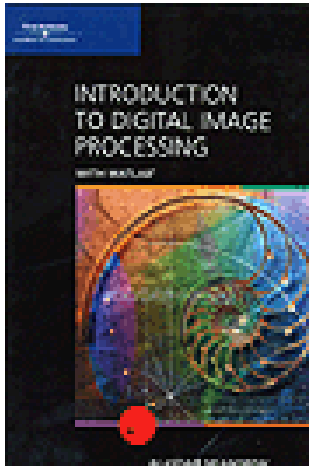
Digital Image Processing with Python

Dr.Su Su Maung
CEIT Department
Yangon Technological University

Text Book & References

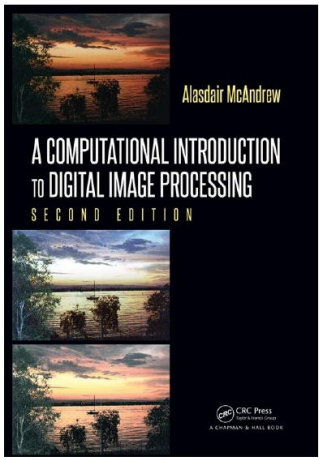


INTRODUCTION TO DIGITAL IMAGE PROCESSING WITH MATLAB, 2nd Edition

Alasdair McAndrew

A COMPUTATIONAL INTRODUCTION TO DIGITAL IMAGE PROCESSING, 2nd Edition

Alasdair McAndrew



Course Outline

Week	Topics
Week 1	Introduction to Image Processing
Week 2	Image Enhancement in Spatial Domain (Point Processing)
Week 3	Image Enhancement in Spatial (Neighborhood Processing)
Week 4	Image Enhancement in Frequency Domain
Week 5	Image Restoration
Week 6	Image Segmentation
Week 7	Mathematical Morphology
Week 8	Color Processing
Week 9	Image Coding and Compression
Week 10	Wavelets Transform
Week 11	Extracting Image Features
Week 12	Image Classification

Exam and Grading System

Grading policy:

1. Final Examination: 80%
2. Practical Assignment: 20%

Lecture 1

Introduction

Outline of Lecture1

- What is an image
- What is image processing
- Some Applications
- Aspects of image processing
- Important Stages in Image Processing
- Types of digital images
- Image Handling and I/O in Python

What is an Image?

- Images and pictures
  - an image is a single picture which represents something. It may be a picture of a person, of people or animals, or of an outdoor scene, or a microphotograph of an electronic component, or the result of medical imaging.

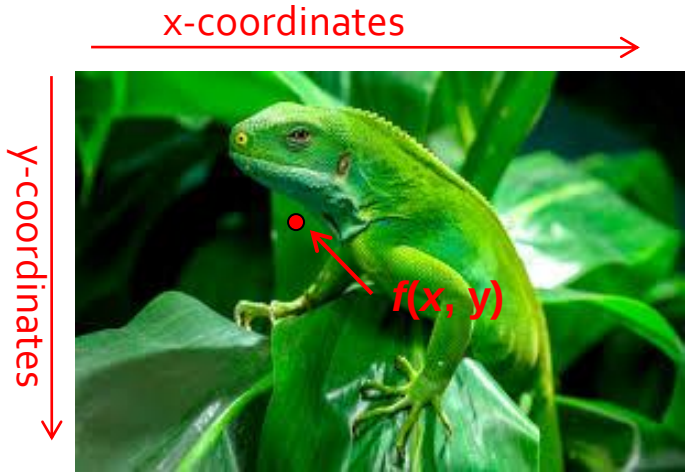
Example of Image

Color image

Gray image

What is an Image?

- An image is nothing more than a two dimensional signal. It is defined by the mathematical function f(x,y) where x and y are the two coordinates horizontally and vertically.
- The value of f(x,y) at any point is gives the pixel value at that point of an image.



What is image processing?

- image processing is processing of images using mathematical operations by using any form of signal processing.
- the input is an image, a series of images or a video, such as a photograph or video frame.
- the output of image processing may be either an image or a set of characteristics or parameters related to the image.

Image processing System

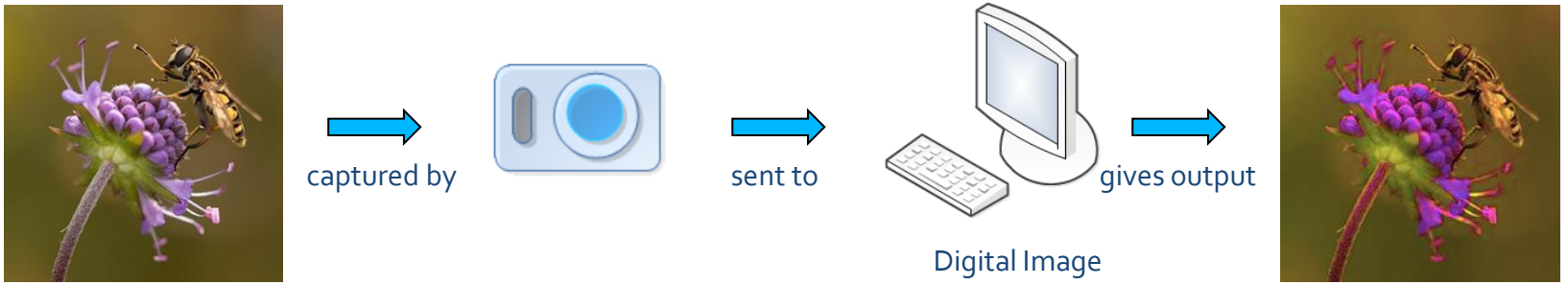


Image processing System

Image Processing to Computer Vision

Low-Level Processing

Noise reduction, Contrast Enhancement, Image sharpening

Image

Middle-Level Processing

Segmentation, Feature Extraction

Attributes

High-Level Processing

Classification, Object Recognition

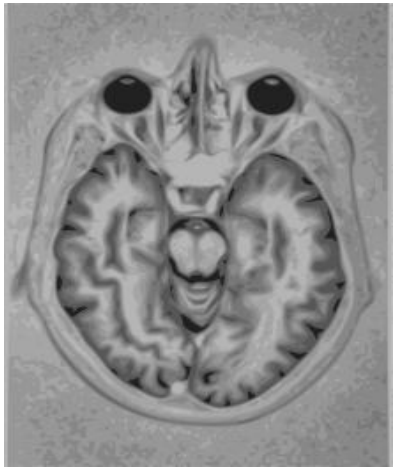
Making Sense

Some applications

- Image processing has an enormous range of applications; almost every area of science and technology can make use of image processing methods.
  - Medicine
  - Agriculture
  - Industry
  - Law enforcement

Some applications

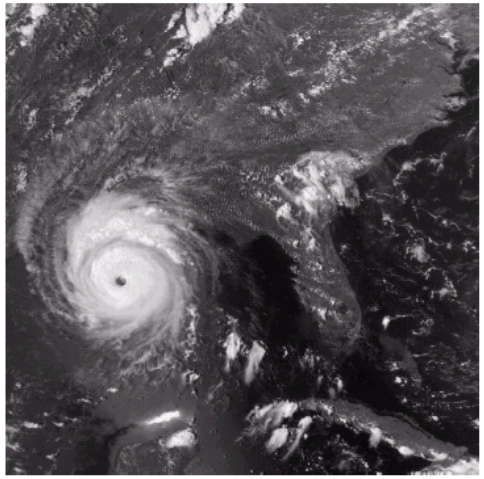
- Medicine
  - Inspection and interpretation of images obtained from X-rays,
  - analysis of cell images, of chromosome karyotypes. MRI or CAT scans,



A brain MRI representation

Some applications

- Agriculture
  - Satellite/aerial views of land, for example to determine how much land is being used for different purposes, or to investigate the suitability of different regions for different crops,
  - inspection of fruit and vegetables - distinguishing good and fresh produce from old.

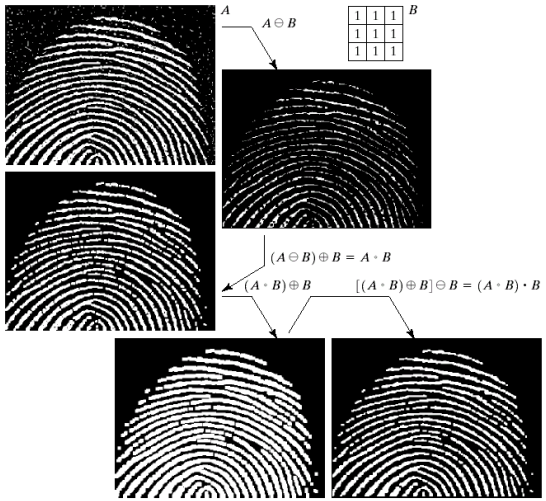


Some applications

- Industry
  - Automatic inspection of items on a production line,
  - inspection of paper samples.

Some applications

- Law enforcement
  - Fingerprint analysis,
  - sharpening or de-blurring of speed-camera images.

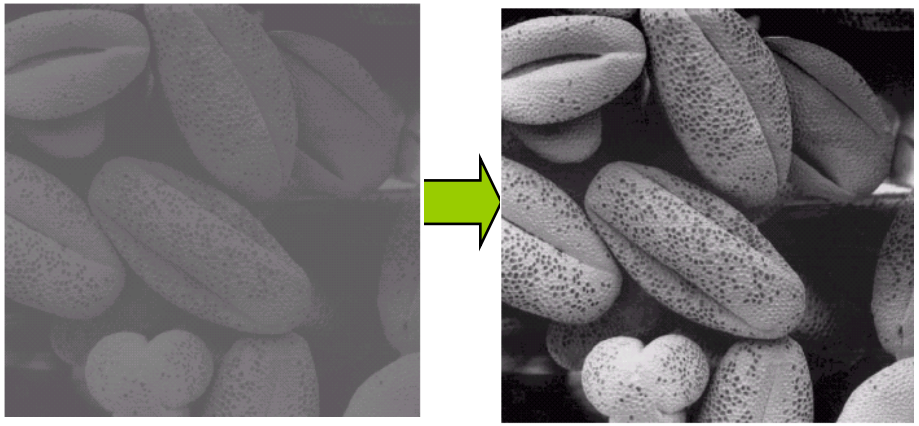


Aspects of image processing

- It is convenient to subdivide different image processing algorithms into broad subclasse. There are different algorithms for different tasks and problems, and often we would like to distinguish the nature of the task at hand.
  - Image enhancement
  - Image restoration
  - Image segmentation

Image enhancement

- This refers to processing an image so that the result is more suitable for a particular application. Example include:
  - sharpening or de-blurring an out of focus image,
  - highlighting edges,
  - improving image contrast, or brightening an image,

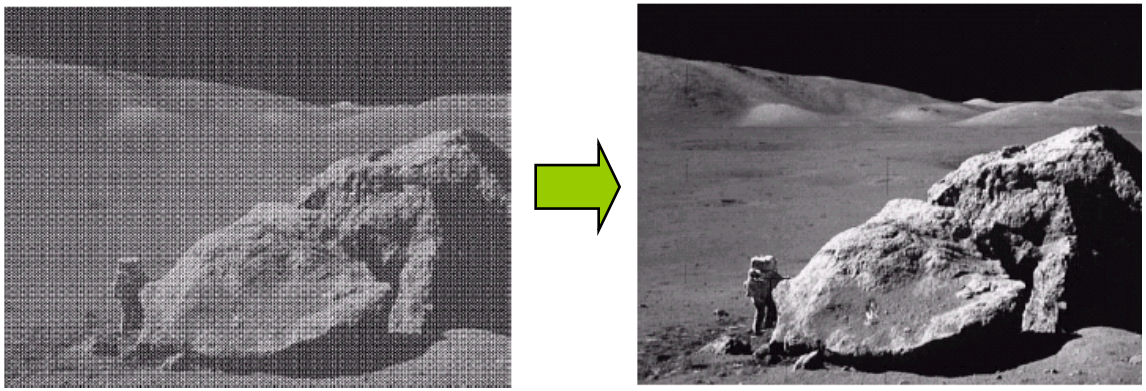


Original

More contrast

Image restoration

- This may be considered as reversing the damage done to an image by a known cause, for example:
  - removing of blur caused by linear motion,
  - removal of optical distortions,
  - removing periodic interference.



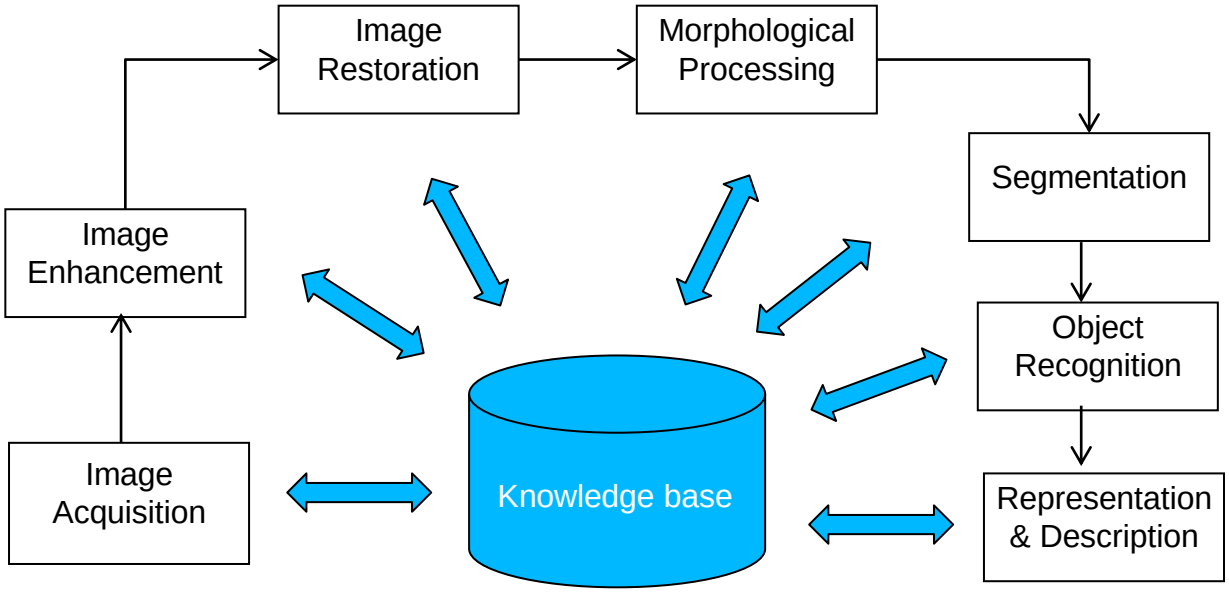
Original

Removing noise

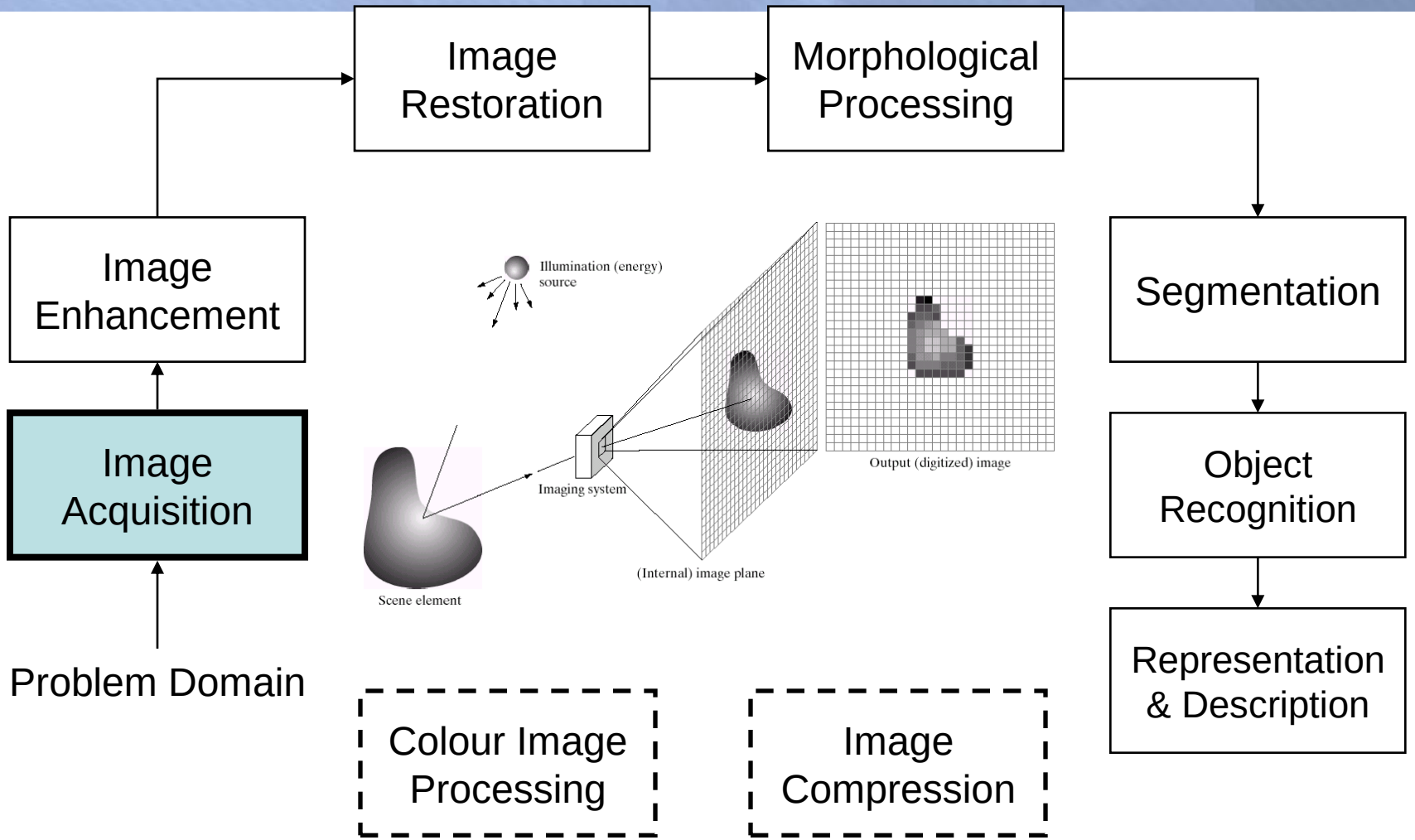
Image segmentation

- This involves subdividing an image into constituent parts, or isolating certain aspects of an image:
  - finding lines, circles, or particular shapes in an image,
  - in an aerial photograph, identifying cars, trees, buildings, or roads.
- These classes are not disjoint; a given algorithm may be used for both image enhancement or for image restoration. However, we should be able to decide what it is that we are trying to do with our image: simply make it look better (enhancement), or removing damage (restoration).

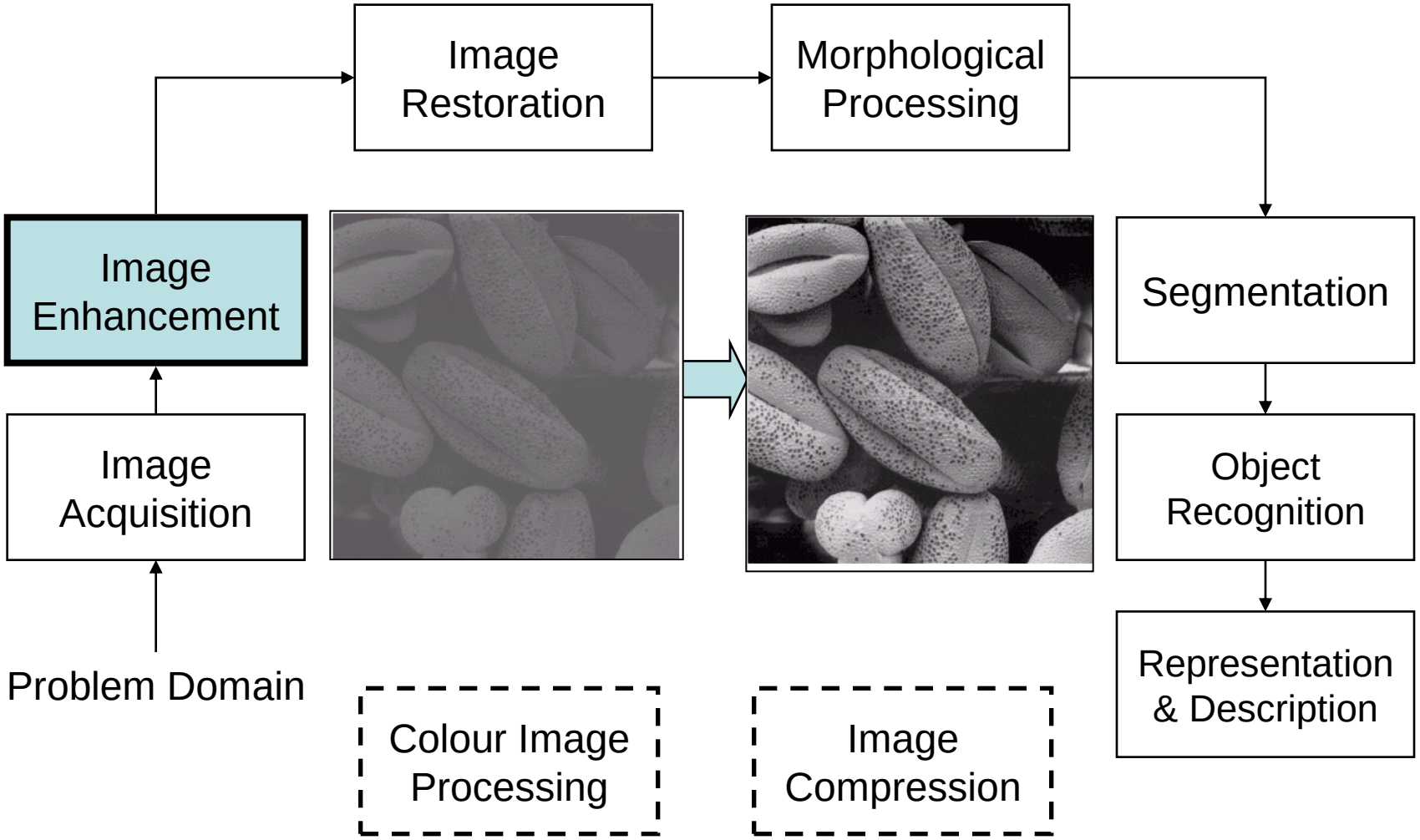
Important Stages in Image Processing



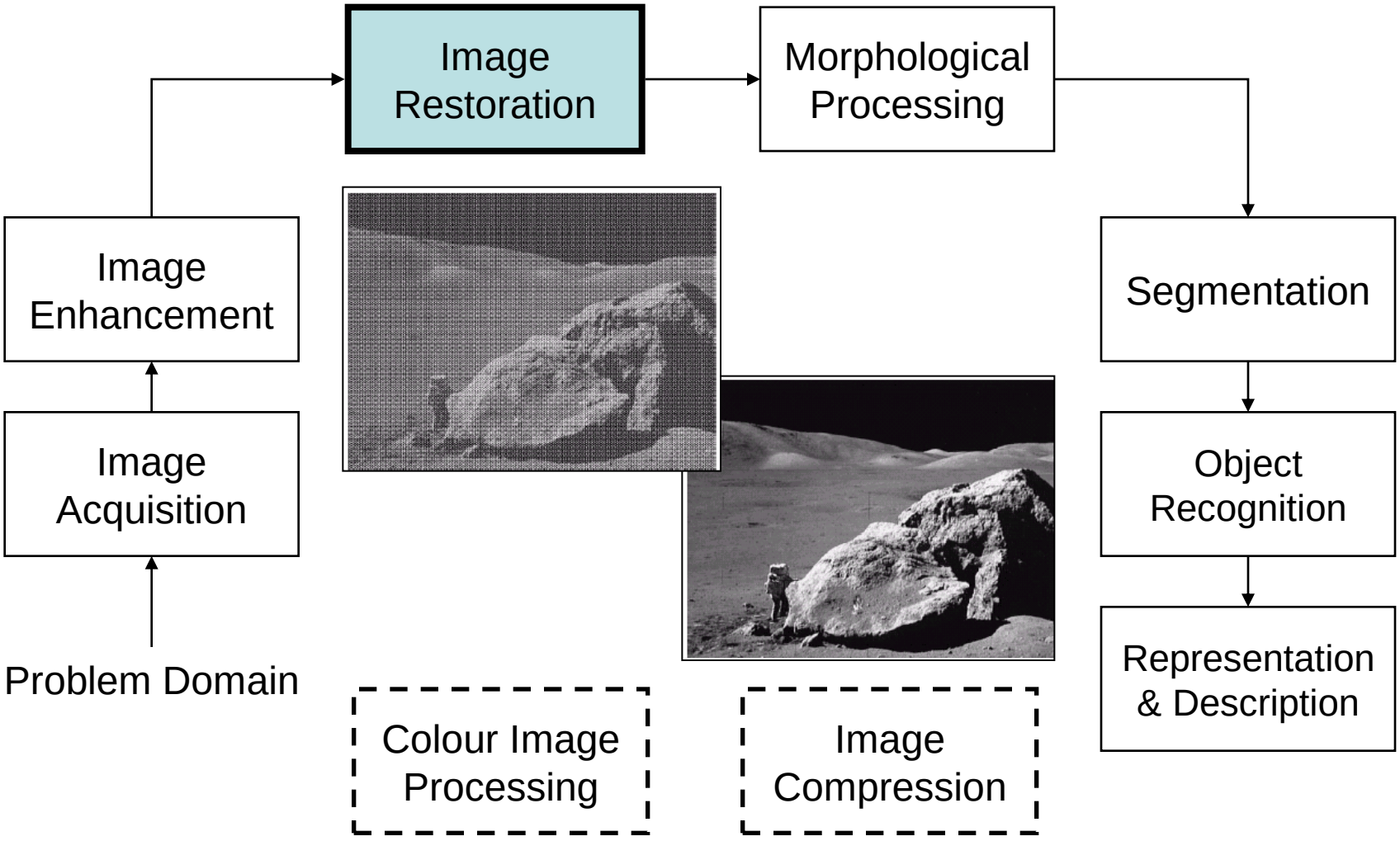
Key Stages in Digital Image Processing: Image Acquisition



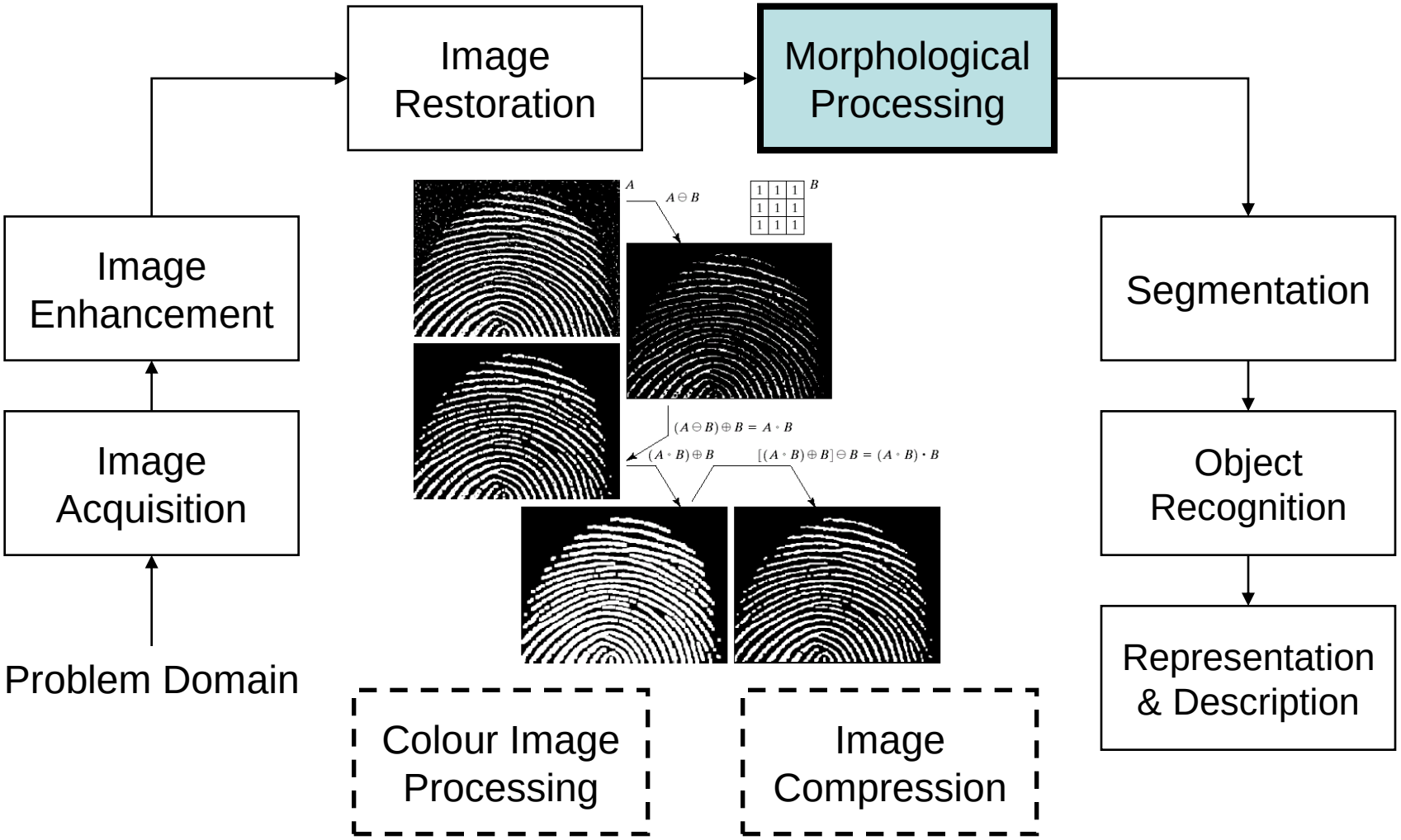
Key Stages in Digital Image Processing: Image Enhancement



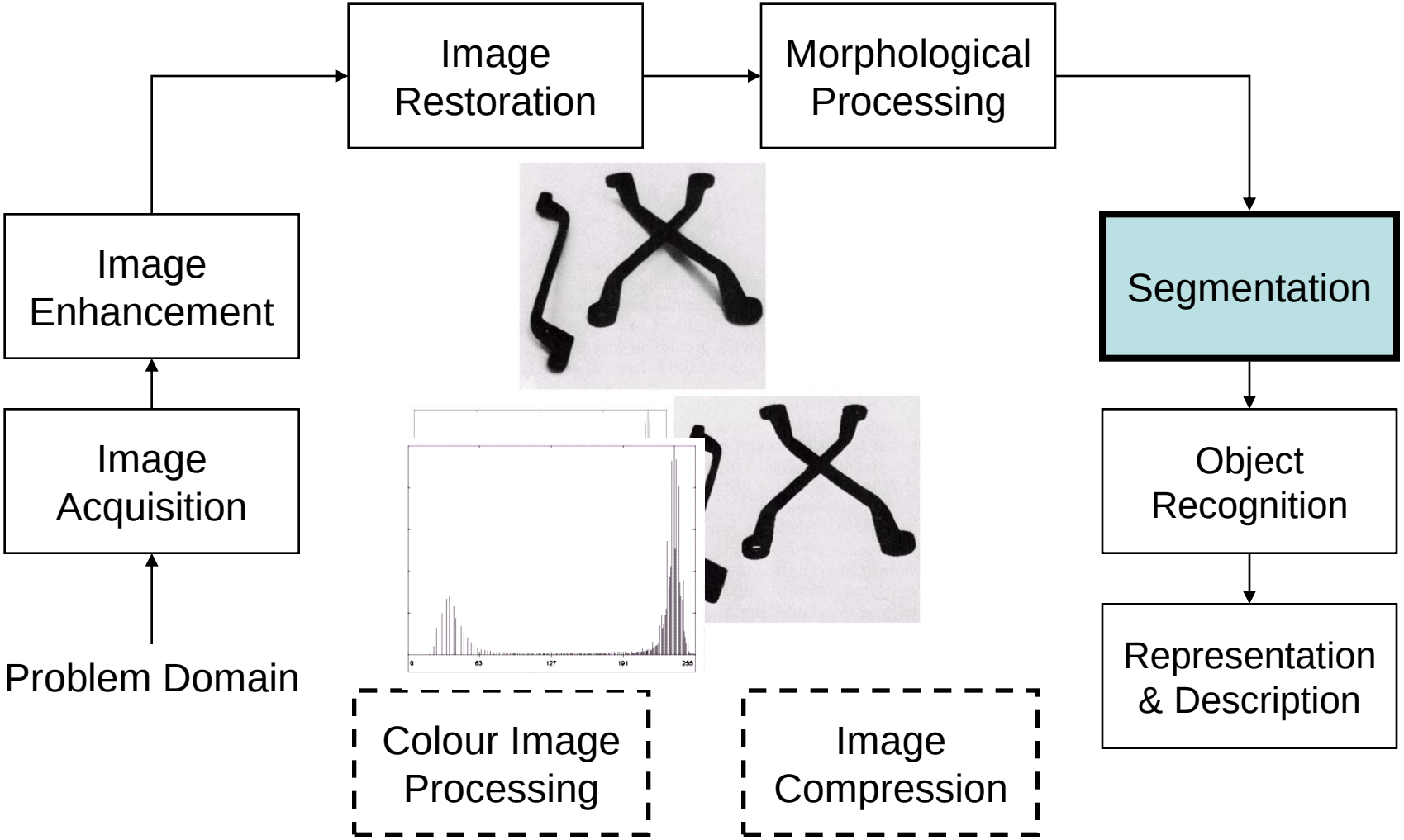
Key Stages in Digital Image Processing: Image Restoration



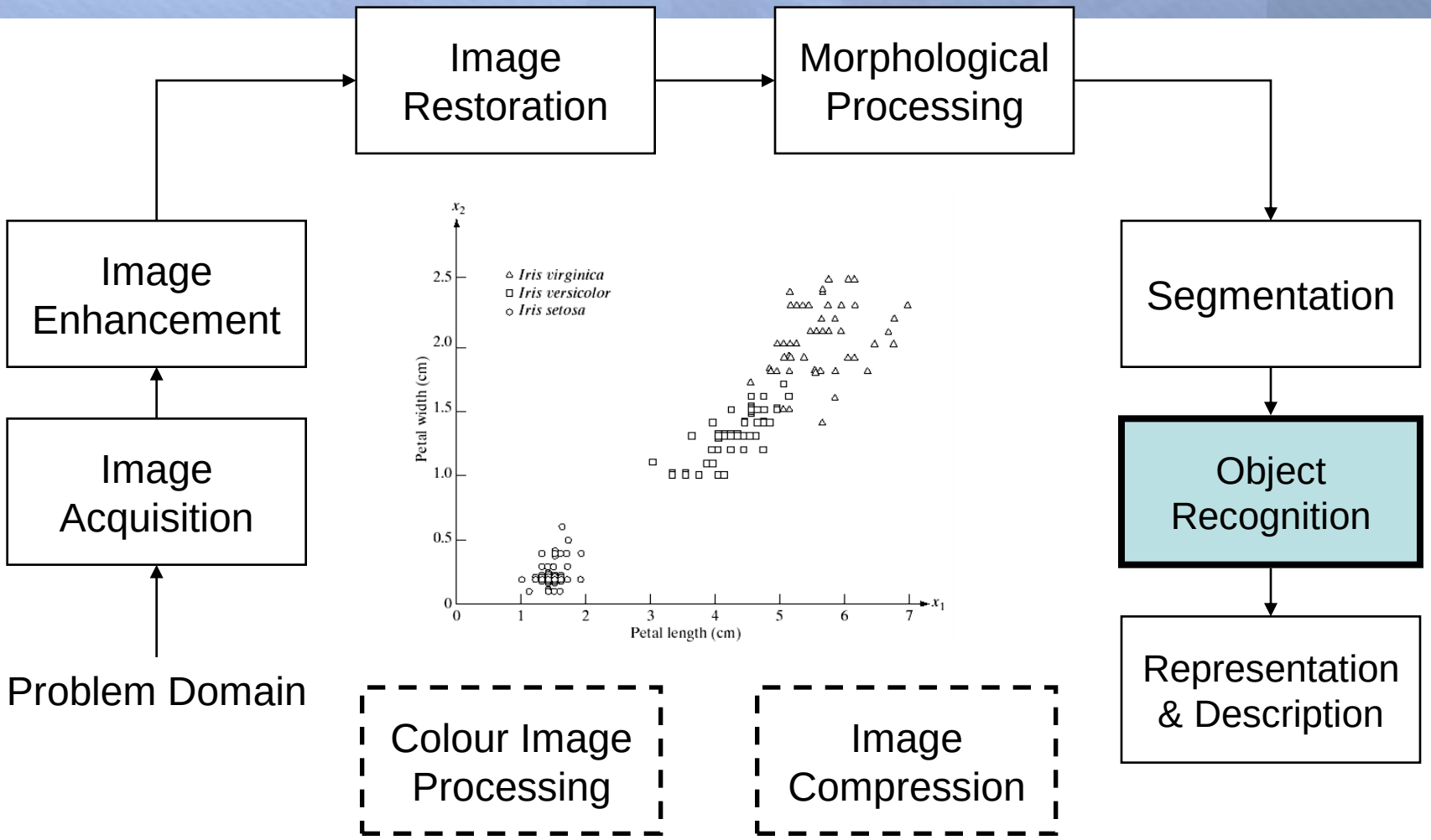
Key Stages in Digital Image Processing: Morphological Processing



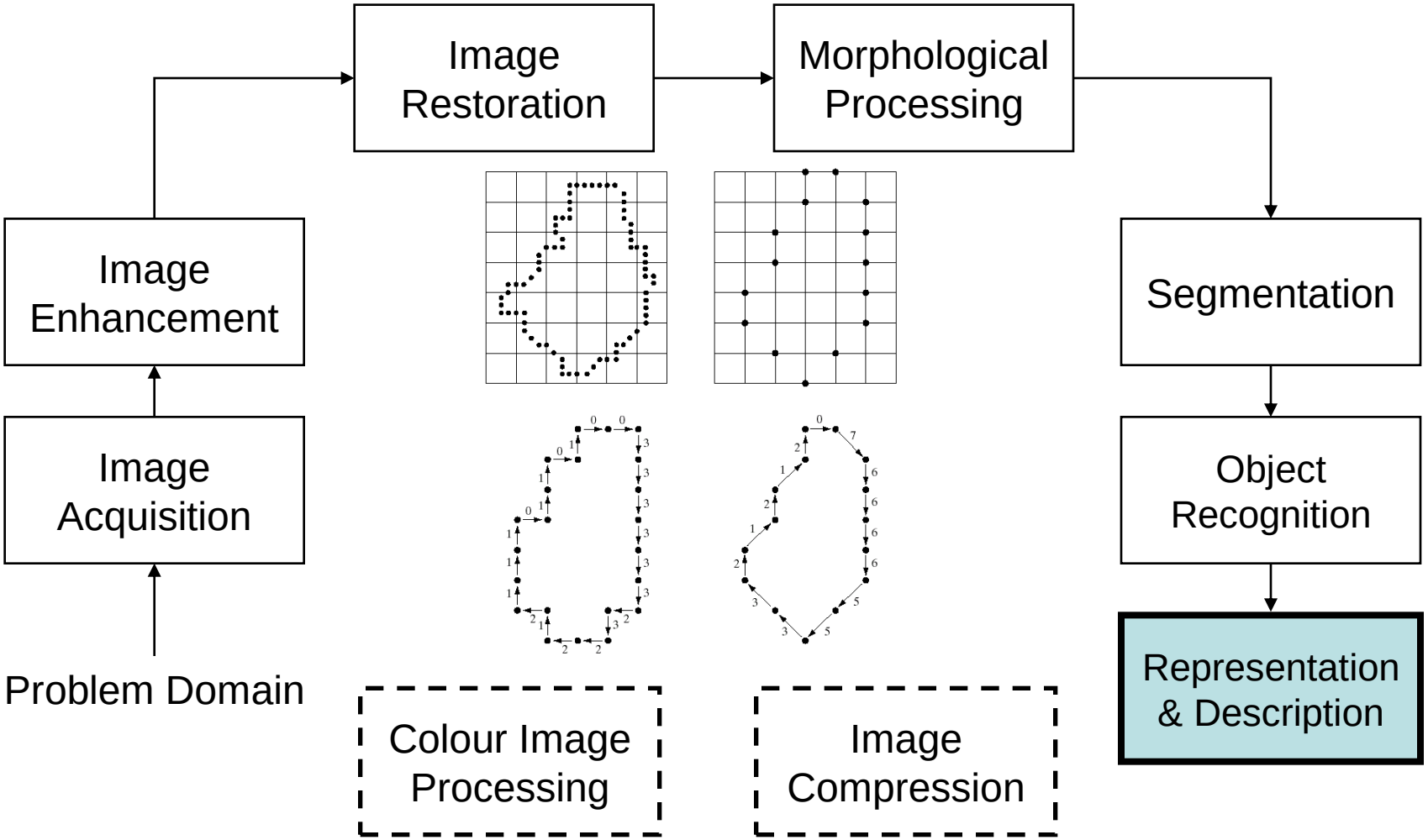
Key Stages in Digital Image Processing: Segmentation



Key Stages in Digital Image Processing: Object Recognition



Key Stages in Digital Image Processing: Representation & Description



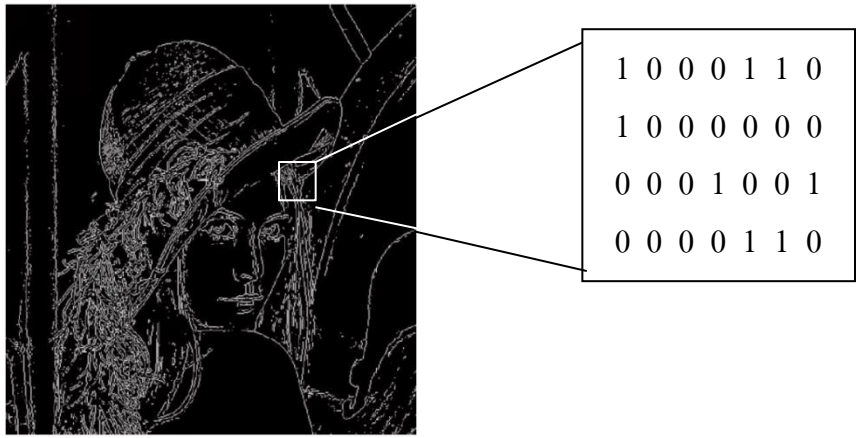
Types of digital images

There are four basic types of images.

- Binary
- Greyscale
- True color, or RGB
- Indexed

Binary Image

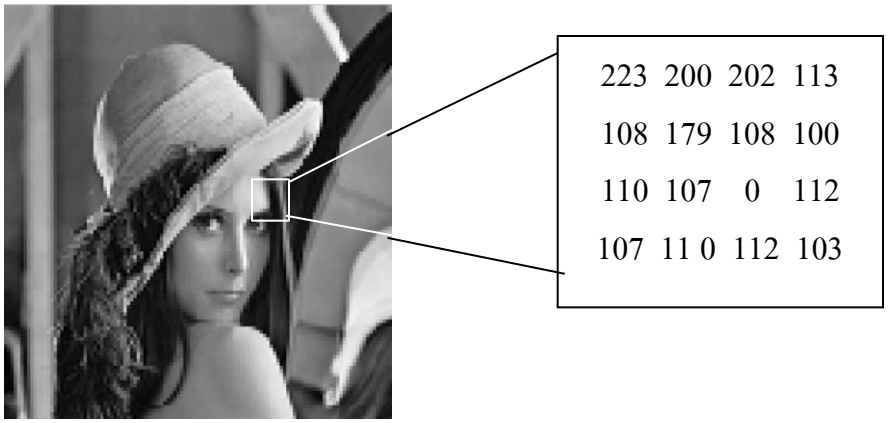
- Each pixel is just black or white.
- only need one bit per pixel.
- very efficient in terms of storage.
- suitable include text (printed or handwriting), fingerprints, or architectural plans.



A Binary image

Greyscale Image

- Each pixel is a shade of grey, normally from 0 (black) to 255 (white).
- Each pixel can be represented by eight bits, or exactly one byte.
- Such images arise in medicine (X-rays), images of printed works.

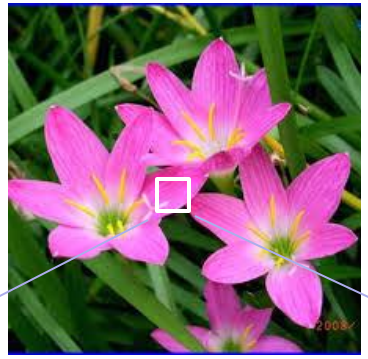


A greyscale image

True colour, or RGB Image

- Here each pixel has a particular colour; that colour being described by the amount of red, green and blue in it.
- If each of these components has a range 0-255, this give a total of 255^3 =16,777,216 different possible colours in the image.
- Since the total number of bits required for each pixel is 24.
- such images are also called 24-bit colour images.

True Color Image



49	55	56	57	52	53
58	60	60	58	55	57
58	58	54	53	55	56
83	78	72	69	68	69
88	91	91	84	83	82
69	76	83	78	76	75
61	69	73	78	76	76

Red

64	76	82	79	78	78
93	93	91	91	86	86
88	82	88	90	88	89
125	119	113	108	111	110
137	136	132	128	126	120
105	108	114	114	118	113
96	103	112	108	111	107

Green

66	80	77	80	87	77
81	93	96	99	86	85
83	83	91	94	92	88
135	128	126	112	107	106
141	129	129	117	115	101
95	99	109	108	112	109
84	93	107	101	105	102

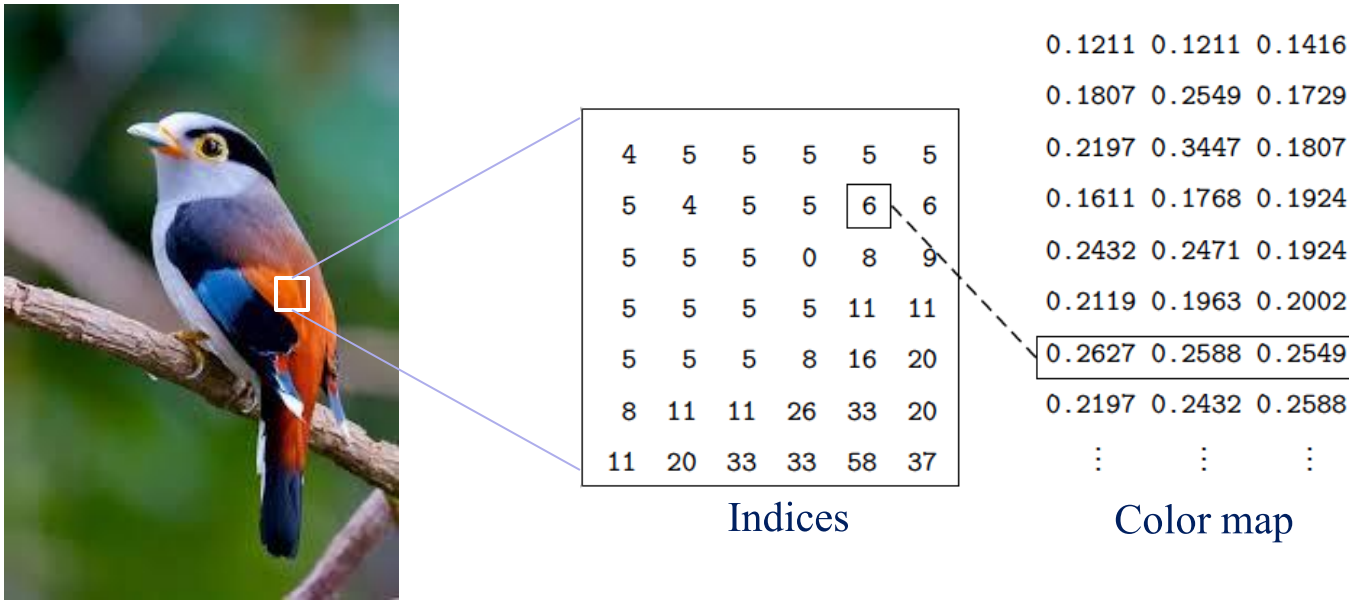
Blue

A true color image

Indexed colour image

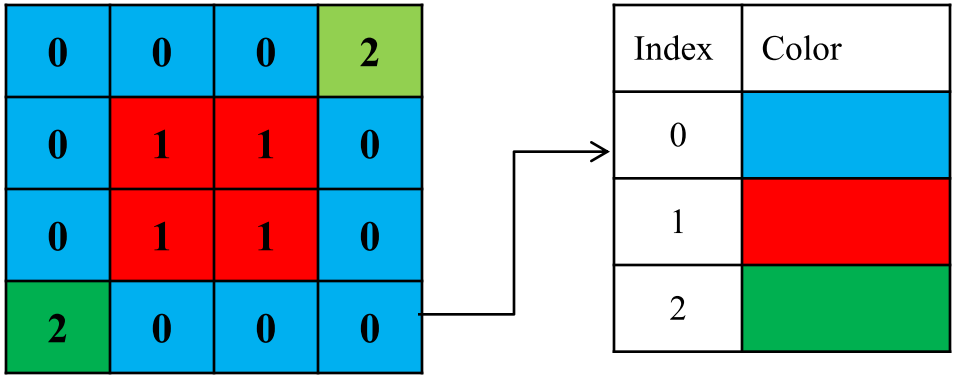
- Most colour images only have a small subset of the more than sixteen million possible colours.
- For convenience of storage and file handling, the image has an associated colour map or colour palette, which is simply a list of all the colours used in that image.
- Each pixel has a value which does not give its colour (as for an RGB image), but an index to the colour in the map.
- It is convenient if an image has 256 colors or less, for then the index values will only require one byte each to store.

Indexed color image



An indexed color image

Indexed color image with 3 gray level



Indexed color image with 4x4 pixels

Image Handling and I/O in Python

- Opening and Viewing Grayscale Images
  - Form the io module of skimage in Python

```
import skimage.io as io
img=io.imread('image.jpg')
```

  - This takes the gray values of all the pixels in the grayscale image image.jpg and puts them all into a matrix img.
  - Now we can display this matrix as a grayscale image in Python.

```
io.imshow(img)
```

  - However, an interactive display is possible using the ImageViewer method from the module

```
from skimage.viewer import ImageViewer as IV
Viewer = IV(img)
Viewer.show()
```

Reading, saving, and displaying an image using PIL

```
#Import libraries
from PIL import Image
# read the image# with the correct path
im = Image.open("images/flower1.jpg")
#display width, height and  image format
print(im.width, im.height, im.mode,
im.format, type(im))
# 225 225 RGB JPEG <class
'PIL.JpegImagePlugin.JpegImageFile'>
# display the image
im.show()
```

```
# convert the RGB color image to a grayscale image
im_g = im.convert('L')
 # save the image to disk
im_g.save('images/flower_gray.jpg')
# read the grayscale image from disk and show
Image.open("images/flower_gray.jpg").show()
```

Reading, saving, and displaying an image using PIL (cont.)

```
from PIL import Image
im = Image.open("images/cats.jpg")
# reflect about the vertical axis
im.transpose(Image.FLIP_LEFT_RIGHT).show()
```

Reading, saving, and displaying an image using Matplotlib

```
from matplotlib import pyplot as plt
import skimage.color as co
import numpy as np
# read image
img=np.uint8(plt.imread('images/orchid.jpg'))
# display image
plt.imshow(img)
imggrey=np.uint8(255*co.rgb2gray(img))
plt.figure()
plt.imshow(imggrey)
# save image
plt.imsave('images/orchid_grey.jpg',imggrey)
```

Original image

After grey conversion

Reading, saving, and displaying an image using scikit-image

```
from skimage.io import imread, imsave, imshow, show
from skimage import color,viewer
import matplotlib.pylab as plt
# read image from disk
im = imread("images/cats.jpg")
print(im.shape, im.dtype, type(im))
```

```
# from RGB to HSV color space
hsv = color.rgb2hsv(im)
# change the saturation
hsv[:, :, 1] = 0.5
# from HSV back to RGB
im1 = color.hsv2rgb(hsv)
# save image to disk
imsave('images/cats_hsv.jpg', im1)
im = imread("images/cats_hsv.jpg")
plt.axis('off'), imshow(im), show()
viewer = viewer.ImageViewer(im)
viewer.show()
```

Next Week Lecture (Week2)

- Lecture2:Image Enhancement in Spatial Domain (Point Processing)

Thank You