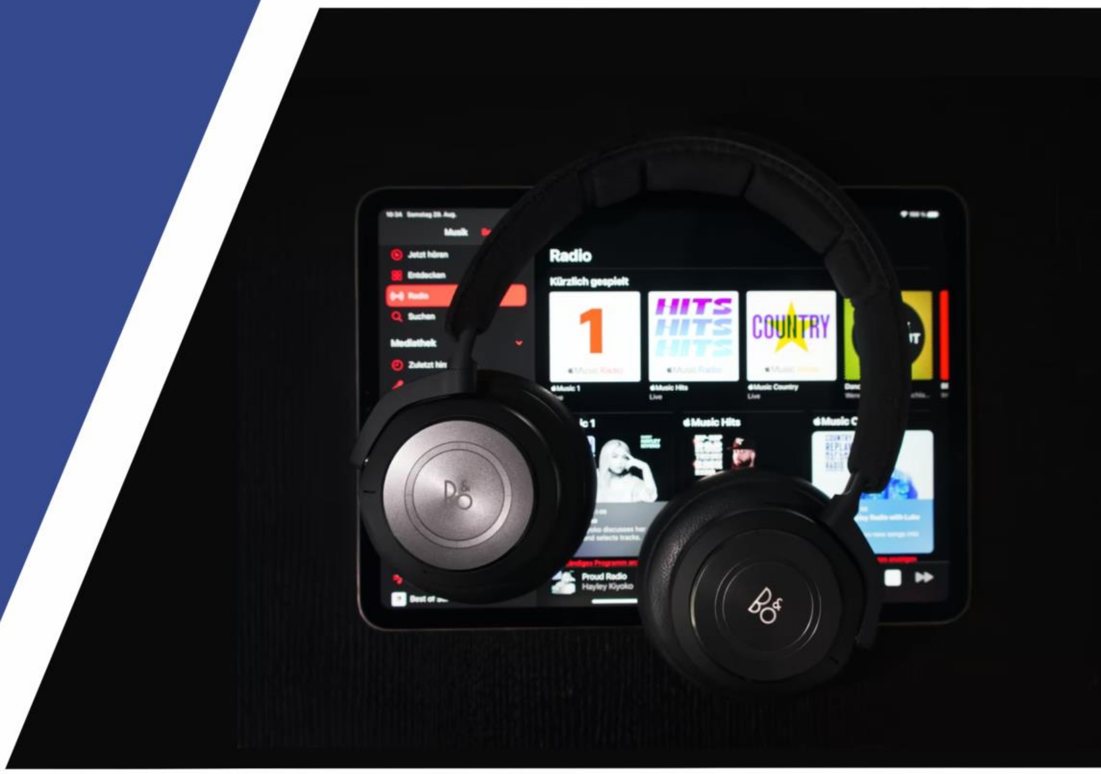
NATIONAL OPEN UNIVERSITY OF NIGERIA

CIT 891

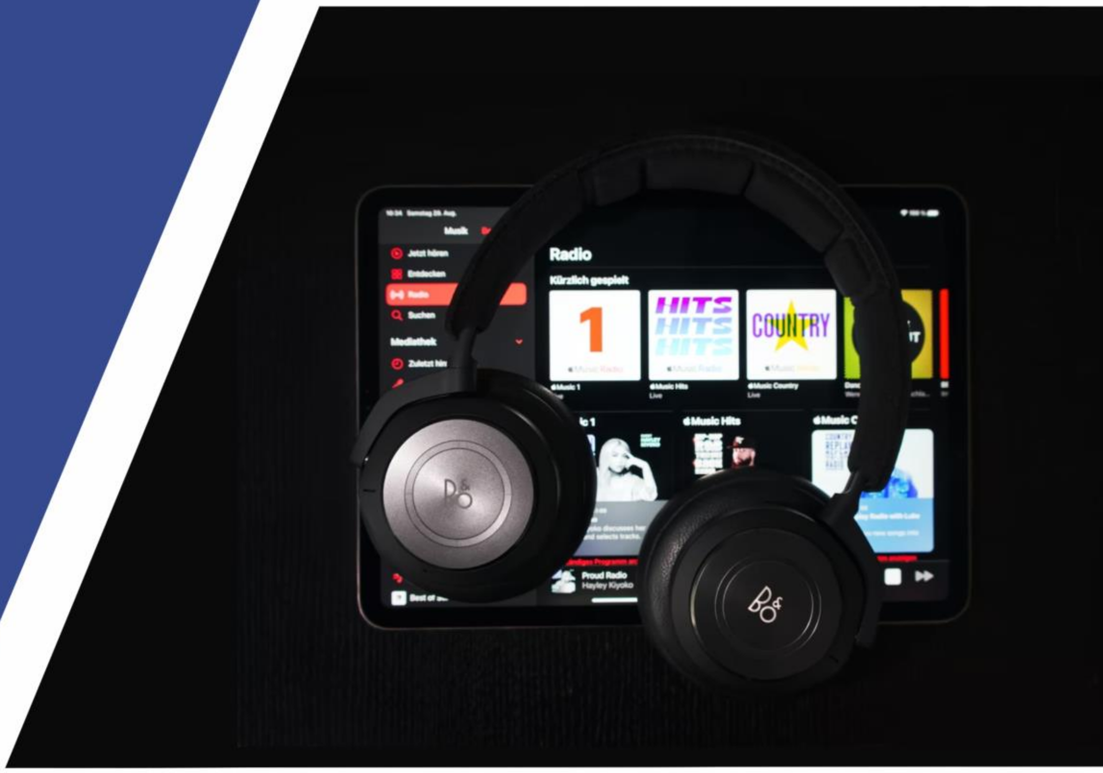


Advanced Multimedia Technology

Module 3

NATIONAL OPEN UNIVERSITY OF NIGERIA

CIT 891



Advanced Multimedia Technology

Module 3

CIT 891
Advanced Multimedia Technology
Module 3

Course Developer/Writer
Dr. Ikhu-Omoregbe, Nicholas, National Open University of Nigeria

Course Coordinator
A. A. Afolorunso, National Open University of Nigeria

Programme Leader
Prof. MoniOluwa Olaniyi, National Open University of Nigeria

Credits of cover-photo: Henry Ude, National Open University of Nigeria

National Open University of Nigeria - University Village, Plot 91, Cadastral Zone, Nnamdi Azikiwe Expressway, Jabi, Abuja-Nigeria.

www.nou.edu.ng centralinfo@nou.edu.ng
oer.nou.edu.ng oerunit@noun.edu.ng OER repository
Published in 2013, 2021 by the National Open University of Nigeria

How to re-use and attribute this content

Unit 1 Image Perception, Image Enhancement with Histogram Analysis

1.0 Introduction

for image enhancement.

2.0 Objectives

Image processing is a very important aspect of digital signal processing (DSP) application area. In this unit, we shall examine briefly the application of DSP techniques in the enhancement of images. To do this we shall first provide some mathematical foundations for histogram analysis and then apply them for the enhancement of images. Also in this unit, we will introduce the concept of filters and attempt to cover how they are used

At the end of this unit, you should be able to:
- explain the meaning of histogram
- describe how histograms are used for image enhancement
- identify sources of noise
- describe how filters are used for the removal of noise.

3.0 Main Content

3.1 Meaning of Histogram

Given a grayscale image, its histogram consists of the histogram of its grey levels; that is, a graph indicating the number of times each grey level occurs in the image. Quite a good number of inferences can be made from the appearance of an image from its histogram. The following statements are results from experiments with histograms.

In a dark image, the grey levels (and hence the histogram) would be clustered at the lower end:

In a uniformly bright image, the grey levels would be clustered at the upper end:

In a well contrasted image, the grey levels would be well spread out over much of the range:

Figure 1.1 shows a poorly contrasted image and its histogram. The histogram was obtained by using   MATLAB software. If you are familiar with MATLAB you can use it to perform a good number of image processing functions. For example, you can view the histogram of an image in MATLAB by using the **imhist** function. Many of the topics or diagrams discussed in this section can easily be obtained by using MATLAB provided you have your data or image to work with. Additional information is provided on MATLAB at the end of this unit.

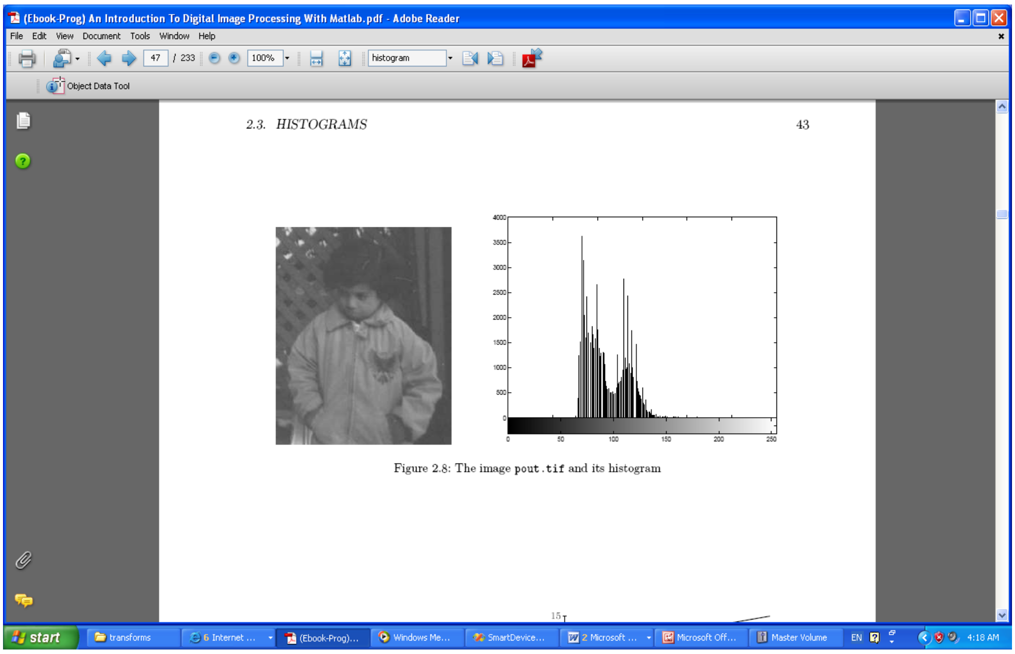


Fig. 1.1: An image with its Histogram

Source: McAndrew,A.(,2004)

3.2 Image Enhancement

Image enhancement is the processing of images to improve their appearance. There are a variety of methods, which are suitable for different objectives. Some objectives are to improve the image quality and visual appearance to human viewer. Other ones include the sharpening of an image to make the processed image better in some sense than the unprocessed one.

In the subsequent sections, we shall discuss some of the approaches to image enhancement. For simplicity, we only use gray-scale images.

3.2.1 Histogram Stretching (Contrast Stretching)

Whenever you have a poorly contrasted image such as shown in figure 1.1 above, you may wish to enhance its contrast by spreading out its histogram. One approach to use is by histogram stretching. Suppose we have an image with the histogram shown in figure 1.1, associated with a table of the numbers of gray values:

Table 1.1 a

Greylevel i	0	1	3	4	5	6	7	8	9	10	11	12	13	14	15
I	15	0	0	0	70	110	45	70	35	0	0	0	0	0	15

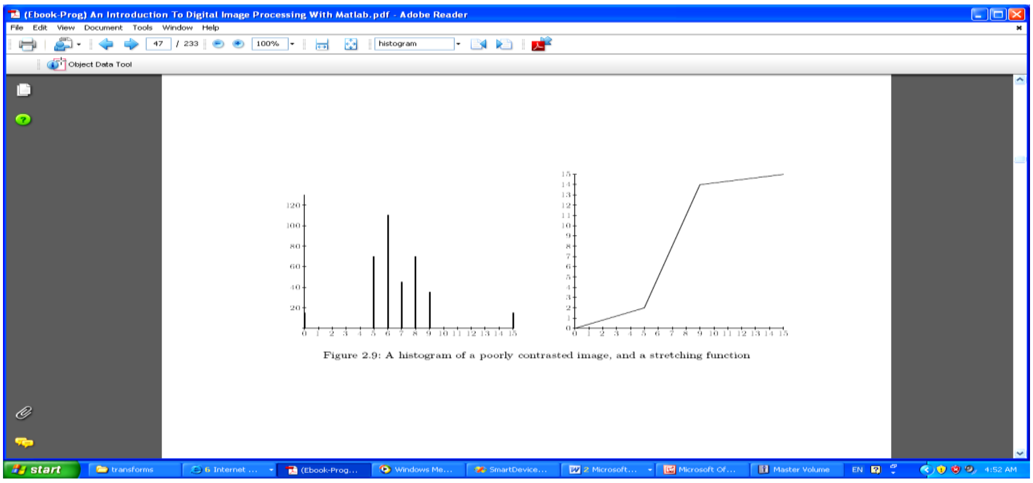


Fig. 1.2a: A Histogram of a Poorly Contrasted Image, and a Stretching Function McAndrew,A. (2004)

Assuming n =360, the grey levels in the centre of the range can be stretch out by using a piecewise linear function. Please, note that you can derive your piecewise linear function. The MATLAB software could assist you to do this. For example the function given by the following equation has the effect of stretching the grey levels 5-9 to grey levels 2-14 according to the equation

$$j = \frac{14-2}{9-5}(i-5)+2$$

where *i* is the original grey level and *j* its result after the function has been applied. Grey levels outside this range are either left alone as in the case under discussion or transformed according to the linear function derivable from the right side diagram in figure 1.1 This yields the corresponding grey values in table 1.1b and histogram in figure 1.2b which indicate an image with a greater contrast than the original.

Table 1.1 b

I	5	6	7	8	9
j	2	5	8	11	14

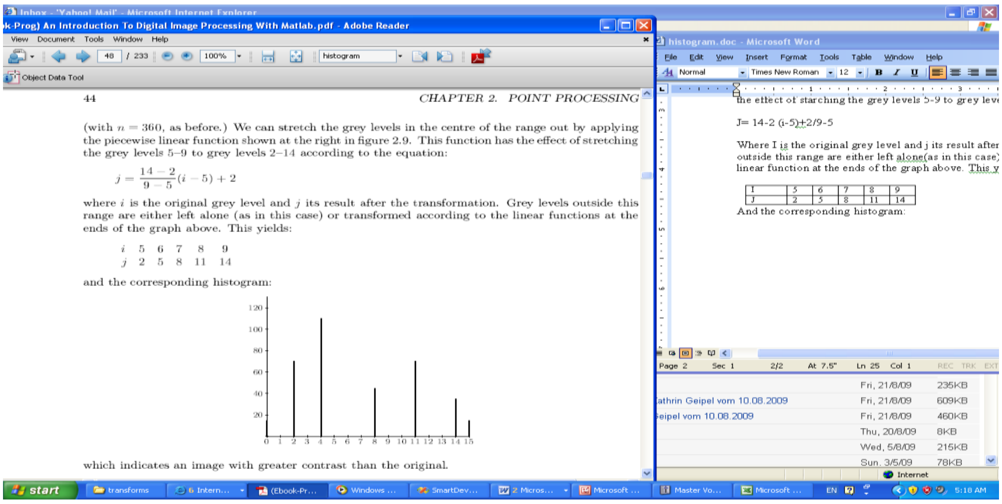


Fig.1.2b: Histogram of Image with Better Contrast

3.2.2 Histogram Equalisation

The major challenge with histogram stretching is that they require user input. Often times, a better approach to image enhancement is provided by histogram equalisation, which is an entirely automatic procedure. The principle here is to change the histogram to one which is uniform; that is one that every bar on the histogram is of the same height, meaning that, each grey level in the image occurs with the same frequency. In the real sense, this looks impracticable, but the fact remains that, the result of histogram equalisation provides very good results for enhancing the quality of an image.

Suppose we have an image with L different grey levels 0, 1, 2, …,.L-1 and that the grey level i occurs n_i times in the image. Assuming also that the total number of pixels in the image is n such that $n_1 + n_2 + n3 + \ldots + n_{L-1} = n$. To transform the grey level to obtain a better contrasted image we change grey level i to

$$\left(\frac{n_o + n_1 + \ldots + n_i}{n}\right)(L-1)$$

with the values obtained rounded to the nearest integer.

Let us now consider the following example in order to have a better understanding of the concept discussed above:

Suppose a 4-bit greyscale image has the histogram shown in figure 1.2 associated with a table of the number n_i of grey values shown in table1.2

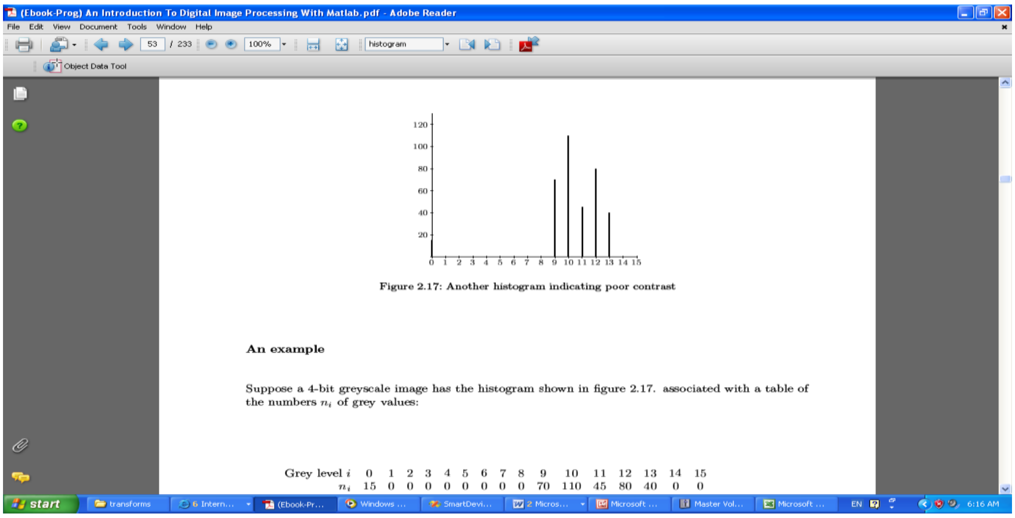


Fig. 1.3a: Histogram of Grey Scale Values before Equalisation

Table 1.2 Grey Scale Values of an Image

Greylevel i	0	1	3	4	5	6	7	8	9	10	11	12	13	14	15
I	15	0	0	0	0	0	0	0	70	110	45	80	40	0	0

We would expect this image to be uniformly bright, with a few dark dots on it. To equalise this histogram, we form running totals of the n_i, and multiply each by 15/360 = 1/24

Table 1.3: Equalised Values

Grey Level i	**n_i**	**∑ni**	**(1/24) ∑ni**	**Rounded Value**
0	15	15	0.63	1
1	0	15	0.63	1
2	0	15	0.63	1
3	0	15	0.63	1
4	0	15	0.63	1
5	0	15	0.63	1
6	0	15	0.63	1
7	0	15	0.63	1

8	0	15	0.63	1
9	70	85	3.63	4
10	110	195	8.13	8
11	45	240	10	10
12	80	320	13.33	13
13	40	360	15	15
14	0	360	15	15
15	0	360	15	15

This will give the following transformation of grey values obtained by reading off the first and last column in the above table:

Table 1.4: Original and Final Grey Values

Original grey level i	0	1	2	3	4	5	7	8	9	10	11	12	13	14	15
Final grey level j	1	1	1	1	1	1	1	1	4	8	10	13	15	15	15

The resulting histogram from the table 1.4 is depicted in Figure 1.4

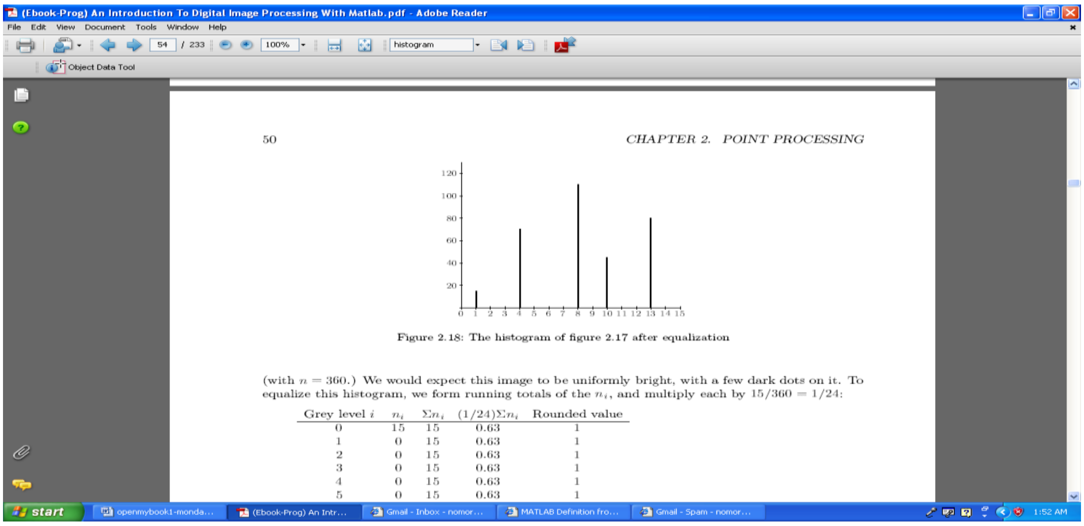


Fig. 1.4: Histogram of Grey Scale Values after Equalisation

3.2.3 Edge Sharpening

Another approach to making edges in an image slightly sharper and crisper, which generally results in an image more pleasing to the human eye, is by performing spatial filtering. The operation is also known as “edge enhancement”, “edge crispening”, or “unsharp masking”. This last term comes from the printing industry. Unsharp masking is well known to photographers and astronomers who used the method as darkroom technique to enhance faint details in photographic prints. As designed by photographers, blurred, reverse-contrast negative (or unsharp mask) is made of the original negative. These two negatives are sandwiched together in perfect registration in the enlarger and a print is made. To perform this operation on a computer, an unsharp mask is produced by blurring and reducing the amplitude of the original image; the unsharp mask is then subtracted from the original to produce a sharpened image. The idea of unsharp masking is to subtract a scaled “unsharp” version of the image from the original. In practice, we can achieve this effect by subtracting a scaled blurred image from the original.

The schema for unsharp masking is shown in figure 1.5a

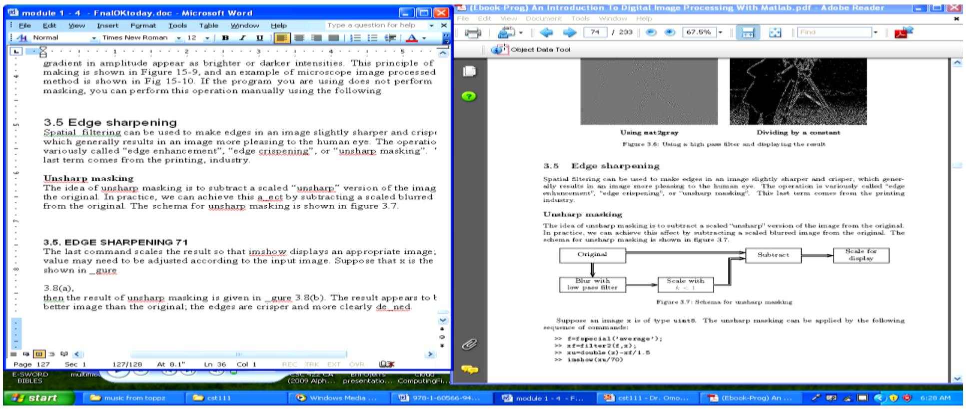


Fig. 1.5a: Schema for Unsharp Masking

	
An Original Image	After Unsharp Masking

Fig.1.5b: Schema for Unsharp Masking

3.4 Filters and Noise Removal in Images

Noise is often introduced during the analog-to-digital conversion process as a side-effect of the physical conversion of patterns of light energy into electrical patterns. Filters usually would have some effect on image processing tasks. To be able to choose the most appropriate filter for image processing, we need to understand the notion of frequency. Roughly speaking, the frequencies of an image are a measure of the amount by which grey values change with distance. High frequency components are characterised by large changes in grey values over small distances; examples of high frequency components are edges and noise. Low frequency components, on the other hand, are parts of the image characterised by little change in the grey values. These may include backgrounds, skin textures. We then say that a filter is a:

high pass filter if it "passes" over the high frequency components, and reduces or eliminates low frequency components, and a

low pass filter if it "passes over" the low frequency components, and reduces or eliminates high frequency components.

Filtering operation selectively reduces or enhances low or high spatial frequency in the object image. This means, that a Low pass filtering, otherwise known as "smoothing", can be employed to remove high spatial frequency noise from a digital image. We shall revisit the topic on noise removal in unit 3 of this module.

3.5 Using MATLAB

MATLAB stands for MATrix LABoratory. It is a programming language for technical computing from The MathWorks, Natick, MA. It is used for a wide variety of scientific and engineering calculations, especially for automatic control and signal processing, MATLAB runs on Windows, Mac and a variety of Unix-based systems. Developed by Cleve Moler in the late 1970s and based on the original LINPACK and EISPACK FORTRAN libraries, it was initially used for factoring matrices and solving linear equations. MATLAB has wide area of application today. Most of the image processing tasks can be handled by it with a little programming skill. You can get a copy from the Internet or other sources for your PC and explore its features.

Self- Assessment Exercise

1. What is the meaning of Edge Sharpening?
2. Access the Internet for more information on MATLAB

4.0 Conclusion

The unsharp filter is a simple sharpening operator which derives its name from the fact that it enhances edges (and other high frequency components in an image) via a procedure which subtracts an unsharp, or smoothed, version of an image from the original image. The unsharp filtering technique is commonly used in the photographic and printing industries for crispening edges.

5.0 Summary

In this unit, we covered histogram and its application in image enhancement. We also took a look at filters and introduce the features of MATLAB which is a software that is currently being used for diverse engineering, science and business applications.

6.0 Self-Assessment Exercise

1. What is a histogram?
2. Describe the processes of Histogram equalisation in image enhancement
3. Explain the term “Edge Sharpening”

7.0 References/Further Reading

Jain,A.K. (1989). *Fundamentals of Image Processing.* Englewood Cliffs, NJ:, Prentice-Hall,

McAndrew,A. (2004). *An Introduction to Digital Image Processing with Matlab* Notes for SCM2511 Image, School of Computer Science and Mathematics, Victoria University of Technology

Mitra,S.& Tinkuacharya, (2003). *Data Mining Multimedia, Soft Computing, and Bioinformatics*. Canada: John Wiley & Sons, Inc, Hoboken. .

Netravali,A.N.& Haskell, B. (1988). *Digital Pictures*. New York: Plenum Press.

Pennenbaker,W.B.& Mitchell,J.L.(1993). JPEG: *Still Image Data*

Compression New York: . StandardChapman & Hall

Unit 2 Morphological Operators

1.0 Introduction

Morphology is a branch of image processing which is particularly useful for analysing shapes in images. We shall examine basic morphological operators that are applied to binary and greyscale images.

2.0 Objectives

At the end of this unit, you should be able to:

- explain the meaning of morphological operators
- .explain the applications of morphological operations in image processing.

3.0 Main Content

3.1 Morphological Operators

The theory of mathematical morphology can be developed in many different ways. We shall adopt one of the standard methods which use operations on sets of points. Some morphological operations are discussed as follows:

3.1.1 Translation and Reflection

3.1.1.1 Translation

Given that A is a set of pixel in a binary image and *w*= *(x,y)* is a particular coordinate point.

Then A_w is the set *A* "translated " in the direction *(x,y)*. This means that: $A^x = \{(a,b)+(x,y): (a,b) \in A\}$

Let us consider a practical example by using figure 2.1a

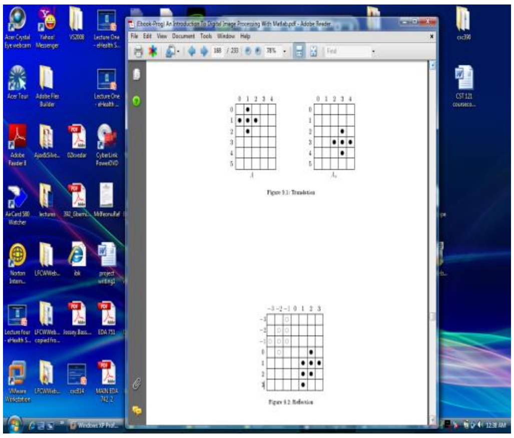


Fig.2.1a: Binary Image

The figure depicts a shaped set and let *w* = *(2,2)*. The set *A* has been shifted in the *x* and *y* directions by the values given in *w*. We observe that here we are using matrix coordinates, rather than Cartesian coordinate, so that the origin is at the top left, *x* goes down and *y* goes across. The result of the translation operation is depicted in figure 2.1b

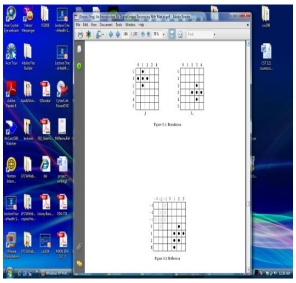


Fig.2.1b: Result of Translation Operation

3.1.1.2 Reflection

If *A* is set of pixel, then its reflection, denoted by *A* is obtained by reflecting *A* in the origin

$$A = \{(-x, -y) : (x, y) \in A\}$$

For example, in Figure 2.1c the open and close circles form sets which are reflections of each other.

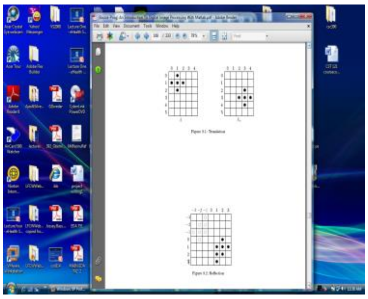


Fig. 2.1c: Result of Reflection Operation

3.1.2 Dilation and Erosion

These are the two basic operations that constitute morphology. All other operations are derived from these two operations.

3.1.2.1 Dilation

Assuming that *A* and *B* are sets of pixels, the dilation of *A* by *B*, denoted by $A \oplus B$ is defined as $A \oplus B = \bigcup_{x \in B} A_x$

This implies that for every point $x \in A$ we translate A by those coordinates. Then we take the union of all these translations. This can also be written as

$A \oplus B = \{(x,y) + (u,v) : (x,y) \in A, (u,v) \in B\}$

From this last definition, dilation is shown to be commutative; that is

$A \oplus B = B \oplus A$

An example of dilation is given in Figure 2.2a. In the translation diagram; the grey squares show the original position of the object. Note that A $_{(0,0)}$ is of course just A itself.

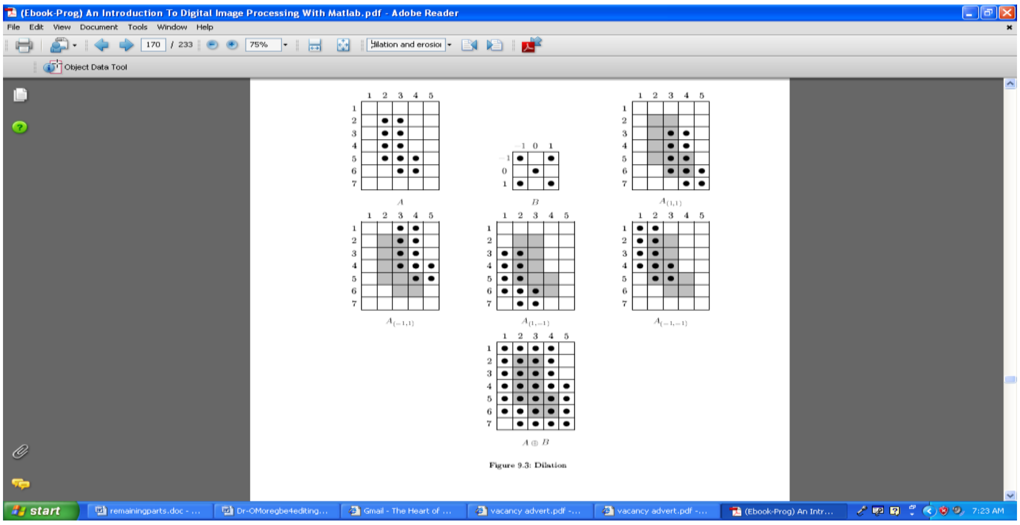


Fig.2.2a: Dilation

Source McAndrew, A.(2004)

In this example, we have

$B = \{(0,0), (1,1), (-1,1)(1,-1), (-1,-1)\}$ and these are the coordinates by which we translate A.

In general $A \oplus B$ can be obtained by replacing every point *(x,y)* in A with a copy of B. placing the (0,0) point of B at (x,y). Equivalently, we can replace every point (u,v) of B with a copy of A.

3.1.2.2 Erosion

Given sets *A* and *B*, the erosion of *A* by *B*, written $A \ominus B$ is defined as

$A \ominus B = \{w: B_w \subseteq A\}$

In other words, the erosion of A by *B* consist of all points w = *(x,y)* for B_w is in A. To perform an erosion, we can move *B* over *A*, and find all the places it will fit, and for each place mark down the corresponding *(0,0)* point of B. The set of all such points will form the erosion. An example of erosion is given in Figure 3.2

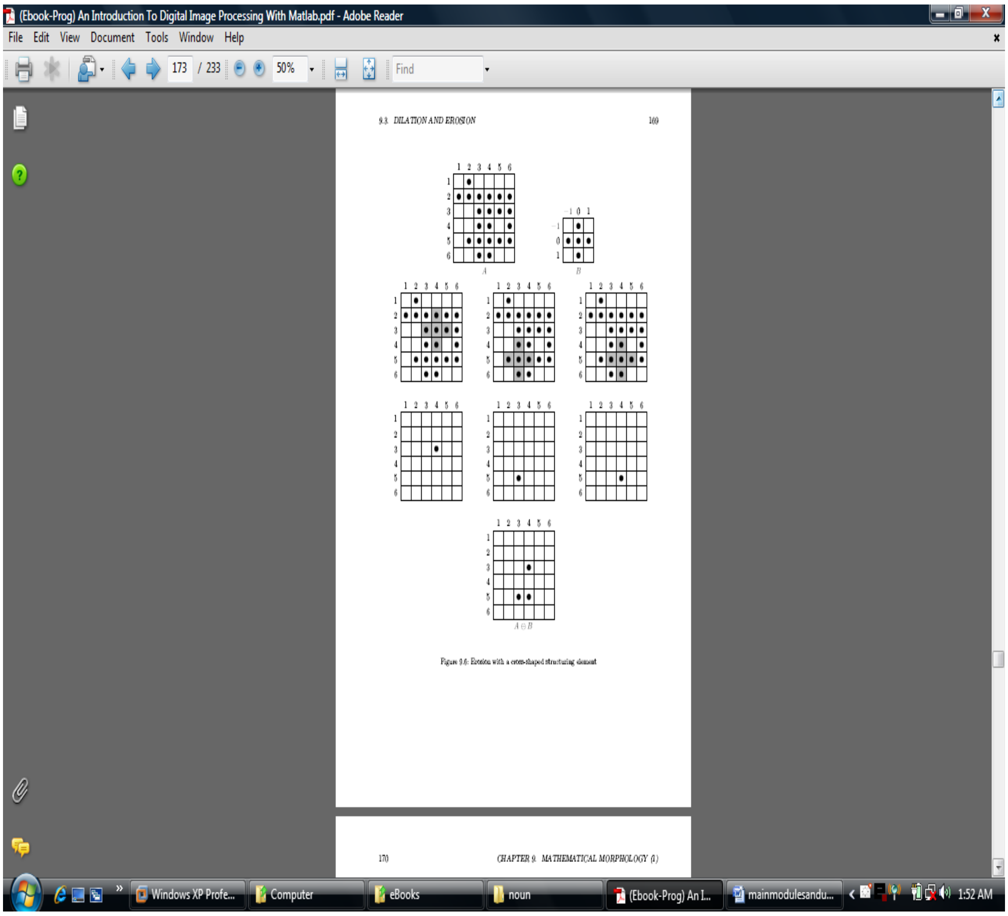


Fig. 3.2b: Erosion

3.2 Relations between Erosion and Dilation

It can be shown that erosion and dilation are "inverses" of each other; more precisely, the complement of erosion is equal to the dilation of the complement. Thus

$$\overline{A \Theta B} = \overline{A} \oplus \hat{B}$$

3.3 Opening and Closing

These operations may be considered as "second level" operations; in that they build on the basic operations of dilation and erosion. They are also, as we shall see, better behaved mathematically.

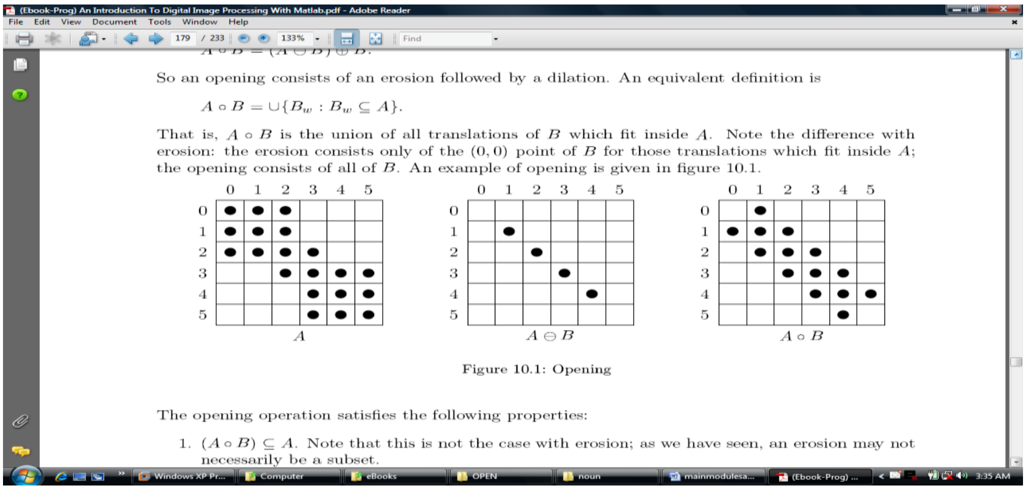
Opening

Given A and a structuring element B, the opening of A by B, denoted by $A \circ B$ is defined as
$A \circ B = (A \Theta B) \oplus B$

So an opening consists of an erosion followed by a dilation. An equivalent definition is

$A \circ B = \{ B_w : B_w \subseteq A\}$

That is $A \circ B$ is the union of all translations of B which fit inside A. Note the difference with erosion: the erosion consist only of the (0,0) point of B for those translations which fit inside A; the opening consist of all of B. An example of opening is given in Figure 3.3a


(Ebook-Prog) An Introduction To Digital Image Processing With Matlab.pdf - Adobe Reader

So an opening consists of an erosion followed by a dilation. An equivalent definition is

$A \circ B = \cup\{B_w : B_w \subseteq A\}.$

That is, $A \circ B$ is the union of all translations of B which fit inside A. Note the difference with erosion: the erosion consists only of the $(0,0)$ point of B for those translations which fit inside A; the opening consists of all of B. An example of opening is given in figure 10.1.

Figure 10.1: Opening

The opening operation satisfies the following properties:

1. $(A \circ B) \subseteq A$. Note that this is not the case with erosion; as we have seen, an erosion may not necessarily be a subset.

Fig. 3.3a: Opening

Source McAndrew,A. (2004)

The opening operation satisfies the following properties:

1 $(A \circ B) \subseteq A$. Note that this is not the case with erosion; as we have seen, an erosion may not necessarily be a subset

2 $(A \circ B) \circ B = A \circ B$). That is, an opening can never be done more than once. This property is called idempotence. Again, this is not the case with erosion; you can keep on applying a sequence of erosion to an image until nothing is left.

3 $(A \subseteq C\}$. Then $(A \circ B) \subseteq (C \circ B)$.

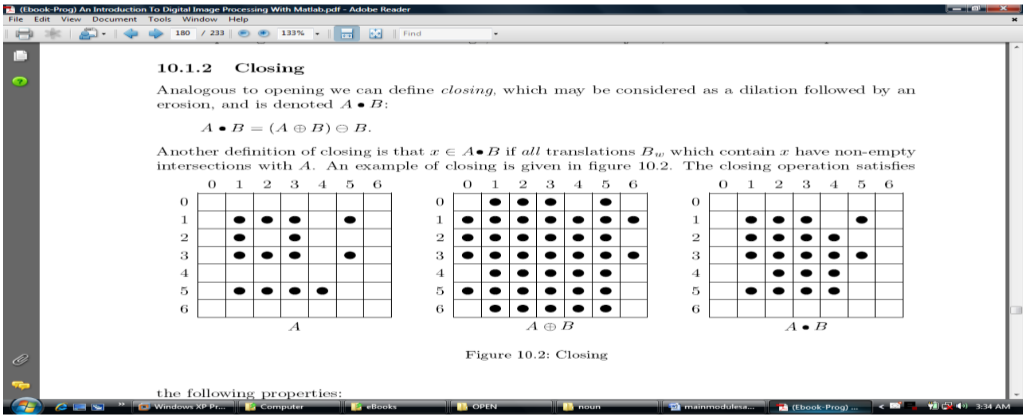
4 Opening tends to "smooth" an image, to break narrow joins, and to remove thin protrusions.

Closing

Related to opening we can define closing, which is considered as a dilation followed by an erosion. It is denoted by $A \bullet B$;

$A \bullet B = (A \oplus B) \ominus B$

Another definition of closing is that $x \in A \bullet B$ if all translations Bw which contains x have non-empty intersections with A. An example of closing is given in figure 3.3b. This closing operation satisfies



10.1.2 Closing

Analogous to opening we can define *closing*, which may be considered as a dilation followed by an erosion, and is denoted $A \bullet B$:

$$A \bullet B = (A \oplus B) \ominus B.$$

Another definition of closing is that $x \in A \bullet B$ if *all* translations B_w which contain x have non-empty intersections with A. An example of closing is given in figure 10.2. The closing operation satisfies

Figure 10.2: Closing

the following properties:

Fig. 3.3b: Closing

1. A $\subseteq$(A • B)

2. (A • B) • B = A • B; that is, closing, like opening, is idempotent

3 If A $\subseteq$C, then (A • B) $\subseteq$(C • B)

4 Closing tends also to smooth an image, but it fuses narrow breaks and thin gulfs and eliminates small holes.

3.4 An Application: Noise Removal

Suppose A is a binary image corrupted by impulse noise-some of the black pixels are white and some of the white pixels are black. An example is given in Figure 3.4. Then A ΘB will remove the single black pixels, but will enlarge the holes. We can fill the holes by dilating twice:

$((A \Theta B) \oplus B) \oplus B$

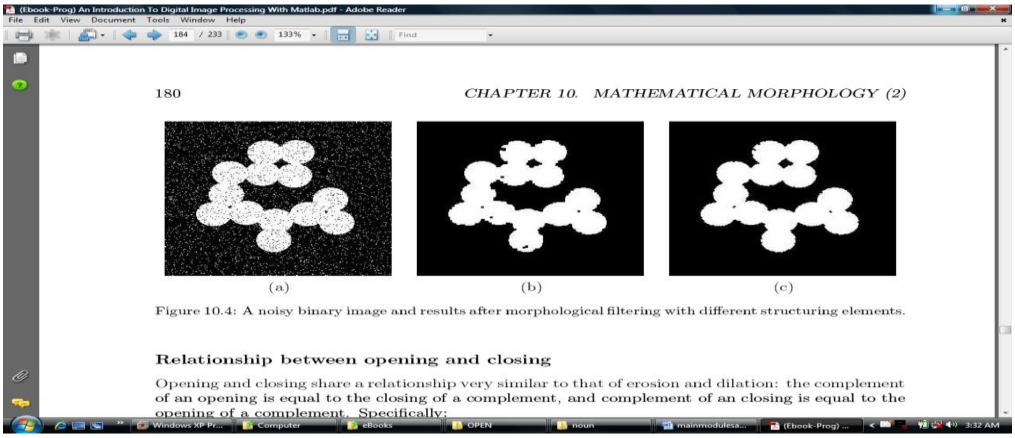
The first dilation returns the holes to their original size; the second dilation removes them. But this will enlarge the object in the image. To reduce them to their correct size, perform a final erosion;

$(((A \Theta B) \oplus B) \oplus B) \Theta B$

The inner two operations constitute an opening; the outer two operations a closing. Thus this noise removal method is in fact an opening followed by a closing.

(A ^{o} B) • B)

This is called morphological filtering


180 CHAPTER 10. MATHEMATICAL MORPHOLOGY (2)

(a) (b) (c)

Figure 10.4: A noisy binary image and results after morphological filtering with different structuring elements.

Relationship between opening and closing

Opening and closing share a relationship very similar to that of erosion and dilation: the complement of an opening is equal to the closing of a complement, and complement of an closing is equal to the opening of a complement. Specifically:

Fig. 3.4: Application of Morphological Operations in Noise Removal

Source McAndrew,A. (2004)

3.5 Relationship between Opening and Closing

Opening and closing share a relationship very similar to that of erosion and dilation; the complement of an opening is equal to the closing of a complement, and complement of a closing is equal to the opening of a complement. Specifically,

$\overline{A \bullet B} = \overline{A} \circ \hat{B}$ and $\overline{A \circ B} = \overline{A} \bullet \hat{B}$

4.0 Conclusion

Morphology is a very important area of mathematics for image processing. This unit has helped to provide a mathematical foundation to validate some concepts discussed in this study material

5.0 Summary

In this unit we have covered some important morphological operators, their relationships and areas of applications in multimedia systems.

6.0 Self-Assessment Exercise

1. Using a relevant example, define the following terms;
a. Dilation
b. Erosion
2. Define an opening operation and state the properties that are usually satisfied by this operation.

7.0 References/Further Reading

McAndrew,A. (2004). *An Introduction to Digital Image Processing with Matlab*. School of Computer Science and Mathematics, Victoria University of Technology.

Serra,J.P. (1982). *Image Analysis and Mathematical Morphology*. Academic Press.

Umbaugh,S.E. (1998). *Computer Vision and Image Processing.* A Practical Approach Using CVIPTools. Prentice-Hall.

Unit 3 Image Restoration, Feature Detection and Pattern Matching

1.0 Introduction

Image restoration is considered one of the major areas of image processing. Image restoration focuses on the removal or reduction of degradations which happened during the acquisition of an image data. The degradations may include noise, which are errors in the pixel values, or optical effects such as out of focus blurring, or blurring due to camera motion. While neighbourhood operations can be used as a dependable technique for image restoration, other techniques require the use of frequency domain processes. In this unit our emphasis shall be on the techniques for dealing with restoration, rather than with the degradations themselves, or the features of digital systems which give rise to image degradation. We shall also consider feature detection and pattern matching.

2.0 Objectives

At the end of this unit, you should be able to:

- describe image degradation model
- explain the concept of noises in image and their removal
- explain the principles of object detection
- describe the technique of object detection.

3.0 Main Content

3.1. A model for Image Degradation

Suppose we have an image $f(x,y)$ and a spatial filter $h(x, y)$ for which convolution with the image results in some form of degradation. Further, let us assume that $h(x,y)$ has a single line of ones, the result of the convolution will be a motion blur in the direction of the line. We represent this by the equation:

$$g(x,y) = f(x,y) * h(x,y)$$

For the resulting degraded image $g(x,y)$ where the symbol $*$ is used to represent a spatial filtering. Further more, we must consider noise, which can be modeled as an additive function to the convolution. Thus if we use $n(x,y)$ to represent the random error which may occur, we will thus have the degraded image expressed as:

$$g(x,y) = f(x,y) * h(x,y) + n(x,y)$$

We can perform the same operations in the frequency domain. To do this we replace the convolution by multiplication, while addition remains as addition, because of the linearity characteristics of the Fourier transform. This resulting expression becomes:

$$G(i,j) = F(i,j)H(i,j) + N(i,j)$$

This expression denotes general image degradation, where of course F, H, and N are the Fourier transformation of f, h, and n respectively.

Once we know the values of H and N we can always recover F by rewriting the above equation as

$$F(i,j) = (G(i,j)-N(i,j))/H(i,j)$$

In real life, this approach may not be as practical as it appears in the mathematical expression. This is because, though, we may have some statistical information about the noise, we may not know the value of n(x,y) or N(i,j) for all, or even any values. Also, when we have the values of H(i,j) which are close to, or equal to zero, we could have some difficulties as implied in the formula.

3.2 Noise

In image digital signal processing systems, the term noise refers to the degradation in the image signal, caused by external disturbance. If an image is being sent electronically from one place to another, via satellite or through networked cable or other forms of channels we may observe some errors at destination points. These errors will appear on the image output in different ways depending on the type of disturbance or distortions in the image acquisition and transmission processed. This gives a clue to what type of errors to expect, and hence the type of noise on the image; hence we can choose the most appropriate method for reducing the effects. Cleaning an image corrupted by noise is thus an important aspect of image restoration.

We shall examine some of the standard noise forms, and provide some details on the different approaches to eliminating or reducing their effects on the image.

3.2.1 Salt and Pepper Noise

This is also referred to as impulse noise, short noise, or binary noise. This degradation can be caused by sharp, sudden disturbances in the image signal; it appears in an image as a randomly scattered white or black (or both) pixels over the image.

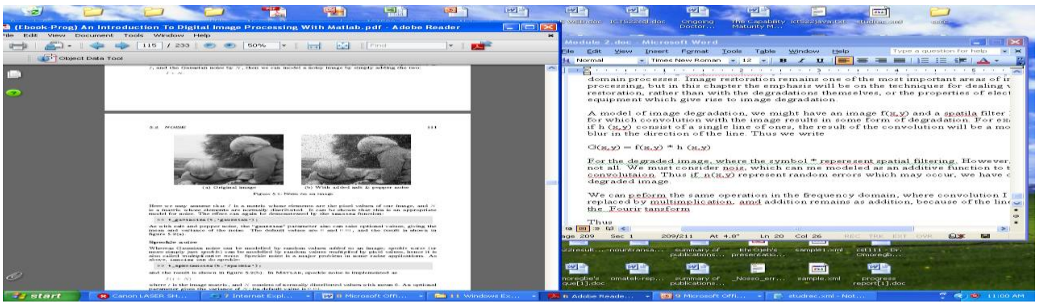


Fig. 3.1a: An Original Image   Fig.3.1b: Effect of Salt and Pepper Noise

Source McAndrew,A. (2004).

3.2.2 Gaussian Noise

Gaussian noise is an ideal case of white noise. It is caused by random fluctuation in the image signal. A very good example of this is by watching a television which is slightly mistuned to a particular channel. Gaussian noise is white noise which is normally distributed. If the image is represented as *I,* and the Gaussian noise by *N*, then we can model a noisy image by simply adding the two represented by *I+ N*

3.2.3 Speckle Noise

As can be seen in the mathematical representation for Gaussian noise, we modeled it by adding random values to an image. On the other hand, speckle noise is modeled by random values multiplied by pixel values; hence it is also called multiplicative noise. This is common with applications that involve radar devices. Figure 3.1 c and Figure 3.1d depict the effects of Gaussian and Speckle noise respectively on an original image shown in Figure 3.1a

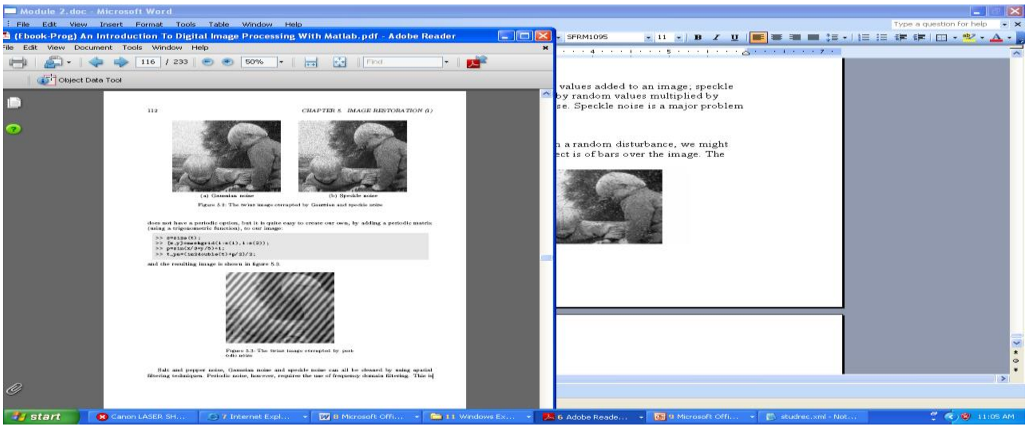


Fig.3.1c: Effect of Gaussian Noise Fig.3.1d: Effect of Speckle Noise

Source McAndrew, A.(2004)

3.2.4 Periodic Noise

This type of noise is used to describe the effect of periodic disturbances on an image signal rather than a random disturbance. The effect appears as bars over an image. This is depicted in figure 3.1 e

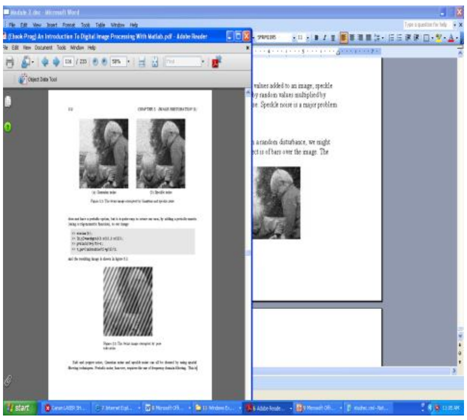


Fig. 3.1e: Effect of Periodic Noise

Source McAndrew,A. (2004)

3.3 Noise Reduction

Now that we have identified the sources of noise in digital signals and some types of noise, we shall describe some of the techniques of reducing or eliminating noise in the image processing. On a general note filters can be used to remove or eliminate noise in an image. The energy of a typical image is primarily in the low frequency region; therefore, a (two-dimensional) low-pass filtering will be good enough in removing a substantial amount of uniform random noise though not without removing some details of the image. On the other hand, the edges that exist in an image usually produce high frequency components. If these components are removed or reduced in energy, the edges will become fuzzier. Median filter are ideal in removing impulse noise while preserving the edges.

They are non-linear filters, however, therefore the process cannot be reversed. In median filtering, a window or mask slides along the image. This window defines a local area around the pixel being processed. The median intensity value of the pixel within that window becomes the new intensity value of the pixel being processed.

Median Filtering
The operations of median filtering make it most suitable for the removal of salt and pepper noise. From the knowledge of simple statistics, you recall that the median of a set is the middle value when they are sorted. If there are even numbers of values, the median is the mean of the middle two. A median filter is an example of a non-linear spatial filter. Using a 3 x 3 mask as an example, the output values is the median of the value in the mask. Let us examine table 3.1

Table 3.1: 3 x 3 Mask

50	65	52
63	255	58
61	60	57

→ 50 52 57 58 I 63 65 255 Here 60 is the median value
60

The operation of obtaining the median means that very large or very small values i.e noisy value will end up at the top or bottom of the sorted list. Thus the median will in generally replace a noisy value with one closer to its surroundings.

3.4 Feature Detection and Recognition (Object Detection Basics)

Object detection is an interesting topic in image processing. It is concerned with locating an object in a scene. To locate an object from a group of other object several questions needs to be answered. Amongst others, the user need to define what is meant by an object? And specify an approach for the object's detection. The ultimate goal of computer vision is to design a system that would be capable of analyzing a scene and determining which items in a scene were relevant objects. We shall now examine Pattern matching by using correlation technique to detect object in a scene.

3.4.1 Pattern Matching Using Correlation

The goal of the pattern matching technique is to find every instance of a specific object in the scene by applying a special template. A template in this context is an image of the object of interest. For example, figure 3.4a is the image of a spherical gas tank which we can use as a template.

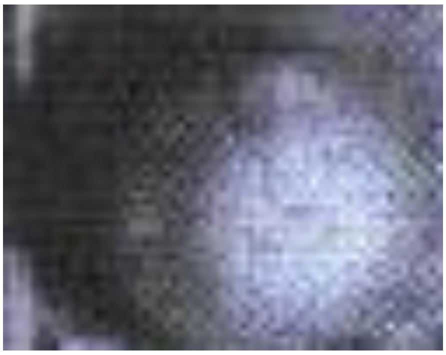
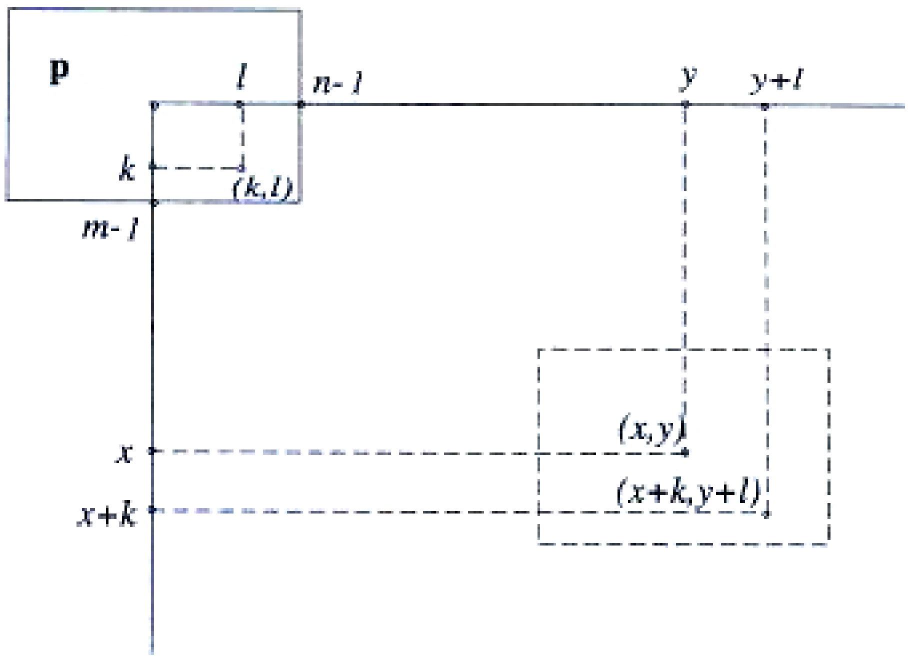


Fig. 3.4a: Spherical Gas

As we can see, this template is a grouping of pixel values that correlate with the object of interest. If our task is to locate all the gas tanks in a factory depicted in figure 3.4b. To accomplish this location task, the gas tank mask is applied to the image in such a way that groupings of pixels that correlate with the template will be close to white while groups of pixels that do not correlate with the template will be close to black

Fig. 3.4b: Spherical Gas

The figure below shows how the template is applied to the image.



The image algebra to accomplish this template application is given by the expression

$$c := a \oplus t$$

In this equation, c is the output image, a is the source image, and t is the template represented by pixel values p. t can be further defined as:

$$t_{(x,y)}(u,v) = \begin{cases} p(u-x, v-y) & \text{if } \; -(m-1) \le u-x \le m-1 \\ & \text{and } \; -(n-1) \le v-y \le n-1 \\ 0 & \text{otherwise.} \end{cases}$$

By applying the above expressions, we obtain a factory picture which is depicted in Figure 3.4c

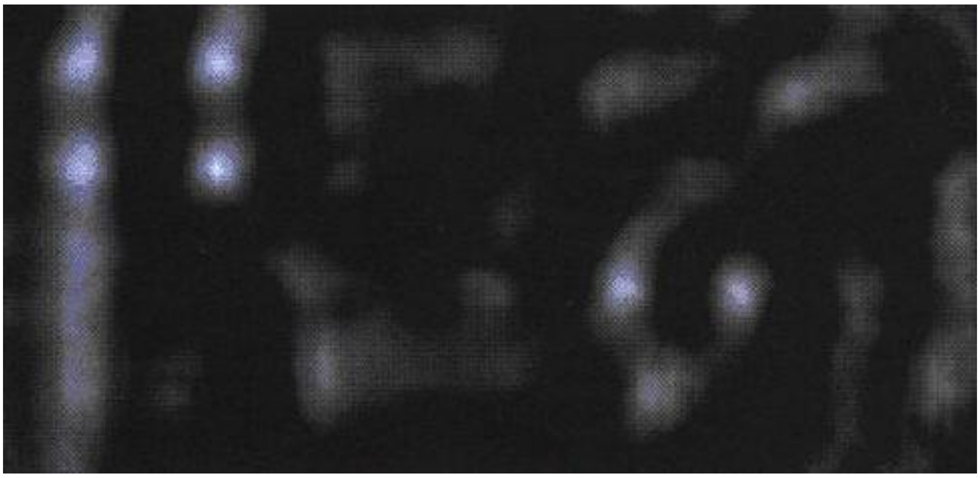


Fig.3.4c: Location of Tanks

As you may observe, in the figure, the locations of the six tanks are in white, which goes to validate the method used here for object detection. However, there are other objects in the scene that are almost detected as gas tanks. While these objects do not have as strong a match as the tanks themselves (as can be seen by the relative whiteness of the pixels at those locations), they could be mistakenly identified as tanks. To make the tank locations clearer, we can apply thresholding on values of the pixel. This type of threshold turns all pixels that are not white enough to black. The result of this operation is shown below.

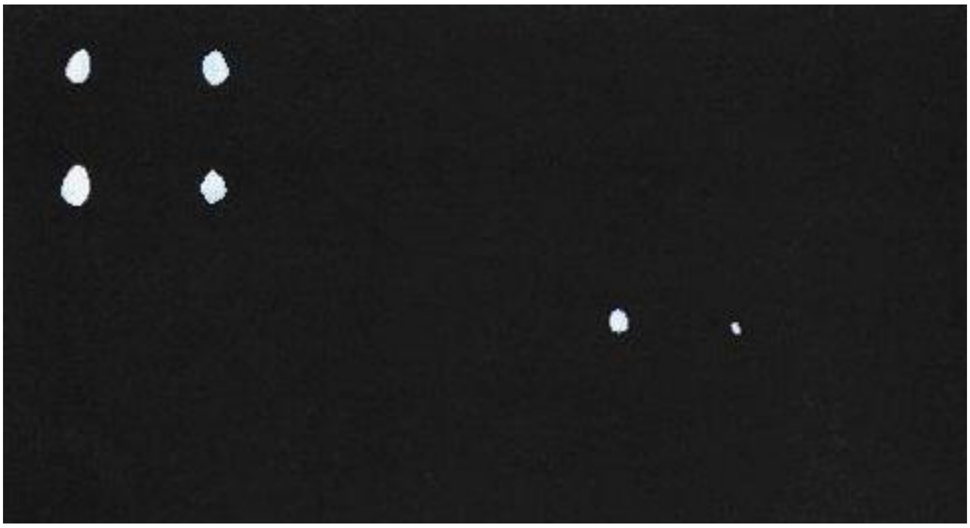


Fig. 3.4d: Object after Thresholding

The locations of the six tanks are now very clearly seen. This same type of process can be performed in the frequency domain.

3.4.2 Some Limitations of Pattern Matching

It requires an accurate image of the desired object as it is likely to appear in the image.

Changes in the object's orientation and size will adversely affect performance.

If the template image was taken in different lighting than the source image, then the pixel values are less likely to line up, even for the object of interest.

4.0 Conclusion

Image restoration focuses on the removal or reduction of degradations which happened during the acquisition or transmission of image data. In digital image processing systems, the term noise refers to the degradation in the image signal, caused by external disturbance. Noise can be removed by designing an appropriate filter.

5.0 Summary

In this unit, we covered image degradation model, identified sources of noise in digital signals and some kinds of noise. We also covered how noise can be removed or reduced by using appropriate filters. Finally we covered pattern matching using correlation.

6.0 Self-Assessment Exercise

1. Explain the term "noise"
2. Briefly explain the terms
   a. Salt and pepper noise
   b. Gaussian noise
   c. Periodic noise
   d. Speckle noise

7.0 References/Further Reading

Benoit,H. (1996). Digital Televison MPEG-1, MPEG-2 and Principles of the DVB System.Oxford:, Focal Press, Linacre House, Jordan Hill.

Gibson,J.D. (Ed.). (2001). Multimedia Communications: Directions and Innovations.USA: Academic Press..

McAndrew,A. (2004). An Introduction to Digital Image Processing with Matlab School of Computer Science and Mathematics, Victoria University of Technology.