WORKSHOP IMAGE PROCESSING

Rieke Adriati Wijayanti, ST., MT.
Nurul Hidayati, ST., MT.
Ahmad Wilda Yulianto, ST., MT.

CERDAS ULET KREATIF
www.cerdas.co.id

Workshop Image Processing

Copyrights © 2022. All Rights Reserved
Hak cipta dilindungi undang-undang

Penulis:
Rieke Adriati Wijayanti, ST., MT.
Nurul Hidayati, ST., MT.
Ahmad Wilda Yulianto, ST., MT.

Penyunting:
Dhega Febiharsa, S.ST., M.Pd.

Desain & Tata Letak:
Tim Penerbit Cerdas Ulet Kreatif

ISBN :
Cetakan Pertama : **2022**
Penerbit :
Cerdas Ulet Kreatif
Jl. Manggis 72 RT 03 RW 04 Jember Lor - Patrang
Jember - Jawa Timur 68118
Telp. 0331-4431347, 412387 Faks. 4431347
e-mail : info@cerdas.co.id

Distributor Tunggal:
Cerdas Ulet Kreatif
Jl. Manggis 72 RT 03 RW 04 Jember Lor - Patrang
Jember - Jawa Timur 68118
Telp. 0331-4431347, 412387 Faks. 4431347
e-mail : info@cerdas.co.id

Undang-Undang RI Nomor 19 Tahun 2002
Tentang Hak Cipta

Ketentuan Pidana
Pasal 72 (ayat 2)

Barang Siapa dengan sengaja menyiarkan, memamerkan, mengedarkan, atau menjual kepada umum suatu ciptaan atau barang hasil pelanggaran Hak Cipta atau hak terkait sebagaimana dimaksud pada ayat (1), dipidana dengan pidana penjara paling lama 5 (lima) tahun dan/atau denda paling banyak Rp. 500.000.000,00 (lima ratus juta rupiah).

PREFACE

Alhamdulillah, we give all praise to Allah who always provides convenience and fluency in completing the textbook entitled Workshop Image Processing.

This textbook that is in your hands is a guide book used in the Workshop Image Processing course. This book contains the basics of digital image processing, operations on digital images to image segmentation which is the basis of pattern recognition.

Our deepest gratitude goes to all colleagues who have helped us and sincerely provided moral support in every stage of the process of making this book.

Finally, we wish you a happy reading and of course, we don't forget to expect criticism and suggestions so that we are always diligent in improving.

Malang, July 2022

Authors

TABLE OF CONTENT

PREFACE iii
TABLE OF CONTENT v
TABLE OF FIGURE viii
CHAPTER 1 FUNDAMENTAL OF DIGITAL IMAGE PROCESSING 1
A. What is Digital Image Processing 1
B. Fundamental Steps in Digital Image Processing 5
C. Element of Digital Image 8
D. Application of Digital Image Processing 10
E. Exercise: 12
CHAPTER 2 OPERATION IN DIGITAL IMAGE 15
A. Levels of Computation 15
B. Arithmetic Operations on Image 20
C. Geometry Operations on Image 23
D. Exercise: 27
CHAPTER 3 HISTOGRAM IMAGE 29
A. What Is Histogram 29
B. Benefits of Histogram 31
C. The Histogram Calculation 31
D. Histogram Equalization 35
E. Exercise: 39
CHAPTER 4 MORPHOLOGICAL IMAGE 41
A. Morphological Operation 41
B. Operation in Morphological Image 43
C. Structure Element 47
D. Exercise: 47
CHAPTER 5 SPATIAL DOMAIN AND FILTERING 49

A. Spatial Domain 49
B. Convolution Theory 50
C. Convolution in Two-Dimension Function 51
D. Spatial Filtering 55
E. Linear Spatial Filtering 55
F. Smoothing (Low pass) Spatial Filters 57
F. Exercise: 61
CHAPTER 6 IMAGE TRANSFORMATION 63
A. Preface 63
B. Fourier Transformation 64
C. Walsh-Hadamard Transformation 65
D. Hadamard Transformation 66
E. Exercise: 67
CHAPTER 7 IMAGE COMPRESSION AND DECOMPRESSION 69
A. Preface 69
B. Image Compression App 70
C. Image Compression Criteria 70
D. Classification of Compression Methods 73
E. Huffman Compression Method 75
F. Run-Length Encoding (RLE) Compression Method 79
G. Quantization Compression Method 81
H. Exercise: 83
CHAPTER 8 IMAGE SEGMENTATION 85
A. Preface 85
B. Pixel-Based Methods 85
C. Region-Based Methods 88
D. Edge-Based Methods 90
E. Exercise: 91

BIBLIOGRAPHY 93

GLOSSARY 94

TABLE OF FIGURE

Figure 1.1 (a) Original Image; (b) Green Channel; (c) Red Channel; (d) Blue Channel 2
Figure 1.2 Representation of pixels with color combination R, G, and B 3
Figure 1.3 RGB to grayscale conversion image 4
Figure 1.4 RGB to binary conversion image 5
Figure 1.5 Fundamentals Steps in Digital Image Processing 6
Figure 1.6 FaceApp application 11
Figure 1.7 Pattern Recognition 11
Figure 2.1 Point level operation on the digital image 15
Figure 2.2 Binary image of Lena 17
Figure 2.3 Negative image of Lena 17
Figure 2.4 (a) Original image; (b) After brightening 18
Figure 2.5 Local level operation 19
Figure 2.6 Edge detection of image Lena 19
Figure 2.7 Global level operation 20
Figure 2.8 (a) Original image; (b) Image translation with m = 30 and n = 25 24
Figure 2.9 Lena image rotation by 90° counterclockwise 25
Figure 2.10 (a) Zoom out with scale factor = 2; (b) Zoom in with scale factor = 1/2 26
Figure 2.11 (a) Image of the city of San Francisco (normal size); (b) Image of the city of San Francisco after enlarged 2 times 26
Figure 2.12 (a) Original image; (b) Flipping horizontal; (c) Flipping vertical 27
Figure 3.1 Image histogram 30
Figure 3.2 (a) Grayscale image; (b) Histogram of grayscale image 31
Figure 3.3 (a) Pepper colour; (b) Red channel histogram; (c) Green channel histogram; (d) Blue channel histogram 32
Figure 3.4 (a) Dark image; (b) Light image; (c) Normal brightness; (d) Normal brightness and high contrast 34
Figure 3.5 (a) Too dark image and it's histogram; (b) Too bright image and it's histogram; (c) Normal image and it's histogram 35
Figure 3.6 Original image and after histogram equalization 37
Figure 4.1 Original coin image 41
Figure 4.2 Object separation 42
Figure 4.3 Foming a filter 42

Figure 4.4 Image labelling....42
Figure 4.5 Object recognition....43
Figure 4.6 Example of kernel....43
Figure 4.7 Dilation operation....44
Figure 4.8 Erosion operation....45
Figure 4.9 Opening operation....46
Figure 4.10 Closing operation....47
Figure 4.11 Structure element....47
Figure 5.1 Illustration of Spatial Domain....49
Figure 5.2 Spatial domain for color image (RGB)....50
Figure 5.3 Illustration of convolution....51
Figure 5.4 The mechanics of linear spatial filtering using a 3 × 3 kernel. The pixels are shown as squares to simplify the graphics. 56
Figure 5.5 Examples of smoothing kernels: (a) is a box kernel; (b) is a Gaussian kernel.....58
Figure 5.6 (a) Test pattern of size 1024 × 1024 pixels; (b)-(d) Results of lowpass filtering with box kernels of sizes 3 × 3, 11 × 11, and 21 × 21, respectively.....59
Figure 6.1The difference between the Walsh matrix and Hadamard matrix....67
Figure 7.1 Original image before image compression with lossless and lossy method 74
Figure 7.2 After compression with lossless method 74
Figure 7.3 After compression with lossy method....75
Figure 8.1 Ideal image for segmentation: a) Image; b) Histogram with bimodal gray value distribution; c) Segmented image with the threshold indicated by the verticalline in the histogram 86
Figure 8.2 Segmentation of an image with uneven illumination: a) Image; b) Histogram; c) Segmented image....88
Figure 8.3 Region growing workflow 89
Figure 8.4 Region splitting and merging workflow....90

CHAPTER 1
FUNDAMENTAL OF DIGITAL IMAGE PROCESSING

After taking this chapter, students should:

- Understand the concept of digital image
- Understand the definition and scope of digital image processing
- Be familiar with the basic processes involved in image processing

A. What is Digital Image Processing

Digital Image Processing is a field of science that studies how an image is formed, processed, and analyzed so as to produce information that can be understood by humans. Digital Image Processing is a series of attempts to transform an image or image into another image using certain techniques. An image is a two-dimensional image that can be displayed on a computer screen as a set or discrete digital value called a pixel or image element. Mathematically, image is a continuous function of the intensity of light on a two-dimensional plane.

Based on the form of the constituent signal, image can be classified into two types, namely analog image and digital image. Analog image is an image formed from a continuous analog signal, while a digital image is an image formed from a discrete digital signal. Analog image is generated from analog image acquisition tools, examples of which are the human eye and analog cameras. Images captured by the human eye and photographs or films captured by analog cameras are examples of analog image. The image has quality with an excellent level of detail or resolution but has disadvantages including that it cannot be stored, processed, and duplicated inside the computer. Digital image is a representation of the function of light intensity in discrete form on a two-dimensional plane. An image is composed of a set of pixels (picture elements) that have coordinates (x,y) and amplitudes f(x,y). Coordinates (x,y) indicate the location/position of pixels in an image, while amplitude f(x,y) indicates the color intensity value of the image.

In general, based on the combination of colors on pixels, images are divided into three types, namely RGB image, grayscale image, and binary image.
The image in Figure 1 includes in the 24-bit true color RGB image type. The image is composed of three-color channels, namely the red canal, the green canal, and the blue canal.

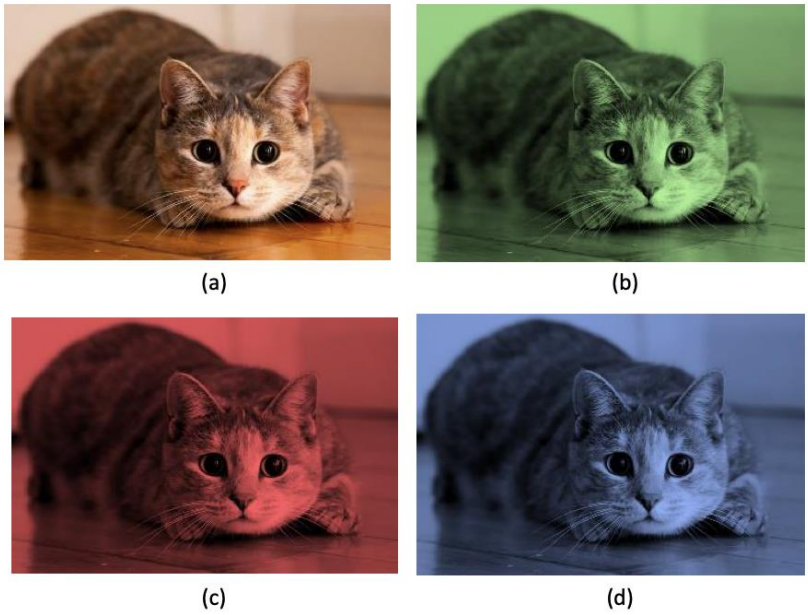


Figure 0.1 (a) Original Image; (b) Green Channel; (c) Red Channel; (d) Blue Channel

Each color channel has a pixel intensity value with a bit depth of 8-bit which means it has a color variation of 2^8 degrees of color (0 to 255). In the red channel, the perfect red color is represented with a value of 255 and perfect black with a value of 0. In the green channel, the perfect green color is represented with a value of 255 and perfect black with a value of 0. Likewise, in blue channels, the perfect blue color is represented with a value of 255 and perfect black with a value of 0. A representation of RGB image and each of its constituent color channels is shown in the following figure:

Figure 0.2 Representation of pixels with color combination R, G, and B

The second type of image is grayscale image. A grayscale image is an image whose pixel intensity value is based on the degree of grayness. In an 8-bit grayscale image, the degrees of black to white are divided into 256 degrees of grayness where the perfect black color is represented with a value of 0 and perfect white with a value of 255. RGB images can be converted to grayscale images so that only one-color channel is generated. The equation commonly used to convert a 24-bit true color RGB image into an 8-bit grayscale image is:

Grayscale = 0.2989*R + 0.5870*G + 0.1140*B

Where:

- scale is the grayscale image intensity value,
- R is the pixel intensity value on the red channel,
- G is the pixel intensity value on the green channel, and
- B is the pixel intensity value on the blue channel.

The image of the RGB to grayscale conversion result is shown in figure below:

Figure 0.3 RGB to grayscale conversion image

The third type of image is binary image. Binary image is an image whose pixels have a bit depth of 1 bit so that it only has two color intensity values, namely 0 (black) and 1 (white). Grayscale images can be converted to binary images through the thresholding process. In the thresholding process, a threshold value is needed as a conversion limiting value. A pixel intensity value greater than or equal to the threshold value will be converted to 1. While the pixel intensity value that is less than the threshold value will be converted to 0. For example, the threshold value used is 128, then pixels that have an intensity of less than 128 will be converted to 0 (black) and those more than or equal to 128 will be converted to 1 (white).

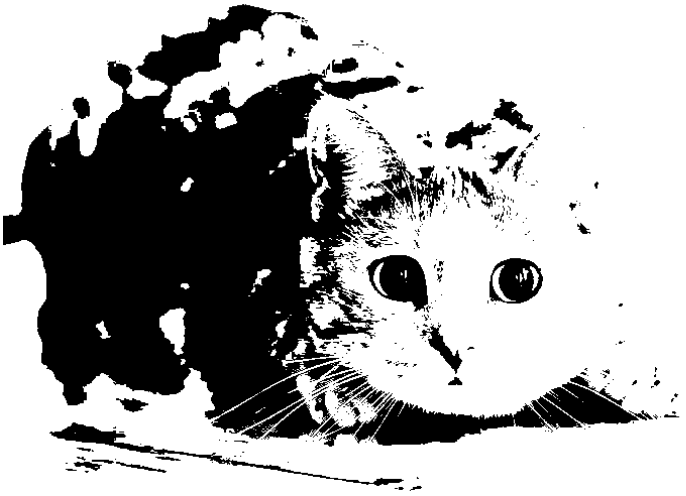


Figure 0.4 RGB to binary conversion image

Thresholding is generally used in the process of image segmentation. The process is done to separate the foreground (desired object) with the background (other unwanted objects). In segmentation results, foreground is represented by white (1) and background is represented by black (0). In the case of segmentation on one image only, we can determine the threshold value with the trial and error method. But in the case of segmentation on a large number of images, a method is needed to determine the threshold value automatically. Threshold values can be obtained automatically using the Otsu method (1979). Digital image is formed by a collection of dots called pixels (pixels or ―picture elements‖). Each pixel is described as one small square. Each pixel has position coordinates.

B. Fundamental Steps in Digital Image Processing

There are several fundamental steps required in digital image processing shown in below figure:

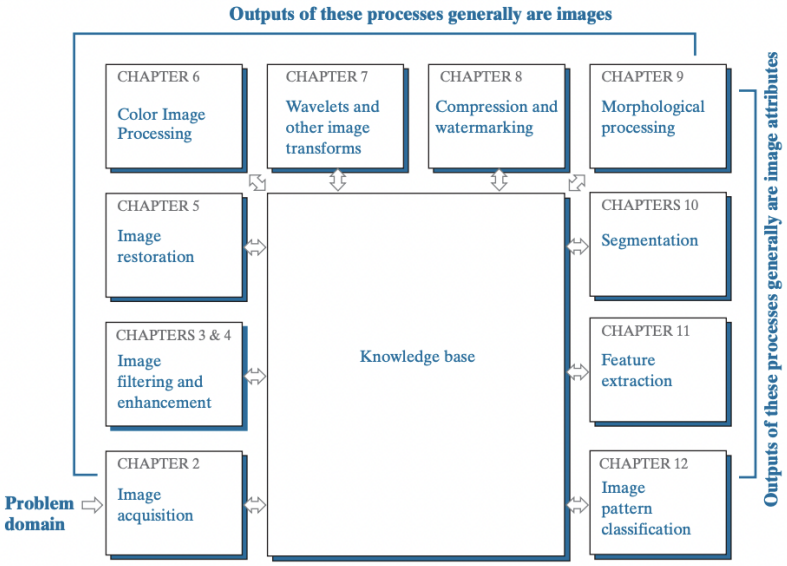


Figure 0.5 Fundamentals Steps in Digital Image Processing

Acquisition could be as simple as being given an image that is already in digital form. Generally, the image acquisition stage involves preprocessing, such as scaling. Image enhancement is the process of manipulating an image so the result is more suitable than the original for a specific application. The word specific is important here, because it establishes at the outset that enhancement techniques are problem oriented. Thus, for example, a method that is quite useful for enhancing X-ray images may not be the best approach for enhancing satellite images taken in the infrared band of the electromagnetic spectrum. There is no general “theory” of image enhancement. When an image is processed for visual interpretation, the viewer is the ultimate judge of how well a particular method works.

Enhancement techniques are so varied, and use so many different image processing approaches, that it is difficult to assemble a meaningful body of techniques suitable for enhancement in one chapter without extensive background development. For this reason, and also because beginners in the field of image processing generally find enhancement

applications visually appealing, interesting, and relatively simple to understand.

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation. Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a “good” enhancement result. Color image processing is an area that has been gaining in importance because of the significant increase in the use of digital images over the internet. Color is used also as the basis for extracting features of interest in an image. Wavelets are the foundation for representing images in various degrees of resolution. In particular, this material is used in the book for image data compression and for pyramidal representation, in which images are subdivided successively into smaller regions.

Compression, as the name implies, deals with techniques for reducing the storage required to save an image, or the bandwidth required to transmit it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. This is true particularly in uses of the internet, which are characterized by significant pictorial content. Image compression is familiar (perhaps inadvertently) to most users of computers in the form of image file extensions, such as the jpg file extension used in the JPEG (Joint Photographic Experts Group) image compression standard. Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape.

Segmentation partitions an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually. On the other hand, weak or erratic

segmentation algorithms almost always guarantee eventual failure. In general, the more accurate the segmentation, the more likely automated object classification is to succeed. Feature extraction almost always follows the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region (i.e., the set of pixels separating one image region from another) or all the points in the region itself. Feature extraction consists of feature detection and feature description.

Feature detection refers to finding the features in an image, region, or boundary. Feature description assigns quantitative attributes to the detected features. For example, we might detect corners in a region, and describe those corners by their orientation and location; both of these descriptors are quantitative attributes. Feature processing methods discussed in this chapter are subdivided into three principal categories, depending on whether they are applicable to boundaries, regions, or whole images. Some features are applicable to more than one category. Feature descriptors should be as insensitive as possible to variations in parameters such as scale, translation, rotation, illumination, and viewpoint. Image pattern classification is the process that assigns a label (e.g., “vehicle”) to an object based on its feature descriptors. In the last chapter of the book, we will discuss methods of image pattern classification ranging from “classical” approaches such as minimum-distance, correlation, and Bayes classifiers, to more modern approaches implemented using deep neural networks. In particular, we will discuss in detail deep convolutional neural networks, which are ideally suited for image processing work.

C. Element of Digital Image

Digital image contains a number of basic elements. Basic elements it is manipulated in image processing and further exploited in computer vision. The basic elements that are important include:

(1) Brightness

Brightness is another word for light intensity. As explained in the inletting section, the brightness at a point (pixel) in the image it is not a real intensity, but it is actually the average intensity of an area that covers it. The human visual system is able to adjust himself with brightness level starting from the lowest to the highest with a range of 10^{10}.

(2) Contrast

Contrast expresses the spread of light (lightness) and darkness (darkness) inside a picture. Low contrast imagery is characterized by most the composition of the image is light or mostly dark. On the image with good contrast, dark and light composition is evenly distributed.

(3) Contour

Contour is a state brought about by a change in intensity in pixels the neighbors. It is because of this change in intensity that our eyes able to detect the edges of objects in the image.

(4) Color

Color is the perception perceived by the human visual system towards lengt light waves reflected by the object. Each color has a length. The wave (λ) is different. The red color has the highest wavelength, while the color purple (violet) has the lowest wavelength. The colors received by the eye (human visual system) are the result the combination of light with different wavelengths. Research shows that color combination that gives the widest color range Red (R), Green (G), and Blue (B). The perception of the human visual system towards color is very relative because it is influenced. by many criteria, one of which is caused by adaptations that give rise to Distortion. For example, gray patches around the green color will appear purplish (distortion of space), or if the eye sees green and then directly quickly see the gray color, then the eyes catch the impression the gray color as purple (distortion of time).

(5) Shape

Shape is an intrinsic property of a three-dimensional object, with the understanding that Shape is a major intrinsic property for the human visual system. Humans more often associate objects with their shape than other elements

(colors for example). In general, the image formed by the eye is a 2-dimensional image, while the object seen is generally 3-dimensional. Object shape information can be extracted from imagery at the beginning of pre-processing and image segmentation. One of the main challenges thing in computer vision is to represent the shape, or aspects of Important of form.

(6) Texture

Texture is characterized as the spatial distribution of the degree of grayness inside a bunch of neighboring pixels. So, texture cannot Defined for a pixel. The human visual system is not receive image information independently on each pixel, but rather considered an image as a whole. The resolution of the observed image is determined by the scale on which the texture is perceived. For example, if we observe the image of the tiled floor remotely, then we observe that texture is formed by the placement of the tiles as a whole, not from perception of patterns within the tile itself. However, if we look at the image the same from a close distance, then only a few tiles appear in Midwives observe, so we perceive that texture is formed by placement of detailed patterns that make up each tile.

D. Application of Digital Image Processing

The use of digital image processing technology is widely attributed to intelligent systems, nowadays it is not foreign for us to see even using image processing and image processing in everyday life. Starting from the features on cameras and videos on smartphones that use artificial intelligence such as cameras that can produce images with unique and interesting filters, CCTV that can monitor movement and so on. Some applications of image processing include being applied to the fields of: Military, medical, biology, education, geography and geology, policing or law, commerce, entertainment and data communication.

(1) In 2020 there are at least dozens of free and paid applications that apply Detection, classification and identification of objects, some of these applications have

shocked the world because of their uniqueness in implementing image object identification. Here are the applications that are widely used, namely Luxad, Blippar, Face2gene, FaceApp, Applock, FacePhi, Log Me, FaceFirst, Pictriev, BetaFace, Kuznech Face Detection and others.

Figure 0.6 FaceApp application

(2) Pattern recognition is a discipline that classifies objects based on image, weight or predetermined parameters into a number of categories or classes.

(3) Machine Vision Pattern recognition becomes the basis of this machine system. This machine captures a group or group of objects with a camera and is then analyzed to describe the object or object.

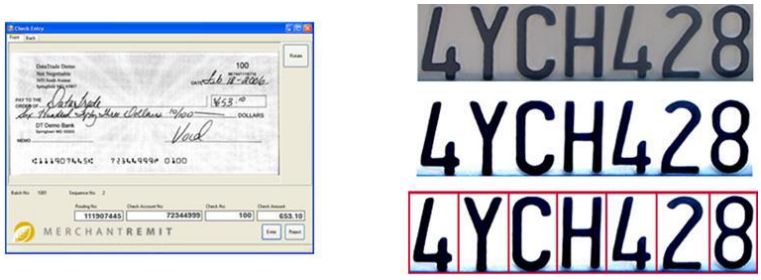


Figure 0.7 Pattern Recognition

(4) Character recognition (OCR) One of the areas of pattern recognition that generally addresses automation and information issues. The OCR system has a front end device

consisting of a light generator, a scanning lens, a document transport and a detector.

(5) Computer aided diagnosis This system helps doctors in making decisions about a diagnosis.

(6) Speech recognition Speech recognition Is one of the applications that are growing today. This system allows us to communicate between humans by entering data into the computer. Increase the efficiency of the manufacturing industry, controlling the machine by talking to that machine.

(7) Face recognition Is a system that recognizes human facial images used in the automation and security of an industry.

(8) Biometrics are used to recognize a pattern of living things that are connected with psychological and behavioral parameters

(9) Image Data Base Retrieval
Image Data Base Retrieval is a system for the return of data base image. A shooting system used to browse, search and retrieve images from a large database of digital images. The most traditional and common shooting methods use several methods of adding metadata such as text, keywords, titles, or descriptions to images so that capture can be done on annotation words.

(10) Data mining is the grouping of object patterns of a number of data that are sorted in the hope of providing useful and desirable information.

(11) Bioinformatics is closely related to the discipline of medicine, the recognition of patterns or images of a disease image or pattern in an analysis of disease diagnosis or pattern recognition related to the biological world in general.

E. Exercise:

Select one of the correct answers by marking the answer sheet.

1. How does Computer Vision work?
   a. Identify and process objects
   b. Count and filter text
   c. Counting Images
   d. Save an Image file

e. Create a 3D image

2. Computer vision divides the images that have been inserted into a form, what form does the computer translate to process the image?
   a. Text Form
   b. Audio Form
   c. Image Shape
   d. Pixel Shape and number shape
   e. Shape

3. In general, based on the color combination on the pixels, the image is divided into several types, namely the image...
   a. CMYK, RGB, Citra grayscale
   b. Binary Image and Analog Image
   c. RGB, grayscale image, and binary image
   d. Grayscale image and Analog image
   e. Binary image, grayscale image and analog image

4. If the yellow representation has a pixel intensity value of R:255 G:255 B:0, then what is the pixel intensity value of White...
   a. R: 0 G:255 B:0
   b. R: 0 G:255 B:255
   c. R: 102 G:255 B:204
   d. R: 255 G:255 B:255
   e. R: 204 G:102 B:204

5. What does not belong to the color type is ...
   a. Biner
   b. RGB
   c. Grayscale
   d. CMYK
   e. Pixel

CHAPTER 2
OPERATION IN DIGITAL IMAGE

After taking this chapter, students should:

- Understand the operation used in digital image processing
- Understand the kind of operation in digital image processing
- Understand how to measure the operation in digital image

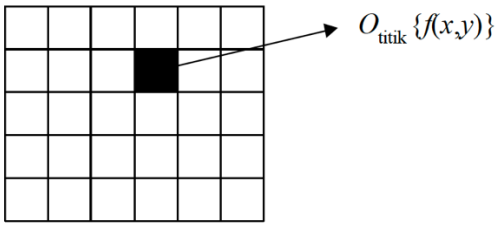
A. Levels of Computation

Digital image is represented by a matrix. Basically, operations on digital images is manipulating the elements of the matrix. Matrix elements manipulated can be a single element (a pixel), a set of adjacent elements, or the entire matrix element. Inside this chapter will outline the basic operations on the processing of digital image. Operations performed on image processing can be grouped into in four levels of computation, namely point level, local level, global level, and object level.

(1) Point Level

Operations at the point level are performed only on a single pixel in the image. Point operation is also known as the pointwise operation. This operation consists of accessing pixels at a given location, modifying them by operation linear or nonlinear operation, and place pixel values new to the corresponding location in the new image. This operation is repeated for the entire pixel in the image.

Mathematically, the operation at the point level is expressed as shown in below figure:



$$f_B(x, y) = O_{titik}\{ f_A(x, y) \}$$

Figure 0.1 Point level operation on the digital image

which in this case f_A and f_B are input image and image respectively output, O_{titik} can be either linear or nonlinear operation.

What is meant by linear operation is an operation that can be declared Mathematically as a linear equation, the opposite is an equation nonlinear. Operations at the point level can be divided into three kinds: based on intensity, based on geometry, or a combination of the two.

a. Based on intensity.

The intensity value of u a pixel is changed by transforming h into a new intensity value v:

v = h(u), u, v ∈ [0, L]

An example of a point operation based on intensity is a thresholding operation. In the seeding operation, the pixel intensity value is mapped to one of two values, a1 or a2, based on the threshold value T:

$$f(x, y)' = \begin{cases} a_1, & f(x,y) < T \\ a_2, & f(x,y) \geq T \end{cases}$$

If a_1 = 0 and a_2 = 1, then the anchoring operation transforms the black-and-white image to a binary image. In other words, the pixel intensity value was originally mapped to two values only: black and white. The threshold value used can apply to the entire pixel or to a specific region only (based on the spread of the intensity value in that region).

The pairing operation on Lena's image with the transformation function:

$$f(x, y)' = \begin{cases} 0, & f(x,y) < 128 \\ 1, & f(x,y) \geq 128 \end{cases}$$

generates a binary image as shown in Figure xx. Equation above states that pixels whose intensity values are below 128 are converted to black (intensity value = 0), while pixels whose intensity values are above 128 are converted to white (intensity value = 1).

Figure 0.2 Binary image of Lena

Other examples of point operations are:

- Negative operation, namely getting a negative image (negative image) mimics a negative film in photography by reducing the pixel intensity value from the maximum grayness value.

Figure 0.3 Negative image of Lena

- Cutting (clipping)
  This operation is performed if the pixel intensity value of an image processing operation is located below the minimum intensity value or above the maximum intensity value.

$$f(x, y)' = \begin{cases} 255, & f(x,y) > 255 \\ f(x, y), & 0 \le f(x,y) \le 255 \\ 0, & f(x,y) < 0 \end{cases}$$

- Image brightening
  The brightness of an image can be improved by adding (or subtracting) a constant to (or from) each pixel in the image. Mathematically this operation is written as f(x, y)' = f(x, y) + b (4.8) If b is positive, the brightness of the image increases, otherwise if b negative the brightness of the image decreases. See an example of image enlightenment in figure below applied to Zelda imagery. Originally Zelda's image appeared dark, but by adding each pixel value with b = 10, Zelda's image became lighter.

(a) (b)

Figure 0.4 (a) Original image; (b) After brightening

b. Based on geometry
The pixel position is changed to a new position, while the intensity does not change. Examples of point operations based on geometry such as playback (rotation), shift (translation), scaling (dilation), correction of innards (distortion) geometry.

c. Combined intensity and geometry
This operation not only changes the pixel intensity value, but also changes its position. For example, image morphing, which is a change in the shape of an object and its intensity value.

(2) Local Level

Operation at the local level produces an output image whose intensity of a pixel depends on the intensity of its neighboring pixels.

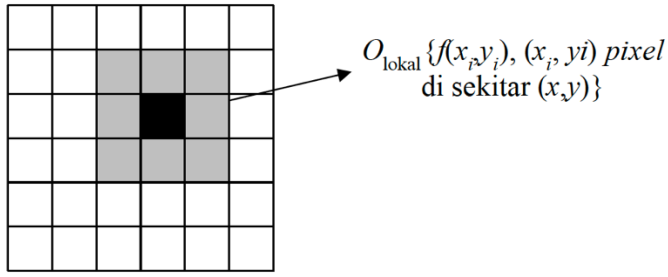


Figure 0.5 Local level operation

$$f_B(x, y)' = O_{local}\{f_A(xi, yj);\quad (xi, yj) \in N(x, y)\}$$

(description: N = neighborhood, i.e. pixels located around (x, y))

Examples of locally adjusted operations are convolution operations to detect edges (edge detection) and image softening (image smoothing). This figure below shows an image of Lena from edge detection.

Figure 0.6 Edge detection of image Lena

(3) Global level

Operations on the global level produce output image whose intensity of a pixel depends on the overall intensity of the pixel.

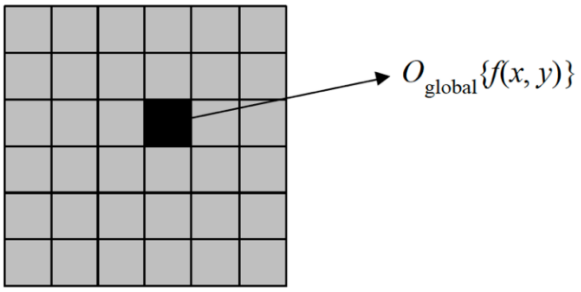


Figure 0.7 Global level operation

$$f_B(x, y)' = O_{global}\{f_A(x, y)\}$$

An example of a globally adjusted operation is a histogram equalization operation to improve image quality.

(4) Object Level
is only performed on certain objects in the image. The purpose of the operation at the level of the object is to recognize the object, for example by calculating the average intensity, size, shape, and other characteristics of the object. The operation of the level of the object is a very difficult operation, because before we must be able to answer: what is the object, how to find it?

B. Arithmetic Operations on Image

Since the digital image is a matrix, the arithmetic operations of the matrix also apply to the image. Matrix operations that can be performed are:

1. Addition or subtraction between two images A and B:
   C (x, y) = A (x, y) ± B (x, y)
2. Multiplication of two images:
   C (x, y) = A (x, y) B (x, y)
3. Addition/subtraction of image A with scalar c:
   B (x, y) = A (x, y) ± c
4. Multiplication/division of image A with a scalar c:
   B (x, y) = c • A (x, y)

Reviewed from the computational level, arithmetic operations are included in the point level operation. The explanation of each arithmetic operation of the matrix is as follows:

(1) The addition of two image

Equations: **C (x, y) = A (x, y) + B (x, y)**

C is a new image whose intensity of each pixel is the sum of the intensity of each pixel in A and B. If the result of the addition of intensity is greater than 255, then the intensity is rounded to 255.

Image addition operations can be used to reduce the effect of noise in the data, by averaging the degree of validity of each pixel of the same image taken many times.

For example for the same image recorded twice, f_1 and f_2, then calculated the average intensity for each pixel:

$$f'(x,y) = \frac{1}{2}\{ f_1(x, y) + f_2(x, y) \}$$

The result of the operation may be of real value, therefore all such real values need to be rounded to the nearest round value, the maximum value is 255.

(2) Reduction of Two Image

Equations: **C (x, y) = A (x, y) – B (x, y)**

C is a new image whose intensity of each pixel is the difference between the pixel intensity on A and B. It is possible that the result of this operation produces a negative value, therefore, the image reduction operation needs to involve clipping operations.

An example of an image reduction operation application is to obtain an object from two images. The first image is for example a photo of an empty room, the second image is a photo of the same room but there are people in it. The result of the second image reduction with the first image produces an image whose background is black, while the foreground (person object) is white. Image reduction can also be used to detect changes that occur during a certain interval of time when the two images taken are images from the same scene. This kind of technique is used in moving images.

(3) Image Multiplication

The equation: **C (x, y) = A (x, y) ● B (x, y)**

Image multiplication is often used to correct sensor delivery by multiplying the image matrix by the correction matrix. So, in this case A is the image while B is the correction matrix. The result of the operation may be of real value, therefore all values are rounded to the nearest round value, the maximum value is 255.

For example multiplying image A by correction matrix:

$$\begin{bmatrix} 0 & 12 & 142 & 255 \\ 1 & 6 & 40 & 254 \\ 24 & 0 & 20 & 255 \\ 30 & 2 & 10 & 240 \end{bmatrix} \begin{bmatrix} 0.3 & 0.4 & 0.1 & 0.1 \\ 0.3 & 0.0 & 0.0 & 0.1 \\ 0.3 & 0.0 & 0.0 & 0.0 \\ 0.4 & 0.1 & 0.0 & 0.1 \end{bmatrix} = \begin{bmatrix} 0 & 17 & 157 & 255 \\ 2 & 6 & 40 & 255 \\ 32 & 0 & 20 & 255 \\ 42 & 3 & 10 & 255 \end{bmatrix}$$

Matriks citra A Matriks koreksi B Matriks keluaran C

(4) Addition/subtraction of image by scalar

The equation: **B (x, y) = A (x, y) ± c**

The addition of image A with scalar c is to add each pixel in the image of a scalar c, and produce a new image B whose intensity is lighter than A. The increase in intensity is the same for the entire pixel, i.e. c. The reduction of image A with scalar c is to subtract each pixel in the image of a scalar c, and produce a new image of B whose intensity is darker than A. The decrease in intensity is the same for the entire pixel, i.e. c.

An example of an image addition/reduction operation with a scalar is an image enlightenment operation (see discussion of point level operations). Both addition and image reduction operations with a scalar involve clipping operations.

(5) Multiplication/division of Image with Scalar

Equations: **B (x, y) = c ● A (x, y), and B (x, y) = A (x, y) / c**

Multiplication of image A with scalar c produces a new image B whose intensity is lighter than A. The increase in intensity of each pixel is proportional to c. Image multiplication operations with scalars are used for brightness calibration (callibration of brightness). The

division of image A with scalar c produces a new image of B whose intensity is darker than A. The decrease in intensity of each pixel is inversely proportional to c. Image division operations with scalars are used for brightness normalization.

C. Geometry Operations on Image

In geometric operations, pixel coordinates change due to transformation, while the intensity remains. This is different from arithmetic surgery where pixel coordinates remain while the intensity changes.

Geometric operations performed such as translation, rotation, image scaling, and image mirroring (flipping). Change geometry from f (x, y) image to image the new f' (x, y) can be written as:

$$\mathbf{f'(x', y') = f (g1 (x, y), g2 (x, y))}$$

in this case, g1(x) and g2(y) are geometric transformation functions. With other words,

x’ = g1(x, y);

y’ = g2(x, y)

(1) Translation

Image translation formula:

$$\mathbf{x' = x + m}$$

$$\mathbf{y' = y + n}$$

which in this case, m is the large shift in the direction of x, while n is Great shift in direction y. If the original image is A and the image is translational result is B, then translation can be implemented by copying the image from A to B:

$$B[x][y] = A [x + m][y + n]$$

Image translation on the camera image is shown in figure below:

(a) (b)

Figure 0.8 (a) Original image; (b) Image translation with m = 30 and n = 25

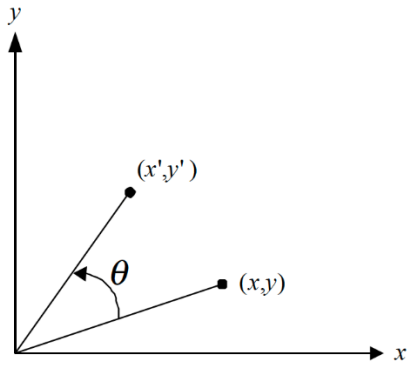
(2) Rotation

Image rotation formula:

$$x' = x\cos(\theta) - y\sin(\theta)$$
$$y' = x\sin(\theta) + y\cos(\theta)$$

which in this case, θ = the angle of rotation counter clockwise. If the original image is A and the image of the rotation result is B, then the rotation of the image from A to B:

$$B[x'][y'] = B[x\cos(\theta) - y\sin(\theta)][x\cos(\theta) + y\cos(\theta)] = A[x][y]$$



If the rotation angle is 90°, then the implementation is easier to do with how to copy row pixels to column pixels in the direction of rotation. 180° rotation is implemented by rotating 90° twice.

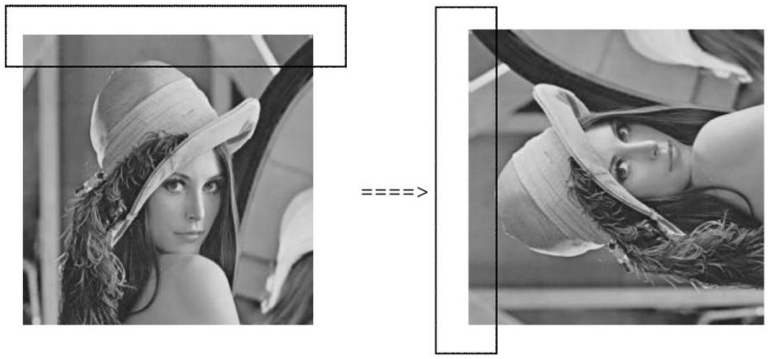


Figure 0.9 Lena image rotation by 90° counterclockwise

(3) Image Scaling

Image scaling, also called image zooming, is resizing the image (zoom out or shrink/zoom in).

Image scaling formula:

$$\mathbf{x' = s_x \bullet x}$$

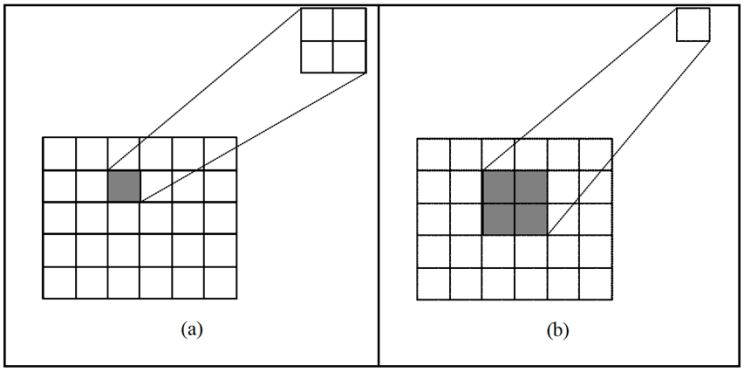
$$\mathbf{y' = s_y \bullet y}$$

which in this case, s_x and s_y are the scale factors respectively in the direction of x and direction y.

If the original image is A and the image of the scaling result is B, then the scaling image is expressed as:

$$B[x'][y'] = B[s_x \cdot x][\, s_y \cdot y] = A[x][y]$$

Zoom out operations by a factor of 2 (i.e., sx = sy = 2) are implemented with copy each pixel 4 times. So, an image of 2 x 2 pixels will be 4 x 4 pixels. An example of a double-magnified image is shown in Figure below:



(a) (b)

Figure 0.10 (a) Zoom out with scale factor = 2; (b) Zoom in with scale factor = 1/2

Zoom in (shrinking) operation with scale factor = 1/2 is performed with takes the average from 4 neighboring pixels to 1 pixel.

(a) (b)

Figure 0.11 (a) Image of the city of San Francisco (normal size); (b) Image of the city of San Francisco after enlarged 2 times

(4) Flipping

Flipping is the same geometry operation as mirroring (image reflection). There are two types of flipping: horizontal and vertical.

(a) (b) (c)

Figure 0.12 (a) Original image; (b) Flipping horizontal; (c) Flipping vertical

Horizontal flipping is the mirroring on the Y-axis (cartesian) of image A become image B, given by:

$$B[x][y] = A[N - x][y]$$

Vertical flipping is the mirroring on the X-axis (cartesian) of image A become image B, given by:

$$B[x][y] = A[x][M - y]$$

Mirroring at the point of origin (cartesian) from image A to image B is given by:

$$B[x][y] = A[N - x][M - y]$$

Mirroring on the line x = y from image A to image B is given by:

$$B[x][y] = A[y][x]$$

D. Exercise:

Select one of the correct answers by marking the answer sheet.

1. Image elements that can be affected by arithmetic operations are:
   a. Image brightness
   b. Image size
   c. Image position
   d. Contour image
   e. The structure of the object in the picture

2. The following statements that correspond to the addition operation on the image are:

a. The image becomes brighter
b. The contrast of the image changes to low
c. The image turns darker
d. The contrast of the image changes to high
e. Image brightness changes to normal

3. Which does not include geometric operations on the image, namely:
   a. Reflection
   b. Translation
   c. Rotation
   d. Dilation
   e. Dilation
4. If it is known that the result of the translation of point B is (9,6) with the translational component is (3,2), then the starting point of B is:
   a. (-12,-8)
   b. (-6,-4)
   c. (3,3)
   d. (12,8)
   e. (6.4)

5. The dilation of point A(4,7) to the center of P(2,5) with a scale factor of 2 is:
   a. (6,-9)
   b. (-2,-5)
   c. (8, 17)
   d. (6,9)
   e. (8.35)

CHAPTER 3
HISTOGRAM IMAGE

After taking this chapter, students should:

- Understand the meaning and use of a histogram.
- Able to display image histograms using software.
- Able to modify the image histogram in image processing

A. What Is Histogram

Important information about the content of digital image can be known by creating image histogram. An image histogram is a graph that describes spread of pixel intensity values of a particular image or part in in imagery. From a histogram can be known the frequency of the appearance of nisbi (relative) of the intensity of the image. Histograms can also show There are many things about the brightness and contrast of a picture.

Therefore, histograms are a valuable tool in work image processing both qualitatively and quantitatively. Suppose the digital image has L degree of validity, which is from the value of 0 to L – 1 (for example in an image with quantization of 8-bit grayish degrees, degree values grayness from 0 to 255). Mathematically the image histogram is calculated by formula which in this case:

ni = number of pixels that have a degree of grayness i

n = the number of all pixels in the image

The plot hi versus fi is called histogram. Figure 3.1 is an example of an image histogram.

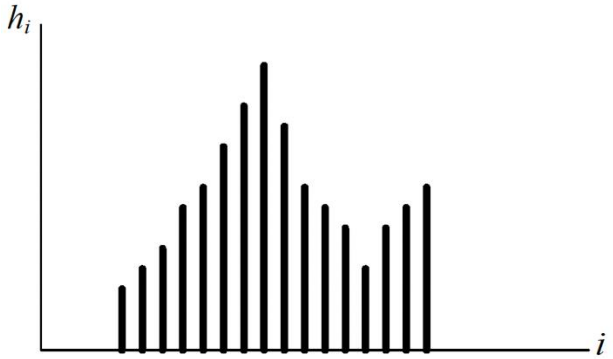


Figure 0.1 Image histogram

Graphically the histogram is displayed with a bar chart. The value ni has been normalized with Divide it by n. The value hi is inside intervals 0 to 1.

For example, suppose the matrix below expresses a defilement image that size 8 x 8 pixels with a degree of grayness from 0 to 15 (there are 16 pieces degree of grayness):

$$\begin{bmatrix} 3 & 7 & 7 & 8 & 10 & 12 & 14 & 10 \\ 2 & 0 & 0 & 0 & 1 & 8 & 15 & 15 \\ 14 & 6 & 5 & 9 & 8 & 10 & 9 & 12 \\ 12 & 12 & 11 & 8 & 8 & 10 & 11 & 1 \\ 0 & 2 & 3 & 4 & 5 & 13 & 10 & 14 \\ 4 & 5 & 0 & 0 & 1 & 0 & 2 & 2 \\ 15 & 13 & 11 & 10 & 9 & 9 & 8 & 7 \\ 2 & 1 & 0 & 10 & 11 & 14 & 13 & 12 \end{bmatrix}$$

The tabulation of the histogram calculation is shown in below table. Easy to see that the greater the value of ni, the greater the value of hi.

i	ni	hi = ni/n (n = 64)
0	8	0.125
1	4	0.0625
2	5	0.078125
3	2	0.03125
4	2	0.03125
5	3	0.046875
6	1	0.015625
7	3	0.046875
8	6	0.09375
9	3	0.046875
10	7	0.109375

11	4	0.0625
12	5	0.078125
13	3	0.046875
14	4	0.0625
15	3	0.046875

B. Benefits of Histogram

The benefits that can achieved from histogram are:

- Useful for observing the spread of color intensity and can be used for decision making, for example in increased brightness or stretching contrast and color distribution.
- Useful for the determination of boundaries in separation object from the background.
- Provide a percentage of color composition and texture intensity for the sake of image identification.
- Histogram does not reflect the arrangement of color positions pixels in the image. Therefore, the histogram cannot be used to guess the shape of an object contained in the image.

C. The Histogram Calculation

The input image has 256 degrees of grayness whose values are from 0 to 255. Pixel intensity is stored in image [0…N-1][0…M-1], while the histogram is stored in the Hist table[0..255]. Figure 3.2 shows the histogram of the ship's image (512 x 512). Some commercial programs such as Adobe Photoshop, Paintshop, and PolyView, can used to evoke image histograms.

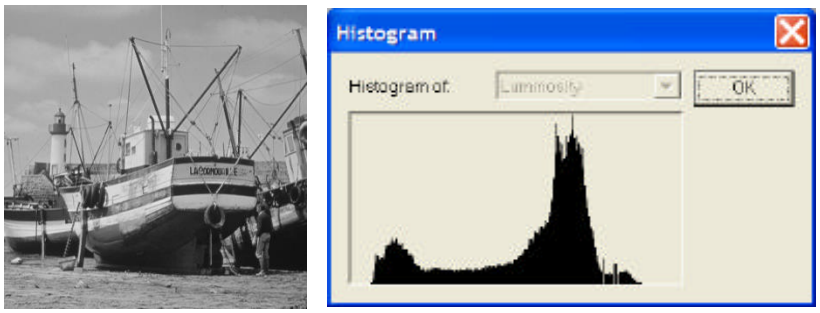
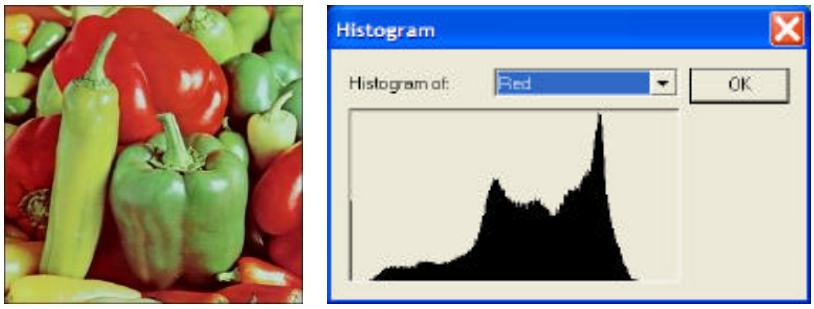


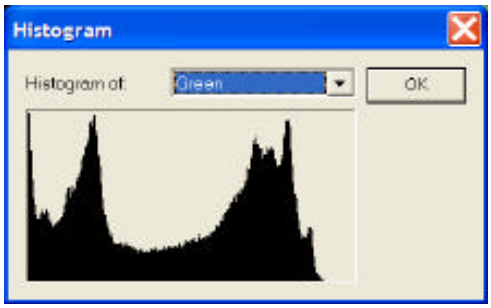
Figure 0.2 (a) Grayscale image; (b) Histogram of grayscale image

Specifically, for color imagery, the histogram is created for each RGB channel (red, green, and blue). For example, pepper color

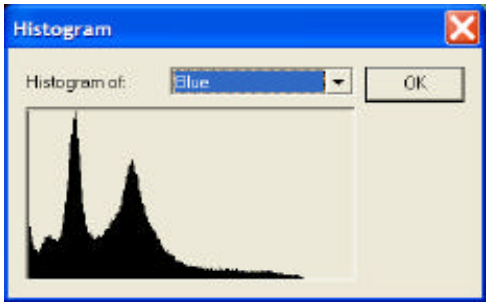
image 512 x 512 8-bit pixels, in Figure 3.3, the histogram has three pieces, each for a component red, green, and blue. The histogram is generated from the program.



(a) (b)



(c)



(d)

Figure 0.3 (a) Pepper colour; (b) Red channel histogram; (c) Green channel histogram; (d) Blue channel histogram

Image histograms provide a lot of important information as follows:

(1) The value hi expresses the probability (probability) of the pixel, P(i), with the degree of validity i. The sum of the entire value of hi is equal to 1, or

$$\sum_{i=0}^{L-1} h_i = 1$$

The chance of a pixel having a degree of grayness is smaller or equal to

$$P(i \le j) = \sum_{i=0}^{j} h_i \,, \qquad 0 \le j \le L-1$$

(2) The peak of the histogram indicates the intensity of the pixel that stands out. Width of the peak shows the contrast range of the image. An image that has contrast is too bright (overexposed) or too dark (underexposed) has a narrow histogram. The histogram looks only using half of the area of the degree of grayness. A good image has a histogram which fills the area of the degree of grayness in full with the distribution that evenly distributed at each pixel intensity value shown in Figure 3.4 below:

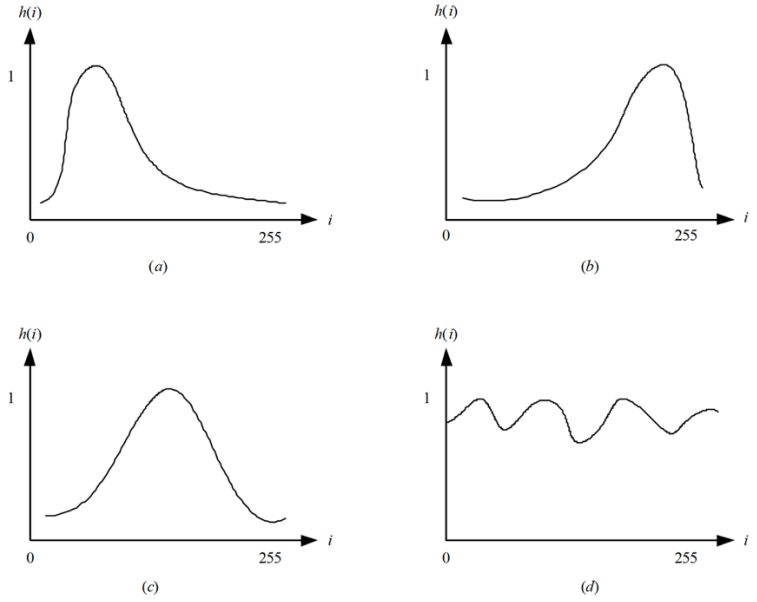
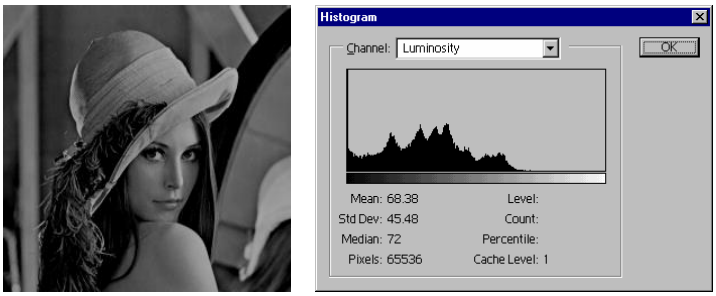
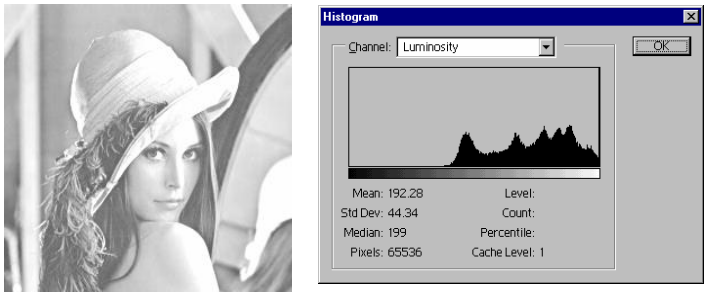


Figure 0.4 (a) Dark image; (b) Light image; (c) Normal brightness; (d) Normal brightness and high contrast

Figure 3.5 shows three images of Lena. The first image of Lena too dark. The histogram accumulates a lot on the left because of the imagery it contains many intensity values close to 0 (black). Image The second Lena was too bright. The histogram accumulates a lot on the part right because the image contains a lot of close intensity values 255 (white). Lena's third image is a normal (good) image.
The histogram is evenly distributed throughout the area of the degree of grayness. Three pieces the histogram is generated with the Adobe Photoshop program.



(a)



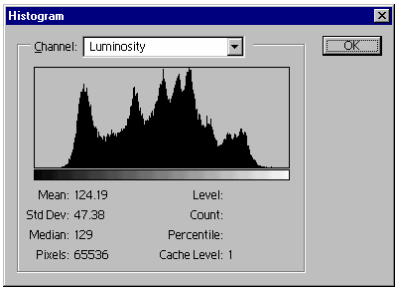
(b)

(c)

Figure 0.5 (a) Too dark image and it's histogram; (b) Too bright image and it's histogram; (c) Normal image and it's histogram

D. Histogram Equalization

The histogram of an image can be modified to obtain the image as desired or improve the quality of the image. One way that can be used to modify the image histogram is histogram *equalization*. Histogram equalization is a process that changes the distribution of the degree of grayness value in an image so that it becomes uniform. The goal of histogram equalization is to obtain an even spread of histograms so that each degree of grayness has a relatively equal number of pixels. Histogram alignment is obtained by changing the degree of validity of a pixel (*r*) with a new degree of validity (*s*) with a T transformation function. Mathematically it can be written with the equation:



$$\boldsymbol{s = T(r)}$$

r can be recovered from s by the inverse transformation as in the equation:

$$r = T^{-1}(s)\ where\ 0 \leq s \leq 1$$

The formula used to calculate *histogram equalization* can be written as in the following equation:

$$\Pr(rk) = \frac{nk}{n} \; where \; rk = \frac{k}{L-1}, 0 \; \leq k \; \leq L-1$$

where n_k is the pixel value at the grayish degree k*, and* n *is the* sum of all pixels in the image. From the formulation it can be interpreted that the degree of grayness (*k*) is normalized to the degree of grayness (*L-1*). The value rk = 0 states black, and rk= 1 states white on the defined grayish scale.

Another formula that can be used to calculate histogram equalization on an image with the k bit grayness scale is as in the following equation:

$$Ko = round\left(\frac{Ci(2^k - 1)}{w.h}\right)$$

where:

Ci = cumulative distribution of the grayish scale value to –*i* of the original image
round = rounding function to the nearest number
Ko = the grayness value of the histogram equalization result
w = image width
h = image height

The following image is an example of an unequalized original image histogram and an image histogram after the equalization process.

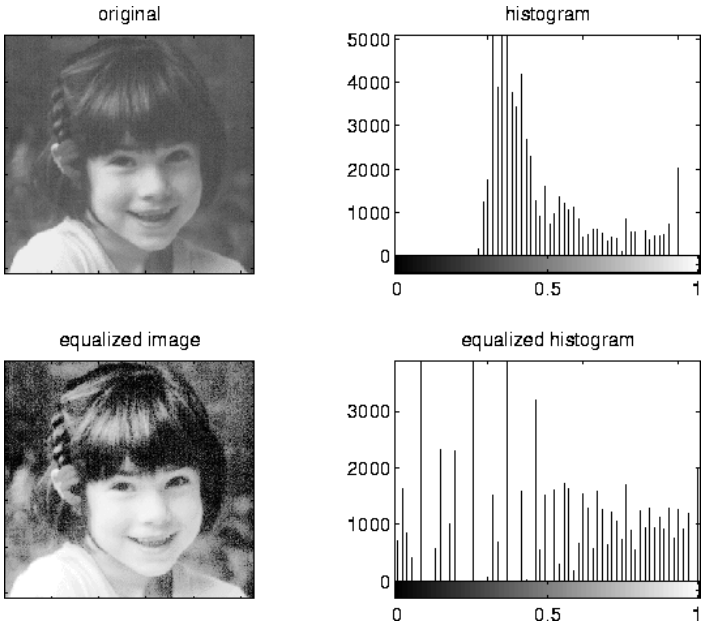


Figure 0.6 Original image and after histogram equalization

While the examples of manual calculation stages for histogram equalization are as follows:

1. Suppose there is a grayish image with pixel values as follows:

(i,j)	0	1	2	3	4	5	6	7
0	29	40	44	39	111	116	81	108
1	40	44	62	90	111	111	108	58
2	186	132	132	154	154	154	150	229
3	136	184	175	155	171	171	171	148
4	254	133	133	136	154	151	151	157
5	140	133	133	131	136	138	149	149
6	115	128	133	136	136	136	138	117
7	122	128	138	143	133	119	139	128

2. From the pixel value in the image matrix, the frequency and cumulative distribution of the grayish scale value are calculated. A list of cumulative distribution frequencies and calculations can be seen in the following table:

Skala Keabuan	Frek	Distribusi Kumulatif	Skala Keabuan	Frek	Distribusi Kumulatif
29	1	1	133	6	26 + 6 = 32
39	1	1 + 1 = 2	136	6	32 + 6 = 38
40	2	2 + 2 = 4	138	3	38 + 3 = 41
44	2	4 + 2 = 6	139	1	41 + 1 = 42
58	1	6 + 1 = 7	140	1	42 + 1 = 43
62	1	7 + 1 = 8	143	1	43 + 1 = 44
81	1	8 + 1 = 9	148	1	44 + 1 = 45
90	1	9 + 1 = 10	149	2	45 + 2 = 47
108	2	10 + 2 = 12	150	1	47 + 1 = 48
111	3	12 + 3 = 15	151	2	48 + 2 = 50
115	1	15 + 1 = 16	154	4	50 + 4 = 54
116	1	16 + 1 = 17	155	1	54 + 1 = 55
117	1	17 + 1 = 18	157	1	55 + 1 = 56
119	1	18 + 1 = 19	171	3	56 + 3 = 59
122	1	19 + 1 = 20	175	1	59 + 1 = 60
128	3	20 + 3 = 23	184	1	60 + 1 = 61
131	1	23 + 1 = 24	186	1	61 + 1 = 62
132	2	24 + 2 = 26	229	1	62 + 1 = 63
			254	1	63 + 1 = 64

3. Calculates the validity value of the cumulative distribution calculation using the formula previously written above. The following is an example of a calculation for the grayness scales 2 and 30:

$$Ko = round\left(\frac{1\ x\ (2^8 - 1)}{8\ x\ 8}\right) = round\ \left(\frac{255}{64}\right) = 4$$

$$Ko = round\left(\frac{2\ x\ (2^8 - 1)}{8\ x\ 8}\right) = round\ \left(\frac{510}{64}\right) = 8$$

The calculation results for the entire grayish scale value can be seen in the following table:

Keabuan Awal	Frek	Keabuan Hasil
29	1	4
39	1	8
40	2	16
44	2	24
58	1	28
62	1	32
81	1	36
90	1	40
108	2	48
111	3	60
115	1	64
116	1	68
117	1	72
119	1	76
122	1	80
128	3	92
131	1	96
132	2	104

Keabuan Awal	Frek	Keabuan Hasil
133	6	127
136	6	151
138	3	163
139	1	167
140	1	171
143	1	175
148	1	179
149	2	187
150	1	191
151	2	199
154	4	215
155	1	219
157	1	223
171	3	235
175	1	239
184	1	243
186	1	247
229	1	251
254	1	255

While the image matrix after the histogram equalization process is as follows:

(i,j)	0	1	2	3	4	5	6	7
0	4	16	24	8	60	68	36	48
1	16	24	32	40	60	60	48	28
2	247	104	104	215	215	215	191	251
3	151	243	239	219	235	235	235	179
4	255	127	127	151	215	199	199	223
5	171	127	127	96	151	163	187	187
6	64	92	127	151	151	151	163	72
7	80	92	163	175	127	76	167	92

E. Exercise:

Select one of the correct answers by marking the answer sheet.

CHAPTER 4
MORPHOLOGICAL IMAGE

After taking this chapter, students should:

- Understand the meaning and use of a histogram.
- Able to display image histograms using software.
- Able to modify the image histogram in image processing

A. Morphological Operation

Morphological operation is a common operation imposed on binary (black-and-white) imagery to change the structure of the shape of the object contained in the image. Morphology is an image processing technique based on form image segment. The purpose of morphological operations is to improve segmentation results.

Examples of morphological applications/operations:

(a) Closing the hole in the image
(b) Separating objects
(c) Forming spatial filters.
(d) Obtaining skeleton (skeleton) objects.
(e) Determine the location of objects in the image.
(f) Obtaining the shape of the object structure.

Figure 0.1 Original coin image

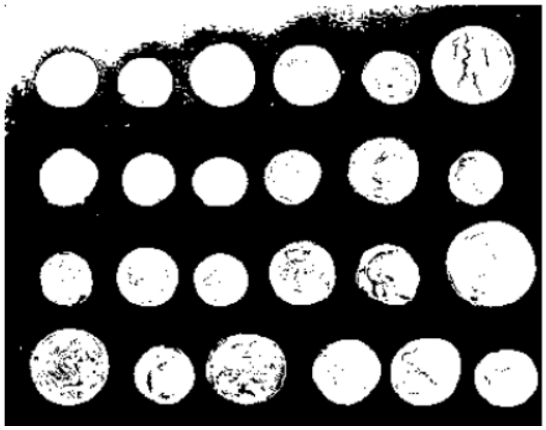


Figure 0.2 Object separation

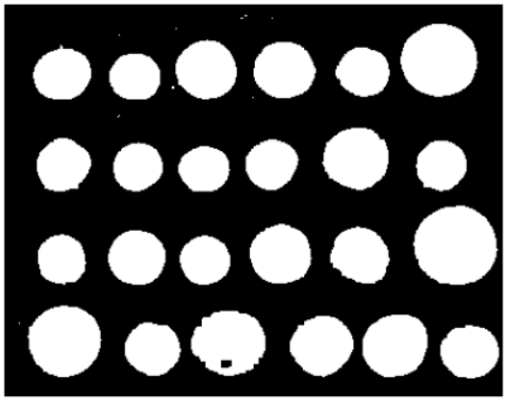


Figure 0.3 Foming a filter

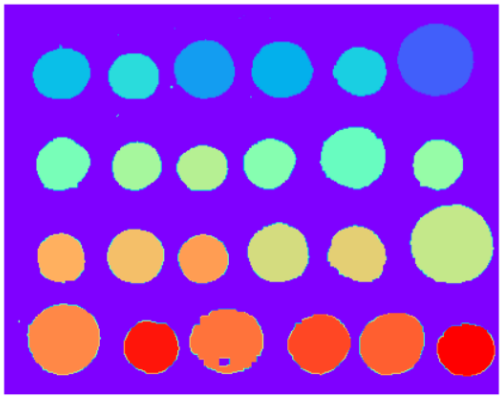


Figure 0.4 Image labelling

Figure 0.5 Object recognition

Morphological operations generally involve 2 (two) inputs:

1. Image of input to be subjected to morphological operations
2. Kernel or structuring element with a central point (hotspot)
   Kernel example:

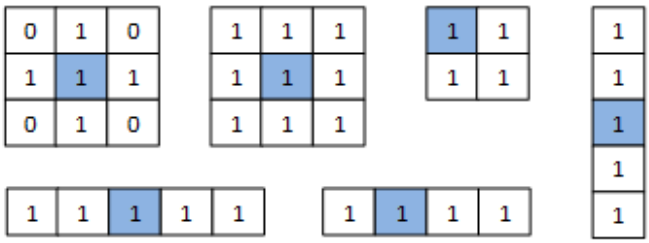


Figure 0.6 Example of kernel

B. Operation in Morphological Image

There are four operation usually used in morphological image:

1. Dilation operations
   Dilation surgery is commonly used to get the effect dilation to pixels that are valued 1.

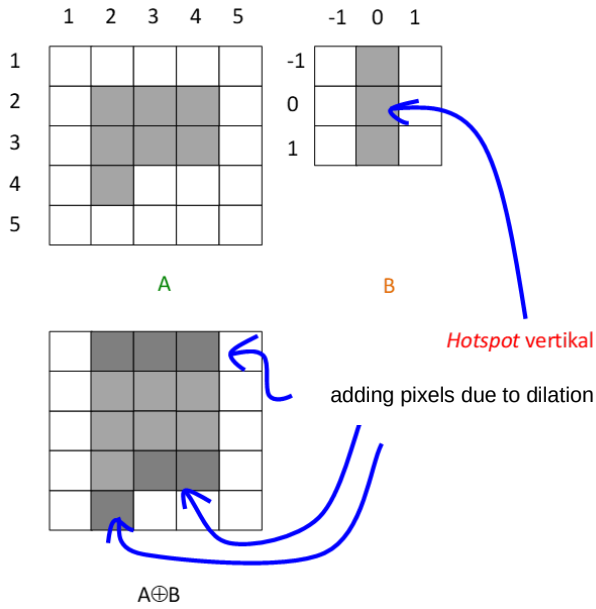


Figure 0.7 Dilation operation

Based on the above figure, we can know the pixel coordinates that are valued 1 are:
A= { (2,2), (2,3), (2,4), (3,2), (3,3), (3,4), (4,2) }
B= { (-1, 0), (0,0), (1,0)}

With thereby,
A ⊕ B = { (2,2) + (-1, 0) , (2,2) + (0, 0) + (2,2) + (1, 0), (2,3) + (-1, 0) , (2,3) + (0, 0) + (2,3) + (1, 0),
(2,4) + (-1, 0), (2,4) + (0, 0) + (2,4) + (1, 0), (3,2) + (-1, 0), (3,2) + (0, 0) + (3,2) + (1, 0),
(3,3) + (-1, 0), (3,3) + (0, 0) + (3,3) + (1, 0), (3,4) + (-1, 0), (3,4) + (0, 0) + (3,4) + (1, 0),
(4,3) + (-1, 0), (4,3) + (0, 0) + (4,3) + (1, 0) }
= { (1,2), (2,2), (3,2), (1,3), (2,3), (3,3), (1,4), (2,4), (3,3), (2,2), (3,2), (4,2),
(2,3), (3,3), (4,3), (2,4), (3,4), (4,4), (3,3), (4,3), (5,2) }
= { (1,2), (1,3), (1,4), (2,2), (2,3), (2,4), (3,2), (3,3), (3,4), (4,2), (4,3), (4,4), (5,2) }

2. Erosion Operations
Erosion operations have a minimizing effect image structure

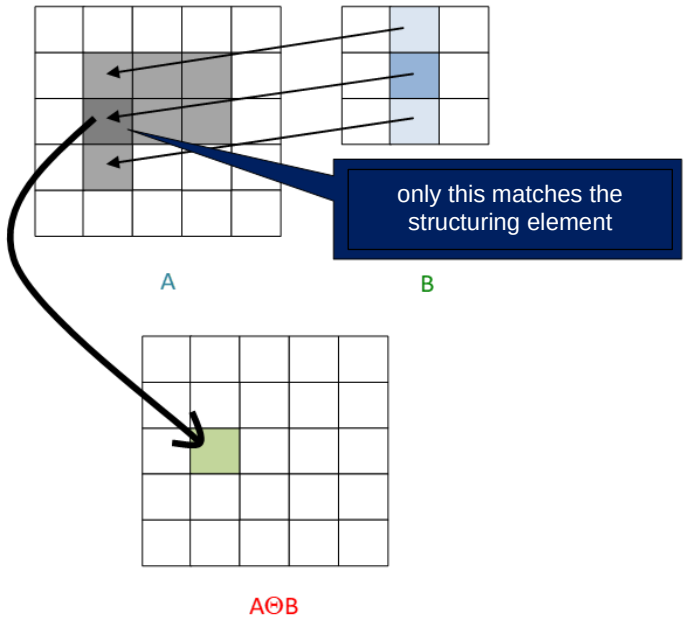


Figure 0.8 Erosion operation

3. Opening operation
   is an erosion operation that follows by dilation using the same structure element. This operation is useful for smoothing the contours of objects and remove all pixels in an area that is too small to be occupied by the structure element.

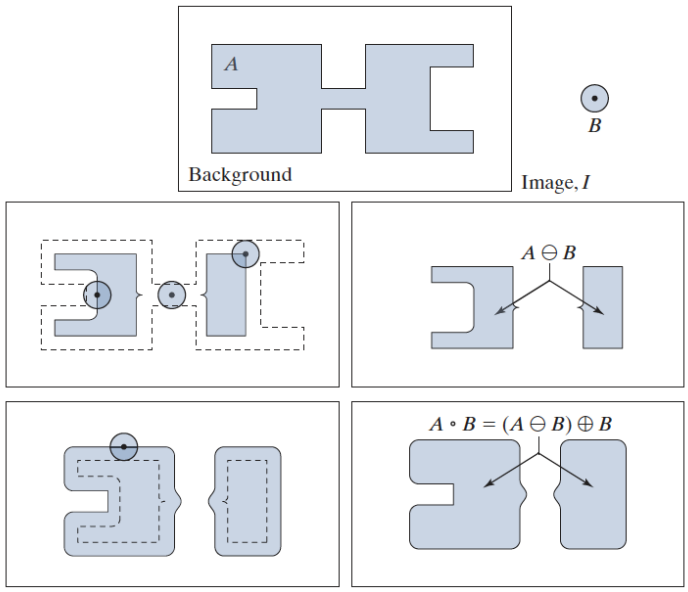


Figure 0.9 Opening operation

4. Closing operations are useful for smoothing contours and remove small holes. Closing operation carried out by performing dilation operations first and then followed by erosion operations.

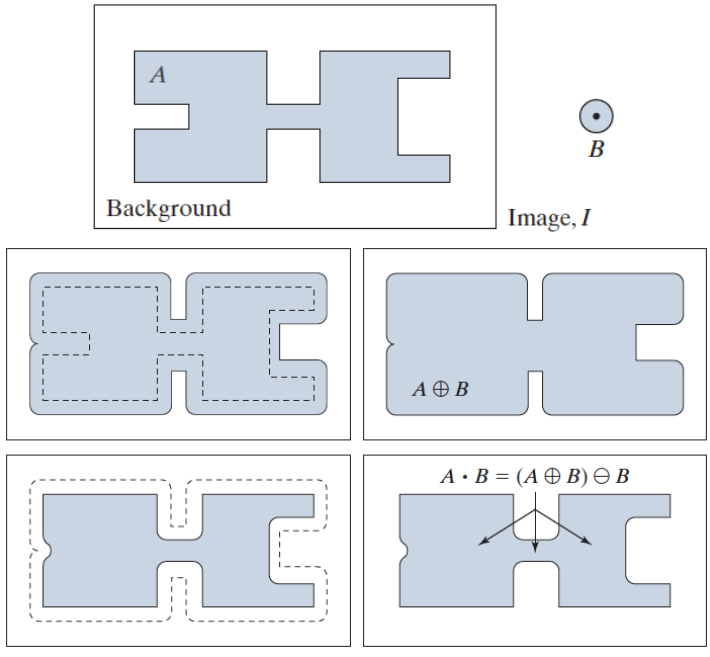


Figure 0.10 Closing operation

C. Structure Element

Size and shape of structure elements also determines the results of morphological operations. Shape commonly used in morphological operations are discs and circles.

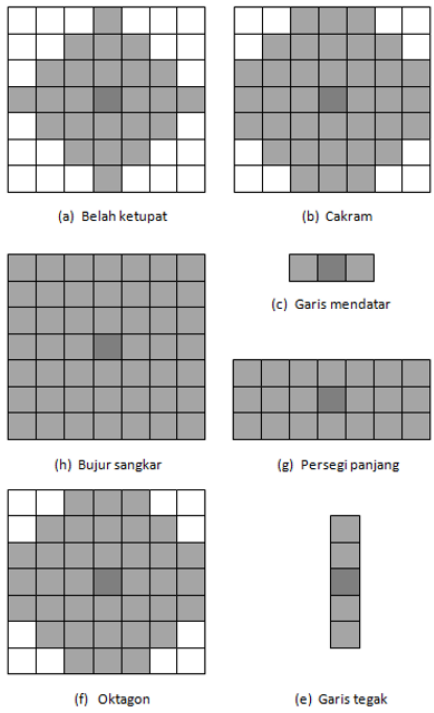


Figure 0.11 Structure element

D. Exercise:

Choose one of the correct answers by marking on the answer sheet.

1. skimage.morphology is usually often used to...
   a. Generate matrix shift
   b. Make the image black-and-white
   c. Generate structuring elements
   d. Duplicate binary images
   e. Remove Some colored imagery

2. Here is the command to call the process image segmentation....
   a. skimage.data

b. skimage.filters
c. skimage.morphology
d. skimage.segmentation
e. skimage.measure

3. Commonly used forms in morphological operations be...
   a. Circle Shape and Rhombus Shape
   b. Octagon shape and Circle shape
   c. Upright line shape and Circle shape
   d. Disc Shape and Circle Shape
   e. Horizontal Line Shape and square shape

4. What is the purpose of morphological operation for digital image processing...
   a. Improve image segmentation results
   b. Change the shape structure of an object
   c. Create image rotation
   d. Change pixels
   e. Pixel interpolation

5. Morphological Operations Section used to smoothing contours and removing holes is...
   a. Dilation Operations
   b. Erosion Operations
   c. Operation Closing
   d. Operation Opening
   e. Basic operations

CHAPTER 5
SPATIAL DOMAIN AND FILTERING

After taking this chapter, students should:

- Able to understand the basic concepts of spatial domains in image processing
- Able to understand the convolution processes in digital image processing
- Able to understand the filtering process on spatial domains

A. Spatial Domain

An image can be represented in the form of a 2D matrix where each element of the matrix represents pixel intensity. This state of 2D matrices that depict the intensity distribution of an image is called Spatial Domain. It can be represented as shown below:

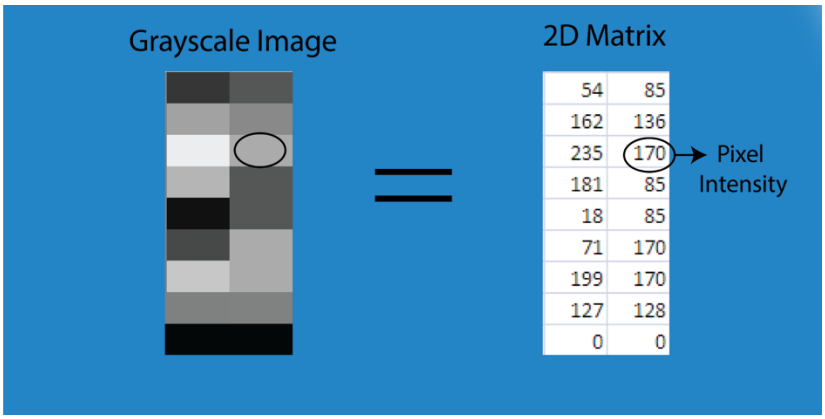


Figure 0.1 Illustration of Spatial Domain

For the RGB image, the spatial domain is represented as a 3D vector of 2D matrices. Each 2D matrix contains the intensities for a single color as shown below:

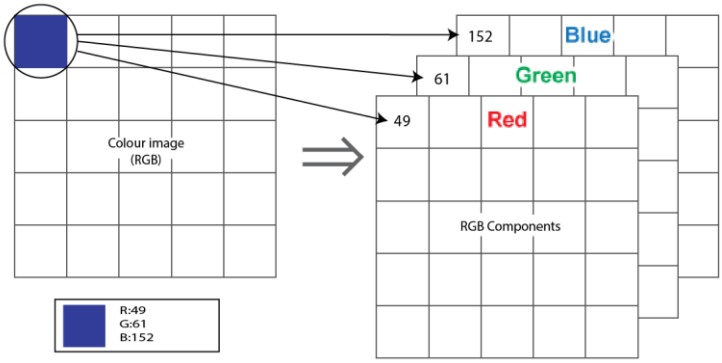


Figure 0.2 Spatial domain for color image (RGB)

Each pixel intensity is represented as I(x,y) where x,y is the coordinate of the pixel in the 2D matrix. Different operations are carried out in this value.

B. Convolution Theory

The fundamental operation in image processing is convolution operations. Convolution of 2 functions f(x) and g(x) is defined as follows:

$$h(x) = f(x) * g(x) = \int_{-\infty}^{\infty} f(a)g(x-a)da$$

in which case, the sign * states the convolution operator, and the variable (variable) a is an auxiliary changer (dummy variable).

For discrete functions, convolution is defined as:

$$h(x) = f(x) * g(x) = \sum_{a=-\infty}^{\infty} f(a)g(x-a)$$

In the convolution operation above, g(x) is called the convolution kernel or filter kernel. The g(x) kernel is a window operated in a shift on the input signal f(x), in which case, the number of multiplications of both functions at each point is the result of convolution expressed by the output h(x).

C. Convolution in Two-Dimension Function

For functions with two-dimensional functions, convolution operations are defined as follows:

1. For continuous function

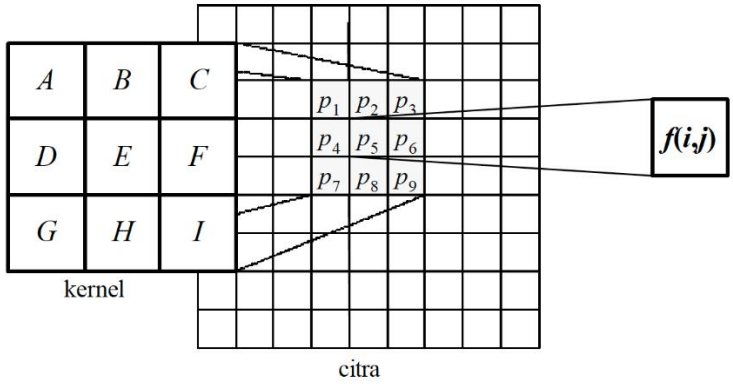
$$h(x,y) = f(x,y) * g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(a,b)g(x-a,y-b)dadb$$

2. For discrete functions

$$h(x,y) = f(x,y) * g(x,y) = \sum_{a=-\infty}^{\infty}\sum_{b=-\infty}^{\infty} f(a,b)g(x-a,y-b)$$

The g(x,y) filter function is also called convolution filter, convolution mask, kernel convolution, or template. In the discrete realm of convolution kernels expressed in matrix form (generally 3x3, but there are also those measuring 2x2 or 2x1 or 1x2). The size of this matrix is usually smaller than the size of the image. Each matrix element is called a convolution coefficient.

An illustration of convolution is shown in below figure:



$$f(i,j) = A\,p_1 + B\,p_2 + C\,p_3 + D\,p_4 + E\,p_5 + F\,p_6 + G\,p_7 + H\,p_8 + I\,p_9$$

Figure 0.3 Illustration of convolution

Convolution operations are performed by sliding the convolution kernel pixels per pixel. Convolution results are stored in the new matrix.

Example:

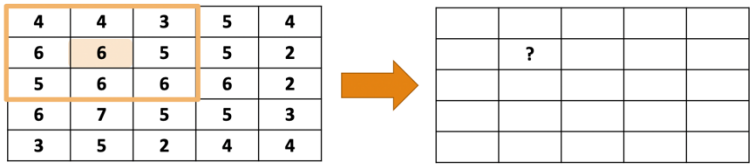
Suppose the image of f(x,y) measuring 5x5 and a kernel or mask measuring 3x3 is each as follows:

$$f(x, y) = \begin{bmatrix} 4 & 4 & 3 & 5 & 4 \\ 6 & 6 & 5 & 5 & 2 \\ 5 & 6 & 6 & 6 & 2 \\ 6 & 7 & 5 & 5 & 3 \\ 3 & 5 & 2 & 4 & 4 \end{bmatrix} \qquad g(x, y) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & \bullet 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

(Description: Sign • represents the position (0, 0) of the kernel)

Convolution operations between f(x, y) images and g(x, y) kernels, f(x, y) * g(x, y), it can be described as follows:

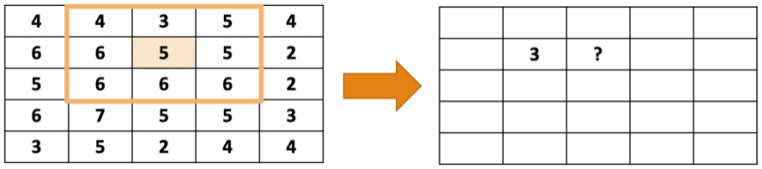
(1) Place the kernel in the upper left corner, then calculate the pixel value in the position (0, 0) of the kernel:



Convolution result = 3. This value is calculated in the following way:

(0x4) + (-1x4) + (0x3) + (-1x6) + (4x6) + (-1x5) + (0x5) + (-1x6) + (0x6) = 3

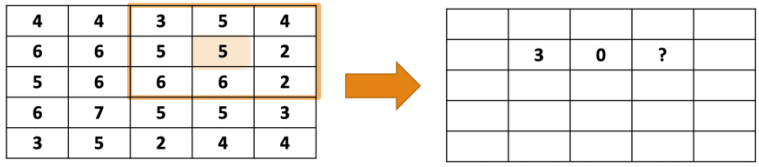
(2) Slide the kernel one pixel to the right, then calculate the pixel value in the position (0,0) of the kernel:



Convolution result = 0. This value is calculated in the following ways:

(0x4) + (-1x3) + (0x5) + (-1x6) + (4x5) + (-1x5) + (0x6) + (-1x6) + (0x6) = 0

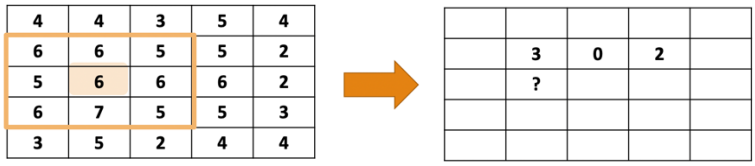
(3) Slide the kernel one pixel to the right, then calculate the pixel value in the position (0, 0) of the kernel:



Convolution result = 2. This value is calculated in the following way:

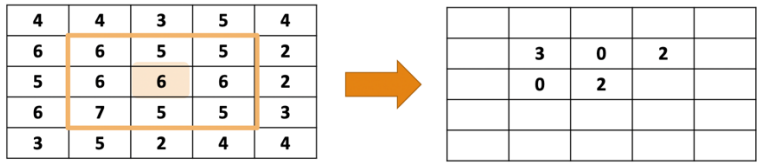
(0x3) + (-1x5) + (0x4) + (-1x5) + (4x5) + (-1x2) + (0x6) + (-1x6) + (0x2) = 2

(4) Next, slide the kernel one pixel down, and then start convolution again from the left side of the image. Each time it convoluts, slide the kernel one pixel to the right:



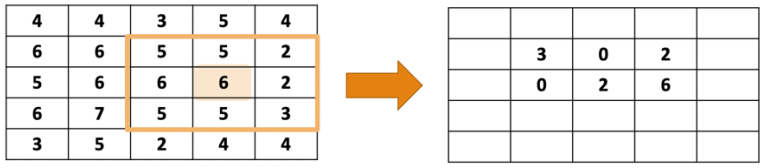
Convolution result = 0. This value is calculated in the following way:

(0x6) + (-1x6) + (0x5) + (-1x5) + (4x6) + (-1x6) + (0x6) + (-1x7) + (0x5) = 0



Convolution result = 2. This value is calculated in the following way:

(0x6) + (-1x5) + (0x5) + (-1x6) + (4x6) + (-1x6) + (0x7) + (-1x5) + (0x5) = 2



Convolution result = 6. This value is calculated in the following way:

(0x5) + (-1x5) + (0x2) + (-1x6) + (4x6) + (-1x2) + (0x5) + (-1 x5) + (0x3) = 6

In the same way as above, the pixels in the third row are convolved so as to produce:

	3	0	2	
	0	2	6	
	6	0	2	

For the record, if the convolution result produces a negative pixel value, then the value is made 0, otherwise if the convolution result produces a pixel value greater than the maximum graying value, then the value is made to the maximum graying value (remember clipping operation). Problems arise when the convoluted pixel is a border pixel, because some convolution coefficients cannot be positioned on the image pixels (the "hanging" effect), such as the example below:

4	4	3	5	4	?
6	6	5	5	2	?
5	6	6	6	2	?
6	7	5	5	3	
3	5	2	4	4	

This kind of "hanging" problem always occurs on the left edge pixels, right, up, and down. The solution to this problem is:

(1) Edge pixels are ignored, not convoluted. This solution is widely used in the library of image processing functions. In this way, the edge pixels of the value remain the same as the original image.

(2) Duplication of image elements, for example the first column element is copied to the column M+1, and vice versa, then convolution can be done to pixel that edge.

(3) Elements marked with "?" are assumed to be worth 0 or constants that other, so that the convolution of edge pixels can be done.

Solutions with the three approaches above assume the edges of the image. The width is very small (only one pixel) relative to the size of the image, so the edge pixels do not show visible effects.

You can see that convolution operations are computations at the local level, because the computation for a pixel on the output image involves a neighboring pixelpixel in its input image.

Convolution is useful in the image processing process such as:

- improvement of image quality (image enhancement)
- noise removal
- reduces innards
- image smoothing/softening
- edge detection, edge sharpening

D. Spatial Filtering

Spatial filtering is used in a broad spectrum of image processing applications, so a solid understanding of filtering principles is important. The name filter is borrowed from frequency domain processing where “filtering” refers to passing, modifying, or rejecting specified frequency components of an image. For example, a filter that passes low frequencies is called a lowpass filter. The net effect produced by a lowpass filter is to smooth an image by blurring it. We can accomplish similar smoothing directly on the image itself by using spatial filters. Spatial filtering modifies an image by replacing the value of each pixel by a function of the values of the pixel and its neighbors. If the operation performed on the image pixels is linear, then the filter is called a linear spatial filter. Otherwise, the filter is a nonlinear spatial filter. We will focus attention first on linear filters and then introduce some basic nonlinear filters.

E. Linear Spatial Filtering

A linear spatial filter performs a sum-of-products operation between an image f and a filter kernel, w. The kernel is an array

whose size defines the neighborhood of operation, and whose coefficients determine the nature of the filter. Other terms used to refer to a spatial filter kernel are mask, template, and window. We use the term filter kernel or simply kernel.

Figure 1 illustrates the mechanics of linear spatial filtering using a 3 × 3 kernel as follows:

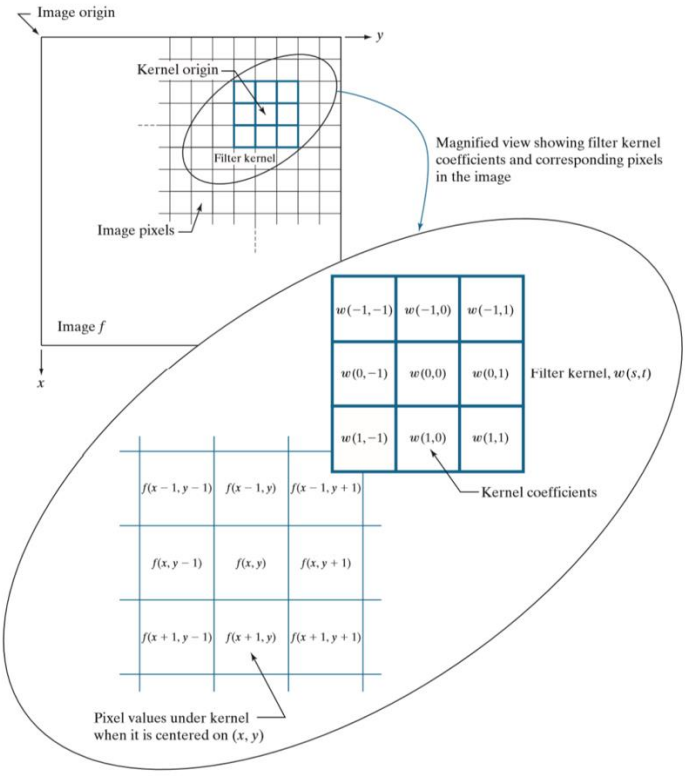


Figure 0.4 The mechanics of linear spatial filtering using a 3 × 3 kernel. The pixels are shown as squares to simplify the graphics.

Note that the origin of the image is at the top left, but the origin of the kernel is at its center. Placing the origin at the center of spatially symmetric kernels simplifies writing expressions for linear filtering.

At any point (x, y) in the image, the response, $g(x, y)$, of the filter is the sum of products of the kernel coefficients and the image pixels encompassed by the kernel:

$$g(x, y) = w(-1, -1)f(x - 1, y - 1) + w(-1,0)f(x - 1, y) + \cdots + w(0,0)f(x, y) + \cdots + w(1,1)f(x + 1, y + 1)$$

As coordinates x and y are varied, the center of the kernel moves from pixel to pixel, generating the filtered image, g, in the process. A filtered pixel value typically is assigned to a corresponding location in a new image created to hold the results of filtering. Observe that the center coefficient of the kernel, $w(0,0)$, aligns with the pixel at location (x, y).

For a kernel of size (m, n), we assume that $m = 2a + 1$ and $n = 2b + 1$, where a and b are nonnegative integers. This means that our focus is on kernels of odd size in both coordinate directions. In general, linear spatial filtering of an image of size $M \times N$ with a kernel of size $m \times n$ is given by the expression:

$$g(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t)f(x + s, y + t)$$

where x and y are varied so that the center (origin) of the kernel visits every pixel in f once.

F. Smoothing (Low pass) Spatial Filters

Smoothing (also called averaging) spatial filters are used to reduce sharp transitions in intensity. Because random noise typically consists of sharp transitions in intensity, an obvious application of smoothing is noise reduction. Smoothing is used to reduce irrelevant detail in an image, where “irrelevant” refers to pixel regions that are small with respect to the size of the filter kernel. Linear spatial filtering consists of convolving an image with a filter kernel. Convolving a smoothing kernel with an image blurs the image, with the degree of blurring being determined by the size of the kernel and the values of its coefficients.

(1) Box Filter Kernels

The simplest, separable lowpass filter kernel is the box kernel, whose coefficients have the same value (typically 1). The name “box kernel” comes from a constant kernel

resembling a box when viewed in 3-D. We show a 3 × 3 box filter in Figure 5.5 (a). An $m \times n$ box filter is an $m \times n$ array of 1's, with a normalizing constant in front, whose value is 1 divided by the sum of the values of the coefficients (i.e., 1 mn when all the coefficients are 1's).

$\frac{1}{9} \times$

1	1	1
1	1	1
1	1	1

(a)

$\frac{1}{4.8976} \times$

0.3679	0.6065	0.3679
0.6065	1.0000	0.6065
0.3679	0.6065	0.3679

(b)

Figure 0.5 Examples of smoothing kernels: (a) is a box kernel; (b) is a Gaussian kernel.

Figure 5.6 (a) shows a test pattern image of size 1024 × 1024 pixels. Figures 3(b)-(d) are the results obtained using box filters of size $m \times m$ with m = 3, 11, and 21, respectively.

For m = 3, we note a slight overall blurring of the image, with the image features whose sizes are comparable to the size of the kernel being affected significantly more.

Such features include the thinner lines in the image and the noise pixels contained in the boxes on the right side of the image. The filtered image also has a thin gray border, the result of zero-padding the image prior to filtering. The padding extends the boundaries of an image to avoid undefined operations when parts of a kernel lie outside the border of the image during filtering. When zero (black) padding is used, the net result of smoothing at or near the border is a dark gray border that arises from including black pixels in the averaging process. Using the 11 × 11 kernel resulted in more pronounced blurring throughout the image, including a more prominent dark border. The result with the 21 × 21 kernel shows significant blurring of all components of the image, including the loss of the characteristic shape

of some components, including, for example, the small square on the top left and the small character on the bottom left. The dark border resulting from zero padding is proportionally thicker than before.

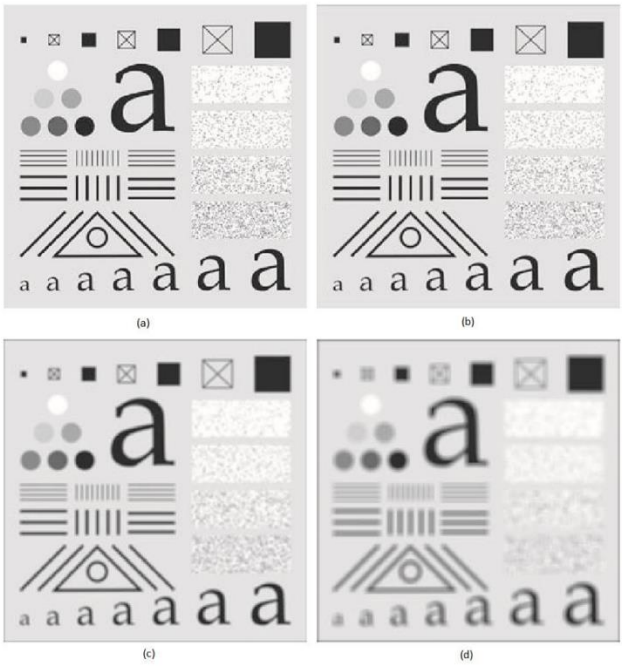


Figure 0.6 (a) Test pattern of size 1024 × 1024 pixels; (b)-(d) Results of lowpass filtering with box kernels of sizes 3 × 3, 11 × 11, and 21 × 21, respectively.

(2) Lowpass Gaussian Filter Kernels

Because of their simplicity, box filters are suitable for quick experimentation and they often yield smoothing results that are visually acceptable. They are useful also when it is desired to reduce the effect of smoothing on edges. However, box filters have limitations that make them poor choices in many applications.

For example, a defocused lens is often modeled as a lowpass filter, but box filters are poor approximations to the blurring characteristics of lenses. Another limitation is the fact that box filters favor blurring along perpendicular directions. In applications involving images with a high level

of detail, or with strong geometrical components, the directionality of box filters often produces undesirable results.

The kernels of choice in applications such as those just mentioned are circularly symmetric (also called isotropic, meaning their response is independent of orientation). As it turns out, Gaussian kernels of the form:

$$w(s,t) = G(s,t) = K.e^{-\frac{(i-u)^2+(j-v)^2}{2(\sigma^2)}}$$

Example:
Specifies a Gaussian filter with a matrix size of 3x3, and standard deviation = 1:

1. Create a matrix according to size and specify its coordinates.

(-1,1)	(0,1)	(1,1)
(-1,0)	(0,0)	(1,0)
(-1,-1)	(0,-1)	(1,-1)

2. Calculate the value of the Gauss kernel matrix element, with the following formula:

$$w(s,t) = G(s,t) = K.e^{-\frac{(i-u)^2+(j-v)^2}{2(\sigma^2)}}$$

Example:

$$G(0,0) = 1.e^{-\frac{(0-0)^2+(0-0)^2}{2(1^2)}}$$

$$G(0,0) = e^0 = 1$$

with the same calculation, obtained:

$G(0,0) = e^0 = 1$

$G(1,0) = G(0,1) = G(-1,0) = G(0,-1) = e^{-\frac{1}{2}} = 0{,}6065$

$G(1,1) = G(1,-1) = G(-1,1) = G(-1,-1) = e^{-1} = 0{,}3679$

From above values, obtained the new table:

0,3679	0,6065	0,3679
0,6065	1	0,6065

0,3679	0,6065	0,3679

3. Find the smallest number from within the table (**0.3679**)
4. The division between 1 and the smallest number (**divResult = 1 / 0.3679**)
5. Multiply all the numbers in the table by **divResult**, and round it to the nearest number.

1	2	1
2	3	2
1	2	1

To compare Gaussian and box kernel filtering, we repeat the example in the box kernel using a Gaussian kernel. Gaussian kernels have to be larger than box filters to achieve the same degree of blurring. This is because, whereas a box kernel assigns the same weight to all pixels, the values of Gaussian kernel coefficients (and hence their effect) decreases as a function of distance from the kernel center.

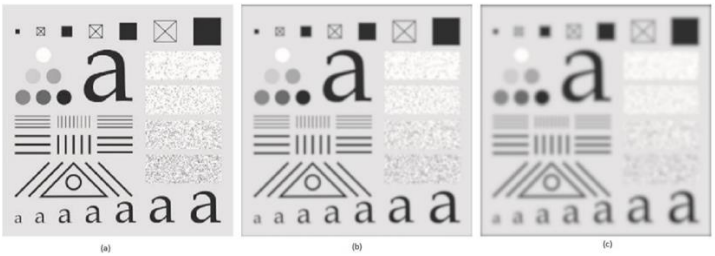


Figure 4: (a) A test pattern of size 1024 × 1024; (b) Result of lowpass filtering the pattern with a Gaussian kernel of size 21 × 21, with standard deviations σ = 3.5; (c) Result of using a kernel of size 43 × 43, with σ = 7

F. Exercise:

Select one of the correct answers by marking the answer sheet.

1. Which of the following statements is true about filtering in the spatial domain?

a. The type of filter used in the edge detection process is a low pass filter
b. Sharpening spatial filter can increase the detail in digital images
c. Smoothing spatial filters can increase the detail in digital images
d. The Robert filter is an extension of the Prewitt filter
e. Gaussian filter is a derivative of Laplacian filter

2. Filters whose pixel values are obtained by finding the middle value are:
   a. Gaussian Filter
   b. Filter Mode
   c. Laplacian Filter
   d. Mean Filter
   e. Median Filter

3. What are not included in the filter to soften the image are:
   a. Gaussian Filter
   b. Filter Mode
   c. Laplacian Filter
   d. Mean Filter
   e. Median Filter

4. What is the element value of the Gaussian mask matrix for the coordinates (2,2) with a standard deviation of 1?
   a. 0.0183
   b. 0.6065
   c. 0.0821
   d. 0.1353
   e. 0.3679

5. Included in the low pass filter are:
   a. Gaussian Filter
   b. Laplacian Filter
   c. Prewitt Filter
   d. Robert Filter
   e. Sobell Filter

CHAPTER 6
IMAGE TRANSFORMATION

After taking this chapter, students should:

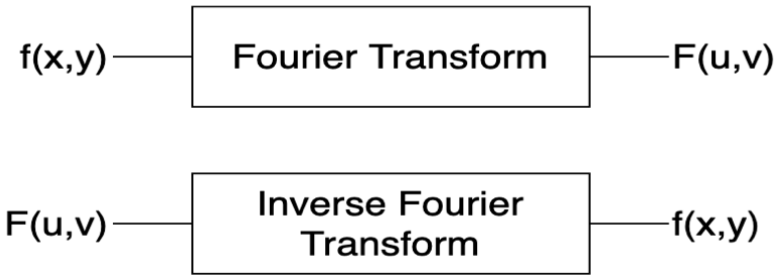
- Understand the measurement way of information in digital image
- Understand the difference between lossy and lossless in image compression
- Understand the principles of image compression methods, and how they work

A. Preface

Since convolution is done per pixel and for each pixel multiplication and summation operations are performed, it is clear that convolution consumes a lot of time. If the image is NxN and the kernel is mxm in size, then the number of multiplications is in the order N^2m^2. For example, if the image is 512x512 and the kernel is 16x16, then there are about 32 million multiplications needed. This is definitely not suitable for real-time processes without dedicated hardware.

One way to reduce computing time is to transform the image and kernel into the realm of frequency (using the Fourier Transformation) then convolution is carried out in the realm of time. The main advantage of the use of the frequency realm is that convolution processes can be applied in the form of direct multiplication.

The process of changing functions from the spatial realm realm to the frequency realm is carried out through the Fourier Transformation. While the change in function from the frequency realm to the spatial realm is carried out through the Fourier Inverse Transformation.



B. Fourier Transformation

Fourier transformation is the most important transformation in the field of signal processing, especially in the field of image processing. Generally, the signal is expressed as a form of plot amplitude versus time (at the function of one matra) or plot amplitude versus spatial position (at the dualmatra function). In some signal processing applications, there is difficulty performing operations due to functions in the time / spatial realm, for example in the convolution operations above. Convolution operations can be applied as a form of direct multiplication when the function is in the realm of propagation. Fourier Transformation is a kakas (tool) to change functions from the time/spatial realm to the frequency realm. For the reverse change used Inverse Fourier Transformation. The essence of the Fourier Transformation is to decompose a signal or wave into a number of sinusoids of various frequencies, the number of which is equivalent to the original wave.

In image processing, Fourier transformations are used to analyze frequencies in operations such as image recording, image quality improvement, image restoration, coding, and others. From the frequency analysis, we can make frequency changes to the image. Frequency changes relate to the spectrum between images whose contrast fog is the same as images rich in visual details. For example, in the image recording process there may be blurring of image contrast. In images that experience contrast blur, there is a gradual change in intensity, which means loss of high-frequency information. To

improve image quality, we use a high-frequency filter so that fuzzy contrasting pixels can be increased in intensity.

1-dimensional Fourier transformation is given by:

$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x) exp^{\left[\frac{-2j\pi ux}{N}\right]}$$

Because with euler's theorem: $e^{j\theta} = cos\theta + jsin\theta$
Thus, the 1-dimensional Fourier Transformation into:

$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)\left(\cos\left(\frac{2\pi ux}{N}\right) - jsin\left(\frac{2\pi ux}{N}\right)\right)$$

The result of the fourier transformation contains real and imaginary numbers expressed in R(u) and I(u). The fourier spectrum (magnitude) is expressed as:

$$|F(u)| = (R(u)^2 + I(u)^2)^{1/2}$$

The angle (phase) of the fourier is expressed as:

$$\theta(u) = tan^{-1}\left[\frac{I(u)}{R(u)}\right]$$

C. Walsh-Hadamard Transformation

The Walsh–Hadamard transform is a non-sinusoidal, orthogonal transformation technique that decomposes a signal into a set of basis functions. These basis functions are Walsh functions, which are rectangular or square waves with values of +1 or –1. Walsh–Hadamard transforms are also known as Walsh, or Walsh-Fourier transforms. The Walsh–Hadamard transform returns sequency values. Sequency is a more generalized notion offrequency and is defined as one half of the average number of zero-crossings per unit time interval.

Each Walsh function has a unique sequency value. It can be used to estimate the signal frequencies in theoriginal signal. Three different ordering schemes are used to store Walsh functions: sequency, Hadamard, and dyadic. Sequency ordering, which is used in signal processing applications, has the Walsh functions in the ordershown in the figure above.

Hadamard ordering, which is used in controls applications, arranges them as 0,4, 6, 2, 3, 7, 5, 1. Dyadic or gray code ordering, which is used in mathematics, arranges them as 0, 1, 3, 2,6, 7, 5, 4.

Like the FFT, the Walsh–Hadamard transform has a fast version, the fast Walsh–Hadamard transform (fwht). Compared to the FFT, the FWHT requires less storage space and is faster to calculate because ituses only real additions and subtractions, while the FFT requires complex values. The FWHT is able torepresent signals with sharp discontinuities more accurately using fewer coefficients than the FFT. Both the FWHT and the inverse FWHT (ifwht) are symmetric and thus, use identical calculation processes.

1-D Walsh transform is given by:

$$W(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x) \prod_{i=0}^{n-1} (-1)^{b_i(x)b_{n-1-i}(u)}$$

Unlike the Fourier transform, which is based on trigonometric functions, the Walsh transform consist of aseries expansion of basis functions whose values are only +1 and -1.

D. Hadamard Transformation

Hadamard Transform is an example of a generalized class of Fourier transforms. It performsan orthogonal, symmetric, involution, linear operation on 2m real numbers (or complex numbers,although the Hadamard matrices themselves are purely real). The Hadamard transform can be regarded as being built out of size-2 discrete Fourier transforms (DFTs), and is in fact equivalent to a multidimensional DFT of size. It decomposes an arbitrary input vector into asuperposition of Walsh functions.In a similar form as the Walsh transform, the 2-D Hadamard transform is defined as,
The Hadamard transform differs from the Walsh transform only in the order of basis functions. The orderof basis functions of Hadamard transform does not allow fast computation of it by using the straight forward modification of FFT. An extended

version of the Hadamard transform is the ordered Hadamard transform for which a fast algorithm called Fast Hadamard transform can be applied.

Two important characteristics of the Walsh functions are their compactness (representing the lower orderfunctions requires fewer samples), and the simplicity and quickness of their constructionThe effectiveness of most of these applications, especially filtering and coding, depends on the ability ofthe transform to pack signal energy into a few transform coefficients.Today Walsh transform is mainly used in multiplexing which is to send several data simultaneously. It does not require high energy packing ability.

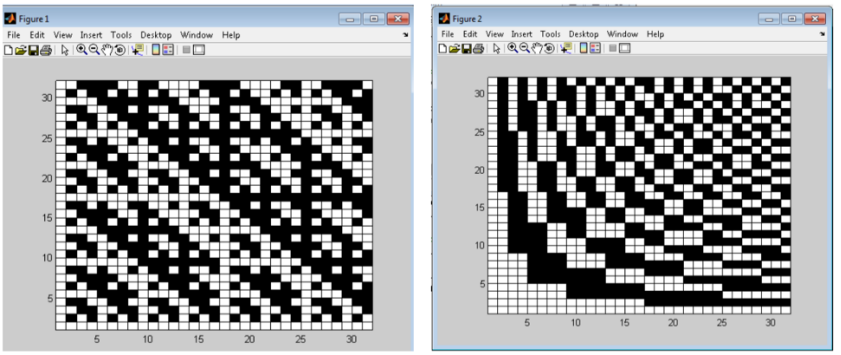


Figure 0.1The difference between the Walsh matrix and Hadamard matrix

E. Exercise:

Select one of the correct answers by marking the answer sheet.

1. If it is known that there is a 1-dimensional image of f(x) = f(0), f(1), f(2), f(3) = (5, 6, 7, 8), what is the magnitude (2) value?
   a. 0.71
   b. 0.5
   c. -0.71
   d. -6.5
   e. 6.5
2. If it is known that the number of data is 8, what are the values of b0(3), b1(6) and b2(4)?

a. 1, 1, 0
b. 0, 1, 0
c. 0, 1, 1
d. 1, 1, 1
e. 1, 0, 1

3. The disadvantages of the spatial domain are:
   a. Long computing time
   b. Using the concept of time domain
   c. Processing on a pixel-by-pixel basis
   d. Able to eliminate periodic noise
   e. Able to manipulate the frequency value that represents the signal

4. The following statements are in accordance with the concept of frequency in digital images, except:
   a. Low frequency means how many pixels it takes to go from white to black
   b. Frequency indicates the rate of change of intensity per pixel
   c. Filters in the frequency domain are used for image enhancement purposes
   d. Frequency in digital images requires changing the domain from spatial to frequency
   e. High frequency means the number of pixels needed to change from white to black

5. Which of the following statements is not true regarding image transformation:
   a. Image transformation is the process of changing the domain from frequency to spatial domain
   b. Filtering in the frequency domain is able to eliminate periodic noise in digital images
   c. The meaning of the frequency at the pixel value of 20 = 0.1 i.e. 1 period every 10 pixels
   d. Fourier transform divides the data into several data based on the frequency
   e. The Inverse Fourier Transform converts F(x,y) to f(x,y)

CHAPTER 7
IMAGE COMPRESSION AND DECOMPRESSION

After taking this chapter, students should:

- Understand the measurement way of information in digital image
- Understand the difference between lossy and lossless in image compression
- Understand the principles of image compression methods, and how they work

A. Preface

In general, digital image representation requires a large memory. For example, Lena's image in bitmap format is 512 x 512 pixels require 32 KB of memory (1 pixel = 1 byte) to representation. The larger the image size, of course, the larger the memory. what he needs. On the other hand, most images contain duplication. data. Duplication of data on an image can mean two things. First, it is likely a pixel with a pixel has the same intensity, so Storage of each pixel wastes space. Second, many images. contains the same part (region), so this same part is not it needs to be encoded repeatedly because it is redundant or redundant.

Today, most apps want an image representation with a need little memory as possible. Image compression or image compression aims to minimize the need for memory to represent digital imagery. The general principle used in the image compression process is reduce duplication of data in the image so that the memory needed to represent an image to be less than an original image representation.

There are two main processes in the issue of image compression:

1. Image compression.
   In this process, the image in the representation is not compressed encoded with representations that minimize memory needs. Image with format Bitmaps are generally

not in compressed form. The image that has been compressed is saved into an archive with a specific format. We know JPG and GIF formats as compressed image formats.

2. Image decompression.
   In this process, the compressed image must be restorable. again (decoding) becomes an unsanctioned representation. This process is required if the image is displayed to the screen or saved to the inside archive with non-compressed format. In other words, image impersonation returns the compressed image to bitmap data.

B. Image Compression App

Image compression contributes great benefits in the industry multimedia at this time. Image compression is useful for applications that do:

1. Data transmission on data communication channels
   Compressed image requires more delivery time short compared to an image that is not compressed. Examples fax image delivery application, videoconferencing, data transmission medical, sending images from space satellites, sending images via cellular phone. Download images from the internet, and so on.
2. Data storage (data storing) in secondary media (storage)
   Compressed imagery requires memory space in the media less storage compared to images that are not compressed. Examples of applications include image database applications, office automation, video storage (such as Video Compact Disc), etc.

C. Image Compression Criteria

Currently, there are many methods of image compression. Criterion the ones used in measuring the image compression method are:

1. Compression and decompression time
   The time of compression of the image and its compression should be fast. Exist compression methods whose compression time is long, but time the compression is fast. There is also a method whose compression time fast but

the compression time is slow. But there are also methods that the compression and compressing time are fast or both are slow.

2. Memory needs
   The memory needed to represent an image should be reduced. meaningfully. There are methods that successfully compress by percentage Some are big, some are small. In some methods, memory size results compression depends on the image itself. Cira containing many duplication elements (e.g. image of a clear sky without clouds, floor imagery ceramics) are generally successfully compressed with less memory compared to compressing an image containing multiple objects (e.g. natural scenery imagery).
3. Compression quality (fidelity)
   Information lost due to compression should be to a minimum so that the quality of the compression results is maintained. Compression quality Memory needs are usually inversely proportional. Good compression quality is generally achieved in the compression process that produces Memory reduction is not so great, as well as preferably. In other words, there is a tradeoff between the quality of the image and the size of compression results. The quality of an image is subjective and relative, depending on the observation of the person who judged it. Someone could have said the quality of an image is good, but others may say less Good, ugly, and so on. We can make the quality measure of the image compression result into a size quantitative by using the amount of PSNR (peak signal-to-noise ratio).

   PSNR is calculated to measure the difference between the original image and the image compression results (of course the image of the compression result must be compressed first) with the original image, with the formula:

$$PSNR = 20 \times \log_{10}\left(\frac{b}{rms}\right)$$

with b is the largest signal value (on black-and-white image, b = 255) and rms is the second-rank root of the difference between the original image and the result image compression. RMS value is calculated by the formula:

$$rms = \sqrt{\frac{1}{width\ x\ height}} \sum_{i=1}^{N} \sum_{j=1}^{M} (fij - f'ij)^2$$

which in this case, f and f' each state the pixel value of the image original and pixel value of the compressed image. PSNR has a decibel unit (dB). Equation above states that PSNR is only calculated after the process of image and PSNR is inversely proportional to RMS. Low RMS value implies that the image of the compression result is not much different from the image originally it would produce a high PSNR, which means quality the compression is good. The greater the value of PSNR, the better the quality the compression. How much a good PSNR value cannot be expressed explicitly, depending on the compressed image. But we can find out this if we do the testing by trying various combinations of compression parameters are used. If PSNR value getting bigger, it means that the compression parameters used are already towards good grades. Image compression parameters depend on the method the compressing used.

4. Output format

   The image format of the compression result should be suitable for delivery and data storage. Image reading depends on how the image it is represented (or stored).

   The exact selection of criteria depends on the user and application. For example, whether the user wants a compression that produces quality that good, but the memory reduction needed is not too great, or Instead. Or if the compression time can be ignored from consideration (assuming that compression is only done once, however compression can be many times), hence the method that

produces time rapid compression that needs to be considered.

There are four approaches used in image compression:

1. Statistical approach
   Image compression is based on the frequency of occurrence of the degree of grayness pixels in all parts of the image. Example method: Huffman Coding.
2. Space approach
   Image compression is based on the spatial relationships between the pixels in the in a group that has the same degree of grayness within an area in the picture. Example method: Run-Length Encoding.
3. Quantization approach
   Image compression is done by reducing the number of degrees of grayness which is available. Example method: quantization compression method.
4. Fractal approach
   Image compression is based on the fact that the similarity of parts in the image can be exploited with a transformation matrix. Example method: Fractal Image Compression.

D. Classification of Compression Methods

The method of compressing image can be classified into two large groups:

1. Lossless method
   The lossless method always produces the right image of the compression result same as the original image, pixel by pixel. No information is missing due to compression. Unfortunately, the ratio of method image compression Lossless is very low. An example of a lossless method is the Huffman method.
   Image compression ratio calculated by formula:

$$\text{Ratio} = 100\% - \left(\frac{compressed\ image}{original\ image}\ x\ 100\%\right)$$

   Lossless method suitable for compressing images containing important information that should not be

damaged by compression. For example, compressing images of medical diagnostic results.

2. Lossy method

   Lossy method produces almost the same compression result image with the original image. There is information lost due to compression, but It can be tolerated by eye perception. The eye cannot distinguish changes. Small in the picture. Lossy compression method results in ratio high compression rather than lossless method. Figure 8.1 is an image before compressing, and Figure 8.2 is the result of image compression ship with lossy method. Examples of lossy methods are JPEG methods and fractal methods.

Figure 0.1 Original image before image compression with lossless and lossy method

Figure 0.2 After compression with lossless method

Figure 0.3 After compression with lossy method

E. Huffman Compression Method

Huffman's compression method uses the principle that values (or degrees) the grayness that often appears in the image will be encoded with the number of bits which is less while the grayish value whose frequency of appearance slightly encoded with a longer number of bits.

Huffman method algorithm:

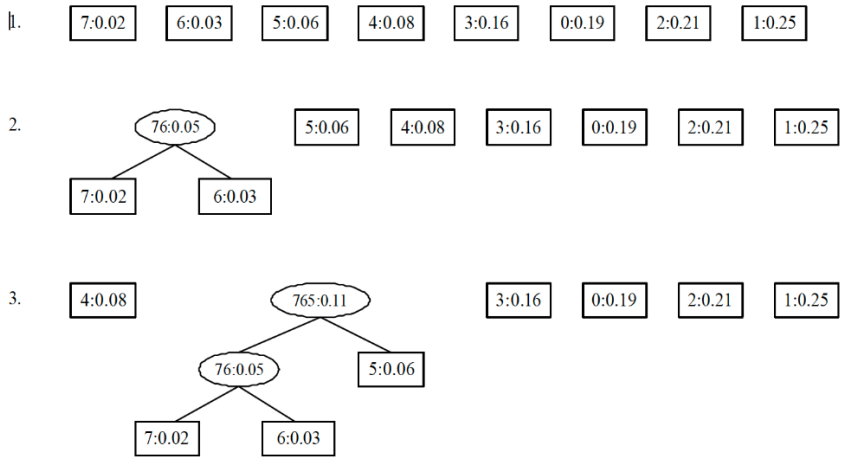
1. Sort ascending orders by default values by frequency of its occurrence (or based on the chance of occurrence, pk, that is the frequency of occurrence (nk) is divided by the number of pixels in the image (n)). Each grayish value is expressed as a single knotted tree. Every node is assigned with the frequency of occurrence of the grayish value.
2. Join two trees that have the smallest frequency of appearance at a root. The root has a frequency that is the sum of the frequency of the two constituent trees.
3. Repeat step 2 until only one binary tree remains. In order for the selection of two trees to be combined to take place quickly, then all existing trees are always sorted up based on frequency.
4. Label each side on a binary tree. The left side is labeled with 0 and the side Right is labeled with 1. The leaf nodes on the binary tree express the grayish value contained in in the original image. To encode every pixel inside an image, Take the following fifth step.

5. Trace the binary tree from root to leaf. Rows of side labels from root to root the leaf expresses huffman code for corresponding degrees of grayness. Each Huffman code is a prefix code, which means there is no binary code. a grayish value that is the prefix for the arithmetic degree binary code other. In this way, there is no ambiguity in the compression process image.

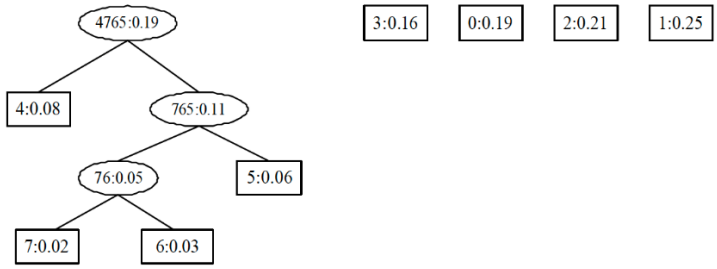
Suppose there is an image measuring 64 x 64 with 8 degrees grayness (k) and the sum of all pixels (n) = 64 x 64 = 4096, as follow:

k	n_k	$p(k) = n_k/n$
0	790	0.19
1	1023	0.25
2	850	0.21
3	656	0.16
4	329	0.08
5	245	0.06
6	122	0.03
7	81	0.02

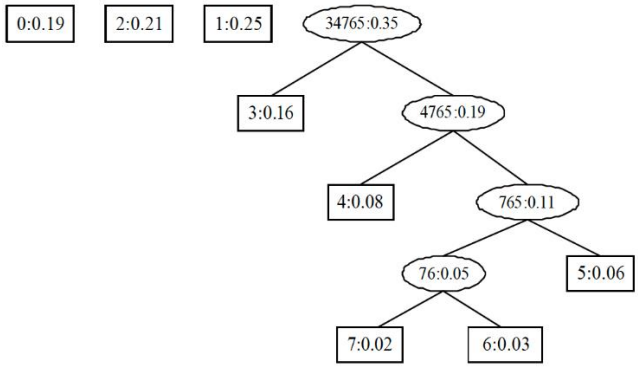
The process of forming the formed Huffman tree can be seen below.



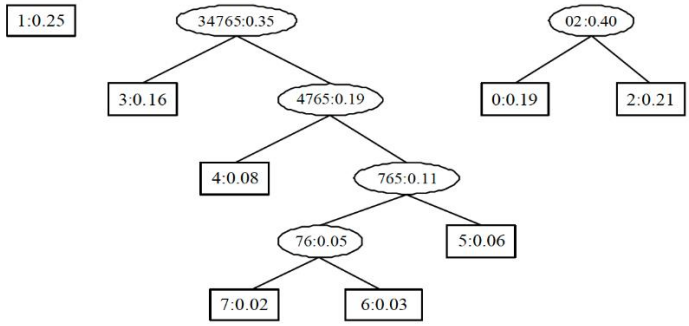
4.


4765:0.19
3:0.16
0:0.19
2:0.21
1:0.25
4:0.08
765:0.11
76:0.05
5:0.06
7:0.02
6:0.03

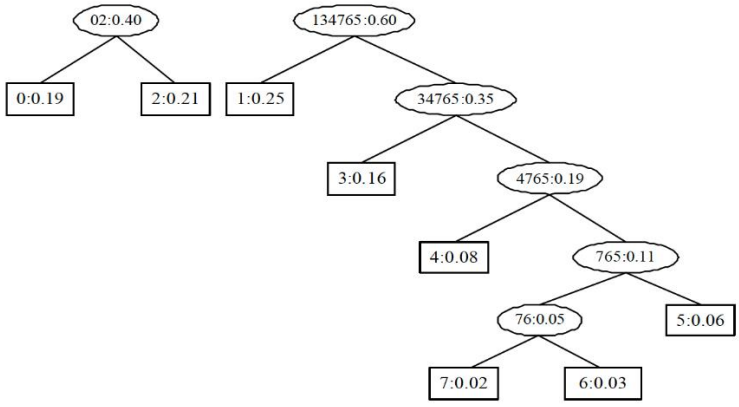
5.


0:0.19
2:0.21
1:0.25
34765:0.35
3:0.16
4765:0.19
4:0.08
765:0.11
76:0.05
5:0.06
7:0.02
6:0.03

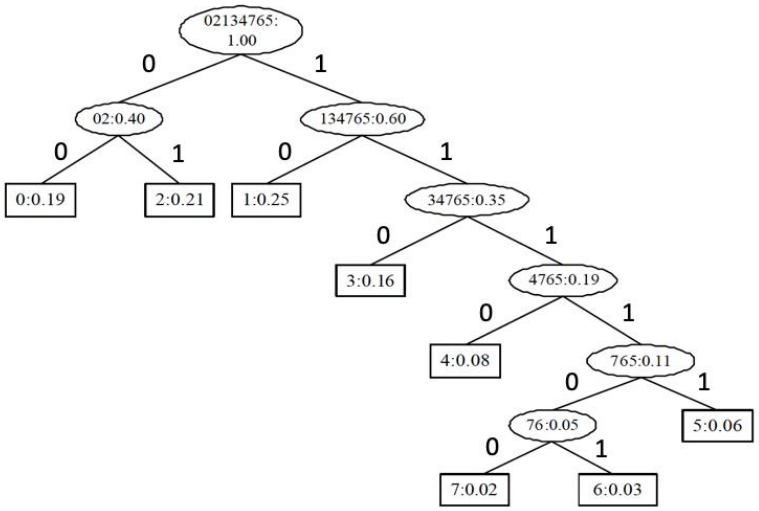
6.


1:0.25
34765:0.35
02:0.40
3:0.16
4765:0.19
0:0.19
2:0.21
4:0.08
765:0.11
76:0.05
5:0.06
7:0.02
6:0.03

7.


02:0.40
134765:0.60
0:0.19
2:0.21
1:0.25
34765:0.35
3:0.16
4765:0.19
4:0.08
765:0.11
76:0.05
5:0.06
7:0.02
6:0.03

8.



Each node in the tree contains a pair of values a:b, which in this case a expresses the value of the grayness and b expresses the probability of the appearance of the value The validity is in the image. From the Huffman tree we get the code for each degree of grayness is as follows:
0 = 00
1 = 10
2 = 01
3 = 110
4 = 1110
5 = 11111
6 = 111101
7 = 111100

Image size before compression (1 degree of grayness = 3 bits) is 4096x3 bits = 12288 bits,
while Image size after compression:
(790x2 bits) + (1023x2 bits) + (850x2 bits) + (656x3 bits) + (329x4 bits) + (245x5 bits) + (122x6 bits) + (81x6 bits) = 11053 bits

So, the memory requirement has been reduced from 12288 bits to 11053 bits. Clearly this doesn't save much, but if 256 grayish

values are used (compared to 8 degrees of grayness as in the example above), savings memory can get bigger.

Compression ratio = $100\% - \left(\frac{11053}{12288} x\ 100\%\right) = 10\%$

which means 10% of the original image has been compressed.

F. Run-Length Encoding (RLE) Compression Method

The RLE method is suitable for compressing images that have the pixel groups are the same grayness. Image compression with The RLE method is performed by creating a series of value pairs (p, q) for each line of pixels, the first value (p) represents the degree of grayness, while the value the second (q) expresses the number of consecutive pixels that have a degree of grayness it (called run length).

Review the 10 x10 pixel image with 8 degrees of grayness expressed as a matrix of degrees of grayness as follows:

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

There are all 100 pieces of value.

Value pairs for each run row generated by the compression method RLE:

(0, 5), (2, 5)
(0, 3), (1, 4), (2, 3)
(1, 10)
(4, 4), (3, 4), (2 2)
(3, 3), (5, 2), (7, 4), (6, 1)
(2, 2), (6, 1), (0, 4), (1, 2), (0, 1)
(3, 2), (4, 2), (3, 1), (2, 2), (1, 2)
(0, 8), (1, 2)
(1, 4), (0, 3), (2, 3)
(3, 3), (2, 3), (1, 4)

there are 31 pairs of values or 31 x 2 = 62 values.

The size of the image before compression (1 degree of grayness = 3 bits) is 100 x 3 bits = 300 bits, while the size of the image after compression (degree of grayness = 3 bits, run length = 4 bits):
(31 x 3) + (31 x 4) bit = 217 bits

Compression ratio = $(100\% - \frac{217}{300} \times 100\%) = 27.67\%$

which means 27.67% from the original image has been compressed.

Another version of the RLE method is to state an entire line of image become a run line, then calculate the run-length for each degree consecutive legitimacy. For example, review an image as follows:
1 2 1 1 1 1
1 3 4 4 4 4
1 1 3 3 3 5
1 1 1 1 3 3
State as a row of grayish degree values:
1 2 1 1 1 1 3 4 4 4 4 1 1 3 3 3 5 1 1 1 1 3 3
There are all 24 values.

Pairs of values of the run generated by the RLE compression method:
(1, 1) (2, 1) (1, 5) (3, 1) (4, 4) (1, 2) (3, 3) (5, 1) (1, 4) (3, 2)

Coding results:
1 1 2 1 1 5 3 1 4 4 1 2 3 3 5 1 1 4 3 2
There are all 20 values. So, we have saved 4 pieces of value.

The RLE method can be combined with Huffman's method for coding values of RLE compression results to increase the

compression ratio. At first do RLE compression, then the result is compressed again with the method Huffman.

G. Quantization Compression Method

This method reduces the number of degrees of grayness, for example from 256 to 16, which of course reduces the number of bits needed to represent an image. Suppose P is the number of pixels in the original image, it will be compressed become n degrees of grayness. The algorithm is as follows: Quantization method algorithm:

1. Create a histogram of the original image (the image to be compressed).
2. Identify n groups in the histogram in such a histogram that each the group has about P/n pixels.
3. Declare each group with a degree of validity of 0 to n –1. Every pixel in the group are recoded with a degree of grayness value the new one.

Review images that are 5 x 13 pixels:

2 9 6 4 8 2 6 3 8 5 9 3 7
3 8 5 4 7 6 3 8 2 8 4 7 3
3 8 4 7 4 9 2 3 8 2 7 4 9
3 9 4 7 2 7 6 2 1 6 5 3 0
2 0 4 3 8 9 5 4 7 1 2 8 3

which will be compressed into an image with 4 degrees of grayness (0 to 3), so each degree of grayness is represented by 2 bits.

Original image histogram:

0 **
1 **
2 *********
3 ***********
4 *********
5 ****
6 *****
7 ********
8 *********
9 ******

There are 65 pixels, grouped into 4 groups of grayish degrees. Each group has an average of 65/4 = 16.25 pixels per group:

	0 **	
13	1 **	0
	2 *********	
20	3 **********	
	4 *********	1
	5 ****	
17	6 *****	2
	7 ********	
15	8 *********	3
	9 ******	

Image after compressed to:

0 3 2 1 3 0 2 1 3 2 3 1 2
1 3 2 1 2 2 1 3 0 3 1 2 1
1 3 1 2 1 3 0 1 3 0 2 1 3
1 3 1 2 0 2 2 0 0 2 2 1 0
0 0 1 1 3 3 2 1 2 0 0 3 0

Image size before compression (1 degree of grayness = 4 bits):

65 x 4 bits = 260 bits

Image size after compression (1 degree of grayness = 2 bits):

65 x 2 bits = 130 bits

Compression ratio = $(100\% - \frac{130}{260} \times 100\%) = 50\%$

which means 50% of the image originally it has been compressed.

The disadvantage of quantization compression method is the amount of information that lost, but loss of this information can

be minimized by guaranteeing Each group has almost the same number of pixels.

H. Exercise:

Select one of the correct answers by marking the answer sheet.

1. What's not right about image compression is:
   a. Image compression aims to minimize memory requirements in representing digital images
   b. Image compression reduces data redundancy in the image
   c. Image encoding that produces less memory representation than the original representation
   d. Image compression returns the compressed image to bitmap data
   e. GIF and JPEG image formats are uncompressed image formats

2. What is image decompression:
   a. The process of reducing the memory size of a digital image
   b. Image compression restores the compressed image to JPEG or GIF format
   c. The compressed image must be able to be returned to an incompressible representation
   d. Images in incompressible representations are encoded with representations that minimize memory requirements
   e. The process of changing the image format from JPEG or GIF to bitmap

3. Examples of applications that take advantage of image compression, except:
   a. Download images from the internet
   b. Pixel thresholding application
   c. Fax
   d. Sending images via the WhatsApp application
   e. Video storage on VCD

4. Which is not an image compression criteria, namely:
   a. Compression time

b. Memory requirement
c. Output format
d. Brightness
e. Fidelity

5. Which is not a type of image compression, namely:
   a. Fractal Image Compression
   b. Prewitt-Sobel Filtering
   c. Huffman Coding
   d. Run-Length Encoding
   e. Quantizing Compression

CHAPTER 8
IMAGE SEGMENTATION

After taking this chapter, students should:

- Understand the basic concept of image segmentation
- Understand the methods used in image segmentation and how they work

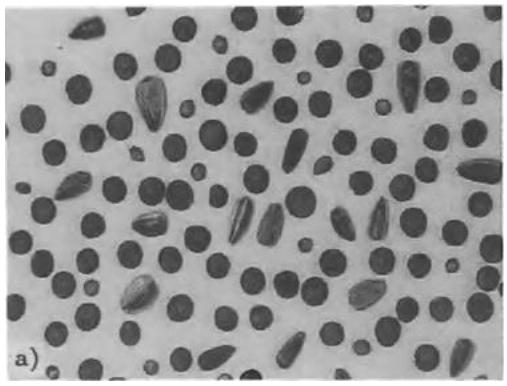
A. Preface

All image processing operations discussed so far have helped us to "recognize" objects of interest, i. e., to find suitable local features which allow us to distinguish them from other objects and from the background. The next step is to check each individual pixel whether it belongs to an object of interest or not. This operation is called segmentation and produces a binary image. A pixel has the value one if it belongs to the object; otherwise it is zero. Segmentation is the operation at the threshold between low-level image processing and the operations which analyze the shape of objects. In this chapter, we discuss several types of segmentation methods.

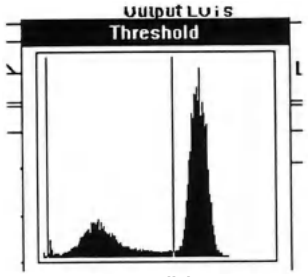
Basically, we can think of three concepts for segmentation. Pixel-based methods only use the gray values ofthe individual pixels. Edge-based methods detect edges and then try to follow the edges. Finally, region-based methods analyze the gray values in larger areas.

B. Pixel-Based Methods

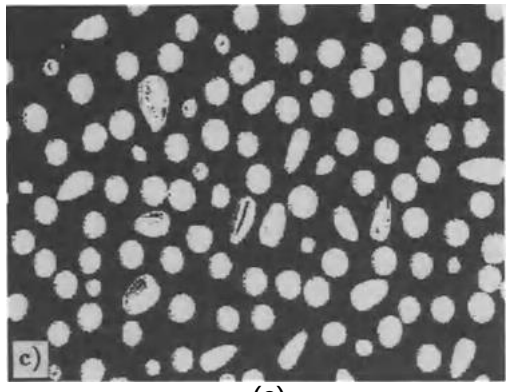
Point-based segmentation only takes the gray value of a pixel in order to decide whether it belongs to the object or not. In ordertoperform this task, we have to find the gray value range which characterizes the object of interest. In the following we will consider a simple scene with one type of object. If we have found a good feature to separate the object from the background, the histogram of the gray values- or more generally feature values- will showabimodal distribution with two distinct maxima like shown in the Figure 9.1 below:



(a)



(b)



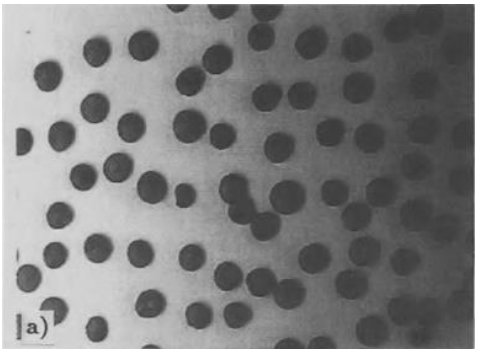
(c)

Figure 0.1 Ideal image for segmentation: a) Image; b) Histogram with bimodal gray value distribution; c) Segmented image with the threshold indicated by the verticalline in the histogram

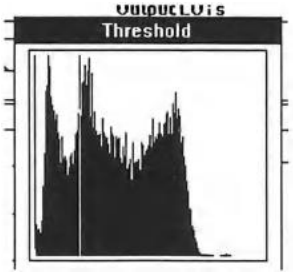
Ideally, a zone will exist between the two maxima where no features exist. Then the histogram is zero in this range and we

can place a threshold anywhere in this range yielding a perfect separation between object and background.

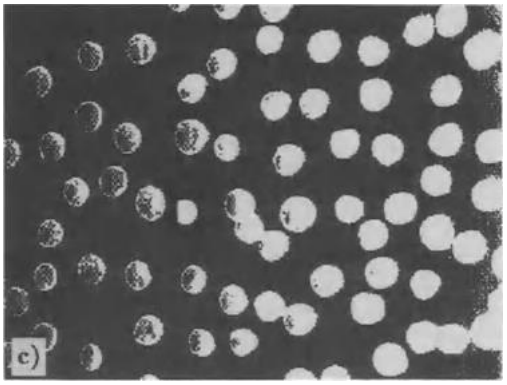
It is the aim oflow-level image processing to achieve this ideal situation. However, we may fall short. It is still quite simple to handle a situation in which a non-zero minimum exists between two well-pronounced maxima. This can occur even when the object and the background have clearly distinct gray values, since intermediate gray values will always occur at the edges of the objects. The probability density functions for the gray values of the object and the background will overlap. This overlap means that some mis-correspondences could be unavoidable. Some object pixels will be recognized as background and vice versa. If we know the probability distribution for the object and the background pixels, we can use a statistical analysis of the decision process to find an optimum threshold with the minimum number of erroneous correspondences. The gray value distributions of the object and the background can be estimated by local histograms which only include areas from either the object or the background. These areas must be selected manually. In less favorable circumstances, the histogram might not show a minimum at all, in which case an adequate threshold does not exist. This situation occurs, for example, in a scene with uneven illumination even if the object clearly juts out of the background like shown in Figure 9.2 below:



(a)



(b)



(c)

Figure 0.2 Segmentation of an image with uneven illumination: a) Image; b) Histogram; c) Segmented image

The literature is full of concepts to handle such cases. We can, for example, compute histograms from smaller areas and use these histograms to find local thresholds.

However, it is much better to solve the problern at its root, i. e., to optimize the illumination of the scene we observe. If this is not possible, we should try to correct for the uneven illumination before we apply a complex segmentation procedure.

C. Region-Based Methods

A region can be classified as a group of connected pixels exhibiting similar properties. The similarity between pixels can be in terms of intensity, color, etc. In this type of segmentation, some predefined rules are present which have to be obeyed by a pixel in order to be classified into similar pixel regions. Region-based segmentation methods are preferred over edge-based segmentation methods in case of a noisy image. Region-Based

techniques are further classified into 2 types based on the approaches they follow:

(1) Region growing method

In the case of the Region growing method, we start with some pixel as the seed pixel and then check the adjacent pixels. If the adjacent pixels abide by the predefined rules, then that pixel is added to the region of the seed pixel and the following process continues till there is no similarity left. This method follows the bottom-up approach. In case of a region growing, the preferred rule can be set as a threshold.

For example: Consider a seed pixel of 2 in the given image and a threshold value of 3, if a pixel has a value greater than 3 then it will be considered inside the seed pixel region. Otherwise, it will be considered in another region. Hence 2 regions are formed in the following image based on a threshold value of 3.

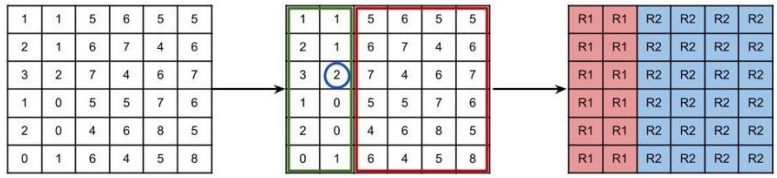


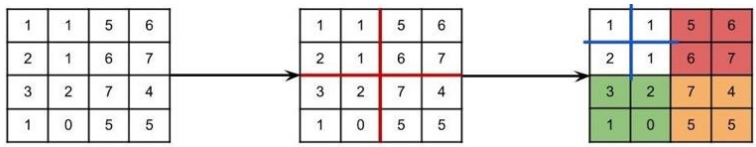
Figure 0.3 Region growing workflow
(a) Original image (b) Region growing process with 2 as the seed pixel (c) Splitting image into 2 regions based on a threshold

(2) Region splitting and merging method

In Region splitting, the whole image is first taken as a single region. If the region does not follow the predefined rules, then it is further divided into multiple regions (usually 4 quadrants) and then the predefined rules are carried out on those regions in order to decide whether to further subdivide or to classify that as a region. The following process continues till there is no further division of regions required i.e every region follows the predefined rules. In Region merging technique, we consider every pixel as an individual region. We select a region as the seed region to check if

adjacent regions are similar based on predefined rules. If they are similar, we merge them into a single region and move ahead in order to build the segmented regions of the whole image. Both region splitting and region merging are iterative processes. Usually, first region splitting is done on an image so as to split an image into maximum regions, and then these regions are merged in order to form a good segmented image of the original image.

In case of Region splitting, the following condition can be checked in order to decide whether to subdivide a region or not. If the absolute value of the difference of the maximum and minimum pixel intensities in a region is less than or equal to a threshold value decided by the user, then the region does not require further splitting.



(a) (b) (c)

Figure 0.4 Region splitting and merging workflow
(a) Original image (b) Region splitting into 4 quadrans (c) Classifying quadrant as a region

D. Edge-Based Methods

Edge-based segmentation relies on edges found in an image using various edge detection operators. These edges mark image locations of discontinuity in gray levels, color, texture, etc. When we move from one region to another, the gray level may change. So if we can find that discontinuity, we can find that edge.

A variety of edge detection operators are available but the resulting image is an intermediate segmentation result and should not be confused with the final segmented image. We have to perform further processing on the image to the segment it. Additional steps include combining edges segments obtained into one segment in order to reduce the number of segments rather than chunks of small borders which might hinder the

process of region filling. This is done to obtain a seamless border of the object. The goal of edge segmentation is to get an intermediate segmentation result to which we can apply region-based or any other type of segmentation to get the final segmented image.

E. Exercise:

Select one of the correct answers by marking the answer sheet.

1. The purpose of image segmentation, namely:
   a. Separating an object from its background (1)
   b. Partition the image into several non-overlapping regions with homogeneous properties (2)
   c. Identify the shape of the object found in the image (3)
   d. Answers (1), (2) and (3) are correct
   e. Answers (1) and (2) are correct

2. Things that affect an image can be segmented except:
   a. Color
   b. Brightness
   c. Texture
   d. Intensity
   e. Frequency

3. Methods that are widely used in image segmentation, except:
   a. Huffman Coding
   b. Split and Merge
   c. Clustering
   d. Thresholding
   e. Region Growing
4. The threshold value depends on the neighboring pixels, not all pixels are the threshold type...
   a. Local threshold
   b. Global threshold
   c. Adaptive threshold
   d. Log threshold
   e. Multilevel threshold

5. Threshold value depends on the whole pixel is the type of threshold…
    a. Local threshold
    b. Global threshold
    c. Adaptive threshold
    d. Log threshold
    e. Multilevel threshold

BIBLIOGRAPHY

Tyagi, M. 2021. *Toward Data Science: Image Segmentation part 2*. Toward Data Science: https://towardsdatascience.com/image-segmentation-part-2-8959b609d268

Sachdev, A. 2019. *Medium: Spatial and Frequency Domain — Image Processing*. 2022. Medium: https://medium.com/vithelper/spatial-and-frequency-domain-image-processing-83ffa3fc7cbc

Jayathilake, C., Perera, A., & Chamikara, M. 2013. Discrete Walsh-Hadamard Transform in Signal Processing. *International Journal of Research in Information Technology*, 80-89.

Gonzales, R. C., & Woods, R. E. 2018. *Digital Image Processing Fourth Edition.* New York: Pearson.

K. Jain, A. 1989. *Fundamentals of Digital Image Processing.* United State of America: Prentice Hall.

Jahne, B. 1995. *Digital Image Processing: Concepts, Algorithm and Scientific Application.* New York: Springer.

Burger, W., & J. Burge, M. 2009. *Principles of Digital Image Processing: Core Algorithms.* London: Springer.

Mutiara, A. B. 2005. *Pengolahan Citra Digital.* Jakarta: Universitas Gunadarma.

Munir, R. 2005. *Pengolahan Citra Digital dengan Pendekatan Algoritmik.* Bandung: Informatika.

Santoso, T. B. 2004. *Pengolahan Citra Digital.* Surabaya: PENS .

Marleny, F. D. 2021. *Pengolahan Citra Digital Menggunakan Python.* Banjarmasin: CV. Pena Persada.

GLOSSARY

A

Addition	
	A point process that blends the values of corresponding pixels in two input images. A single parameter controls which input image dominates the output image.
Area Processes	
	A category of image-processing techniques that calculate the value of each output-image pixel from the corresponding input-image pixel and its neighbors. Examples include halftoning, sharpening and median filtering.
Average Greyscale	
	The shade of grey produced by adding the brightnesses of the RGB components of a colour pixel and dividing by three.

B

Bitmap Image	
	An image composed of black and white pixels.
Blurring	
	An area process that produces an effect similar to an out-of-focus photograph. Blurring removes the detail in an image by making each pixel more like its neighbors.
Brightness	
	Determines the intensity of the colour presented by a pixel in a colour image, or the shade of grey presented by a pixel in a greyscale image.
Brightness Transformation	
	A point process that maps input brightnesses onto output brightnesses with a linear or non-linear mathematical function.

C

Capture	
	The process of measuring and visualizing physical phenomena such as visible and non-visible electromagnetic radiation. Examples include taking a photograph and scanning a document.
Closing	

	A morphological operation produced by following a dilation by an erosion. Often used for filling holes in bitmap images.
Colour Model	
	Determines how the colour in a digital image is represented numerically. Examples include the RGB and HSB colour models.
Composition	
	A point process that overlays the pixels of a foreground input image onto the pixels of a background input image.
Contrast	
	The difference between the lightest and darkest regions of an image.
Contrast Expansion	
	An image-processing technique that re-distributes the brightness in an image to eliminate regions that are either too dark or too light. Examples include basic and ends-in contrast expansion.
Convolution	
	A method of calculating the new value of a central pixel in a neighborhood by multiplying each pixel in the neighborhood by a corresponding weight; the new value of the central pixel is the sum of the multiplications.
Corrupted Pixel	
	A pixel value altered by noise.
Cropping	
	A geometric process that reduces the size of an image by discarding the pixels outside a specified region called the crop selection.

D

Digital Camera	
	An imaging device that focuses visible light onto a CCD.
Digital Image	
	An image captured by an imaging device and represented in a computer as a rectangular grid of pixels.
Dilation	

	A morphological operation that increases the size of objects in an image by adding a layer of foreground pixels around each object.

E

Edge	
	Edges mark the boundaries between the objects in a scene. A large change in pixel brightness over a small number of pixels often indicates the presence of an edge.
Edge Detector	
	An image-processing routine that flags the large changes in pixel brightness that indicate potential edges. Edge detectors often visualize their results in edge maps. Examples include the Sobel, Prewitt, Kirsch and Laplacian edge detectors.
Electromagnetic Spectrum	
	The complete range of electromagnetic radiation from short wavelength gamma radiation to long wavelength radio waves.
Erosion	
	A morphological operation that decreases the size of objects in an image by removing a layer of foreground pixels around each object. Often used for removing projections and blobs in bitmap images.

F

Flipping	
	A geometric process that swaps the pixels in an image across the horizontal, vertical and diagonal axes.
Frame Averaging	
	A point process that removes noise from a series of input images taken of the same subject. Each output-image pixel value is the average of the corresponding input-image pixel values.

G

Gaussian Noise	
	A form of image noise that adds small positive and negative deviations to the pixels in an image, often caused by the random variations between the elements of a CCD sensor. Plotting the number of occurrences of each deviation on a

	histogram produces the bell-shaped curve of the normal distribution, which is also called the Gaussian distribution.
Geometric Process	
	A category of image-processing techniques that change the size and shape of an image rather than its contents. Examples include cropping, scaling and rotation.
Greyscale Image	
	An image composed of pixels that present shades of grey.

H

High-contrast Image	
	An image with large numbers of pixels in the shadows and highlights.
High Frequency	
	The high frequency information in an image is represented by large changes in pixel brightness over a small number of pixels.
High-pass Filter	
	A filter that preserves or amplifies the high frequency information in an image. Sharpening is implemented by a high pass filter.
Histogram	
	The histogram of an image visualizes the distribution of the brightness in the image by plotting the number of occurrences of each brightness.
Histogram Equalization	
	An image-processing technique that reveals detail hidden in images with a poorly-distributed range of brightnesses.
Horizontal Axis	
	The line that runs through the centre of an image from the left of the image to the right.
HSB	
	A colour model that represents each colour with three numbers that specify the hue (H), the saturation (S) and the brightness (B) of the colour.
Hue	
	The colour in the HSB colour model.

I

Image	
	An image records a visual snapshot of the world around us.
Image Processing	
	The field of computer science that develops techniques for enhancing digital images to make them more enjoyable to look at, and easier to analyze by computers as well as humans.
Imaging Device	
	A piece of equipment that captures an image. Examples include digital cameras, side-scan sonar systems and scanning electron microscopes.
Interpolation	
	A method of creating new pixel values from existing pixel values. Examples include nearest-neighbor and bi-linear interpolation.
Input Image	
	The image transformed by an image-processing routine.

K

Kernel	
	A rectangular grid of convolution weights.

L

Line Edge	
	A line chain of pixels that separates a region of light pixels from a region of dark pixels.
Linear Brightness Transformation	
	A category of brightness transformations that lighten and darken images using mathematical functions with curved graphs.
Low-contrast Image	
	An image that uses only a small range of the available brightness. Low-contrast images are mostly dark, mostly dull or mostly light.
Low-key Image	
	An image that represents a naturally dark subject.
Low Frequency	

	The low frequency information in an image is represented by small changes in pixel brightness over a small number of pixels.
Low-pass Filter	
	A filter that discards or attenuates the high frequency information in an image and preserves the low frequency information. Removing the high frequency information from an image removes the detail and blurs the image. Blurring is implemented by a low pass filter.

M

Median Filtering	
	An area process that removes noise by replacing the central pixel in a neighborhood with the median pixel value of the neighborhood.
Morphological Operation	
	A category of image-processing techniques that operate on the structure of the objects in an image.

N

Noise	
	Unwanted changes to the values of the pixels in an image, often introduced by the imaging device during capture. Examples include impulse noise and Gaussian noise.
Non-linear Brightness Transformation	
	A category of brightness transformations that change the brightness of an image using mathematical functions with straight-line graphs. Examples include inversion and posterization.
Non-primary Colour	
	A colour created by mixing the red, green and blue primary colours of the RGB colour model.

O

Opening	
	A morphological operation produced by following an erosion by a dilation. Often used for filling holes in bitmap images.
Output Image	

	An image that contains the results of applying an image-processing routine to an image.

P

Photo Restoration	
	The application of a series of image-processing routines to enhance a damaged photograph.
Pixel	
	A square unit of visual information that represents a tiny part of a digital image.
Pixel Neighborhood	
	A region of pixels processed by an area process. Typical neighborhood dimensions are 3x3 pixels and 5x5 pixels.
Primary Colours	
	The colours red, green and blue from which all other colours in the RGB colour model are mixed.

Q

Quantization	
	The calculation that maps the fractional measurements made by imaging devices onto proportional integer pixel brightnesses.

R

Resolution	
	The number of pixels available to represent the details of the subject of a digital image.
RGB	
	A colour model that represents each colour with three numbers that specify the amounts of red (R), green (G) and blue (B) that produce the colour.
RGB Colour Cube	
	Visualizes the amounts of red, green and blue required to produce each colour in the RGB colour model as a point in a cube at co-ordinates (x, y, z).
Rotation	
	A geometric process that turns an image about its centre by a specified angle.

S

Sampling	
	The process of mapping a continuous quantity of electromagnetic radiation, such as light or X-rays falling on a sensor, onto a discrete, rectangular grid of pixels.
Saturation	
	The component of the HSB colour model that controls the amount of white mixed into the hue.
Scale Factor	
	A fractional number that controls whether a scaling process enlarges or reduces an image. Scale factors between zero and one reduce images; scale factors greater than one enlarge images.
Scaling	
	A geometric process that changes the size of an image.
Scanner	
	An imaging device that focuses light reflected from a document onto a CCD that moves across and down the document.
Shadows	
	The range of pixel brightnesses that represent the darker regions of an image.
Sharpening	
	An area process that emphasizes the detail in an image.
Step Edge	
	The ideal edge shape characterized by a large, immediate change in pixel brightness from a region of light pixels to a region of dark pixels.
Structuring Element	
	The rectangular grid of binary values used by morphological operations to assign a new value to a pixel in a bitmap image.
Subtraction	
	A point process that identifies the pixels that differ between two input images.

T

Thresholding	
	A point process that produces a bitmap version of a greyscale image. Black bitmap pixels represent greyscale pixels darker

	than a threshold brightness; white bitmap pixels represent greyscale pixels lighter than the threshold.
Thermal Camera	
	An imaging device that measures the amount of infra-red light emitted by the subject of the image. Hotter objects emit more infra-red light, which show up as the brighter regions of a thermal image.
Thermal Image	
	A greyscale image captured by a thermal camera. Often enhanced with pseudo-colour to assign the same colour to regions of pixels that represent similar temperatures.