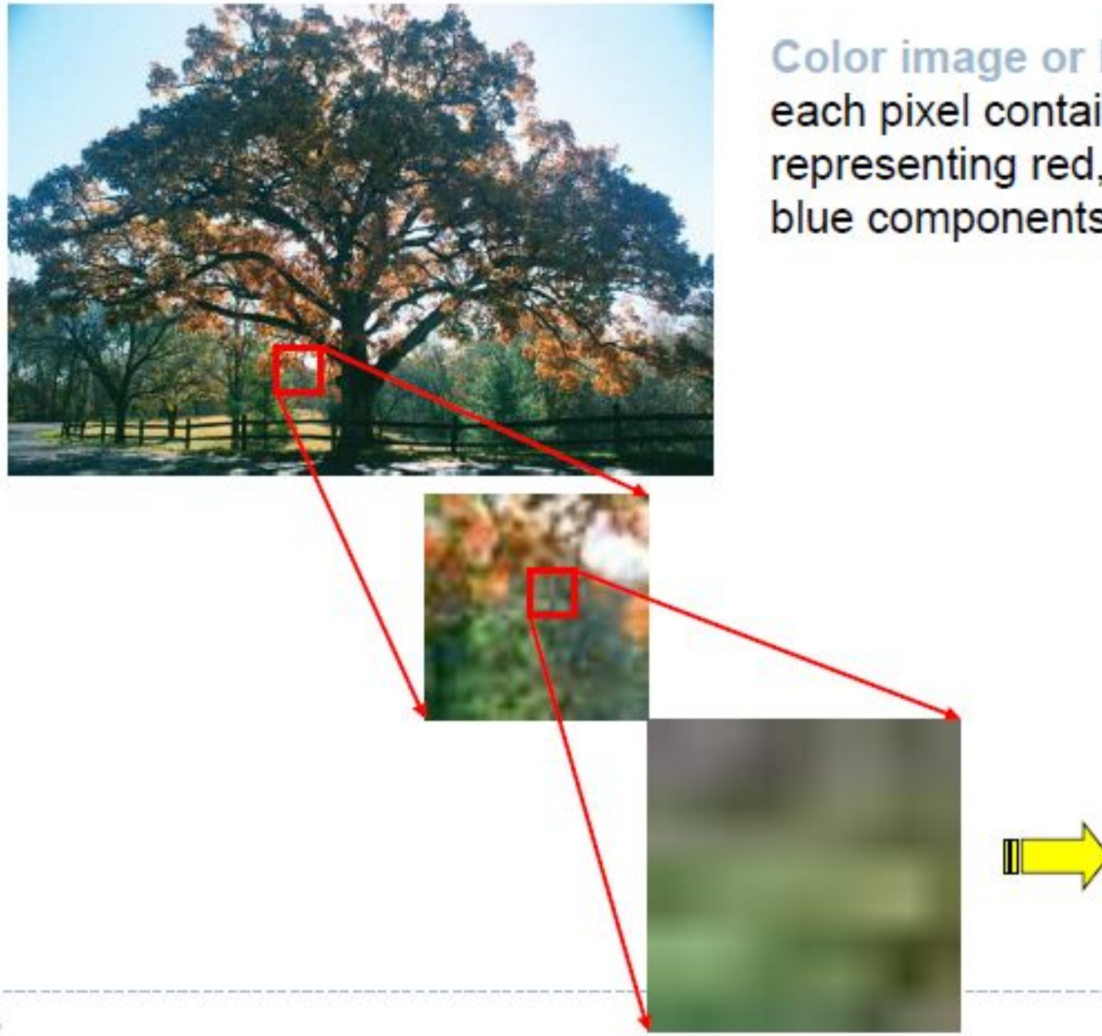


Gray scale values

$$\begin{bmatrix} 10 & 10 & 16 & 28 \\ 9 & 6 & 26 & 37 \\ 15 & 25 & 13 & 22 \\ 32 & 15 & 87 & 39 \end{bmatrix}$$

Digital Image Types : RGB Image

Color image or RGB image: each pixel contains a vector representing red, green and blue components.



RGB components

$$\begin{bmatrix} 99 & 70 & 56 & 78 \\ 60 & 90 & 96 & 67 \\ 85 & 85 & 43 & 92 \\ 32 & 65 & 87 & 99 \end{bmatrix}$$

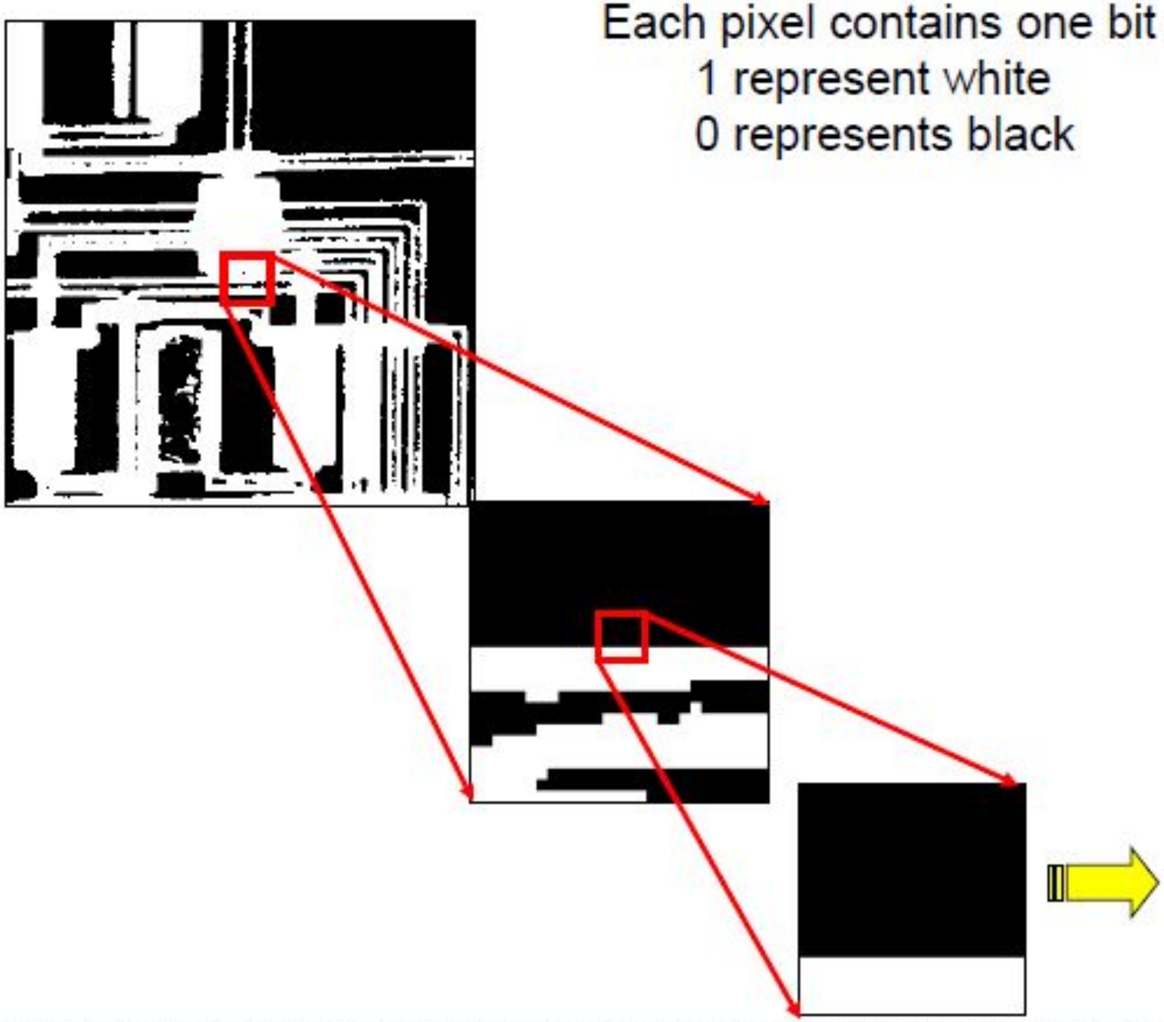
Image Types : Binary Image

Binary image or black and white image

Each pixel contains one bit :

1 represent white

0 represents black



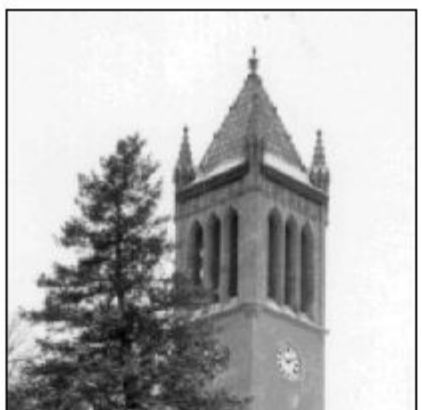
Binary data

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

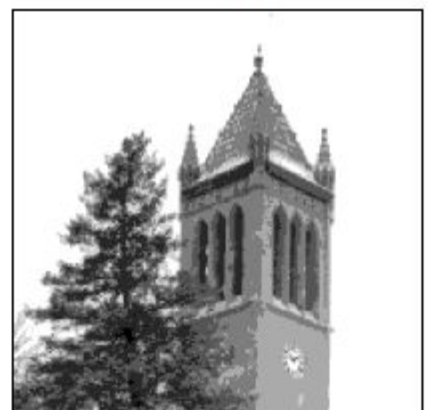
Effect of Quantization Levels or Intensity resolution

256 levels

128 levels



64 levels



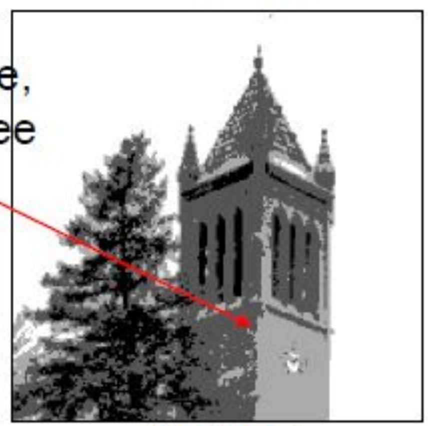
32 levels

Effect of Quantization Levels (cont.) or intensity resolution

16 levels

8 levels

In this image, it is easy to see false contour.



4 levels

2 levels

How to select the suitable size and pixel depth of images

The word "suitable" is subjective: depending on "subject".

Low detail image

Lena image

Medium detail image

Cameraman image

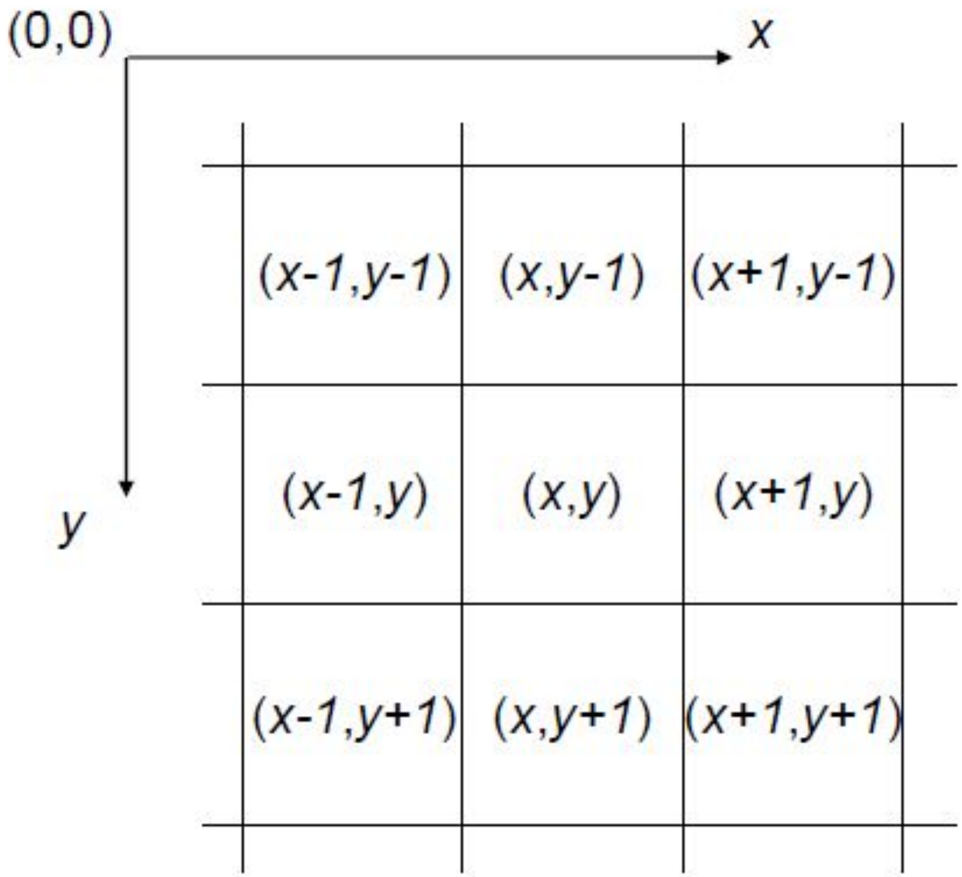
High detail image

To satisfy human mind

1. For images of the same size, the low detail image may need more pixel de
2. As an image size increase, fewer gray levels may be needed.

(Images from Rafael C. Gonzalez and Richard

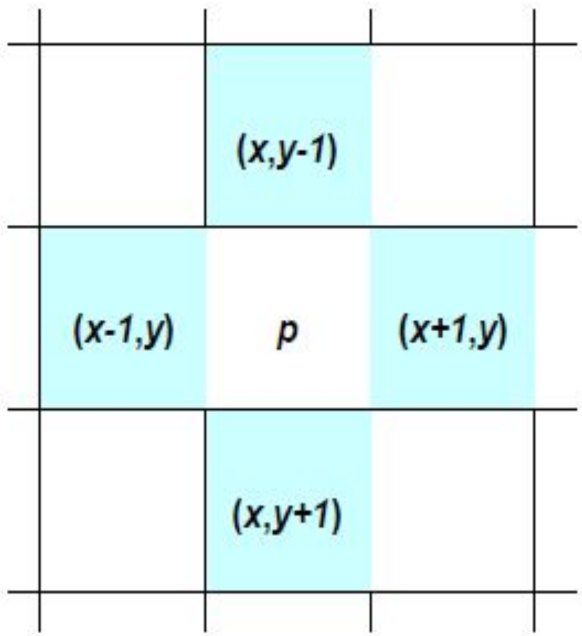
Basic Relationship of Pixels



Conventional indexing method

Neighbors of a Pixel

Neighborhood relation is used to tell adjacent pixels. It is useful for analyzing regions.



4-neighbors of *p*:

$$N_4(p) = \left\{ \begin{array}{c} (x-1,y) \\ (x+1,y) \\ (x,y-1) \\ (x,y+1) \end{array} \right\}$$

4-neighborhood relation considers only vertical and horizontal neighbors.

Note: $q \in N_4(p)$ implies $p \in N_4(q)$

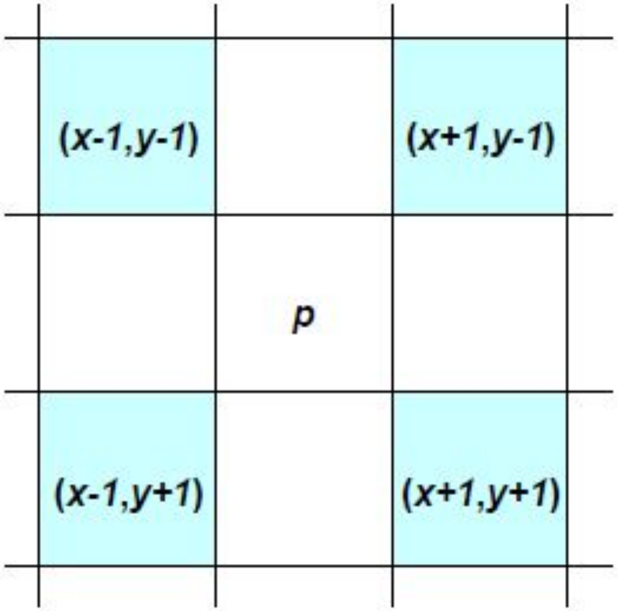
(*x-1,y-1*)	(*x,y-1*)	(*x+1,y-1*)
(*x-1,y*)	*p*	(*x+1,y*)
(*x-1,y+1*)	(*x,y+1*)	(*x+1,y+1*)

8-neighbors of *p*:

$$N_8(p) = \left\{ \begin{array}{c} (x-1,y-1) \\ (x,y-1) \\ (x+1,y-1) \\ (x-1,y) \\ (x+1,y) \\ (x-1,y+1) \\ (x,y+1) \\ (x+1,y+1) \end{array} \right\}$$

8-neighborhood relation considers all neighbor pixels.

Neighbors of a Pixel (cont.)



Diagonal neighbors of *p*:

$$N_D(p) = \left\{ \begin{matrix} (x-1,y-1) \\ (x+1,y-1) \\ (x-1,y+1) \\ (x+1,y+1) \end{matrix} \right\}$$

Diagonal -neighborhood relation considers only diagonal neighbor pixels.

Mathematical Operations on Images

- Image Addition

```
a1 = imread('D:\Matlab\bin\cameraman.jpg'); % Read the image
a2 = imread('D:\Matlab\bin\lena.jpg'); % Read the image
a1=imresize(a1,[255,255]);
a2=imresize(a2,[255,255]);
g=imadd(a1,a2,'uint8');
figure(1)
        subplot(1,3,1),imshow(a1),title('Original Image');
subplot(1,3,2),imshow((a2)),title('Image 2');subplot(1,3,3),imshow(g);
```

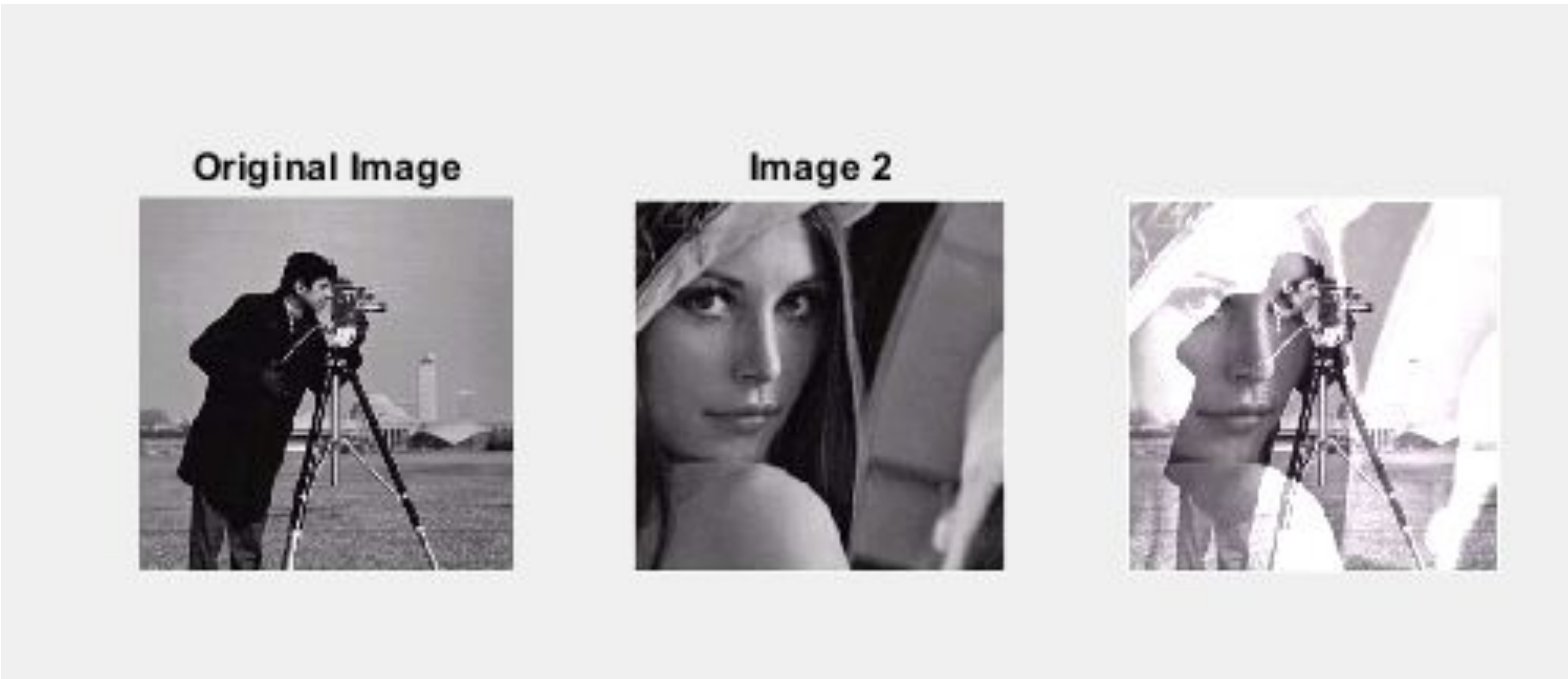

Original Image
Image 2

Image Subtraction

```
a1 = imread('D:\Matlab\bin\cameraman.jpg'); % Read the image
a2 = imread('D:\Matlab\bin\lena.jpg'); % Read the image
a1=imresize(a1,[255,255]);
a2=imresize(a2,[255,255]);
g=imadd(a1,a2,'uint8');
figure(1)
        subplot(1,3,1),imshow(a1),title('Original Image');
subplot(1,3,2),imshow((a2)),title('Image 2');subplot(1,3,3),imshow(g);
a3 = imread('D:\Matlab\bin\cameraman.jpg'); % Read the image
a4 = imread('D:\Matlab\bin\lena.jpg'); % Read the image
a3=imresize(a3,[255,255]);
a4=imresize(a4,[255,255]);
figure(2)
        subplot(1,3,1),imshow(a3),title('Original Image');
subplot(1,3,2),imshow((a4)),title('Image 2');subplot(1,3,3),imshow(z);
```


Original Image
Image 2

Mathematical Operations on Images

- **Image Multiplication**
- $G(x,y)=f1(x,y)*f2(x,y)$
- **Image Division**
- $G(x,y)=f1(x,y)/f2(x,y)$

Image Enhancement

- objective of image enhancement is to improve the interpretability of the information present in images for human viewers.

- enhancement algorithm is one that yields a better-quality image for the purpose of some particular application which can be done by either suppressing the noise or increasing the image contrast.

Spatial Domain Methods

- Image-enhancement techniques can be classified into two broad categories as

(1) spatial domain method

- The spatial domain method operates directly on pixels or raw data.

(2) **transform domain method**

- Transform domain method operates on the Fourier transform of an image and then transforms it back to the spatial domain.

Spatial Domain - Point Processing

1) Image Negative or Digital Negative

- F(x,y) may be an image here F is grey level value(0 to 255).
- 0 black
- 255 white
- Here x,y represents image coordinates

Spatial Domain - Point Processing

- The inverse transformation reverses light and dark.
- Application – X-ray image
- Inverting grey levels.
- $g(m, n) = 255 - f(m, n)$
- Here $g(m,n)$ is negative image
- $f(m,n)$ is original image

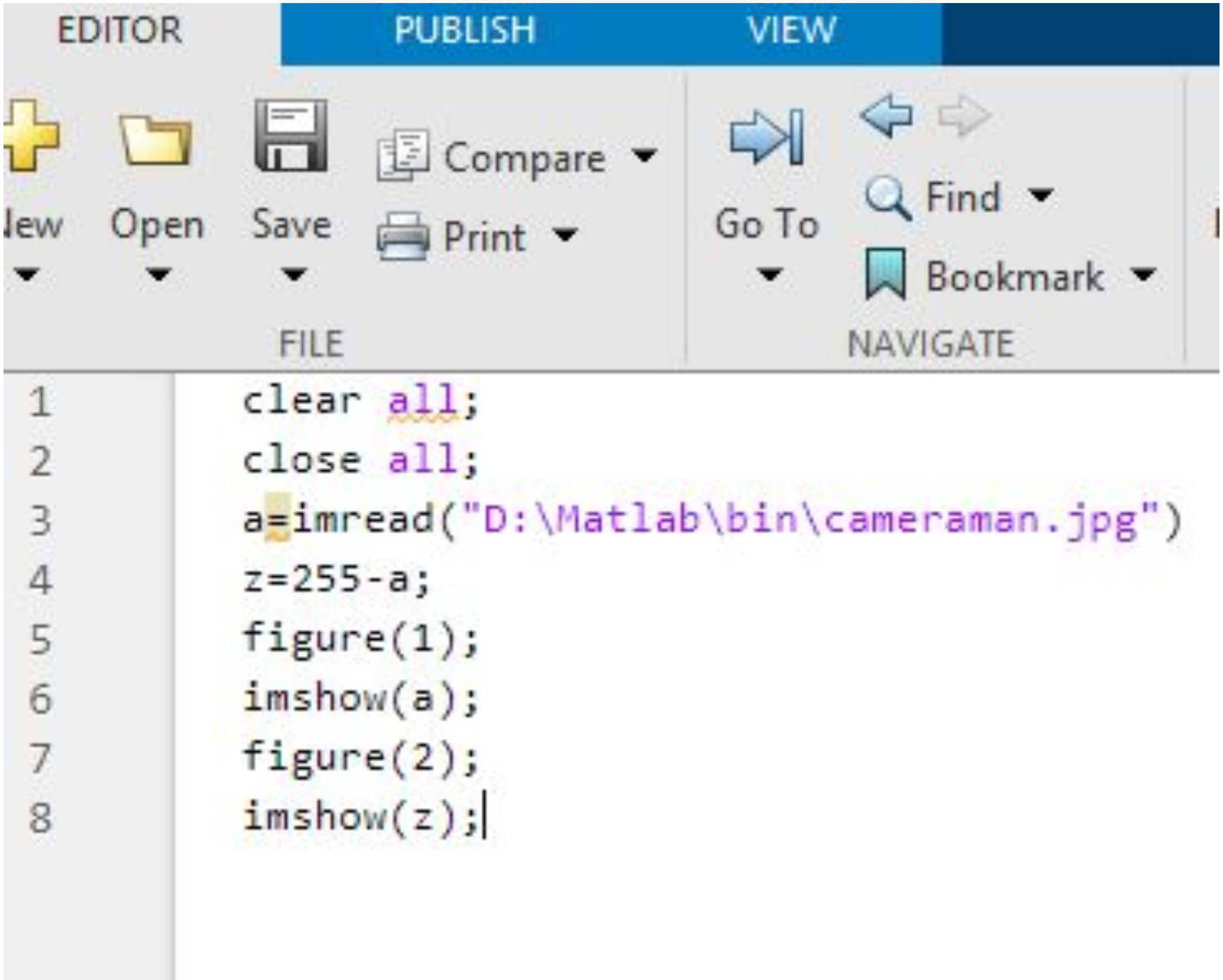

EDITOR
PUBLISH
VIEW
Open
Save
Compare
Print
Go To
Find
Bookmark
FILE
NAVIGATE
clear all;
close all;
a=imread("D:\Matlab\bin\cameraman.jpg")
z=255-a;
figure(1);
imshow(a);
figure(2);
imshow(z);


Figure 1
File
Edit
View
Insert
Tools
Desktop
Window
Help
Figure 2
File
Edit
View
Insert
Tools
Desktop
Window
Help

Spatial Domain - Point Processing

2) Contrast Stretching or Contrast Adjustment

- Poor illumination images are not clear.

- Contrast stretching is increase the contrast of the images by making the dark portions darker and the brighter portions brighter.

- Contrast adjustment is done by scaling all the pixels of the image by a constant k. It is given by
- $g[m, n] = f[m, n] * k$

Spatial Domain - Point Processing

- Specifying a value above 1 will increase the contrast by making bright samples brighter and dark samples darker.

- A value below 1 will do the opposite and reduce a smaller range of sample values.

```
clear all
close all
b=imread("D:\Matlab\bin\butterfly.jpg")
%b=rgb2gray(b);
z=b*0.4;  %Decerease contrast
k=b*40;   %Increase contrast
figure(1)
imshow(b);
figure(2)
imshow(z);
figure(3)
imshow(k);
```

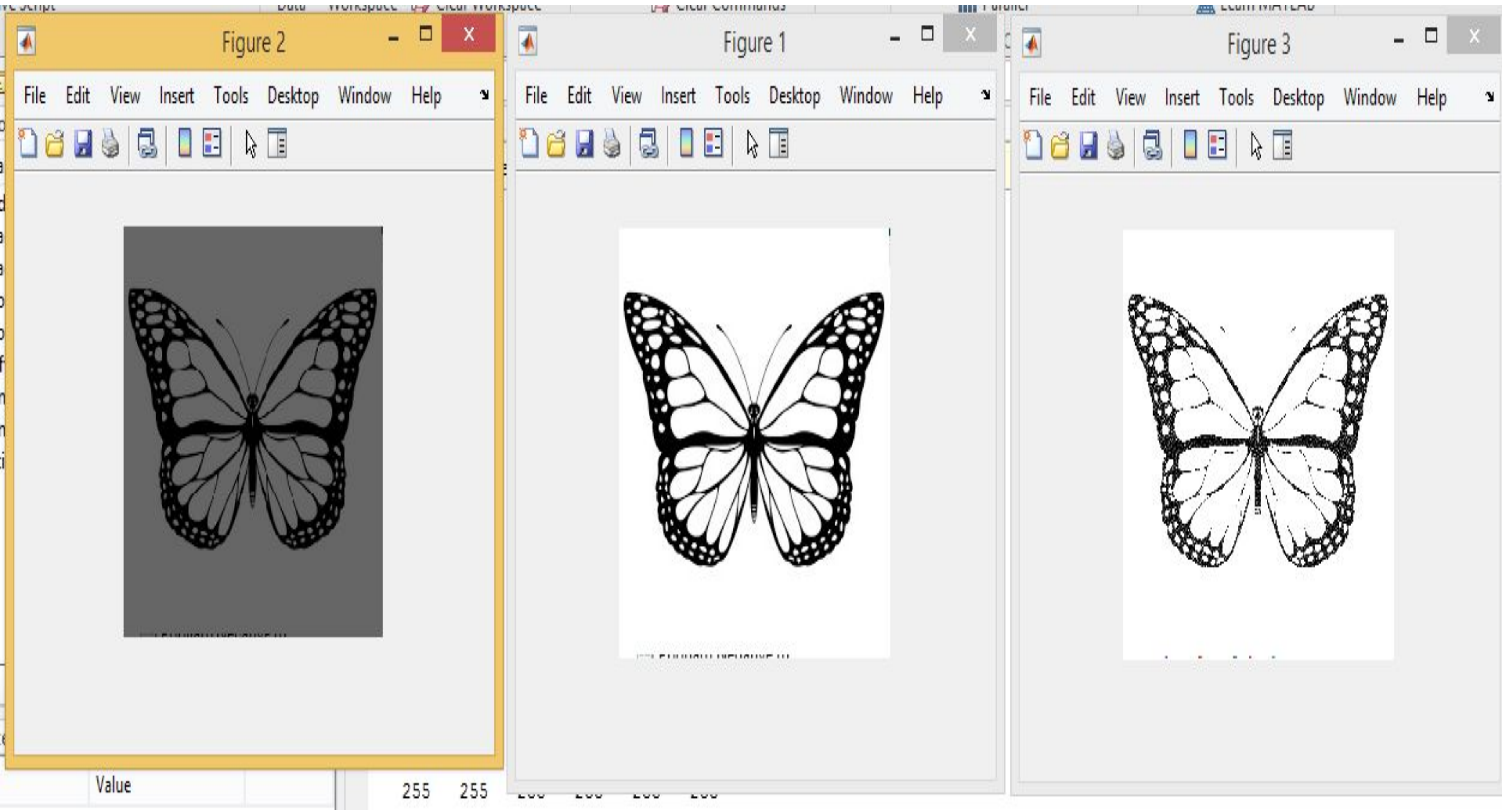


Figure 2
File Edit View Insert Tools Desktop Window Help
Figure 1
File Edit View Insert Tools Desktop Window Help
Figure 3
File Edit View Insert Tools Desktop Window Help
Value
255 255

Spatial Domain - Point Processing

3) Thresholding

- Extreme contrast stretching yields in thresholding.
- S=255; if r(pixel intensity value) >=a (threshold value)

- S=0 if r(pixel intensity value) < a(threshold value)

```
clear all;
close all;
a=imread("D:\Matlab\bin\tumor.jpg");
a=rgb2gray(a);
p=a;
[r c]=size(a)  %returns the size of image a
T=input('Enter the value of Threshold');
for i=1:1:r
    for j=1:1:c
        if(p(i,j)<T)
            a(i,j)=0;
        else
            a(i,j)=255;
        end
    end
end
figure(1)
imshow(a)
figure(2)
imshow(p)
```

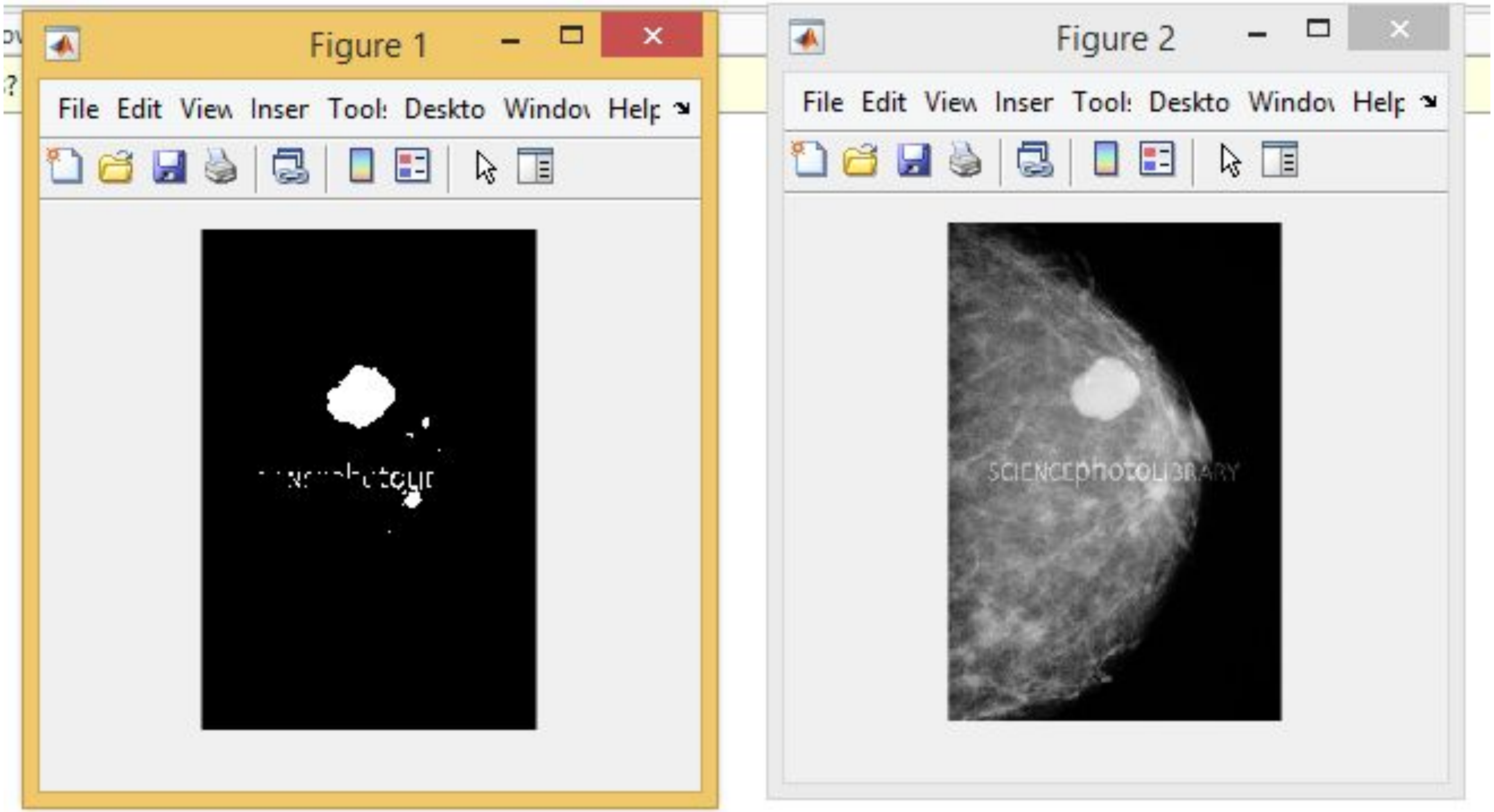


Figure 1
File Edit View Inser Tool: Deskto Windo Help
Figure 2
File Edit View Inser Tool: Deskto Windo Help
SCIENCEphotoLIBRARY

Spatial Domain - Point Processing

4) Grey Level Slicing (Intensity slicing)

- Thresholding splits the grey level into two parts.

- Highlight specific range of grey values we use grey level slicing. Useful for highlighting features in an image

- Band of grey level is selected.

Spatial Domain - Point Processing

- **Grey Level without Background**
- S=255; if a(low threshold value) <=r(pixel intensity value) <= b(high threshold value)
- S=0; otherwise
- display a high value for all gray levels in the range of interest and a low value for all other gray levels.

- **Grey Level with Background**
- S=255; if a(low threshold value) <=r(pixel intensity value) <= b(high threshold value)
- S=r; otherwise
- based on the transformation brightens the desired range of gray levels but preserves gray levels unchanged.

```matlab
clear all;
close all;
a=imread("D:\Matlab\bin\skull.jpg");
a=rgb2gray(a);
p=a;
[r c]=size(a)  %returns the size of image a
T1=input('Enter the value of Threshold T1');
T2=input('Enter the value of Threshold T2');
for i=1:1:r
    for j=1:1:c
        if((p(i,j)>=T1)&&(p(i,j)<=T2)) %without background
            a(i,j)=255;
        else
            a(i,j)=0;
        end
    end
end
figure(1)
imshow(p)
figure(2)
imshow(a)
```

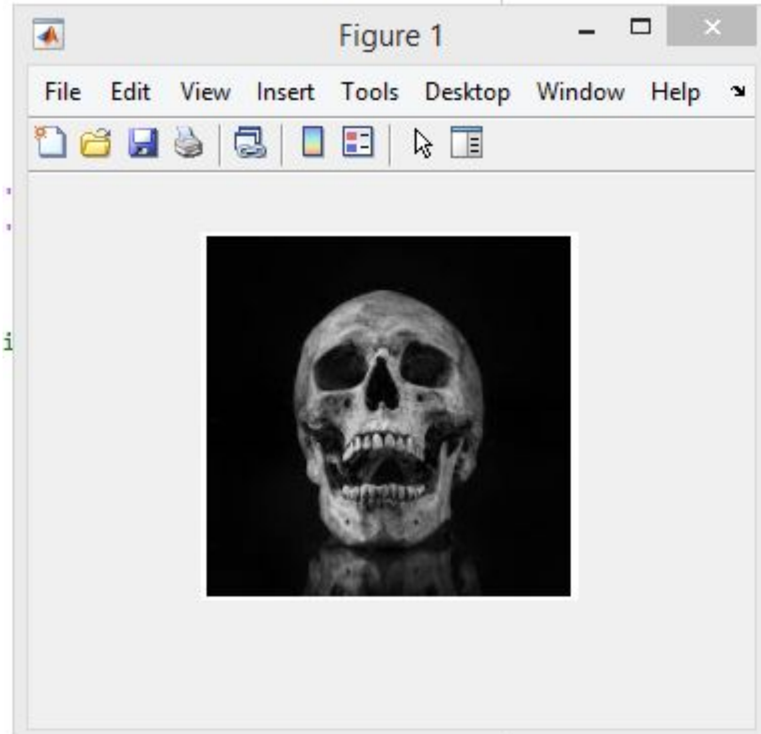


Figure 1
File Edit View Insert Tools Desktop Window Help

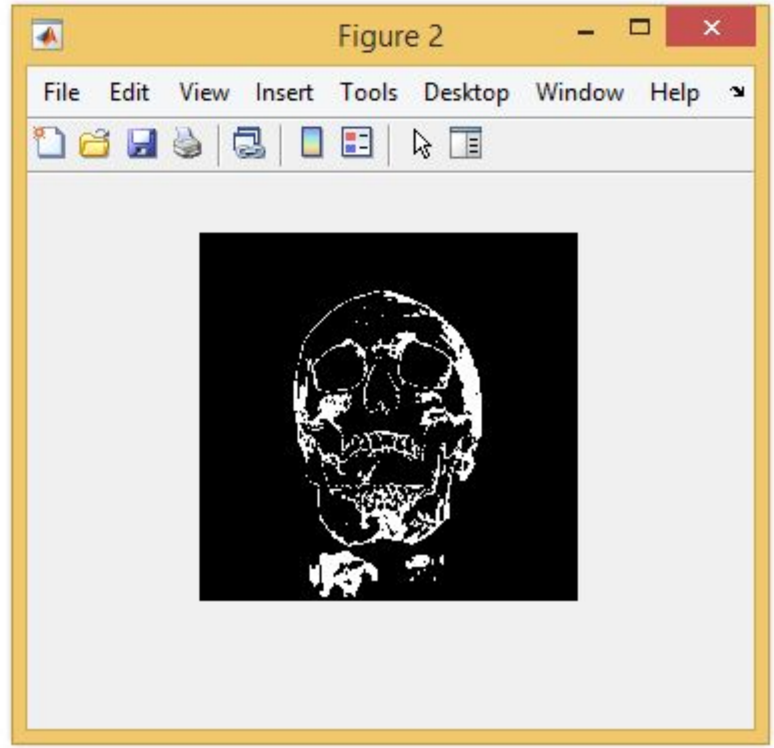


Figure 2
File Edit View Insert Tools Desktop Window Help

```matlab
clear all;
close all;
a=imread("D:\Matlab\bin\skull.jpg");
a=rgb2gray(a);
p=a;
[r c]=size(a)  %returns the size of image a
T1=input('Enter the value of Threshold T1');
T2=input('Enter the value of Threshold T2');
for i=1:1:r
    for j=1:1:c
        if((p(i,j)>=T1)&&(p(i,j)<=T2)) % with background
            a(i,j)=255;
        else
            a(i,j)=p(i,j);
        end
    end
end
figure(1)
imshow(p)
figure(2)
imshow(a)
```

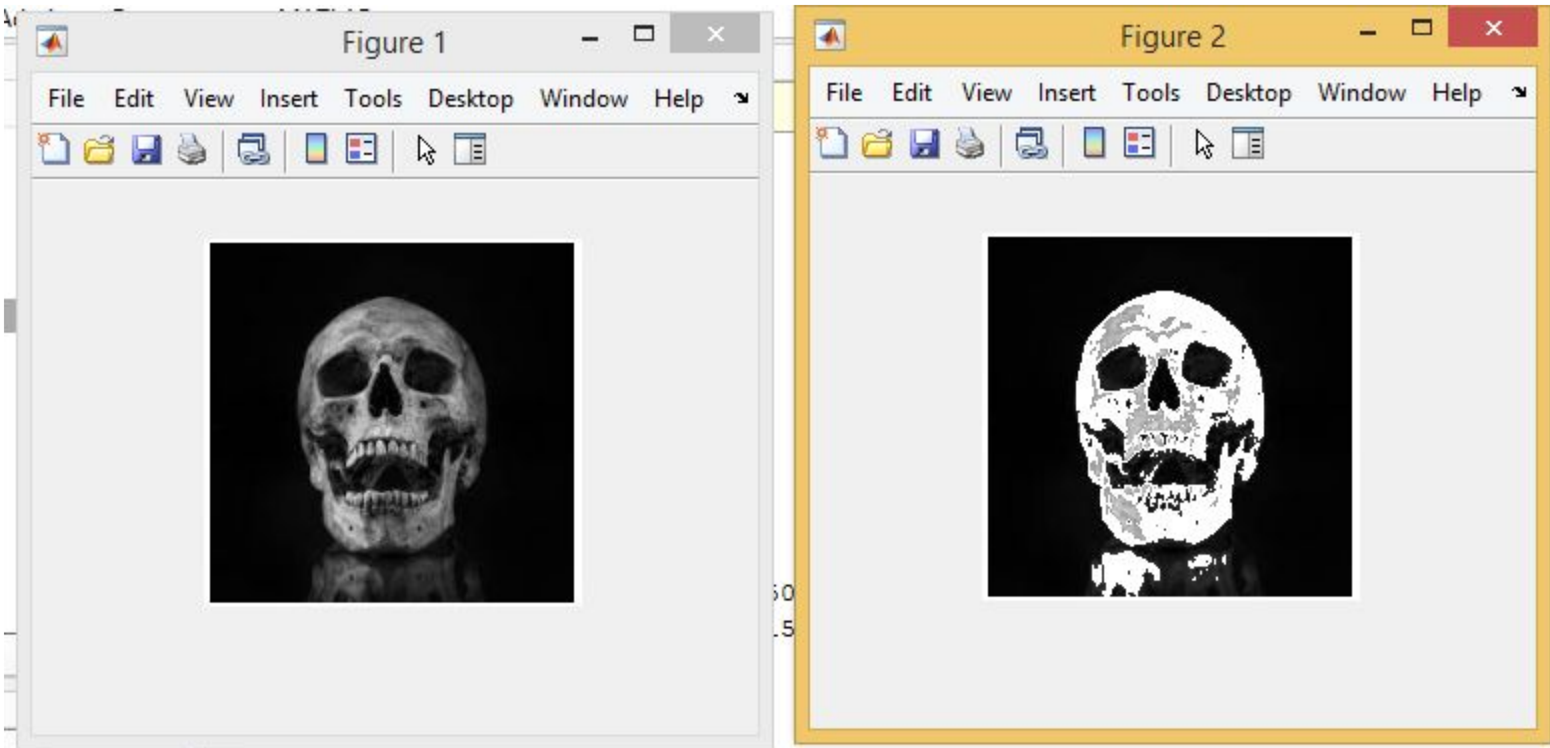


Figure 1
File Edit View Insert Tools Desktop Window Help
Figure 2
File Edit View Insert Tools Desktop Window Help

Spatial Domain - Point Processing

5) Bit Plane Slicing

- Pixels are digital numbers, each one composed of bits. Instead of highlighting gray-level range, we could highlight the contribution made by each bit.

- This method is useful and used in image compression.

Spatial Domain - Point Processing

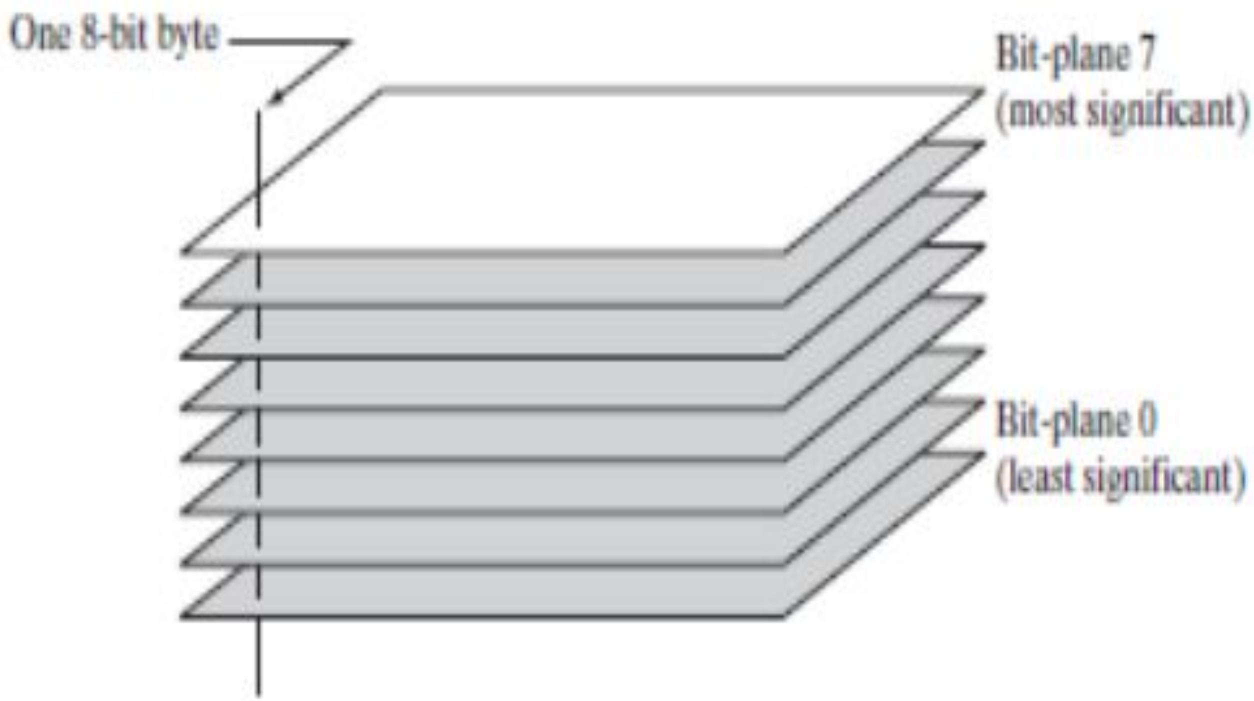
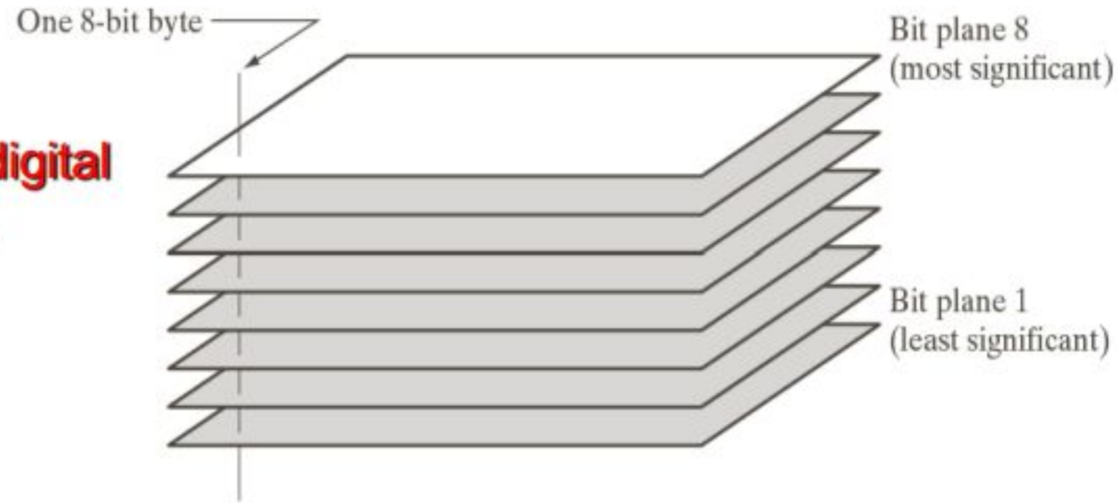


FIGURE 3.12 Bit-plane representation of an 8-bit image.

Remember that pixels are digital numbers composed of bits.



8-bit Image composed of 8 1-bit planes

Spatial Domain - Point Processing

- Often by isolating particular bits of the pixel values in an image we can highlight interesting aspects of that image.

– Higher -order bits usually contain most of the significant visual information.

– Lower -order bits contain subtle details.

Spatial Domain - Point Processing

Ex. 7.1 : Given a 3 × 3 image, plot its bit planes.

1	2	0
4	3	2
7	5	2

Soln. :

Since 7 is the maximum grey level, we need only 3-bits to represent the grey levels. Hence we will have 3-bit planes. Converting the image to binary we get,

001	010	000
100	011	010
111	101	010

Binary image

1	0	0
0	1	0
1	1	0

LSB plane

0	1	0
0	1	1
1	0	1

Middle bit plane

0	0	0
1	0	0
1	1	0

MSB plane

```
clear all;
close all;
clc;
a=imread('D:\Matlab\bin\lena.jpg');
b1=[];
b2=[];
b3=[];
b4=[];
b5=[];
b6=[];
b7=[];
b8=[];
for m=1:256
    for n=1:256
        t=de2bi(a(m,n),8,'left-msb');
        b1(m,n)=t(1,1);
        b2(m,n)=t(1,2);
        b3(m,n)=t(1,3);
        b4(m,n)=t(1,4);
        b5(m,n)=t(1,5);
        b6(m,n)=t(1,6);
        b7(m,n)=t(1,7);
        b8(m,n)=t(1,8);
    end
end

subplot(3,3,1);
imshow(a);
title('image of cameramen','color','r');
```

```
subplot(3,3,1);
imshow(a);
title('image of cameramen','color','r');

subplot(3,3,2);
imshow(b8);
title('image of bit-1','color','r');
subplot(3,3,3);
imshow(b7);
title('image of bit-2','color','r');
subplot(3,3,4);
imshow(b6);
title('image of bit-3','color','r');
subplot(3,3,5);
imshow(b5);
title('image of bit-4','color','r');
subplot(3,3,6);
imshow(b4);
title('image of bit-5','color','r');

subplot(3,3,7);
imshow(b3);
title('image of bit-6','color','r');

subplot(3,3,8);
imshow(b2);
title('image of bit-7','color','r');
```

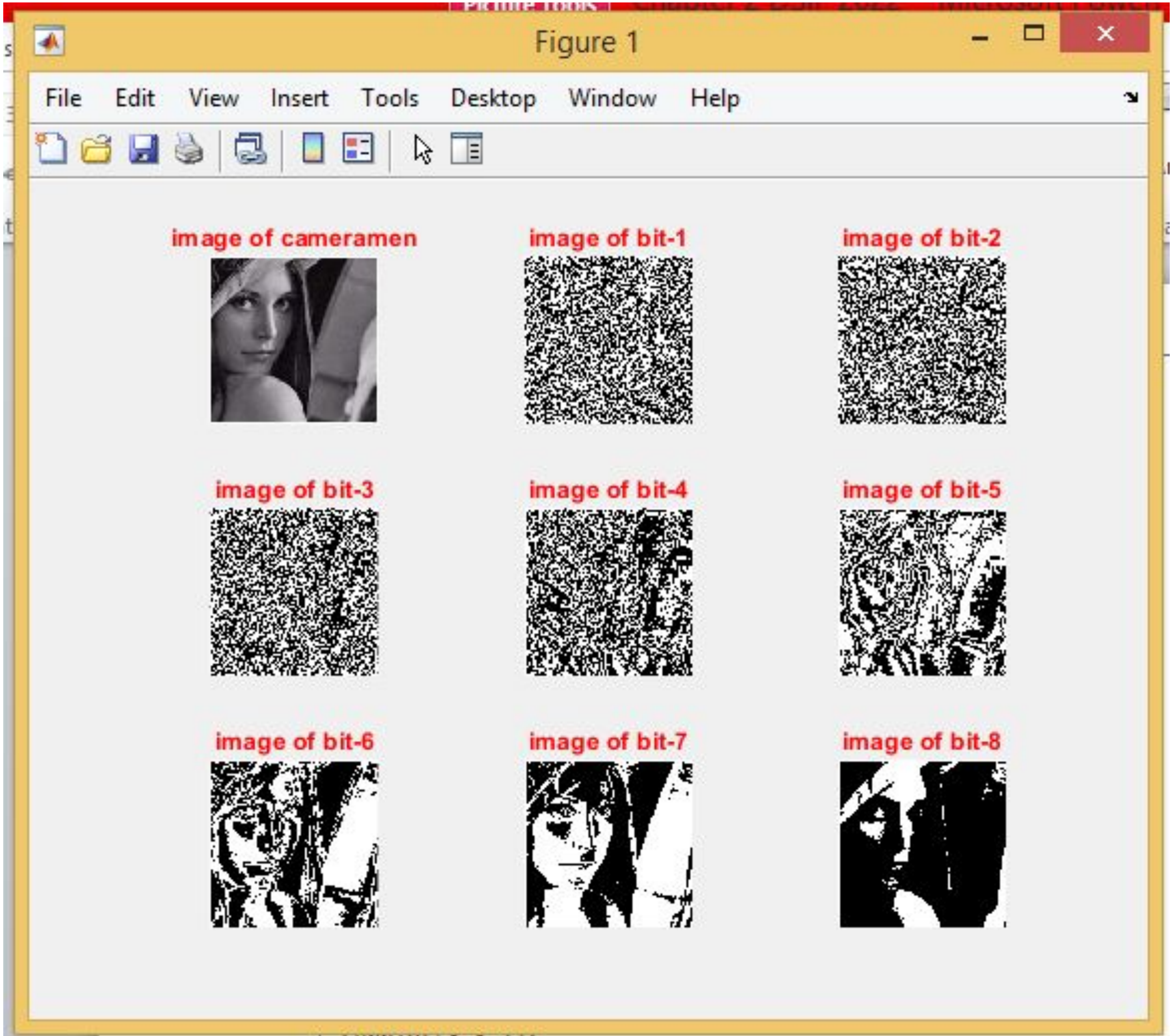


Figure 1
File Edit View Insert Tools Desktop Window Help
image of cameramen
image of bit-1
image of bit-2
image of bit-3
image of bit-4
image of bit-5
image of bit-6
image of bit-7
image of bit-8

Spatial Domain - Point Processing

6) Dynamic Range Compression (Log Transformation)

- Dynamic range of the image exceeds the capability of the display device.
- Some pixel values are so high that low pixel values get over shadowed.
- To be able to see the small value pixels dynamic range compression is used.

Spatial Domain - Point Processing

- Log operator being the excellent compressing function.
- Dynamic range compression is achieved by using log operator.
- Log Transformation = C*log(1+|a|)…(a=image)

```
a1 = imread('D:\Matlab\bin\saturn.jpg'); % Read the image
a = double(a1)/256; % Normalized Image
c = 2; % Constant
f = c*log(1 + (a)); % Log Transform
subplot(1,2,1),imshow(a1),title('Original Image');
subplot(1,2,2),imshow((f)),title('Log Transformation Image');
```

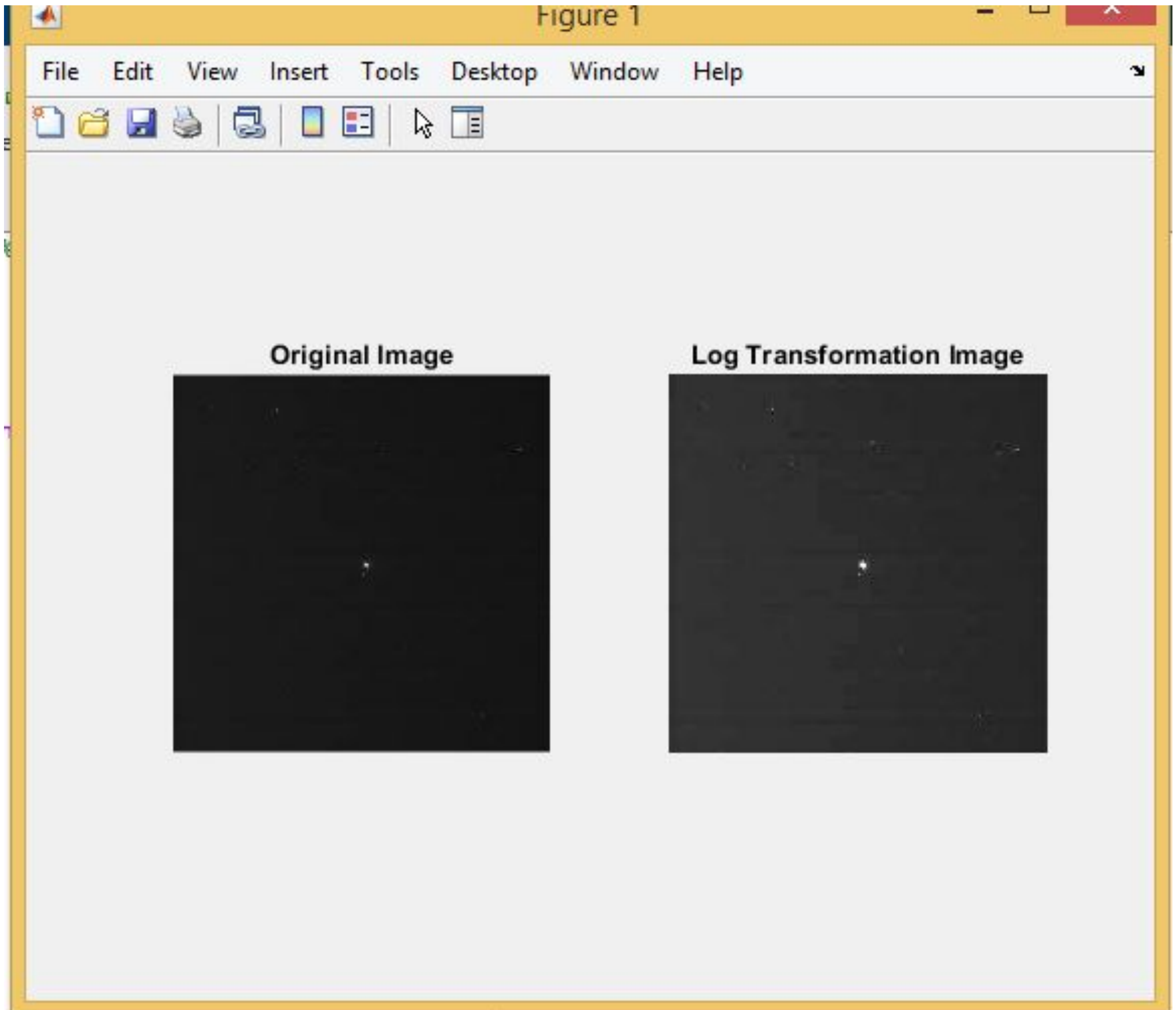
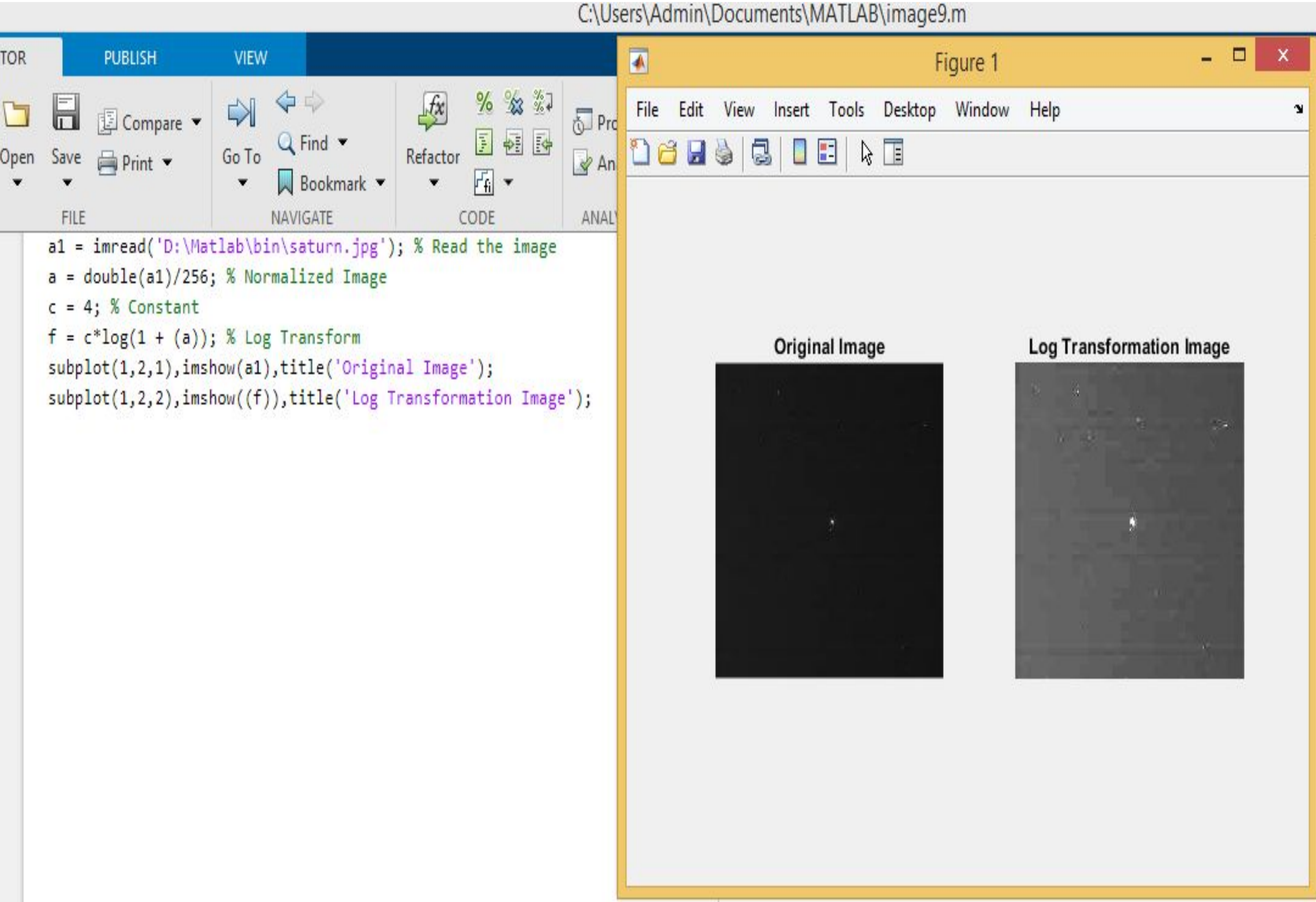
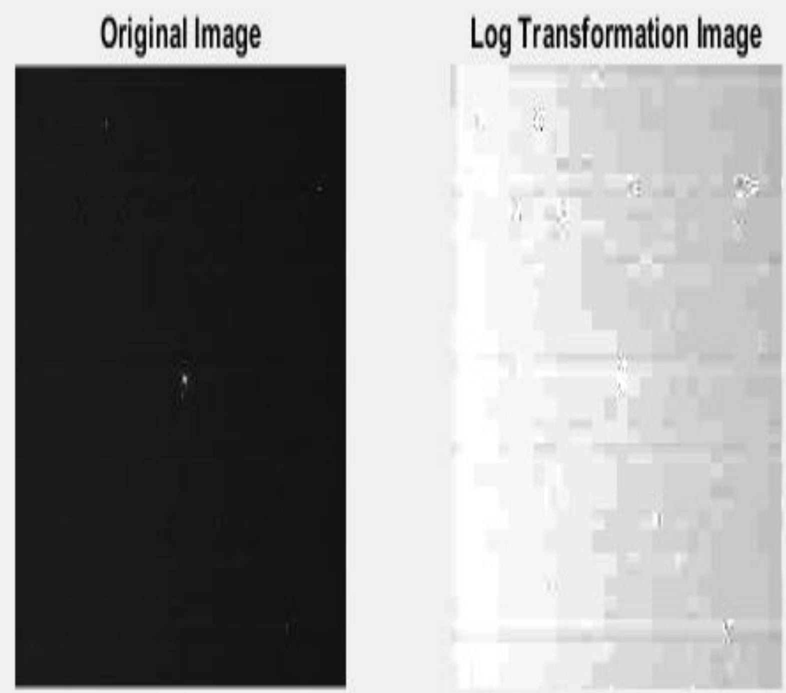


Figure 1
File Edit View Insert Tools Desktop Window Help
Original Image
Log Transformation Image


C:\Users\Admin\Documents\MATLAB\image9.m
TOR
PUBLISH
VIEW
Open
Save
Compare
Print
FILE
Go To
Find
Bookmark
NAVIGATE
Refactor
CODE
ANAL
a1 = imread('D:\Matlab\bin\saturn.jpg'); % Read the image
a = double(a1)/256; % Normalized Image
c = 4; % Constant
f = c*log(1 + (a)); % Log Transform
subplot(1,2,1),imshow(a1),title('Original Image');
subplot(1,2,2),imshow((f)),title('Log Transformation Image');
Figure 1
File Edit View Insert Tools Desktop Window Help
Original Image
Log Transformation Image

```
a1 = imread('D:\Matlab\bin\saturn.jpg'); % Read the image
a = double(a1)/256; % Normalized Image
c = 10; % Constant
f = c*log(1 + (a)); % Log Transform
subplot(1,2,1),imshow(a1),title('Original Image');
subplot(1,2,2),imshow((f)),title('Log Transformation Image');
```



Spatial Domain - Point Processing

7) Power Law Transformation

- human perception of brightness - more sensitive to changes in dark as compared to bright.
- display devices like computer screen have Intensity to voltage (non linear) - which is a power function with exponents(Gamma) varying from 1.8 to 2.5.
- any input signal(say from a camera), the output will be transformed by gamma because of non-linear intensity to voltage relationship of the display screen - This results in images that are darker than intended.

Spatial Domain - Point Processing

$$s = c*r^{\gamma}$$

- we apply gamma correction to the input signal.
- This input cancels out the effects generated by the display and we see the image as it is.

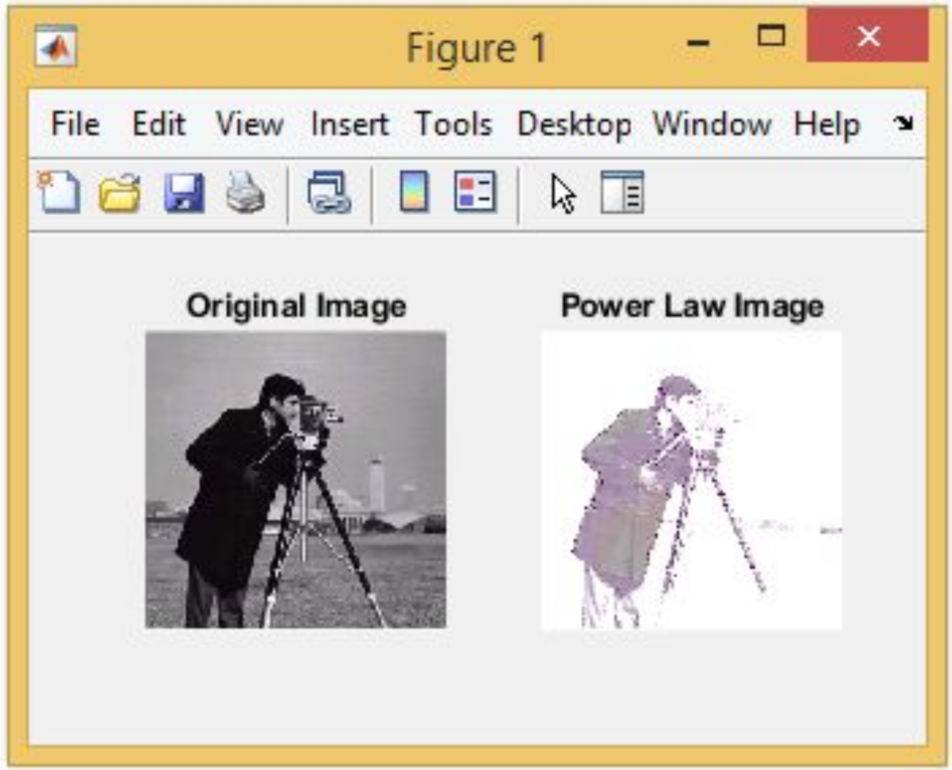
```
a1 = imread('D:\Matlab\bin\cameraman.jpg'); % Read the image
a = double(a1)/256; % Normalized Image
gamma=input('Enter the correction factor gamma value:')
c=2;
f = c*a.^gamma;
subplot(1,2,1),imshow(a1),title('Original Image');
subplot(1,2,2),imshow((f)),title('Power Law Image');
```

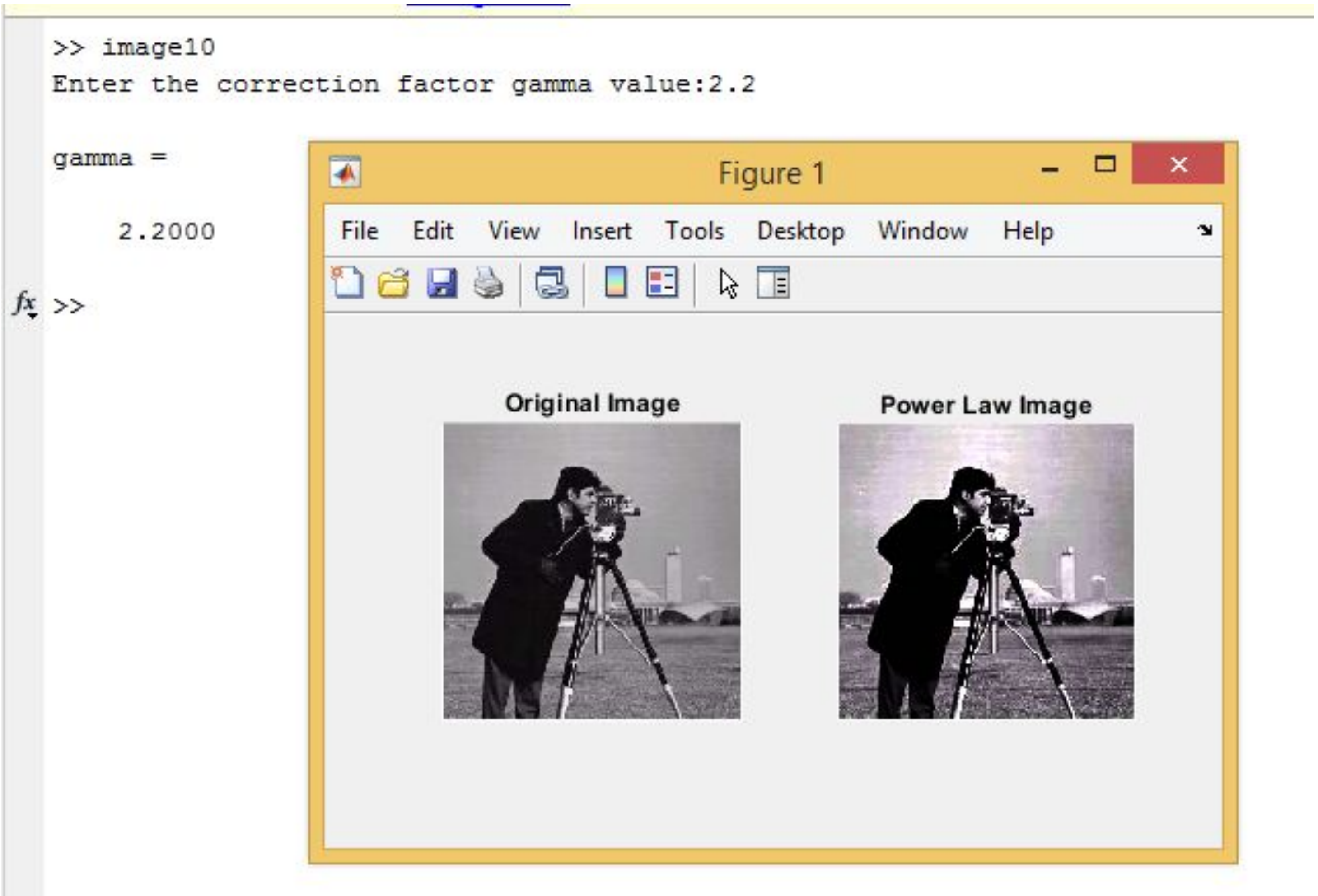
```
>> image10
Enter the correction factor gamma value:0.4

gamma =

    0.4000

>>
```




>> image10
Enter the correction factor gamma value:2.2
gamma =
2.2000
>>
Figure 1
File Edit View Insert Tools Desktop Window Help
Original Image
Power Law Image

Feature	Power Law Transformation	Contrast Stretching
Transformation Type	Nonlinear (depends on the exponent γ\gammaγ)	Linear (stretches pixel values over a specified range)
Effect on Image	Adjusts **brightness** and **local contrast** based on intensity	Increases **global contrast** by expanding pixel values
Parameter	Gamma (γ\gamma γ) determines the nonlinearity	Minimum and maximum values determine contrast enhancement
Pixel Range	Does not necessarily change the pixel value range, but alters the distribution	Expands the pixel range from input min/max to desired output min/max
Visual Impact	Changes pixel intensity to brighten/darken based on the exponent	Increases overall contrast, making dark areas darker and bright areas brighter
Use Case	Correcting brightness issues (e.g., for display calibration)	Improving visibility in images with low contrast

Neighbourhood Processing

- In neighbourhood operation, the pixels in an image are modified based on some function of the pixels in their neighbourhood.

- Instead of 3*3 neighbourhood , we could also use 5*5 or a 7*7 …….

Low Pass Averaging Filter (Box Filter)

- If an image has Gaussian noise present in it a low pass averaging filter is used to eliminate it.
- The mean filter replaces each pixel by the average of all the values in the local neighbourhood.

- The size of the neighbourhood controls the amount of filtering.

Low Pass Averaging Filter

- spatial averaging operation, each pixel is replaced by a weighted average of its neighbourhood pixels.

- The low-pass filter preserves the smooth region in the image and it removes the sharp variations leading to blurring effect.

Low Pass Averaging Filter

$$\text{3 by 3 low-pass spatial mask} = \frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\text{Similarly, the 5 by 5 averaging mask} = \frac{1}{25} \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Low Pass Averaging Filter

- It is to be noted that the sum of the elements is equal to 1 in the case of a low-pass spatial mask.

- The blurring effect will be more with the increase in the size of the mask.

- the size of the mask will be odd so that the central pixel can be located exactly.

→ Low Pass Averaging Filter

Given Image :-

8×8 Image

$$\begin{bmatrix} 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\ 50 & 50 & 50 & 50 & 50 & 50 & 50 & 50 \\ 50 & 50 & 50 & 50 & 50 & 50 & 50 & 50 \\ 50 & 50 & 50 & 50 & 50 & 50 & 50 & 50 \\ 50 & 50 & 50 & 50 & 50 & 50 & 50 & 50 \end{bmatrix}$$

Mask that will be used is 3×3

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

Iteration 1:—

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\ 10 & \underline{10} & 10 & 10 & 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\ 50 & 50 & 50 & 50 & 50 & 50 & 50 & 50 \\ 50 & 50 & 50 & 50 & 50 & 50 & 50 & 50 \\ 50 & 50 & 50 & 50 & 50 & 50 & 50 & 50 \\ 50 & 50 & 50 & 50 & 50 & 50 & 50 & 50 \end{bmatrix}$$

After multiplication & average of each element we get center pixel as 10 so no change.

→ Similarly the mask will be shifted to the next position row-wise and column-wise

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}\begin{bmatrix}10 & 10 & 10 & 10 & 10 & 10 & 10 & 10\\ 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10\\ 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10\\ 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10\\ 50 & 50 & 50 & 50 & 50 & 50 & 50 & 50\\ 50 & 50 & 50 & 50 & 50 & 50 & 50 & 50\\ 50 & 50 & 50 & 50 & 50 & 50 & 50 & 50\\ 50 & 50 & 50 & 50 & 50 & 50 & 50 & 50\end{bmatrix}$$

$\Rightarrow 1\times10 + 1\times10 + 1\times10 + 1\times10 + 1\times10$

$+ 1\times10 + 1\times50 + 1\times50 + 1\times50$

$\Rightarrow 23.33$

→ Similarly the other values can be calculated.

→ The result of convolving the image with the 3×3 averaging mask

$$\begin{bmatrix}10 & 10 & 10 & 10 & 10 & 10 & 10 & 10\\ 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10\\ 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10\\ 23.33 & 23.33 & 23.33 & 23.33 & 23.33 & 23.33 & 23.33 & 23.33\\ 36.6 & 36.6 & 36.6 & 36.6 & 36.6 & 36.6 & 36.6 & 36.6\\ 50 & 50 & 50 & 50 & 50 & 50 & 50 & 50\\ 50 & 50 & 50 & 50 & 50 & 50 & 50 & 50\\ 50 & 50 & 50 & 50 & 50 & 50 & 50 & 50\end{bmatrix}$$

Low Pass Averaging Filter

- Low frequency regions remain unchanged while sharp edges become blur.

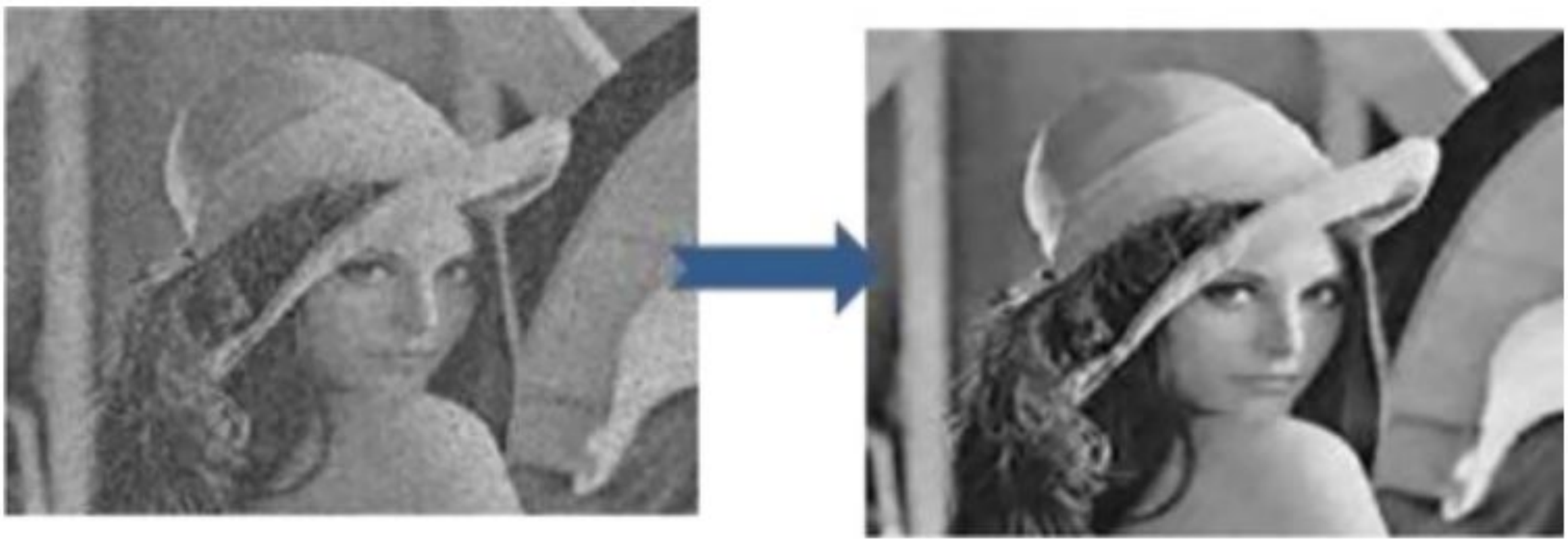
- Grey level transition from 10 to 50 reduces to 10 -23.33 -36.6-50 blurring effect.

Low Pass Averaging Filter

- Limitations of Averaging Filter

1. Averaging operation **leads to the blurring** of an image. Blurring affects feature localisation.
2. If the averaging operation is applied to an image corrupted by impulse noise then the impulse noise is attenuated and diffused but not removed.
3. A single pixel with unrepresentative value can affect the mean value of the centre pixel.

Impulse Noise



Low Pass Averaging Filter

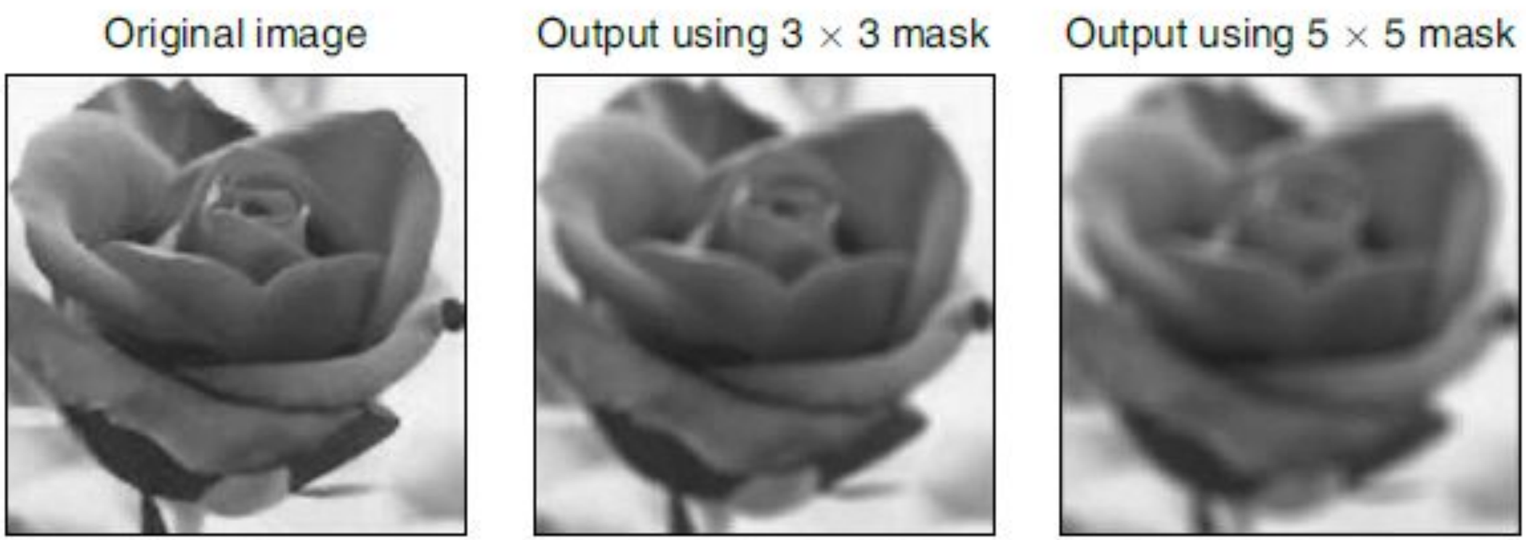


Fig. 5.33 *Results of spatial domain filters*

Weighted Average Filter

- The mask of a weighted average filter is given by

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

- the pixels nearest to the centre are weighted more than the distant pixels

Weighted Average Filter

- In the mask the pixel at the center of the mask is multiplied by a higher value than any other, thus giving this pixel more importance in the calculation of the average.

Median Filter

- Median filters are statistical non linear filters in spatial domain.
- Median filter smoothens the image by utilising the median of the neighbourhood.
- **Steps performed by median filter to find each pixel value in processed image:**

1. All pixels in the neighbourhood of the pixel in the original image which are identified by the mask are sorted in the ascending (or) descending order.
2. The median of the sorted value is computed and is chosen as the pixel value for the processed image.

Median Filter

Example 5.5 *Compute the median value of the marked pixel shown in Fig. 5.40 using a 3 × 3 mask.*

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

Fig. 5.40 *Data for Example 5.5*

Step 1 First, the pixel values are arranged in ascending order as follows:

$$1\ 1\ 2\ 2\ 3\ 4\ 5\ 6\ 7$$

Step 2 The median value of the ordered pixel is computed as follows:

$$\cancel{1}\ \cancel{1}\ \cancel{2}\ \cancel{2}\ 3\ \cancel{4}\ \cancel{5}\ \cancel{6}\ \cancel{7}$$

The median value is computed to be 3. Then, the original pixel value of 4 will be replaced by the computed median value of 3.

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

Original image data — After median filtering

Example 5.6 *Compute the median value of the marked pixels shown in Fig. 5.41 using a 3 × 3 mask.*

$$\begin{bmatrix} 18 & 22 & 33 & 25 & 32 & 24 \\ 34 & 128 & 24 & 172 & 26 & 23 \\ 22 & 19 & 32 & 31 & 28 & 26 \end{bmatrix}$$

Fig. 5.41 *Input image*

Step 1 **To compute the median value of the marked pixel 128**

$$\begin{bmatrix} 18 & 22 & 33 & 25 & 32 & 24 \\ 34 & 128 & 24 & 172 & 26 & 23 \\ 22 & 19 & 32 & 31 & 28 & 26 \end{bmatrix}$$

Step 1a The eight neighbours of the marked pixel '128' are now arranged in ascending order as follows:

18 19 22 22 24 32 33 34 128

Step 1b The median value is computed as **24**.

$$\cancel{18}\ \cancel{19}\ \cancel{22}\ \cancel{22}\ 24\ \cancel{32}\ \cancel{33}\ \cancel{34}\ \cancel{128}$$

Step 2 **To Compute the median value of the marked pixel 24**

Step 2a Arrange the pixels in the neighbourhood (shown by dotted lines) in the ascending order.

$$\begin{bmatrix} 18 & 22 & 33 & 25 & 32 & 24 \\ 34 & 128 & 24 & 172 & 26 & 23 \\ 22 & 19 & 32 & 31 & 28 & 26 \end{bmatrix}$$

After arranging the pixels in the neighbourhood in the ascending order, the values are displayed as follows:

19 22 24 25 31 32 33 128 174

Step 2b The median value is computed to be **31**.

$$\cancel{19}\ \ \cancel{22}\ \ \cancel{24}\ \ \cancel{25}\ \ 31\ \ \cancel{32}\ \ \cancel{33}\ \ \cancel{128}\ \ \cancel{174}$$

Step 3 **To compute the median value of the marked pixel 172**

Step 3a Arrange the pixels in the neighbourhood of 172 in the ascending order.

$$\begin{bmatrix} 18 & 22 & 33 & 25 & 32 & 24 \\ 34 & 128 & 24 & 172 & 26 & 23 \\ 22 & 19 & 32 & 31 & 28 & 26 \end{bmatrix}$$

After arranging the pixel values in the neighbourhood in the ascending order, we have

$$24\ 25\ 26\ 28\ 31\ 32\ 32\ 33\ 172$$

Step 3b The median value is computed as **31**.

$$\cancel{24}\ \ \cancel{25}\ \ \cancel{26}\ \ \cancel{28}\ \ 31\ \ \cancel{32}\ \ \cancel{32}\ \ \cancel{33}\ \ \cancel{172}$$

Step 4 **To compute the median value of the marked pixel 26**

Step 4a Arrange the pixels in the neighbourhood of 26 in the ascending order:

$$\begin{bmatrix} 18 & 22 & 33 & 25 & 32 & 24 \\ 34 & 128 & 24 & 172 & 26 & 23 \\ 22 & 19 & 32 & 31 & 28 & 26 \end{bmatrix}$$

After arranging the pixel values in the neighbourhood in the ascending order, we have

23 24 25 26 26 28 31 32 172

Step 4b The median value is computed as 26.

$$\cancel{23}\ \cancel{24}\ \cancel{25}\ \cancel{26}\ 26\ \cancel{28}\ \cancel{31}\ \cancel{32}\ \cancel{172}$$

The original pixel values and the values replaced by their median are shown side by side below:

$$\begin{bmatrix} 18 & 22 & 33 & 25 & 32 & 24 \\ 34 & 128 & 24 & 172 & 26 & 23 \\ 22 & 19 & 32 & 31 & 28 & 26 \end{bmatrix} \longrightarrow \begin{bmatrix} 18 & 22 & 33 & 25 & 32 & 24 \\ 34 & 24 & 31 & 31 & 26 & 23 \\ 22 & 19 & 32 & 31 & 28 & 26 \end{bmatrix}$$

Original pixel value **Median value**

Median Filter

- When we take the median value, the pixel values which are very different from their neighbouring pixels are replaced by a value equal to the neighbouring pixel value.
- median filter is capable of reducing salt-and-pepper noise.
- A median filter is an effective tool to minimise salt-and-pepper noise.

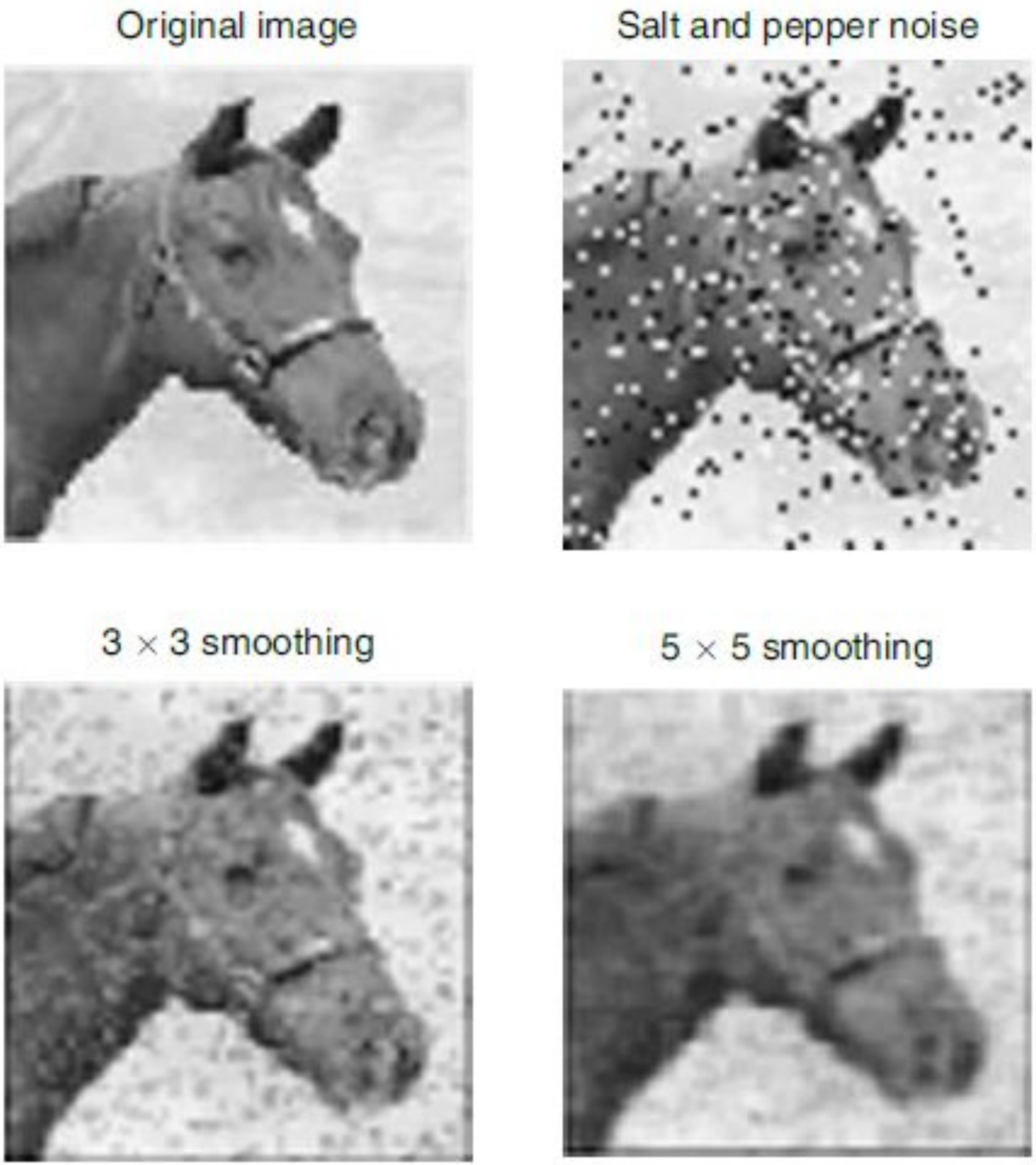

Original image
Salt and pepper noise
3 × 3 smoothing
5 × 5 smoothing

Fig. 5.43 *Output of the box and median-filtered image*

Comparison Between Filters:

Parameters	Average Filter	Weighted Filter	Median Filter
Noise Reduction	Reduces Noise but it introduces blurring effect at edges.	Blurring effect is less as compared with Average filter	Blurring effect is less as compared with Average filter
Percentage of noise Reduction	100% noise Not Reduced	100% noise Not Reduced	Almost 100% noise reduced.
Size of Filter	As we increase the size of the filter mask, Noise reduces but blurring effect increases.	As we increase the size of the filter mask, Noise reduces but blurring effect increases.	As we increase the size of the filter mask, 100% of Noise reduces but blurring effect at edges increases.
Mask	1/9x[1,1,1;1,1,1;1,1,1]	1/16x[1,2,1;2,4,4;1,2,1]	Pixel value is replaced by median value of neighborhood.
MATLAB Function	filter2(mask,Noisy_img)	filter2(mask,Noisy_img)	medfilt2(Noisy_img,[3 3])

Highpass Filtering

- High pass filtering eliminates the low frequency regions while retaining or enhancing the high frequency components.
- High pass image will have no background as it’s a low frequency region.
- The main aim of image sharpening is to highlight fine details in the image.
- Image sharpening is used to enhance the high-frequency components.

Highpass Filtering

The spatial mask which performs image sharpening is given as $\frac{1}{9}\times\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$

Highpass Filtering

- Sum of coefficients of high pass mask has to be zero to eliminate the low frequency regions.

Example of High Pass Filtering

$$\begin{bmatrix} 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\ 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 \end{bmatrix}$$

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

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -270 & -270 & -270 & -270 & -270 & -270 & -270 & -270 \\ 270 & 270 & 270 & 270 & 270 & 270 & 270 & 270 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

1) Pixel values cannot be negative. They always start with zero. Consider all negative values as zero.

2) The values of the original image at the edge are very large and there is a tendency of them going out of range due to center weight of +8.
1/9 is simply scaling factor.

final result using mask & negative values zero.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 30 & 30 & 30 & 30 & 30 & 30 & 30 & 30 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

High-boost Filtering

- High-boost filter is also known as a high-frequency emphasis filter.
- A high-boost filter is used to retain some of the low-frequency components(background) to aid in the interpretation of an image.
- In high-boost filtering, the input image f(m, n) is multiplied by an amplification factor A before subtracting the low-pass image.

High-boost Filtering

- High pass filter gets rid of the complete background.
- Thus, the high-boost filter expression becomes
  A is amplification factor

$$\text{high boost} = (A-1) \times f(m,n) + \text{high pass}$$

HBF Mask

- A or K amplification factor

The HBF mask is given by,

$$w = 1/9 \begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline -1 & 9k-1 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$$

Zooming

- When an image has fine details, it may not be possible to examine the detail easily in a standard display on a monitor.
- zooming operation helps to view fine details in the image.
- Zooming is equivalent to holding a magnifying glass in front of the screen.
- The simplest zooming operation is through ‘replication’ of pixels.

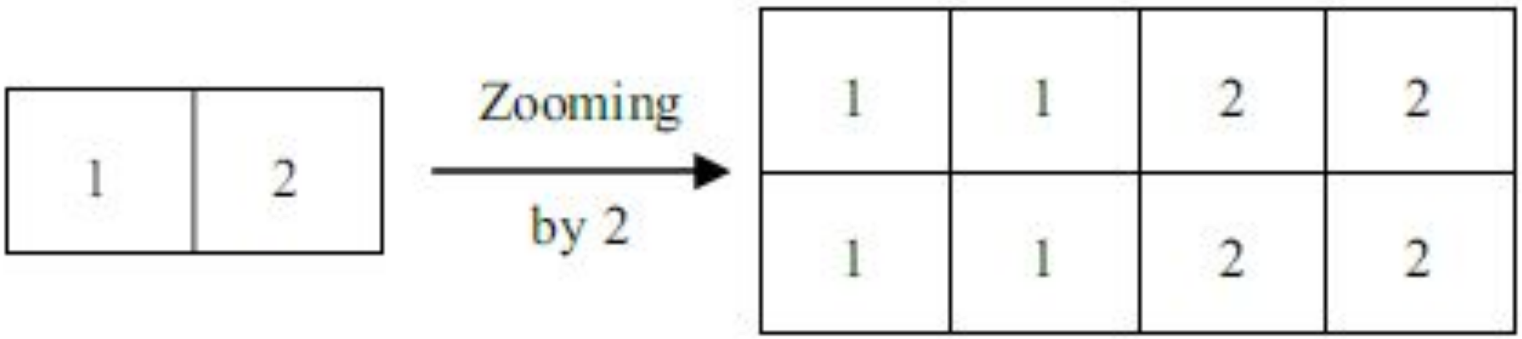


Fig. 5.73 *Zooming illustration*

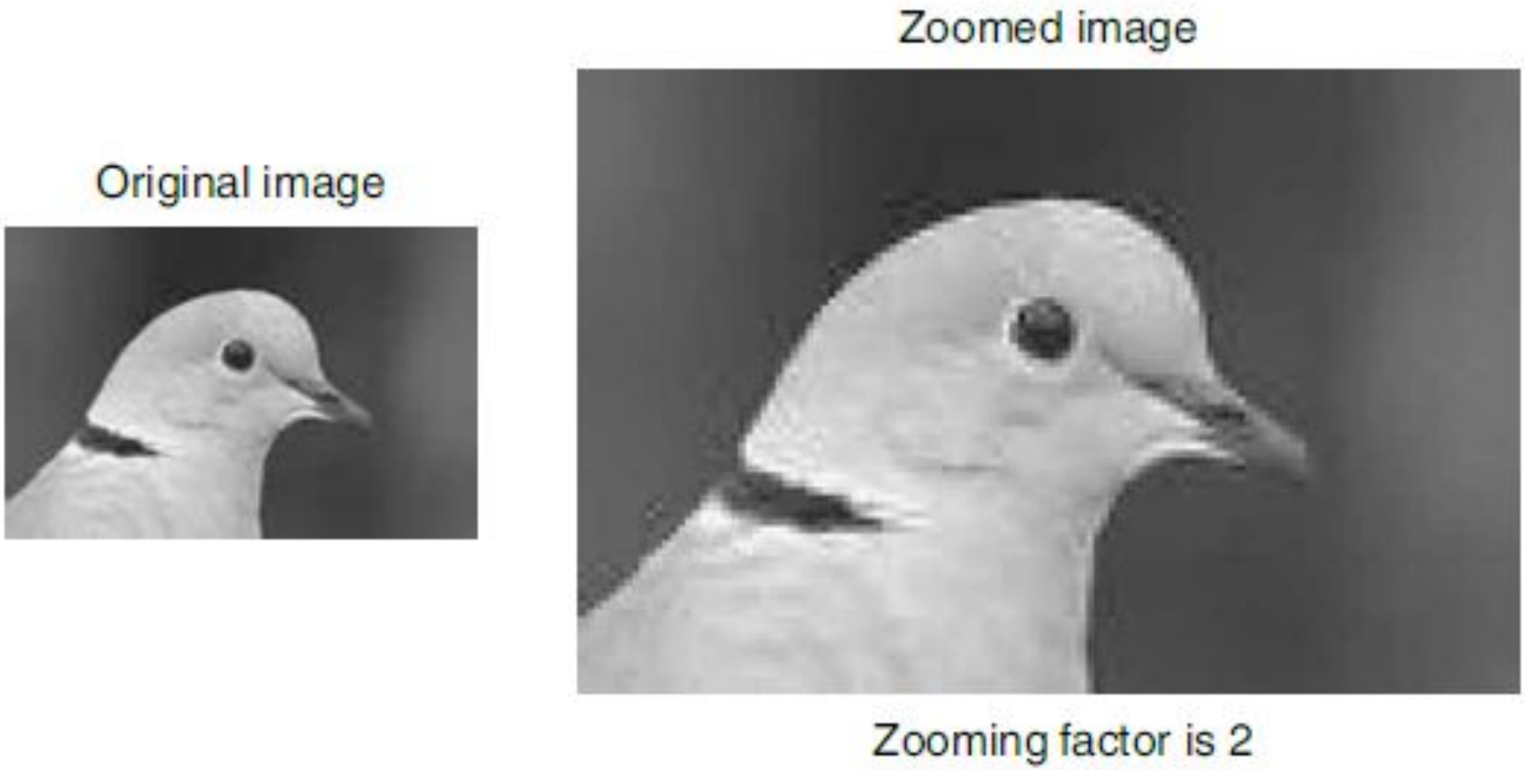


Fig. 5.75 *Output of zooming operation*

Histogram Modelling

- Histogram of images provides global description of the appearance of an image.
- Histogram of an image represents the relative frequency of occurrence of the various grey levels in an image.

Histogram Modelling

- Two methods of plotting histogram.

Method 1 :

Grey level	Number of pixels (n_k)
0	40
1	20
2	10
3	15
4	10
5	3
6	2

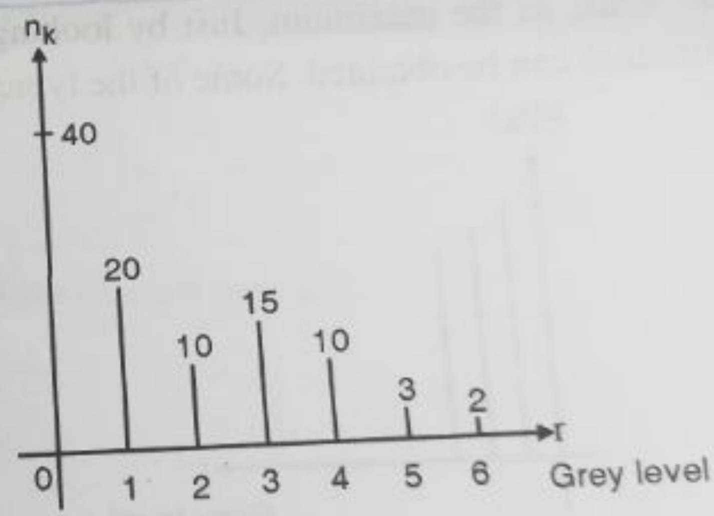


Fig. 8.1

Histogram Modelling

Grey level	Number of pixels (n_k)	$p(r_k) = \frac{n_k}{n}$
0	40	0.4
1	20	0.2
2	10	0.1
3	15	0.15
4	10	0.1
5	3	0.03
6	2	0.02
	n = 100	

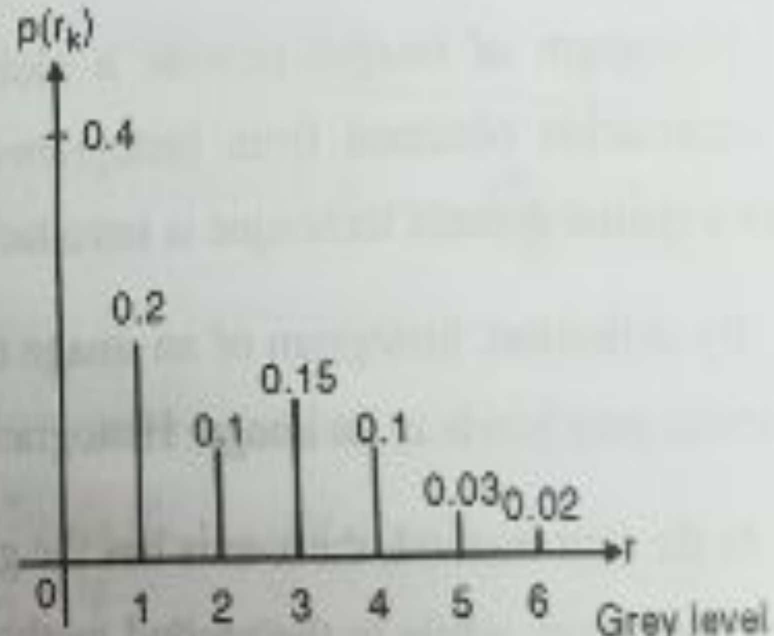


Fig. 8.2

Histogram Modelling

- Histogram plotted in the second method is normalised histogram.
- Maximum value to be plotted will be 1.
- Normalised histogram provides clear information about the image.

Linear Stretching

- Histogram stretching is one way to increase dynamic range of the image.
- Here the basic shape of the histogram is not altered , instead the histogram is spread to cover the entire dynamic range.

$$s = T(r) = \frac{s_{max} - s_{min}}{r_{max} - r_{min}} (r - r_{min}) + s_{min}$$

s_{max} → Maximum grey level of output image

s_{min} → Minimum grey level of output image

r_{max} → Maximum grey level of input image

r_{min} → Minimum grey level of input image

Histogram Equalization

- Equalisation is a process that attempts to spread out the gray levels in an image so that they are evenly distributed across their range.

- Histogram equalisation reassigns the brightness values of pixels based on the image histogram.

- Histogram equalisation is a technique where the histogram of the resultant image is as flat as possible.

Histogram Equalization

- Histogram equalisation provides more visually pleasing results across a wider range of images.

(c) *Procedure to Perform Histogram Equalisation*

Histogram equalisation is done by performing the following steps:

1. Find the running sum of the histogram values.
2. Normalise the values from Step (1) by dividing by the total number of pixels.
3. Multiply the values from Step (2) by the maximum gray-level value and round.
4. Map the gray level values to the results from Step (3) using a one-to-one correspondence.

Example 5.1 *Perform histogram equalisation of the image* $\begin{bmatrix} 4 & 4 & 4 & 4 & 4 \\ 3 & 4 & 5 & 4 & 3 \\ 3 & 5 & 5 & 5 & 3 \\ 3 & 4 & 5 & 4 & 3 \\ 4 & 4 & 4 & 4 & 4 \end{bmatrix}$.

Solution The maximum value is found to be 5. We need a minimum of 3 bits to represent the number. There are eight possible gray levels from 0 to 7. The histogram of the input image is given below:

Gray level	0	1	2	3	4	5	6	7
Number of pixels	0	0	0	6	14	5	0	0

Step 1 Compute the running sum of histogram values.

The running sum of histogram values is otherwise known as cumulative frequency distribution.

Gray level	0	1	2	3	4	5	6	7
Number of pixels	0	0	0	6	14	5	0	0
Running sum	0	0	0	6	20	25	25	25

Step 2 Divide the running sum obtained in Step 1 by the total number of pixels. In this case, the total number of pixels is 25.

Gray level	0	1	2	3	4	5	6	7
Number of pixels	0	0	0	6	14	5	0	0
Running sum	0	0	0	6	20	25	25	25
Running Sum/Total number of pixels	0/25	0/25	0/25	6/25	20/25	25/25	25/25	25/25

Step 3 Multiply the result obtained in Step 2 by the maximum gray-level value, which is 7 in this case.

Gray level	0	1	2	3	4	5	6	7
Number of pixels	0	0	0	6	14	5	0	0
Running Sum	0	0	0	6	20	25	25	25
Running sum / Total number of pixels	0/25	0/25	0/25	6/25	20/25	25/25	25/25	25/25
Multiply the above result by maximum gray level	$\frac{0}{25} * 7$	$\frac{0}{25} * 7$	$\frac{0}{25} * 7$	$\frac{6}{25} * 7$	$\frac{20}{25} * 7$	$\frac{25}{25} * 7$	$\frac{25}{25} * 7$	$\frac{25}{25} * 7$

The result is then rounded to the closest integer to get the following table:

Gray level	0	1	2	3	4	5	6	7
Number of pixels	0	0	0	6	14	5	0	0
Running Sum	0	0	0	6	20	25	25	25
Running Sum/Total number of pixels	0/25	0/25	0/25	6/25	20/25	25/25	25/25	25/25
Multiply the above result by maximum gray level	0	0	0	2	6	7	7	7

Step 4 Mapping of gray level by a one-to-one correspondence:

Original gray level	Histogram equalised values
0	0
1	0
2	0
3	2
4	6
5	7
6	7
7	7

The original image and the histogram equalised image are shown side by side.

$$\begin{bmatrix} 4 & 4 & 4 & 4 & 4 \\ 3 & 4 & 5 & 4 & 3 \\ 3 & 5 & 5 & 5 & 3 \\ 3 & 4 & 5 & 4 & 3 \\ 4 & 4 & 4 & 4 & 4 \end{bmatrix} \xrightarrow[\text{Equalisation}]{\text{Histogram}} \begin{bmatrix} 6 & 6 & 6 & 6 & 6 \\ 2 & 6 & 7 & 6 & 2 \\ 2 & 7 & 7 & 7 & 2 \\ 2 & 6 & 7 & 6 & 2 \\ 6 & 6 & 6 & 6 & 6 \end{bmatrix}$$

Original image **Histogram equalised image**

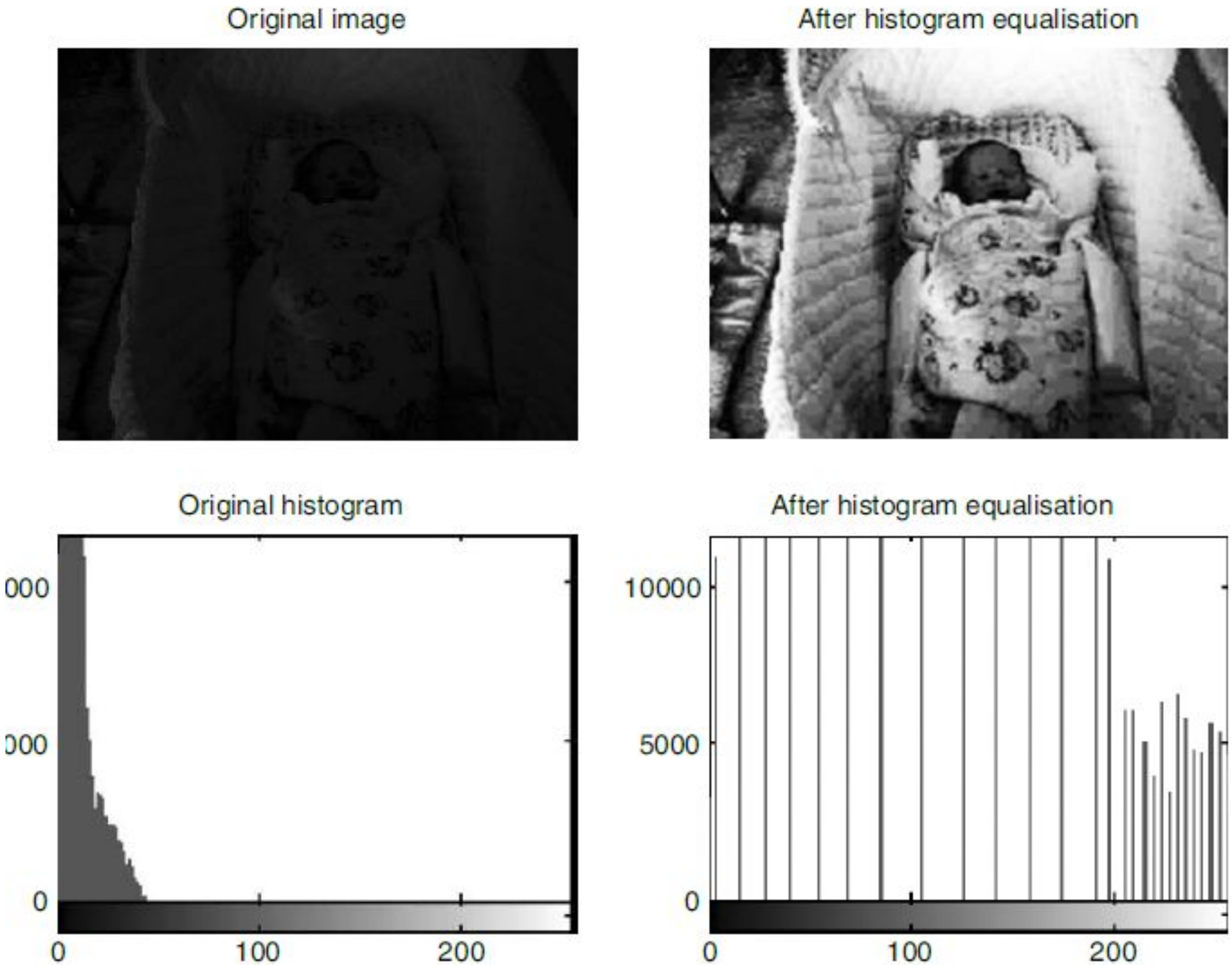


Fig. 5.11 *Results of histogram equalisation*