`

CHAPTER-1

1. INTRODUCTION TO IMAGE PROCESSING

1.1 DEFINITION OF AN IMAGE:

An image is an array, or a matrix, of square pixels (picture elements) arranged in columns and rows. In a (8-bit) gray scale image each picture element has an assigned intensity that ranges from 0 to 255. A gray scale image is what people normally call a black and white image, but the name emphasizes that such an image will also include many shades of gray.

A digital image is a representation of a two-dimensional image as a finite set of digital values, called picture elements or pixels. Pixel values typically represent gray levels, colors.

Digitization implies that a digital image is an approximation of a real scene.

1.2 IMAGE PROCESSING:

In electrical engineering and computer science, image processing is any form of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image processing techniques involve treating the image as a two-dimensional signal and applying standard signal processing techniques to it. Image processing usually refers to digital image processing, but optical and analog image processing also are possible.

1.3 DIGITAL IMAGE PROCESSING:

It is the use of computer algorithms to perform image processing on digital images. As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise during processing. Since images are defined over two dimensions (perhaps more) digital image processing may be modeled in the form of multidimensional systems.

Digital image processing focuses on two major tasks

– Improvement of pictorial information for human interpretation.

– Processing of image data for storage, transmission and representation for autonomous machine perception.

1.4 Applications of Digital Image Processing:

- Medical applications
- Image enhancement/restoration
- Image transmission and coding
- Color processing
- Remote sensing
- Robot Vision

1. ***Remote Sensing*** – For this application, sensors capture the pictures of the earth's surface in remote sensing satellites or multi – spectral scanner which is mounted on an aircraft. These pictures are processed by transmitting it to the Earth station. Techniques used to interpret the objects and regions are used in flood control, city planning, resource mobilization, agricultural production monitoring, etc.

2. ***Moving object tracking*** – This application enables to measure motion parameters and acquire visual record of the moving object. The different types of approach to track an object are:

a. Motion based tracking
b. Recognition based tracking

3. ***Defense surveillance*** – Aerial surveillance methods are used to continuously keep an eye on the land and oceans. This application is also used to locate the types and formation of naval vessels of the ocean surface. The important duty is to divide the various objects presenting the water body part of the image. The different parameters such as length, breadth, area, perimeter, compactness are set up to classify each of divided objects. It is important to recognize the distribution of these objects in different directions that are east, west, north, south, northeast, northwest, southeast and south west to explain all possible formations of the vessels. We can interpret the entire oceanic scenario from the spatial distribution of these objects.

`

4. *Biomedical Imaging techniques* – For medical diagnosis, different types of imaging tools such as X- ray, Ultrasound, computer aided tomography (CT) etc are used.

Some of the applications of Biomedical imaging applications are as follows: Heart disease identification– The important diagnostic features such as size of the heart and its shape are required to know in order to classify the heart diseases. To improve the diagnosis of heart diseases, image analysis techniques are employed to radiographic image.

1.5 FUNDAMENTAL STEPS IN DIGITAL IMAGE PROCESSING:

There are some fundamental steps in Digital Image Processing. The fundamental steps are described below with a neat diagram.

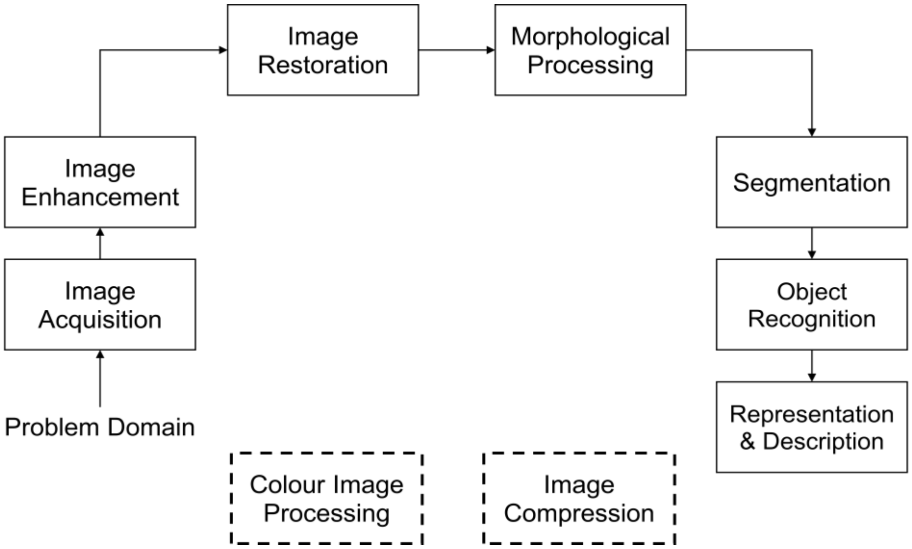


Fig. 1.1 Key Stages in Digital Image Processing

1.5.1. Image Acquisition:

Images are typically generated by illuminating a scene and absorbing the energy reflected by the objects in that scene. Image Acquisition involves an imaging system which consists of an imaging sensor. This sensor collects all the light rays i.e., reflected or absorbed (refracted). This sensor should be capable of giving digital output. So Image Acquisition consists of SAMPLING and QUANTISATION to get the digital output.

`

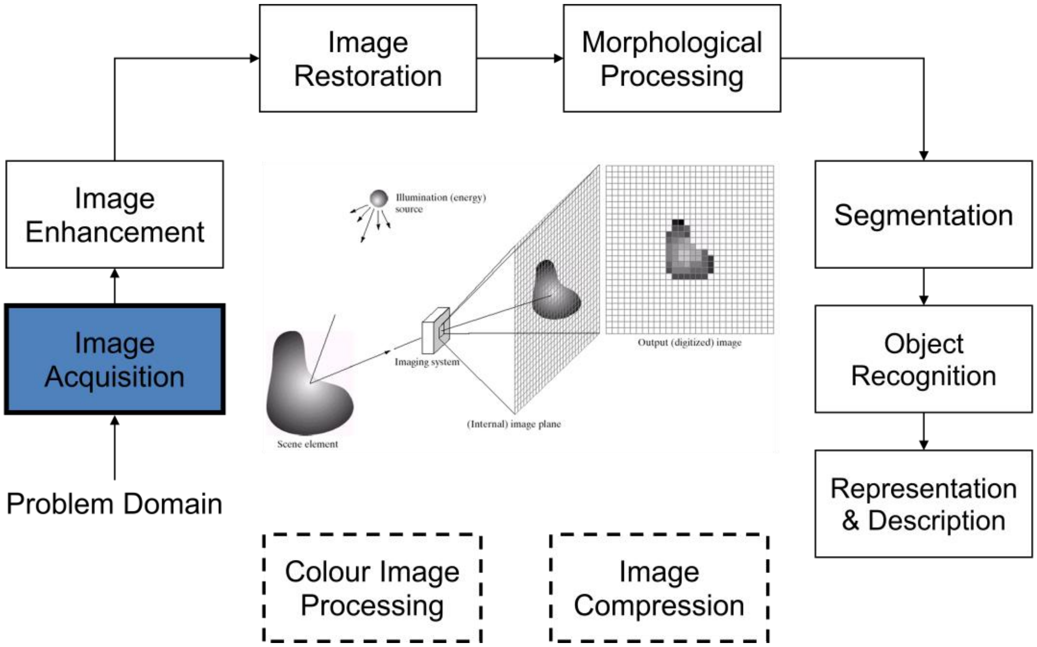


Fig. 1.2 Key Stages in Digital Image Processing: Image Acquisition

1.5.2. Image Enhancement:

It is a subjective type of methodology. Image enhancement is the improvement of digital image quality without knowledge about the source of degradation.

The aim of image enhancement is to improve the interpretability or perception of information in images for human viewers, or to provide `better' input for other automated image processing techniques.

`

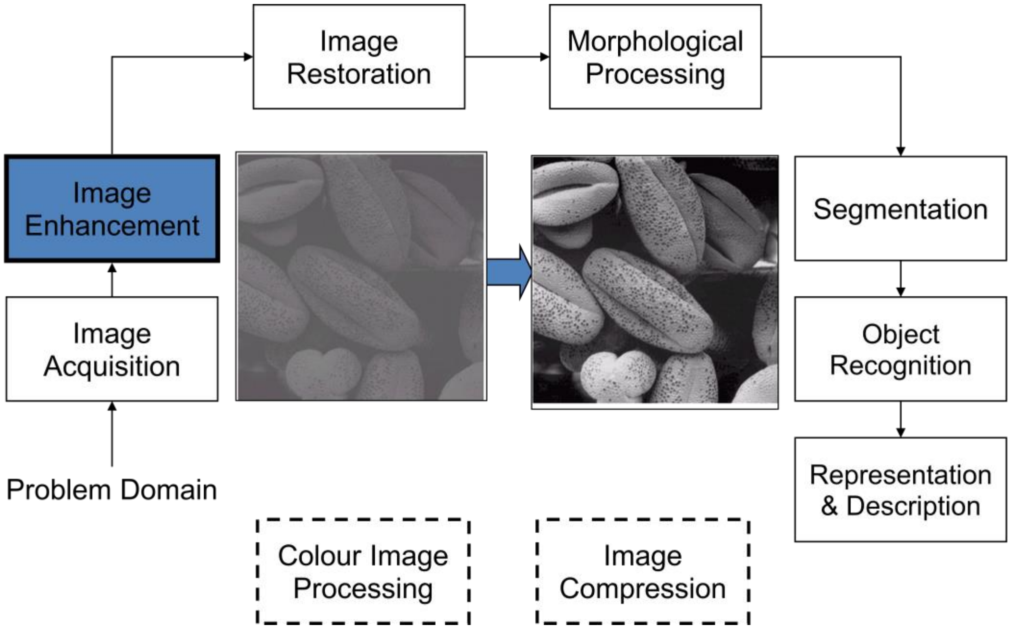


Fig. 1.3 Key Stages in Digital Image Processing: Image Enhancement

1.5.3 Image Restoration:

It is an objective type of methodology. Images are produced to record or display useful information. Due to imperfections in the imaging and capturing process, however, the recorded image invariably represents a degraded version of the original scene. The undoing of these imperfections is crucial to many of the subsequent image processing tasks. There exists a wide range of different degradations that need to be taken into account, covering for instance noise, geometrical degradations, illumination and color imperfections (under/over-exposure, saturation), and blur.

In addition to blurring effects, noise always corrupts any recorded image. Noise may be introduced by the medium through which the image is created (random absorption or scatter effects), by the recording medium (sensor noise), by measurement errors due to the limited accuracy of the recording system, and by quantization of the data for digital storage.

The field of image restoration (sometimes referred to as image de-blurring or image de-convolution is concerned with the reconstruction or estimation of the uncorrupted image from a blurred and noisy one. Essentially, it tries to perform an operation on the image that is the inverse of the imperfections in the image formation system. In the use of image restoration methods, the characteristics of the degrading system and the noise are assumed to be known a priori.

The main objective of restoration is to improve the quality of a digital image which has been degraded due to various phenomena like:

- Motion
- Improper focusing of Camera during image acquisition.
- Atmospheric turbulence
- Noise

Noise is usually dealt separately, while the other phenomena are modeled by degradation functions.

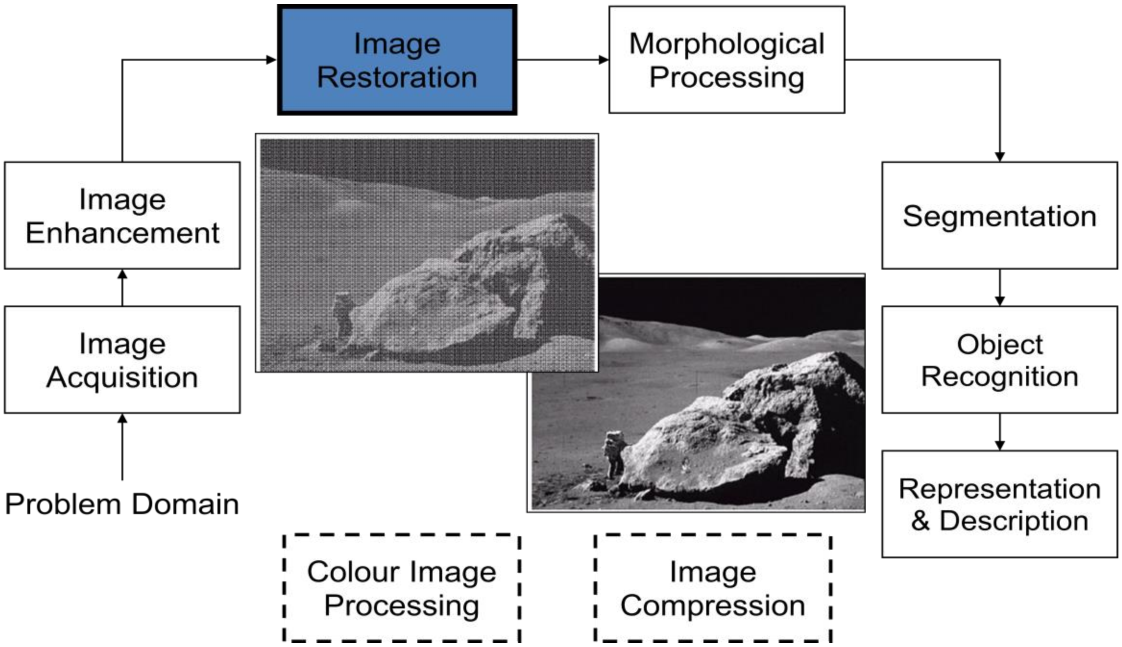


Fig. 1.4 Key Stages in Digital Image Processing: Image Restoration

1.5.4 Morphological Processing:

Morphological image processing techniques are useful for extracting image components that are useful in representing and describing region shapes. The filters can be described using set theoretic notation and implemented using simple computer algorithms

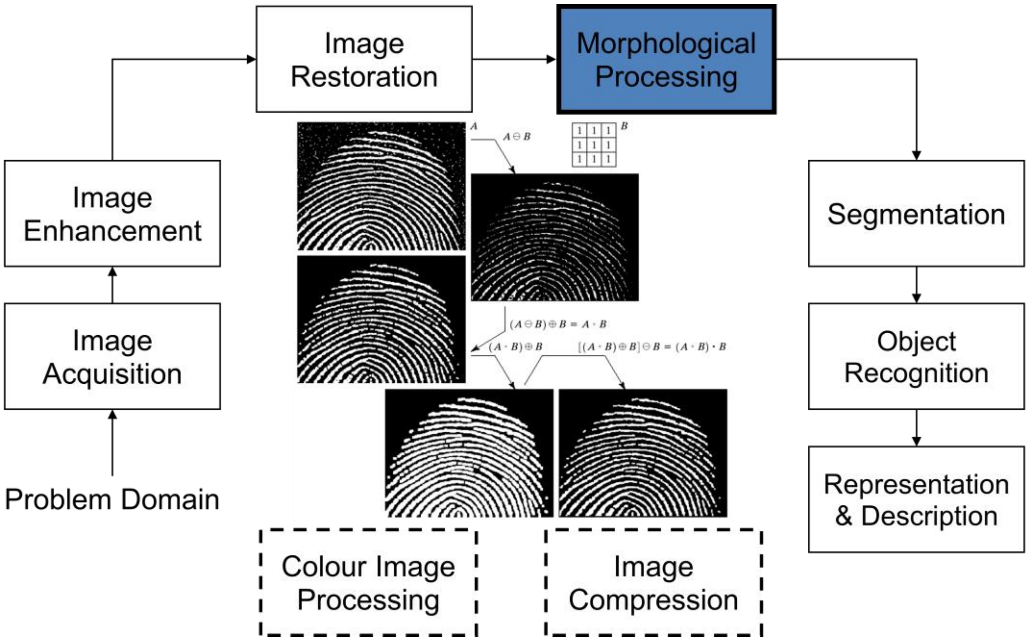


Fig. 1.5 Key Stages in Digital Image Processing: Morphological Processing

1.5.5 Image Segmentation:

Segmentation is to distinguish objects from background. For intensity images (i.e., those represented by point-wise intensity levels) four popular approaches are: Threshold techniques, Edge-based methods, Region-based techniques, and Connectivity-preserving relaxation methods.

Threshold techniques, which make decisions based on local pixel information, are effective when the intensity levels of the objects fall squarely outside the range of levels in the background. Edge-based methods center around contour detection: their weakness in connecting together broken contour lines make them, too prone to failure in the presence of blurring.

A region-based method usually proceeds as follows: the image is partitioned into connected regions by grouping neighboring pixels of similar intensity levels. Adjacent regions are then merged under some criterion involving perhaps homogeneity or sharpness of region boundaries. Over stringent criteria create fragmentation; lenient ones overlook blurred boundaries and over merge.

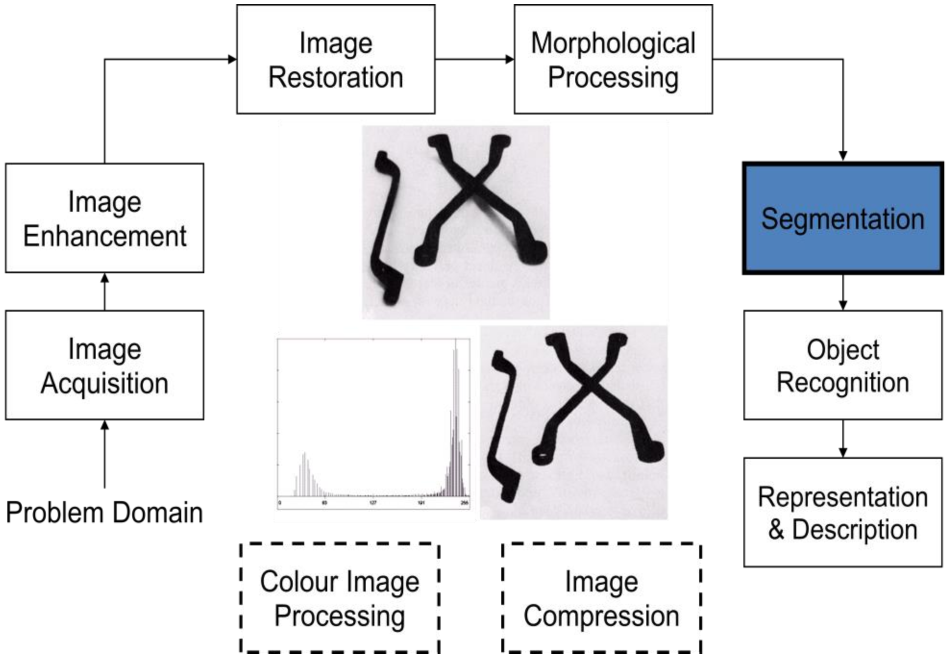


Fig. 1.6 Key Stages in Digital Image Processing: Segmentation

1.5.6 Object Recognition:

Object recognition in computer vision is the task of finding a given object in an image or video sequence. Humans recognize a multitude of objects in images with little effort, despite the fact that the image of the objects may vary somewhat in different viewpoints, in many different sizes/scale or even when they are translated or rotated. Objects can even be recognized when they are partially obstructed from view.

Object recognition is to determine which, if any, of a given set of objects appear in a given image or image sequence. Thus object recognition is a problem of matching models from a database with representations of those models extracted from the image luminance data. Early work involved the extraction of three-dimensional models from stereo data, but more recent work has concentrated on recognizing objects from geometric invariants extracted from the two-dimensional luminance data.

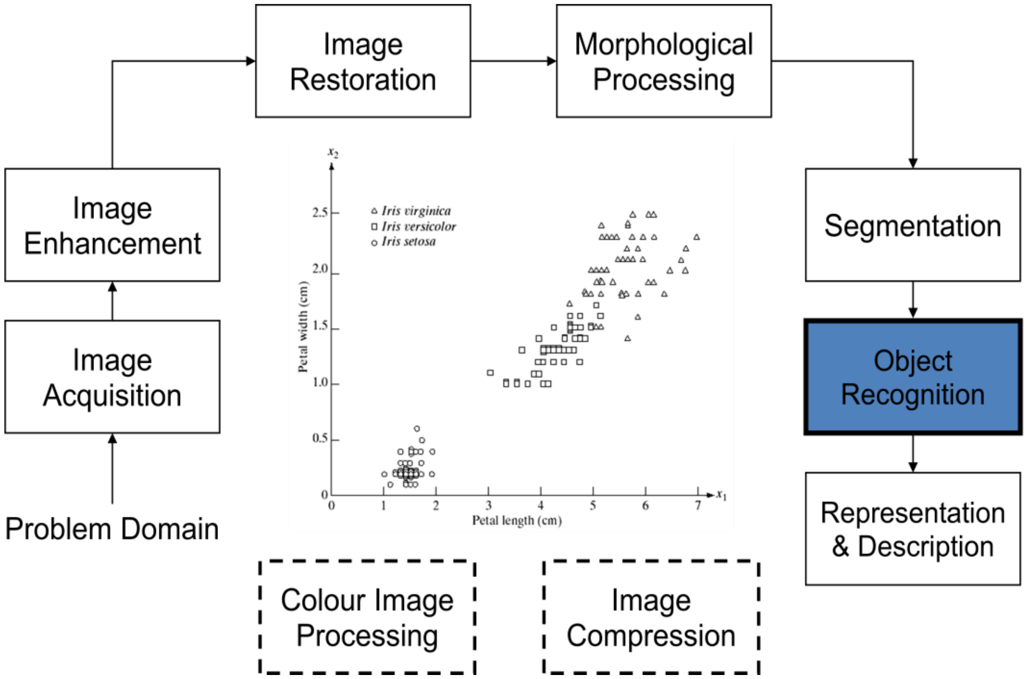


Fig. 1.7 Key Stages in Digital Image Processing: Object Recognition

1.5.7 Representation and Description:

It is a process which transforms raw data into a form suitable for subsequent computer processing. The first decision is to be choosing between boundary representation and regional representation and regional representation. Boundary representation is used when the details of external shape characteristics is important, whereas the regional representation is used when the regional representation is used when the internal properties are important.

The next decision is the feature selection, which deals with extracting features that result in some quantitative information of or features that result in some quantitative information of interest or features that are basic for representing one class of objects from another.

`

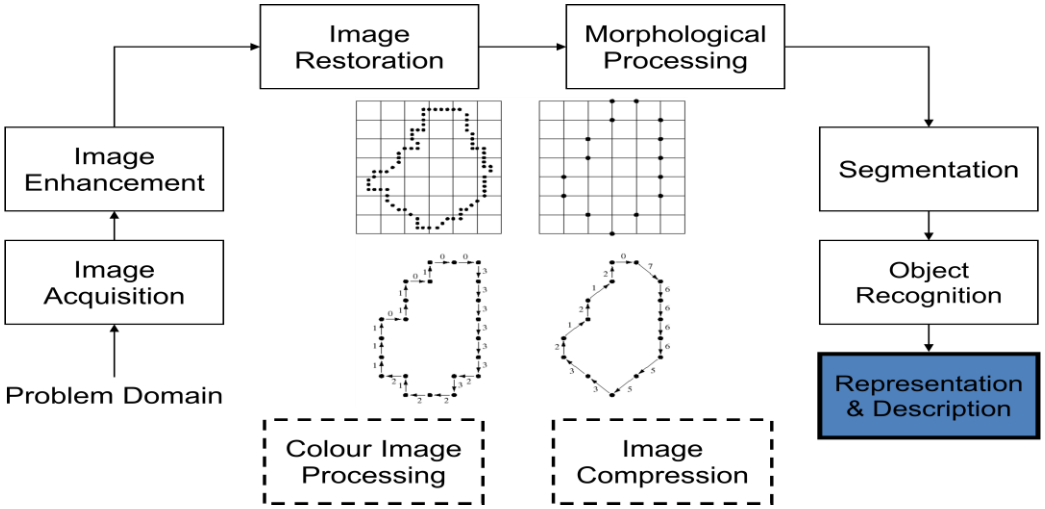
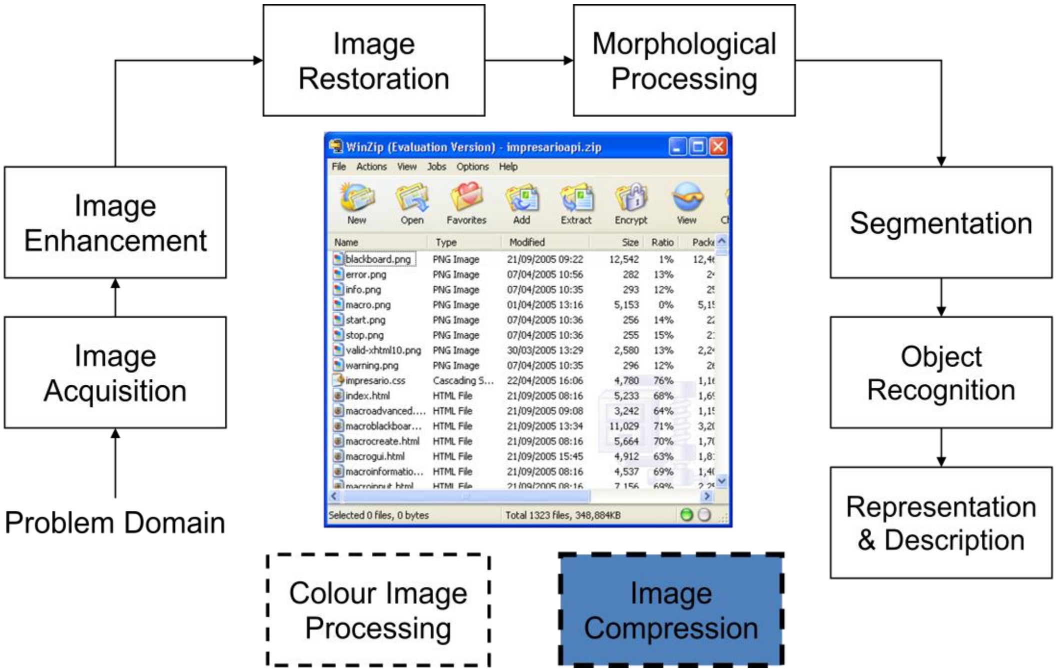


Fig. 1.8 Key Stages in Digital Image Processing: Representation & Description

1.5.8 Image Compression:

The objective of image compression is to reduce irrelevance and redundancy of the image data in order to be able to store or transmit data in an efficient form.

Image compression may be lossy or lossless. Lossless compression is preferred for archival purposes and often for medical imaging, technical drawings, clip art, or comics. This is because lossy compression methods, especially when used at low bit rates, introduce compression artifacts. Lossy methods are especially suitable for natural images such as photographs in applications where minor (sometimes imperceptible) loss of fidelity is acceptable to achieve a substantial reduction in bit rate. The lossy compression that produces imperceptible differences may be called visually lossless.



Name	Type	Modified	Size	Ratio	Pack
blackboard.png	PNG Image	21/09/2005 09:22	12,542	1%	12,4
error.png	PNG Image	07/04/2005 10:56	282	13%	2
info.png	PNG Image	07/04/2005 10:35	293	12%	2
macro.png	PNG Image	01/04/2005 13:16	5,153	0%	5,1
start.png	PNG Image	07/04/2005 10:36	256	14%	2
stop.png	PNG Image	07/04/2005 10:36	255	15%	2
valid-xhtml10.png	PNG Image	30/03/2005 13:29	2,580	13%	2,2
warning.png	PNG Image	07/04/2005 10:35	296	12%	2
impresario.css	Cascading S...	22/04/2005 16:06	4,780	76%	1,1
index.html	HTML File	21/09/2005 08:16	5,233	68%	1,6
macroadvanced....	HTML File	21/09/2005 09:08	3,242	64%	1,1
macroblackboar...	HTML File	21/09/2005 13:34	11,029	71%	3,2
macrocreate.html	HTML File	21/09/2005 08:16	5,664	70%	1,7
macrogui.html	HTML File	21/09/2005 15:45	4,912	63%	1,8
macroinformatio...	HTML File	21/09/2005 08:16	4,537	69%	1,4
macroinput.html	HTML File	21/09/2005 08:16	7,156	69%	2,2

Selected 0 files, 0 bytes

Total 1323 files, 348,884KB

Fig. 1.9 Key Stages in Digital Image Processing: Image Compression

1.5.9 Colour Image Processing:

The human visual system can distinguish hundreds of thousands of different color shades and intensities, but only around 100 shades of gray. Therefore, in an image, a great deal of extra information may be contained in the color, and this extra information can then be used to simplify image analysis, e.g. object identification and extraction based on color.

Three independent quantities are used to describe any particular color. The hue is determined by the dominant wavelength. Visible colors occur between about 400nm (violet) and 700nm (red) on the electromagnetic spectrum.

The saturation is determined by the excitation purity, and depends on the amount of white light mixed with the hue. A pure hue is fully saturated, i.e. no white light mixed in. Hue and saturation together determine the chromaticity for a given color. Finally, the intensity is determined by the actual amount of light, with lighter corresponding to more intense colors.

`

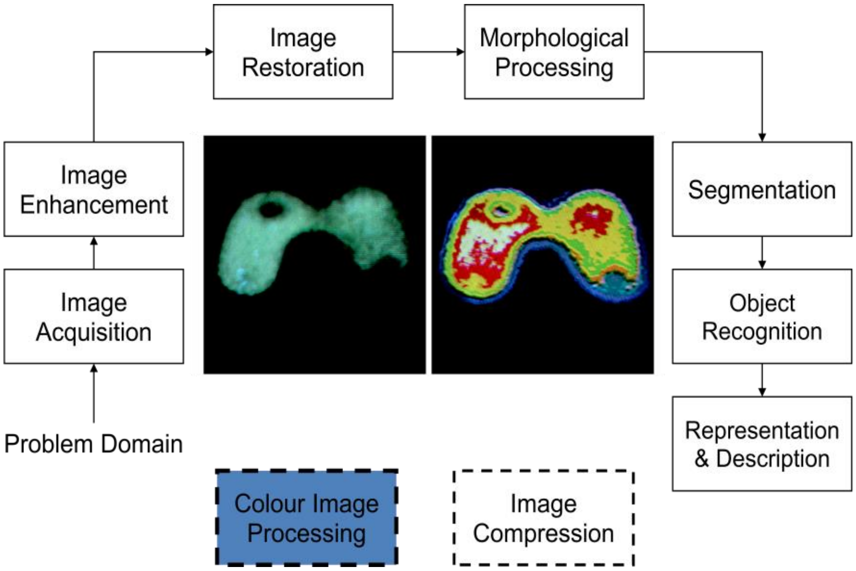


Fig. 1.10 Key Stages in Digital Image Processing: Colour Image Processing

Of all the fundamental steps of image processing, image restoration is the key step which gives most reliable results. True images are usually degraded during image acquisition. Image restoration is for restoring true images from their observed but degraded versions; itis often used for preprocessing observed images so that subsequent imageprocessing and analysis becomes more reliable.

1.6 NOISE:

The principal source of noise in digital images arises during image transmission.

For instance, an image transmitted over a wireless network might be corrupted as a result of lighting or other atmospheric disturbance.

Noise: Image noise is random (not present in the object imaged) variation of brightness or color information in images, and is usually an aspect of electronic noise.

1.7 Some Important Noise Probability Density Functions:

Due to limitations of computational and storage capacity, it is practically always impossible to take into account all the available knowledge about minor details which possibly have some impact on the observation. It is then reasonable to ignore the details and model only their net effect, which manifests itself in minor fluctuations not predictable from the things included in the model. These fluctuations are called noise.

a) *Gaussian (normal) noise:*

Gaussian noise is statistical noise that has its probability density function equal to that of the normal distribution, which is also known as the Gaussian distribution In other words, the values that the noise can take on are Gaussian-distributed. A special case is white Gaussian noise, in which the values at any pairs of times are statistically independent (and uncorrelated). In applications, Gaussian noise is most commonly used as additive white noise to yield additive white Gaussian noise.

The probability density function is given by equation

$$\mathrm{P(z)} = \frac{1}{\sqrt{2\pi\sigma}} \mathrm{e}^{-(\mathrm{z}-\bar{\mathrm{z}})^2/2\sigma^2} \qquad (1.1)$$

Where

z= intensity

$\bar{z}$ = mean of z

σ= standard deviation of z

σ^2= variance of z

- Frequently used in practice since it is mathematically tractable in both the spatial and frequency domains
- 70% of z's values fall into the range.$[(\bar{z} - \sigma), (\bar{z} + \sigma)]$.
- 95% of z's values fall into the range.$[(\bar{z} - 2\sigma), (\bar{z} + 2\sigma)]$.
- Arising in an image due to factors such as electronic circuit noise and sensor noise due to poor illumination and/or high temperature.

`

(b) ***Rayleigh noise:***

The probability density function is given by equation

$$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-\frac{(z-a)^2}{b}} & \text{for } z \geq a, \bar{z} = a + \sqrt{\frac{\pi b}{4}} \\ 0 & \text{for } z < a\,, \sigma^2 = \frac{b(4-\pi)}{4} \end{cases} \tag{1.2}$$

(c) ***Erlang (Gamma) noise:***

The probability density function is given by equation

$$p(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{for } z \leq 0 \;\; \bar{z} = \frac{b}{a}, a \in R^+, b \in Z^+ \\ \qquad 0 & \text{for } z < 0 \quad \sigma^2 = \frac{b}{a^2} \end{cases} \tag{1.3}$$

- Developed by Erlang to model telephone traffics.
- It is called Gamma noise if the denominator is the gamma function.

(d) ***Exponential noise:***

The probability density function is given by equation

$$p(z) = \begin{cases} a\, e^{-az} & \text{for } z \geq 0 \quad \bar{z} = \frac{1}{a} \\ 0 & \text{for } z < 0 \quad \sigma^2 = \frac{1}{a^2} \end{cases} \tag{1.4}$$

- A special case of the Erlang density, with b=1.

(e) ***Uniform noise:***

The probability density function is given by equation

$$p(z) = \begin{cases} \frac{1}{b-a} \text{ if } a \leq z \leq b & \bar{z} = \frac{a+b}{2} \\ 0 \quad \text{otherwise} & \sigma^2 = \frac{(b-a)^2}{12} \end{cases} \tag{1.5}$$

(f) ***Impulse (salt-and-pepper) noise:***

The probability density function is given by equation

$$p(z) = \begin{cases} p_a & \text{for } z = a \\ p_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases} \tag{1.6}$$

- Bipolar if neither P_a nor P_b is zero; in practice, for an 8-bit image, b=255 (white) and a = 0 (black).
- Bipolar one, also known as salt-and-pepper, data-drop-out and spike noise.
- It is called unipolar if either P_a or P_b is zero.
- It is caused by either sensors failure (**pepper**, *black*) or sensors saturation in color (**salt**, *white*).

Estimation of Noise Parameters:

- The parameters of periodic noise typically estimated by inspecting the image’s Fourier spectrum.
- The parameters of noise PDFs may be known partially from sensor specifications often necessary to be estimated for a particular imaging arrangement.
- Capturing a set of images of “flat” environments.
- Possible to be estimated from small patches of reasonably constant background intensity, when only images already generated by a sensor are available.

1.8 NOISE FILTERING:

Noise filtering is used to reduce the noise components on an image.

1.8.1 Spatial Filtering:

Spatial filtering is suitable when only additive random noise is present. Filtering is a technique for modifying or enhancing an image. Spatial domain operation or filtering (the processed value for the current pixel processed value for the current pixel depends on both itself pixel neighborhood of the corresponding input pixel. A pixel's neighborhood is some set of the surrounding pixels. Hence Filtering is a neighborhood operation, in which the value of any given pixel in the output image is determined by applying some algorithm to the values of the pixels in pixels, defined by their locations relative

`

The following spatial filters will be employed in spatial filtering

- Arithmetic mean filter
- Geometric mean filter
- Harmonic mean filter
- Contra harmonic mean filter

Mean Filter:

The mean filter is a simple spatial filter .It is a sliding-window filter that replaces the center value in the window. It replaces with the average mean of all the pixel values in the kernel or window. The window is usually square but it can be of any shape

An Example of mean filtering of a 3x3 kernel of values is shown below.

In this Center value which is previously 1 in the unfiltered value is replaced by the mean of all nine values that is 5.

Mean Filter

$$\begin{matrix}8&4&7\\2&1&9\\5&3&6\end{matrix} \quad \longrightarrow \quad \begin{matrix}*&*&*\\ *&5&*\\ *&*&*\end{matrix}$$

Since 8+4++2+1+9++573+6=45 45 / 9 = 5

(a) Arithmetic mean filter:

The filter equation is given by

$$R(x,y) = \frac{1}{MN}\sum_{(s,t)\in S_{xy}} g(s,t) \tag{1.7}$$

- The simplest mean filters representing the restored pixel value at (x,y) by the arithmetic mean computed within the filter window.
- Smoothing local variations in an image → blurring.
- Noise-reducing as a by-product of blurring.

(b) Geometric mean filter:

The filter equation is given by

$$R(x,y) = \left[\prod_{(s,t)\in S_{xy}} g(s,t)\right]^{\frac{1}{MN}} \tag{1.8}$$

- Each restored pixel value given by the product of all the pixel values in the filter window, raised to the power$\frac{1}{mn}$.
- Achieves smoothing comparable to the arithmetic mean filter, but tends to lose less image detail in the process.

(c) ***Harmonic mean filter*****:**

The filter equation is given by

$$R(x, y) = \frac{MN}{\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}} \tag{1.9}$$

- Working well for salt and Gaussian noises but failing for pepper noise.

(d***) Contra harmonic mean filter:***

The filter equation is given by

$$R(x, y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}} \tag{1.10}$$

- This filter well handles or virtually eliminates the effects of salt-and-pepper noise.
- However, this filter is unable to eliminate both salt and pepper noises simultaneously.
- Eliminating pepper noise when $Q\in R+$.
- Eliminating salt noise when $Q\in R$.

1.8.2 Order-Statistic Filters:

Order-statistic filters (OSF) are those whose response is based on ordering (ranking) the values of the pixels contained in the filter window.

Some of the order-static filters are as follows:

- Median filter
- Max and min filters
- Midpoint filter
- Alpha-trimmed mean filter

(a) Median filter:

Median Filter is a simple and powerful non-linear filter which is based order statistics. It is easy to implement method of smoothing images. Median filter is used for reducing the amount of intensity variation between one pixel and the other pixel. In this filter, we do not replace the pixel value of image with the mean of all neighboring pixel values, we replaces it with the median value. Then the median is calculated by first sorting all the pixel values into ascending order and then replace the pixel being calculated with the middle pixel value. If the neighboring pixel of image which is to be considered containsan even numbers of pixels, then the average of the two middle pixelsused toreplacevalues is. The median filter gives best result when the impulse noise percentage is less than 0.1 %. When the quantity of impulse noise is increased the median filter not gives best results. The filter equation is given by

$$R(x, y) = \mathrm{median}_{(s,t)\in S_{xy}}\{g(s, t)\} \qquad (1.11)$$

Example:

10	5	20
14	80	11
8	3	22

Median value: 3 5 8 10 11 14 20 22 8

Central value 80 is replaced by 11

- Median filter is the best-known of order-statistic filters.
- It represents the restored pixel value at (x,y) by the median (ranked in the 50th percentile) of intensity levels in the filter window.
- For certain types of noise, this filter provides excellent noise-reduction capabilities, with considerably less blurring than linear smoothing filters on the same basis.
- This filter is particularly effective in the presence of both bipolar and unipolar impulse noise.

`

(b) Max and Min filter:

Maximum Filter is also known as dilation filter .This filter replaces every pixel by the maximum value in the neighborhood pixels.

Minimum Filter is also known as erosion filter. This filter replaces every pixel by the minimum value in the neighborhood pixels.

The filter equation is given by

$$R(x, y) = \begin{cases} \text{Max}_{(s,t) \in S_{xy}} \{g(s, t)\} \text{ for the Max Filter} \\ \text{Min}_{(s,t) \in S_{xy}} \{g(s, t)\} \text{ for the Min Filter} \end{cases} \tag{1.12}$$

- This filter represents the restored pixel value at (x, y) by the maximum/minimum of intensity levels in the filter window.
- The max filter greatly reduces pepper noise (black dots).
- The min filter greatly reduces salt noise (white dots).

(c) *Midpoint filters:*

The filter equation is given by

$$R(x, y) = \frac{1}{2} [\text{Max}_{(s,t) \in S_{xy}} \{g(s, t)\} + \text{Min}_{(s,t) \in S_{xy}} \{g(s, t)\}] \tag{1.13}$$

- This filter represents the restored pixel value at (x,y) by the midpoint between the darkest and brightest points in the filter window. This filter works best for randomly distributed noise, e.g., Gaussian or uniform noise.

(d) *Alpha trimmed mean filter:*

The filter equation is given by

$$R(x, y) = \frac{1}{MN - d} \sum_{(s,t) \in S_{xy}} g_r(s, t) \tag{1.14}$$

Where'd' value ranges from 0 to MN-1.

- A filter formed by averaging the pixels that are remained after trimming is called alpha trimmed mean filter.

`

CHAPTER-2

2. EXISTING SYSTEMS

2.1 MEDIAN FILTER:

The Median filter is a nonlinear digital filtering technique, often used to remove impulse noise. Median filtering is one kind of smoothing technique. All smoothing techniques are effective at removing noise in smooth patches or smooth regions of a signal, but adversely affect edges. Often though, at the same time as reducing the noise in a signal, it is important to preserve the edges. Edges are of critical importance to the visual appearance of images, for example. For small to moderate levels of noise, the median filter is better at removing noise while preserving edges of a restored image.

However, its performance is not that much better than for high noise density, whereas, for speckle noise and salt and pepper noise (impulsive noise), it is particularly effective. Hence, median filtering is used for removal of low density noise in digital image processing. Typically, by far the majority of the computational effort and time is spent on calculating the median of each window. Because the filter must process every entry in the signal, for large signals such as images, the efficiency of this median calculation is a critical factor in determining how fast the algorithm can run.

Consider Peppers image corrupted by 10% and 60% noise. When this image is passed through the Median filter, a more amount of the noise remains in the filtered image. This can be illustrated by the following figures.

`

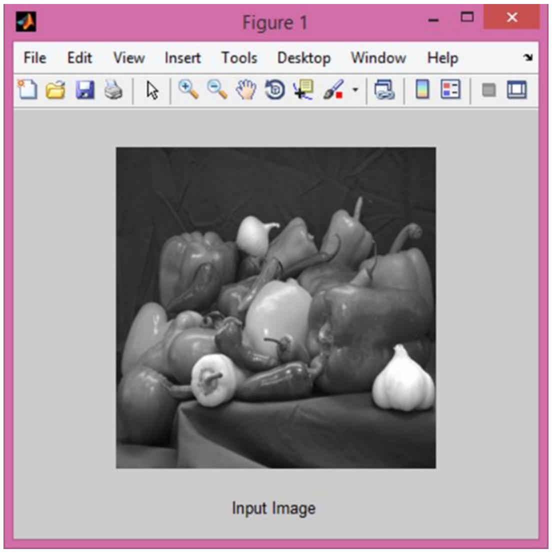
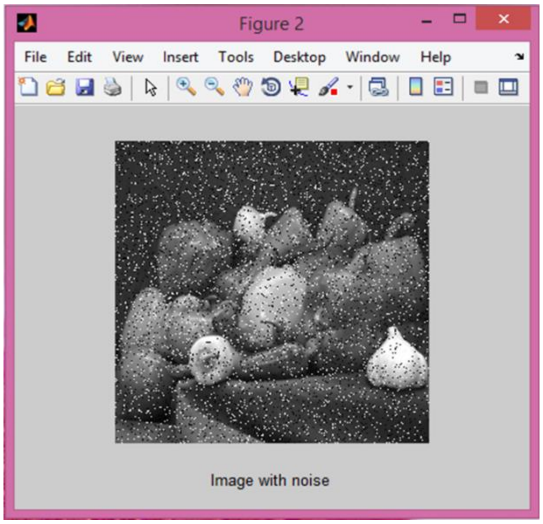


Fig. 2.1(a) Original Image



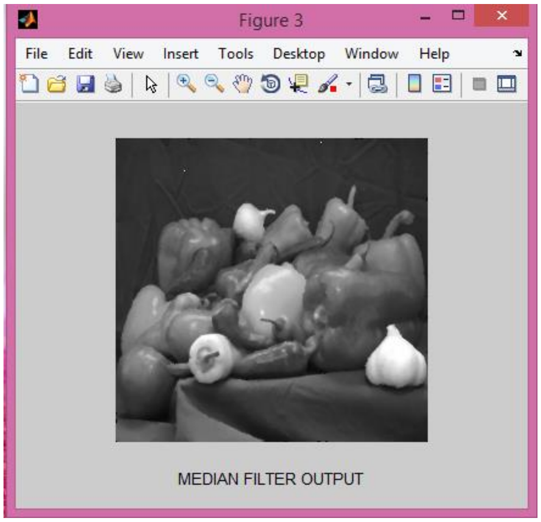


Fig. 2.1 (b) Image Corrupted by 10% Noise

Fig.2.1 (c) Median filter output Image

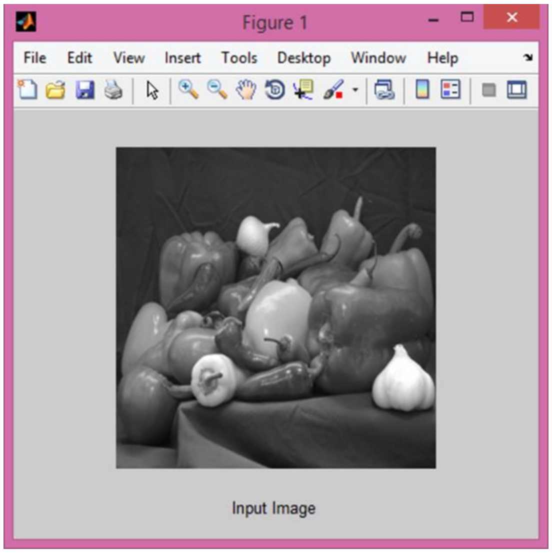
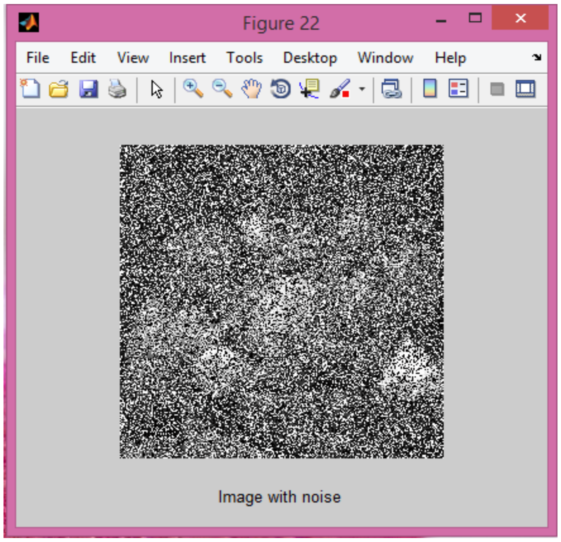


Figure 2.2(a) original image



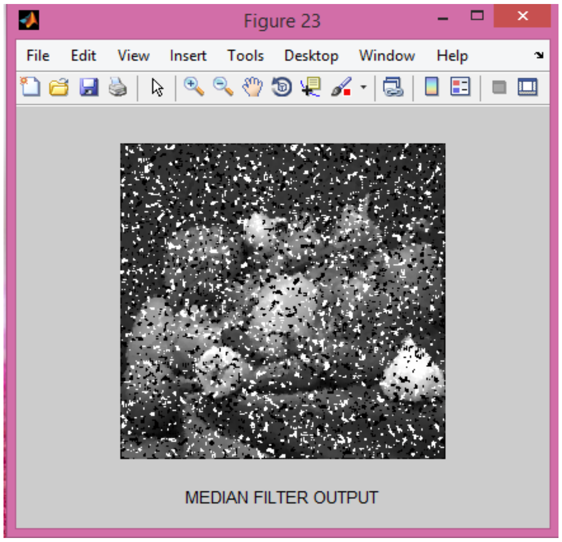


Fig. 2.2(b) image corrupted by 60% noise,

Fig. 2.2(c) median filter output image

2.2 WIENER FILTER:

Wiener uses a pixel wise adaptive Wiener method based on statistics estimated from a local neighborhood of each pixel. Consider Peppers image corrupted by 10%and 60% noise. When this image is passed through the Wiener filter, a more amount of the noise remains in the filtered image. This can be illustrated by the following figures.

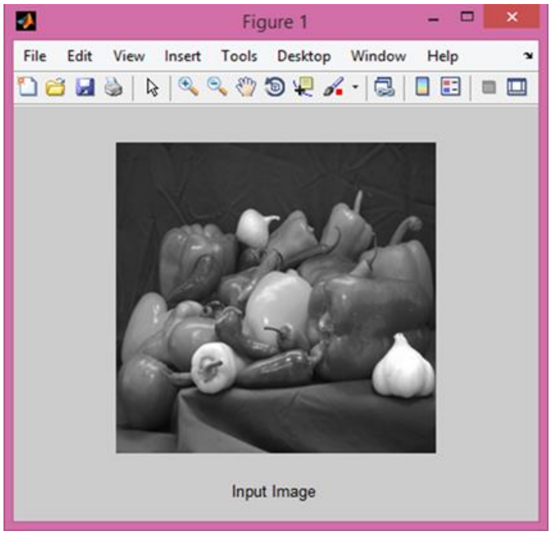


Fig. 2.3 (a) Original Image

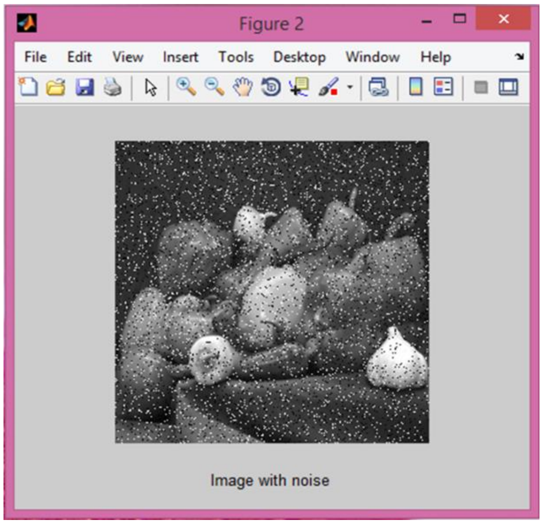


Fig. 2.3 (b) Image Corrupted by 10% Noise

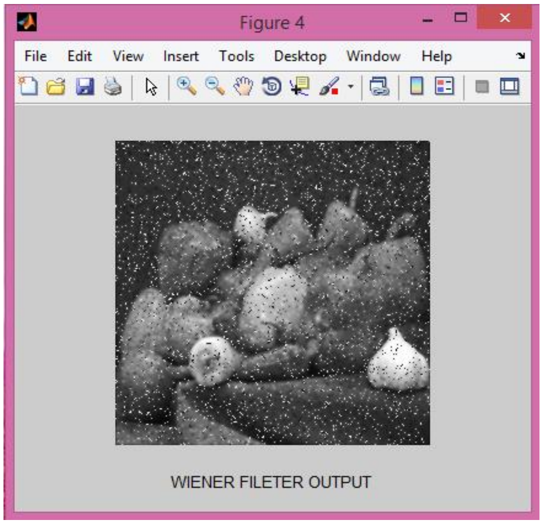


Fig.2.3(c) Wiener filter output Image

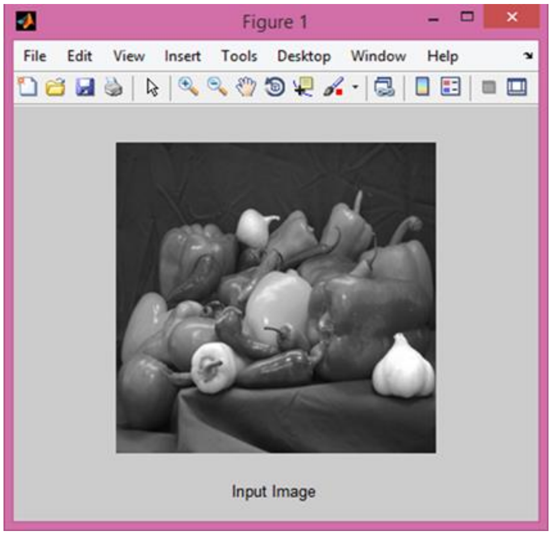


Fig.2.4 (a) original image

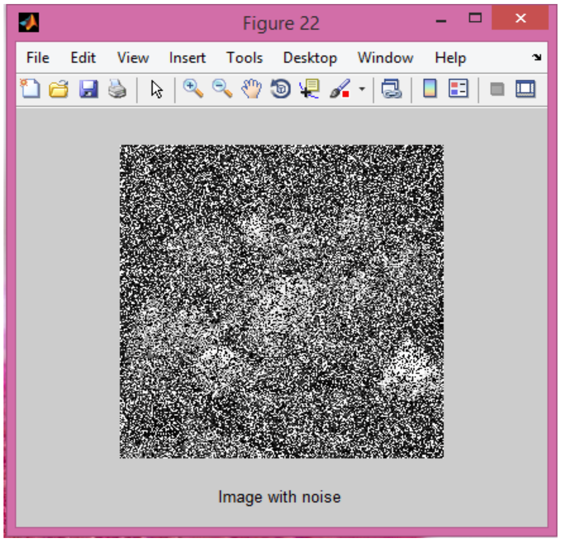


Fig. 2.4(b) image corrupted by 60% noise,

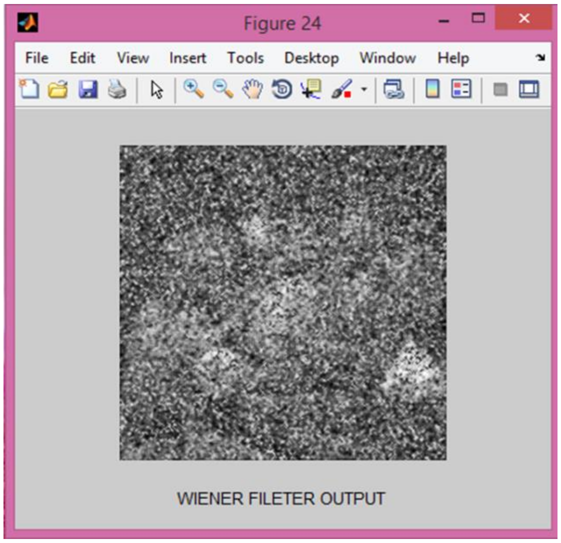


Fig.2.4(c) wiener filter output image

2.3 DRAWBACKS:

These filters performance is not that much better for high noise density. When an image with high noise is passed through any one of above filter, a large amount of noise is present in restored image.

`

CHAPTER-3

3. ADAPTIVE WEIGHTED ALGORITHM

3.1INTRODUCTION:

In Adaptive Weighted approach the output is a weighted sum of the image and de-noising factor. These weighted coefficients depend on a state variable. The state variable is the difference between the current pixel and the average of the remaining pixels in the window.

Adaptive weighted algorithm is used to remove the salt and pepper noise from images. In this algorithm first calculate weights for the noisy image and then adapt that weight's to noisy image.

The proposed method consists of two major steps, detection and filtering.

In this method for the noise image initially, we have to estimate the noise density. Based on the noise density we use mean filtering or adaptive weight algorithm.

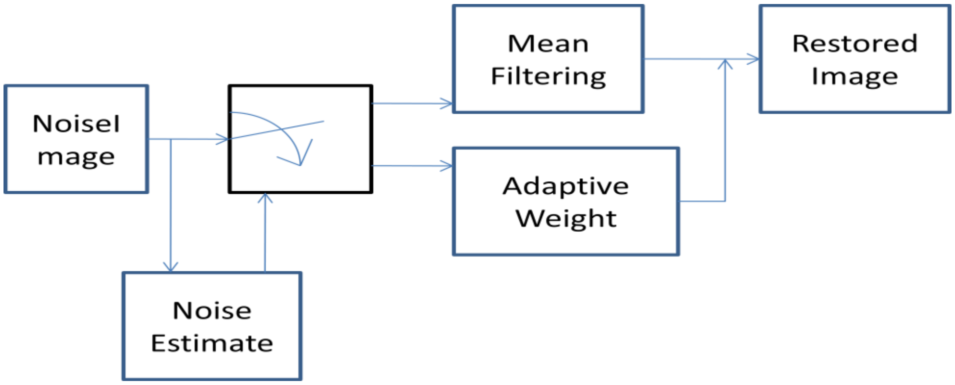


Fig 3.1 Block diagram of adaptive weighted algorithm

Adaptive weighted algorithm consists of two steps.

1. Noise Detection
2. Noise Filtering.

`

3.1.1 Noise Detection:

Noise detection is used to separate the signal pixels and noise pixels.

Let I be the image whose size is M×N, $X_{i,j}$be the gray value of pixel at location (i, j) ,f_{ij}be the value in the label matrix at pixel location (i, j) which is defined by

$$f_{i,j}= \begin{cases} 1 & x_{i,j} = 255 \\ -1 & x_{\mathrm{i,j}} = 0 \\ 0 & else \end{cases} \qquad (3.1)$$

The salt and peppper noise value is the max or min values in the image according to the characteristic of the noise sprinkiling on images,If $f_{i,\,j}$=0, it means the pixel is a signal point, else it is a noise one.

3.1.2 Noise Filtering:

Noise filtering is used to reduce the noise components on an images.

In Noise Filtering first calculate the Noise Density.

Calculation of Noise Density:

Calculate the number of noise pixels in the image.After calculating the number of noise pixels calculate Noise Density by using bellow formula

$$P = \frac{\text{The number of noise pixel}}{\text{M}*\text{N}} \qquad (3.2)$$

After calculating Noise density,based on Noise density we use two methods for filtering of noise.

a)Adjacent signal pixels mean method.
b)Adaptive weighted algotithm.

a) Adjancent signal pixel mean method :

If Noise density is less than 30% ,we can go for Adjancent signal pixel mean method.Small amount of noise can be removed easiley by using Adjancent signnal pixel mean method because there are maney signal pixels around the noise pixels.

Let W_3 be the window of size 3×3 centered at (i,j) and Mean $_{i,j}$ be the mean value of all signal pixels in W_3 .If $f_{i,j}$=0 , $x_{i,j}$ is kept same,because if $f_{i,j}$=0 the corresponding pixel belongs to signal pixel therfore we kept same $x_{i,j}$,else replaced with Mean $_{i,j}$.

$$x_{i,j} = \begin{cases} Mean_{i,j} & f_{i,j} \neq 0 \\ x_{i,j} & f_{i,j} = 0 \end{cases} \qquad (3.3)$$

b) ***Adaptive weighted algorithm :***

If Noise density is more than 30% , go for Adaptive weighted algorithm.If $f_{i,j}$=0 , the corsponding pixel is signal pixel and $x_{i,j}$ is kept same,else take a 3×3 window(W_3)from image,$x_{i,j}$ as the center pixel in $W_3$and compute the number of signal pixels(S_3) in the W_3.If the number of signal pixels(S_3)in $W_3$is zero it means that they are all noise pixels in W_3 then move the window center to the next pixel and again calculate the number of signal pixels(S_3) in the window,if there are signal pixels in window then compute D-value between $x_{i+a,j+b}$(|a,b|≤1 and a.b≠0) and Mean $_{i,j}$ and then compute their inverse marked as $U_{a,b}$

$$U_{a,b} = \begin{cases} \frac{1}{x_{i+a,j+b} - Mean_{i,j}} & f_{i+a.j+b} = 0 \\ 0 & else \end{cases} \qquad (3.4)$$

Because of the correlation between image pixels depending on their spatial location relationship,Generally,the closer the distance between two pixels are,the higher the correlation..So it can make full use of spatial location relation between the neighborhood pixels and the center pixel to determine the neighborhood pixels position weight marked as $V_{a,b}$

(-1,1) 0.25	(0,1) 0.5	(1,1) 0.25
(-1,0) 0.5	(0,0) 0	(1,0) 0.5
(-1,-1) 0.25	(0,-1) 0.5	(1,-1) 0.25

`

$$V_{a,b}= \begin{cases} 0 & a = b = 0 \\ 0.25 & |a| = |b| = 1 \\ 0.5 & else \end{cases} \qquad (3.5)$$

Let $T_{a,b}$ be the initial weight coefficients , $d_{a,b}$ be the normilized weight coefficients.

$$T_{a,b}=U_{a,b}\times V_{a,b} \qquad (3.6)$$

$$d_{a,b}=\frac{T_{a,b}}{\sum_{a=-1}^{1}\sum_{b=-1}^{1}T_{a,b}} \qquad (3.7)$$

After calculating the initial weight coefficients and normalized weight coefficients then multiply normalized weight coefficient by the corresponding $x_{i+a,j+b}$,count up the figure.Their summation is $x'_{i,j}$:

$$x'_{i,j} = \sum_{a=-1}^{1}\sum_{b=-1}^{1} x_{i+a,j+b} \times d_{a,b} \qquad (3.8)$$

$x_{i,j}$ will be substituted by $x'_{i,j}$ and the pixel will be regaeded as a signal point .

$$x_{i,j} = \begin{cases} x'_{i,j} & f_{i,j} \neq 0 \; and \;\; S_3 > 0 \\ x_{i\;j} & else \end{cases} \qquad (3.9)$$

$$f_{i,j} = \begin{cases} f_{i,j} & S_3 = 0 \\ 0 & S_3 \neq 0 \end{cases} \qquad (3.10)$$

After getting the restored image,check whether there is noise present in restored image.If thereis still noise present in restored image,then give restored image as input to the adaptive weighted algorithm.Repeat this untill there is no noise in restored image.

`

CHAPTER-4

SIMULATION RESULTS AND DISCUSSIONS

4.1 INTRODUCTION:

In this section, results are presented to illustrate the performance of the proposed algorithm. Image corrupted with different noise densities are selected for demonstrating the algorithm. A quantitative comparison is performed between several noise removal filters and the proposed algorithm in terms of Noise Density of restored image, Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE).The Proposed Adaptive Weighted algorithm removes salt and pepper noise even for higher noise densities and retains the edges.

4.2 OUTPUT OF PROPOSED METHOD FOR VARIOUS NOISES DENSITIES:

4.2.1 For 10% Noise Density:

Consider peppers image corrupted by 10% noise. As noise density is less than 30% the noisy image is passed through adjacent signal pixel mean method, so the restored image obtained is illustrated in figure 4.1 (b).

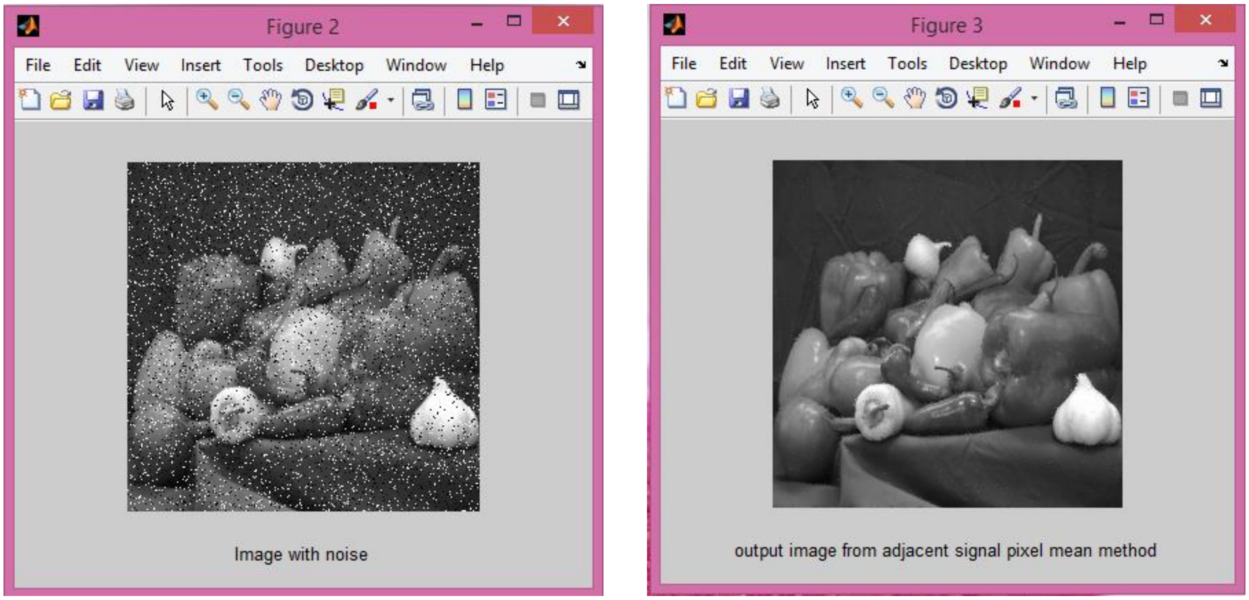


Figure 2
File Edit View Insert Tools Desktop Window Help
Image with noise
Figure 3
File Edit View Insert Tools Desktop Window Help
output image from adjacent signal pixel mean method

(a) (b)

Fig. 4.1 (a) Peppers Image Corrupted by 10% noise (b) Restored image

4.2.2 For 60% Noise Density

Consider peppers image corrupted by 60% noise. As noise density is greater than 30% the noisy image is passed through Adaptive weight algorithm, so the restored image obtained is illustrated in figure 4.2 (b).

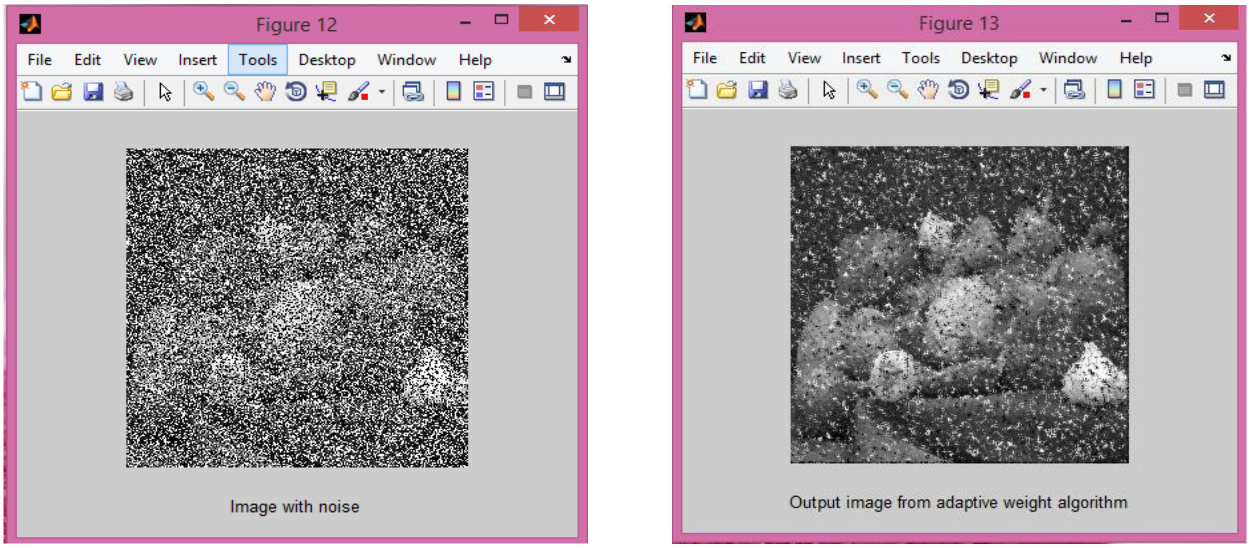


Figure 12
File Edit View Insert Tools Desktop Window Help
Image with noise
Figure 13
File Edit View Insert Tools Desktop Window Help
Output image from adaptive weight algorithm

(a) (b)

Fig. 4.2 (a) Peppers Image Corrupted by 60% noise Fig 4.2(b) Restored image

`

4.3 COMPARISON OF MEAN SQUARE ERROR (MSE) VALUES OF EXISTING AND PROPOSED METHOD FOR DIFFERENT NOISE DENSITIES:

4.3.1 Mean Square Error (MSE):

Mean Square Error (MSE) measures the average of the squares of the “error”. The error is the amount by which the value implied by the estimator differs from the quantity to be estimated. The difference occurs because of randomness or because the estimator doesn't account for information that could produce a more accurate estimate.

For an image, MSE is defined as the square of the difference between restored image and original image. Generally MSE should be a less value.

The MSE equation is given by

$$\mathrm{MSE} = \frac{1}{\mathrm{MN}} \sum_{\mathrm{i}=1}^{\mathrm{M}} \sum_{\mathrm{J}=1}^{\mathrm{N}} \left(\mathrm{r_{ij}} - \mathrm{x_{ij}}\right)^{2} \qquad (4.1)$$

4.3.2 MSE Values for Existing and Proposed Methods:

The MSE values for Existing and Proposed Methods for different noise densities is shown in table 4.1

`

Table 4.1: Comparisons of MSE values for existing and proposed methods

Noise level	Filters(MSE Values)		
	Wiener Filter	Median Filter	Proposed Method
0.1	18.56	0.8809	0.1126
0.2	33.54	1.639	0.527
0.3	40.81	4.018	1.243
0.4	48.78	11.28	6.758
0.5	56.52	37.6	14.35
0.6	58.75	93.95	24.97

4.4 COMPARISON OF PEAK SIGNAL TO NOISE RATIO (PSNR)VALUES OF EXISTING AND PROPOSED FILTERS FOR DIFFERENT NOISE DENSITIES:

4.4.1 Peak Signal to Noise Ratio (PSNR):

The peak signal-to-noise ratio, often abbreviated PSNR, is a term for the ratio between the maximum possible power of a signal and the power of corrupting noise. PSNR is usually expressed in terms of the logarithmic decibel scale.

The PSNR is given by equation

$$psnr = 10log_{10}(\frac{255^2}{MSE}) \qquad (4.2)$$

The PSNR is most commonly used to measure of quality of reconstruction. The signal in this case is the original data, and the noise is the error introduced. In some cases reconstruction may appear to be closer to the original than another, even though it has a lower PSNR (a higher PSNR would normally indicate that the reconstructed image is of high quality).

4.4.2 PSNR Values for Existing and Proposed Methods:

The PSNR values for Existing and Proposed Methods for different noise densities is shown in Table 4.2

Table 4.2: Comparison of PSNR values For Existing and Proposed Filter

Noise level	Filters(PSNR Values)		
	Wiener Filter	Median Filter	Proposed method
0.1	35.45	48.68	57.62
0.2	32.87	45.98	50.91
0.3	32.02	42.09	47.19
0.4	31.25	37.61	39.83
0.5	30.61	32.38	36.56
0.6	30.44	28.4	34.16

The Peak Signal to Noise Ratio (PSNR) of proposed Adaptive Weighted Algorithm is calculated for the densities from 10% to 60% corrupted images. We can clearly observe the difference of PSNR values for existing filters and proposed filter. For example, for 60% corrupted Peppers image when restored with Wiener filter and Median filter the PSNR values are 30.44 dB and 28.4 dB respectively. But when it is passed through Proposed Filter the MSE obtained is 34.16 dB.

It can observe from the Table 4.2, that the PSNR values for proposed method are becoming better for high noise and low noise densities.

`

4.5 GRAPHICAL ANALYSIS:

4.5.1 Graph of Noise Density vs. MSE for Existing and Proposed Filters:

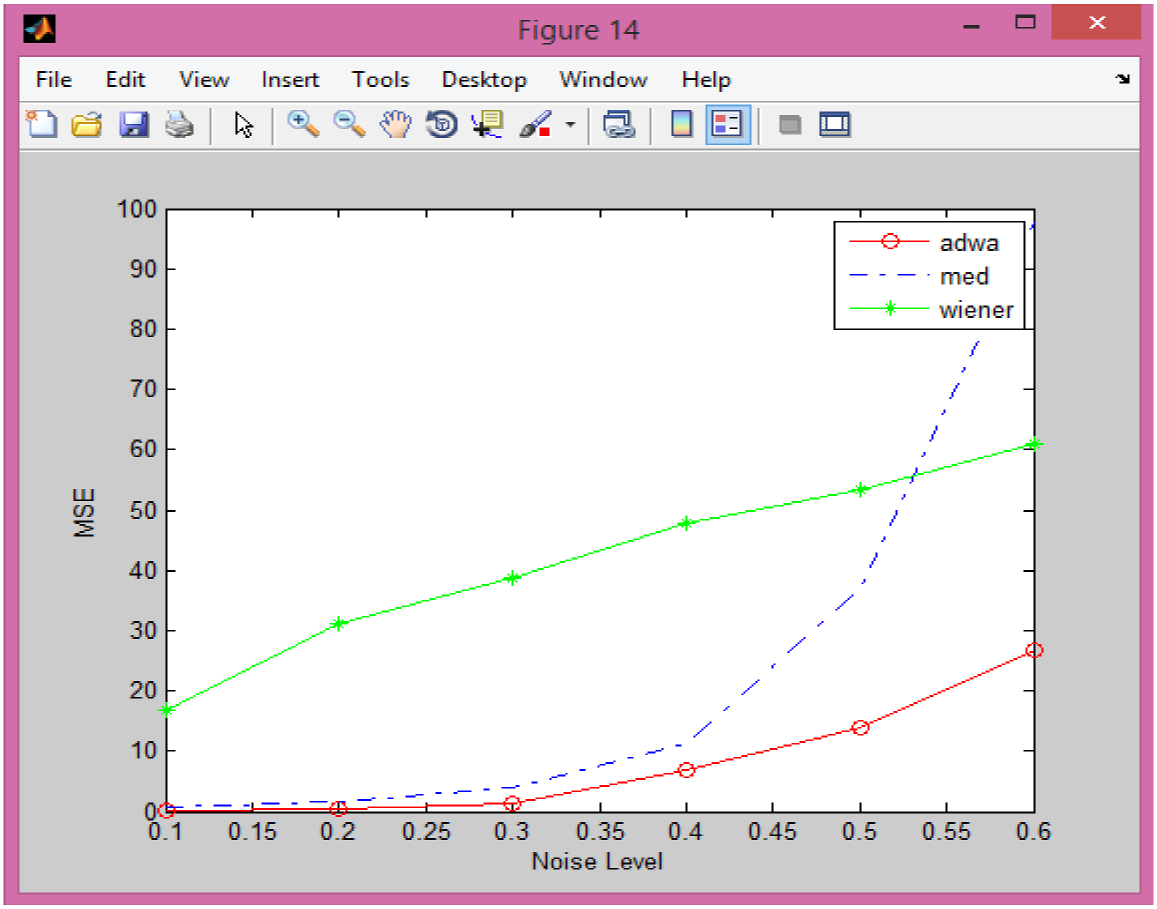


Fig. 4.3 Graph for Noise Density vs. MSE

The Fig 4.3 gives the plot of Noise Density vs. MSE for existing filters MEDIAN, WIENER and proposed filters (Adaptive Weighted Algorithm). We can analyze from the graph that the RED line which indicates proposed filter MSE graph is very much less than that of existing methods which is the desired response. So, proposed filter provides best MSE values compared to existing ones.

`

4.5.2 Graph of Noise Density vs. PSNR for Existing and Proposed Filters:

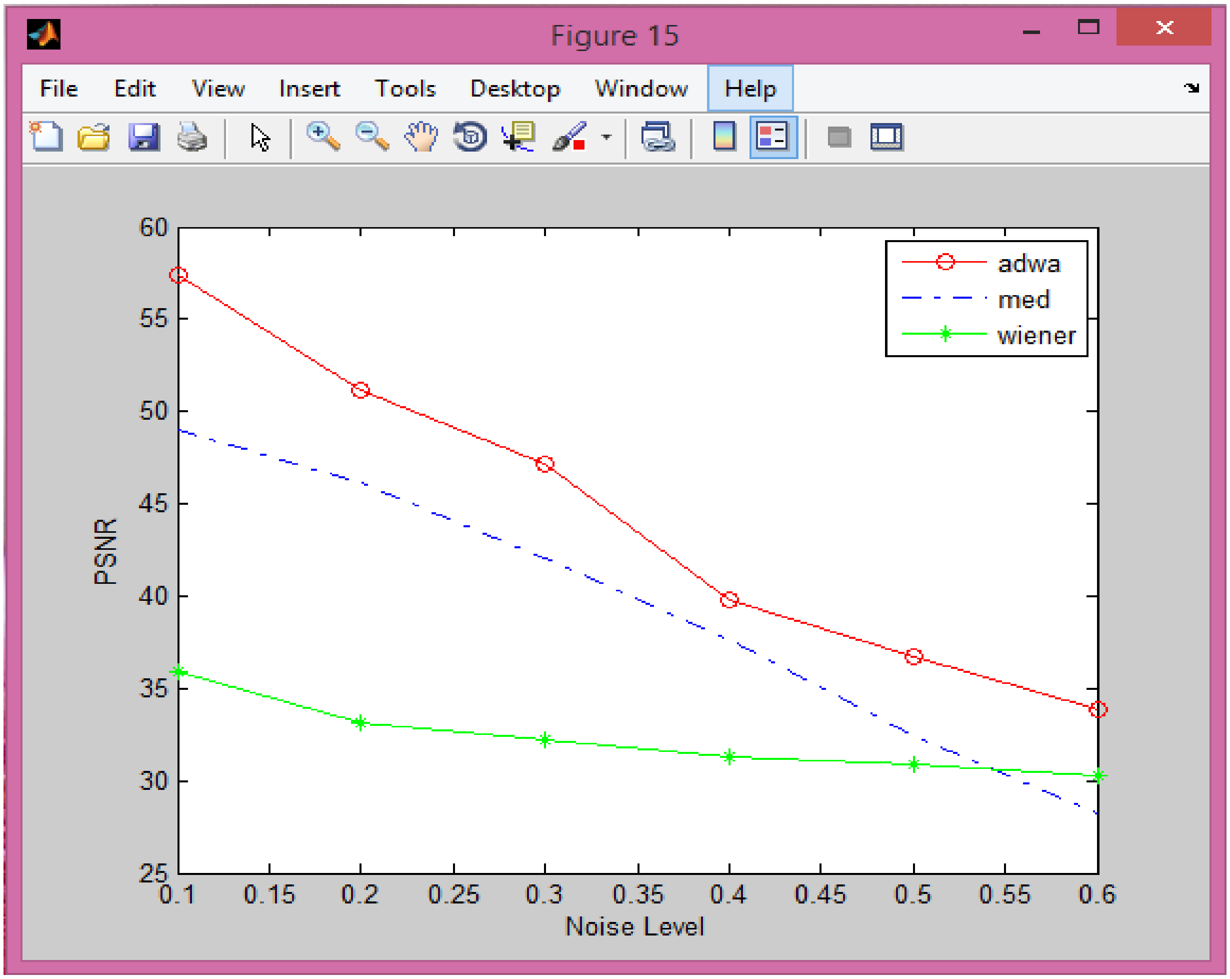


Fig. 4.4 Graph for Noise Density vs. PSNR

The Fig 4.4 gives the plot of Noise Density vs. PSNR for existing filters (MEDIAN, WIENER) and proposed filters (Adaptive Weighted Algorithm). We can analyze from the graph that the RED line which indicates proposed filter PSNR graph is very much higher than that of existing systems which is the desired response. So, proposed filter provides best PSNR values compared to existing ones.

`

CHAPTER-5

CONCLUSION AND FUTURE SCOPE

An effective filtering algorithm is proposed for removing the salt and pepper noise from images for different noise densities. In this algorithm first detect noise pixels, then different methods (adjacent signal pixel mean method and adaptive weighted algorithm) are applied for various noise densities. Simulation results show that this algorithm can suppress noise effectively.

By considering the R, G, B values for a color images, Adaptive Weighted Algorithm can be used to remove the Salt and Pepper noise from color images in future.

`

APPENDIX

MATLAB

A.1 Introduction

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. MATLAB stands for matrix laboratory, and was written originally to provide easy access to matrix software developed by LINPACK (linear system package) and EISPACK (Eigen system package) projects. MATLAB is therefore built on a foundation of sophisticated matrix software in which the basic element is array that does not require pre dimensioning which to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of time.

MATLAB features a family of applications specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow learning and applying specialized technology. These are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control system, neural networks, fuzzy logic, wavelets, simulation and many others.

Typical uses of MATLAB include: Math and computation, Algorithm development, Data acquisition, Modeling, simulation, prototyping, Data analysis, exploration, visualization, Scientific and engineering graphics, Application development, including graphical user interface building.

A.2 Basic Building Blocks of MATLAB

The basic building block of MATLAB is MATRIX. The fundamental data type is the array. Vectors, scalars, real matrices and complex matrix are handled as specific class of this basic data type. The built in functions are optimized for vector operations. No dimension statements are required for vectors or arrays.

A.2.1 MATLAB Window

The MATLAB works based on five windows: Command window, Workspace window, Current directory window, Command history window, Editor Window, Graphics window and Online-help window.

A.2.1.1 Command Window

The command window is where the user types MATLAB commands and expressions at the prompt (>>) and where the output of those commands is displayed. It is opened when the application program is launched. All commands including user-written programs are typed in this window at MATLAB prompt for execution.

A.2.1.2 Work Space Window

MATLAB defines the workspace as the set of variables that the user creates in a work session. The workspace browser shows these variables and some information about them. Double clicking on a variable in the workspace browser launches the Array Editor, which can be used to obtain information.

A.2.1.3 Current Directory Window

The current Directory tab shows the contents of the current directory, whose path is shown in the current directory window. For example, in the windows operating system the path might be as follows: C:\MATLAB\Work, indicating that directory “work” is a subdirectory of the main directory “MATLAB”; which is installed in drive C. Clicking on the arrow in the current directory window shows a list of recently used paths. MATLAB uses a search path to find M-files and other MATLAB related files. Any file run in MATLAB must reside in the current directory or in a directory that is on search path.

A.2.1.4 Command History Window

The Command History Window contains a record of the commands a user has entered in the command window, including both current and previous MATLAB sessions. Previously entered MATLAB commands can be selected and re-executed from the command history window by right clicking on a command or sequence of commands. This is useful to select

`

various options in addition to executing the commands and is useful feature when experimenting with various commands in a work session.

A.2.1.5 Editor Window

The MATLAB editor is both a text editor specialized for creating M-files and a graphical MATLAB debugger. The editor can appear in a window by itself, or it can be a sub window in the desktop. In this window one can write, edit, create and save programs in files called M-files.

MATLAB editor window has numerous pull-down menus for tasks such as saving, viewing, and debugging files. Because it performs some simple checks and also uses color to differentiate between various elements of code, this text editor is recommended as the tool of choice for writing and editing M-functions.

A.2.1.6 Graphics or Figure Window

The output of all graphic commands typed in the command window is seen in this window.

A.2.1.7 Online Help Window

MATLAB provides online help for all it's built in functions and programming language constructs. The principal way to get help online is to use the MATLAB help browser, opened as a separate window either by clicking on the question mark symbol (?) on the desktop toolbar, or by typing help browser at the prompt in the command window. The help Browser is a web browser integrated into the MATLAB desktop that displays a Hypertext Markup Language (HTML) documents. The Help Browser consists of two panes, the help navigator pane, used to find information, and the display pane, used to view the information. Self-explanatory tabs other than navigator pane are used to perform a search.

A.3 MATLAB Files

MATLAB has three types of files for storing information. They are: M-files and MAT-files.

`

A.3.1 M-Files

These are standard ASCII text file with ‘m’ extension to the file name and creating own matrices using M-files, which are text files containing MATLAB code. MATLAB editor or another text editor is used to create a file containing the same statements which are typed at the MATLAB command line and save the file under a name that ends in .m. There are two types of M-files:

1. Script Files

It is an M-file with a set of MATLAB commands in it and is executed by typing name of file on the command line. These files work on global variables currently present in that environment.

2. Function Files

A function file is also an M-file except that the variables in a function file are all local. This type of files begins with a function definition line.

A.3.2 MAT-Files

These are binary data files with .mat extension to the file that are created by MATLAB when the data is saved. The data written in a special format that only MATLAB can read. These are located into MATLAB with ‘load’ command.

A.4 the MATLAB System:

The MATLAB system consists of five main parts:

A.4.1 Development Environment:

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and Command Window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

A.4.2 the MATLAB Mathematical Function:

This is a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigen values, Bessel functions, and fast Fourier transforms.

A.4.3 the MATLAB Language:

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

A.4.4 Graphics:

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

A.4.5 the MATLAB Application Program Interface (API):

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

A.5 SOME BASIC COMMANDS:

pwd prints working directory

Demo demonstrates what is possible in **Mat lab**

Who lists all of the variables in your Mat lab workspace?

Whose list the variables and describes their matrix size

`

figure   creates an empty figure window

hold on holds the current plot and all axis properties so that subsequent

graphing commands add to the existing graph

hold off sets the next plot property of the current axes to "replace"

find   find indices of nonzero elements e.g.:

```
d = find(x>100) returns the indices of the vector x that are greater than 100
```

break   terminate execution of m-file or WHILE or FOR loop

for   repeat statements a specific number of times,

the general form of a FOR statement is:

```
 FOR variable = expr, statement, ..., statement END

        for n=1:cc/c;

        magn(n,1)=NaNmean(a((n-1)*c+1:n*c,1));

        end
```

diff   difference and approximate derivative e.g.:

```
DIFF(X) for a vector X, is [X(2)-X(1)  X(3)-X(2) ... X(n)-X(n-1)].
```

save   saves all the matrices defined in the current session into the file,

matlab.mat, located in the current working directory

load   loads contents of matlab.mat into current workspace

save filename x y z   saves the matrices x, y and z into the file titled filename.mat

save filename x y z /ascii   save the matrices x, y and z into the file titled filename.dat

`

load filename loads the contents of filename into current workspace; the file can be a binary (.mat) file

load filename.dat loads the contents of filename.dat into the variable filename

xlabel(‘ ’) : Allows you to label x-axis

ylabel(‘ ‘) : Allows you to label y-axis

title(‘ ‘) : Allows you to give title for plot

subplot( ) : Allows you to create multiple plots in the same window

A.6 SOME BASIC PLOT COMMANDS:

Kinds of plots:

plot(x,y) creates a Cartesian plot of the vectors x & y

plot(y) creates a plot of y vs. the numerical values of the elements in the y-vector

semilogx(x,y) plots log(x) vs y

semilogy(x,y) plots x vs log(y)

loglog(x,y) plots log(x) vs log(y)

polar(theta,r) creates a polar plot of the vectors r & theta where theta is in radians

bar(x) creates a bar graph of the vector x. (Note also the command stairs(x))

bar(x, y) creates a bar-graph of the elements of the vector y, locating the bars according to the vector elements of 'x'

`

Plot description:

grid creates a grid on the graphics plot

title('text') places a title at top of graphics plot

xlabel('text') writes 'text' beneath the x-axis of a plot

ylabel('text') writes 'text' beside the y-axis of a plot

text(x,y,'text') writes 'text' at the location (x,y)

text(x,y,'text','sc') writes 'text' at point x,y assuming lower left corner is (0,0)

and upper right corner is (1,1)

axis([xmin xmax ymin ymax])sets scaling for the x- and y-axes on the current plot

A.8 MATLAB WORKING ENVIRONMENT:

A.8.1 Matlab Desktop

Matlab Desktop is the main Matlab application window. The desktop contains five sub windows, the command window, the workspace browser, the current directory window, the command history window, and one or more figure windows, which are shown only when the user displays a graphic.

The command window is where the user types MATLAB commands and expressions at the prompt (>>) and where the output of those commands is displayed. MATLAB defines the workspace as the set of variables that the user creates in a work session.

The workspace browser shows these variables and some information about them. Double clicking on a variable in the workspace browser launches the Array Editor, which can be used to obtain information and income instances edit certain properties of the variable.

The current Directory tab above the workspace tab, shows the contents of the current directory, whose path is shown in the current directory window. For example, in the windows

`

operating system the path might be as follows: C:\MATLAB\Work, indicating that directory "work" is a subdirectory of the main directory "MATLAB"; WHICH IS INSTALLED IN DRIVE C. clicking on the arrow in the current directory window shows a list of recently used paths. Clicking on the button to the right of the window allows the user to change the current directory.

MATLAB uses a search path to find M-files and other MATLAB related files, which are organize in directories in the computer file system. Any file run in MATLAB must reside in the current directory or in a directory that is on search path. By default, the files supplied with MATLAB and math works toolboxes are included in the search path.The easiest way to see which directories is soon the search path, or to add or modify a search path, is to select set path from the File menu the desktop, and then use the set path dialog box. It is good practice to add any commonly used directories to the search path to avoid repeatedly having the change the current directory.

The Command History Window contains a record of the commands a user has entered in the command window, including both current and previous MATLAB sessions. Previously entered MATLAB commands can be selected and re-executed from the command history window by right clicking on a command or sequence of commands.

This action launches a menu from which to select various options in addition to executing the commands. This is useful to select various options in addition to executing the commands. This is a useful feature when experimenting with various commands in a work session.

A.8.2 Using the MATLAB Editor to create M-Files:

The MATLAB editor is both a text editor specialized for creating M-files and a graphical MATLAB debugger. The editor can appear in a window by itself, or it can be a sub window in the desktop. M-files are denoted by the extension .m, as in pixelup.m.

The MATLAB editor window has numerous pull-down menus for tasks such as saving, viewing, and debugging files. Because it performs some simple checks and also uses color to differentiate between various elements of code, this text editor is recommended as the tool of choice for writing and editing M-functions.

`

To open the editor, type edit at the prompt opens the M-file filename.m in an editor window, ready for editing. As noted earlier, the file must be in the current directory, or in a directory in the search path.

A.8.3 Getting Help:

The principal way to get help online is to use the MATLAB help browser, opened as a separate window either by clicking on the question mark symbol (?) on the desktop toolbar, or by typing help browser at the prompt in the command window. The help Browser is a web browser integrated into the MATLAB desktop that displays a Hypertext Markup Language (HTML) documents. The Help Browser consists of two panes, the help navigator pane, used to find information, and the display pane, used to view the information. Self-explanatory tabs other than navigator pane are used to perform a search**.**

REFERENCES:

1. T.A. Nodes and N.C. Gallagher, Jr., “The output distribution of median type filters” IEEE Trans.Commum., 32(5):532-541.
2. “Wang and D.Zhang”, Progressive switching median filter for the removal of impulse noise from highly corrupted image”, IEEE Trans Circuits syst. II, Analog Digit, Signal Pricess,46(1):78-80.
3. S.J Ko and Y.H.Lee “center weighted median filter and their applications to image enhancement”, Pattern Recognit.Lett.15 (4):341-347.
4. E.Abreu, M.Lighstone, S.Mitrs and K.Arakawa”, A New efficient approach for the removal of impulse noise from highly corrupted images”, IEEE Trans.Image Processing, 5(6):1012-1025.
5. Shuqun Zhang and Mohammad A.Karim.”A New impulse detector for switching median filters”, IEEE Trans on image processing, 13(2):234-237.
6. T.Kasparis,N.Tzannes and Q.Chen,”Detail-Preservating adaptive conditional median filters” ,Electron Img,1(4):358-364.
7. H.Hwang and R.A.Hadded,”Adaptive median filters:New algotithms and results”,IEEE Trans.Image process,4(4):499-502.
8. H.L Eng and K.K Ma “Noise Adaptive Soft-Switching Median Filter”IEEE Trans.Image process. Vol.10 No.2, pp, 242-251.
9. S.Zhang and M.A.Karim,”A new impulse detector for switching median filters”IEEE signal process, Lett, .Vol.9, No.11, pp.360-363.
10. H.Hwang and R.A.Haddad.” Adaptive median filters:New algorithms and results”,IEEE Trans.Image Process,Lett., Vol.4,No.4,pp 499-502.
11. S.J.Ko and Y.H.Lee,”Center weighted median filters and their applications to image enhancement”IEEE Trans,Circuit systems,Vol.38,no.9,pp.984-993