# *Implementation of Fuzzy Filters for Smoothing & Sharpening of Gray Images*

**A MAJOR THESIS**

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENT FOR THE DEGREE OF

**MASTER OF ENGINEERING**

IN

ELECTRONICS & COMMUNICATION

ENGINEERING

BY

**HIMANSHU GARG**

UNDER THE GUIDANCE OF

**Shri M. KULKARNI**

DEPARTMENT OF

ELECTRONICS & COMMUNICATION ENGINEERING

DELHI COLLEGE OF ENGINEERING

(UNIVERSITY OF DELHI), DELHI

Feb 2005

# CERTIFICATE

This is to certify that the thesis entitled "Implementation of Fuzzy Filters for Smoothing & Sharpening of Digital Images " being submitted by Himanshu Garg in the partial fulfillment of the requirement for the degree of Master of Engineering in Electronics & Communication in the Department of Electronics & Communication, Delhi College of Engineering, University of Delhi is a record of bonafide work done by him under my supervision and guidance. It is also certified that the dissertation has not been submitted elsewhere for any other degree.

(Shri M. Kularni)
Assistant Professor,
Deptt. of Electronics & Communication
Delhi College of Engineering
Delhi-110042

(Prof. A. BHATTACHARYYA)
Head of Department,
Deptt. of Electronics &Communication
Delhi College of Engineering
Delhi-110042

# ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without a mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

I am grateful to Dr. A.BHATTACHARYA (HOD, ECE deptt.) for providing us an opportunity to undertake this project. I am highly indebted to Mr. M.Kulkarni ( ECE deptt.), my project guide for taking keen interest in my work and for his constant monitoring and invaluable guidance and support throughout the course of our project. I profusely thank him for having patience to clear my doubts and channelise my efforts his cheerful disposition made my work all the more enjoyable.



HIMANSHU GARG

Roll No. 14/EC/02

Delhi College Of Engineering,

(University of Delhi), Delhi

# Abstract

Image smoothing is an important step in many image-processing applications. In the presence of noise, smoothing is an important pre-processing step if they are to be followed by other tasks such as edge detection, feature extraction and object / pattern recognition. The most effective approaches of image smoothing are non-linear and adaptive in nature. Noise smoothing and edge sharpening are inherently conflicting processes, since smoothing a region might destroy an edge and sharpening edges might lead to unnecessary noise.The type of algorithm to be used depends upon the objective to be achieved by the smoothing process as well as the particular application. The objective of the thesis is development of fuzzy filters, which are adaptive in nature and remove different noises like salt and pepper and Gaussian noise.

Most of the classical filters that remove noise simultaneously also blur the edges, but fuzzy filters have the ability to combine edge-preservation and smoothing. Compared to other non-linear techniques, fuzzy filter are able to represent knowledge in a comprehensible way.

# Contents

# List of tables

# Chapter 1

# Introduction

## 1.1 Introduction

Image smoothing is an important step in many image-processing applications.The problem of image smoothing can be stated as that of filtering out impulse noise and smoothing non-impulse noise or enhancing other salient structures present in the input image. Noise smoothing and edge detection are inherently conflicting processes, since smoothing a region might destroy an edge and sharpening edges might lead to unnecessary noise.

Noise smoothing can be viewed as replacing the gray-level of every pixel with a new value depending on the local context. Ideally, the smoothing algorithm should vary from pixel to pixel based on the local context. For example, if the local region is relatively smooth, then the new pixel value may be taken to be the average of local values. On the other hand, if the local region contains edge or non-impulse noise pixels, a different type of smoothing should be used to preserve edges. However, it is extremely hard, if not impossible, to set the conditions under which a certain filter should be selected. Since the local conditions can be evaluated only vaguely in some portions of an image, need arises for a filtering system to be capable of performing reasoning with vague and uncertain information.

There is uncertainty in many aspects of image processing and computer vision. Visual patterns are inherently ambiguous, image feature are corrupted and destroyed by the acquisition process, object definition are not always crisp, knowledge about the objects in the scene can be described only in vague terms, and output of low level processing provides vague, conflicting, or erroneous inputs to higher level algorithm. Fuzzy set theory and fuzzy logic are ideally suited for dealing with such uncertainty. For example, consider the following rule of thumb in image filtering for smoothing:

IF region is very noisy

THEN apply a large window-based smoothing operator.

Here, the antecedent clause is vague, and the consequent clause is fuzzy action that can be described only in imprecise terms. By constructing fuzzy rules in terms of condition – action relations can easily represent this type of knowledge.

Fuzzy sets offer a problem-solving tool for performing reasoning because they form the bridge between the precision of classical mathematics and the inherent imprecision of the real world. The imprecision of an image is contained within a grey or colour value to be handled using fuzzy sets. An image can be considered as an array of fuzzy singletons having membership values that denote the degree of some image property. For the purpose of image smoothing, we can take that property as noise distribution. Based on this property, we can develop the image-smoothing algorithm.

Image enhancement is usually one of the many task applied to an image in a computer vision task. The principal objective of the enhancement technique is to process a given image so that result is more suitable than original image for specific application. Typically we want enhancement process capable of removing noise, smoothing regions where gray level do not change significantly and sharpening abrupt gray level changes. It is however hard to incorporate all these requirements in a single framework, since smoothing a region might destroy a line or edges, and sharpening might lead to unnecessary noise. A good enhancement process is, therefore, required to be adaptive so that it can process each region differently based on region properties.

Since fuzzy logic can easily incorpoatre heuristic knowledge about a specific application in the form of the rules, it is ideally suited for building an image enhancement system. This leads to the development of a variety of image enhancement methods based on fuzzy logic.

The most effective approaches of image smoothing are non-linear and adaptive in nature so we are trying to develop an adaptive fuzzy filter to improve its performance for noise removal.

## 1.2 Objective of the Thesis

The objectives of this thesis are as follows:

1. To develop fuzzy filters, which can remove different types of noises like Salt and Pepper noise and Gaussian noise.
2. To make the fuzzy filters adaptive such that it can remove the noises based on the information obtained from the image.

### 1.3 Scope of the current work

It is well known that fuzzy filters have a more robust performance than classical filters. For example, most classical filters that remove noise simultaneously blur the edges, while fuzzy filters have the ability to combine edge-preservation and smoothing. Compared to other non-linear techniques, fuzzy filters are able to represent knowledge in a comprehensible way. Some fuzzy filters already exit in the literature, but we are looking for adaptive fuzzy filters, which can remove noises only from the image information itself.

We present three fuzzy filters: The first one removes the **Gaussian** noise based on a parameter '$\alpha$' which is fixed for each pixel for the entire image. In fact we require two values of '$\alpha$'; one for smoothing and other for sharpening of the image. $\alpha_{smooth}$ has a value greater than $\alpha_{sharp}$. We have taken largest possible value of $\alpha_{smooth}$ which is L-1 or 255.

The second fuzzy filter removes **Salt and Pepper** noise. This filter is extension of the work of the previous filter. In this we have made the parameter '$\alpha$' adaptive. The parameter is not a constant, but varies from pixel to pixel depending upon the image information. Two separate membership functions, one each for smoothing & sharpening has been defined.

Third filter removes both kind of noises. This filter has the ability to remove the noise based on three parameter (t, d, a), the parameter 'd' is estimated from the image information itself on the other hand 't' & 'a' are evaluated from the trial and error procedure. For removing salt and peeper noise we take negative value of 't' whereas for

Gaussian noise we take positive value of ‘t’ .The range of ‘t’ is quite broad so it is easy to set this parameter.The value of ‘a’ for Gaussian noise larger as compared to the salt and pepper noise.

## 1.4 Motivation

Image processing is an important field for the automated visual inspection. In many of applications, the acquired images must pass through a stage of image preprocessing in order to remove distracting and useless information from the images. So preprocessing techniques can play a very important role in increasing the accuracy of subsequent tasks such as parameter estimation and object recognition.

Image enhancement is an important step in many image-processing applications. In this respect, contrast enhancement is often necessary in order to highlight important features embedded in the image data. The enhancement of noisy image, however, is a very critical process because the sharpening operation can significantly increase the noise. Many fuzzy filters exist but our motive is to develop a fuzzy filter that is adaptive, so that it can remove the noise from the information of image itself.

## 1.5 Outline of the Thesis

In the next chapter, we present a literature review of the different classical filtering methods. In chapter 3, we have discussed about fuzzy logic, different approaches of fuzzy filtering like pure fuzzy filtering, fuzzy extension of existing filter and soft fusion of existing filter. In chapter 4, we present the method of development of fuzzy filter for gaussian noise. In chapter 5, we present the method of development of fuzzy filter for salt & pepper noise. In chapter 6, we have presented sigmoid filter for salt & pepper and gaussian noise. Finally conclusions of this thesis along with the suggestions for future work are given in Chapter 7.

# Chapter 2

# Review

In this chapter we present noise models followed by simple conventional filters, the noises include uniform noise, salt and pepper noise and Gaussian noise .The conventional filters include linear filters, rank filters and adaptive filter .in this Chapter we emphasize the need for fuzzy based noise filtering.

## 2.1 Noise models

There are two ways of image corruption by noise: noise addition and noise multiplication. A model of an image degraded by additive random noise is given by

$$g(x,y) = f(x,y) + n(x,y) \tag{2.1}$$

Where $n(x,y)$ represents the signal independent additive random noise. The level of noise is generally expressed by its variance. For example, if a color image is digitized with RGB values in the range (0,.. 255), additive Gaussian noise with variance, $\sigma_n^{\ 2}$ =1 would not be visible. A moderate noise level with $\sigma_n^{\ 2}$ = 100 makes an image grainy, while, $\sigma_n^{\ 2}$ =1000; the noise level obscures the image. Similar effects are observed with other noise distributions. In order to compare the performance of the original, degraded and processed images, some measures of error are necessary. The signal-to-noise ratio (SNR) is often used for the characterization of signal

$$SNR \ in \ dB = 10 \times \log_{10} \frac{\sigma_f^2}{\sigma_n^2} \tag{2.2}$$

Where $\sigma_f^{\ 2}$ and $\sigma_n^{\ 2}$ are the variances in the signal and noise respectively. The normalized mean square error (NMSE) between the original image $f(x, y)$ and the processed image $p(x, y)$ is defined as

$$NMSE[f(x,y), p(x,y)] = 100 \times \frac{\text{var}[f(x,y) - p(x,y)}{\text{var}[f(x,y)]} \quad (2.3)$$

Where var is the variance.

Digital images acquired with an electronic camera are typically corrupted with noise due to the optical system, light sensor and associated electronics. The transmission of video images is often accompanied by noise depending on environmental conditions. Video images transmitted via satellite are very susceptible to the electronic interference due to sunspot activity.

The classification of noise is based upon the shape of its probability density function (pdf). The mean and variance are important parameters to characterize the noise. Mean value $\overline{m}$, gives the average brightness of the noise and square root of variance $\sigma$ gives the average peak-to-peak gray level deviation of the noise. The mean and variance are defined as

$$\overline{m} = \sum_{k=0}^{g_{\max}} k \quad p_n(k) \quad (2.4)$$

$$\sigma^2 = \sum_{k=0}^{g_{\max}} (kp_n(k) - \overline{m})^2 \quad (2.5)$$

Where, $p_n(k)$ is the frequency of occurrence of noise amplitude, $k$. Ideally, $k$ varies from $-\infty$ to $+\infty$, however, since the pixel levels are limited in the range $[0, L\text{-}1]$, the noise amplitude level $k$ also lies in $[0, L\text{-}1]$. Now we want to describe different types of noise that we have used in our thesis.

### 2.1.1 Uniform noise

Uniform noise produces noise values with equal probability in the range from $g_{\max}$ to $g_{\min}$. The histogram of the uniform noise is

$$p_n(k) = \begin{cases} \dfrac{1}{g_{\max} - g_{\min}} \; ; \; g_{\min} < k < g_{\max} \\ 0 \; ; \; otherwise \end{cases} \tag{2.6}$$

The mean and standard daviation are computed from $g_{\max}$ and $g_{\min}$ as :

$$\overline{m} = \frac{g_{\max} + g_{\min}}{2} \tag{2.7}$$

$$\sigma = \frac{g_{\max} - g_{\min}}{\sqrt{12}} \tag{2.8}$$

### 2.1.2 Gaussian Noise

The most common type of noise that is found in an image is Gaussian noise, which is the result of many unknown noises from independent sources added together. Gaussian noise is expressed by the probability density function (pdf) as:

$$p_n(k) = \frac{1}{\sigma\sqrt{2\pi}} \; e^{-\frac{(k-\overline{m})^2}{\sigma^2}} \; ; \qquad -\infty < k < \infty \tag{2.9}$$

In the pdf, mean is located at the peak, having highest probability of occurrence and the width is determined by the standard deviation. Gaussian noise is defined over infinite range; however, the digitized image has finite range. Hence the noise values that exceed the gray level range are deposited at the 0 and $g_{max}$ points on the pdf. For 99.7% of the gray levels, the peak-to-peak gray level deviation is equal to $6\sigma$. For example, consider an image containing Gaussian noise with a mean gray level of 128 and a standard deviation of 10. For 99.7% of the pixels in this image, the peak-to-peak gray level deviation will be 60. This results in the image's gray levels varying between 98 and 158.

### 2.1.3 Impulse (salt and pepper) Noise

The probability density function (PDF) of (bipolar) impulse noise is given by

$$p(z) = \begin{cases} p_a & \text{for } z=a \\ p_b & \text{for } z=b \\ 0 & \text{otherwise} \end{cases} \quad (2.10)$$

In the contents of image if $b>a$, gray level b will appear as light dot in the image. Conversely, level $a$ will appear dark dot in the image. If either $P_a$ or $P_b$ is zero, the impulse noise is called unipolar. If neither probability is zero, especially if $a$ and $b$ are approximately equal, impulse noise will resemble salt and pepper granules randomly distributed over the image .For this reason, bipolar impulse noise is also called salt-and-pepper noise. Shot and spike noises also refer to this type of noise. Noise impulse can be positive or negative.

## 2.2 Conventional Filters

Filters are mainly used to suppress either the high frequencies in the image, *i.e., for* smoothing the image, or the low frequencies, *i.e.,* for enhancing or detecting edges in the image.

Suppose that an image-processing operator *F* acts on the two input images A and *B* and produces output images *C* and *D* respectively. If the operator *F* is ***linear,*** then

$F(a \times A + b \times B) = a \times C + b \times D$

Where *a* and *b* are constants. This means that each pixel in the output of a *linear* operator is the weighted sum of a set of pixels in the input image.

For example, the threshold operator is *non-linear*, because individually, corresponding pixels in the two images A and B may be below the threshold, whereas the pixel obtained by adding A and B may be above threshold. Similarly, the absolute value operation is non-linear:

$|-1+1| \sim= |-1|+|1|$

### 2.2.1 Linear Filters

The idea of mean filtering is simply to replace each pixel value in an image with the mean (`average') value of its neighbors, including itself. This has the effect of eliminating pixel values, which are unrepresentative of their surroundings. Often a 3×3 square kernel shown in Figure 2.1 is used, although larger kernels (*e.g.*, 5×5 squares) can be used for more severe smoothing. ( a small kernel can be applied more than once in order to produce a similar ,but not identical effect as a single pass produces with a large kernel)

| | | |
|---|---|---|
| $\frac{1}{9}$ | $\frac{1}{9}$ | $\frac{1}{9}$ |
| $\frac{1}{9}$ | $\frac{1}{9}$ | $\frac{1}{9}$ |
| $\frac{1}{9}$ | $\frac{1}{9}$ | $\frac{1}{9}$ |

**Figure 2.1: 3×3 averaging kernel used in mean filtering**

Computing the straightforward convolution of an image with this kernel performs the mean filtering that is most commonly used as a simple method for reducing noise in an image.

There are two main problems with mean filtering, which are:

- A single pixel with a very unrepresentative value can significantly affect the mean value of all the pixels in its neighborhood.
- When the filter neighborhood straddles an edge, the filter will interpolate new values for pixels on the edge thereby blurring the edge. This may be a problem if sharp edges are required in the output.

### 2.2.2 Rank filters

For a linear filter, the output is a linear combination (i.e., weighted average) of neighboring pixels. Problem with linear filter is that edges are blurred by its application. To improve upon them **non-linear** filters are used.

The rank filters are a class of non-linear filters. With a rank filter operating in a window about a pixel ij , the N pixel within the neighborhood is ranked as :

$$g^1_{ij} \leq g^2_{ij} \leq \dots \leq g^N_{ij}$$

Then ,the output value is chosen.

$$f_{ij} = R_k ( g_{ij} )$$

Where g is the input image, f is the output image and $k_{th}$ rank value in the window is chosen.

Three special rank filters are the $min_n$ , $max_n$ and median filters.

$$min_n (g) = R_1(g)$$

$$max (g) = R_N(g)$$

$$median(g) = R_{\lceil N/2 \rceil}(g)$$

The net effect of three rank filters is to reduce the variance in the image. Well-known properties of median filters are :

- They preserve sharp edges
- They eliminate spike (salt and pepper) noise

### 2.2.2.1 Median filter

This involves sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value so as to calculate the median. If the neighborhood under consideration contains an even number of pixels, the average of the two middle pixel values is used. Figure 2.2 illustrates calculation of median value.

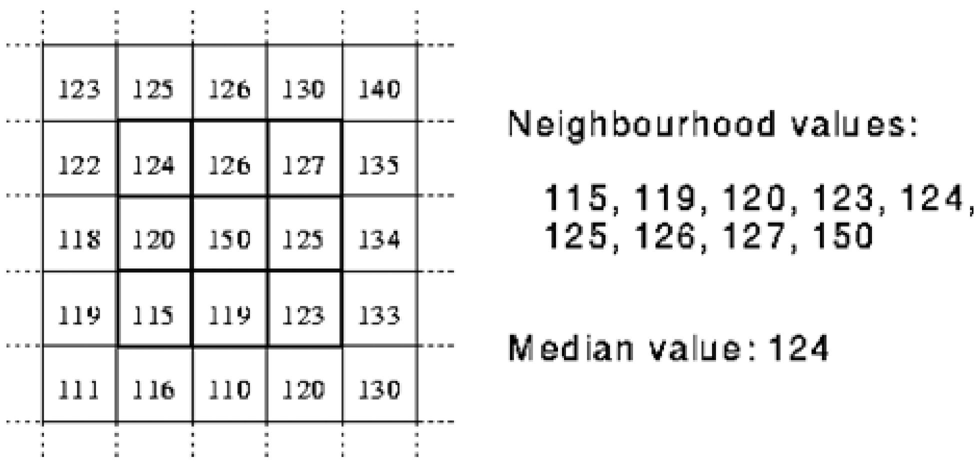



**Figure 2.2 : The median value of a pixel neighborhood.**

As can be seen, the central pixel value of 150 is rather unrepresentative of the surrounding pixels and is replaced with the median value: 124. A 3×3 square neighborhood is used here , however ,larger neighborhoods will produce more severe smoothing.

By calculating the median value of a neighborhood rather than the mean filter, the median filter has two main advantages over the mean filter:

- The median is more robust than the mean and so a single very unrepresentative pixel in a neighborhood will not affect the median value significantly.
- Since the median value must actually be the value of one of the pixels in the neighborhood, the median filter does not create new unrealistic pixel values when

the filter straddles an edge. For this reason the median filter is much better at preserving sharp edges than the mean filter.

In General, the median filter allows a great deal of high spatial frequency detail to pass while remaining very effective at removing noise on images where less than half of the pixels in a smoothing neighborhood have been effected. (As a consequence of this, median filtering can be less effective at removing noise from images corrupted with Gaussian noise)

Unlike the mean filter, the median filter is non-linear. This means that for two images A (x) and B (x) ,we have

$$\text{Median} |A(x)+B(x)| \neq \text{Median} |A(x)| + \text{Median}|B(x)|$$

### 2.2.3 Adaptive filters

Once selected, the filters are applied on image without regard for how the image characteristics vary from point to point another. The adaptive filter whose behavior changes is based on statistical characteristic of image inside the filter region defined by the $m \times n$ rectangular window. The price paid for improved filtering power is an increase in filter complexity.

The simplest statistical measures of a random variable are its mean and variance. These are reasonable parameters on which to base an adaptive filter because they are quantities closely related to appearance of an image. The mean gives a measure of average gray level in the region over which the mean is computed, and the variance gives measures of average contrast in that region.

# Chapter 3

# Fuzzy Logic

## 3.1 Fuzzy set theory

Fuzzy set theory is the extension of conventional (crisp) set theory .It handles the concept of partial truth (truth values between 1 (completely true) and 0 (completely false)). It was introduced by Prof**. Lotfi A. Zadeh** of UC/Berkeley in 1965 as a means to model the vagueness and ambiguity in complex systems.

The idea of fuzzy sets is simple and natural. For instance, we want to define a set of gray levels that share the property dark. In classical set theory, we have to determine a threshold, say the gray level 100. All gray levels between 0 and 100 are elements of this set; the others do not belong to the set . But the darkness is a matter of degree. So, a fuzzy set can model this property much better. To define this set, we also need two thresholds, say gray levels 50 and 150. All gray levels that are less than 50 are the full members of the set, all gray levels that are greater than 150 are not the member of the set. The gray levels between 50 and 150, however, have a partial membership in the set (right image in Figure 3.1)

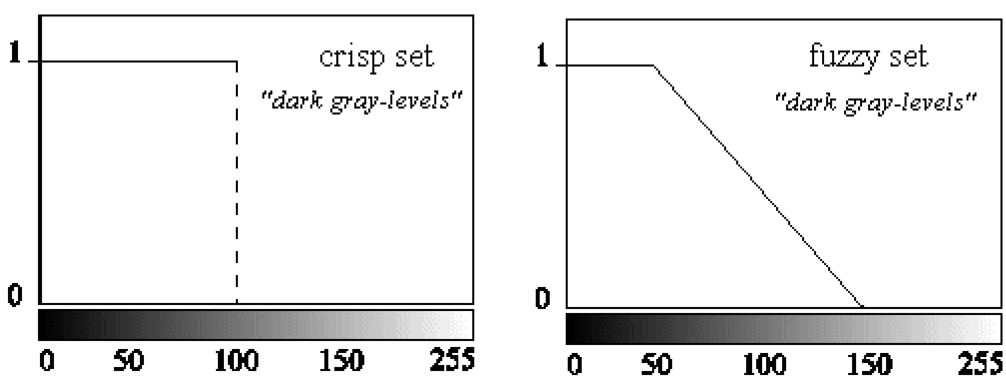



**Figure 3.1. : Representation of "dark gray-levels" with a crisp and a fuzzy set**

## 3.2 Fuzzy Image Processing

It is a collection of different fuzzy approaches to image processing.

Fuzzy image processing has three main stages:

- Image fuzzification
- Modification of membership values
- if necessary, image defuzzification.

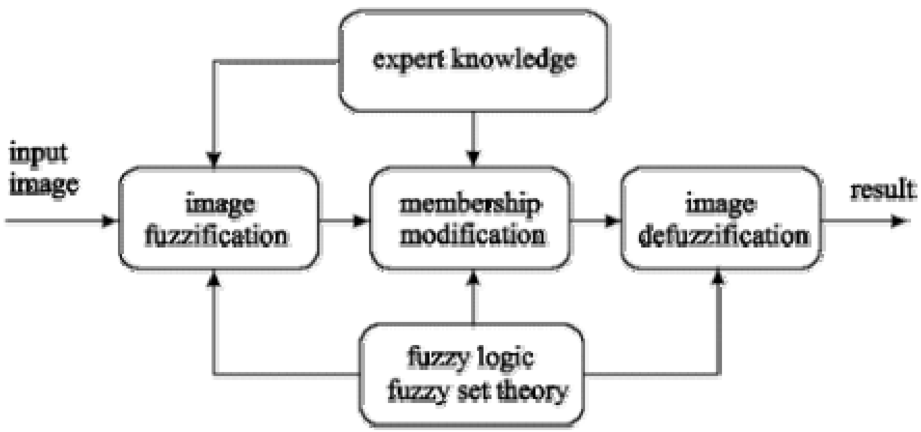



**Figure 3.2 The general structure of fuzzy image processing**

The main power of fuzzy image processing is in the middle step (modification of membership values). After the image data are transformed from gray-level plane to the membership plane (fuzzification), appropriate fuzzy techniques modify the membership values. This can be a fuzzy clustering; a fuzzy rule-based approach, a fuzzy integration approach and so on. These steps are shown in Figure 3.3 .

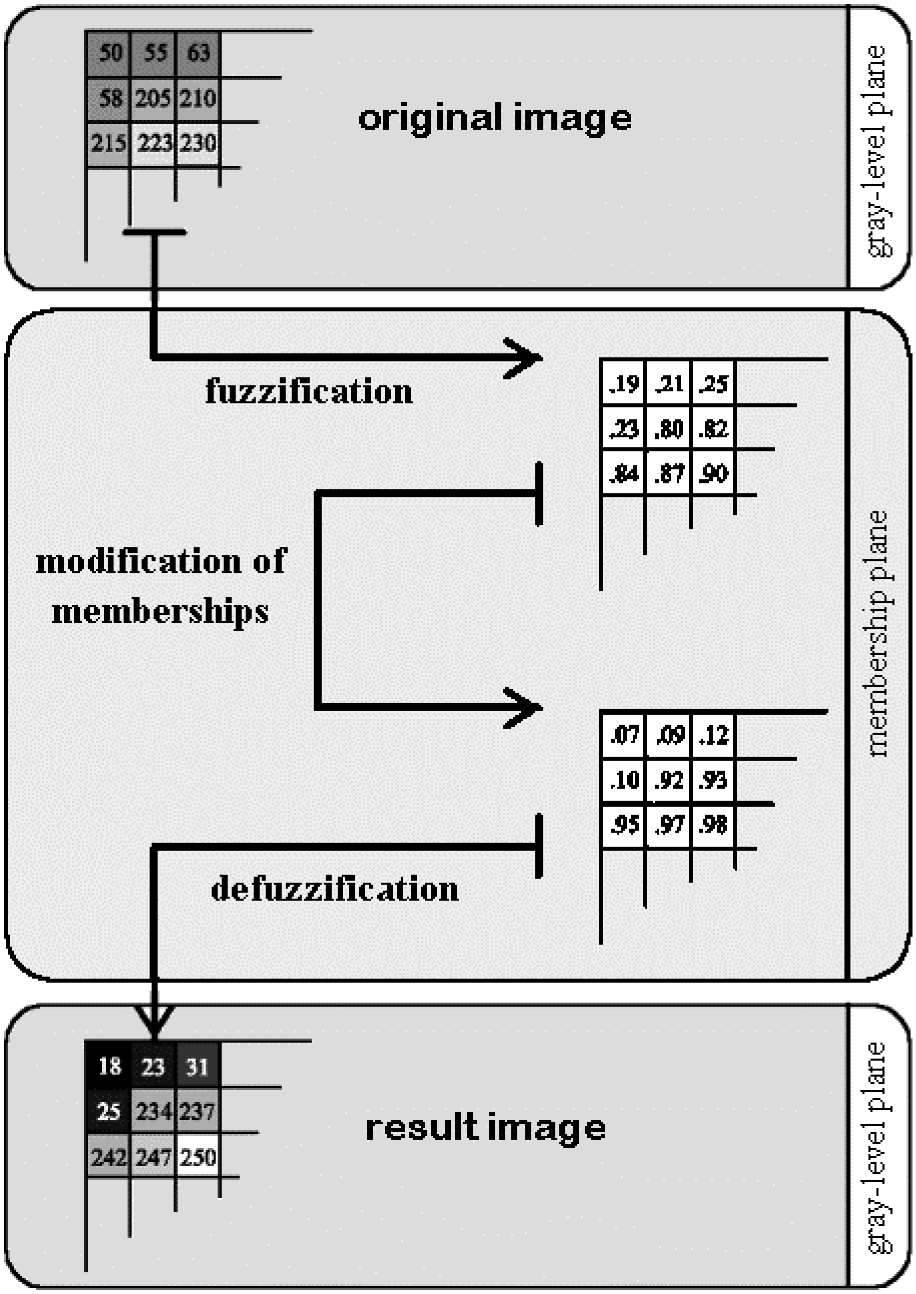



**Figure 3.3: Steps in fuzzy image processing**

## 3.3 Why Fuzzy Image Processing

Why we should use fuzzy techniques in image processing? There are many reasons to do this. The most important of them are as follows:

- Fuzzy techniques are powerful tools for knowledge representation and processing

- Fuzzy techniques deal with vagueness and ambiguity efficiently

In many image-processing applications, we have to use expert knowledge to overcome the difficulties. Fuzzy set theory and fuzzy logic offer us powerful tools to represent and process human knowledge in the form of fuzzy if-then rules. On the other side, many difficulties in image processing arise because the data/results are uncertain. This uncertainty, however, is not always due to the randomness but to the ambiguity and vagueness. Beside randomness, which can be managed by probability theory, we can distinguish between three other kinds of imperfection in the image processing

- Geometrical fuzziness
- Vague (complex/ill-defined) knowledge

The main difference to other methodologies in image processing is that input data X (histogram, gray level, features,…) will be processed in the so – called membership plane where one can use great diversity of fuzzy logic ,fuzzy set theory and fuzzy measure theory to modify/aggregate the membership value ,classify data , or make decision using fuzzy inference. The new membership values are retransformed in the gray level plane to generate new histogram, modify gray levels, image segments, or class objects.

## 3.4 Fuzzy Filtering

Image filtering is necessary to enhance edges and / or suppress noise. Various classical techniques already exist in the literature. The main problem of filtering is the dilemma of concurring image properties, such as sharpness and smoothness, respectively. While removing noise from the image, the fine details are also usually filtered out. On the other hand, enhancing the edges and fine structures, the noise will also be amplified. A widespread solution to this dilemma is to use adaptive filtering. Here, the image pixels $x_j$ in each $n$x$n$ neighborhood ($n$ is an odd number) are assigned with the suitable weights $w_j$. The outcome of the filtering can be achieved by:

$$y = \sum_j w_j . x_j , \quad with \quad \sum_j w_j = 1 \tag{3.1}$$

Where $j$ = 1,2, … $n$. The weights [8] are chosen so as to get the desired filtering effect. The performance of these techniques depends on the correctness of weights [9].

In the recent years, many fuzzy approaches to image filtering have been proposed .One can distinguish between the following approaches:

- Pure fuzzy filtering: uses only fuzzy if-then rules,
- Fuzzy extension of existing algorithm: use membership function or fuzzy rules to extend the classical filter to fuzzy set, and
- Fuzzy fusion technique: aggregate the result of different filters to combine their Advantages.

### 3.4.1 Pure fuzzy filtering

Pure fuzzy filters are mainly based on fuzzy if – then rules, where the desired filtering effect can be achieved using a suitable set of linguistic rules. Russo proposed FIRE (Fuzzy Inference Ruled by Else- action) operators for image filtering. A fuzzy smoother, for example, can be designed using luminance differences between the center pixel of a neighborhood and its surrounding pixels. Fuzzy rules for smoothing are of the following (general) form:

If (a pixel is d*arker* than its neighboring pixels), Then (make it *brighter* ),

If (a pixel is *brighter* than its neighboring pixels),Then (make it *darker*),

Else (leave it unchanged)

### 3.4.2 Fuzzy Extension of Existing Filters

Another possibility to apply the concept of fuzziness to image filtering is the extension of existing filters. The extension sometimes make use of a simple membership function that substitutes a threshold function, or applies the fuzzy rules to adapt the parameters of the corresponding filter. In the following, we will see some possible ways of extending gaussian, median and mean filters.

### 3.4.3 Soft Fusion of Existing Filter

The filtering procedure has to fulfill different requirements. For instance, the image should be smoothed without loss of fine details .In spite of all approaches to image filtering, it is not possible to develop a super filter that solves all conflicts in all possible situation (each filter has its own advantage and disadvantage). Therefore, it is often more appropriate to combine the existing filters using their advantages, and simultaneously, excluding their shortcomings. Fuzzy technique enable us implement the fusion in a robust way because fusion decision is possible. This property is fundamental since in real images it is nearly impossible to find a crisp answer to the question whether a neighborhood is edgy, noisy or smooth. Choi and Krishnapuram proposed fusion technique with fuzzy if then rules using a degree of compatibility of the center pixel to the neighbors:

If the compatibility is small (outlier), then use the filter $F_1$,

If the compatibility is medium (edge), then use the filter $F_2$,

If the compatibility is large (smooth), then use the filter $F_3$.

The filters $F_1$ (outlier filter),$F_2$ (edge sharpening) and $F_3$ (smoothing) can be selected regarding to specific requirements of the actual application .

From the above discussion, it is clear that an adaptive filter should also address the conflicting goals:

- removing impulse noise,
- smoothing out non-implulse noise,
- enhancing edge features.

Fuzzy techniques offer a new and flexible framework for the development of image enhancement algorithms. They are non-linear, knowledge based and robust. The potential of fuzzy set theory with respect to image enhancement are still investigated as well as other established methodologies.

# Chapter 4

## Fuzzy Filter to remove Gaussian Noise Combining Sharpening and Noise Reduction

### 4.1 Introduction

Improving the quality of sensor data is a key issue in image based instrumentation. Indeed, preprocessing techniques can play a very relevant role in increasing the accuracy of subsequent tasks such as parameter estimation and object recognition. In this respect, contrast enhancement is often necessary in order to highlight important features embedded in the image data. The enhancement of noisy data, however, is a very critical process because the sharpening operation can significantly increase the noise.

In this chapter, a nonlinear method for the enhancement of noisy data is presented. The proposed approach consists in a multiple output processing system that adopts fuzzy networks in order to combine contrast enhancement and noise reduction. Indeed, the fuzzy paradigm is well suited to address conflicting tasks such as detail sharpening and noise smoothing. In addition the multiple-output structure increases the overall performance of the fuzzy processing because the operation can repeatedly be applied to the image data.

Key features of the proposed technique are good performance in the enhancement of mages corrupted by Gaussian noise without using complicated tuning of fuzzy set parameters. In fact, the overall nonlinear behavior of the enhancement system is very easily controlled by one parameter only.

### 4.2 Filter Design

Fuzzy networks have been shown to be very effective for image processing applications such as noise removal. For the sake of clarity, the basic operation of a fuzzy network is explained in details.

### 4.2.1 Image Smoothing

Let us consider a digitized image having L gray levels. Let $X_{i,j}$ be the pixel luminance at location [i,j] , $(0 <= X_{i,j} <= L - 1)$ , as represented in Fig. 4.1.The noise amplitude estimate is computed by considering (fuzzy) relations between the central pixel and its

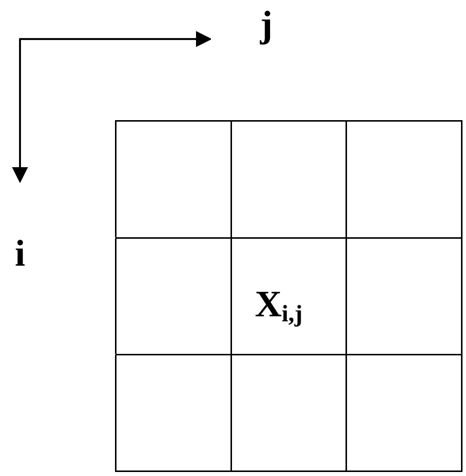



Figure 4.1: Showing $3\times3$ window with pixel to be processed

neighbors. For the sake of simplicity, let **A** denote the set of neighboring pixel. The output is yielded by the following relationship:

$$\Delta X_{i,j} = \left(\frac{K_1\,\alpha_{smooth}}{N}\right)\left[\sum_{X_{m,n}\,\varepsilon\,A} \mu_{R1}\left(X_{i,j}, X_{m,n}, \alpha_{smooth}\right) - \sum_{X_{m,n}\,\varepsilon\,A} \mu_{R2}\left(X_{i,j}, X_{m,n}, \alpha_{smooth}\right)\right] \quad ....(4.1)$$

Where **$K_1$=1** & **Rq (q = 1,2)** represents the class of fuzzy relations described by the parameterized membership function.

$$\mu_{Rq}(u,v,\alpha) = \begin{cases} MAX\{\,1-\,(|u-v-\alpha|)\,/\,(2*\alpha)\;,\,0\,\} & ;\,q=1 \\ MAX\{\,1-\,(|u-v+\alpha|)\,/\,(2*\alpha)\;,\,0\,\} & ;\,q=2 \end{cases} \quad ....(4.2)$$

It should be observed that, by varying the value of $\alpha_{smooth}$ ($0 < \alpha_{smooth} <= L-1$), different nonlinear behaviors can be obtained. For example, let us focus on fuzzy relation $R_1$. When $\alpha_{smooth}$ is large, the fuzzy relation represents "u is much larger than v".

Conversely, when $\alpha_{smooth}$ is small, the fuzzy relation becomes “u is close to v.” The (possibly) noise-free value $Y_{i,j}$ of the pixel luminance at location [i,j] is then obtained by subtracting the noise estimate $\Delta X_{i,j}$ from the original pixel luminance $X_{i,j.}$

$$y_{i,j} = x_{i,j} - \Delta x_{i,j} \qquad \text{....(4.3)}$$

The processing defined by (1)–(3) performs data smoothing and is typically applied to a noisy input image. Large values of $\alpha_{smooth}$ increase the noise cancellation at the price of a possible increase of the detail blur. The optimal choice depends on the amount of noise corruption and is typically a tradeoff between noise removal and detail preservation.

### 4.2.2 Image Sharpening

Now, let us consider a noise-free image. A sharpening effect can easily be implemented by using the same fuzzy network operations defined by (1) and (2). In this case, we choose $K_2 = 1$, $\alpha_{sharp} = L - 1$, and we add the corresponding network output $\Delta x'_{i,j}$ to the original pixel luminance $x_{i,j.}$

$$y_{i,j} = x_{i,j} \oplus \Delta x'_{i,j} \qquad (4.4)$$

where symbol $\oplus$ denotes the bounded sum: $a \oplus b = \min\{a + b, L - 1\}$.In fact, we can think of the sharpening effect as the opposite of the smoothing action. Notice that the bounded sum is formally required in order to limit the output value as follows:

$$y_{i,j} <= L - 1$$

The parameter settings are based on a heuristic approach.

If the input image is noisy, we can combine a sharpening and a smoothing network in the same processing system. The former aims at increasing the luminance difference between the central pixel and its neighborhood, while the latter aims at reducing the noise increase.

$$y_{i,j} = x_{i,j} \oplus (\Delta x'_{i,j} - \Delta x_{i,j}) \qquad \text{....(4.5)}$$

An appropriate choice of in the smoothing network permits us to remove noise in the uniform regions of the image, where the effect is more annoying from the point of view of the human perception.

## 4.3 Combination of Fuzzy Networks

More fuzzy networks can be adopted in the same structure in order to increase the enhancement effect. The proposed multiple-output system includes three fuzzy networks and deals with a 4x 4 neighborhood, as shown in Figs. 2–4. Each processing step involves two different operations dealing with an appropriate choice of pixel patterns. The first operation performs smoothing only. This operation evaluates the output $Y_{i,j}$ by processing the luminances in the pixel pattern A (Fig. 3). This processing is performed by fuzzy network 1 (Fig. 4). According to the mechanism described in the previous section, the output $Y_{i,j}$ is given by the following relationship:

$$Y_{i,j} = X_{i,j} - (\Delta X_{i,j})^{A} \qquad \text{….(4.6)}$$

$$(\Delta X_{i,j})^{A} = \left(\frac{K_1 \, \alpha_{smooth}}{N}\right) \left[ \sum_{X_{m,n} \, \varepsilon \, A} \mu_{R1}\left(X_{i,j}, X_{m,n}, \alpha_{smooth}\right) - \sum_{X_{m,n} \, \varepsilon \, A} \mu_{R2}\left(X_{i,j}, X_{m,n}, \alpha_{smooth}\right) \right]$$

….(4.7)

The processing is recursive, i.e., the new value $Y_{i,j}$ is immediately assigned to $X_{i,j}$ and reused for further processing.

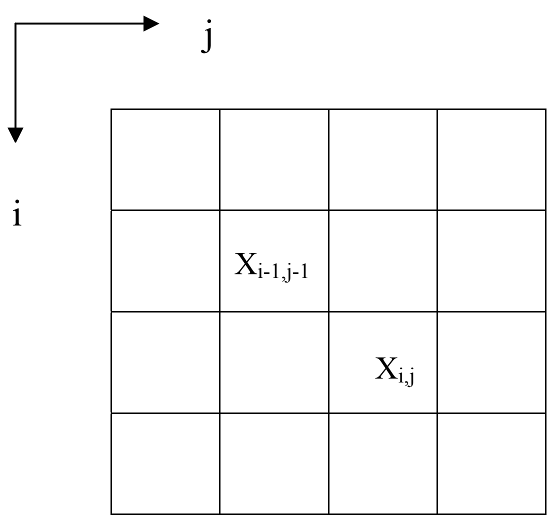



Figure 4.2: 4X 4 window

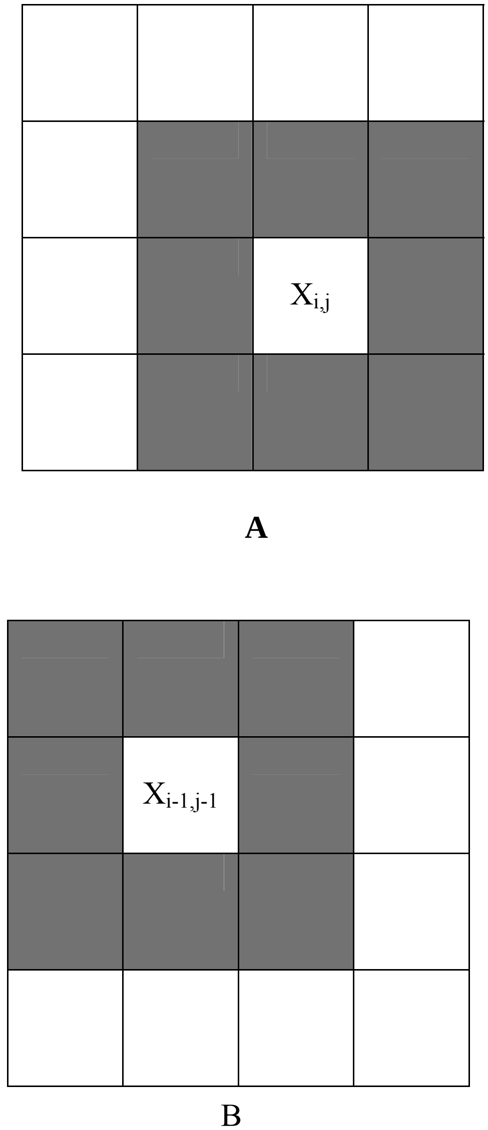



Fig. 4.3 Different pixel patterns for the multiple-output processing

The second operation is performed by fuzzy networks 2 & 3. It evaluates the output $Y_{i-1,j-1}$ by processing the luminances in the pixel pattern B (see Fig. 3). It should be observed that these luminances represent the results of the first operation because the processing is recursive and the window scans the image from left to right and from top to bottom. Since the second operation acts on prefiltered data, the effectiveness of the image enhancement process increases. The output is evaluated by the following relations:

$$y_{i-1,j-1} = x_{i-1,j-1} \oplus \left( (\Delta z_{i-1,j-1})^{B2} - (\Delta x_{i-1,j-1})^{B1} \right) \quad \text{....(4.8)}$$

$$\left(\Delta X_{i-1,j-1}\right)^{B} = \left(\frac{K_1\,\alpha_{smooth}}{N}\right)\left[\sum_{X_{m,n}\,\varepsilon\,B}\mu_{R1}\left(X_{i-1,j-1},X_{m,n},\alpha_{smooth}\right)-\sum_{X_{m,n}\,\varepsilon\,B}\mu_{R2}\left(X_{i-1,j-1},X_{m,n},\alpha_{smooth}\right)\right]$$

....(4.9)

$$\left(\Delta Z_{i-1,j-1}\right)^{B} = \left(\frac{K_1\,\alpha_{sharp}}{N}\right)\left[\sum_{X_{m,n}\,\varepsilon\,B}\mu_{R1}\left(X_{i-1,j-1},X_{m,n},\alpha_{sharp}\right)-\sum_{X_{m,n}\,\varepsilon\,B}\mu_{R2}\left(X_{i-1,j-1},X_{m,n},\alpha_{sharp}\right)\right]$$

4.10)

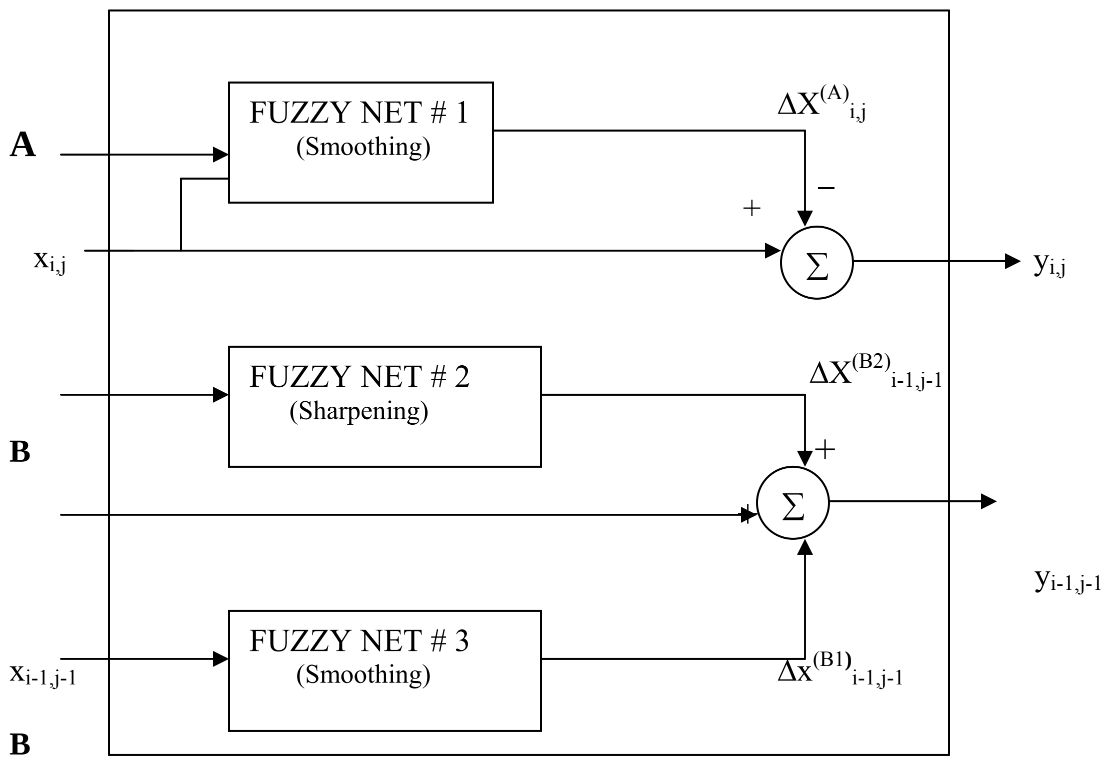


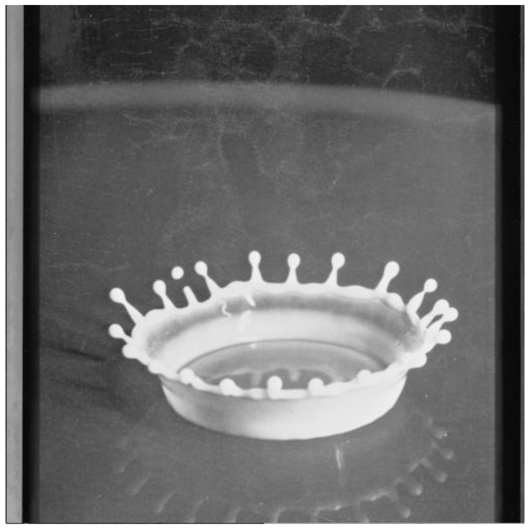

Fig. 4.4 Block diagram of multiple output system.

It is worth pointing out that the nonlinear behavior of the overall system is easily controlled by the value of parameter $\alpha$ only ($0 < \alpha <= L-1$). In fact, according to (10), the strength of the sharpening action is assigned. On the contrary, the effectiveness of the smoothing effect depends on [see relations (8) and (9)]. A large value increases the noise removal. A small value decreases this effect.

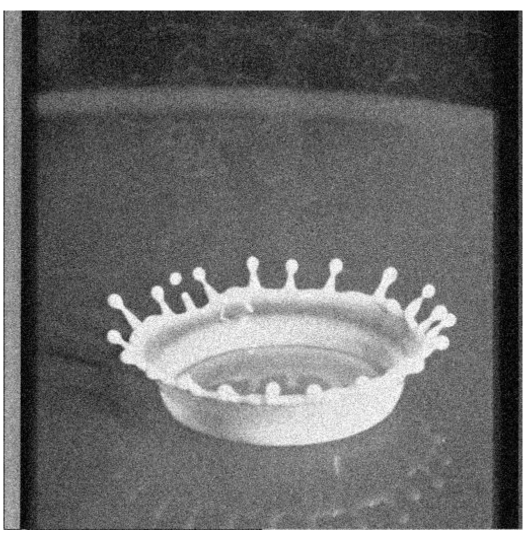
## 4.4 Results And Analysis

### 4.4.1 Result set 1

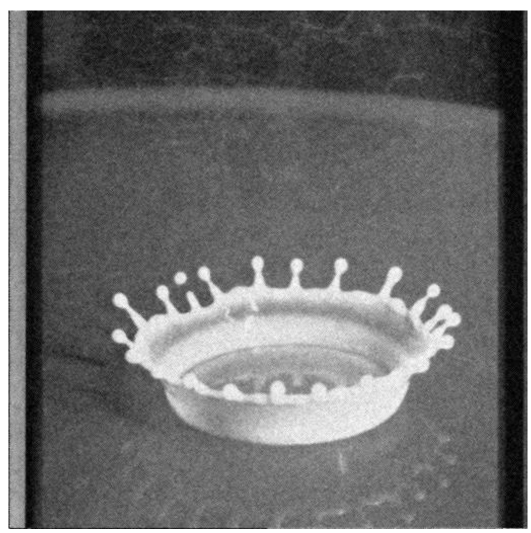
The filter was tested with varying levels of Gaussian noise for different images. The results are listed below.



(a) Original image

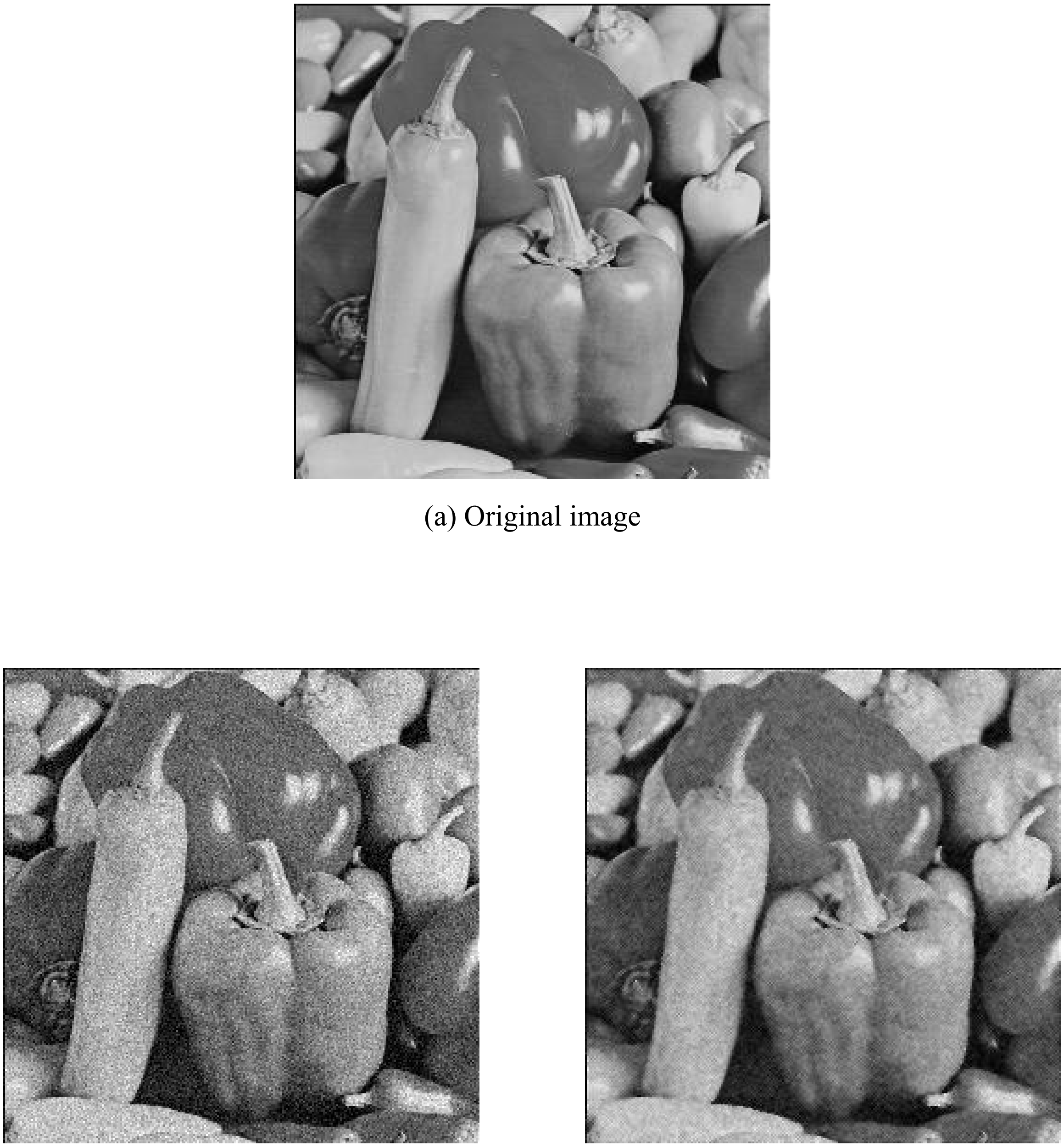


(b) Noisy image



(c) Corrected image

Fig 4.5: (a) Original Ashtray Image (b) Image corrupted with mean = 0 and variance = 0.005 Gaussian noise (c) Image Corrected by the proposed method



(a) Original image

(b) Noisy image

(c) Corrected image

Fig 4.6: (a) Original Pepper Image (b) Image corrupted with mean = 0 and variance = 0.006 Gaussian noise (c) Image Corrected by the proposed method

(a) Original image

(b) Noisy image

(c) Corrected image

Fig 4.7: (a) Original Image (b) Image corrupted with mean = 0 and variance = 0.004 Gaussian noise (c) Image Corrected by the proposed method

(a) Original image

(b) Noisy image

(c) Corrected image

Fig 4.8: (a) Original Boat Image (b) Image corrupted with mean = 0 and variance = 0.006 Gaussian noise (c) Image Corrected by the proposed method

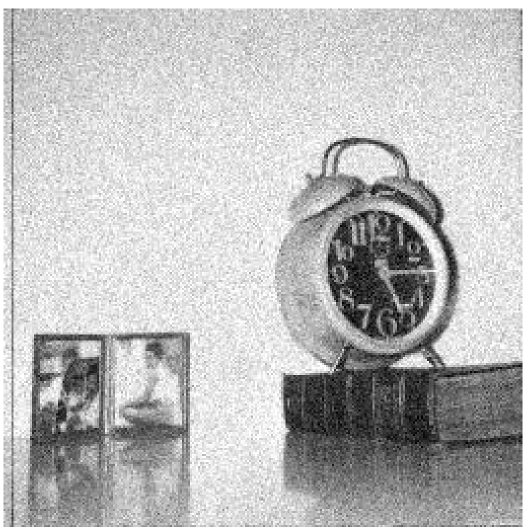
(a) Original image

(b) Noisy image

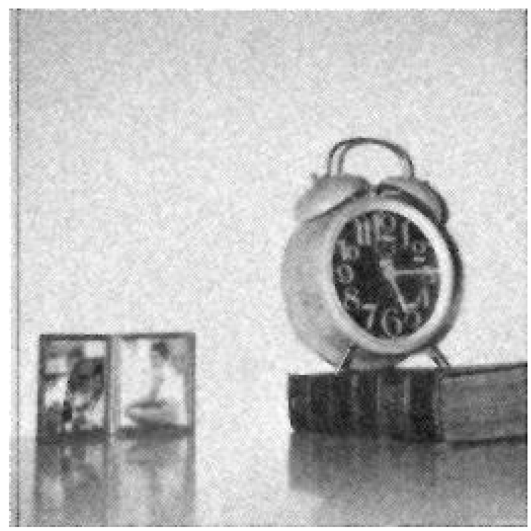
(c) Corrected image

Fig 4.9: (a) Original Lena Image (b) Image corrupted with mean = 0 and variance = 0.005 Gaussian noise (c) Image Corrected by the proposed method

(a) Original image



(b) Noisy image



(c) Corrected image

Fig 4.10: (a) Original Clock Image (b) Image corrupted with mean = 0 and variance = 0.005 Gaussian noise (c) Image Corrected by the proposed method

**Table No. 4.1**

**RMSE & SNR** values for different test images with mean = 0; varying values of variance and keeping $\alpha_{smooth} = 90$

| S. No. | **Images** | **Size** | Varia-nce | RMSE(Noisy Image) | RMSE(Corr-ected Image) | **SNR** |
|---|---|---|---|---|---|---|
| 1 | Mary | 256 X 256 | 0.004 | 16.1223 | 9.9724 | 9.3095 |
| 2 | Red House | | | 16.0493 | 10.4405 | 9.4403 |
| 3 | Clock | | | 15.7618 | 11.1553 | 12.3864 |
| 4 | Elaine | 512 X 512 | | 16.0963 | 9.8973 | 9.7552 |
| 5 | Woman | | | 16.0346 | 9.5369 | 9.4635 |
| 6 | Camera Man | 256 X 256 | 0.005 | 17.3930 | 12.3469 | 7.8446 |
| 7 | Bridge | | | 17.7790 | 14.7014 | 6.0641 |
| 8 | Lena | | | 17.9031 | 11.4848 | 7.8695 |
| 9 | Flintstones | 512 X 512 | | 17.0884 | 13.0253 | 9.4739 |
| 10 | Ashtray | | | 17.8079 | 10.6000 | 7.6066 |
| 11 | Gordios | 423 X 629 | | 17.4753 | 11.0202 | 11.1733 |
| 12 | Peppers | 256 X 256 | 0.006 | 19.5243 | 13.3465 | 7.1470 |
| 13 | Boat | 512 X 512 | | 19.6118 | 12.5558 | 7.3986 |
| 14 | Barbara | | | 19.5630 | 13.9434 | 6.4731 |
| 15 | House | | | 19.6529 | 12.5637 | 8.9627 |

## 4.4.2 Analysis from result set 1 & table 4.1

After carefull analyzing the figure we can make out that the performance of the given filter is not same for all the images. We get best results for Ashtray and Gordios figures and worst results for Bridge and Flintstones figures.

## Table No. 4.2

Comparision of RMSE and SNR values for different values of $\alpha_{smooth}$ ( $0 < \alpha_{smooth} <= L-1$) ; with mean = 0; variance = 0.005.Test applied on famous Lena Image.

| S.No. | $\alpha_{smooth}$ | 3 RMSE | 4 SNR |
|---|---|---|---|
| 1 | 5 | 28.5877 | 3.0876 |
| 2 | 10 | 27.2629 | 3.2909 |
| 3 | 18 | 23.1663 | 4.1110 |
| 4 | 25 | 19.1106 | 5.0793 |
| 5 | 35 | 15.2663 | 6.2950 |
| 6 | 50 | 12.7616 | 7.2590 |
| 7 | 60 | 12.2070 | 7.4785 |
| 8 | 75 | 11.7060 | 7.7458 |
| 9 | 80 | 11.6009 | 7.8249 |
| 10 | 90 | 11.4848 | 7.8695 |
| 11 | 95 | 11.5399 | 7.8274 |
| 12 | 100 | 11.5684 | 7.8191 |
| 13 | 105 | 11.6023 | 7.7858 |
| 14 | 110 | 11.6569 | 7.7849 |
| 15 | 125 | 11.9081 | 7.6994 |
| 16 | 150 | 12.4362 | 7.5000 |
| 17 | 175 | 12.7958 | 7.4265 |
| 18 | 200 | 13.0512 | 7.3825 |
| 19 | 225 | 13.1742 | 7.3290 |
| 20 | 250 | 13.1245 | 7.3415 |
| 21 | 255 | 13.1377 | 7.3332 |

## 4.4.3 Analysis from table 4.2

The RMSE value for the corrupted image with mean = 0 and variance = 0.005 Gaussian Noise applied on 256 X 256 Lena Image is 17.9031. From the table 4.2 we find that for small values of $\alpha_{smooth}$ we don't get any filteration; in fact on passing the noisy image from the filter the noise is enhanced. As we increase the value of $\alpha_{smooth}$ the noise starts decreasing; it attains best value but on furthur increase in $\alpha_{smooth}$ the noise furthur starts increasing. We get best results for $\alpha_{smooth} = 90$.
The values of RMSE and SNR does not change significantly for $\alpha_{smooth}$ in the range of 75 to 255.

**Table No. 4.3**

Comparision of **RMSE** and **SNR** values for different values of variance; mean = 0, keeping $\alpha_{smooth}$ = 90.Test applied on 256 X 256 Lena Image.

| S.No. | Variance | RMSE(Noisy Image) | RMSE(Corrected Image) | 5 **SNR** |
|---|---|---|---|---|
| 1 | 0.001 | 8.0461 | 6.7021 | 14.4065 |
| 2 | 0.002 | 11.4176 | 8.1745 | 11.3406 |
| 3 | 0.003 | 13.9080 | 9.4482 | 9.6809 |
| 4 | 0.004 | 16.1761 | 10.5903 | 8.5397 |
| 5 | 0.005 | 17.9031 | 11.4848 | 7.8695 |
| 6 | 0.006 | 19.7270 | 12.4529 | 7.2114 |
| 7 | 0.007 | 21.2188 | 13.3104 | 6.7534 |
| 8 | 0.008 | 22.7439 | 14.2109 | 6.3011 |
| 9 | 0.009 | 24.0720 | 14.9582 | 5.9683 |
| 10 | 0.01 | 25.3224 | 15.6855 | 5.6854 |
| 11 | 0.02 | 35.2792 | 22.3094 | 4.0105 |
| 12 | 0.03 | 42.3763 | 28.1479 | 3.1903 |
| 13 | 0.04 | 48.3660 | 34.4198 | 2.6443 |
| 14 | 0.05 | 53.2514 | 40.0399 | 2.2787 |

## 4.4.4 Analysis from table 4.3

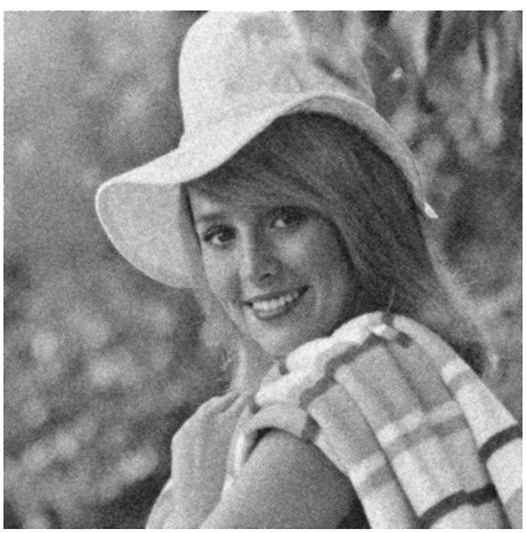
It is evident from the table that as we increase the noise addition or the variance the performance of filter is affected. For larger values of variance the output image is highly noisy. In that case we should go for multiple pass filtering.

### 4.4.5 Result set 2

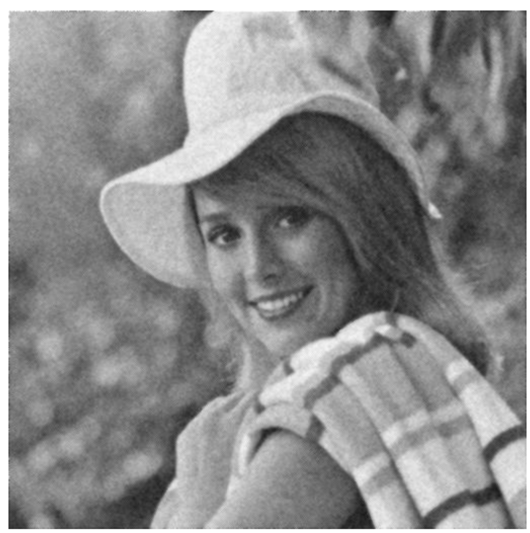
Results of **Multi Pass Filtering**

(a) Original Elaine Image

(b) Corrupted Image



(c) First Iteration



(d) Second Iteration

(e) Third Iteration

(f) Fourth Iteration

(g) Fifth Iteration

Fig 4.11 (a) Original Elaine Image (b) Corrupted Image with mean = 0 & variance = 0.006 Gaussian Noise (c) Image after first iteration (d) Image after second iteration (e) Image after third iteration (f) Image after fourth iteration (g) Image after fifth iteration

Table No. 4.4

Comparision of RMSE & SNR after different passes with $\alpha_{smooth}$ = 90 & with Gaussian noise of mean = 0, variance = 0.006.Test applied on 512 X 512 Elaine image.

| **Pass** | **RMSE** | **SNR** |
|---|---|---|
| 1 | 11.8209 | 8.1521 |
| 2 | 9.7171 | 10.3162 |
| 3 | 9.3958 | 11.2762 |
| 4 | 9.6782 | 11.4491 |
| 5 | 10.2162 | 11.2627 |

## 4.4.6 Analysis from Result set 2 & table 4.4

From the Result set 6 and table 4.4 it is very much clear that multiple passing the image from the filter decreases the Root mean square error & increases the Signal to Noise Ratio for some passes; but if we furthur keep on passing the Image from the filter blurring starts taking place and RMSE starts to increase and SNR starts to decrease. We get best results after third pass. Hence there is no point passing the image beyond third pass.

# Chapter 5

## An improved Fuzzy Image Enhancement by Adaptive Parameter Selection

## 5.1 Intoduction

An effective method for contrast enhancement in an image was presented in chapter 4, which was controlled by trial and error tuning of one parameter. The same parameter was used for entire image, resulting in over, blurring or sharpening of features in some parts of the image. In this chapter we apply that algorithm on Impulse Noise and propose an efficient method for obtaining the parameter adaptively. Each pixel location is adaptively assigned a different parameter value by evaluating the local features. Results of proposed method show that it performs better than other techniques for the enhancement of images corrupted with impulse noise.

## 5.2 Tuning of parameter $\alpha$

We propose to improve the performance of previous method by tuning the $\alpha$ parameter according to a membership function, which reflects the local noise pattern: $\mu(X_{i,j})$ represents the degree of compatibility of a neighbors pixel $x_{m,n}$ with respect to $x_{i,j}$. Thus, the membership function $\mu(X_{i,j})$ is a decreasing function of the sealed residual

$$\frac{(X_{i,j} - X_{m,n})^2}{\beta(X_{i,j})}$$

The fuzzy membership function is defined by :

$$\mu(X_{i,j}, X_{m,n}) = \exp\left[-\frac{(X_{i,j} - X_{m,n})^2}{\beta(X_{i,j}, X_{m,n})}\right] \quad \text{....5.1}$$

in which $\beta$ is the **scale parameter**. The parameter $\beta$ can be determined on the basis of variations in pixel intensities in a given spatial window. Since $\beta$ is an estimate of scale, it should reflect the variance (dispersion ) of the luminance differences between the center pixel and its neighboring pixel. We can simply take the mean of $(X_{i,j} - X_{m,n})^2$ in the neighborhood as described by the function:

$$\beta_{xi,j} = \frac{1}{N-1} \sum_{X_{m,n} \in A} (x_{i,j} - x_{m,n})^2 \qquad ....5.2$$

In order to find out whether a particular center pixel is an impulse noise pixel or part of a uniform area, we have to consider the compatibilities of all the neighboring pixels $X_{m,n}$ in A with respect to the center pixel $X_{m,n}$. To evaluate this this property, we can simply take the mean of $\mu(X_{i,j},X_{m,n})$, as described by the function:

$$\mu_C = \frac{1}{\mathbf{N-1}} \sum_{A} \mu_{xi,j} \qquad 0 <= \mu_C <= 1$$

....5.3

If the center pixel $X_{i,j}$ is an impulse noise pixel, the compabilities $\mu x_{i,j}$ will be small, resulting in a small $\mu_C$. If the center pixel $X_{i,j}$ is an edge or detail pixel, the would be medium. Conversely if the center pixel $X_{i,j}$ is part of a uniform area, $\mu_C$ would be large. This would give us a good idea of the regions in which a small or large would work better.

As mentioned earlier, a large $\alpha$ would cause more smoothing as well as sharpening action: We denote $\alpha$ for smoothing as $\alpha_{smooth}$ and for sharpening as $\alpha_{sharp}$. Since both actions of smoothing and sharpening are opposite to each other, it is to be noted that, in general, $\alpha_{smooth} \neq \alpha_{sharp}$. When impulse noise is present, more smoothing action should be assigned. Presence of impulse noise is indicated by a small $\mu_C$. Hence when $\mu_C$ is small $\alpha_{smooth}$, should be large. On the other hand $\alpha_{smooth}$ should decrease as $\mu_C$ increases so as not to blur edges. So $\alpha_{smooth}$ is given by the following relationship:

$$\alpha_{smooth} = \min(\exp(1/\mu_C), L-1) \qquad ....5.4$$

Notice that the min operator is formally required to limit the output value of $\alpha$ as follows: $\alpha <= L - 1$

On the contrary, in the presence of impulse noise, sharpening should be kept to a minimum. Hence, $\alpha_{sharp}$ should be small in the presence of impulse noise and larger in areas, which are relatively noiseless. The resulting membership function of $\alpha_{sharp}$ is an increasing function of $\mu_C$ given by:

$$\alpha_{sharp} = \min \{ \mu_C \times (L - 1), L - 1 \} \qquad \ldots.5.5$$

By separating smooth and detail or noisy areas of an image, the algorithm treats them differently and thus avoids excessive enhancement of noise, which in another common problem for many existing contrast enhancement techniques. This method for parameter tuning results in almost no ringing artifacts around sharp transition regions, which is often seen in images processed by conventional contrast enhancement techniques.

## 5.3 Results And Analysis

### 5.3.1 Result set 1

The proposed filter was tested with different values of fuzzy parameter K with an impulse noise of probability 0.1 on a 512 x 512 Image of Woman. The results are depicted below.

(a) Input Image

(b) Noisy Image

(c) Corrrected Image with $K_1 = 1$ & $K_2 = 1$

(d) Corrrected Image with $K_1 = 1$ & $K_2 = 2$

(e) Corrrected Image with $K_1 = 2$ & $K_2 = 1$

(f) Corrrected Image with $K_1 = 2$ & $K_2 = 2$

(g) Corrrected Image with $K_1 = 3$ & $K_2 = 2$

(h) Corrrected Image with $K_1 = 3$ & $K_2 = 3$

Fig 5.1: (a) Original Woman Image (b) Image corrupted with d = 0.1 Salt & Pepper Noise (c) Image Corrected by the proposed method taking $K_1 = 1$ & $K_2 = 1$ (d) Image Corrected taking $K_1 = 1$ & $K_2 = 2$ (e) Image Corrected taking taking $K_1 = 2$ & $K_2 = 1$ (f) Image Corrected taking $K_1 = 2$ & $K_2 = 2$ (g) Image Corrected taking $K_1 = 3$ & $K_2 = 2$ (h) Image Corrected taking $K_1 = 3$ & $K_2 = 3$

# Table 5.1

| K1 | K2 | RMSE | SNR |
|---|---|---|---|
| 1 | 1 | 13.9611 | 11.1329 |
| 1 | 2 | 15.8489 | 8.7982 |
| 2 | 1 | 6.1444 | 22.3236 |
| 2 | 2 | 6.6442 | 21.2411 |
| 3 | 2 | 16.7108 | 7.3371 |
| 3 | 3 | 19.8490 | 5.8916 |

Table 5.1 showing Root Mean Square Error (RMSE) & Signal to Noise Ratio (SNR) for different values of Fuzzy Parameter K.

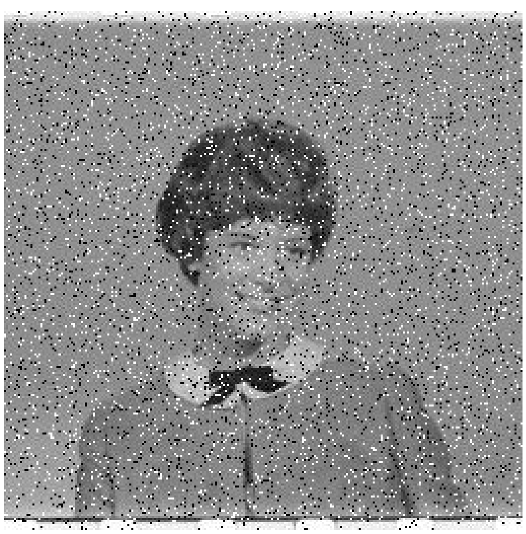
## 5.3.2 Analysis from Fig. 5.1(a) to 5.1(h) & Table 5.1

We conclude from the figures and table that for K1 = 2, K2 = 1 we get best correction. As we keep on increasing the value of K beyond 2 the filter starts performing badly.

### 5.3.3 Result set 2

The filter was tested with Salt & Pepper noise with probability 0.1 for different images.The results are listed below.

(a) Original Image

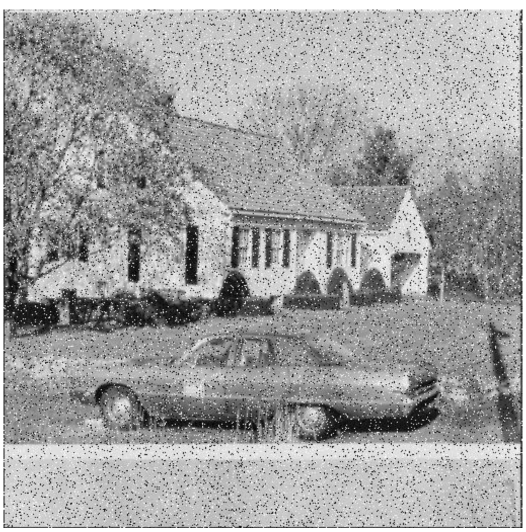


(b) Noisy Image

(c) Corrected Image

Fig 5.2 Result set 2.1: (a) Original Mary Image (b) Image corrupted with d = 0.1; $K_1 = 2$, $K_2 = 2$ Salt & Pepper noise (c) Image Corrected by the proposed method.

(a) Original Image



(b) Noisy Image

(c) Corrected Image

Fig 5.3 Result set 2.2: (a) Original Image of House (b) Image corrupted with d = 0.1; $K_1$ = 2, $K_2$ = 2 Salt & Pepper noise (c) Image Corrected by the proposed method.

(a) Original Image

(b) Noisy Image

(c) Corrected Image

Fig 5.4 Result set 2.3: (a) Original Image (b) Image corrupted with d = 0.1; $K_1$ = 2, $K_2$ = 2 Salt & Pepper noise (c) Image Corrected by the proposed method.

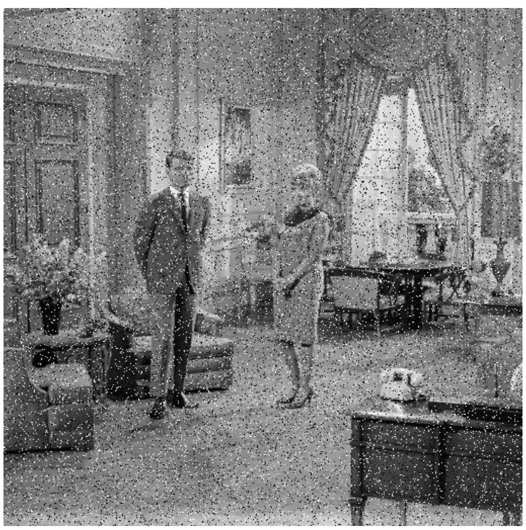
(a) Original Image

(b) Noisy Image

(c) Corrected Image

Fig 5.5 Result set 2.4: (a) Original Image (b) Image corrupted with d = 0.1; $K_1$ = 2, $K_2$ = 2 Salt & Pepper noise (c) Image Corrected by the proposed method.

(a) Original Image



(b) Noisy Image

(c) Corrected Image

Fig 5.6 Result set 2.5: (a) Original Image (b) Image corrupted with d = 0.1; $K_1 = 2$, $K_2 = 2$ Salt & Pepper noise (c) Image Corrected by the proposed method.

## Table 5.2

**RMSE & SNR** values for different test images with d = 0.1, $K_1$ = 2, $K_2$ = 2 Salt & Pepper Noise.

| **S.No.** | 6 Images | **Size** | 7 RMSE | **SNR(Noisy Image)** | **SNR(Corr-ectedImage** |
|---|---|---|---|---|---|
| 1 | Mary | 256 X 256 | 10.8839 | 6.2846 | 18.5769 |
| 2 | Red House | | 10.9040 | 6.5327 | 13.9848 |
| 3 | Clock | | 14.4235 | **11.2057** | **19.5508** |
| 4 | Peppers | | 13.1686 | 5.5975 | 12.0409 |
| 5 | Camera Man | | 16.9141 | 5.4766 | 9.6190 |
| 6 | Lena | | 12.9651 | 5.3680 | 11.9723 |
| 7 | Bridge | | **21.1330** | 4.8603 | **4.9978** |
| 8 | Elaine | 512 X 512 | 8.4827 | 6.3525 | 14.9483 |
| 9 | Woman | | **6.6442** | 4.7532 | **21.2411** |
| 10 | Flintstones | | 18.2023 | 7.5054 | 10.1076 |
| 11 | Ashtray | | **7.0009** | 4.0443 | **19.0881** |
| 12 | Boat | | 12.1123 | 5.8693 | 10.4112 |
| 13 | Barbara | | **16.9731** | 5.2134 | **6.9486** |
| 14 | House | | 11.9241 | 8.7297 | 14.8895 |
| 15 | Gordios | 423 X 629 | 9.8616 | 9.0514 | 20.4057 |

# 5.3.4 Analysis from Fig. 5.2 to 5.6 & Table 5.2

1. The Signal to noise ratio for noisy Clock image is exceptionally high
2. The Image of Woman and Ashtray are corrected show best filtration results
3. Bridge & Barbara Images are not corrected satisfactorily by the proposed filter

### 5.3.5 Result set 3

Comparison of RMSE & SNR for different levels of Salt & Pepper Noise being added with $K_1 = 2$, $K_2 = 1$. Test applied on 423 X 629 Gordios Image.

(a) Original Image

(b) Image with noise of probability d = 0.05

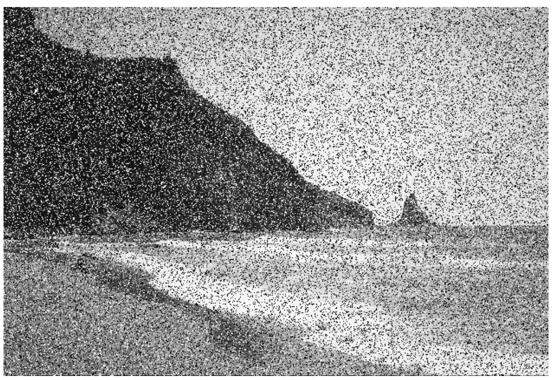
(c) Corrected Image

(d) Image with noise of probability d = 0.1

(e) Corrected Image

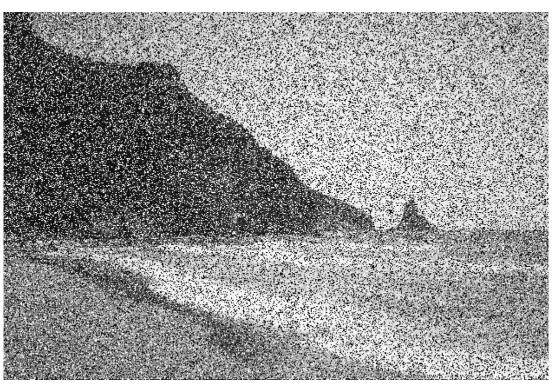
(f) Image with noise of probability
d = 0.2

(g) Corrected Image



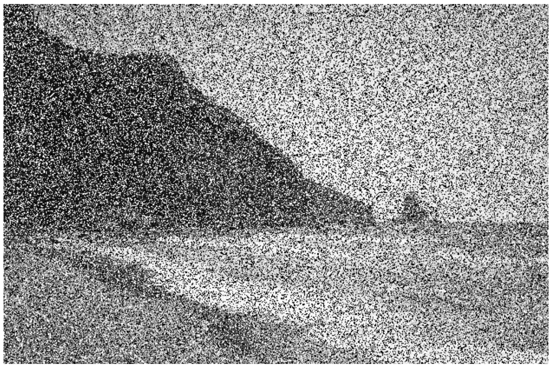
(h) Image with noise of probability
d = 0.3

(i) Corrected Image



(j) Image with noise of probability
d = 0.4

(k) Corrected Image



(l) Image with noise of probability
d = 0.5

(m) Corrected Image

Fig 5.7

## Table 5. 3

List of **RMSE** & **SNR** values for different levels of Salt & Pepper noise added to 423 X 629 Gordios Image with $K_1 = 2$ , $K_2 = 1$

| **S. No.** | **d** | RMSE(Noi-sy Image) | **RMSE(Corr-**ected Image) | **SNR(Noisy Image)** | **SNR(Corr-ected Image)** |
|---|---|---|---|---|---|
| 1 | 0.05 | 33.5762 | 9.2373 | 18.0904 | 21.9684 |
| 2 | 0.1 | 47.4679 | 9.7565 | 9.0166 | 20.6394 |
| 3 | 0.2 | 67.2516 | 12.2464 | 4.5195 | 16.2653 |
| 4 | 0.3 | 82.3633 | 17.5363 | 3.0049 | 11.5008 |
| 5 | 0.4 | 95.2193 | 26.8882 | 2.2500 | 7.7338 |
| 6 | 0.5 | 106.0929 | 38.0333 | 1.8126 | 5.2602 |

## 5.3.6 Analysis from Fig. 5.7 & Table 5.3

We have tested the proposed filter with different levels of Salt & Pepper Noise. We have started with 5% of noise. Upto 20% of noise filter removes noise almost completely; with a very clear output image. Beyond that level the filter removes much of noise but the output image is not that clear. Even when we add 50% of noise the filter decreases RMSE from 1066.0929 to 38.0333 and increases SNR from 1.8126 to 5.2602.

### 5.3.7 Result Set 4

Results of  Multipass Filtering : Experiment applied on 512 X 512 image with K1 = 2, K2 = 1.

(a) Original Image

(b) Corrupted Image with d = 0.2

(c) First Iteration

(d) Second Iteration

(e) Third Iteration

(f) Fourth Iteration

Fig 5.8 Result Set 4.1: (a) Original Image of woman (b) Image corrupted with d = 0.2 Salt & Pepper Noise with $K_1 = 2$, $K_2 = 1$. (c) Image after first iteration (d) Image after Second Iteration (e) Image after Third Iteration (f) Image after fourth Iteration

(a) Original Image

(b) Corrupted Image with d = 0.3

(c) First Iteration

(d) Second Iteration

(e) Third Iteration

Fig 5.9 Result Set 4.2: (a) Original Image of woman (b) Image corrupted with d = 0.3 Salt & Pepper Noise with $K_1 = 2$, $K_2 = 1$. (c) Image after first iteration (d) Image after Second Iteration (e) Image after Third Iteration

# Table 5. 4

List of **RMSE** & **SNR** values for different number of iterations of the proposed filter, with $K_1 = 2$, $K_2 = 1$ ; for Salt & Pepper noise added to 512 X 512 Image (Woman)

| No. of Passes | d | RMSE | SNR |
|---|---|---|---|
| 1 | 0.1 | 6.1444 | 22.3236 |
| 2 | | 7.1625 | 18.0899 |
| 3 | | 8.2692 | 15.2592 |
| 4 | | 9.2113 | 13.4532 |
| 1 | 0.2 | 8.5923 | 15.6398 |
| 2 | | 8.3900 | 15.4623 |
| 3 | | 9.2767 | 18.8466 |
| 4 | | 10.0465 | 12.5727 |
| 1 | 0.3 | 15.0239 | 9.5589 |
| 2 | | 12.8146 | 11.0637 |
| 3 | | 13.0871 | 10.7965 |
| 4 | | 13.4239 | 10.2857 |

## 5.3.8 Analysis from Fig. 5.8,5.9 & Table 5.4

1. For d = 0.1, with the increase of number of iterations we are actually distorting the image. We get best results by passing the distorted image only once from the filter.
2. For d = 0.2, RMSE decreases marginally after second pass but at the cost of image blurring. For more no. of iterations the image gets more & more distorted. Hence multiple pass technique is not helping any way.
3. For d = 0.3, RMSE & SNR values improve significantly for two iterations but subsequent iterations blur the image. Hence for d = 0.3 two iterations are advisable.

# Chapter 6

# Sigmoid filter for Impulse & Gaussian Noise

## 6.1 Sigmoid Membership Function

The sigmoid function is no longer stranger to fuzzy processing. Sigmoid function is used as smoothing intensification operator, which involves parameter 't' and 'a' for enhancement of images. A visible improvement in the quality of smoothness in the image is required.

So we use **sigmoid** membership function with two parameters, defined by

$$\text{sigmoid}(t,d,a) = \frac{1}{1+e^{t(d-a)}} \qquad \textbf{(6.1)}$$

We calculate the value of 'd' from the image information. The above sigmoid function with two parameters 't' and 'a', can be used to control the steepness and position of curve respectively. Using the sigmoid function, we can design a filter, to filter out salt and pepper noise depending on the parameter supplied

## 6.2 Filter Design

Let an image I of size M × N and intensity level in the range (0 L-1) be considered as collection of fuzzy singletons in the fuzzy set notation

$$I = U\{\mu_x(x_{mn})\} = \{\mu_{mn} / x_{mn}\};\ m=1,2\ldots M;\ n=1,2\ldots N \qquad \textbf{(6.2)}$$

Where $\mu_{mn}(x_{mn})$ represents the membership or grade of some property $\mu_{mn}$ of $x_{mn}$.

$x_{mn} = 0,1\ldots L-1$ is the intensity at $(m,n)^{th}$ pixel .For the transformation of the intensity **X** in the range (0,L-1) to the fuzzy property plane in the interval (0,1),a membership function of the **sigmoid** type is used . This technique operates on a window.

Let us consider the $3\times3$ window, to be applied on the image having L gray levels. Let $x_{m,n}$ be pixel luminance at location (m,n) shown in Figure 6.1.

| | | |
|---|---|---|
| | | |
| | $x_{mn}$ | |
| | | |

**Figure 6.1: Showing $3\times3$ window with pixel to be processed**

## 6.2.1 Membership Functions for Impulse Noise

First we rank the neighborhood pixels surrounding the central pixel, based on intensity. A reasonable estimate of a non-noise contaminated pixel is the middle ranking pixel. Therefore we take the difference the middle-ranking pixel (i.e., in the case of 3 × 3 window, with a set of N = 8 neighboring pixels, the middle ranking pixel would be rank 4 and rank 5) and the pixel being processed. Then, we obtain an estimate of noise amplitude. To do this, we use membership functions, which consider the fuzzy relations between the center pixel and its neighbors. They are :

$$\mu_1\left(x_{i,j}, t, a\right) = \frac{1}{1 + e^{\frac{t\cdot\left(\max\left(X_{rank\left(\frac{N}{2}\right)} - X_{i,j}, 0\right) - a\right)}{L-1}}} \qquad \ldots\ldots.(6.3)$$

$$\mu_2\left(x_{i,j}, t, a\right) = \frac{1}{1 + e^{\frac{t\cdot\left(\max\left(X_{i,j} - X_{rank\left(\frac{N}{2}+1\right)}, 0\right) - a\right)}{L-1}}} \qquad \ldots\ldots(6.4)$$

Fuzzy membership $\mu_1$ stands for "central pixel is much darker than the middle ranking neighborhood pixel ". Conversely, $\mu_2$ stand for "central pixel is much brighter than middle ranking pixel".

It may be noted that by varying the value of 't', different nonlinear behavior could be obtained. In order to detect the presence of impulse noise, 't' is set to be negative. The value of 't' should be large and negative in order to remove the salt and pepper noise. The value of 't' is obtained by trial and error procedure. Now the range of 't' is broad which means for many values of 't' it gives the more or less same quality of image output.

The noise amplitude is:

$$n = \max\left[\mu_x\left(X_{i,j}\right)\right] \cdot \left(X_{i,j} - \frac{X_{rank\left(\frac{N}{2}\right)} + X_{rank\left(\frac{N}{2}+1\right)}}{2}\right) \qquad \mu_x = \mu_1 \text{ or } \mu_2 \tag{6.5}$$

Where $\mu_x$ (where x =1,2) represents the class of fuzzy relations described by the parameterized membership function:

### 6.2.2 Membership Function for Gaussian Noise

To detect the presence of Gaussian noise, 't' is set to be positive. In this scenario, we need only to detect whether a central pixel is similar to its neighbors, it is none of our concern to know whether this pixel is brighter or darker than its neighboring pixels. Therefore we have only one membership function, which stands for the linguistic "central pixel's intensity is close to neighborhood pixel's intensity".

The membership function for Gaussian noise is given by

$$\mu_3\left(x_{i,j}, t, a\right) = \frac{1}{1 + e^{\frac{t\left(\left|x_{i,j} - x_{m,n}\right| - a\right)}{(L-1)}}} \tag{6.6}$$

The noise amplitude is:

$$n = \left(\frac{1}{N}\right) \sum_{x_{m,n} \varepsilon A} \mu_3 \left(x_{i,j}, t, a\right) \cdot \left(x_{i,j} - x_{m,n}\right)$$

**…….(6.7)**

In all the membership functions, the value of 'a' determines the selectivity and the sign of 't' tells which kind of noise to detect. It should be observed that by varying the value of t and a (0 < a < = L-1), different nonlinear behavior could be obtained. For impulse noise, a smaller value of 'a' increases noise removal. Conversely, for removing Gaussian noise, a bigger value of 'a' will smooth more.

Finally, the possible noise free value $y_{m,n}$ of the pixel luminance at location (m,n) is then obtained by subtracting the noise estimate n from the original pixel.

$$\mathbf{y_{m,n} = x_{m,n} - n} \qquad \textbf{(6.8)}$$

## 6.3 Multipass Filtering

Noise contamination can occur in any noise type, the most common type being a combination of Impulse and Gaussian like noises. To effectively remove these noises, multipass filtering is used. In the first pass, the noise image is passed through an impulse noise filter. Then, in the second pass, the result of the first pass is filtered with a Gaussian noise filter. This effectively removes most of the noise spikes in the first pass, leaving behind the background noise to be removed in the second pass.

## 6.4 Results & Analysis

### 6.4.1 Result set 1

The filters were tested with different input images. The results are listed below:

(a) Original Image

(b) Corrupted by Impulse Image

(c) Corrected Image

(d) Corrupted by Gaussian Noise

(e) Corrected Image

(f) Corrupted by both kind of Noises

(g) After first Iteration

(h) After Second Iteration

Fig. 6.2 : (a) Original Lena Image
(b)Image corrupted by d = 0.1 Salt and Pepper Noise
(c) Image Corrected by proposed filter
(d) Image corrupted by variance = 0.005 and mean = 0 Gaussian Noise
(e) Image Corrected by proposed filter
(f) Image Corrupted by both type of noises added together
(g) Image after first iteration
(h) Image after second iteration

(a) Original Image

(b) Corrupted by Impulse Image

(c) Corrected Image

(d) Corrupted by Gaussian Noise

(e) Corrected Image

(f) Corrupted by both kind of Noises

(g) After first Iteration

(h) After Second Iteration

Fig. 6.3 : (a) Original Boat Image (b)Image corrupted by d = 0.1 Salt and Pepper Noise (c) Image Corrected by proposed filter (d) Image corrupted by variance = 0.005 and mean = 0 Gaussian Noise (e) Image Corrected by proposed filter (f) Image Corrupted by both type of noises added together (g) Image after first iteration (h) Image after second iteration

(a) Original Image

(b) Corrupted by Impulse Image

(c) Corrected Image

(d) Corrupted by Gaussian Noise

(e) Corrected Image

(f) Corrupted by both kind of Noises

(g) After first Iteration

(h) After Second Iteration

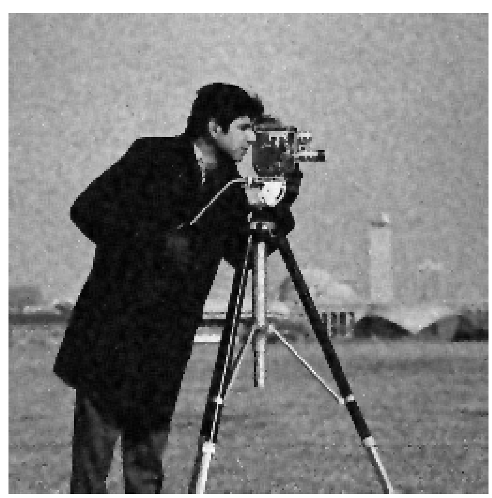
Fig. 6.4: (a) Original Flint Stones Image (b)Image corrupted by d = 0.1 Salt & Pepper Noise (c) Image Corrected by proposed filter (d) Image corrupted by variance = 0.005 and mean = 0 Gaussian Noise (e) Image Corrected by proposed filter (f) Image Corrupted by both type of noises added together (g) Image after first iteration (h) Image after second iteration

(a) Original Image

(b) Corrupted by Impulse Image

(c) Corrected Image

(d) Corrupted by Gaussian Noise



(e) Corrected Image

(f) Corrupted by both kind of Noises

(g) After first Iteration

(h) After Second Iteration

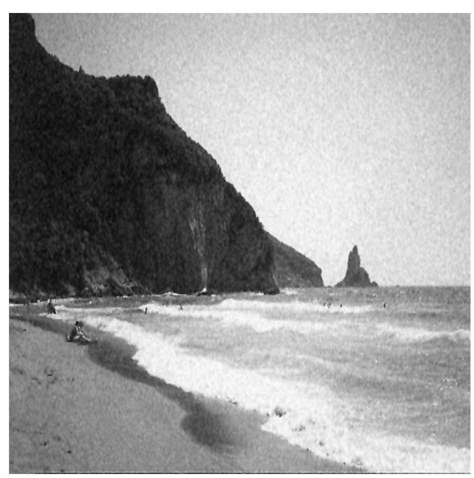
Fig. 6.5: (a) Original Cameraman Image (b)Image corrupted by d = 0.1 Salt & Pepper Noise (c) Image Corrected by proposed filter (d) Image corrupted by variance = 0.005 and mean = 0 Gaussian Noise (e) Image Corrected by proposed filter (f) Image Corrupted by both type of noises added together (g) Image after first iteration (h) Image after second iteration

(a) Original Image

(b) Corrupted by Impulse Image

(c) Corrected Image

(d) Corrupted by Gaussian Noise



(e) Corrected Image

(f) Corrupted by both kind of Noises

(g) After first Iteration

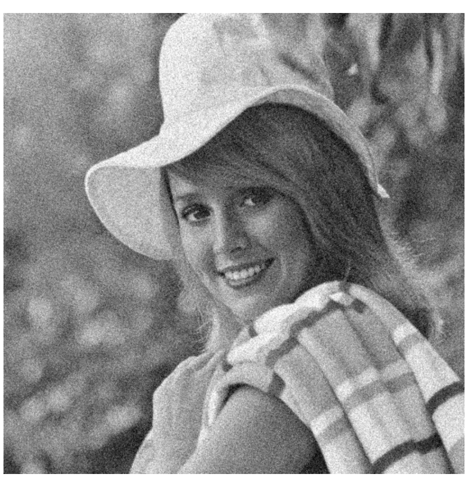
(h) After Second Iteration

Fig. 6.6: (a) Original Gordios Image (b)Image corrupted by d = 0.1 Salt & Pepper Noise (c) Image Corrected by proposed filter (d) Image corrupted by variance = 0.005 and mean = 0 Gaussian Noise (e) Image Corrected by proposed filter (f) Image Corrupted by both type of noises added together (g) Image after first iteration (h) Image after second iteration

(a) Original Image

(b) Corrupted by Impulse Image

(c) Corrected Image



(d) Corrupted by Gaussian Noise

(e) Corrected Image

(f) Corrupted by both kind of Noises

(g) After first Iteration

(h) After Second Iteration

Fig.6.7:(a)Original ElaineImage(b)Image corrupted by d = 0.1Salt & Pepper Noise (c) Image Corrected by proposed filter (d) Image corrupted by variance = 0.005 and mean = 0 Gaussian Noise (e) Image Corrected by proposed filter (f) Image Corrupted by both type of noises added together (g) Image after first pass (h) Image after second pass

## 6.4.2 Analysis from Images

**(a)** From the above experiments it is clear that proposed filter removes Salt and Pepper noise very much satisfactorily. The filter also remove Gaussian noise, but the results obtained are not that satisfactory.

**(b)** When both the noises are added together we use multipass filtering. In the first pass Salt and Pepper noise is removed and in the second pass Gaussian noise is removed.

## Table 6.1

The calculated root mean square error (**RMSE**) for some images for impulse and Gaussian noises using proposed(Sigmoid) filter is tabulated. Also the corresponding RMSE values for the previous filters are presented

| **Figure** | **Size** | **Impulse Noise** | | **Gaussian Noise** | | **Impulse Noise + Gaussian Noise** |
|---|---|---|---|---|---|---|
| | | **Sigmoid Filter** | **Adaptive α Filter** | **Sigmoid Filter** | **Fixed α Filter** | **Sigmoid Filter** |
| Lena | 256X256 | 8.9447 | 12.9651 | 8.7158 | 11.4848 | 12.1909 |
| Boat | 512X512 | 7.1211 | 12.1123 | 9.4232 | 11.6426 | 11.4071 |
| Flintstones | 512X512 | 11.4673 | 18.2023 | 11.2855 | 13.0253 | 16.0426 |
| Clock | 256X256 | 9.2976 | 14.4235 | 8.7842 | 11.9943 | 12.0043 |
| Cameraman | 256X256 | 12.7519 | 16.9141 | 9.8437 | 12.3469 | 15.2184 |
| Ashtray | 512X512 | 5.9539 | 7.0009 | 7.1380 | 10.6000 | 8.6766 |
| House | 512X512 | 10.0046 | 11.9241 | 9.4438 | 11.6906 | 13.2874 |
| Gordios | 423X629 | 8.1935 | 9.8616 | 8.4382 | 11.0202 | 10.6298 |
| Elaine | 512X512 | 3.2805 | 8.4827 | 8.8469 | 10.8936 | 8.8536 |

## 6.4.3 Analysis from Table 6.1

**(a)** RMSE value varies from image to image. The variation for Salt and Pepper noise is greater compared to Gaussian noise.

**(b)** For Salt and Pepper noise the RMSE values by Sigmoid Filter are much better compared to adaptive parameter selection method.

**(c)** Also for Gaussian Noise the RMSE values by Sigmoid filter are better than sharpening and noise reduction method.

## Table 6.2

The calculated Signal to Noise Ratio (**SNR**) for some images for impulse (d = 0.1) and Gaussian (variance = 0.005, mean = 0) noises using proposed (Sigmoid) filter is tabulated. Also the corresponding SNR values for the previous filters are presented

| Figure | Size | Impulse Noise | | Gaussian Noise | | Impulse Noise + Gaussian Noise |
|---|---|---|---|---|---|---|
| | | Sigmoid Filter | Adaptive α Filter | Sigmoid Filter | Fixed α Filter | Sigmoid Filter |
| Lena | 256X256 | 45.0206 | 11.9723 | 10.8874 | 7.8695 | 9.5219 |
| Boat | 512X512 | 45.8521 | 10.4112 | 10.0330 | 7.9856 | 9.1944 |
| Flintstones | 512X512 | 34.9924 | 10.1076 | 11.5380 | 9.4739 | 9.7382 |
| Clock | 256X256 | 73.4313 | 19.5508 | 17.0381 | 11.3705 | 15.1469 |
| Camera-man | 256X256 | 26.1881 | 9.6190 | 10.3467 | 7.8446 | 8.7306 |
| Ashtray | 512X512 | 109.4466 | 19.0881 | 11.5284 | 7.6066 | 11.3228 |
| House | 512X512 | 53.7293 | 14.8895 | 12.8118 | 9.6661 | 11.2702 |
| Gordios | 423X629 | 71.3559 | 20.4057 | 15.2842 | 11.1733 | 14.4045 |
| Elaine | 512X512 | 104.507 | 14.9483 | 11.3247 | 8.8401 | 11.5607 |

### 6.4.4 Analysis from Table 6.2

(a) SNR value varies from image to image . The variation for Salt and Pepper noise is greater compared to Gaussian noise.
(b) SNR values for Impulse Noise is much much better than the Gaussian Noise for the Sigmoid Filter
(c) For Salt and Pepper noise the SNR values by Sigmoid Filter are much much better compared to adaptive parameter selection method.
(d) Also for Gaussian Noise the SNR values by Sigmoid filter are better than sharpening and noise reduction method.

# Table 6.3

Comparison of **R.M.S.E.** and **S.N.R.** values for different values of **'t'** for Impulse Noise(d = 0.1) keeping the value of **'a'** = 20 (constant). Test applied on famous 256 X 256 Lena image.

| S.No. | t | RMSE | SNR |
|---|---|---|---|
| 1 | $-5 X 10^{100}$ | 9.0573 | 47.3146 |
| 2 | $-5 X 10^{10}$ | 8.9730 | 47.6374 |
| 3 | $-5 X 10^{8}$ | 9.1572 | 46.2922 |
| 4 | -2000 | 9.1767 | 45.2354 |
| 5 | -1000 | 9.1155 | 46.3541 |
| 6 | -500 | 8.9652 | 46.7694 |
| 7 | -100 | 9.0291 | 45.4015 |
| 8 | -70 | 8.9585 | 44.3433 |
| 9 | -40 | 8.8952 | 40.5132 |
| 10 | -20 | 8.6179 | 34.3431 |
| 11 | -10 | 8.8236 | 28.0647 |
| 12 | -1 | 17.2729 | 10.1334 |
| 13 | -0.5 | 19.9182 | 9.0552 |
| 14 | -0.1 | 21.9927 | 8.4214 |
| 15 | -0.01 | 22.4403 | 8.2097 |
| 16 | -0.001 | 22.1316 | 8.4069 |
| 17 | +0.001 | 22.2251 | 8.3627 |
| 18 | 0.01 | 22.4265 | 8.2727 |
| 19 | 0.1 | 22.2598 | 8.2979 |
| 20 | 1 | 21.7131 | 8.3763 |
| 21 | 10 | 15.4759 | 10.4413 |
| 22 | 100 | 10.7556 | 16.5341 |
| 23 | 1000 | 10.7555 | 16.3080 |
| 24 | 10000 | 10.8482 | 16.2539 |
| 25 | $10^{5}$ | 11.1309 | 16.1309 |
| 26 | $10^{6}$ | 10.9014 | 16.3389 |
| 27 | $10^{50}$ | 11.0366 | 16.2193 |
| 28 | $10^{101}$ | 10.8303 | 16.5005 |

### 6.4.5 Analysis from table 6.3

This table gives the effect on **RMSE** & **SNR** with the change of parameter 't' while keeping the other parameter 'a' constant at 20, when Impulse noise is applied. Experiment is performed on famous Lena Image.

(a) From the table we observe that the range of 't' is very large & hence there is not much difficulty in setting the value of the parameter.
(b) We notice that as we move from minus infinity to -100 the values of RMSE & SNR do not show much change. But as we start increasing the value of 't' towards zero, SNR starts decreasing and RMSE value shows an upward trend. At 't' = -10 there is no problem on the front of RMSE but the value of SNR becomes poor; hence we get a blurred image.
(c) Now as we further increase the value of 't' RMSE increases & SNR decreases. Hence suitable value of parameter 't' is towards negative infinity. Hence the magnitude is large.

## Table 6.4

Comparision of **R.M.S.E.** and **S.N.R.** values for different values of **'a'** for Impulse noise(d = 0.1) keeping the value of **'t'** = -100 (constant). Test applied on famous 256 X 256 Lena image.

| S.No. | 'a' | RMSE | SNR |
|---|---|---|---|
| 1 | 2 | 10.3050 | 20.9438 |
| 2 | 5 | 9.7195 | 25.2817 |
| 3 | 10 | 9.9317 | 31.7565 |
| 4 | 15 | 9.3627 | 38.5077 |
| 5 | 17 | 9.2231 | 41.5505 |
| 6 | 19 | 9.0005 | 44.9898 |
| 7 | 25 | 8.7880 | 50.5773 |
| 8 | 28 | 8.7037 | 53.2365 |
| 9 | 31 | 8.7086 | 55.6270 |
| 10 | 35 | 8.4474 | 59.3317 |
| 11 | 40 | 8.3739 | 61.4844 |
| 12 | 45 | 8.4474 | 60.6905 |
| 13 | 50 | 8.7385 | 57.2172 |
| 14 | 55 | 9.0091 | 53.7444 |
| 15 | 60 | 9.6391 | 50.0590 |
| 16 | 70 | 9.8021 | 46.7770 |
| 17 | 80 | 11.0293 | 41.2441 |
| 18 | 90 | 12.0332 | 34.7970 |

| | | | |
|---|---|---|---|
| 19 | 100 | 14.2182 | 25.2454 |
| 20 | 110 | 16.6128 | 19.9334 |
| 21 | 130 | 21.8503 | 13.3488 |
| 22 | 150 | 27.5843 | 9.6118 |
| 23 | 180 | 35.1938 | 7.0150 |
| 24 | 200 | 38.3062 | 6.1974 |
| 25 | 220 | 42.7391 | 5.3852 |
| 26 | 255 | 43.0216 | 5.3456 |

## 6.4.6 Analysis from table 6.4

This table gives the effect on **RMSE & SNR** with the change of parameter 'a' while keeping the other parameter 't' constant at –100, when Impulse noise is applied. Experiment is performed on famous Lena image.

(a) Parameter 'a ' can take values between 0 and L.As we start increasing the values of 'a' from 1 to 255 we see that initially RMSE decreases and SNR increases .We get best results at a = 40.

(b) If we further increase the parameter RMSE keep on increasing and SNR keep on decreasing. We get bad results for higher values of 'a'.

# Table 6. 5

Comparision of **R.M.S.E.** and **S.N.R.** values for different values of **'t'** for **Gaussian Noise** (variance = 0.005, mean = 0), keeping the value of **'a'** = 70 (constant). Test applied on famous 256 X 256 Lena image.

| S.No. | t | RMSE | SNR |
|---|---|---|---|
| 1 | $10^{-100}$ | 10.6013 | 8.4906 |
| 2 | $10^{-50}$ | 10.6211 | 8.4398 |
| 3 | $10^{-20}$ | 10.6002 | 8.4806 |
| 4 | $10^{-11}$ | 10.6562 | 8.4432 |
| 5 | $10^{-6}$ | 10.5572 | 8.5050 |
| 6 | $10^{-3}$ | 10.6191 | 8.4758 |
| 7 | 0.01 | 10.5762 | 8.5291 |
| 8 | 0.1 | 10.5100 | 8.5496 |
| 9 | 1 | 10.2909 | 8.7868 |
| 10 | 10 | 8.5613 | 10.8852 |
| 11 | 20 | 8.4732 | 11.1617 |
| 12 | 40 | 8.6445 | 10.9962 |
| 13 | 50 | 8.7037 | 10.9432 |
| 14 | 70 | 8.8115 | 10.7970 |
| 15 | 90 | 8.8189 | 10.8100 |
| 16 | 130 | 8.8298 | 10.7620 |
| 17 | 170 | 8.9000 | 10.7405 |
| 18 | 250 | 8.9623 | 10.6100 |
| 19 | 1000 | 8.9487 | 10.6827 |
| 20 | $10^{4}$ | 8.9866 | 10.5848 |
| 21 | $10^{6}$ | 8.9359 | 10.7113 |
| 22 | $10^{10}$ | 8.9832 | 10.6385 |
| 23 | $10^{20}$ | 8.9481 | 10.6227 |
| 24 | $10^{50}$ | 8.9968 | 10.6191 |
| 25 | $10^{100}$ | 8.9742 | 10.6464 |

## 6.4.7 Analysis from table 6.5

This table gives the effect on RMSE & SNR with the change of parameter 't' while keeping the other parameter 'a' constant at 70, when Gaussian noise is applied. Experiment is performed on famous Lena image.

We see that 't' takes a very large range of values. For Gaussian Noise 't' has to be positive. As we increase value of 't' from zero to infinity we don't find much change in either SNR or RMSE. We get best results for t = 20.

# Table 6.6

Comparision of **R.M.S.E.** and **S.N.R.** values for different values of **‘a’** for **Gaussian Noise** (variance = 0.005, mean = 0) keeping the value of **‘t’** = 50 (constant). Test applied on famous 256 X 256 Lena image.

| S.No. | ‘a’ | RMSE | SNR |
|---|---|---|---|
| 1 | 10 | 17.3524 | 5.1476 |
| 2 | 20 | 15.9674 | 5.7124 |
| 3 | 30 | 14.1026 | 6.7084 |
| 4 | 40 | 11.6984 | 8.2467 |
| 5 | 50 | 9.9096 | 9.7564 |
| 6 | 60 | 8.9344 | 10.6857 |
| 7 | 70 | 8.7037 | 10.9432 |
| 8 | 80 | 8.7958 | 10.8590 |
| 9 | 90 | 9.0670 | 10.5899 |
| 10 | 100 | 9.4032 | 10.2973 |
| 11 | 110 | 9.6863 | 10.1587 |
| 12 | 120 | 9.9179 | 9.9464 |
| 13 | 130 | 10.1099 | 9.8397 |
| 14 | 140 | 10.3662 | 9.7298 |
| 15 | 150 | 10.4561 | 9.7286 |
| 16 | 160 | 10.7072 | 9.5988 |
| 17 | 170 | 10.7968 | 9.5666 |
| 18 | 180 | 10.9329 | 9.5324 |
| 19 | 190 | 10.9421 | 9.5157 |
| 20 | 200 | 11.0551 | 9.4918 |
| 21 | 210 | 11.1240 | 9.4425 |
| 22 | 220 | 11.0993 | 9.4862 |
| 23 | 230 | 11.0811 | 9.5154 |
| 24 | 240 | 11.1479 | 9.4585 |
| 25 | 255 | 11.1865 | 9.4486 |

## 6.4.8 Analysis from table 6.6

This table gives the effect on RMSE & SNR with the change of parameter ‘a’ while keeping the other parameter ‘t’ constant at 50, when Gaussian noise is applied. Experiment is performed on famous Lena Image.

As we start increasing the value of ‘a’ from zero SNR increases and RMSE decreases. Both these attain their best values and then on further increase in ‘a’ SNR and RMSE starts detiorating. We get best results for **a = 70**

# Chapter 7

# Conclusions and Suggestions for Future work

## 7.1 Conclusions

A fuzzy filter to remove Gaussian Noise has been presented in chapter four. It is a very simple filter and no complex tuning of fuzzy set parameters is required. In fact, the overall nonlinear behavior of the enhancement system is very easily controlled by one parameter '$\alpha$' only .

An effective method to remove Salt & Pepper Noise has been explained in chapter five. An effective method for contrast enhancement in an image was presented in chapter four, which was controlled by trial and error tuning of one parameter. The same parameter was used for entire image, resulting in over blurring or sharpening of features in some parts of the image. In this chapter we apply that algorithm on impulse noise and propose an efficient method for obtaining the parameter adaptively. Each pixel location is adaptively assigned a different parameter value by evaluating the local features.

A fuzzy filter for denoising of images corrupted with Salt & Pepper and Gaussian noises is proposed in chapter six. The filter developed to remove salt and pepper noise removes the noise successfully and the results are very satisfactory. The filter to remove Gaussian noise also removes noise to a great extent but the results are not that satisfactory.
The main feature of the Fuzzy filter is that it distinguishes between local variation due to noise and due to the image structure, using a sigmoid membership function that model the image information in the spatial domain.

## 7.2 Future work

In chapter five, we have presented the filter for Salt & Pepper Noise. The filter is based on a parameter '$\beta$' which is variable for each pixel and calculated from image information. Now this method is able to remove Salt & Pepper noise successfully but not the Gaussian noise. There is furthur scope to extend this filter to remove Gaussian noise as well.

In the Chapter six, we have presented the filter for Salt & Pepper Noise, we have formulated the value of ‘d’ from the image information and the values of the parameter changes depend on the local context, other parameter ‘t’ is select by trial and error procedure and we have suggested the large negative values is sufficient for good filtering of Salt and pepper noise, whereas, small positive values is required for filtering Gaussian noise. Instead of the trial and error procedure we can further look to automate the value of ‘t’ also. Also there is a furthur scope to develop a better membership function for Gaussian noise which would remove noise more satisfactorily.We have suggested two different membership functions for Impulse and Gaussian noise respectively. There is a further scope for uniting the membership function for both the noise.

# Bibliography


[1] Russo F.*" An Image Enhancement Technique Combining Sharpening and Noise Reduction"*. IEEE Trans. Instrumentation and Measurement , vol 51, No. 4, August 2002

[2] H.S. Kam, M. Handmandlu "*An Improved Fuzzy Image Enhancement by Adaptive Parameter Selection"*, proceedings of the IEEE TENCON (Convergent Technologies for the Asia-Pacific Region) 2003, pp.2001-2006

[3] H.S. Kam, M. Hanmandlu and W.H. Tan "*A new tunable fuzzy filter for image processing"*, accepted at the IEEE International Symposium on Signal Processing and Information Technology,14-17 December 2002, Darmstadt, Germany.

[4] Dimtri Van De Ville, Mike Nachtegael , Dietrich Vander Weken, Etienne E. Kerre, Wilfried Philips, and Ignace Lemahieu ." *Noise Reduction by Fuzzy Image Filtering* ". IEEE Transaction on Fuzzy Systems, vol 11, No. 4,August 2003

[5] Choi, Y.S., Krishnapuram, R., 1997. *"A robust approach to image enhancement based on fuzzy logic"*. IEEE Trans. Image Process. 6 (6), 808–825.

[6] Hanmandlu, M., Tandon, S.N., Mir, A.H., 1997.*"A new fuzzy logic based image enhancement"*, 34th Rocky Mountain Symposium on bioengineering, Dayton, Ohio, USA, pp. 590–595.

[7] Russo, M., Ramponi, G., 1995. " A fuzzy operator for the enhancement of blurred and noisy images". IEEE Trans. Image Process. 4 (8), 1169–1174.

[8] H.S. Kam, M. Handmandlu and W. H. Tan “*An Adaptive fuzzy filter system for smoothing images”*, proceedings of the IEEE TENCON (Convergent Technologies for the Asia-Pacific Region) 2003,15-17 October 2003,Banglore, India, vol. 4 pp.1614-1617

[9] H.S. Kam, M. Handmandlu and W. H. Tan, “An Improved Image Enhancement Combining Smoothing and Sharpening”, *Proceedings of the IEEE TENCON* (Convergent Technologies for the Asia-Pacific Region) 2003,15-17 October 2003,Banglore, India, Vol. 4, pp.1614-1617

[10] Lee Jong-Sen, *Digital Image Enhancement and noise filtering*, IEEE Trans. Pattern Anal. Machine Intell., 2, 165-168.

[11] Saint-Marc, P. Chen, J., and Medioni, G., *Adaptive smoothing: A general tool for early vision*, IEEE Trans. Pattern Anal. Machine Intelli., 13(6) : 514-529.

[12] Rafael C. Gonzales and Richard E. Woods, *“Digital Image Processing”*, Second Edition, Prentice Hall, 2002.

[13] Yvonne Kam, Multimedia University “*Image denoising using a fuzzy parameter tuned filter*”

[14] Jerry M. Mendel, fellow IEEE “*Fuzzy Logic Systems for Engineering*”