

C1

Noise Filtering with Edge Preservation in Digital Images

by

Claudio Gustavo Rey

B. A. Sc., The University of British Columbia, 1983

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE**

in

THE FACULTY OF GRADUATE STUDIES

Department of Electrical Engineering

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

December 1985

© Claudio Gustavo Rey, 1985

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Electrical Engineering

The University of British Columbia
1956 Main Mall
Vancouver, Canada
V6T 1Y3

Date January 31st, 1986

Abstract

The widespread use of the absolute gradient and the sample variance in present day local noise filters in digital image processing is pointed out. It is shown that the sample variance and the absolute gradient can be viewed as measures of the modelling error for a simple zeroth order local image model. This is shown to lead to a general formulation of local noise filtering applicable to the great majority of current local noise filters for digital images. This formulations describes local noise filtering as a two step process. In the first step a robust estimation of every pixel z_o is obtained. In the second step a better estimate of z_o is obtained by performing a weighted sum within a neighborhood of z_o . The weights in the second step are related to some measure of modelling error of the above zeroth order model; namely, the absolute gradient or the sample variance.

Of the above two measures of modelling error, the sample variance is shown to be the least sensitive to noise. Furthermore, the sample variance is also more sensitive to faint image edges. Therefore, the sample variance is the most desirable of the above two measures of modelling error. Unfortunately, its use until now has been hampered by its poor edge localization. To solve this problem a new measure of modelling error is introduced which achieves far superior edge localization than the sample variance (though still lower than the absolute gradient) but maintains the low noise sensitivity. A filter is designed based on this new measure of modelling error (of the zeroth order model described above) which is shown to perform better, in a least squares sense, than all other local noise filters for non-impulsive additive and multiplicative noise. A practical implementation of this original filter is also presented.

TABLE OF CONTENTS

Abstract	ii
Table of Contents	iii
List of Tables	v
List of Figures	vi
Acknowledgement	x
Chapter 1: Introduction	1
Chapter 2: Image Modelling	3
2.1 The Modelling Error	5
2.2 A Measure of Performance: the Mean Square Error	7
2.3 Analysis of the Mean Square Error for Non-Linear Filters	10
Chapter 3: Robust Filters	13
3.1 The Median Filter	14
3.2 The Ordered Statistics Filter	15
3.3 The M-Smoother	18
3.4 The Modified Trimmed Mean Filter	19
Chapter 4: Gradient-Based Filters	23
4.1 The Sigma Filter	25
4.2 The Gradient Inverse Filter	27
4.3 A Generalized Gradient Based Filter	29
4.4 A General Formulation for Gradient-Based Methods and the DWMTM filter.	30
Chapter 5: Variance-Based Filters	32
5.1 Nagao's Smoother	35
5.2 J. S. Lee's Local Statistics Smoother	38
5.3 An Original Localized Variance Smoother	40
Chapter 6: A Comparative Study of Local Noise Filtering Techniques	46
6.1 A Root Mean Square Analysis	47
6.2 A Visual Analysis	51

Conclusion	53
Appendix A	55
Appendix B: For Further Research	56
References	

List of Tables

Table 6.1 - root mean square (RMS) performance of the filters for the image T degraded by varying amounts of additive noise and multiplicative noise	48
Table 6.2 - root mean square (RMS) performance of the filters for the image F degraded by varying amounts of additive noise and multiplicative noise.	48
Table 6.3 - root mean square (RMS) performance of the filters for the image C degraded by varying amounts of additive noise and multiplicative noise.	49
Table 6.4 - root mean square (RMS) performance of the sigma, DWMTM and generalized gradient filters in removing the noise of the image CMG0.2. Up to two iterations used	49
Table A.1 - the three components of the mean square error for several filters applied to the image TAG10.	55

List of Figures

Figure 2.1 - (a) - original ideal step. (b) - 5 pixel ramp resulting from the application of the running mean filter with window size 5	4
Figure 2.2 - (a) - original "Telephone" image. (b) - original image degraded by gaussian noise with $\sigma_{\epsilon}^2 = 25$. (c) - degraded image smoothed by the running mean filter with window size 5x5.	5
Figure 2.3 - (a) - original image and degraded image; (b) - model within a window containing pixels 4, 5, 6, 7 and 8; (c) - error sequence—additive white and non-impulsive; (d) - modelling error.	6
Figure 2.4 - Graph showing the tradeoff between the two terms of the RHS of (2.13).	9
Figure 2.5 - Method for extracting \tilde{x}_{ij}	12
Figure 3.1 - (a) - "Telephone" smoothed with median filter with window size 3x3; (b) - same as above but with window size 5x5	14
Figure 3.2 - (a) - original image; (b) - output after smoothing with a 3x3 median filter	15
Figure 3.3 - (a) - original "Camera" image; (b) - Camera image degraded by "salt" noise; (c) - noise removed using a median filter of size 3x3; (d) - noise removed using a median filter of size 5x5.	16

Figure 3.4 -	(a) - "Telephone" degraded by AWGN with $\sigma_\epsilon^2 = 225$;	
	(b) noise removed using window size 9 (3x3) running mean filter;	
	(c) noise removed using window size 9 (3x3) median;	
	(d) noise removed using window size 9 (3x3) OSF with	
	weights: (0, 0, 0, 1/3, 1/3, 1/3, 0, 0, 0)	17
Figure 3.5 -	(a) - "Telephone" degraded by additive white Gaussian noise	
	(AWGN) with $\sigma_\epsilon = 15.0$;	
	(b) - noise removed of degraded image of Figure 3.4b with the	
	m-smoother of window 5x5;	
	(c) - noise removed with DWMTM filter with median window size	
	of 3x3 and smoothing window size of 7x7;	
	(d) - noise removed with 5x5 median.	21
Figure 4.1 -	The \mathfrak{S} function corresponding to the sigma filter.	25
Figure 4.2 -	(a) - noise removed image of Figure 2.2b using a sigma filter	
	of size 7X7 and $p = 12.5$;	
	(b) - noise removed using a DWMTM filter with window sizes 3x3	
	and 7x7 and threshold $p = 12.5$.	25
Figure 4.3 -	(a) - noise removed from image of Figure 3.3b using a sigma	
	filter of window size 7X7 and threshold $p = 3.0$;	
	(b) - noise removed using a running mean filter of window	
	size 3x3; (c) - noise removed using the generalized filter of	
	Section 4.4.	27
Figure 4.4 -	(a) - "Camera" degraded by AWGN with $\sigma = 15.$;	
	(b) - noise removed using a sigma filter with window	
	size 7X7 and threshold $p = 37.5$;	
	(c) - noise removed using a median filter with window	
	size 5x5.	28

Figure 4.5 - The \mathfrak{S} function of the gradient inverse weighting technique	29
Figure 4.6 - The \mathfrak{S} function of the generalized gradient-based filter	29
Figure 5.1 - (a) - variance map of the “Telephone” using a 3x3 window; (b) - variance map of the “Telephone” using a 5x5 window; (c) - variance map of the degraded “Telephone” of Figure 3.4 using a 3x3 window.	36
Figure 5.2 - normalized variance map of the “Telephone” degraded by MWGN with $\sigma_v = .2$ using a 3x3 window.	37
Figure 5.3 - Three of the regions checked by Nagao’s filter.	37
Figure 5.4 - noise removed from the degraded image of Figure 3.5a using J. S. Lee’s local statistics filter.	38
Figure 5.5 - noise removed from the degraded image of Figure 2.2b using J. S. Lee’s local statistics filter with window size 5x5;	39
Figure 5.6 - (a) - undegraded “Face” image; (b) - “Face” degraded by MWGN with $\sigma_v = 0.2$; (c) - noise removed using J. S. Lee’s local statistics filter with window size 7x7; (d) - noise removed using J. S. Lee’s sigma filter with window size 7x7; (e) - noise removed using the localized variance smoother of Section 5.3 with window with window size 5x5 for computing the local variance map and window size 7x7 for smoothing.	41
Figure 5.7 - Example showing the two 3x3 neighborhood sample variances	

which must be computed to determine M_{ij} at pixel z_{ij}	42
Figure 5.8 - edge localization by the new mismodelling measure: M_{ij}	43
Figure 5.9 - noise removed of degraded images of Figure 4.4a and 3.4a smoothed using localized variance smoother with a window size 5x5 for computing the local variance map and window size 7x7 for smoothing.	45
Figure 6.1 - (a) - "Face" degraded by AWGN with $\sigma_\epsilon = 15.0$; (b) - noise removed using J. S. Lee's local statistics filter with window size 7x7; (c) - noise removed using J. S. Lee's sigma filter with window size 7x7; (d) - noise removed using the DWMTMF with median window size 3x3 and smoothing window size 7x7; (e) - noise removed using the localized variance smoother of Section 5.3 with window with window size 5x5 for computing the local variance map and window size 7x7 for smoothing.	52

Acknowledgement

I would like to express my gratitude to my supervisor, Professor Rabab K. Ward for her invaluable support and encouragement.

I would also like to thank Professor Mabo R. Ito, who supervises me in my current employment, for his understanding of the constraints involved with the writing of this thesis.

Chapter 1: Introduction

Noise filters in digital image processing must be able to handle many different types of noise. Some of these are not correlated with the image, but others are; some are white; while others are colored. In some cases, such as channel noise or the scanning by a vidicon television camera, the noise is additive and uncorrelated with the image [11]. In other cases, such as synthetic aperture radar images the noise is multiplicative and, under certain restrictions, it is approximately uncorrelated with the image [9]. A more complicated example is photographic graininess which arises because of the finite size of the Silver Halide “grains” in the photographic emulsion.

Noise filters (or smoothers) for digital image processing can be separated into two main categories. In the first category, the noise degraded image is processed as a whole or in separate non-overlapping regions to produce the filtered image. This category includes filters using Wiener or least squares filtering, constrained deconvolution or recursive filters such as those applying one-dimensional or two-dimensional Kalman filtering. These filters often assume that the noise is uncorrelated with the signal and cannot be easily adapted to signal dependent noise. A majority of them also require statistical models for the undegraded image [11]. Images smoothed using these techniques display blurred edges and conceal subtle detail. In addition, the techniques do not allow for parallel processing. In the second category, local operators are applied to noisy images. Only those pixels in a small neighbourhood of the concerned pixel are involved in the computation. The immediate advantage of these techniques is their efficiency—pixels can be processed in parallel. A further advantage is that multiplicative noise and other image dependent noise sequences can be filtered. Recent research in image noise filtering favors the local techniques.

Local noise filtering algorithms can also be separated into three main categories. The first category contains robust filters which are based upon robust estimation techniques.

The most well known representative of robust filters is the median filter [10]-[11]. Since the introduction of the median filter several other related noise removing algorithms have appeared. The most relevant of these are: the ordered statistics smoother of A. C. Bovik *et al* [2], and the standard trimmed mean and the double-window modified trimmed mean filter of Y. H. Lee and S. A. Kassam [1].

The second category of local noise filtering algorithms contains the gradient-based filters. The filters in this category include J. S. Lee's sigma filter [3] and the gradient inverse weighted smoother of D. Wang *et al* [5].

The third category of local noise filtering algorithms contains the variance-based filters. The filters in this category include J. S. Lee's local statistics smoother [7] and the edge preserving smoother of M. Nagao *et al* [6].

In the following chapters, an underlying principle behind local noise filtering algorithms is presented. It is shown that a simple zeroth order local image model can be used for analyzing the present local noise filtering algorithms. The use of the above image model is shown to lead to a generalization of the gradient based methods and robust methods. It is also shown that the current variance-based filters suffer from a lack of edge localization. A solution for this problem is given and a new filter is introduced which can produce better least squares performance than all other current local noise removing filters for non-impulsive noise degradation.

Chapter 2: Image Modelling

This chapter discusses the use of a zeroth order local image model for image noise removal. Based on this model, this chapter describes the relationship between the modelling error (in the above image model) and the edge degradation of linear filters. Furthermore, the role of this modelling error in the least squares performance of all current local noise filters is discussed.

The models for the only two different forms of noise degradation used are:

$$z_{ij} = x_{ij} + \epsilon_{ij} \quad \text{for additive noise degradation;} \quad (2.1)$$

and

$$z_{ij} = x_{ij}\nu_{ij} \quad \text{for multiplicative noise degradation.} \quad (2.2)$$

where: $\left\{ \begin{array}{l} x_{ij} \text{ represents the image intensity of the undegraded image;} \\ z_{ij} \text{ is the measured grey level;} \\ \epsilon_{ij} \text{ is a noise sequence;} \\ \nu_{ij} \text{ is a noise sequence.} \end{array} \right.$

ϵ_{ij} and ν_{ij} must satisfy:

$$\left\{ \begin{array}{l} E\{\epsilon_{ij}\} = E\{\nu_{ij} - 1\} = 0 \\ E\{\epsilon_{ij}^2\} = \sigma_\epsilon^2 \text{ (constant)} \\ E\{\nu_{ij}^2\} = \sigma_\nu^2 \text{ (constant)} \\ E\{\epsilon_{ij}\epsilon_{mn}\} = E\{\nu_{ij}\nu_{mn}\} = 0 \quad (i, j) \neq (m, n) \\ E\{\epsilon_{ij}x_{mn}\} = E\{(\nu_{ij} - 1)x_{mn}\} = 0 \quad \forall i, j, m, n \end{array} \right. \quad (2.3)$$

The noise sequences ϵ_{ij} (or ν_{ij}) are classified as impulsive or non-impulsive according to their statistical distribution. Types of distributions that lead to non-impulsive noise are Gaussian, Uniform, Triangular and Generalized Exponential with $\beta > 2$. Types of distributions that lead to impulsive noise are: Laplacian, Cauchy and Generalized Exponential with $\beta < 2$. For a description of the Generalized Exponential distribution see [2].

A difficulty with using (2.1) and (2.2) for signal estimation (to remove noise) is that no model for the signal, either statistic or deterministic is given. Clearly, some assumption

must be made. One simple assumption is that for a "small" neighborhood of an image the grey level is constant:

$$x_{ij} = x_o \quad (2.4)$$

where: $\begin{cases} i, j & \text{encompass only a small neighborhood of the image} \\ & \text{(usually, a square array of dimensions varying from } 3 \times 3 \text{ to } 11 \times 11); \\ x_{ij} & \text{is the undegraded grey level at } (i, j); \\ x_o & \text{is the grey level of the neighborhood's center pixel.} \end{cases}$

For an example of another local signal model, see [4].

Some local smoothing algorithms make use of the model of (2.4) for removing noise; i. e. for estimating the grey level x_o . It is easily shown that according to the model of (2.4) the (least squares) optimum algorithm for removing additive or multiplicative noise is the running mean filter. The running mean filter estimates the gray level of a pixel x_o by the average within a small neighborhood of x_o .

The application of the running mean filter causes, as shown in Figure 2.2, a pronounced blurring effect. This undesirable blurring effect is due in part to edge degradation. To illustrate this consider a one-dimensional ideal step signal. When filtered by a running mean smoother the step is transformed into a ramp whose width equals the size of the smoothing window. See Figure 2.1.

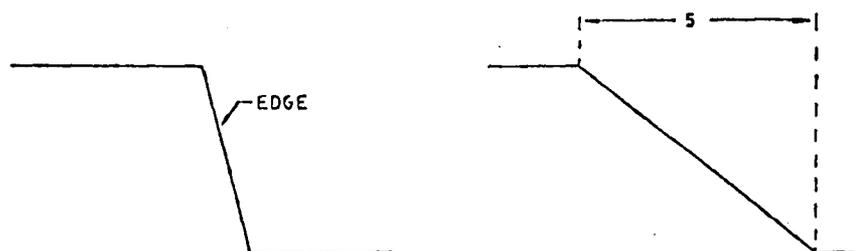


Figure 2.1 - (a) - original ideal step; (b) - 5 pixel ramp resulting from the application of the running mean filter with window size 5.

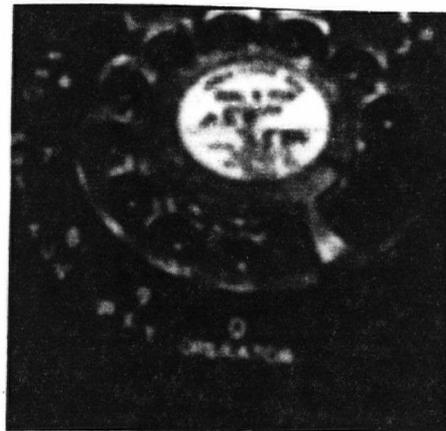
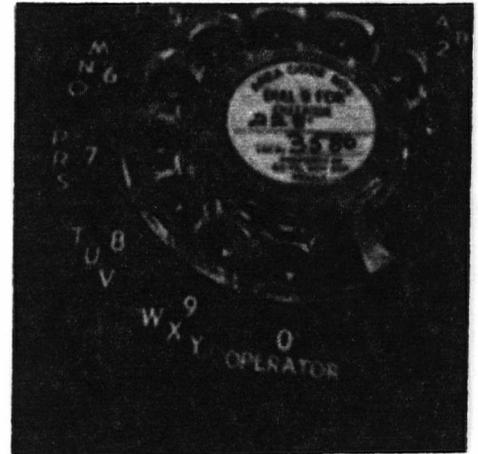
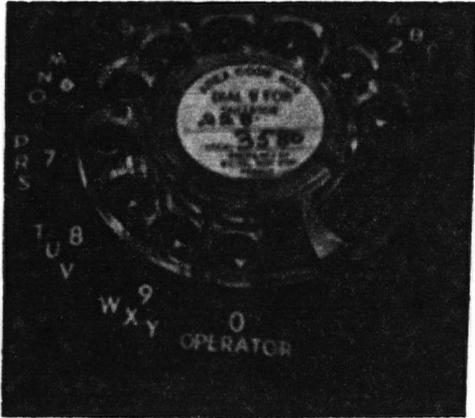


Figure 2.2 - (a) - original "Telephone" image; (b) - original image degraded by gaussian noise with $\sigma_e^2 = 25$.; (c) - degraded image smoothed by the running mean filter with window size 5x5.

2.1 The Modelling Error

The very undesirable blurring effect caused by the application of the running mean filter shows how limited the model of (2.4) is. As illustrated by Figure 2.1, no matter how small a neighborhood is used, the image model of (2.4) will always fail in areas of sharp contrast. In general, $x_o \neq x_{ij}$ and the difference is called the modelling error η_{ij} :

$$x_{ij} = x_o + \eta_{ij} \tag{2.5}$$

Therefore:

$$\eta_o = 0$$

(2.6)

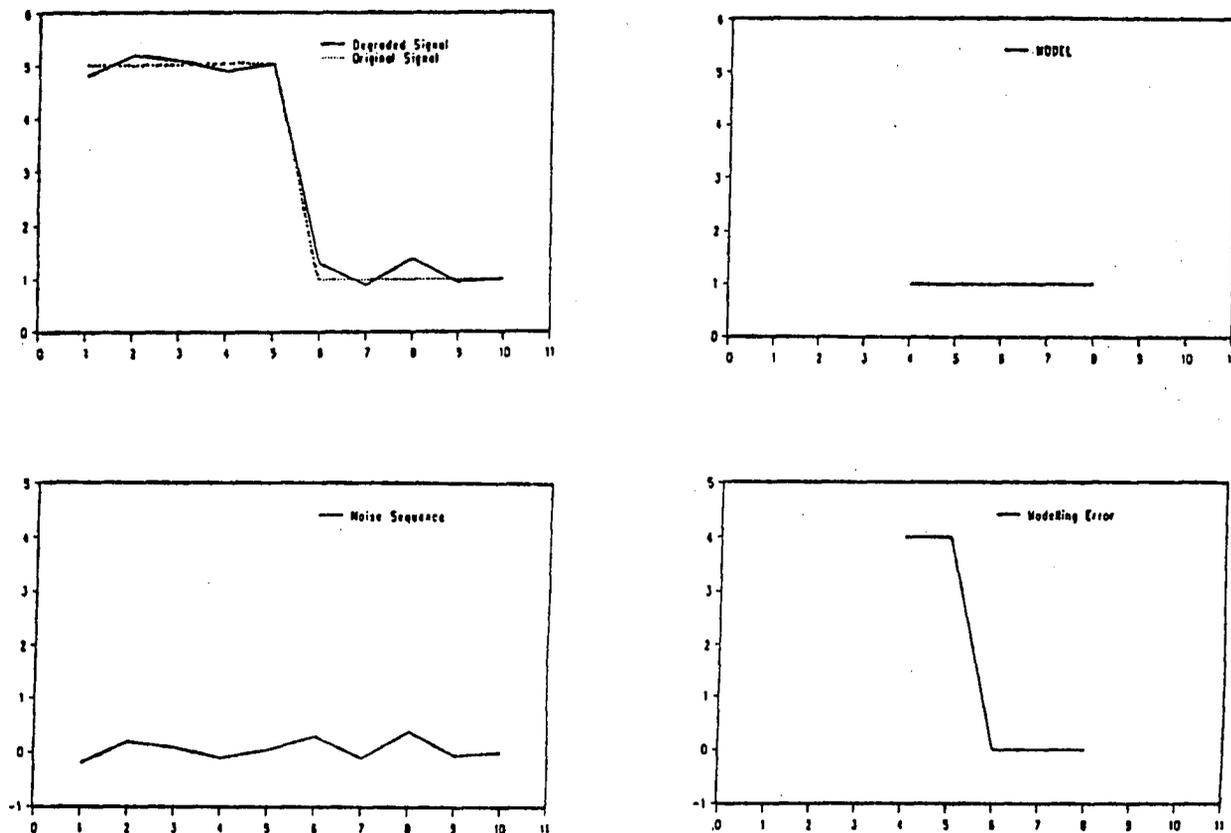


Figure 2.3 - (a) - original image and degraded image; (b) - model within a window containing pixels 4, 5, 6, 7 and 8; (c) - error sequence—additive white and non-impulsive; (d) - modelling error.

Figure 2.3 illustrates the modelling error for a one-dimensional step signal degraded by additive white non-impulsive noise. Only the window containing pixels 4, 5, 6, 7 and 8 is analyzed.

As illustrated by Figure 2.2 the running mean filter's blurring effect may cause a

degradation of an image which may exceed the noise degradation. Because the blurring effect can be traced to the signal model of (2.4), it follows that this model is inappropriate. On the other hand, the difficulty with model 2.4 is shared by all deterministic image models.

Present-day noise filtering algorithms use data dependent weighted sums which are only indirectly based on model 2.4. These algorithms replace a pixel by a weighted sum of it and its immediate neighbors. Each weight is assigned proportionally to some measure of local confidence on the model of (2.4). Ideally, if at a given neighborhood the image is “fairly” smooth, the weights used within this neighborhood should all be equal; that is, the weighted sum becomes the local average. If, on the other hand, there is a sharp contrast within the neighborhood used for smoothing, great care must be taken to avoid blurring this contrast.

Chapters 3, 4, and 5 will define different measures of local model confidence used by current data dependent local smoothers.

2.2 A Measure of Performance: the Mean Square Error

As shown by Figures 2.1 and 2.2 the running mean filter does not only reduce the noise degradation but also creates an undesirable blurring effect. The previous section traced the cause of this blurring effect to the error of the model of (2.4). This section analyzes the mean square error of the result of applying any linear filter to an image degraded by additive noise as a function of the modelling error and the noise sequence. This analysis is then generalized for multiplicative noise and for non-linear filters.

Because of the modelling error η_{ij} , if the running mean filter is applied to an image which has not been degraded by noise, the filter output still differs from the original image. That is: $\hat{x}_{ij} \neq x_{ij}$. This inequality also holds for all other linear smoothers, except

the identity.

If only linear filters are considered the property of superposition can be applied. Because of this property, the output of a linear filter can be separated as follows:

$$\hat{z}_{ij} = \text{estimate of } x_{ij} = \vec{h}_{ij} \otimes (\vec{x}_{ij} + \vec{e}_{ij}) = \vec{h}_{ij} \otimes \vec{x}_{ij} + \vec{h}_{ij} \otimes \vec{e}_{ij} = \hat{x}_{ij} + \hat{e}_{ij} \quad (2.7)$$

$$\text{where: } \begin{cases} \otimes & \text{denotes convolution;} \\ \hat{x}_{ij} & \text{is the estimate of } x_{ij} \text{ if there were no degradation;} \\ \hat{e}_{ij} & \text{is the noise residual after filtering;} \\ h_{ij} & \text{is the impulse response of the filter.} \end{cases}$$

The variance of \hat{e}_{ij} is well known to be:

$$\sigma_{\hat{e}}^2 = \left(\sum h_{ij}^2 \right) \sigma_e^2 \quad (2.8)$$

The performance of the noise removing filter cannot be judged by considering $\sigma_{\hat{e}}^2$ solely, because, as it was illustrated before: $\hat{x}_{ij} \neq x_{ij}$.

A better measure of filter performance lies in mean square (*MS*) error:

$$MS = (1/N) \sum (\hat{z}_{ij} - x_{ij})^2 \quad (2.9)$$

Where N is the total number of pixels in the image.

Thus:

$$MS = (1/N) \sum (\hat{x}_{ij} - x_{ij} + \hat{e}_{ij})^2 = (1/N) \sum (\hat{x}_{ij} - x_{ij})^2 + (1/N) \sum \hat{e}_{ij}^2 + (2/N) \sum ((\hat{x}_{ij} - x_{ij})\hat{e}_{ij}) \quad (2.10)$$

Because of (2.3), $(1/N) \sum ((\hat{x}_{ij} - x_{ij})\hat{e}_{ij}) \approx 0$

And $(1/N) \sum \hat{e}_{ij}^2 \approx \sigma_{\hat{e}}^2$ So:

$$MS \approx (1/N) \sum (\hat{x}_{ij} - x_{ij})^2 + \sigma_{\hat{e}}^2 \quad (2.11)$$

Or:

$$MS \approx (1/N) \sum (\hat{x}_{ij} - x_{ij})^2 + (\sum h_{ij}^2) \sigma_e^2 \quad (2.12)$$

For the running mean filter:

$$MS \approx (1/N) \sum (\hat{x}_{ij} - x_{ij})^2 + (1/W) \sigma_e^2 \quad (2.13)$$

(W is the number of pixels in the local neighbourhood)

Thus, by increasing W , σ_e^2 can be made arbitrarily small. This, in practice, causes the first term of the RHS of (2.13) to increase and eventually offset any reduction obtained in the second term. In other words, there is a tradeoff between noise reduction represented by the term $\frac{1}{W} \sigma_e^2$ and the term $(1/N) \sum (\hat{x}_{ij} - x_{ij})^2$. Figure 2.4 illustrates this.

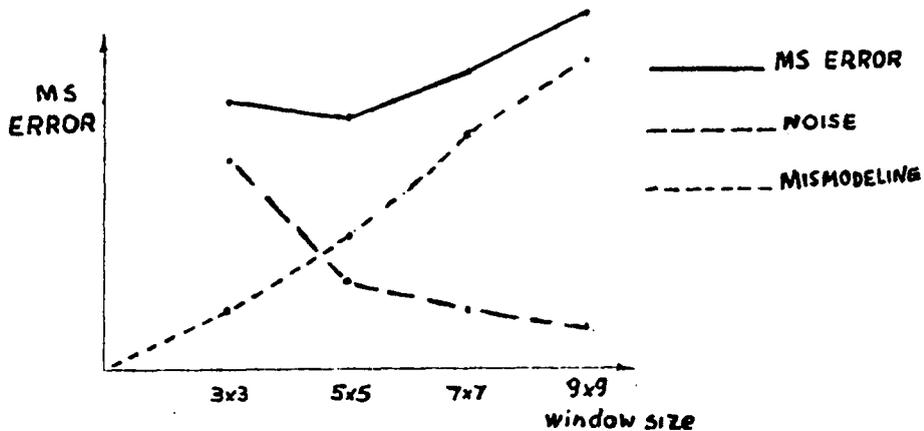


Figure 2.4 - Graph showing the tradeoff between the two terms of the RHS of (2.13).

It is clear that, to be most effective, image smoothing algorithms have to take the modelling error η_{ij} into account. This is a difficult task since η_{ij} is usually highly correlated to the extent that no statistical model can be applied to it. See the example in Figure 2.3.

The goal of current research is to improve the tradeoff between noise reduction and image blurring. This involves giving up the (least squares) optimum noise reducing properties

of the running mean filter in exchange for inhibition of modelling error. As described in the previous section, present-day noise removing algorithms use data dependent weighted sums for estimating the image. Inhibition of modelling error is achieved by making the weights proportional to the confidence of the model of (2.4).

Consider the multiplicative noise model of (2.2).

(2.2) can be rewritten as:

$$z_{ij} = x_{ij} + x_{ij}(\nu_{ij} - 1) \quad (2.14)$$

Define $\epsilon_{ij} = x_{ij}(\nu_{ij} - 1)$

Note that because of (2.3): $E\{\epsilon_{ij}\} = 0$

And also

$$E\{x_{ij}^2(\nu_{ij} - 1)\} = 0 \quad (2.15)$$

If the filter is linear:

$$E\{x_{ij}^2(\nu_{ij} - 1)\} = 0 \Rightarrow E\{((\hat{x}_{ij} - x_{ij})\hat{\epsilon}_{ij})\} = 0 \Rightarrow (1/N) \sum ((\hat{x}_{ij} - x_{ij})\hat{\epsilon}_{ij}) \approx 0 \quad (2.16)$$

Using (2.10):

$$MS \approx (1/N) \sum (\hat{x}_{ij} - x_{ij})^2 + \sum \hat{\epsilon}_{ij}^2 \quad (2.17)$$

Unlike (2.12), simplification of (2.17) is not possible; both terms of the RHS of (2.17) must be computed from actual images.

2.3 Analysis of the Mean Square Error for Non-Linear Filters

Here it will be shown that a revised form of (2.17) can be used for the analysis of the performance of non-linear filters. It is assumed that all existing non-linear local noise re-

moving filters can be described as data dependent weighted sums. (This will be elaborated upon in the following chapters.)

Given a vector \vec{z}_{ij} of pixels in the neighbourhood surrounding (and including) a center pixel z_o each non-linear (or linear) filtering scheme generates a weighting vector \vec{w}_{ij} to estimate z_o :

$$\hat{z}_o = \vec{w}_{ij} \bullet \vec{z}_{ij} \quad (2.18)$$

with the constraint:

$$\sum w_{ij} = 1 \quad (2.19)$$

Note that the model of (2.14) can be represented by (2.1) without any loss of generality by equating:

$$\epsilon_{ij} = x_{ij}(\nu_{ij} - 1)$$

Thus (2.18) becomes:

$$\hat{z}_o = \vec{w}_{ij} \bullet (\vec{x}_{ij} + \vec{\epsilon}_{ij}) \quad (2.20)$$

$$\text{Define: } \begin{cases} \tilde{x}_o = \vec{w}_{ij} \bullet \vec{x}_{ij} \\ \tilde{\epsilon}_o = \vec{w}_{ij} \bullet \vec{\epsilon}_{ij} \end{cases} \quad (2.22)$$

Then:

$$\hat{z}_o = \tilde{x}_o + \tilde{\epsilon}_o \quad (2.22)$$

Note that, unless the filter is linear:

$$\begin{cases} \tilde{x}_o \neq \hat{x}_o \\ \tilde{\epsilon}_o \neq \hat{\epsilon}_o \end{cases}$$

where \hat{x}_o and $\hat{\epsilon}_o$ are the results of filtering the original signal and the noise separately.

The mean square error of a non-linear filter is:

$$MS = (1/N) \sum (\hat{z}_{ij} - x_{ij})^2 = (1/N) \sum (\tilde{x}_{ij} - x_{ij})^2 + (1/N) \sum \tilde{\epsilon}_{ij}^2 + (2/N) \sum ((\tilde{x}_{ij} - x_{ij})\tilde{\epsilon}_{ij}) \quad (2.23)$$

terms of the RHS of (2.23) play a role in the mean square error. This is of importance in that it shows that the simple model of (2.4), which can only predict the term $(1/N) \sum(\tilde{\epsilon}_{ij}^2)$, is an inappropriate tool for analyzing filter performance.

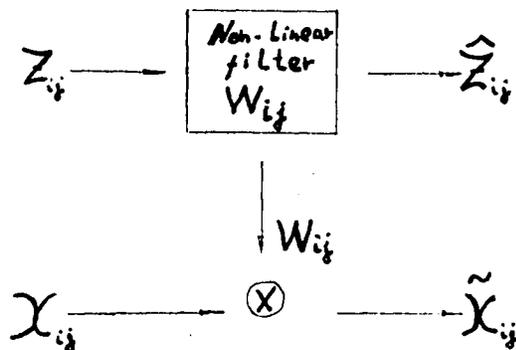


Figure 2.5 - Method for extracting \tilde{x}_{ij}

An apparent difficulty with (2.23) is that it involves the computation of $\tilde{\epsilon}_{ij}$ and \tilde{x}_{ij} . A simple algorithm for obtaining $\tilde{\epsilon}_{ij}$ and \tilde{x}_{ij} is based upon (2.22); it records the weighting vector (\tilde{w}_{ij}) used by the non-linear filter and applies this vector directly on x_{ij} or ϵ_{ij} . See Figure 2.5.

Using this algorithm the three terms of the RHS of (2.23) can be computed separately. This will be done in Appendix A 5 for several non-linear algorithms.

Chapter 3: Robust Filters

Based on (2.4), it is predicted in Chapter 2 that the running mean filter is the (least squares) optimum local noise removing filter. However, the application of the running mean filter is shown (Figure 2.2) to cause very undesirable image blurring. This chapter describes filters which provide sharper images than the running mean filter and other linear filters. This greater sharpness is achieved through better edge preservation but at the cost of thin line and spot suppression.

Robust filters can be described as data dependent weighted sum estimators. The common denominator of these filters is that, prior to smoothing, they eliminate outliers within the smoothing window. This chapter shows how outlier elimination leads to an improved edge preservation and makes these filters capable of handling impulsive noise. Furthermore, this chapter also shows that robust algorithms tend to eliminate thin lines and subtle detail.

The filters presented in this chapter are:

- * The median filter.
- * The ordered statistics filter (OSF)
- * The standard trimmed mean filter (STMF)
- * The m-smoother
- * The double window modified trimmed mean filter (DWMTMF)

It will be shown in Chapter 6 that of these filters the DWMTMF seems best. Original variations of the DWMTMF including a new version that can handle multiplicative noise are introduced.

3.1 The Median Filter

The median filter outputs at each pixel the median grey level of a neighborhood of the pixel. The median is a robust estimator in that it clearly discards outliers; consequently, the median is very effective in eliminating impulse noise. Unfortunately, this means that the median removes all image features which can be improperly identified as impulse noise, these include thin lines, as well as highlights and lowlights in images. In particular all thin lines whose width is $N/2$, N being the size of the median window, are removed.

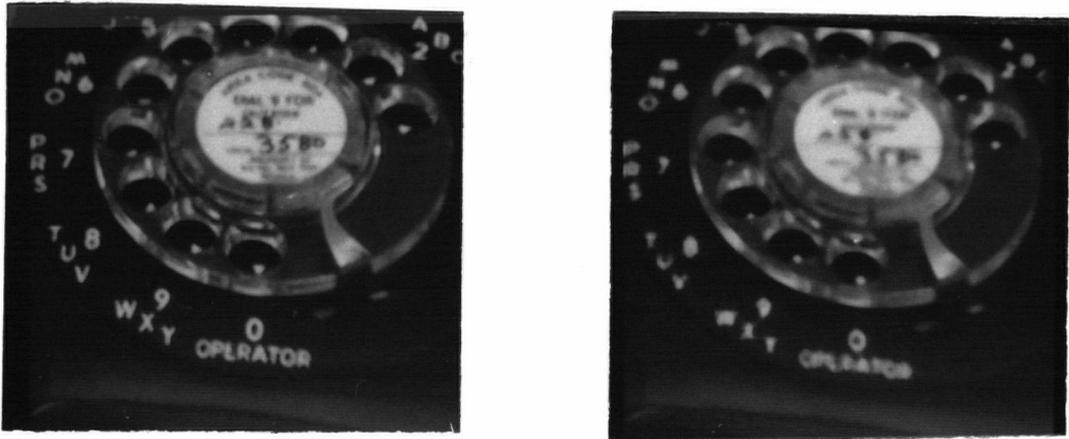


Figure 3.1 - (a) - "Telephone" of Figure 2.2 smoothed with median filter with window size 3x3; (b) - same as above but with window size 5x5

The median filter is excellent at preserving edges; in fact, an ideal step is completely preserved by the median. The median, however, removes corners when applied using a square window. See Figure 3.2.

Superior to the running mean filter at preserving edges and removing impulsive noise, the median also achieves minor smoothing of non-impulsive noise components. In fact, while the running mean is optimum in a least squares sense, the median filter is optimum

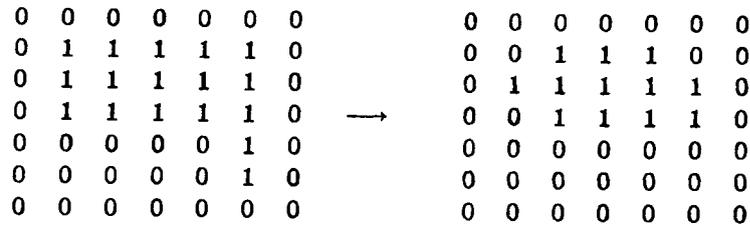


Figure 3.2 - (a) - original image; (b) - output after smoothing with a 3x3 median filter.

in a minimum average absolute deviation sense. Average absolute deviation is defined by:

$$d = (1/W) \sum |z_{ij} - \hat{z}_o| \quad (3.1)$$

where z_{ij} belongs to a neighborhood of z_o , \hat{z}_o is a local estimate of z_o and W is the number of pixels in the neighborhood.

However, the median filter's smoothing capabilities will often not suffice. This has led to much research into how to obtain the smoothing capabilities of the running mean filter while maintaining the edge preservation properties of the median filter and its ability to handle multiplicative noise. Some of these methods will be discussed in the remainder of the chapter.

3.2 The Ordered Statistics Filter (OSF)

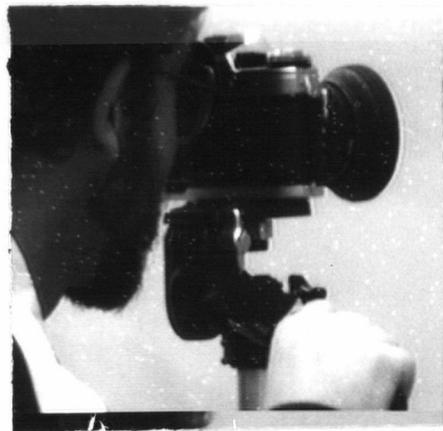
The OSF was recently introduced by Bovic, Huang and Munson [2]. The OSF outputs at each pixel a linear combination of the ordered values in the pixel's neighborhood. Consider a pixel z_o and let \hat{z}_o be its estimate; let x_1, \dots, x_n be the ordered sequence of pixels in a given neighborhood of z_o . Then:

$$z_o = \sum w_i x_i \quad (3.2)$$

with the constraint: $\sum w_{ij} = 1$; where w_1, \dots, w_n is a predetermined set of weights.



(a)



(b)



(c)



(d)

Figure 3.3 - (a) - original "Camera" image; (b) - Camera image degraded by "salt" noise; (c) - noise removed using a median filter of size 3x3; (d) - noise removed using a median filter of size 5x5;

By proper choice of these weights the OSF can be tuned to be a running mean filter, a median filter or a min(max) filter. Note that unless the center weight is one and all others zero, ideal edges will be smeared. Indeed, good edge preservation requires that most of the weight be placed on the center pixel. This can be shown to also achieve suppression of impulse noise components.

On the other hand, to achieve adequate smoothing of non-impulsive noise components the weights must be as evenly spread as possible. Bovik *et al* give tables containing

optimum weights for different noise distributions. Their work is limited to additive noise and deals mainly with Model 2.4, so its interest here is limited.



(a)



(b)



(c)



(d)

Figure 3.4 - (a) - "Telephone" degraded by AWGN with $\sigma_e^2 = 100$; (b) noise removed using window size 9 (3x3) running mean filter; (c) noise removed using window size 9 (3x3) median; (d) noise removed using window size 9 (3x3) OSF with weights: (0, 0, 0, 1/3, 1/3, 1/3, 0, 0, 0).

3.3 M-Smoother

An estimate \hat{z}_o inside a neighborhood of z_o is defined by the solution of the equation:

$$\sum \Psi(z_{ij} - \hat{z}_o) = 0 \quad (3.3)$$

where Ψ is an odd, continuous, increasing function.

An m-smoother outputs at each pixel an m-estimate of the gray level of a neighborhood of the pixel. M-smoothers are superior to OSF's in that they are more adaptive to the signal, giving a better tradeoff between edge preservation and non-impulsive noise smoothing. As with the OSF the m-smoother can be tuned to be either a running mean filter (making $\Psi(x) = ax$, $a > 0$), or a median filter (making $\Psi(x) = sig(x)$.)

An interesting m-smoother subclass introduced by Y. H. Lee and S. A. Kassam [1] is the Standard Trimmed Mean Filter (STMF). The STMF is an m-smoother whose generating function Ψ satisfies:

$$\Psi(x) = \begin{cases} 1 & \text{for } x > p \\ x/p & \text{for } |x| \leq p \\ -1 & \text{for } x < -p \end{cases} \quad (3.4)$$

In the limiting cases as $p \rightarrow \infty$ the STMF becomes the running mean filter and as $p \rightarrow 0$ the STMF becomes the median filter.

STMF's differ from the OSF's in that the STMF's do not use ordering of data as their sole means of detecting outliers. Instead, the STMF uses the difference in grey level between the local median and the actual pixels. Consider an ideal one-dimensional step whose height satisfies:

$$p \ll H(N+1)/N$$

Y. H. Lee and S. A. Kassam have shown that an STMF with parameter p applied to this step tends to smooth only those samples lying on the side of the step containing the median value.

In general, the STMF has been reported to perform better than the OSF. See [1]. The superior performance of the STMF can be attributed to the STMF's greater ability for detecting outliers. Y. H. Lee and S. A. Kassam have shown that the STMF does not disregard outliers completely, but that it "hardlimits" samples to some range of the median $[m_i - q_1, m_i + q_2]$ where q_1 and q_2 are data dependent.

The STMF, as described by Y. H. Lee and S. A. Kassam can deal only with additive noise. It can be controlled through the parameter p and the size of the smoothing window. Typically the parameter p is twice the standard deviation of the noise:

$$p = 2\sigma_\epsilon \quad (3.5)$$

In the case of multiplicative noise the STMF can also be used but its parameter p must be made to float with the local image brightness. A suggested value for p is:

$$p = 2\sigma_\nu \hat{z} \quad (3.6)$$

Where σ_ν is the standard deviation of the noise and \hat{z} is an estimate of the image brightness. For example, \hat{z} can be the local median.

Y. H. Lee and S. A. Kassam have introduced the MTMF which is a modified version of the STMF which is easier to implement and is reported to perform as well or better.

3.4 The Modified Trimmed Mean Filter (MTMF)

Inside the sliding smoothing window the MTMF first obtains the median (m); then all pixels within the window satisfying $|m - z_{ij}| < p$ are averaged. In the case of additive noise, p is a predetermined constant which depends on the noise level. In the case of multiplicative noise p should be proportional to the image brightness. Typical values for p in the cases of additive and multiplicative noise are given by (3.5) and (3.6). Note that for $p = 0$ the MTMF becomes the median filter and for $p \rightarrow \infty$ it becomes the running mean filter.

In general, it is important to maintain the window size for computing the median as small as possible so that thin lines and subtle detail are preserved. On the other hand, averaging must be made in as large a window as possible so as to achieve acceptable smoothing of non-impulsive noise components. To achieve these seemingly contradictory goals Y. H. Lee and S. A. Kassam have introduced a “double window” version of the STMF which uses a separate window size for the median operation than for the subsequent averaging. Y. H. Lee and S. A. Kassam have named this new filter double window modified trimmed mean filter (DWMTMF).

The window size used for obtaining the median can be tuned to remove impulsive noise components while the parameter p and the window size for the averaging can be tuned to remove non-impulsive noise components.

The DWMTMF will preserve an ideal edge provided that $p \leq H$, where H is the height of the edge. If $p > H$ the edge is transformed into a ramp whose width is equal to the window size used for averaging.

In the case of additive noise, a typical value for p is $2\sigma_\epsilon$ (as with the STM filter), where σ_ϵ is the standard deviation of the noise.

A different scheme for inhibiting the influence of outliers is through the use of the following weighting scheme:

$$w_{ij} = \begin{cases} \frac{C}{threshold^n} & \text{for } |z_{ij} - median| \leq threshold \\ \frac{C}{(z_{ij} - median)^n} & \text{for } |z_{ij} - median| > threshold \end{cases} \quad (3.7)$$

where:

$$\sigma_\epsilon \leq threshold \leq 2\sigma_\epsilon \text{ for additive noise} \quad (3.8)$$

$$1 \leq n < \infty$$

C is such that :

$$\sum w_i = 1 \quad (3.9)$$



(a)



(b)



(c)



(d)

Figure 3.5 (a) - "Telephone" degraded by additive white Gaussian noise (AWGN) with $\sigma_\epsilon = 15.0$; (b) - noise removed of degraded image of Figure 3.4b with the m-smoother of window 5×5 ; (c) - noise removed with DWMTM filter with median window size of 3×3 and smoothing window size of 7×7 ; (d) - noise removed with 5×5 median

Note that as $n \rightarrow \infty$ the weighting scheme becomes a simple thresholding algorithm.

In the case of multiplicative noise the threshold p of the DWMTM must be made to float with the image intensity. One typical value for p for the case of multiplicative noise is given by (3.6). Replacing \hat{z}_o by the local median (*median*):

$$p = 2\sigma_p \text{median} \quad (3.10)$$

in which σ_ν is the standard deviation of the multiplicative noise.

For a combination of additive and multiplicative noise, one typical value for p is:

$$p = 2(\sigma_\nu \text{median} + \sigma_\epsilon) \quad (3.11)$$

For this general case of both additive and multiplicative noise components the threshold used for (3.7) is:

$$(\sigma_\nu \text{median} + \sigma_\epsilon) \leq \text{threshold} \leq 2(\sigma_\nu \text{median} + \sigma_\epsilon) \quad (3.12)$$

An improvement in the performance of the DWMTM smoother can be made by calling attention to its two steps:

STEP 1 - obtain a robust estimate of the center pixel.

STEP 2 - obtain a better estimate of the center pixel by performing a weighted sum within a smoothing window using (3.7)

Note that STEP 2 can be iterated to improve the signal estimate. The improvement due to this iterative scheme will be studied further in Chapter 6.

Chapter 4: Gradient-Based Filters

Chapter 2 describes how the zeroth order Model of (2.4) (also called Model 2.4) predicts the running mean filter to be the optimal least squares filter. However, in regions of high contrast the Model fails and the running mean causes severe image smearing.

The approach taken by current researchers ([1]-[3], [5]-[7]) to avoid this undesirable smearing effect is to replace the running mean filter indiscriminate averaging by a scheme using a data dependent weighted sum estimation. This scheme assigns a weight to each pixel within a smoothing window such that if the confidence in Model 2.4 is high at a certain pixel in the window the filter assigns it a large weight, conversely, if the confidence is low at a certain pixel its corresponding weight is small. In other words, each weight is assigned proportionally to some measure of local confidence in Model 2.4. The filters discussed in this chapter are said to be gradient based because an absolute gradient is used as a measure of the local confidence in Model 2.4.

Consider some “small” neighborhood used for estimating the gray level of the pixel z_o . Because of (2.5) and (2.1), for additive noise:

$$z_{ij} = x_o + \eta_{ij} + \epsilon_{ij} \quad (4.1)$$

And for multiplicative noise:

$$z_{ij} = (x_o + \eta_{ij})\nu_{ij} \quad (4.2)$$

The measure for local confidence in the model of (2.4) is the absolute value of the difference $z_{ij} - z_o$, also called absolute gradient. Using (2.6):

$$z_{ij} - z_o = \eta_{ij} + \epsilon_{ij} - \epsilon_o \quad (4.3)$$

for additive noise; and:

$$z_{ij} - z_o = \eta_{ij}\nu_{ij} + x_o(\nu_{ij} - \nu_o) \quad (4.4)$$

for multiplicative noise.

If the noise sequences ϵ_{ij} and ν_{ij} satisfy the conditions of (2.3); then:

$$E\{(\epsilon_{ij} - \epsilon_o)^2\} = 2\sigma_\epsilon^2 \quad (4.5)$$

and:

$$E\{x_o^2(\nu_{ij} - \nu_o)^2\} = 2x_o^2\sigma_\nu^2 \quad (4.6)$$

If $|z_{ij} - z_o|$ is large the confidence in Model (2.4) is low at pixel z_{ij} , thus the probability of mismodelling is high. On the other hand, if the $|z_{ij} - z_o|$ is small the probability of modelling error at z_{ij} is low. Note that there is no modelling error at the pixel z_o ; see (2.6).

Consider an estimate of z_o of the form:

$$\hat{z}_o = \sum_{ij} w_{ij} z_{ij} \quad \text{satisfying:} \quad \sum_{ij} w_{ij} = 1$$

To minimize the effect of mismodelling in the estimation, the weights assigned to each pixel must be in direct relation with the model confidence at that pixel. For this reason, gradient-based filters assign weights in an inverse relation with the local absolute gradient, that is:

$$w_{ij} = \mathfrak{F}(|z_{ij} - z_o|) \quad (4.7)$$

where \mathfrak{F} is a decreasing function defined for positive argument.

In this chapter, the sigma filter of J. S. Lee and the gradient inverse filtering Scheme by David C. Wand *et al* will be discussed along with a generalization of these techniques. Furthermore, the relationship of these techniques with the DWMTMF of Y. H. Lee and S. A. Kassam and a further generalization of the gradient-based methods to deal with impulsive noise are studied.

4.1 The Sigma Filter

The sigma filter, recently introduced by J. S. Lee, can be considered a special case of the DWMTM filter of Y. H. Lee and S. A. Kassam where the window size for computing the median has been collapsed to just one pixel. That is, the first step of the DWMTM filter—to obtain a robust initial estimate—has been eliminated. This allows the sigma filter to preserve thin lines and other subtle details that are suppressed by the application of the median operation.

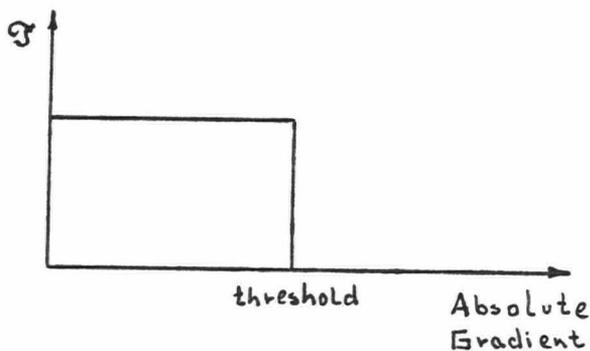


Figure 4.1 - The \mathfrak{S} function corresponding to the sigma filter.

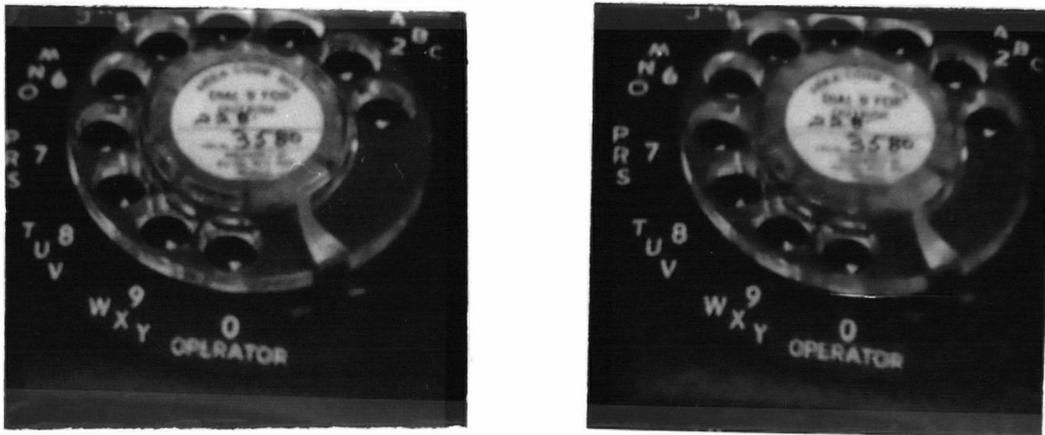


Figure 4.2 - (a) - noise removed image of Figure 2.2b using a sigma filter of size 7×7 and $p = 12.5$; (b) - noise removed using a DWMTM filter with window sizes 3×3 and 7×7 and threshold $p = 12.5$.

The sigma filter is a gradient-based smoother for which the inverse relation function satisfies:

$$\mathfrak{S} : \begin{cases} C & \text{for } |z_{ij} - z_o| < p \\ 0 & \text{for } |z_{ij} - z_o| \geq p \end{cases} \quad (4.8)$$

More specifically, the sigma filter outputs at a pixel z_o the average of pixels, within a neighborhood of z_o , whose grey level satisfies $|z_{ij} - z_o| < p$. p has the same typical values for the corresponding parameter of the DWMTM filter namely:

$$p = \begin{cases} 2\sigma_\epsilon & \text{for additive noise;} \\ 2z_o\sigma_\nu & \text{for multiplicative noise;} \\ 2(\sigma_\epsilon + z_o\sigma_\nu) & \text{for a combination of additive and multiplicative noise.} \end{cases} \quad (4.9)$$

Note that (4.9) differs from the corresponding formula for the DWMTM filter in that the level estimation is the actual intensity of the center pixel.

The sigma filter as it has been presented has several drawbacks: The sigma filter is not as robust as the DWMTM because of the lack of the median estimation. The center pixel could itself be an outlier, in which case no smoothing will take place. Thus, the sigma filter cannot filter out impulsive noise components. The filter also has difficulties with outliers of some non impulsive noise components (like additive white Gaussian noise).

The lack of robustness of the sigma filter can be even more damaging for the case of multiplicative noise. This is due to the fact that the threshold used is also dependent upon the center pixel's brightness.

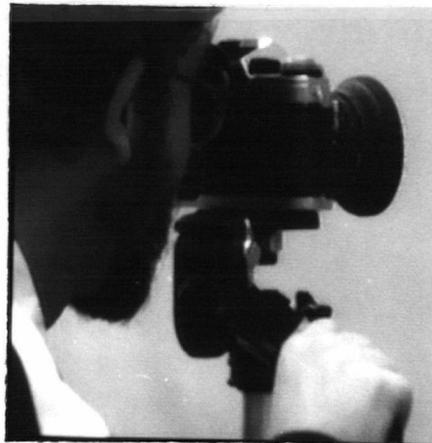
As with the DWMTM filter, it is again possible to produce an iterative procedure for improving the performance of the sigma filter. This iterative procedure would feed back the output estimate of the sigma filter (\hat{z}_o) and smooth all other pixels in the smoothing window which satisfy $|z_{ij} - \hat{z}_o| < p$. The new result can be fed back for another iteration. A study of the performance improvement due to this iterative scheme is shown in Chapter 6.



(a)



(b)



(c)

Figure 4.3 - (a) - noise removed from image of Figure 3.3b using a sigma filter of window size 7×7 and threshold $p = 3.0$; (b) - noise removed using a running mean filter of window size 3×3 ; (c) - noise removed using the generalized filter of Section 4.4.

4.2 The Gradient Inverse Filter

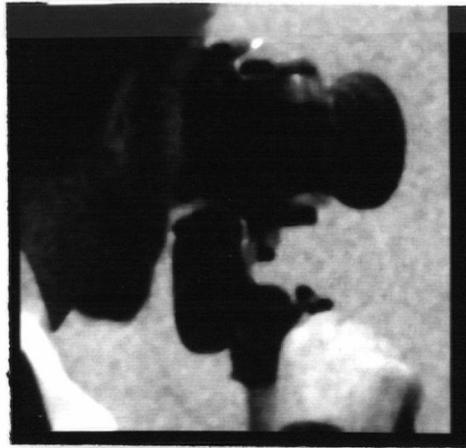
Gradient inverse filtering was introduced by David C. C. Wand, A. H. Vagucci and C. C. Li. Their algorithm uses a 3×3 smoothing window where the center pixel is estimated by a weighted sum of the 9 pixels in the window. The weighting coefficients, as the name of the algorithm implies, are the normalized gradient inverses. Therefore, the inverse relation function \mathfrak{S} is an inverse proportionality. See Figure 4.5.



(a)



(b)



(c)

Figure 4.4 - (a) - "Camera" degraded by AWGN with $\sigma = 15$; (b) - noise removed using a sigma filter with window size 7×7 and threshold $p = 37.5$; (c) - noise removed using a median filter with window size 5×5 .

More specifically the gradient inverse weighting algorithm uses the following weights:

$$w_{ij} = \begin{cases} 2C & \text{for } z_{ij} = z_0 \\ \frac{C}{|z_{ij} - z_0|} & \text{for } z_{ij} \neq z_0 \end{cases} \quad (4.10)$$

where C is such that: $\sum_{ij} w_{ij} = 1$

The method is simple but the edge preservation procured is not as good as that of the sigma filter. Furthermore, as pointed out by the authors themselves, to obtain sufficient

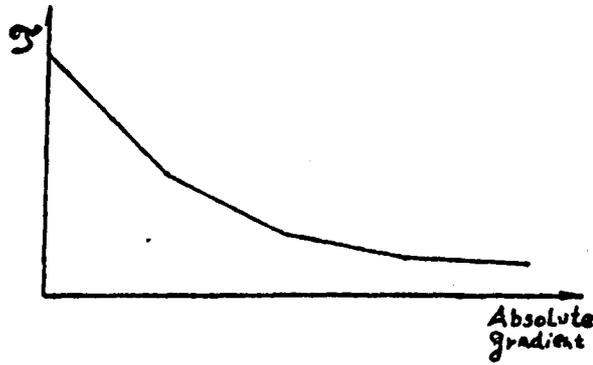


Figure 4.5 - The \mathfrak{S} function of the gradient inverse weighting technique

noise smoothing it is necessary to apply this method several times. As reported by J. S. Lee, this method is not as effective as the sigma filter in edge preservation and noise smoothing properties.

4.3 A Generalized Gradient-Based Filter

As with the DWMTM Filter the sigma filter's weighting scheme can be generalized through the use of a weighting scheme similar to that of (3.7):

$$w_{ij} = \begin{cases} \frac{C}{\text{threshold}^n} & \text{for } |z_{ij} - z_o| \leq \text{threshold} \\ \frac{C}{|z_{ij} - z_o|^n} & \text{for } |z_{ij} - z_o| > \text{threshold} \end{cases} \quad (4.11)$$

where: *threshold*, *n*, and *C* are described in (3.8) and (3.9).

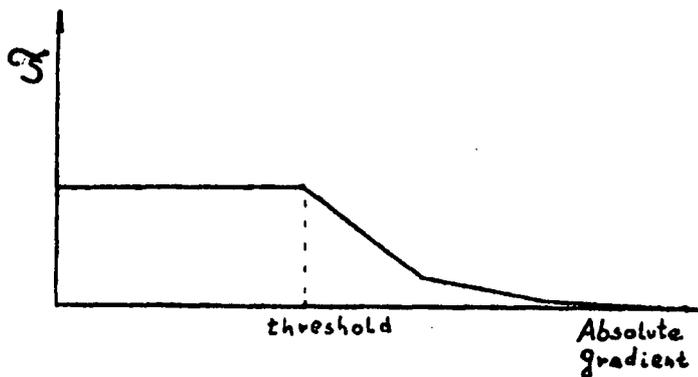


Figure 4.6 - The \mathfrak{S} function of the generalized gradient-based filter

Observe that for $n \rightarrow \infty$ the generalized gradient-based filter becomes the sigma filter.

Also for $n = 1$ and $threshold = 0.5$ the generalized gradient-based filter becomes the gradient inverse weighting filter. The generalized method, then, contains a rich variety of possible filters including both the sigma filter and the gradient inverse filter.

As with the DWMTM filter, an iterative scheme can also be developed for the generalized method. The presentation of such a scheme is presented in the next section.

In the next section a higher level of generalization will be achieved through the use of outlier elimination prior to smoothing.

4.4 A General Formulation for Gradient-Based Methods and the DWMTM Filter

In Chapter 3, a generalized form of the DWMTM filter is presented which is described in two steps. Step 1 consists of a robust estimation of the center pixel using the median filter. Step 2 does the actual smoothing using the weighting scheme of (3.7). Note the resemblance between (3.7) and (4.11). In fact, the generalized gradient-based filter can be described as a simplified form of the DWMTM Filter.

The use of the median filter in step 1 of the DWMTM is shown in Chapter 3 to cause suppression of thin lines and other subtle details in images. J. S. Lee has proposed a simple scheme for removing outliers which appears to have fewer drawbacks than the median filter. The scheme consists of replacing a pixel by a neighborhood estimation if it is judged to be an outlier; otherwise it remains unchanged.

To decide if a pixel is indeed an outlier the number of pixels x_{ij} in a neighborhood of x_o such that $|x_{ij} - x_o| < threshold$ is counted. (In general the threshold is given by (4.9).) For a 3x3 neighborhood, if the count is less than two the point x_o is judged to be an outlier. For a 5x5 neighborhood the count must be less than three for x_o to be judged an outlier. If x_o is judged to be an outlier it must be replaced by some interpolation technique. J.S. Lee suggests an average of the 8 neighboring points. A better alternative is to use the 3x3 (or

possibly 5x5) median. Note that if any count is accepted the method becomes the median filter.

By using the above outlier removal scheme instead of the median for Step 1 of the generalized formulation of the DWMTM filter, a new composite method is obtained which includes the DWMTMF as a particular case. The method is as follows:

STEP 1: Apply outlier removal scheme to obtain an initial estimate (\hat{z}_o) of the center pixel.

STEP 2: Apply the weighting scheme of (4.11). (Where instead of the center pixel \hat{z}_o is used.)

STEP 2 can be iterated as many times as desired to improve the signal estimate. The improvement due to this iterative scheme will be studied further in Chapter 6.

Chapter 5: Variance-Based Filters

Chapter 4 has described the use of the absolute gradient in several present day noise filtering algorithms. It has been shown, in Chapter 4, that the absolute gradient gives a measure of modelling error. This chapter introduces another measure of modelling error known as sample variance. As it is shown below, the sample variance has the very desirable advantage upon the absolute gradient in that it is less sensitive to noise. However, the usefulness of the sample variance is seriously hampered by its poor localization of modelling error. This chapter analyzes this difficulty of the use of the sample variance and presents an original solution to the localization problem.

The local sample variance is defined by:

$$S = \frac{1}{W-1} \sum_{ij} \left(z_{ij} - \frac{\sum_{ij} z_{ij}}{W} \right)^2 \quad (5.1)$$

or:

$$S = \frac{1}{W-1} \sum_{ij} z_{ij}^2 - \frac{1}{W-1} \left(\frac{\sum_{ij} z_{ij}}{W} \right)^2 \quad (5.2)$$

where W is the number of elements used for computing the variance.

For the special case of $W = 2$, and pixels z_{ij} and z_o ; S becomes:

$$S = \frac{|z_{ij} - z_o|^2}{2} \quad (5.3)$$

The sample variance S applied to just two points (5.3), is very similar to the absolute gradient of Chapter 4; moreover, (5.3) gives an equivalent measure of the modelling error at pixel z_{ij} . However for $N > 2$ the sample variance cannot be said to give a measure of the modelling error at any particular pixel, rather, the sample variance gives a neighborhood measure of modelling error. That is, the sample variance, except for the trivial case of $N = 2$, cannot localize modelling error.

On the other hand, the use of more than two pixels for obtaining the sample variance makes it less sensitive to noise than the absolute gradient. So, localization of modelling error is given up in exchange for lower noise sensitivity. A reduced noise sensitivity is highly desirable because it allows for the detection of fainter image edges.

Note that as with the gradient-based techniques and for similar reasons it is necessary to assume here that the noise is non-impulsive. Prefiltering of outliers is a must if the noise is impulsive; in this case, similar techniques as those in Section 3.4 are necessary.

As a validation for variance-based techniques consider the following example where modelling error has been likened to a non-stationary noise source:

$$\text{Consider the model: } z_{ij} = x + e_{ij} \quad (5.4)$$

$$\text{where: } \begin{cases} e_{ij} \text{ is a nonimpulsive noise sequence;} \\ E\{e_{ij}\} = 0 \\ E\{e_{ij}^2\} = \sigma_{ij}^2 \\ E\{e_{ij}e_{mn}\} = 0 \quad \forall (i, j) \neq (m, n) \\ E\{e_{ij}x_{mn}\} = 0 \quad \forall i, j, m, n \end{cases} \quad (5.5)$$

Consider a sliding smoothing window and that an estimate of x is given as a weighted sum pixels within the smoothing window. See (2.20):

$$\hat{z}_o = \sum_{ij} w_{ij} z_{ij} \quad (5.6)$$

For a least squares estimate of z_o the following cost function must be minimized:

$$C = E\{(z_o - x)^2\} \quad (5.7)$$

$$\begin{aligned} C &= E \left\{ \left(\sum_{ij} w_{ij} z_{ij} - x \right)^2 \right\} \\ C &= E \left\{ \left(\sum_{ij} w_{ij} x - x + \sum_{ij} w_{ij} e_{ij} \right)^2 \right\} \end{aligned} \quad (5.8)$$

If the estimate is to be unbiased:

$$\sum_{ij} w_{ij} = 1 \quad (5.9)$$

From (5.8) and (5.9):

$$C = E \left\{ \left(\sum_{ij} w_{ij} \epsilon_{ij} \right)^2 \right\} = \sum_{ij} w_{ij}^2 \sigma_{ij}^2$$

Thus:

$$C = \sum_{ij} w_{ij}^2 \sigma_{ij}^2$$

must be minimized with the constraint (5.9)

Using Lagrange multipliers:

$$q = \sum_{ij} (w_{ij}^2 \sigma_{ij}^2) + \lambda \left(1 - \sum_{ij} w_{ij} \right)$$

$$\frac{\partial q}{\partial w_{ij}} = 2w_{ij} \sigma_{ij}^2 - \lambda = 0$$

$$\lambda = 2\sigma_{ij}^2 w_{ij}; \quad w_{ij} = \frac{\lambda}{2\sigma_{ij}^2}$$

$$\sum_{ij} w_{ij} = \frac{\lambda}{2} \sum_{ij} \frac{1}{\sigma_{ij}^2}$$

$$\lambda = 2 \left(\sum_{ij} \frac{1}{\sigma_{ij}^2} \right)^{-1}$$

$$\text{Thus: } w_{mn} = \left(\frac{1}{\sigma_{mn}^2} \right) / \left(\sum_{ij} \left(\frac{1}{\sigma_{ij}^2} \right) \right) \quad (5.10)$$

e_{ij} is intended here to model the effects of both the noise and the modelling error:

$$e_{ij} = \epsilon_{ij} + \eta_{ij} \quad \text{for additive noise; and} \quad (5.11)$$

$$e_{ij} = \nu_{ij} \eta_{ij} + (\nu_{ij} - 1)x_o \quad \text{for multiplicative noise} \quad (5.12)$$

According to the model of (5.4) an inverse quadratic proportion between the local variance and the weights is optimal. However, as it is pointed out in Chapter 2, it is

not possible to model η_{ij} and, furthermore, η_{ij} is usually highly correlated so that the assumptions for (5.4) are invalid. The intention here for developing (5.10) is to motivate the methods described in this chapter.

The use of the variance as a means of detecting η_{ij} can be illustrated through the use of a local sample variance map. A sample variance map replaces each pixel by the sample variance of a small neighborhood (usually 3x3 or 5x5 squares) of the pixel. Figure 5.1 illustrates the sample variance of an image with and without additive noise. Note that large values of the modelling error η_{ij} due to edges and high contrast in general create highlights in the local sample variance map.

To illustrate the use of the variance map for dealing with multiplicative noise, a local variance map of an image degraded by multiplicative noise with $\sigma_v = .2$ is shown in Figure 5.2. The displayed grey level is normalized as follows:

$$d_{ij} = 255 \left(\sqrt{\frac{S_{ij}}{\bar{z}_{ij}}} \right)$$

$$\text{where: } \begin{cases} \bar{z}_{ij} & \text{is the mean intensity in the local neighborhood of } z_{ij}; \\ d_{ij} & \text{is the normalized grey level displayed;} \\ S_{ij} & \text{is the local sample variance in the local neighborhood of } z_{ij}; \\ 255 & \text{is a scale factor used to obtain a suitable dynamic range in the display} \end{cases}$$

Thus, through the use of the local variance map it is possible to detect the most outstanding modelling errors. These coincide with image edges, thin lines and spots.

Three filters are presented in this chapter that make use of the sample variance for smoothing images while preserving edges. The filters presented are: Nagao's smoother, J.S. Lee's local statistics smoother and an original smoother introduced here.

5.1 Nagao's Smoother

As described in the previous section the local sample variance gives a measure of the modelling error; in other words, the lower the variance the higher the confidence in the model described by (2.4). Nagao and Matsuyama have introduced an algorithm which

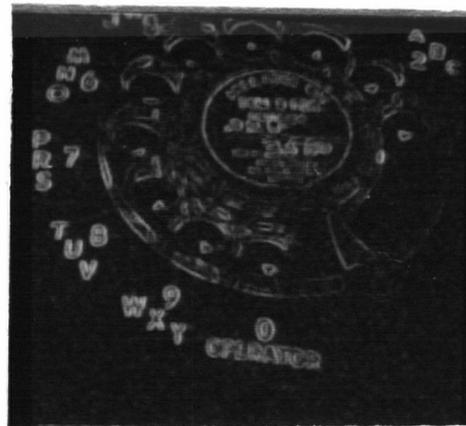
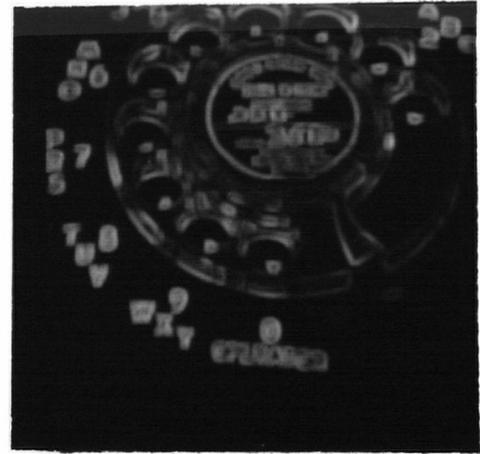
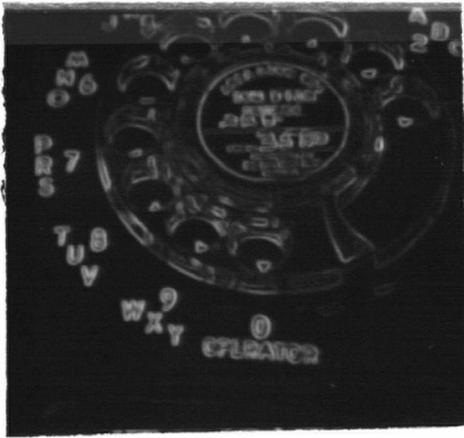


Figure 5.1 - (a) - variance map of the "Telephone" using a 3x3 window; (b) - variance map of the "Telephone" using a 5x5 window; (c) - variance map of the degraded "Telephone" of Figure 3.4 using a 3x3 window.

uses a 5x5 sliding window. This sliding window is divided into nine overlapping regions. See Figure 5.4.

The means and variances of the nine subregions are computed, and the centre pixel replaced by the mean of the subregion having the least variance. Nagao's filter will round off corners of less than 90 degrees; it will eliminate thin lines; it will not eliminate impulse noise but it will reduce the impulses considerably.

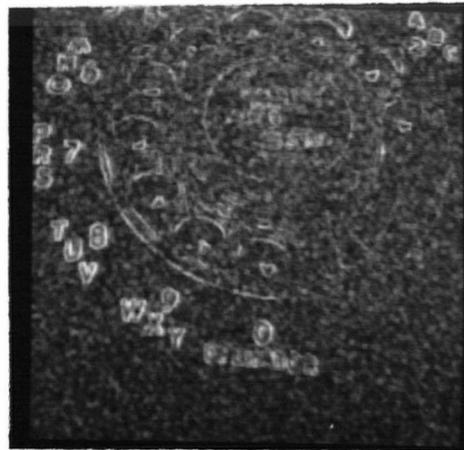


Figure 5.2 - variance map of the "Telephone" degraded by MWGN with $\sigma_v = .2$ using a 3x3 window.

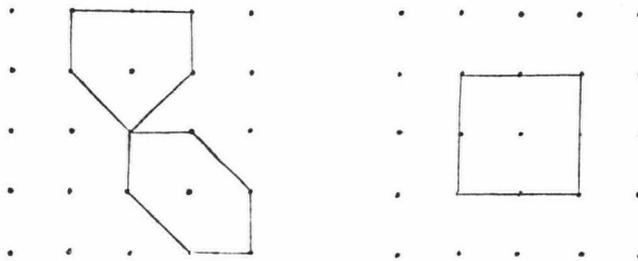


Figure 5.3 - Three of the subregions checked by Nagao's filter.

An important drawback of Nagao's filter is the need of computing the variances for all the nine subregions per image pixel. Furthermore, J. S. Lee has shown through simulation that the noise smoothing performance of Nagao's filter is at par with that of the 3x3 median filter, which requires much less computation. Note that some of the undesired properties of this filter are due to the fact that, for regions of the image with subtle detail or thin lines, all of Nagao's subregions will have a high variance. That is, the chosen subregion for smoothing may itself contain high contrast and thus be undesirable for smoothing. To preserve fine detail, Nagao's filter should preserve the value of the centre pixel when it is not able to find a suitable subregion for smoothing. The two other filters in this chapter

will offer such an option.

5.2 J. S. Lee's Local Statistics Smoother

J.S. Lee has introduced a simple but powerful adaptive method for smoothing images degraded by additive noise, multiplicative noise or both. The main idea behind J.S. Lee's method is to replace a pixel's grey level by its weighted sum with the neighborhood average gray level; i.e. the estimate is:

$$\hat{z}_o = kz_o + (1 - k) \sum_{ij} \frac{z_{ij}}{W} \quad (5.13)$$

where: $\begin{cases} W & \text{is the number of pixels in the neighborhood} \\ k & \text{is a data dependent parameter which will be determined below} \end{cases}$



Figure 5.4 - noise removed from degraded image of Figure 3.5a using J. S. Lee's local statistics filter.

Consider first the case of non-impulsive additive white noise:

$$z_{ij} = x_o + \epsilon_{ij} + \eta_{ij} \quad (5.14)$$

To find an appropriate value of the parameter k consider that the noise ϵ_o is approximately independent of the mean noise $\bar{\epsilon}$ within the smoothing window. (Actually $E\{\epsilon_o \bar{\epsilon}\} = \sigma_\epsilon^2/N$).

Furthermore, note that there is no mismodelling in z_o so that $E\{(z_o - x_o)^2\} = \sigma_\epsilon^2$ and that there may be mismodelling in the mean. Assume that the local sample variance approximates the true variance of the local mean. Thus:

$$\sigma_{\bar{z}}^2 = S_{ij} \tag{5.15}$$

using (5.10) for two measurements z_{ij} and \bar{z} with variances σ_ϵ^2 and S_{ij} :

$$k = \left(\frac{1}{\sigma_\epsilon^2} \right) / \left(\frac{1}{\sigma_\epsilon^2} + \frac{1}{S_{ij}} \right)$$

$$k = \frac{S_{ij}}{S_{ij} + \sigma_\epsilon^2} \tag{5.16}$$

Note that $k \in [0, 1]$. For a low contrast area S_{ij} is small and $\hat{z}_o \approx \bar{z}$; whereas for an image edge or a high contrast area S_{ij} is large and $\hat{z}_o \approx z_o$;



Figure 5.5 - noise removed from the degraded image of Figure 2.2b by using J. S. Lee's local statistics filter with window size 5x5;

This algorithm produces good edge preservation but fails to smooth noise near image edges. Also, this algorithm is not meant for impulse noise—impulse noise is actually likely to be enhanced by the algorithm. Note that unlike Nagao's filter this algorithm can be

tuned to the amount of noise present; therefore it is excellent for images which are just slightly degraded by noise.

For multiplicative noise the model for e_{ij} given by (5.12) applies. To obtain the parameter k for (5.13), consider that from (2.2) and (2.15):

$$\sigma_e^2 = x_o^2 \sigma_v^2 \quad (5.17)$$

Thus:

$$k = \frac{S_{ij}}{S_{ij} + x_o^2 \sigma_v^2} \quad (5.18)$$

Note that x_o in (5.17) is unknown. A first approximation of x_o is the mean gray level in the neighborhood \bar{z} . Also note that \hat{z}_o can be feedback to (5.17) to yield a more accurate estimate of x_o thus yielding an iterative technique.

As with the case of additive noise the parameter $k \in [0, 1]$: In low contrast areas $S_{ij} \approx x_o^2 \sigma_v^2$ and $\hat{z}_o \approx \bar{z}_{ij}$; in high contrast regions, S_{ij} is large compared with $x_o^2 \sigma_v^2$ and $\hat{z}_o \approx z_o$.

Note that the output of applying J. S. Lee's Local Statistics Filter shows "artifacts" around image edges similarly to the effect upon images degraded by additive noise.

The local statistics filter can also be tuned to deal with a combination of multiplicative and additive noise: In this case the parameter k can be shown easily to be approximated by:

$$k = \frac{S_{ij}}{S_{ij} + x_o^2 \sigma_v^2 + \sigma_e^2} \quad (5.19)$$

5.3 An Original Localized Variance Smoother

As described at the beginning of this chapter, the sample variance, similarly to the absolute gradient used in Chapter 4, is a measure of the modelling error of (2.4). The sample variance is preferable over the absolute gradient because it is less sensitive to



(a)



(b)



(c)



(d)



(e)

Figure 5.6 (a) - undegraded "Face" image. (b) - "Face" degraded by MWGN with $\sigma_v = 0.2$; (c) - noise removed using J. S. Lee's local statistics filter with window size 7×7 ; (d) - noise removed using J. S. Lee's sigma filter with window size 7×7 ; (e) - noise removed using the localized variance smoother of Section 5.3 with window with window size 5×5 for computing the local variance map and window size 7×7 for smoothing.

noise. However, the sample variance cannot localize modelling error as precisely as the absolute gradient. An example of the difficulties encountered when using the variance-based techniques is illustrated by J. S. Lee's local statistics smoother of the previous section. J. S. Lee's method cannot localize edges and this causes a grainy texture to appear near image edges. To solve this problem J. S. Lee has been forced to use gradient masks [8]. Here an original method using sample variance is presented which has excellent edge localization and does not require the use of gradient masks.

The method is based on a new measure of modelling error based on sample variance which localizes edges. Such a measure is introduced here. Consider that a fixed window size is used for computing the sample variance—e. g. 3x3; consider, that a fixed window size used for smoothing—e. g. 5x5. Given a pixel z_o center of a smoothing window and another pixel z_{ij} inside the smoothing window; a new measure of the modelling error at pixel z_{ij} is:

$$M_{ij} = (\min\{S_{mn}\})^{\frac{1}{2}} \quad (5.20)$$

where the set $\{S_{mn}\}$ contains the sample variances of all neighborhoods containing both z_{ij} and z_o of the predetermined size (usually either 3x3 or 5x5). See Figure 5.7.

Note that if the sample variance is restricted to only two pixels then:

$$M_{ij} = \frac{|z_{ij} - z_o|}{\sqrt{2}} \quad (5.21)$$

which is just another formulation of the absolute gradient.

As a motivation for the measure of mismodelling of (5.20), consider the example of Figure 5.8 where, again, a 3x3 window is used for computing the sample variance. Of all the possible windows chosen for computing M_{ij} , the only one that shows the modelling error at z_{ij} is also the one with the least sample variance (in this particular case the variance is zero); this shows that (5.20) does give the correct result for pixel z_{ij} . However (5.20) will not give the correct result for pixel z_{mn} .

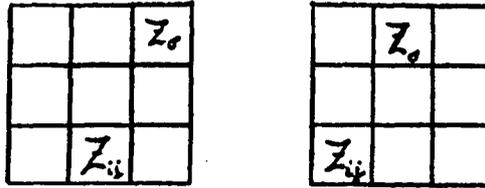


Figure 5.7 - Example showing the two 3x3 neighborhood sample variances which must be computed to determine M_{ij} at pixel z_{ij} .

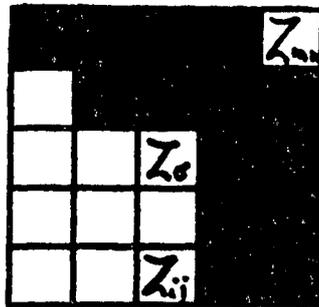


Figure 5.8 - edge localization by the new mismodelling measure: M_{ij} .

As shown in the previous paragraph M_{ij} , is not as localized as the absolute gradient. However, as a sample variance technique M_{ij} is less sensitive to noise than the absolute gradient. In other words, the new measure of mismodelling M_{ij} offers a new tradeoff between low noise sensitivity and edge localization.

This new measure of mismodelling M_{ij} can be used for designing a filter in a similar way that the absolute gradient was used in Chapter 4; namely this new filter is a data dependent weighted sum such that:

$$w_{ij} = \begin{cases} \frac{C}{\text{threshold}^n} & \text{for } M_{ij} \leq \text{threshold} \\ \frac{C}{M_{ij}^n} & \text{for } M_{ij} > \text{threshold} \end{cases} \quad (5.22)$$

with:

$$1 \leq n < \infty \quad \text{typically } n = 4$$

$$C \text{ is such that: } \sum_{ij} w_{ij} = 1$$

Typically $threshold = \sigma_{ij}$, where $\sigma_{ij} = \sigma_e + \bar{z}_{ij}\sigma_v$. σ_{ij} is the standard deviation of any possible combination of additive and multiplicative noise sequences.

Equation 5.22 is valid for all pixels affected by modelling error, as noted in Chapter 2, this excludes the center pixel z_o . This exception is easily handled by imposing that $M_o = 0$, which simply says that there is no modelling error at z_o , prior to applying (5.22).

I. e. imposing that $w_o = \frac{C}{\sigma^n}$.

As described, the algorithm requires the computation of the local sample variance for for several neighborhoods every time a single pixel is estimated. This would clearly require a prohibitive computational time. For example, if a 3x3 window is used for computing the sample variance, a total number of sample variances which must be computed for the whole image is 9N. (Where N is the total number of pixels in the image.) One simple scheme can reduce this computational time by 9 at the expense of extra memory usage. This scheme consists on precomputing a variance map of the image for the desired window size and storing the results in a temporary buffer. To find the sample variance of any neighborhood of the image it suffices to simply read the appropriate value from the above buffer.

However, in spite of the reduction of computational complexity afforded by the above scheme, it has been found that the logic required to determine which sample variances belong to the set $\{S_{mn}\}$ of (5.20) is very costly in computational time. So, in practice only a simplified formulation of the filter is used. This simplified formulation uses only a one-dimensional version of the filter (for which the number of elements in the set $\{S_{mn}\}$ is much reduced.) The filter in a one-dimensional formulation is applied to every column (row) of the image; to the result the same is done to every row (column) of the image.

The algorithm produces excellent edge preservation and can smooth near image edges. As with J. S. Lee's local statistics smoother and the sigma filter this algorithm is designed to deal with non-impulsive noise. If the use of this algorithm is desired when some impulse noise is present it is necessary to prefilter the signal as it was done in Section 3.5. This filter is shown, in Chapter 6, to give less mean square error than any of the other filters discussed for non-impulsive additive or multiplicative noise filtering. See Figures 5.6 and 5.9.

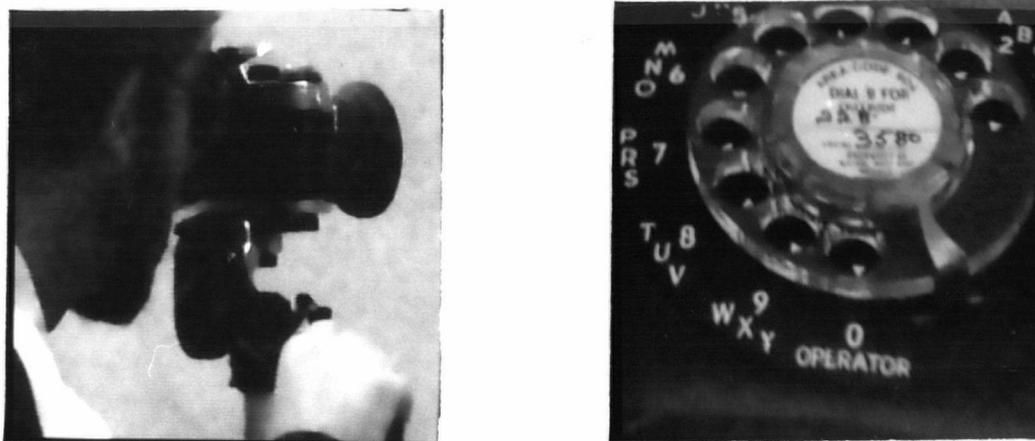


Figure 5.9 - noise removed of degraded images of Figure 4.4a and 3.4a smoothed using localized variance smoother with a window size 5x5 for computing the local variance map and window size 7x7 for smoothing.

Chapter 6: A Comparative Study of Local Noise Filtering Techniques

This chapter studies the performance of the most relevant filters described in Chapters 3, 4 and 5. Performance is studied subjectively through visual examination of filter outputs and objectively by using mean square error analysis. The filters whose performance is studied in this chapter are:

- * running mean filter.
- * median filter
- * sigma filter
- * double window modified trimmed mean filter (DWMTMF)
- * generalized gradient filter
- * J. S. Lee's local statistics filter
- * localized variance filter

Three images are used for analyzing the performance of the filters discussed; they are: the "camera", the "face" and the "telephone." All three images (as well as any others here) are represented by 256x256 arrays and have 256 (from 0 to 255) gray levels.

For testing the filters, the three images have been degraded by additive or multiplicative noise sequences. To the degraded images, all the above filters are applied. Section 6.1 displays tables of the root mean square error resulting in applying the filters. Section 6.2 displays sample images and describes some of the visual effects and artifacts introduced by the filters.

The different settings used in the above filters are:

- ▷ *Running mean filter* - window size is 3x3
- ▷ *Median filter* - window size is 3x3.

- ▷ *Sigma filter* - window size 7 for additive noise and window size 5 for multiplicative noise; a threshold of $2.5\sigma_e$ is used for additive noise and a threshold of $2.5\bar{z}_i\sigma_v$ (signal dependent) is used for multiplicative noise.
- ▷ *Double window modified trimmed mean filter (DWMTMF)* - the settings for this filter are identical to the ones for the sigma filter except that a 3x3 window is used for obtaining the median.
- ▷ *Generalized gradient filter* - the method suggested in section 3.4 is used, a 3x3 window and a count of 2 are used for Step 1; Step 2 has identical settings to the sigma filter.
- ▷ *J. S. Lee's local statistics filter* - window size used is 7x7 for additive noise degradation and 5x5 for multiplicative noise degradation.
- ▷ *Localized variance filter* - for additive noise window size 7x7 is used and a window size 5x5 is used for obtaining the variance map; for multiplicative noise window sizes 5x5 and 3x3 are used for smoothing and for obtaining the variance map respectively. In both cases the power used is four ($n = 4$ in (4.22)).

6.1 A Root Mean Square Analysis

Consider that the "telephone" image is used first. As a shorthand this image will be named "T." Four Gaussian noise sequences are added to T using a pseudo-random number generator the standard deviation of this sequences are 10.0, 15.0, 20.0 and 25.0. Each one of the above degraded images is called TAG10, TAG15, TAG20 and TAG25, respectively. Also, multiplicative noise is used with a standard deviation of 0.2—the multiplicative noise sequence is also a Gaussian distributed pseudo-random sequence. The image T degraded by the above noise sequence is called TMG0.2.

	TAG10	TAG15	TAG20	TAG25	TMG0.2
median	7.767	9.223	10.853	12.551	10.354
running mean	7.895	8.831	9.880	11.074	9.421
sigma	5.886	7.946	9.726	11.401	11.442
local statistics	6.135	8.172	9.834	11.189	10.093
generalized gradient	5.970	7.948	9.709	11.377	11.016
DWMTM	7.832	8.916	10.086	10.843	10.900
localized variance	5.499	7.412	9.141	10.643	9.213

Table 6.1 - root mean square (RMS) performance of the filters for the image T degraded by varying amounts of additive noise and multiplicative noise.

The images F and C have been degraded by the same noise sequences that degrade the image T above. The notation used for the different degraded images obtained from F and C is analogous to the notation used for the image T; e. g. FMG0.2 represents the F image degraded by multiplicative noise with $\sigma_v = 0.2$.

	FAG10	FAG15	FAG20	FAG25	FMG0.2
median	7.804	9.287	10.928	12.647	11.128
running mean	8.317	9.118	10.122	11.275	10.229
sigma	6.711	8.691	10.338	11.833	11.557
local statistics	6.944	8.989	10.652	12.028	10.062
generalized gradient	6.780	8.684	10.316	11.566	10.915
DWMTM	7.968	9.054	10.144	11.122	10.038
localized variance	6.469	8.211	9.678	10.952	8.677

Table 6.2 - root mean square (RMS) performance of the filters for the image F degraded by varying amounts of additive noise and multiplicative noise.

Tables 6.1, 6.2 and 6.3 show clearly that the local variance filter performs better, in a mean square sense, than any of the other filters examined. Some notes on tables 6.1, 6.2 and 6.3:

- (1) - As stated in section 3.1, the sigma filter is excellent for images with low noise degradation. This may be verified for the RMS results after filtering CAG10, FAG10 and TAG10 with the sigma filter.

	CAG10	CAG15	CAG20	CAG25	CMG0.2
median	5.377	7.019	8.723	10.444	7.049
running mean	5.206	6.459	7.924	9.520	6.243
sigma	4.571	6.087	7.487	8.998	7.189
local statistics	4.694	6.109	7.428	8.728	6.134
generalized gradient	4.631	6.120	7.494	8.971	6.706
DWMTM	4.695	5.615	6.679	7.879	5.637
localized variance	4.160	5.302	6.516	7.845	4.947

Table 6.3 - root mean square (RMS) performance of the filters for the image C degraded by varying amounts of additive noise and multiplicative noise.

- (2) - As with the previous note, J. S. Lee's local statistics smoother is predicted to have excellent performance for low noise degradation. This is also verified.
- (3) - The DWMTM filter of Y. H. Lee and S. A. Kassam is very robust and its relative performance improves with increasing noise standard deviation to the point of approximating the performance of the localized variance filter.
- (4) - The generalized gradient filter does not seem to perform better (in a mean square sense) than the sigma filter for additive noise. However, the generalized gradient filter does perform better in the case of multiplicative noise degradation.

	no iterations	1 iteration	2 iterations
sigma	7.189	6.253	6.187
generalized gradient	6.706	6.191	6.263
DWMTM	5.637	5.859	6.046

Table 6.4 - root mean square (RMS) performance of the sigma, DWMTM and generalized gradient filters in removing the noise of the image CMG0.2. Up to two iterations used.

The iterative scheme introduced in Chapters 3 and 4 does not produce an improvement of the RMS performance of the sigma filter or the DWMTMF when the noise is additive.

However, for multiplicative noise, iterating can produce a radical RMS improvement. See Table 6.4.

6.2 A Visual Analysis

This section is organized in point form with each point describing the major advantages and disadvantages of each filter.

▷ *Local statistics filter*

It has excellent edge preservation; but it cannot localize edges so it leaves an undesirable noise strip surrounding all image edges highlights or lowlights. See Figure 6.1b.

▷ *Sigma filter*

It is superb at localizing and preserving edges; however, it does not remove noise outliers. See Figure 6.1c.

▷ *Double window modified trimmed mean filter (DWMTMF)*

It is also excellent at localizing and preserving edges, and it does remove noise outliers—greater robustness. However, it also removes thin lines, lowlights and highlights. See the lack of the reflection on the left eye of the filtered “Face” of Figure 6.1d.

▷ *Generalized gradient filter*

As the DWMTMF, it does remove outliers while preserving edges. It is better than the DWMTMF, however, in that it does not remove thin image lines. (Illustration not shown.)

▷ *Localized variance smoother*

It trades some of the edge preservation quality of the sigma filter for lower noise sensitivity. See Figure 6.1e. The highlight in the left eye

is preserved, as well as the highlights in the hat feathers. Note that, except for the above detail, the output of localized variance filter is very similar to the output of the DWMTMF.



(a)



(b)



(c)



(d)



(e)

Figure 6.1 (a) - "Face" degraded by AWGN with $\sigma_\epsilon = 15.0$: (b) - noise removed using J. S. Lee's local statistics filter with window size 7×7 (c) - noise removed using J. S. Lee's sigma filter with window size 7×7 (d) - noise removed using the DWMTMF with median window size 3×3 and smoothing window size 7×7 (e) - noise removed using the localized variance smoother of Section 5.3 with window with window size 5×5 for computing the local variance map and window size 7×7 for smoothing.

Conclusion

Chapters 3, 4 and 5 have shown the widespread use of the absolute gradient and the sample variance in present day noise filters in digital image processing. Both the absolute gradient and the sample variance have been shown to have much in common. In fact, they both measure modelling error of the local zeroth order model represented by (2.4). Therefore it has been shown that the local zeroth order model of (2.4) is the common denominator behind the majority of current non-linear local noise filtering algorithms which, up to now, have appeared to be unrelated and ad hoc. Furthermore, it has been shown that all present day non-linear noise filters can be viewed as data dependent weighting schemes whose weights are related to some measure of confidence in the model of (2.4). This has led to a general formulation applicable to a majority of the above mentioned filters. This general formulation views the non-linear filters as being composed of two steps:

STEP 1: obtain a robust estimate of a pixel z_o .

STEP 2: obtain a better estimate of z_o by performing a weighted sum within a neighborhood of z_o .

STEP 2 can be iterated if desired.

The local sample variance is computed for more than the two pixels required to obtain the absolute gradient, and therefore, the local sample variance is less sensitive to noise. Therefore, the local sample variance is more desirable as a measure of the modelling error of (2.4). Unfortunately, its use has been, until now, hampered by its lack of edge localization and it seems to have been put aside, for that reason, by current researchers. However, Chapter 5 has introduced a new measure of modelling error, based on the sample variance, which achieves far superior edge localization but maintains the low sensitivity to noise. As expected, a filter based on this new measure of mismodelling confidence has been

shown to perform better, in a least squares sense, than all other current local non-linear non-impulsive noise filtering algorithms. Also, the suggested implementation of this filter makes it very efficient and, as such, the filter is a good practical choice.

Appendix B describes another original algorithm which will be subjected to future research by the author. A preliminary implementation of this algorithm has shown it to be excellent in edge preservation and noise removal.

Appendix A

As described in (1.23), the mean square error of any of the non-linear filters discussed in Chapters 2, 3 and 4 can be separated into three terms; namely: $(1/N) \sum (\tilde{x}_{ij} - x_{ij})^2$, $(1/N) \sum \tilde{\epsilon}_{ij}^2$ and $(2/N) \sum ((\tilde{x}_{ij} - x_{ij})\tilde{\epsilon}_{ij})$.

This appendix displays a table of the three above components of the mean square error for several non-linear filters applied to the image TAG10. The settings used in the filters are the same as those specified in Section 5.1.

	$(1/N) \sum (\tilde{x}_{ij} - x_{ij})^2$	$(1/N) \sum \tilde{\epsilon}_{ij}^2$	$(2/N) \sum ((\tilde{x}_{ij} - x_{ij})\tilde{\epsilon}_{ij})$	<i>MS</i>
median	59.15	40.20	-38.02	60.33
running mean	52.07	11.26	-0.50	63.76
sigma	31.26	8.52	-5.14	34.64
local statistics	32.19	5.25	0.17	37.61
DWMTM	63.71	3.79	-6.16	61.34
localized variance	16.43	13.49	0.29	30.21

Table A.1 - the three components of the mean square error for several filters applied to the image TAG10.

Table A.1 shows that the localized variance filter has a relatively lower modelling error than the other filters in the table. On the other hand, the localized variance filter has a relatively higher residual noise.

Note that the cross term $(2/N) \sum ((\tilde{x}_{ij} - x_{ij})\tilde{\epsilon}_{ij})$ cannot be neglected. In fact, this term is often large: see the case of the median filter in Table A.1.

Appendix B For Further Research

This appendix describes another original filter for smoothing images which is suggested as a future research topic.

A motivation for this new filter is the excellent performance of the original filter described in Chapter 5. From the results presented in Chapter 6, it is clear that methods based on the sample variance perform better than those based on the absolute gradient if localization of edges is achieved. As the new filter uses sample variance and achieves better localization of modelling error than its counterpart in Chapter 5, it is expected that it will perform better, in a least squares sense, than all other the filters discussed.

Like all other non-linear filters described in this thesis, the new filter can be described as a data dependent local weighting scheme. In particular, the scheme consists of averaging only those pixels which are “likely” to follow the model of (2.4). In the sigma filter, the “likelihood” of following the model of (2.4) is described by the inequality: $|z_{ij} - z_o| < threshold$. However, the use of the absolute gradient $|z_{ij} - z_o|$ is undesirable because of its sensitivity to noise. A more noise-insensitive definition of which pixels are “likely” to follow the model of (2.4) is described below.

Consider a small window from which the estimate of the pixel z_o is to be obtained. Order all these pixels according to their absolute difference with z_o (absolute gradient), starting with z_o . Name this sequence (y_i) . Clearly, y_2 is “very likely” to follow the model of (2.4), y_3 is “somewhat less likely,” y_4 is even “less likely” and so on. At this point, the sigma filter classifies all the pixels y_1, \dots, y_k with $|y_k - z_o| < threshold$, as “likely” to follow the model (of (2.4)). However, this definition is, as stated before, very sensitive to noise. A new measure of “likelihood” introduced here is:

$$L_k = \text{Variance of } \{y_1, \dots, y_k\} \quad (B.1)$$

The new filter uses this measure. That is, the points y_1, \dots, y_k are said to be “likely” to follow the model of (2.4) iff:

$$L_k < \textit{threshold} \quad \text{where this } \textit{threshold} \text{ is to determined} \quad (\textit{B.2})$$

Preliminary studies of this filter show that threshold should be dependent upon the variable k . A suggested formula for this relationship is:

$$\textit{threshold} = \sigma_\epsilon \left(\frac{k+1}{k-1} \right) \quad (\textit{B.3})$$

where σ_ϵ represents the standard deviation of the noise.

(B.3) reflects the decrease of noise sensitivity for increasing k .

A preliminary implementation shows that this new filter enhances image edges and is excellent at removing noise. A drawback of the filter, however, is its computational complexity. Further research is required to reduce this complexity and to test the mean square error of this filter against that of its counterpart in Chapter 5.

References

1. Y. H. Lee and S. A. Kassam, "Generalized median filtering and related nonlinear filtering techniques," *IEEE Trans. on ASSP*, vol. ASSP-33, No.3, pp. 672-683, June 1985.
2. A. C. Bovik, T. S. Huang and D. C. Munson Jr., "A generalization of median filtering using linear combinations of order statistics," *IEEE Trans. on ASSP*, vol. ASSP-31, pp. 1342-1350, Dec. 1983.
3. J. S. Lee, "Digital image smoothing and the sigma filter," *Computer Vision and Image Processing* 24, pp. 255-269 (1983).
4. C. Rey and R. K. Ward, "An adaptive minimum-variance predictor-smoother algorithm for image processing," *Proc. 22nd Annual Allerton Conference on Comm. Control and Computing*, University of Illinois, Urbana-Champlain, pp. 122-131, Oct. 1984.
5. D. Wang, A. Vagnucci and C. Li, "Image enhancement by gradient inverse smoothing scheme," *Computer Graphics and Image Processing* 15, pp. 167-181, 1981.
6. M. Nagao and T. Matsuyama, "Edge preserving smoothing," *Computer Graphics and Image Processing* 9, pp. 394-407, 1979.
7. J. S. Lee, "Digital image enhancement and noise filtering by use of local statistics," *IEEE Trans. on PAMI*, vol. PAMI-2, pp. 165-168, March 1980.
8. J. S. Lee, "Refined filtering of image noise using local statistics," *Computer Graphics and Image Processing* 15, pp. 380-389, 1981.
9. J. S. Lee, "Speckle analysis and smoothing of synthetic aperture radar image" *Computer Graphics and Image Processing* 17, pp. 24-32, 1981.
10. W. K. Pratt, *Digital Picture Processing*, New York, John Wiley and Sons, 1978.
11. A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, Vol. 1, 2nd Ed., Academic Press, 1982.