IMAGE ENHANCEMENT USING

NONLINEAR TECHNIQUES

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology in

Electronics and Communication Engineering

under the Supervision of

Mr. PARDEEP GARG

By

SIMANDEEP SINGH (091049)

ARJUN AGARWAL (091050)

KEVIN JAMES (091086)

Jaypee University of Information and Technology

Waknaghat, Solan – 173234, Himachal Pradesh

Table of Contents

Certificate………………………………………………………………………III

Acknowledgement…………………………………………………………….. IV

List of figures………………………………………………………………..V

Abstract……………………………………………………………………...VI

1. Digital Image……………………………………………………………..1
2. Digital Image Enhancement……………………………………….......4
   - 2.1 Introduction………..……………………………………………..4
   - 2.2 Spatial Domain Method……..5
     - 2.2.1 Enhancement Techniques in Spatial Domain …………6
     - 2.2.2 Histogram Equalization..6
     - 2.2.3 Removal of Noise by Filtering.......................................7
     - 2.2.4 Image Negative..9
     - 2.2.5 Gamma Transformation..10
     - 2.2.6 Contrast Stretching...12
     - 2.2.7 Image Averaging..13
     - 2.2.8 Log Transformation..14
     - 2.2.9 Bit Plane Slicing..16
     - 2.2.10 Gray Level Slicing...17
     - 2.2.11 Image Subtraction..18

3 Frequency Domain Methods………………………………..........................20
   - 3.1 Introduction...20
   - 3.2 Spatial Filtering...20
   - 3.3 Smoothing Spatial filter...20
     - 3.3.1 Gaussian Smoothing Operator…………………………....21
   - 3.4 Order Statistics Filter...23
   - 3.5 Sharpening Filters..24
   - 3.6 First Order and Second Order Derivative...24
     - 3.6.1 First Order Derivatives………………………………...24
     - 3.6.2 Second Order Derivatives……………………………......25
     - 3.6.3 The Laplacian……………………………………….....26

4 Noises in Image…………………………...28

4.1 Various types of Noises……………………………………………...29

4.2 Gaussian Noise……………………………………………………...29

4.3 Rayleigh Noise………………………………………………………31

4.4 Salt & Pepper Noise…………………………………………………31

4.5 Impulse Noise………………………………………………………..31

5 Linear Image Enhancement……………………………………………..34

5.1 Introduction………………………………………………………….34

5.2 Removing Noise by Linear Filtering………………………………...35

5.3 Removing Noise by Median Filtering……………………………......35

5.4 Removing Noise by Mode Filtering…………………………………35

6 Non-Linear Image Enhancement……………………………………….36

7 Methodology…………………………………………………………….37

8 Algorithm………………………………………………………………..39

9 Conclusion & Results…………………………………………………...42

References……………………………………………………………….44

Certificate

This is to certify that project report entitled "**Image Enhancement Using Nonlinear Techniques**", submitted by Simandeep Singh , Arjun Agarwal and Kevin James in partial fulfillment for the award of degree of Bachelor of Technology in Electronics and Communication Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

DATE **Mr. PARDEEP GARG**

Sr. Lecturer

Acknowledgment

We would like to take this opportunity to thank all those who have been supporting us during the making of the project.

Our deepest regards to **Mr. Pardeep Garg,** our project mentor, for his guidance and support at every step with attention and care. He has taken great pains to go through the project and make necessary corrections as and when needed and without him this project would have been a distant reality.

We would also like to thank our institution for the work environment and all faculty members, especially **Prof. Dr. Sunil Bhooshan,** our H.O.D**.**

DATE:

SIMANDEEP SINGH

ARJUN AGARWAL

KEVIN JAMES

LIST OF FIGURES

1. Digital Image
1.1 Image representation using 2-D array……………………………………………1
2. Digital Image Enhancement
2.1 Image after Histogram equalization…..……………………………….……..5

2.2 Image after 2x2 filtering……..…………………………………………………….6

2.3 Image negative……….…………………………………………..........……….7

2.4 Gamma Transformation graph..………………………………………………….9

2.5 Image with gamma correction……..……………………………………...10

2.6 histogram after logarithmic transformation………..………………………….13
2.7 Bit Plane Slicing……………………………………………………………14
2.8 Gray level slicing……………………………………………………….15
3. Frequency Domain Methods
3.1 1-D Gaussian distribution ……………………………………………………19

3.2 2-D Gaussian distribution ………………………………………………..……19

4. Noises in Image
4.1 Gaussian Noise …………………………………………...............………25

4.2 Impulse Noise ……………………………………………………………..28

Abstract

An image enhancement algorithm based on a neighborhood dependent nonlinear model is presented to improve visual quality of digital images.

In our project, we are using various non linear digital enhancement techniques so as to remove noise in a digital image as well as to improve the visual quality of digital images as well as images that exhibits dark shadows due to limited dynamic range of imaging.

We have used nonlinear image enhancement because nonlinear image enhancement tools are less susceptible to noise. Noise is always present due to the physical randomness of image acquisition systems. For example, underexposure and low-light conditions in analog photography conditions lead to images with film-grain noise which, together with the image signal itself, are captured during the digitization process.

Nonlinear methods effectively preserve edges and details of images while methods using linear operators tend to blur and distort them.

We have developed a GUI in which we have used mean filtering non linearly to enhance an image which is able to enhance the luminance in the dark shadows keeping the overall tonality consistent with that of the input image .

We would first start by describing what an image is and then we will focus on different image enhancement techniques including linear and non linear image enhancement.

CHAPTER 1

DIGITAL IMAGE

1.1 INTRODUCTION

An image is a two-dimensional function f(x,y), where x and y are the spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x , y) is called the intensity of the image at that level.

If (x , y) and the amplitude values of f are finite and discrete quantities, we call the image a digital image. A digital image is composed of a finite number of elements called pixels, each of which has a particular location and value of f at (x, y) is proportional to the brightness of the image at that point. It is represented by a 2-dimensional integer array. The digitized brightness value is called the gray level value. Each element of the array is called a pixel. A digital N×N image looks like this:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(N-1,0) & f(N-1,1) & \cdots & f(N-1,N-1) \end{bmatrix}$$

[1]

Figure1.1

1.2 TYPES OF DIGITAL IMAGES

Four types of images are considered in digital image processing and they are as follows:

1.2.1 Black and White Images

A black and white image is made up of pixels each of which holds a single number corresponding to the gray level of the image at a particular location. These gray levels span the full range from black to white in a series of very fine steps, normally 256 different gray levels.

1.2.2 Color Images

A color image is made up of pixels each of which holds three numbers corresponding to the red, green, and blue levels of the image at a particular location. Red, green, and blue (sometimes referred to as RGB) are the primary colors for mixing light.

Any color can be created by mixing the correct amounts of red, green, and blue light. Assuming 256 levels for each primary, each color pixel can be stored in three bytes (24 bits) of memory. This corresponds to roughly 16.7 million different possible colors.

1.2.3 Binary or Bi-level Images

Binary images use only a single bit to represent each pixel. Since a bit can only exist in two states—on or off, every pixel in a binary image must be one of two colors, usually black or white. This inability to represent intermediate shades of gray is what limits their usefulness in dealing with photographic images.

1.2.4 Indexed Color Images

Some color images are created using a limited palette of colors, typically 256 different colors. These images are referred to as indexed color images because the data for each pixel consists of a palette index indicating which of the colors in the palette applies to that pixel.

CHAPTER-2

DIGITAL IMAGE ENHANCEMENT

2.1 INTRODUCTION

The principal objective of enhancement is to process an image so that the result is more suitable than the original image for a specific application.

Digital image enhancement and analysis have played, and will continue to play, an important role in scientific, industrial, and military applications. In addition to these applications, image enhancement and analysis are increasingly being used in consumer electronics and Internet Web users.

As an example, image enhancement might be used to emphasize the edges within the image. This edge-enhanced image would be more visually pleasing to the naked eye, or perhaps could serve as an input to a machine that would detect the edges and perhaps make measurements of shape and size of the detected edges. Image enhancement is important because of its usefulness in virtually all image processing applications.

Image enhancement tools are often classified into (a) **point operations** and (b) **spatial operations**. Point operations include contrast stretching, noise clipping, histogram modification, and pseudo-coloring. Point operations are, in general, simple nonlinear operations that are well known in the image processing literature.

Spatial operations used in image processing today are, on the other hand, typically linear operations. The reason for this is that spatial linear operations are simple and easily implemented. Although linear image enhancement tools are often adequate in many applications, significant advantages in image enhancement can be attained if nonlinear techniques are applied. Nonlinear methods effectively preserve edges and details of images while methods using linear operators tend to blur and distort them.

Image enhancement approaches fall into two broad
Categories:

- **Spatial domain methods**
- **Frequency domain methods**

2.2 SPATIAL DOMAIN METHODS:

Spatial domain refers to the aggregate of pixels composing an image. Spatial domain methods are procedures that operate directly on the pixels

$$g(x,y) = T[f(x,y)]$$

f(x,y) : input image

g(x,y) : output image

T[ ] : an operator defined over some neighbourhood of (x,y)

The principal approach in defining a neighbourhood about a point (x,y) is to use a square or rectangular sub image area centred at (x,y).

The centre of the subimage is moved from pixel to pixel starting at the top left corner. The operator T is applied at each location (x,y) to yield the output g, at that location.

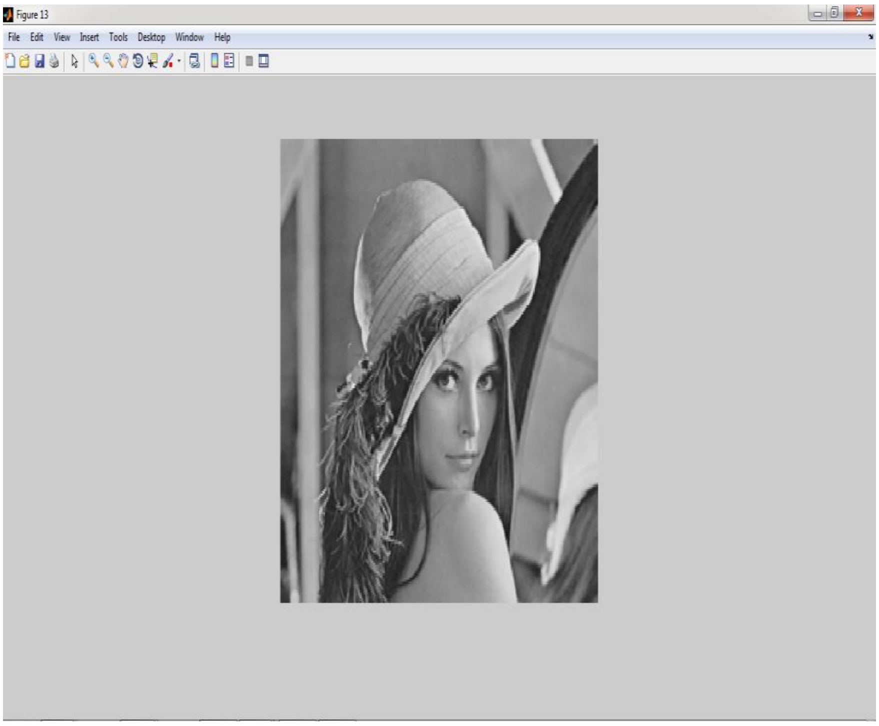
2.2.1 Enhancement techniques in spatial domain:

- Histogram Equalization.
- Noise Removal & Filtering.
- Image Negative.
- Gamma Transformation.
- Contrast Stretching.

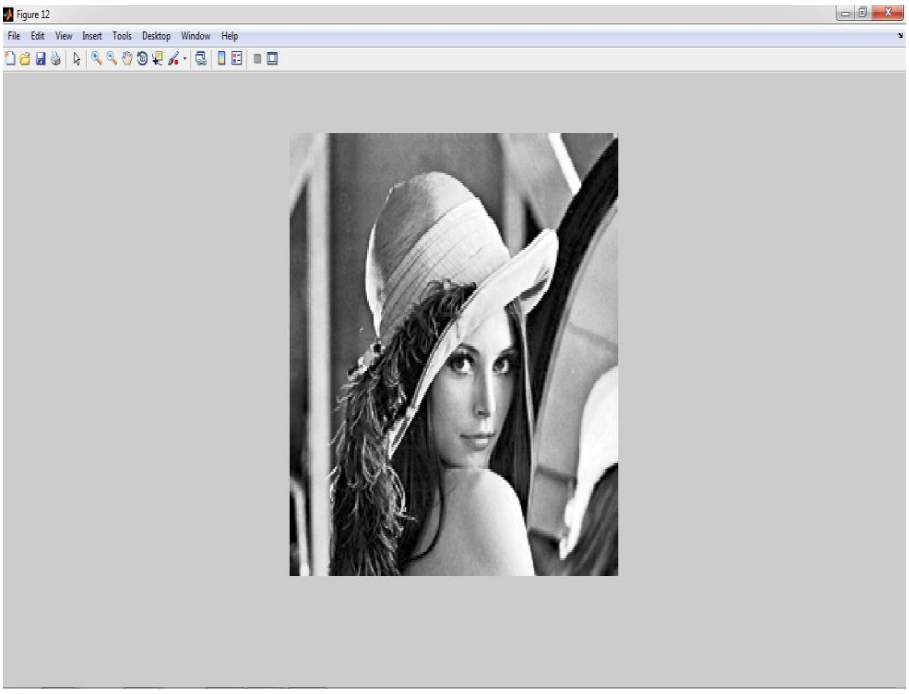
2.2.2 Histogram Equalization

Consider a transformation such as s = T(r), $0 \le r \le 1$ that produces a level S for every pixel value R in the original Image.

In this process output Image is obtained by mapping each pixel with level R(k) in the input Image into a corresponding pixel with level S(k) in the output Image , via transformation function T mentioned above.



(a)



(b)

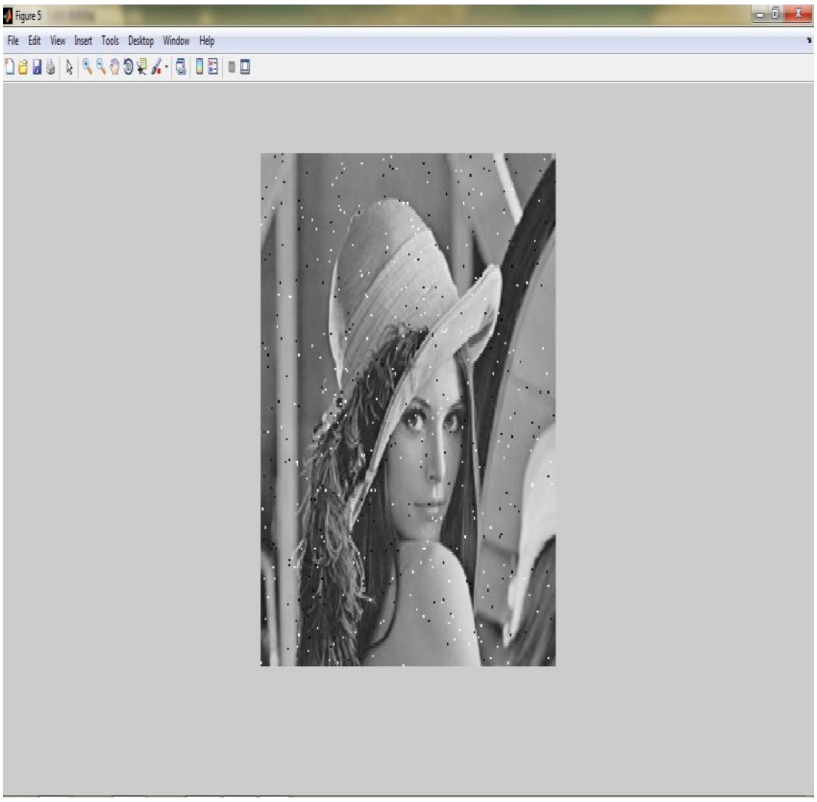
Fig2.1: (a)Original image, (b)enhanced image from histogram equalization

2.2.3 Removal of Noise By Filtering

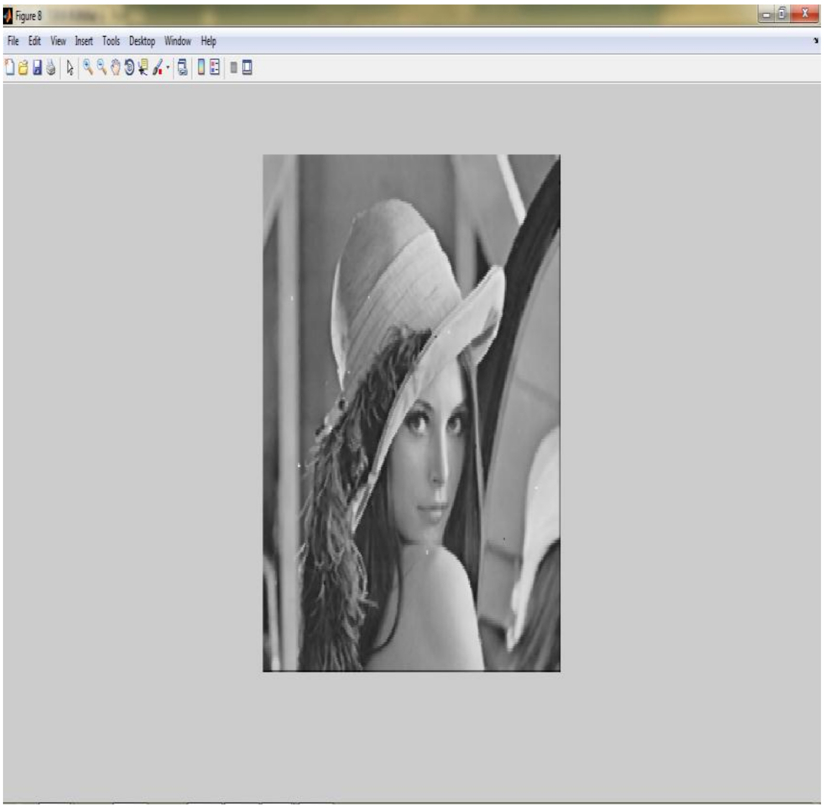
Different types of noises from a digital image can be removed using various types of filters.

Types of noises are:

- Gaussian noise.
- Poisson noise.
- Salt & Pepper noise.
- Speckle noise



(a)



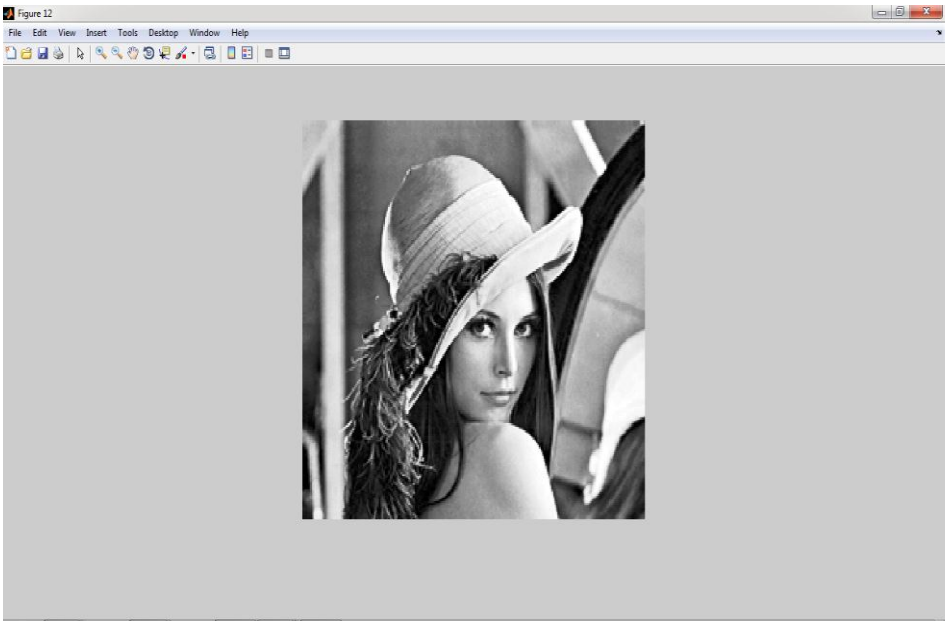
(b)

Fig2.2: (a)Image with noise, (b) Filtered image using 2 X 2 filter

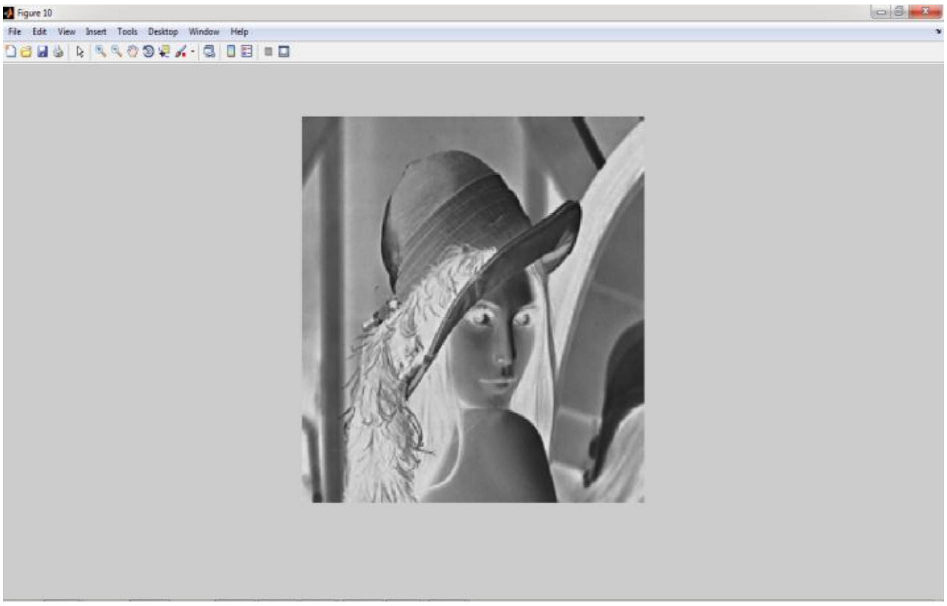
2.2.4 Image Negative

Image in the range [0, L-1] is obtained by using the negative transformation, which is given by:

$$S = L - 1 – R$$



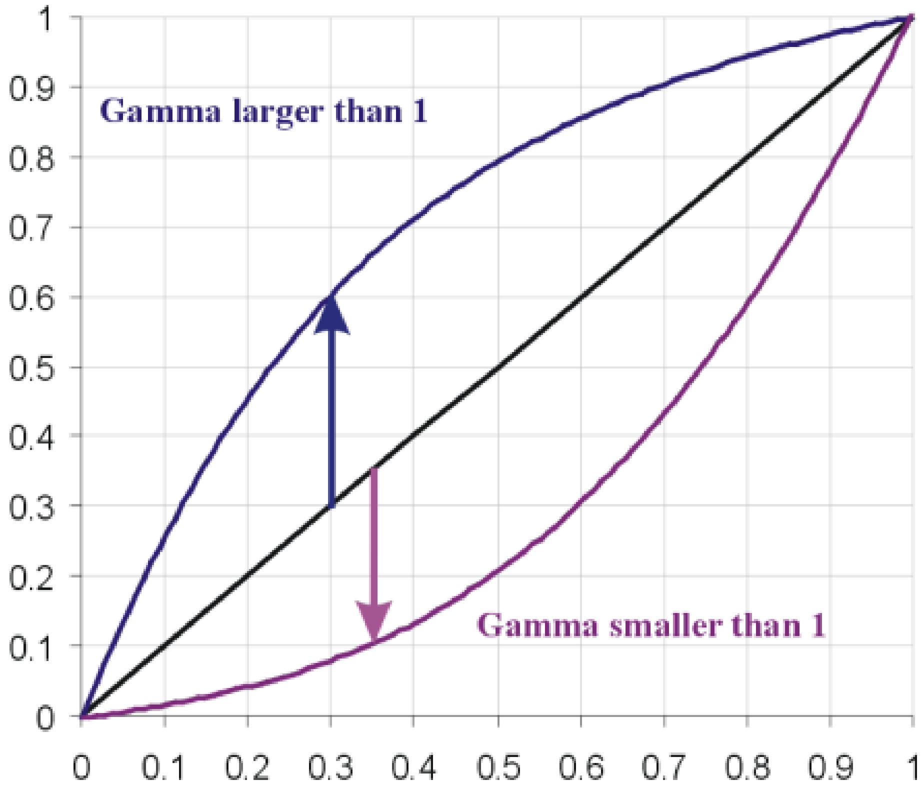
(a)



(b)

Fig2.3: (a)Original image. (b)Image negative

2.2.5 Gamma Transformation



[2]

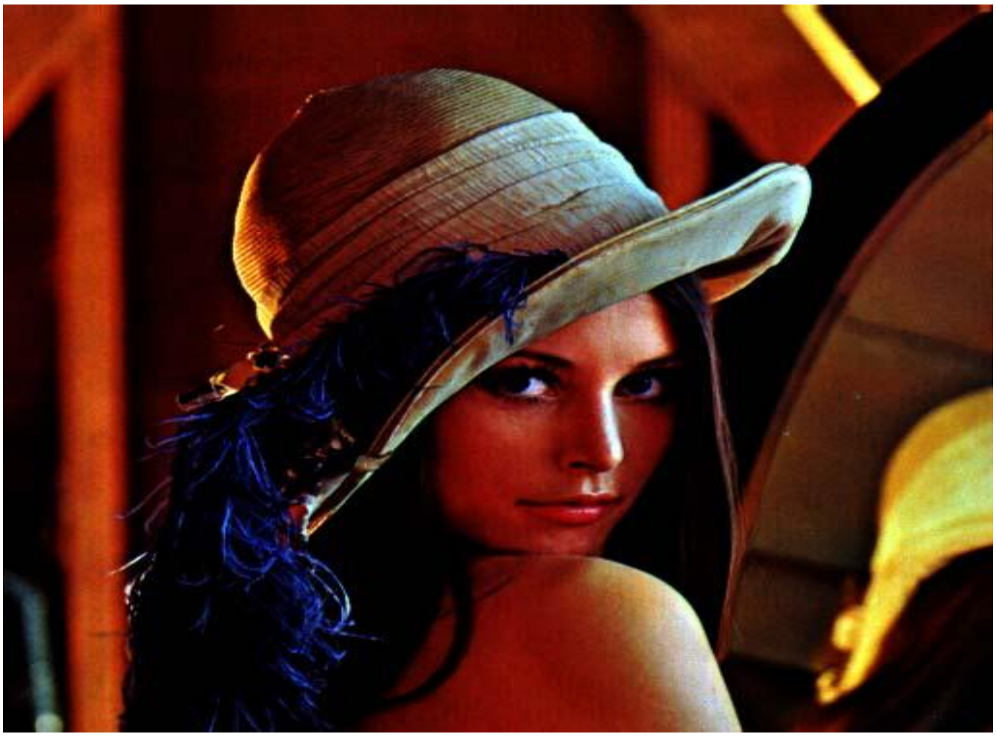
Fig2.4:Gamma Transformation graph

Gamma transformation has a basic form:

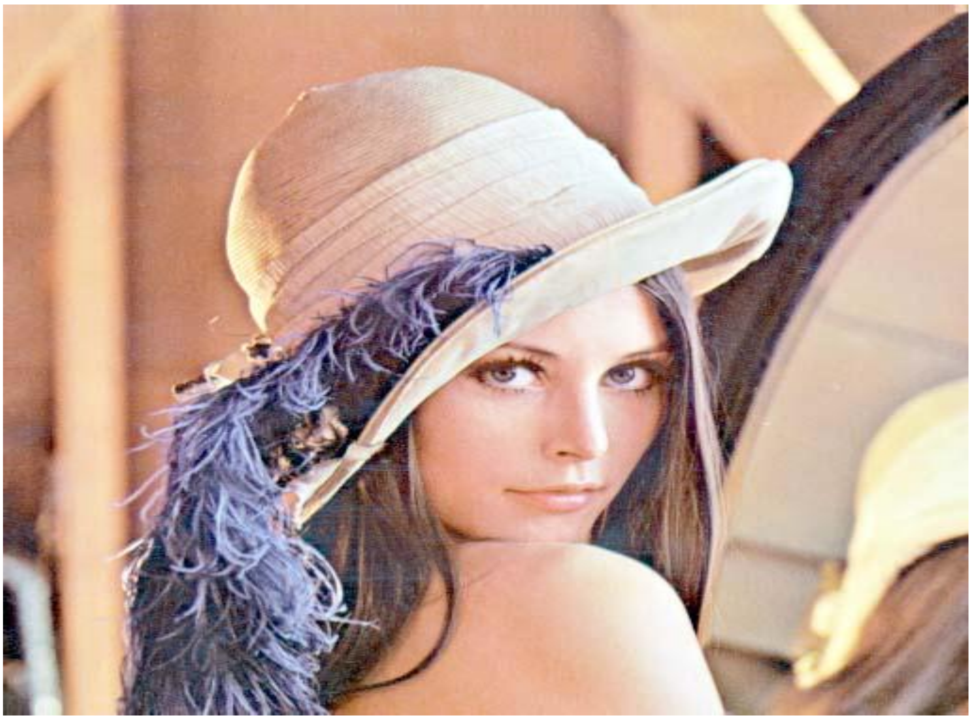
S = C R^ ɣ

Where C & ɣ are constants.

Gamma is an important but seldom understood characteristic of virtually all digital imaging systems. It defines the relationship between a pixel's numerical value and its actual luminance. Without gamma, shades captured by digital cameras wouldn't appear as they did to our eyes.



(a) Gamma=0.25



[3]

(b) Gamma=2.0

Fig 2.5: (a)image with gamma correction=0.25 (b)image with gamma correction=2.0

2.2.6 Contrast Stretching

Images with poor illumination, lack of dynamic range in the image sensor or wrong setting of a lens aperture.

The contrast stretching increases the dynamic range of the gray levels.

In this the values of the output and input gray levels decide the type of transformation.

Depending on these values transformation function can be linear function which does not change any output gray level values.

Transformation function can be a thresholding function which will provide an output image, according to the given input conditions.

r = Input Gray Level

s = Output Gray Level

The location of the points (r1 , s1) and (r2 , s2) controls the shape of the transformation, and the type of transformation function is decided.

Various values of(r1 , s1) and (r2 , s2)produce various degrees of spread in the gray levels of the output image , thus affecting its contrast. In general the values of (r1 , s1) and (r2 , s2) , are such assumed so that the function is single valued and monotonically increasing.

This condition preserves the order of gray level, thus preventing the creation of intensity artifacts in the processed image.

Highlighting a specific range of gray levels in an image often is desired. Applications include enhancing features such as masses of water in satellite imagery and enhancing flaws in an X-ray images. There are several ways of doing level slicing ,but most of them are variations of two basic themes.

One approach is display a high value for all the gray levels in the range of interest and a low value for all the other gray levels.

The second approach, based on the transformation brightens the desired gray levels and preserves the background and gray-level tonalities in a image.

2.2.7 Image Averaging

Image noise can compromise the level of detail in your digital or film photos, and so reducing this noise can greatly enhance your final image or print. The problem is that most techniques to reduce or remove noise always end up softening the image as well. Some softening may be acceptable for images consisting primarily of smooth water or skies, but foliage in landscapes can suffer with even conservative attempts to reduce noise.

Image averaging works on the assumption that the noise in your image is truly random. This way, random fluctuations above and below actual image data will gradually even out as one averages more and more images. If you were to take two shots of a smooth gray patch, using the same camera settings and under identical conditions (temperature, lighting, etc.), then you would obtain images similar to those shown on the left.

Image averaging is common in high-end astrophotography, but is arguably underutilized for other types of low-light and night photography. Averaging has the power to reduce noise without compromising detail, because it actually increases the signal to noise ratio (SNR) of your image. An added bonus is that averaging may also increase the bit depth of your image — beyond what would be possible with a single image.

The image averaging works by adding each image into a moving average of image. This effectively creates the same effect of averaging many image without the significant memory and time that averaging hundreds of images would take.

The averaging effect can be used to remove fast moving objects from images if a high image averaging is used. If fewer images are averaged the effect creates a washing or strobe like effect.

2.2.8 Log Transformation:

Image enhancement simply means, transforming an image *f* into image *g* using *T*. Where *T* is the transformation. The values of pixels in images *f* and *g* are denoted by *r*and *s*, respectively. As said, the pixel values *r* and *s* are related by the expression,

$$s = T(r)$$

where *T* is a transformation that maps a pixel value *r* into a pixel value *s*. The results of this transformation are mapped into the grey sclale range as we are dealing here only with grey scale digital images. So, the results are mapped back into the range [0, L-1], where $L=2^k$, k being the number of bits in the image being considered. So, for instance, for an 8-bit image the range of pixel values will be [0, 255].

The log transformation is given by the expression,

$$s = c \log(1 + r)$$

where *c* is a constant and it is assumed that $r \geq 0$. The shape of the log curve in fig. A tells that this transformation maps a narrow range of low-level grey scale intensities into a wider range of output values. And similarly maps the wide range of high-level grey scale intensities into a narrow range of high level output values. The opposite of this applies for inverse-log transform. This transform is used to expand values of dark pixels and compress values of bright pixels.

The logarithm function tends to squeeze together the larger values in your data set and stretches out the smaller values.The following illustration shows the histogram of a log-normal distribution (left side) and the histogram after logarithmic transformation.

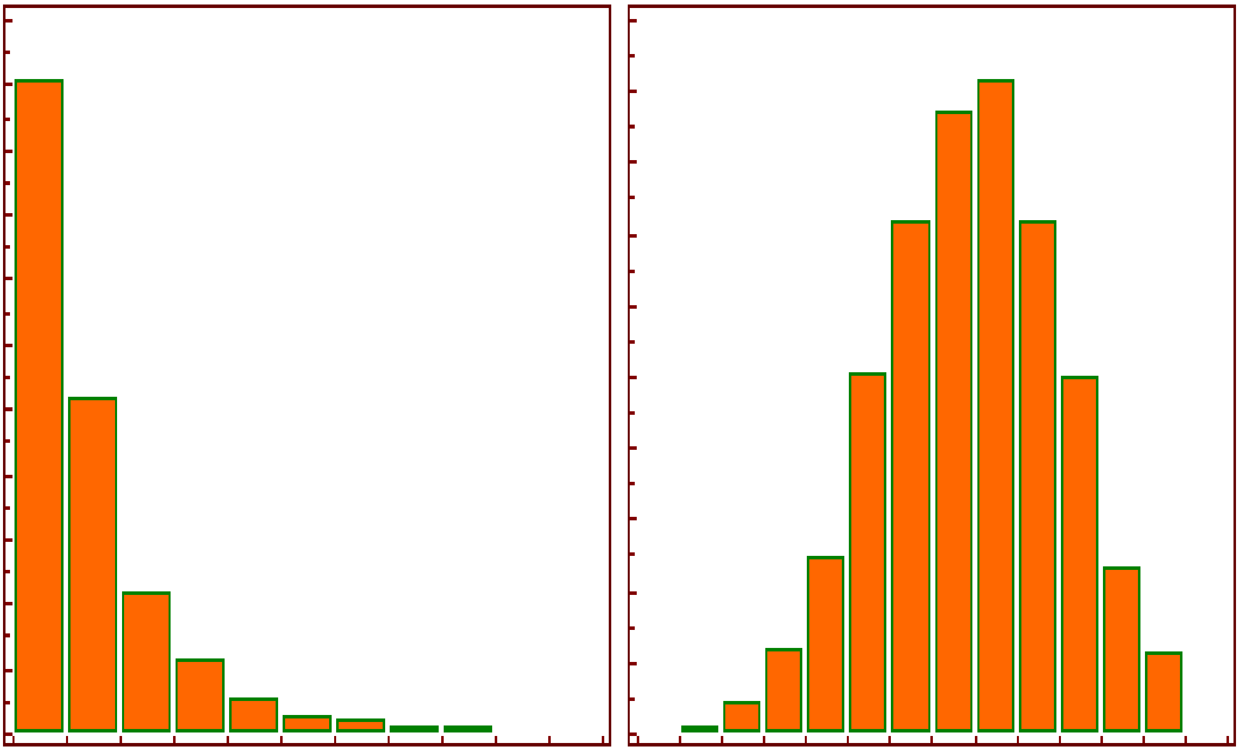


Fig2.6

When you select log transformation, MedCalc computes the base-10 logarithm of each data value and then analyses the resulting data. For ease of interpretation, the results of calculations and tests are backtransformed to their original scale.

Original number = x

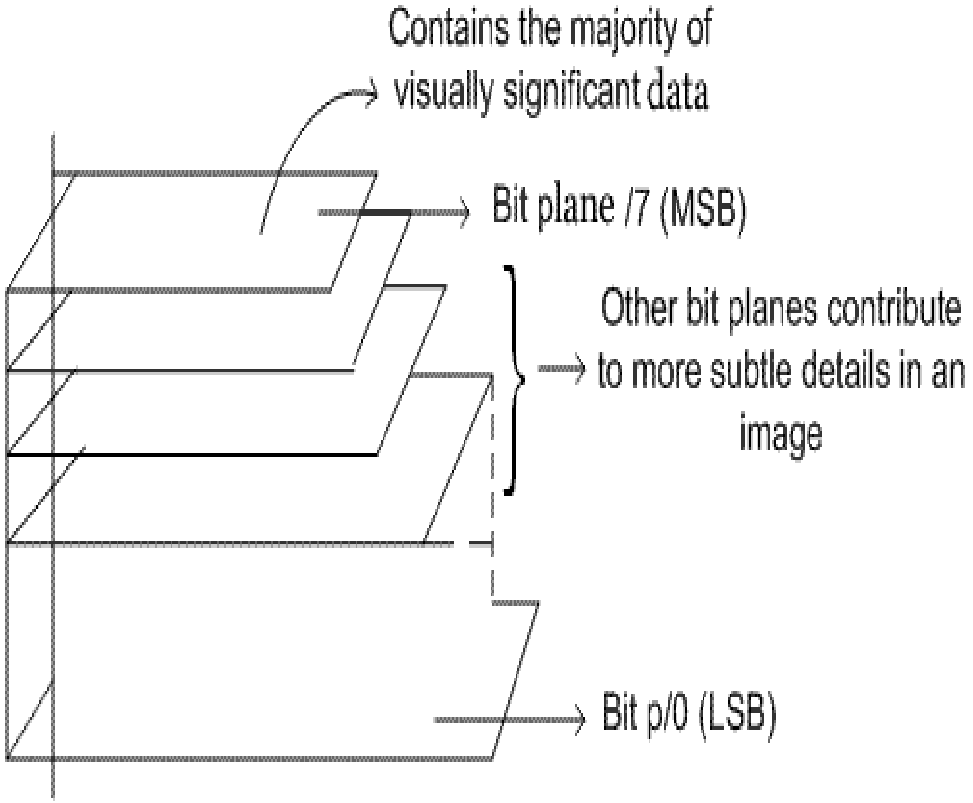
Transformed number x'=$\log_{10}$(x)

`

2.2.9 Bit Plane Slicing

Instead of highlighting gray level images, highlighting the contribution made to total image appearance by specific bits might be desired. Suppose that each pixel in an image is represented by 8 bits. Imagine the image is composed of 8, 1-bit planes ranging from bit plane1-0 (LSB)to bit plane 7 (MSB).

In terms of 8-bits bytes, plane 0 contains all lowest order bits in the bytes comprising the pixels in the image and plane 7 contains all high order bits.


Contains the majority of visually significant data
Bit plane /7 (MSB)
Other bit planes contribute to more subtle details in an image
Bit p/0 (LSB)

[2]

Fig2.7

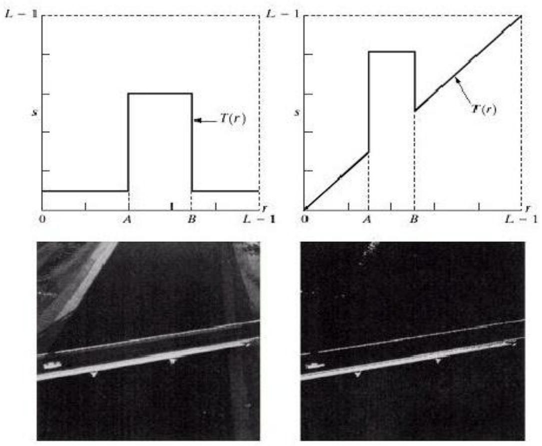
Separating a digital image into its bit planes is useful for analyzing the relative importance played by each bit of the image, implying, it determines the adequacy of numbers of bits used to quantize each pixel , useful for image compression.

In terms of bit-plane extraction for a 8-bit image, it is seen that binary image for bit plane 7 is obtained by proceeding the input image with a thresholding gray-level transformation function that maps all levels between 0 and 127 to one level and maps all levels from 129 to 253 to another (eg. 255)

2.2.10 Gray Level Slicing:

Gray-scale modification (also called gray-level scaling) methods belong in the category of point operations and function by changing the pixel's (gray-level) values by a mapping equation. The mapping equation is typically linear (nonlinear equations can be modeled by piecewise linear models) and maps the original gray-level values to other, specified values. Typical applications include contrast enhancement and feature enhancement.

The primary operations applied to the gray scale of an image are to compress or stretch it. We typically compress gray-level ranges that are of little interest to us and stretch the gray-level ranges where we desire more information. This is illustrated in Figure 4.2-1a, where the original image data are shown on the horizontal axis and the modified values are shown on the vertical axis. The linear equations corresponding to the lines shown on the graph represent the mapping equations. If the slope of the line is between zero and one, this is called gray-level compression, whereas if the slope is greater than one, it is called gray-level stretching. In Figure 4.2-1a, the range of graylevel values from 28 to 75 is stretched, while the other gray values are left alone. The original and modified images are shown in Figures 4.2-1b, c, where we can see that stretching this range exposed previously hidden visual information. In some cases we may want to stretch a specific range of gray levels, while clipping the values at the low and high ends.



[2]

Fig2.8: Gray Level Slicing

2.2.11 Image subtraction:

The pixel subtraction operator takes two images as input and produces as output a third image whose pixel values are simply those of the first image minus the corresponding pixel values from the second image. It is also often possible to just use a single image as input and subtract a constant value from all the pixels. Some versions of the operator will just output the absolute difference between pixel values, rather than the straightforward signed output.

How It Works

The subtraction of two images is performed straightforwardly in a single pass. The output pixel values are given by:

$$Q = P1(I , j) – P2(I , j)$$

Or if the operator computes absolute differences between the two input images then:

$$Q = |P1(I , j) – P2(I , j)|$$

Or if it is simply desired to subtract a constant value *C* from a single image then:

$$Q = P1(I , j) – c$$

If the pixel values in the input images are actually vectors rather than scalar values (*e.g.* for color images) then the individual components (*e.g.* red, blue and green components) are simply subtracted separately to produce the output value.

Implementations of the operator vary as to what they do if the output pixel values are negative. Some work with image formats that support negatively-valued pixels, in which case the negative values are fine (and the way in which they are displayed will be determined by the display color map). If the image format does not support negative numbers then often such pixels are just set to zero (*i.e.* black typically). Alternatively, the operator may`wrap' negative values, so that for instance -30 appears in the output as 226 (assuming 8-bit pixel values).

If the operator calculates absolute differences and the two input images use the same pixel value type, then it is impossible for the output pixel values to be outside the range that may be represented by the input pixel type and so this problem does not arise. This is one good reason for using absolute differences.

CHAPTER 3

FREQUENCY DOMAIN METHODS

3.1 INTRODUCTION

•Fourier Series: Any function that periodically repeats itself can be expressed as the sum of sines/cosines of different frequencies, each multiplied with a different coefficient.

•Fourier Transform: Functions that are not periodic, whose area under the curve is finite, can be expressed as the integral of sines and/cosines multiplied by a weighting function.

3.2 Spatial Filtering

Neighbourhood operations use a subimage that has the same dimension as the neighbour hood. The subimage is called a filter, mask, kernel, template, or window. The values in filter subimage are referred to as coefficients, rather than pixels. The response of the filter at a point is calculated using a predefined relationship. For linear spatial filtering, the response is given by a sum of products of the filter coefficients and the corresponding image pixels in the area spanned by the filter mask.

3.3 Smoothing Spatial Filters

Smoothing Filters are used for blurring and for noise reduction

Blurring is used for

- Removal of small details prior to object extraction.
- Bridging of small gaps in lines or curves.

Smoothing Linear Filters (Averaging Filters)

- Replace the average value defined by the filter mask.
- Have the undesirable effect of blur edges

3.3.1 Gaussian Smoothing Operator:

The Gaussian smoothing operator is a 2-D convolution operator that is used to `blur' images and remove detail and noise. In this sense it is similar to the filter, but it uses a different kernel that represents the shape of a Gaussian (`bell-shaped') hump. This kernel has some special properties which are detailed below.

How It Works:

The Gaussian distribution in 1-D has the form:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma}} e -\frac{x^2}{2\sigma^2}$$

where σ is the standard deviation of the distribution. We have also assumed that the distribution has a mean of zero (*i.e.* it is centered on the line *x*=0). The distribution is illustrated in Figure 4.1.

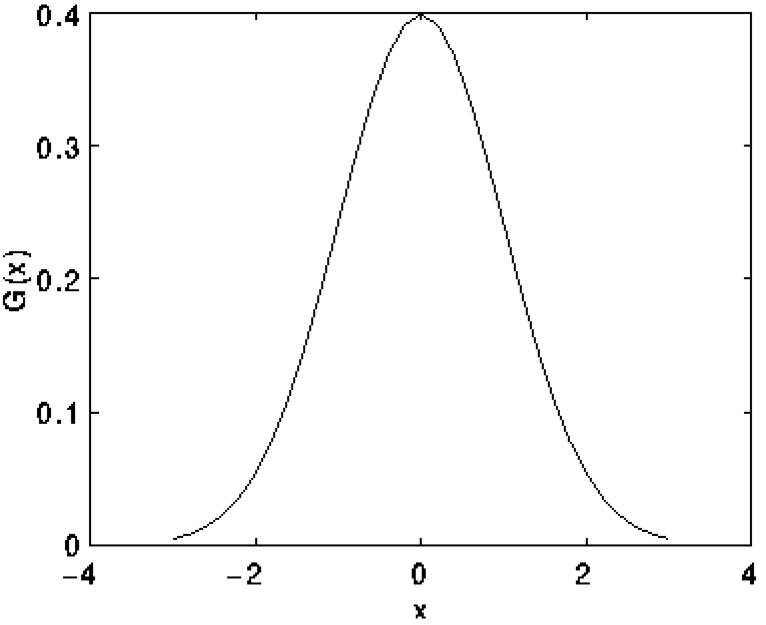


Figure 3.1 1-D Gaussian distribution with mean 0 and σ =1

In 2-D, an isotropic (*i.e.* circularly symmetric) Gaussian has the form:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma}} e -\tfrac{x^2}{2\sigma^2}$$

This distribution is shown in Figure 2.

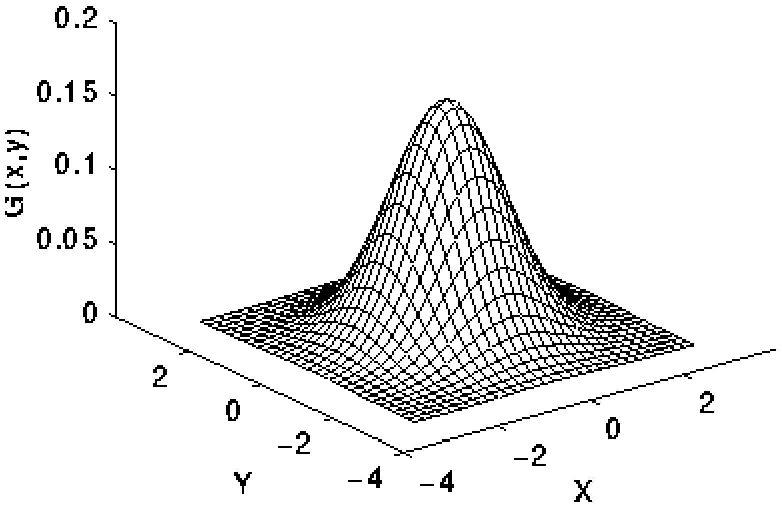


Figure 3.2 2-D Gaussian distribution with mean (0,0) and σ = 1

3.4 Order-Statistics Filters

- Nonlinear spatial filters
- Response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter.
- Replacing of the centre pixel with the value determined by the ranking result
- Median filter
    - ➢ Replaces the pixel value by the median value in the neighbourhood
    - ➢ Excellent noise-reduction capabilities with less blurring than linear smoothing filters
    - ➢ Effective for impulse noise (salt-and-pepper noise)

Order-statistics filters are nonlinear spatial filters whose response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determined by the ranking result.

3.5 Sharpening Filters

- The principal objective of sharpening is to highlight fine detail in an image or to enhance detail that has been blurred
- While averaging is analogous to integration, sharpening could be accomplished by spatial differentiation
- Image differentiation enhances edges and other discontinuities

Human perception is highly sensitive to edges and fine details of an image, and since they are composed primarily by high frequency components, the visual quality of an image can be enormously degraded if the high frequencies are attenuated or completed removed. In contrast, enhancing the high-frequency components of an image leads to an improvement in the visual quality. Image sharpening refers to any enhancement technique that highlights edges and fine details in an image. Image sharpening is widely used in printing and photographic industries for increasing the local contrast and sharpening the images.

In principle, image sharpening consists of adding to the original image a signal that is proportional to a high-pass filtered version of the original image.

3.6 First- and second-order derivatives

3.6.1 First-order derivatives:

- ➢ Produce thicker edges in an image
- ➢ Stronger response to a gray-level step
- ➢ Edge extraction

The derivative of digital functions are defined in terms of differences. There are different ways of defining these differences. However , we require that first order derivative must be

- Nonzero in flat areas.
- Nonzero at the onset of the gray-level step.
- nonzero along ramp

Since we are dealing with digital quantities whose values are finite, the maximum possible gray-level change also is finite, and the shortest distance is between the adjacent pixels.

A basic definition of the first-order derivative of a one-dimensional function f(x) is the difference

$$\frac{\delta f}{\delta x} = f(x+1) - f(x),$$

We used a partial derivative here in order to keep the notation the same as when we consider an image function of two variables, f(x , y), at which time we will considering the partial derivatives along the two spatial axes. Use of a partial derivative in this condition does not affect it.

3.6.2 Second-order derivatives:

- ➢ Produce double response at step changes in gray level
- ➢ Stronger response to fine detail

Any definition of the second order must satisfy

- ➢ Zero in flat areas.
- ➢ Nonzero at the onset and end of a gray-level step.
- ➢ Zero along ramps of constant slope.

We define the second-order derivative as the difference,

$$\frac{\delta^2 f}{\delta^2 x} = f(x+1) + f(x-1) - 2f(x)$$

It is easily verified that these two definitions satisfy the conditions stated previously regarding derivatives of the first and the second order.

Let us consider the properties of the first and second order derivative as we traversed the profile from left to right. First we note that the first order derivative is nonzero along the entire ramp, while the second order derivative is nonzero only at the onset an at the end of the ramp.

Because of the edges in an image resembles the type of transition ,we conclude that first-order derivatives have thick edges, and second-order derivatives have a much finer one.

In summary, comparing the response between the first and second-order derivative we arrive at the following conclusion:

- First-order derivatives generally produce thicker edges.
- Second-order derivatives have a stronger response to the finer details ,such as thin lines and isolated points.
- First-order derivatives generally have a stronger response to the gray level step.
- Second-order derivatives produce a double response at step changes in gray level

3.6.3 The Laplacian (2nd order derivative):

Shown by Rosenfeld and Kak[1982] that the simplest isotropic derivative operator is the Laplacian is defined as

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Discrete form of derivative:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

AND

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

The digital implementation of the 2-Dimensional Laplacian is obtained by summing 2 components

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial x^2}$$

$$\nabla^2 f = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

Implementation

$$g(x, y) = \begin{cases} f(x, y) - \nabla^2 f(x, y) & \textit{If the center coefficient is negative} \\ f(x, y) + \nabla^2 f(x, y) & \textit{If the center coefficient is positive} \end{cases}$$

Where f(x, y) is the original image is Laplacian filtered image g(x , y) is the sharpen image.

CHAPTER 4

Noises in an Image

Any real world sensor is affected by a certain degree of noise, whether it is thermal, electrical or otherwise. This noise will corrupt the true measurement of the signal, such that any resulting data is a combination of signal and noise.

Additive noise, probably the most common type, can be expressed as:

$I(t) = S(t) + N(t)$ (1)

where I(t) is the resulting data measured at time t, S(t) is the original signal measured, and N(t) is the noise introduced by the sampling process, environment and other sources of interference.

Digital images are prone to a variety of types of noise. Noise is the result of errors in the image acquisition process that result in pixel values that do not reflect the true intensities of the real scene. There are several ways that noise can be introduced into an image, depending on how the image is created. For example:

- If the image is scanned from a photograph made on film, the film grain is a source of noise. Noise can also be the result of damage to the film, or be introduced by the scanner itself.
- If the image is acquired directly in a digital format, the mechanism for gathering the data (such as a CCD detector) can introduce noise.
- Electronic transmission of image data can introduce noise.

4.1 Various Types of noises:

1. Gaussian Noise
2. Rayleigh Noise
3. Salt &Pepper Noise
4. Impulse Noise

4.2 Gaussian Noise:

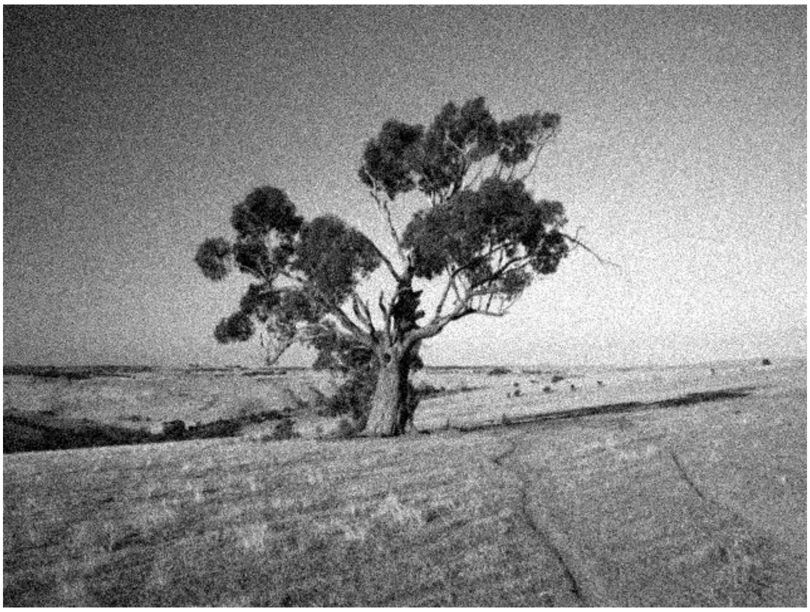
Gaussian noise is properly defined as the noise with a Gaussian amplitude distribution. This says nothing of the correlation of the noise in time or of the spectral density of the noise. Labeling Gaussian noise as *white* describes the correlation of the noise. It is necessary to use the term **white Gaussian noise** to be precise. Gaussian noise is sometimes equated to white Gaussian noise, but this is not necessarily the case.

In Gaussian noise models, the noise density follows a Gaussian normal distribution

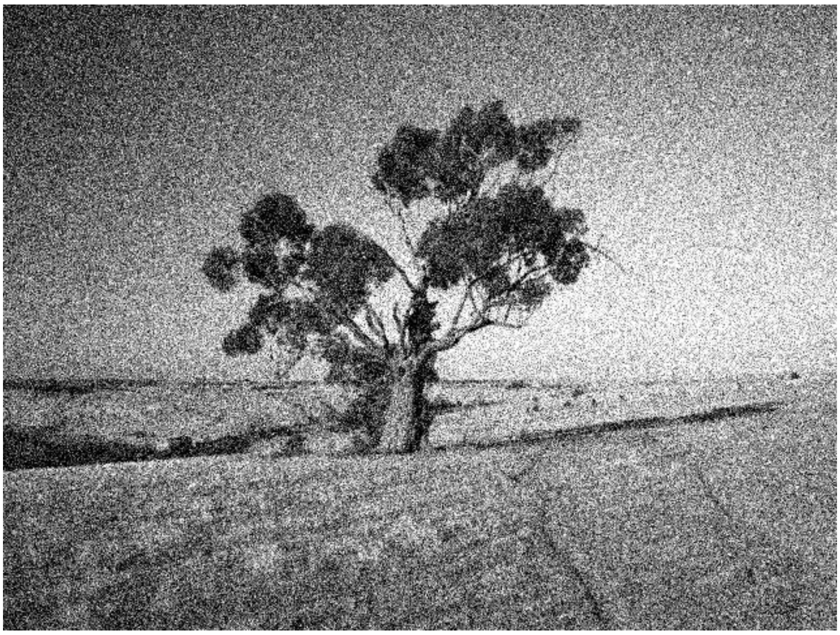
G(¯x, σ) defined by the mean ¯x and standard deviation σ.

The itk::Additive Gaussian Noise Filter can be used to add noise to a volume.

(a) $\sigma = 10$



(b) $\sigma = 20$



(c) $\sigma = 50$

Figure 4.1: (a), (b), (c) Gaussian Noise: the effect of standard deviation (σ)

4.3 Rayleigh Noise:

Rayleigh noise is the noise described by the random process with the Rayleigh distribution of probability density function.

$$w\left({}^{u}\!/\sigma\right) = {}^{u}\!/\sigma \exp\left({}^{u}\!/\sqrt{2}\sigma\right), \text{ u > 0}$$

$$= 0 \text{ , u < 0}$$

Where u is instantaneous noise voltage

σ is noise variance.

4.4 Salt & Pepper Noise:

An image containing salt-and-pepper noise will have dark pixels in bright regions and bright pixels in dark regions. This type of noise can be caused by analog-to-digital converter errors, bit errors in transmission, etc.

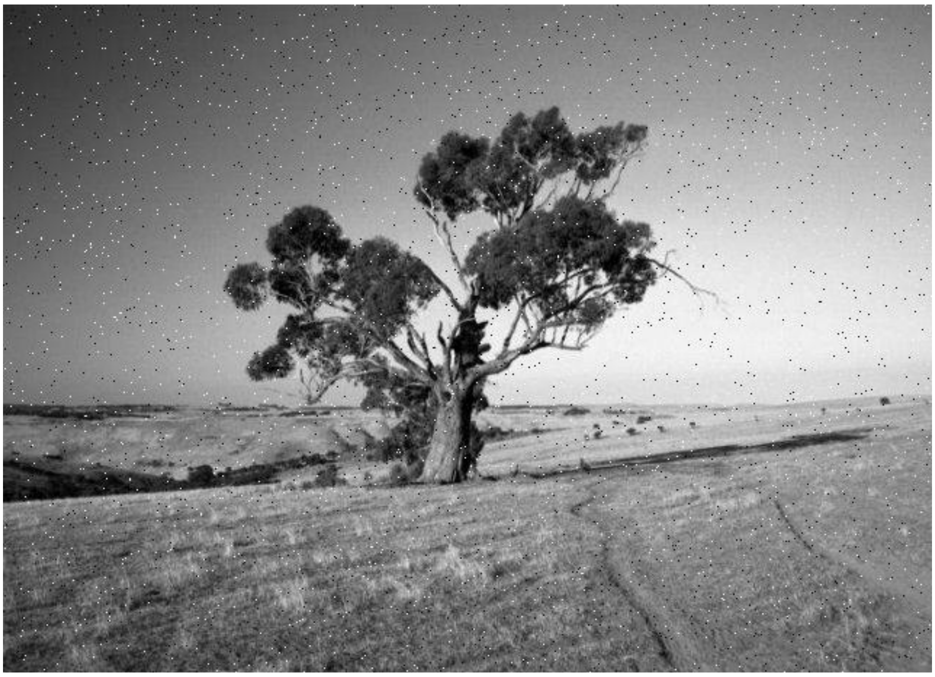
For an 8-bit image, the typical value for pepper noise is 0, and 255 for salt-noise.

4.5 Impulse Noise

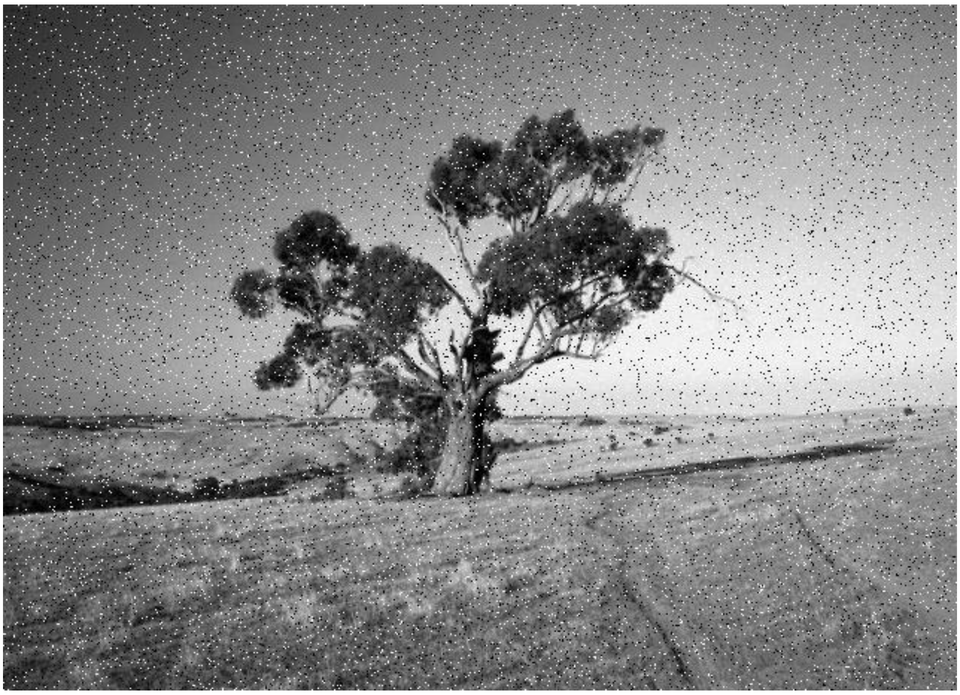
Impulse noise, often found in digital transmission and storage, can be described by the following model:

I(t) = (1 − e)S(t) + eN(t) (3)

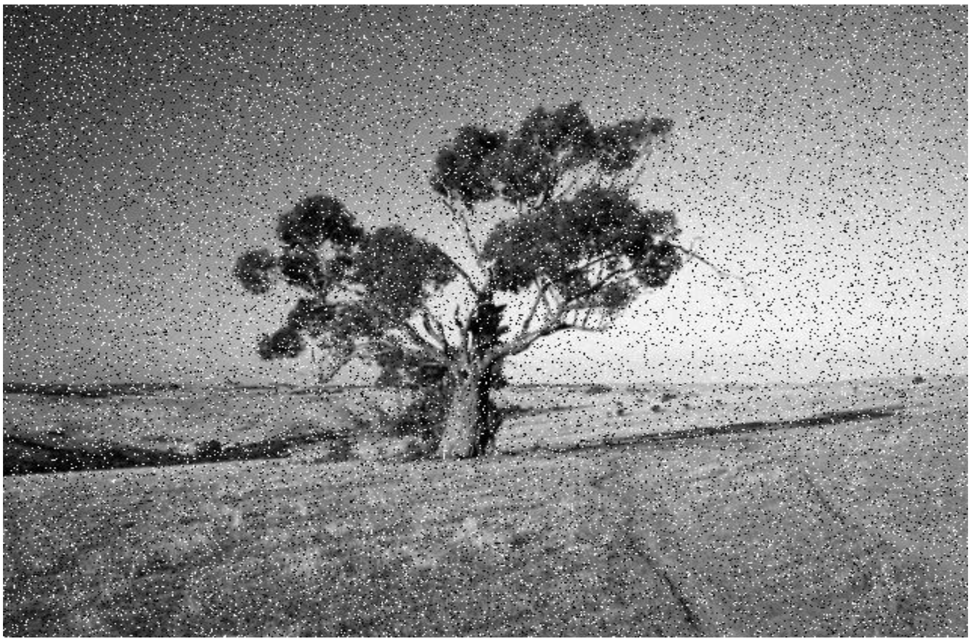
where e = {0, 1} with a probability P. Impulse noise is also known as “speckle” or “salt and pepper” noise. For this filter, the distribution between salt and pepper is 50%. An image progressively degraded with impulse noise is shown below.



(a) p=0.01



(b) p=0.05



(c) p=0.1

Figure 4.2: (a), (b), (c) Impulse Noise: the effect of probability *p*

CHAPTER 5

Linear Image Enhancement

5.1 INTRODUCTION

Linear image enhancement techniques are those which are applied on the whole image itself. We implement linear image enhancement by the use of various linear filters. Linear image enhancement also makes subtle variations within the data more obvious.

Linear filtering is the basis for linear spatial enhancement techniques. An interesting subcategory of linear filtering is space-invariant linear filtering.

A useful property of space-invariant filtering is that the input–output relation remains constant over the entire image. Thus, the value of the impulse response depends only on the distance between the input and output pixels and not on their spatial location.

Another useful property of linear space-invariant filtering is that it can be performed both in the spatial and in the Fourier frequency domains.

Thus, linear space-invariant filtering is performed either by convolving the input image with the impulse response of the filter or by multiplying on a point-by-point basis the Fourier transform of the image with the frequency response of the filter. The ability to perform linear filtering in both the spatial and the Fourier domains has a number of advantages. First, it facilitates the design of filters because it helps separate the part of the signal that should be retained from the part that should be attenuated or discarded. Second, it helps improve the speed of computation by utilizing a fast Fourier transform (FFT) algorithm for computing the DFT.

5.2 Removing Noise by Linear Filtering

You can use linear filtering to remove certain types of noise. Certain filters, such as averaging or Gaussian filters, are appropriate for this purpose. For example, an averaging filter is useful for removing grain noise from a photograph. Because each pixel gets set to the average of the pixels in its neighborhood, local variations caused by grain are reduced.

5.3 Removing Noise by Median Filtering:

Median filtering is similar to using an averaging filter, in that each output pixel is set to an average of the pixel values in the neighborhood of the corresponding input pixel. However, with median filtering, the value of an output pixel is determined by the *median* of the neighborhood pixels, rather than the mean. The median is much less sensitive than the mean to extreme values (called *outliers*). Median filtering is therefore better able to remove these outliers without reducing the sharpness of the image.

5.4 Removal of Noise by Mode Filtering:

Mode filtering is similar to using an median filter, in that each output pixel is set to an average of the pixel values in the neighborhood of the corresponding input pixel. However, with mode filtering, the value of an output pixel is determined by the *mode* of the neighborhood pixels, rather than the mean.

In this method we consider different neighbouring pixel values and then calculate their Mode.

CHAPTER 6

Non Linear Image Enhancement

Linear operations are exceptionally important in Image processing because they are based on a significant body of well-understood theoretical and practical results. Although Non Linear operations sometimes offer better performance, they are not always predictable, but in some cases they are of great importance.

Non linear image enhancement techniques are those which are applied on a specific part of the image. There are no specific techniques to implement Non linear image enhancement but we can use the linear image enhancement techniques for that specific part of the image.

Let H be an operator whose input and output are images, H is said to be a Non linear operator if it does not follow the following relation:

$$H(a*f + b*g) = a*H(f) + b*H(g)$$

Where f and g are 2 images, a and b are any two scalars.

By using Non linear image enhancement, loss of important information from a image can be easily avoided.

In some cases we can afford to have some noise in a image rather than loosing some of the vital information so we prefer Non linear techniques.

CHAPTER 7

METHODOLOGY

We have worked on pattern recognition of Braille language. We have developed an algorithm which helps to remove misalignment of Braille which occurs during manual scanning.

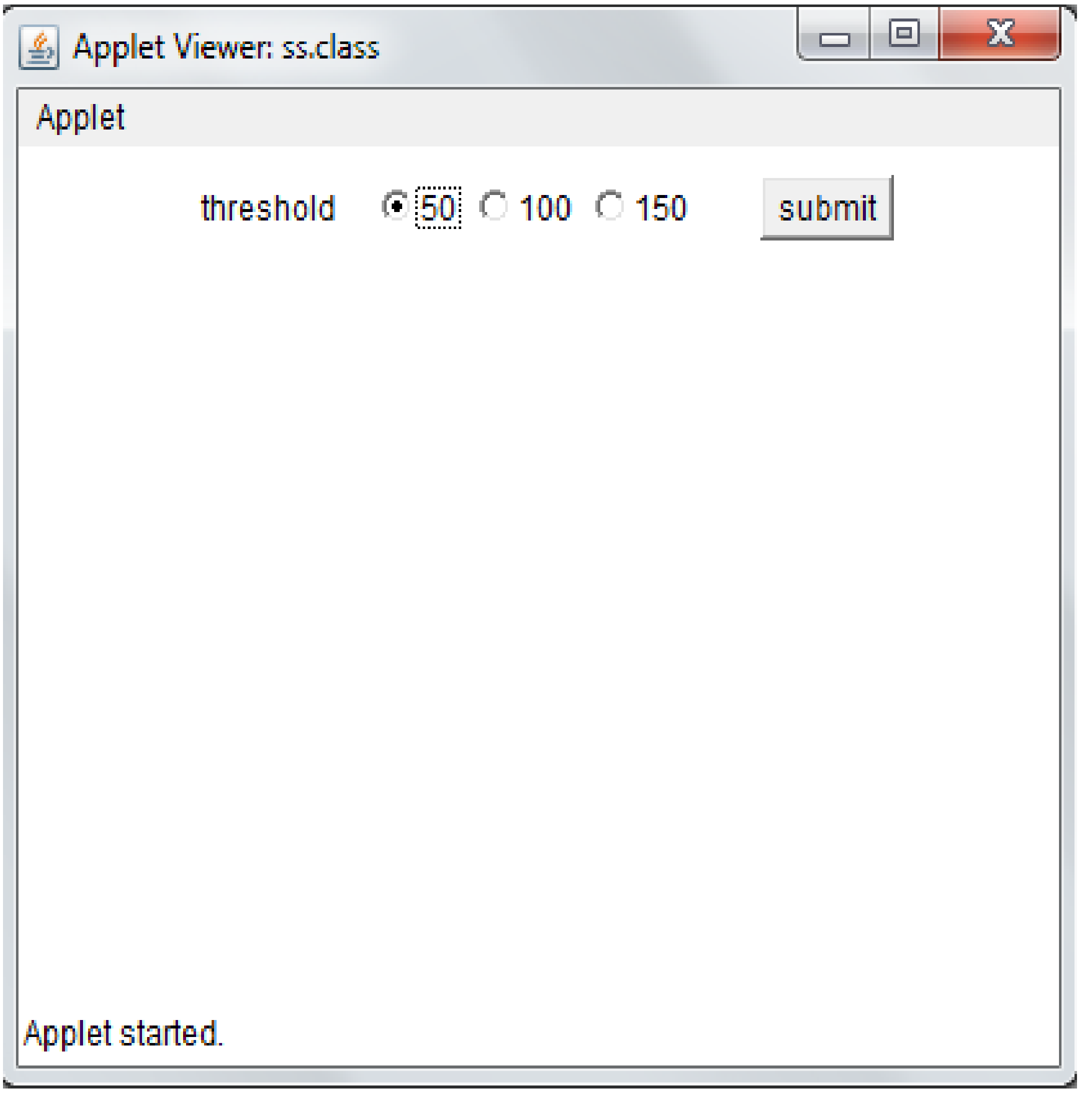
During our research we found various linear enhancement techniques and the main drawback of linear enhancement technique is that some vital information is lost because it works on that part of the image also that doesn't need to be enhanced. So to avoid this we chose to work on non linear methods.

So far

We have used a noisy image which is corrupted with 'Salt & Pepper' noise. We have first created an image containing 'Salt &Pepper' noise using MATLAB R2008a . We have further enhanced the image using java. We have developed an algorithm which enhances the image in a non linear way.

Image Library of Java has been used to read the image.

GUI (Graphic User Interface) has been designed in Java.



threshold ● 50 ○ 100 ○ 150

Applet started.

CHAPTER 8

ALGORITHM

Flow Chart

The following flow chart describes the algorithm we have used to Non Linearly Enhance an Image.

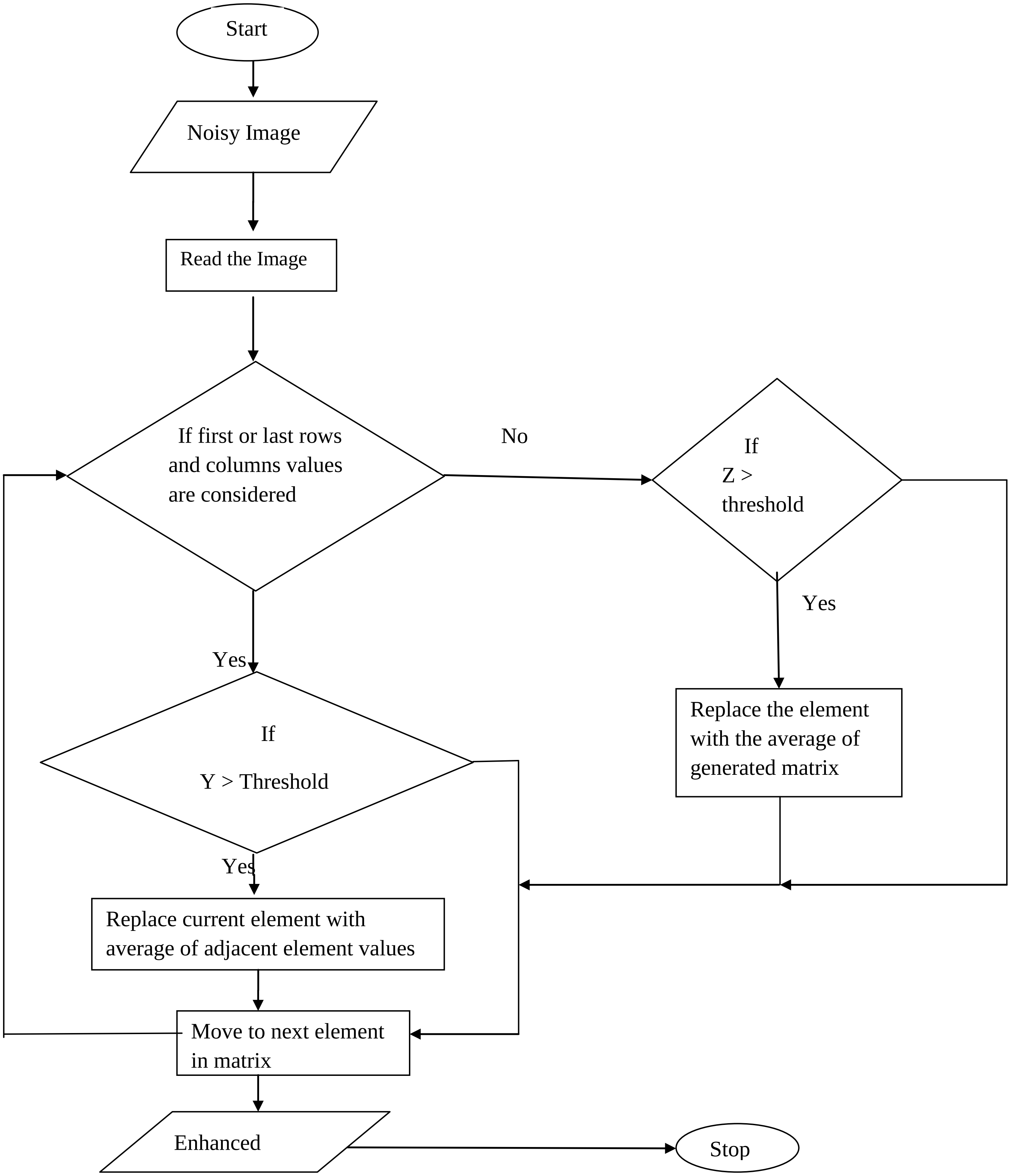
We read the gray image and generated a matrix of the same order as that of the image.

Terms used:

Y = difference of adjacent elements in the matrix.

Z = difference of the element with all the neighboring elements in the 3 X 3 matrix generated by considering the element to be processed as the center of the matrix.

Threshold = Input taken from the user.


Start
Noisy Image
Read the Image
If first or last rows and columns values are considered
No
If Z > threshold
Yes
Yes
If Y > Threshold
Replace the element with the average of generated matrix
Yes
Replace current element with average of adjacent element values
Move to next element in matrix
Enhanced
Stop

Algorithm

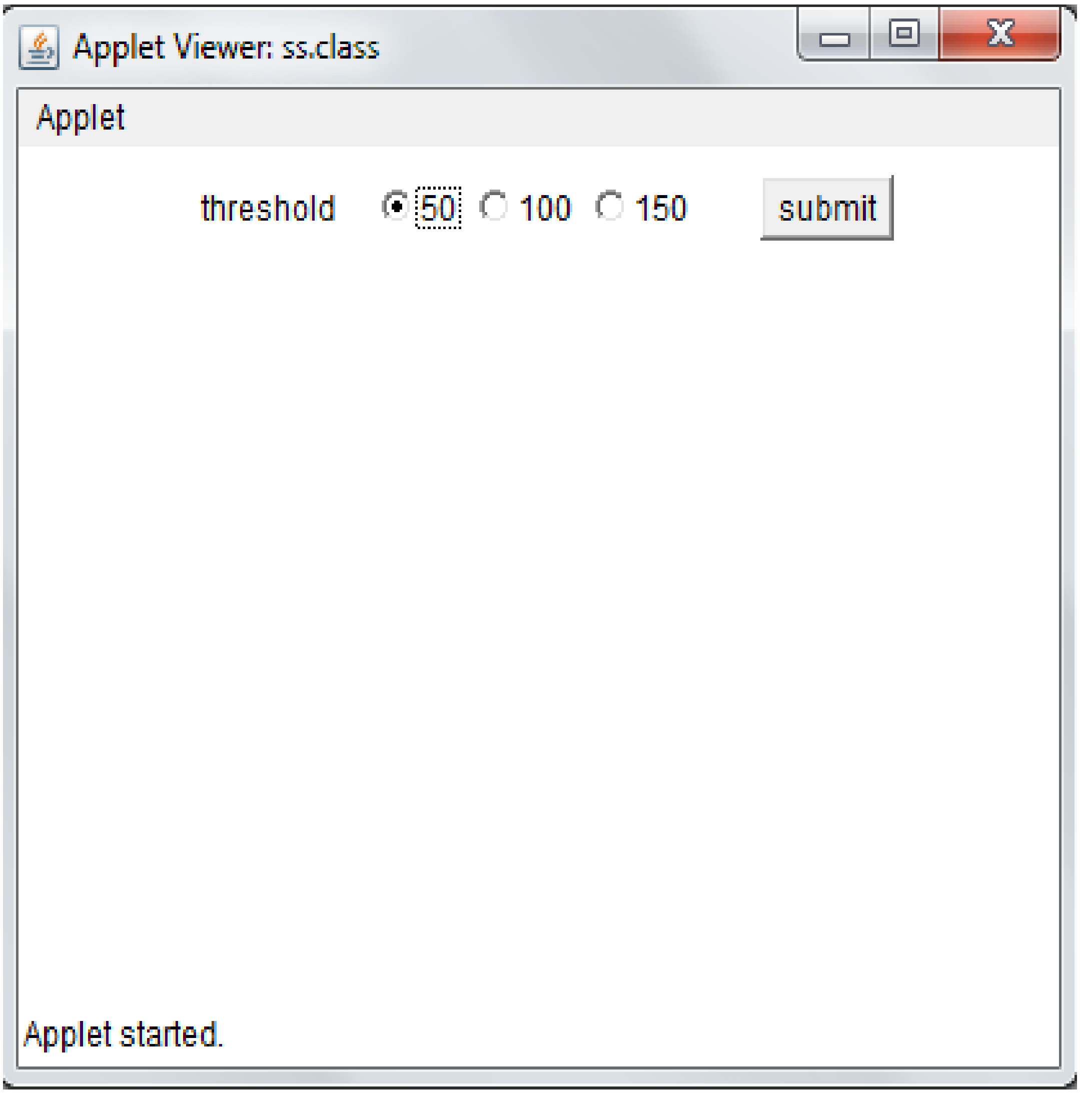
- We have designed a Graphic User Interface in JAVA.
- The interface allows user to choose the conditions according to his needs.
- First check for the first and last row’s and column’s.
- Take the difference of adjacent pixels and check the corresponding difference according to the given condition by the user.
- If the difference satisfies the given conditions then store the pixel location in an array.
- Apply mean filtering technique to that particular pixels only keeping rest of the image intact.
- Substitute the noise which is present in the original image by the values calculated by the mean filtering.
- Now consider the rest elements of the matrix , compare the respective element with every other element of a 3X3 matrix and if the difference from any other element is greater than user defined threshold .
- Replace the current element value with the average of all the elements of the 3X3 matrix.

Main advantage of this algorithm is we do not loose information of an image as pixels containing noise is only affected.

CHAPTER 9

CONCLUSION & RESULTS

Below we have shown the developed Graphic User Interface.



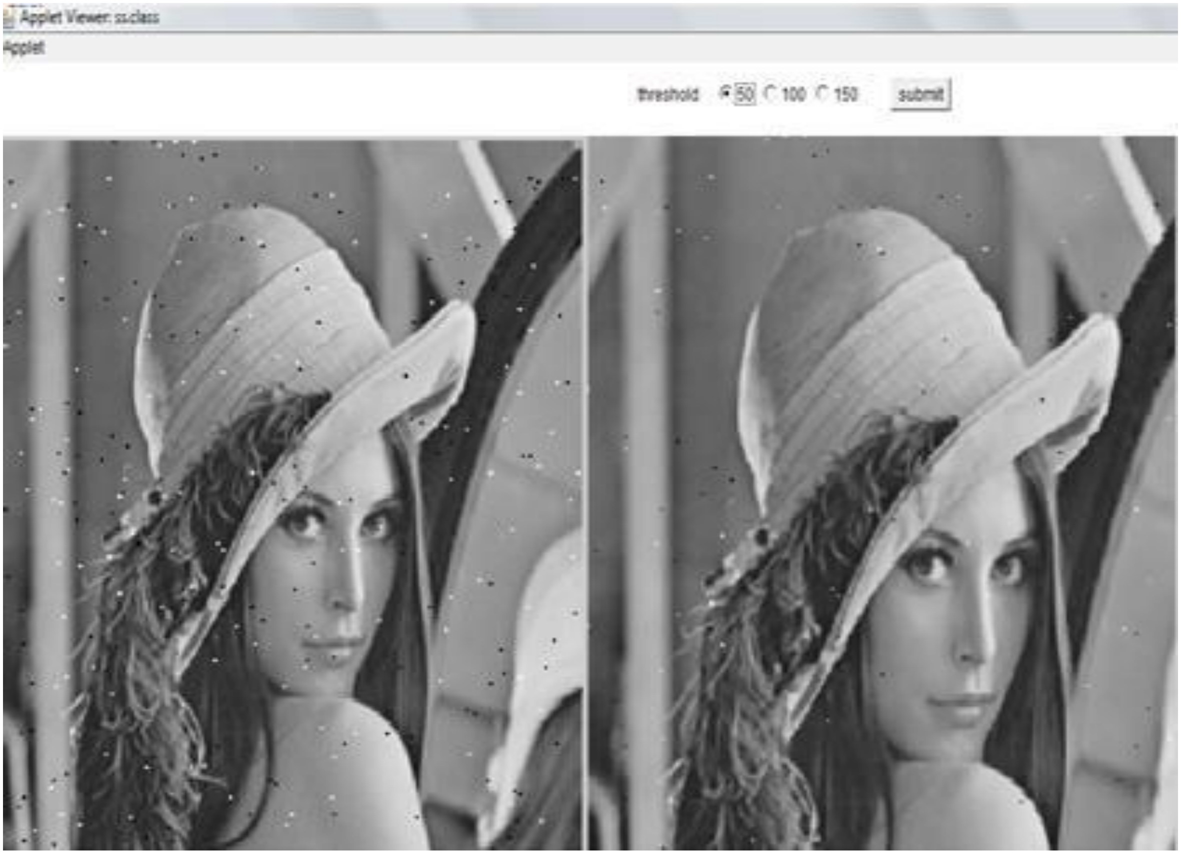
threshold ● 50 ○ 100 ○ 150

Applet started.

Further in GUI we compare the original image with the 2 enhanced images, one enhanced using median filter and the other one enhanced using our proposed algorithm. We are able to easily see the affect of our algorithm, as image obtained using our algorithm is better than the original image.

We have been successful in removing some extent of the noise (Salt & Pepper) but still some

noise is present in the image, but in the overall sense no important information has been lost.



threshold ● 50 ○ 100 ○ 150

ORIGNAL IMAGE

OUTPUT IMAGE 1

(Threshold=50)

References

- http://www.ece.rice.edu/~wakin/images/ [1]
- Digital Image Processing 2nd Edition –R Gonzalez , R Woods[2]
- http://www.cambridgeincolour.com/tutorials/gamma-correction.htm[3]
- http://www.cvip.uofl.edu
- http://www.mathworks.in
- Whiteboard Scanning and Image Enhancement

  -Zhengyou Zhang

  -Li-wei He

(Technical Report- MSR-TR-2003-39)