To the Graduate Council:

I am submitting herewith a thesis written by Hamed Sari–Sarraf entitled "Iterative Noise Filtering." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

Dragana Brzakovic, Major Professor

We have read this thesis
and recommend its acceptance:

Accepted for the Council:

Vice Provost
and Dean of the Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at the University of Tennessee, Knoxville, I agree that the library shall make it available to borrowers under rules of the library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in her absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature Hamed Sari-Sarraf

Date Aug 31, 86

ITERATIVE NOISE FILTERING

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Hamed Sari-Sarraf

December 1986

ACKNOWLEDGMENTS

I would like to thank Dr. D. Brzakovic for patiently guiding me throughout my career as a research assistant, for being a source of support and advice and for giving me the opportunity to work with her.

I extend many thanks to Drs. W. L. Green and R. E. Bodenheimer for being on my thesis committee.

Thanks are due to Danny Newport for tremendous amount of help with the computer and Jenny Craven for final formatting of this document. Many thanks to Hal Beck for proof reading and final preparation of this thesis.

ABSTRACT

Two methods for additive noise filtering are proposed in this thesis. The first method, Automated Noise Filtering (ANF), is a modified and automated version of an already existing noise removal method called Noise Filtering by use of Local Statistics (NFLS). ANF is capable of processing images with different degradation levels. Furthermore, it is completely automated, and thus, requires no a priori knowledge of signal or noise parameters. ANF is able to preserve vital signal information, i.e., edges and details. It is computationally efficient and well suited for parallel processing. The second method, Iterative Noise Filtering (INF), is an iterative version of ANF. INF possesses all the abilities of Automated Noise Filtering, and in addition, it is capable of removing the additive noise completely from the highly degraded images. This iterative process terminates automatically after the additive noise is entirely removed. Performance of the methods of Automated and Iterative Noise Filtering are evaluated based on their applications to various images. Each of the two methods are shown to compare favorably to an alternative technique of noise removal.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE PAGE

CHAPTER 1

INTRODUCTION

Noise reduction and removal are recognized to be crutial in many image processing tasks. This is particularly true when the images to be processed are degraded by very high levels of noise. Generally speaking, the task of noise removal may be approached in two different ways. If a great deal of a priori knowledge on the images is available it is possible to model the original image and fit that model to the data at hand. On the other hand, if there exists little or no knowledge about the images, one can attempt to model the source of degradation and subsequently remove its effects. In essence, the task of noise removal reduces to an estimation problem. Whatever the approach, noise filtering is not a trivial process and requires thorough investigation of image properties and noise statistics. If care is not taken, noise removal can result in signal removal or loss of vital signal information. Over the years, numerous techniques have been introduced which deal particularly with noise removal, e.g., [1-3,5-16,18-20]. While these techniques are effective, they are aimed towards specific applications, and therefore, lack generality. In addition, no general criteria for evaluating the performance of a method exist.

The study described in this thesis is aimed towards developing a noise filtering method which is:

1. Adaptive (because of conceptual simplicity and computational efficiency associated with adaptive filters [14]).

2. Suitable for parallel processing.

3. Able to remove noise from images that are degraded by different levels of noise.

4. Able to preserve vital signal information, i.e., edges and details.

5. Automated and general, i.e., requires no a priori knowledge of signal or noise parameters.

6. Able to remove noise completely from highly degraded images.

The main body of this thesis is organized into three logical units. The first of these, Chapter 2, focuses on theoretical development and implementation of a noise removal technique that meets the first two of the six requirements. This technique is called Noise Filtering by use of Local Statistics [10], and is chosen to be the basis for the methods developed in this work. Initially, considerations were given to other filtering methods which also deal with noise removal as an estimation problem. These methods are based on: (1) frequency domain techniques [6], or (2) application of Kalman filtering algorithm [8], or Bayesian estimation extended to two–dimesional arrays which lead to the concept of a recursive filtering algorithm [7,10,16]. Because of the inabilities of these methods to meet the first two requirements, they have not been pursued any further in this thesis.

In the second part, Chapter 3, the method of Automated Noise Filtering which is a modified and automated version of NFLS is proposed. Modifications introduced to the equation of NFLS provide for processing images that are degraded by different levels of noise (third requirement). Automation of NFLS on the other hand, enables Automated Noise Filtering to be applied to arbitrary images. The proposed procedure for automatic estimation of the noise variance allows ANF to preserve vital signal information. Other filtering techniques which utilize noise removal without loss of signal information were considered, e.g., [1,9,10,11,15,19]. However, the method proposed by Lee [11] is singled out for comparison with ANF since the essence of Lee's method is also Noise Filtering by use of Local Statistics. In general, the method of Automated Noise Filtering outperforms Lee's method. The method of Automated Noise Filtering meets all the requirements stated at the beginning of this Chapter except for the last one.

Finally the third unit, Chapter 4, focuses on the development of an iterative algorithm called the method of Iterative Noise Filtering. This method meets all the stated requirements. Other filtering techniques which also utilize iterative algorithms were considered, e.g., [5,13,14]. In this work, performance of the method developed by Lev et al. [13] is compared to that of Iterative Noise Filtering. The method developed by Lev et al. makes use of the iterative weighted averaging of the intensities of a pixel's immediate neighbors to estimate the true intensity of that pixel. Comparison shows that INF outperforms this method. This is specially evident when the input images contain edges with orientations other than $0°, 45°, 90°$, and $135°$.

In summary, the underlying theory of the methods of Automated and Iterative Noise Filtering are both based on Noise Filtering by use of Local Statistics. The modifications proposed provide for their applications to arbitrary images with different degradation levels. Implementation of each of the methods of Automated and Iterative Noise Filtering is straight forward, and is outlined by the computational procedures given in this thesis.

CHAPTER 2

NOISE FILTERING BY USE OF LOCAL STATISTICS

The noise removal techniques developed in this work have their foundations based on a filtering method called Noise Filtering by use of Local Statistics [10]. Thus, in this Chapter, attention is focused on the theoretical development of NFLS as well as its implementation. Consequently, strong points and shortcomings encountered in employing this method for the purpose of additive noise removal are discovered.

2.1 THEORETICAL DEVELOPMENT

This Section presents the theoretical foundation of Noise Filtering by use of Local Statistics. First, the underlying assumptions and notations are introduced, and then the filter equation is derived.

Consider a noisy image array, M x N, wherein a pixel value, $g(x, y)$, $x = 1, 2, ..., \mathrm{M}$ and $y = 1, 2, ..., \mathrm{N}$, is modeled as the sum of the true pixel value, $f(x, y)$, and the additive noise, $n(x, y)$, i.e.,

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

The objective of the method of NFLS is to estimate $f(x, y)$ based on the measured value, $g(x, y)$, the local mean

$$\overline{g}(x,y) = \frac{1}{(2n+1)\ \mathrm{x}\ (2m+1)} \sum_{k=x-n}^{x+n} \sum_{q=y-m}^{y+m} g(k,q)\ , \tag{2.2}$$

and variance

$$V(x,y) = \frac{1}{(2n+1)\ \mathrm{x}\ (2m+1)} \sum_{k=x-n}^{x+n} \sum_{q=y-m}^{y+m} \left[g(k,q) - \overline{g}(x,y)\right]^2 . \tag{2.3}$$

calculated over a $(2n+1)$ x $(2m+1)$, $m,n = 1,2,...,$ image subregion. [1]

The problem of estimating the true pixel value can be simplified by assuming that additive noise has zero mean, i.e.,

$$\mathrm{E}\left[n(x,y)\right] = \overline{n}(x,y) = 0\ , \tag{2.4}$$

and is spatially uncorrelated so that:

$$\mathrm{E}\left[n(x,y)n(K,L)\right] = \sigma^2\ \delta_{xK}\ \delta_{yL}\ . \tag{2.5}$$

The symbol δ in Equation 2.5 denotes the Kronecker delta function, E is the expectation operator, and σ^2 is the noise variance.

Under these assumptions, it follows from Equation 2.1 that:

$$\mathrm{E}\left[g(x,y)\right] = \overline{g}(x,y) = \mathrm{E}\left[f(x,y)\right] = \overline{f}(x,y)\ , \tag{2.6}$$

and

$$\begin{aligned} Q(x,y) &= \mathrm{E}\left\{\left[f(x,y) - \overline{f}(x,y)\right]^2\right\} \\ &= \mathrm{E}\left\{\left[g(x,y) - n(x,y) - \overline{f}(x,y)\right]^2\right\} \end{aligned}$$

[1] $(2n+1)$ x $(2m+1)$, $m,n = 1,2,...,$ image subregion is refered to as *window* in this work.

$$
\begin{aligned}
&= \mathrm{E}\left\{[g(x,y)-\overline{g}(x,y)-n(x,y)]^2\right\} \\
&= \mathrm{E}\left\{[g(x,y)-\overline{g}(x,y)]^2\right\}+\mathrm{E}\left\{[n(x,y)]^2\right\}- \\
&\quad 2\mathrm{E}\left\{[g(x,y)-\overline{g}(x,y)]\,n(x,y)\right\} \\
&= \mathrm{E}\left\{[g(x,y)-\overline{g}(x,y)]^2\right\}+\mathrm{E}\left\{[n(x,y)]^2\right\}- \\
&\quad 2\mathrm{E}\left\{[f(x,y)+n(x,y)-\overline{g}(x,y)]\,n(x,y)\right\} \\
&= \mathrm{E}\left\{[g(x,y)-\overline{g}(x,y)]^2\right\}-2\mathrm{E}\left\{f(x,y)n(x,y)\right\}- \\
&\quad 2\mathrm{E}\left\{[n(x,y)]^2\right\}+2\overline{g}(x,y)\,\mathrm{E}\left\{n(x,y)\right\}+ \\
&\quad \mathrm{E}\left\{[n(x,y)]^2\right\}.
\end{aligned}
\tag{2.7}
$$

Furthermore, if noise and signal are uncorrelated

$$
\mathrm{E}\left\{f(x,y)n(x,y)\right\}=0\ , \tag{2.8}
$$

and Equation 2.7 becomes

$$
\begin{aligned}
Q(x,y) &= \mathrm{E}\left\{[g(x,y)-\overline{g}(x,y)]^2\right\}-\mathrm{E}\left\{[n(x,y)]^2\right\} \\
&= \mathrm{E}\left\{[g(x,y)-\overline{g}(x,y)]^2\right\}-\sigma^2 \\
&= V(x,y)-\sigma^2.
\end{aligned}
\tag{2.9}
$$

Now that the basic assumptions and notations have been introduced, a weighted least-squares estimate of $f(x,y)$, $\hat{f}(x,y)$, can be obtained by considering the criterion function [4]:

$$
T=\frac{1}{2}\left\{\frac{\left[\hat{f}(x,y)-\overline{f}(x,y)\right]^2}{Q(x,y)}+\frac{\left[g(x,y)-\hat{f}(x,y)\right]^2}{\sigma^2}\right\}. \tag{2.10}
$$

The objective to obtain an estimate ($\hat{f}(x,y)$) that minimizes T is accomplished by considering

$$\frac{dT}{d\hat{f}} = \frac{\left[\hat{f}(x,y) - \overline{f}(x,y)\right]}{Q(x,y)} + \frac{\left[\overline{f}(x,y) - g(x,y)\right]}{\sigma^2} = 0 \qquad (2.11)$$

which yields

$$\hat{f}(x,y) = \overline{f}(x,y) + \frac{Q(x,y)}{Q(x,y)+\sigma^2}\left[g(x,y) - \overline{f}(x,y)\right]. \qquad (2.12)$$

Using Equations 2.6 and 2.9, Equation 2.12 becomes

$$\hat{f}(x,y) = \overline{g}(x,y) + \frac{V(x,y)-\sigma^2}{V(x,y)}\left[g(x,y) - \overline{g}(x,y)\right]. \qquad (2.13)$$

Computational procedure shown in Figure 2.1 illustrates the steps required in implementation of Noise Filtering by use of Local Statistics. As illustrated in this figure, implementation of NFLS involves determination of two parameters (shaded boxes): (1) selection of the size of the image subregion, $(2n+1)$ x $(2m+1)$, within which local statistics are computed and (2) determination of the noise variance. The choice of these two parameters and their effects on the obained results are discussed in the next Section.

2.2 IMPLEMENTATION AND RESULTS

The effects of window size within which local statistics are to be computed, and the value of the additive noise variance on the outcome of the filtering process were studied experimentally. The study was conducted by first experimenting with a set of artificially generated images containing regions of uniform intensity degraded by random noise of known characteristics. Then, experiments were performed on Forward Looking InFrared (FLIR) and TV images.

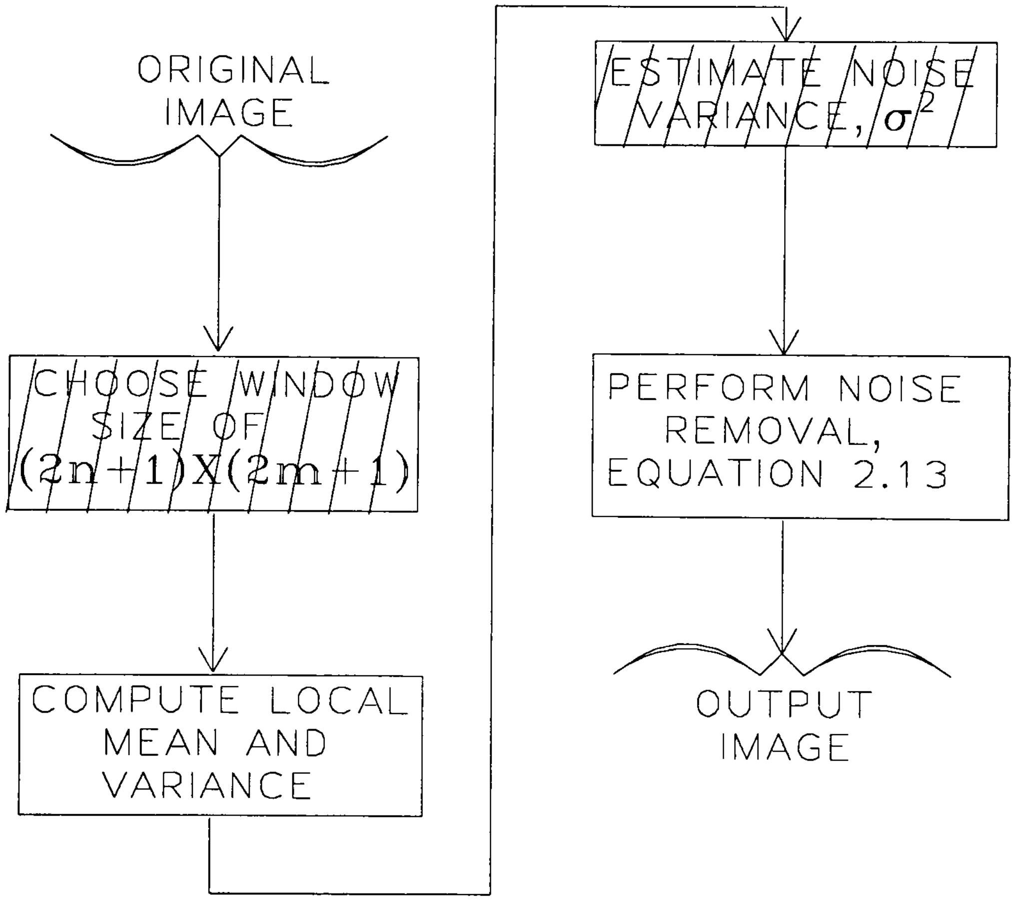

Figure 2.1: Computational Procedure for NFLS.

__Window Size.__The size of the window which characterizes the neighborhood of each degraded pixel, playes a crucial role in effective implementation of NFLS. The criteria for choosing the window size are the obtained results, theoretical considerations, and computational efficiency.

A typical example of the obtained results is the noisy image of Figure 2.2a that was processed by NFLS using window sizes which ranged from 3 x 3 to 11 x 11 (Figures 2.2b to 2.2f). It is evident by inspection of these results that selecting a small window size, i.e., 3 x 3 or 5 x 5, introduces no significant improvement over the noisy input image. This observation can be confirmed theoretically since the choice of a small window size causes $\overline{g}(x,y) \approx g(x,y)$ and $\hat{f}(x,y) \approx g(x,y)$, Equation 2.13.

On the other hand, by choosing a very large window size, pixels which belong to different regions of the input image will be assigned to the same neighborhood. Processing the regions within large windows therefore, smears the present edges and blurrs the output image. It was found based on experimental evidence that window sizes 7 x 7 and 9 x 9 are the most effective for the classes of images under investigation, Figures 2.2d and 2.2e. Since implementation with a 9 x 9 window is computationally less efficient, a 7 x 7 window is chosen as the best compromise, i.e., $m = n = 3$. These findings are in agreement with Lee [10] who reported best results with the same window size.

__Additive Noise Variance.__Determining the value of the additive noise variance, σ^2, is an important factor in effective implementation of NFLS. The criteria for determining a value for σ^2 is similar to that of choosing the window size, with the exception that neither a correct nor an incorrect estimate will have any effect on the computational efficiency of the method.

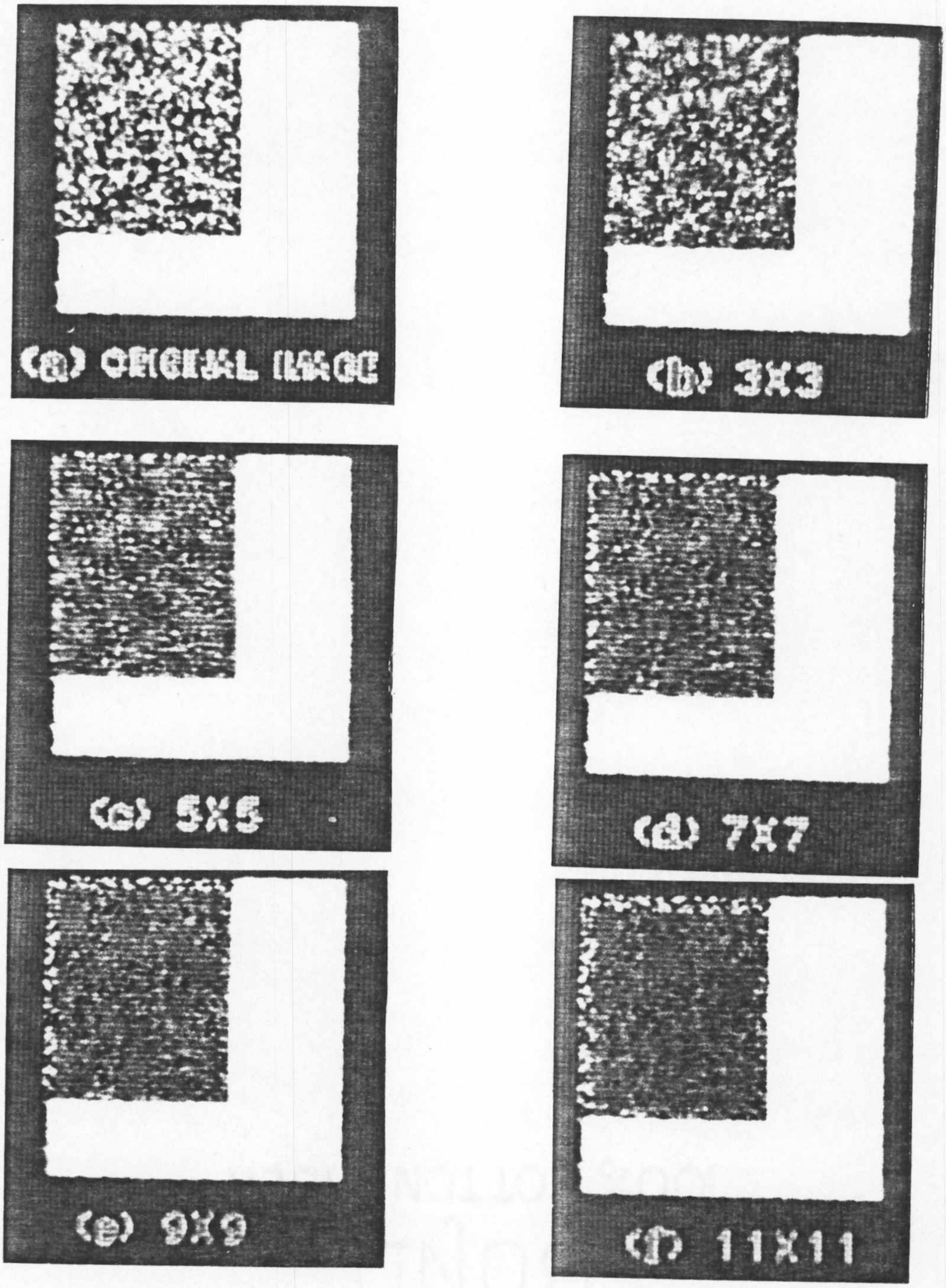

Figure 2.2: Effect of Window Size on the Outcome of NFLS (noise variance is known to be 2500).

The artificial image shown in Figure 2.3a was contaminated by random noise of known characteristics and then processed by NFLS using: (1) the actual value of σ^2, Figure 2.3b, (2) a value which was considerably less than the actual value of σ^2, Figure 2.3c and (3) a value which was considerably greater than the actual value of σ^2, Figure 2.3d. It is evident by inspection of these results that the choice of a noise variance significantly smaller than its real value results in ineffective filtering. This conclusion is confirmed theoretically by considering Equation 2.13, since as $\sigma^2 \longrightarrow 0 \Longrightarrow \hat{f}(x,y) \longrightarrow g(x,y)$. On the other hand, very large values of σ^2 will cause image blurring. Theory also shows that as $\sigma^2 \longrightarrow V(x,y) \Longrightarrow \hat{f}(x,y) \longrightarrow \overline{g}(x,y)$, Equation 2.13.

For the real images [2] under consideration, characteristics of the imaging device were not available, and therefore, values of the additive noise variance had to be determined experimentally. Initially, extreme values for σ^2 were considered. Typically, $\sigma^2_{max} = 500.0$ was chosen as the upper bound of the range of values for σ^2, and $\sigma^2_{min} = 25.0$ as the lower bound. Then, values within the range $[\sigma^2_{min}, \sigma^2_{max}]$ were tested by incrementing/decrementing the extreme values. It was found that values of $\sigma^2 = 100.0$ and $\sigma^2 = 150.0$ were the best estimates of the additive noise variance for the FLIR images under consideration. The effect of noise variance on the obtained results are shown in Figure 2.4. It is seen that underestimates of σ^2 resulted in very little change, while overestimates introduced undesirable changes in the image. The best result is obtained for $\sigma^2 = 125.0$, Figure 2.4d.

While results obtained by using Equation 2.13 are generally satisfactory, there are definite shortcomings in applying NFLS to the classes of images under investigation. These deficiencies limit the ability of NFLS to process

[2]Term real images refers to FLIR and TV images throughout this document.

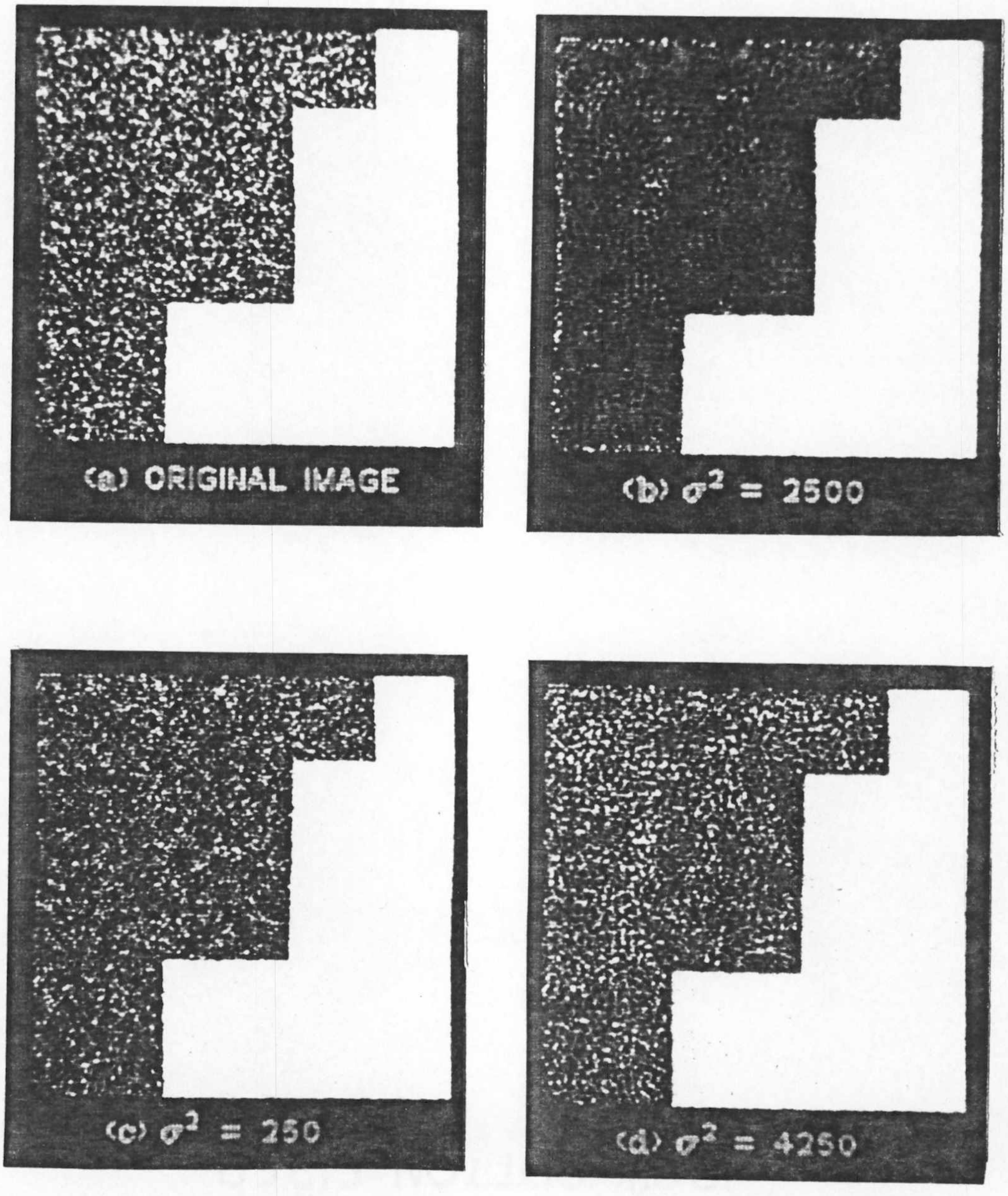

Figure 2.3: Effect of Noise Variance on the Outcome of NFLS (window size of 7 x 7 is used).

Figure 2.4: Effect of Noise Variance on the Outcome of NFLS (window size of 7 x 7 is used).

images with regions of semi-uniform [3] or uniform intensity. Furthermore, NFLS can not be applied to arbitrary images due to the fact that a priori knowledge of σ^2 is required. Finally, since estimating the true pixel value of a degraded pixel is based on its neighbors (within a 7 x 7 window), NFLS is not capable of processing the boundary pixels [4]. Appropriate modifications of Equation 2.13 as well as its automation, which will overcome these drawbacks, are proposed in Chapter 3.

[3]Regions where intensities of the pixels have a maximum variation of 2.

[4]Boundary pixels are those that lie at image boundaries, and thus, do not have adequate number of neighbors within a given window.

CHAPTER 3

AUTOMATED NOISE FILTERING

The method of Noise Filtering by use of Local Statistics, described in Chapter 2, is not capable of processing:

1. Images that contain regions of semi-uniform or uniform intensity.
2. Arbitrary images – since a priori knowledge of noise variance, σ^2, is required.
3. Pixels lying along the boundaries of the degraded image.

The method of Automated Noise Filtering [18] is developed to overcome these shortcomings, and is discussed in detail in this Chapter.

3.1 MODIFICATION OF NFLS

Application of Noise Filtering by use of Local Statistics to images which contain regions of uniform or semi-uniform intensity result in two types of problems.

Problem 1: From Equation 2.13 it follows that $V(x, y) = 0$ causes division by zero and breaks the computational procedure. This particular problem is encountered when processing subregions of uniform intensity which are noise free, such as, a 7 x 7 subregion of the encircled region in Figure 3.1. This region has the following intensities:

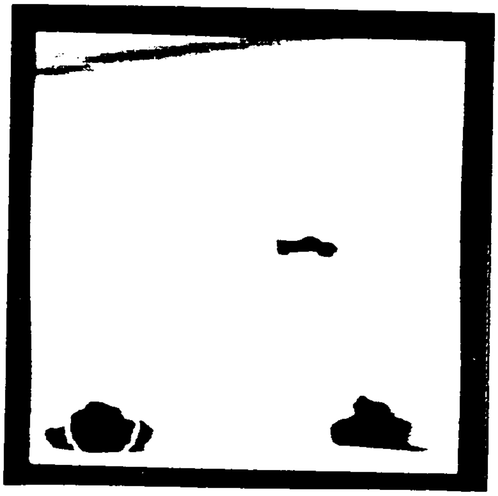

Figure 3.1: Image Containing Regions of Uniform Intensity.

1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1

The local mean in this case is $\overline{g}(x, y) = 1$ and the local variance is $V(x, y) = 0$.

Solution 1: The following modification:

$$\hat{f}(x,y) = \begin{cases} \overline{g}(x,y) + \dfrac{V(x,y) - \sigma^2}{V(x,y)} [g(x,y) - \overline{g}(x,y)] & \text{if } V(x,y) \neq 0 \\ g(x,y) & \text{if } V(x,y) = 0 \end{cases} \tag{3.1}$$

overcomes the described problem.

Problem 2: If the local variance computed from the original image is $V(x, y) \ll \sigma^2$, $Q(x, y)$ in Equation 2.9, which represents the variance of $f(x, y)$, becomes negative. This problem is encountered when processing subregions of semi-uniform intensity embedded within a noisy image. An example of a noisy image which contains regions of semi-uniform intensity is shown in Figure 3.2a. A 7 x 7 subregion of the encircled region in this figure has the following intensities:

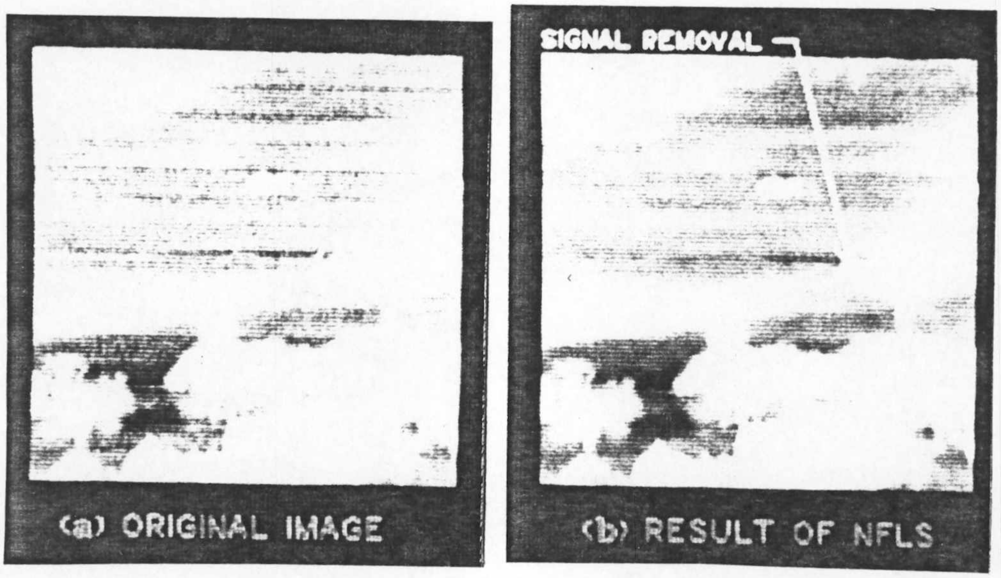

Figure 3.2: Noisy Image Containing Regions of Semi-Uniform Intensity.

254	255	254	255	254	255	254
254	255	254	255	254	255	254
254	255	254	255	254	255	254
254	255	254	255	254	255	254
254	255	254	255	254	255	254
254	255	254	255	254	255	254
254	255	254	255	254	255	254

The local mean is $\overline{g}(x,y) = 254.43$ and the local variance is $V(x,y) = 0.245$. Substitution of $\sigma^2 = 105$ (which has been found to be a suitable value for noise variance in this case) in Equation 2.13 gives an estimate of $\hat{f}(x,y) = 11$. Clearly, this is not the desired intensity and application of Equation 2.13 has resulted in signal removal instead of noise removal. The effects of these undesired estimates are observed in Figure 3.2b.

<u>**Solution 2:**</u>Modification

$$\hat{f}(x,y) = \begin{cases} \overline{g}(x,y) + \dfrac{V(x,y)-\sigma^2}{V(x,y)}\left[g(x,y)-\overline{g}(x,y)\right] & \text{if } V(x,y) \geq \sigma^2 \\ g(x,y) & \text{if } V(x,y) < \sigma^2 \end{cases} \tag{3.2}$$

guardes against this problem.

<u>**Final Solution:**</u>Combination of Equations 3.1 and 3.2 results in

$$\hat{f}(x,y) = \begin{cases} \overline{g}(x,y) + \dfrac{V(x,y)-\sigma^2}{V(x,y)}\left[g(x,y)-\overline{g}(x,y)\right] & \text{if } V(x,y) \geq \sigma^2 \\ g(x,y) & \text{if } V(x,y) = 0 \\ g(x,y) & \text{if } V(x,y) < \sigma^2. \end{cases} \quad (3.3)$$

The rationale behind these modifications is that a pixel in a region of uniform or semi-uniform intensity should retain its original value while the new estimate is appropriate in cases when an image subregion is degraded by noise. Equation 3.3 provides for successful processing of both FLIR and TV images, i.e., images which are degraded by high and low levels of noise.

3.2 AUTOMATION OF NFLS

As described in Section 2.2, implementation of NFLS requires a priori knowledge about σ^2. While the results obtained by using appropriate values of σ^2 are generally satisfactory, the applicability of the algorithm is limited to images with known properties.

A procedure for automatic determination of additive noise variance, which allows Automated Noise Filtering to be applied to arbitrary images is proposed. The proposed procedure is developed under the premise that at least K% of an image corresponds to regions of uniform intensity. Local variances obtained from regions of uniform intensity are the smallest variances, and are good estimates of the noise variance.

The proposed procedure consists of three steps as follows:

Step 1: Calculate local means and variances for each pixel $g(x,y)$ of the degraded image using Equations 2.2 and 2.3, respectively. As described in

Section 2.2, the size of the window, within which local statistics are computed, is chosen to be 7 x 7, i.e., $m = n = 3$.

Step 2: Find K% of the smallest local variances, $V(x,y)$. Generally, the choice of K is related to the choice of the window size. For the classes of images under consideration and a window size of 7 x 7, K is chosen to be 5.

Step 3: Choose the noise variance, σ^2, as follows:

$$\sigma^2 = \frac{1}{N \times M \times \frac{K}{100}} \sum_{i=1}^{N \times M \times \frac{K}{100}} Var(i) \tag{3.4}$$

where:

$Var(i)$: Local variances arranged in ascending order.

$M \times N$: Dimensions of the noisy image.

Experimental results have shown that a sample size of K=5% was a good compromise in the case where a 7 x 7 window size was chosen. The choice of a smaller sample size of the smallest local variances, e.g., 1%, results in underestimation of σ^2 and as a result no noise is removed. On the other hand, if the sample size is too large, e.g., 25%, the value of σ^2 is likely to be overestimated which results in: (1) image blurring – since $\sigma^2 \approx V(x,y)$ in regions of the image where there is an edge present, and therefore, $\hat{f}(x,y) \approx \overline{g}(x,y)$ (Equation 3.3), or (2) no noise removal – since $\hat{f}(x,y) \approx g(x,y)$ (Equation 3.3).

The computational procedure for this method of Automated Noise Filtering is shown in Figure 3.3. An evaluation of its performance is discussed in Section 3.4.

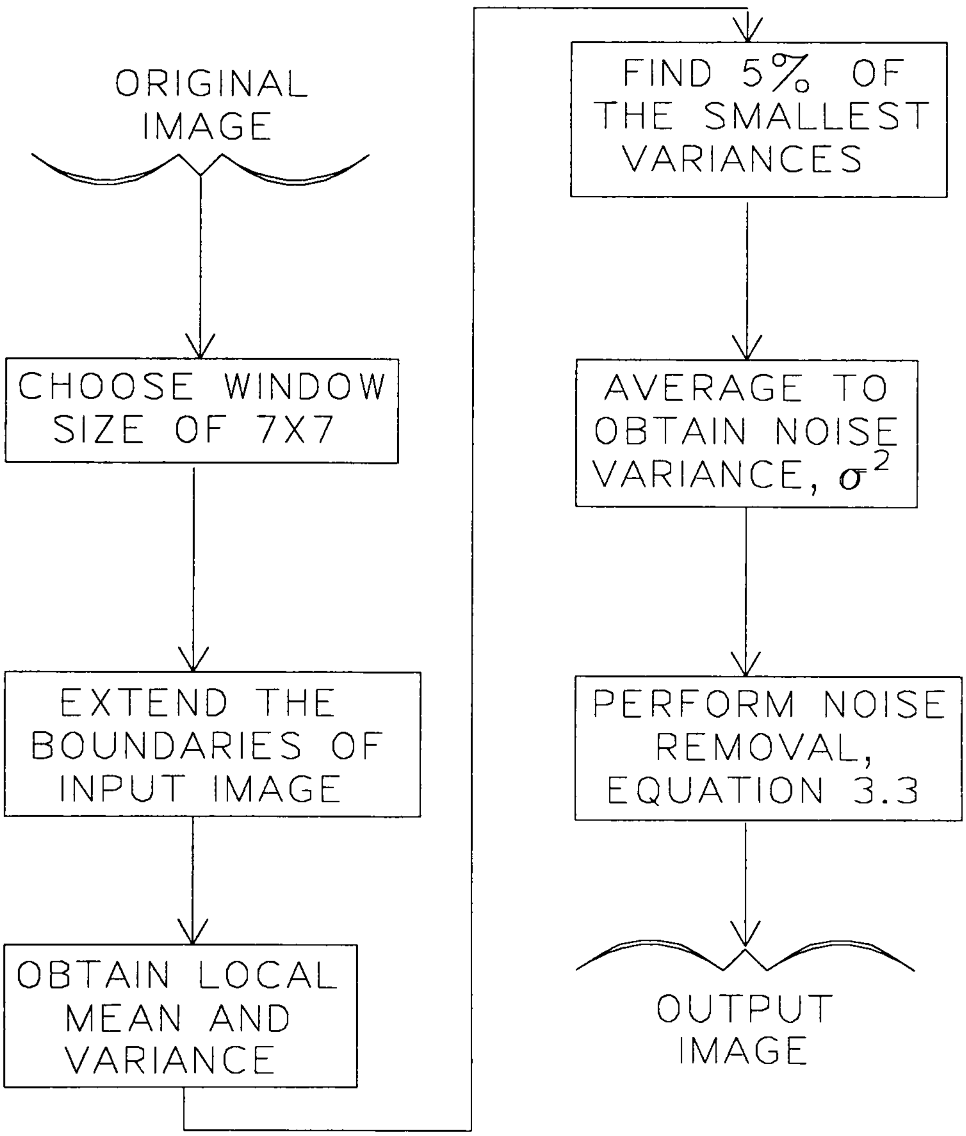

Figure 3.3: Computational Procedure for Automated Noise Filtering.

3.3 PROCESSING OF BOUNDARY PIXELS

Frequently in image processing, the objective is to generate an output image whose elements $f_O(i,j)$, $i,j = 1,2,...,\mathrm{N}$, are obtained by manipulating the pixels lying within a $(2n+1)$ x $(2m+1)$ neighborhood centered at pixel $f_I(i,j)$ of the input image array (Figure 3.4). Difficulties arise when the $(2n+1)$ x $(2m+1)$ neighborhood is centered at a pixel which lies in rows 1 thru n or $(\mathrm{N}-n+1)$ thru N or columns 1 thru m or $(\mathrm{N}-m+1)$ thru N. In these cases windows fall beyond the boundaries of f_I, and therefore, particular pixels do not have complete neighborhoods, Figure 3.5.

In this work, the method of pixel reflection is utilized to generate appropriate neighbors for pixels lying at image boundaries. Accordingly, $f_I(i,j-1)$, the neighbor of pixel $f_I(i,j)$ (Figure 3.6), is generated as $f_I(i,j-1) = f_I(i,j+1)$. Similarly, $f_I(i-1,j) = f_I(i+1,j)$, $f_I(i-1,j+1) = f_I(i+1,j+1)$. A neighborhood of a pixel generated in this way is a good estimate of an extension of regions having uniform intensities or textural properties which are likely to appear at image boundaries.

In general when using a $(2n+1)$ x $(2m+1)$ window, in order for all pixels of the original N x N image [1] to have an adequate number of neighbors, Figure 3.7, it is necessary to:

1. Reflect m rows around the first row, such that:

$$f_I(-i,j) = f_I(i+1,j) \tag{3.5}$$

[1]Dimensions of N x N are chosen for ease of derivation. Original image can just as well be M x N, where M $\neq$ N.

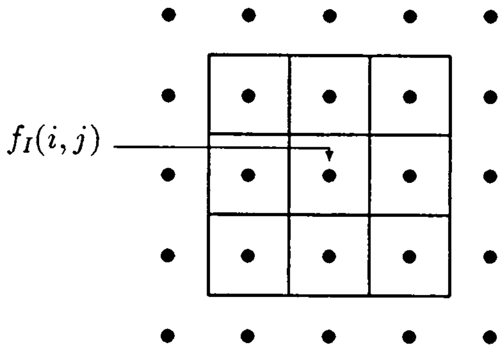

Figure 3.4: $f_I(i,j)$ and its Neighborhood for $m = n = 1$.

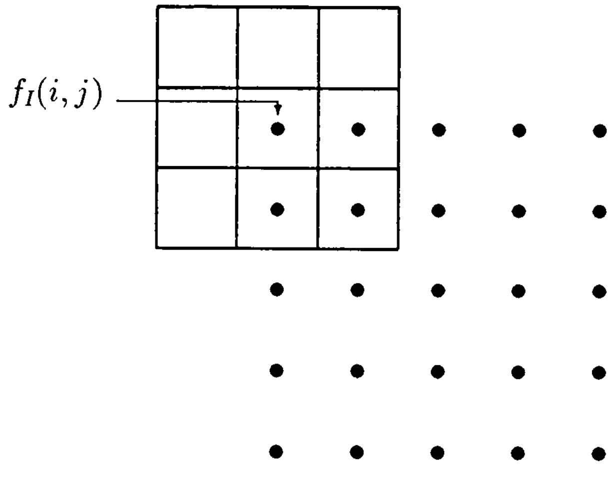

Figure 3.5: A 3 x 3 Window Falling Beyond the Boundaries of an N x N Image, f_I.

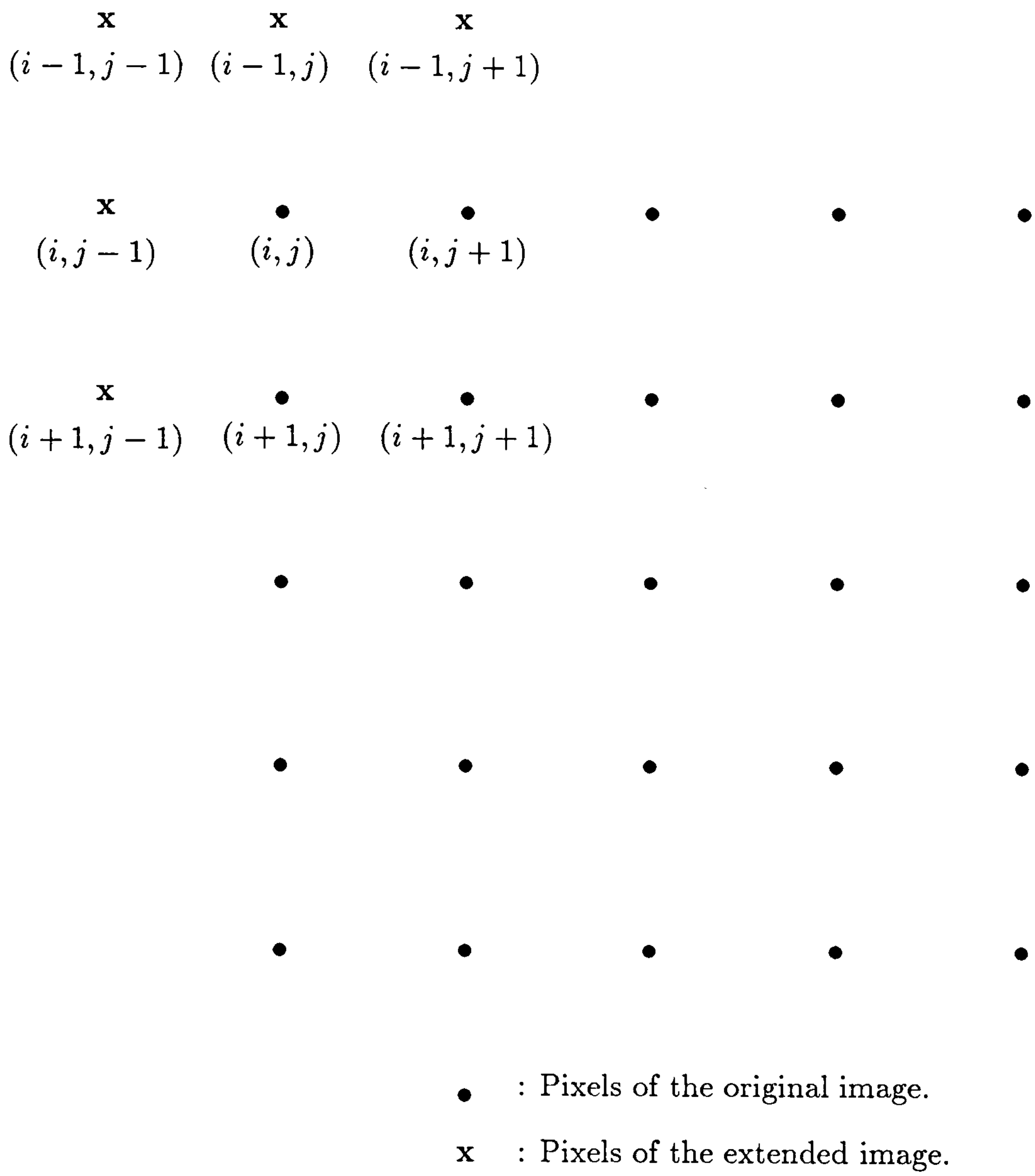

Figure 3.6: Generation of Neighbors for $f_I(i, j)$.

m Rows Reflected Around The First Row

ORIGINAL IMAGE

(-m, -m)	•••	(-m, -1)	(-m, 1)	(-m, 2)	(-m, 3)	•••	(-m, N-1)	(-m, N)	(-m, N+2)	•••	(-m, N+m+1)
⋮		⋮	⋮	⋮	⋮		⋮	⋮	⋮		⋮
(-2, -m)	•••	(-2, -1)	(-2, 1)	(-2, 2)	(-2, 3)	•••	(-2, N-1)	(-2, N)	(-2, N+2)	•••	(-2, N+m+1)
(-1, -m)	•••	(-1, -1)	(-1, 1)	(-1, 2)	(-1, 3)	•••	(-1, N-1)	(-1, N)	(-1, N+2)	•••	(-1, N+m+1)
(1, -m)	•••	(1, -1)	(1, 1)	(1, 2)	(1, 3)	•••	(1, N-1)	(1, N)	(1, N+2)	•••	(1, N+m+1)
(2, -m)	•••	(2, -1)	(2, 1)	(2, 2)	(2, 3)	•••	(2, N-1)	(2, N)	(2, N+2)	•••	(2, N+m+1)
(3, -m)	•••	(3, -1)	(3, 1)	(3, 2)	(3, 3)	•••	(3, N-1)	(3, N)	(3, N+2)	•••	(3, N+m+1)
⋮		⋮	⋮	⋮	⋮		⋮	⋮	⋮		⋮
(N-1, -m)	•••	(N-1, -1)	(N-1, 1)	(N-1, 2)	(N-1, 3)	•••	(N-1, N-1)	(N-1, N)	(N-1, N+2)	•••	(N-1, N+m+1)
(N, -m)	•••	(N, -1)	(N, 1)	(N, 2)	(N, 3)	•••	(N, N-1)	(N, N)	(N, N+2)	•••	(N, N+m+1)
(N+2, -m)	•••	(N+2, -1)	(N+2, 1)	(N+2, 2)	(N+2, 3)	•••	(N+2, N-1)	(N+2, N)	(N+2, N+2)	•••	(N+2, N+m+1)
(N+3, -m)	•••	(N+3, -1)	(N+3, 1)	(N+3, 2)	(N+3, 3)	•••	(N+3, N-1)	(N+3, N)	(N+3, N+2)	•••	(N+3, N+m+1)
⋮		⋮	⋮	⋮	⋮		⋮	⋮	⋮		⋮
(N+m+1, -m)	•••	(N+m+1, -1)	(N+m+1, 1)	(N+m+1, 2)	(N+m+1, 3)	•••	(N+m+1, N-1)	(N+m+1, N)	(N+m+1, N+2)	•••	(N+m+1, N+m+1)

m Columns Reflected Around The First Column

m Rows Reflected Around The Last Row

m Columns Reflected Around The Last Column

Figure 3.7: Neighbors are generated at coordinates shown when the pixel reflection method is applied to the original N x N image. The window size is $(2n+1)$ x $(2m+1)$, $n = m = 1, 2, \ldots$.

$$\begin{aligned} \text{for every } i &= 1, 2, ..., m \\ \text{let } j &= 1, 2, ..., N \end{aligned}$$

2. Reflect m rows around the last row, such that:

$$f_I(\mathrm{N} + i + 1, j) = f_I(\mathrm{N} - i, j) \tag{3.6}$$

$$\begin{aligned} \text{for every } i &= 1, 2, ..., m \\ \text{let } j &= 1, 2, ..., N \end{aligned}$$

3. Reflect m columns around the first column, so that:

$$f_I(i, -j) = f_I(i, j + 1) \tag{3.7}$$

$$\begin{aligned} \text{for every } i &= -m, ..., -2, -1, 1, 2, ..., \mathrm{N} + m + 1 \\ \text{let } j &= 1, 2, ..., m \end{aligned}$$

4. Reflect m columns around the last column, such that:

$$f_I(i, \mathrm{N} + j + 1) = f_I(i, \mathrm{N} - j) \tag{3.8}$$

$$\begin{aligned} \text{for every } i &= -m, ..., -2, -1, 1, 2, ..., \mathrm{N} + m + 1 \\ \text{let } j &= 1, 2, ..., m \end{aligned}$$

Examples shown in Figures 3.8 and 3.9 illustrate the described procedure.

99	114	93	104	90
110	98	105	122	74
79	81	124	106	94
93	104	115	113	69
101	82	119	78	117

Figure 3.8: A 5 x 5 Digital Image.

98	*110*	*98*	*105*	*122*	*74*	*122*
114	**99**	**114**	**93**	**104**	**90**	*104*
98	**110**	**98**	**105**	**122**	**74**	*122*
81	**79**	**81**	**124**	**106**	**94**	*106*
104	**93**	**104**	**115**	**113**	**69**	*113*
82	**101**	**82**	**119**	**78**	**117**	*78*
104	*93*	*104*	*115*	*113*	*69*	*113*

Figure 3.9: Result of applying pixel reflection to the image of Figure 3.8, for $m = n = 1$. **Bold** numbers represent original pixel values, while *italic* numbers represent extended pixel values.

3.4 PERFORMANCE EVALUATION OF ANF

The method of Automated Noise Filtering has been tested on numerous images. In addition, performance of ANF has been compared to an alternative noise removal technique developed by Lee [11]. The comparison is based on:

1. The ability to remove noise without destroying vital signal information when applied to:
   (a) An artificial image specially generated to illustrate how and when either of the two methods is effective.
   (b) FLIR and TV images.
2. Computational efficiency.

The essence of Lee's method is also Noise Filtering by Use of Local Statistics. Lee estimates noise variance for each pixel $g(x, y)$ separately by averaging the five smallest variances associated with pixels in a 7 x 7 window centered at $g(x, y)$. Then, edge preservation is employed by redefining the neighborhood (where the local statistics are computed) of a pixel near the regions where the degraded image shows abrupt changes (these changes are detected by an edge detection method [17]). Redefining the neighborhood of a pixel occurs only when the corresponding local variance computed from its 7 x 7 neighborhood is greater than some arbitrarily chosen threshold variance (TVAR).

3.4.1 Ability to Remove Noise and Preserve Vital Signal Information

Processing an Artificial Image. Implementation of Automated Noise Filtering, Figure 3.3, is straight forward. It does not require a priori knowledge about any parameter or any particular conditions regarding the image properties. On the other hand, as it will be illustrated, Lee's algorithm requires: (1) a priori knowledge about the threshold variance, and (2) that there exists no more than one edge within the 7 x 7 region of the degraded image. If either one of these requirements is not met, this technique will fail to produce the desired results, i.e., remove noise and preserve edges.

The artificial image shown in Figure 3.10a was generated to illustrate the effects of threshold variance on the obtained results. In this example, no more than one edge occurs in each 7 x 7 window. In addition, the exact value of the desired threshold variance is known (TVAR = 2984.69). Using these ideal conditions, the best possible result (Figure 3.10b) has been generated. However, such conditions are rarely met when dealing with real images. A more realistic result that is likely to be obtained by Lee's technique is shown in Figure 3.10d, where a threshold value of TVAR = 800 was chosen. In this case, underestimation of TVAR has caused notable changes in the resulting image. In essence, underestimation of TVAR causes neighborhoods to be redefined unnecessarily. On the other hand, overestimation of this parameter allows only a limited number of neighborhoods to be redefined, making this method equivalent to NFLS. The result obtained by using ANF is shown in Figure 3.10c.

In conclusion Lee's method has the potential to perform very well only when the ideal conditions are met. Under realistic conditions ANF performs

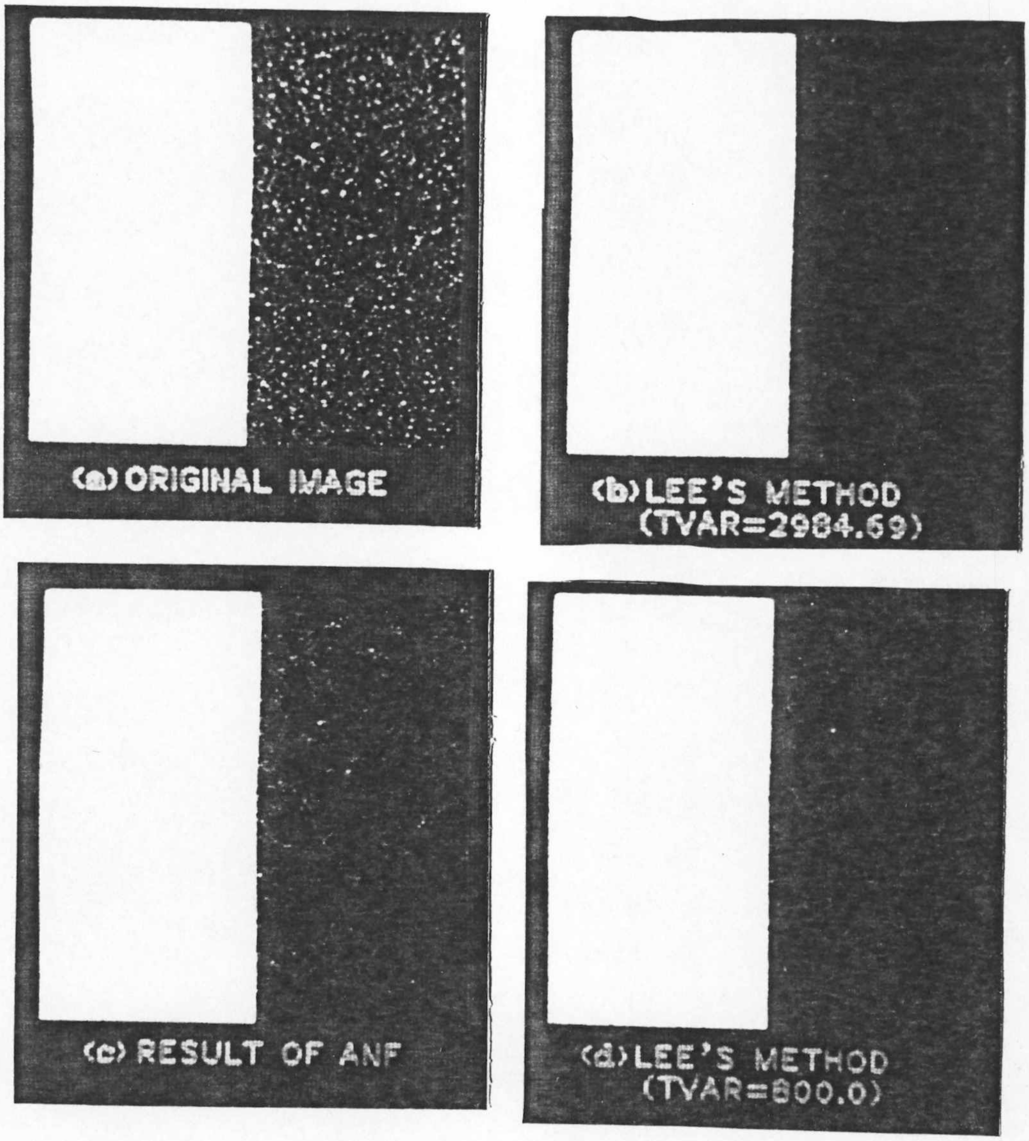

Figure 3.10: Comparison of Obtained Results.

consistantly, and on the average outperforms Lee's algorithm. Typical results are shown in Figure 3.11.

Processing Real Images. Images shown in Figure 3.12 show the ability of the two methods to preserve edges and details. As illustrated in these images, the method of Automated Noise Filtering proved to be very successful in preserving edges and small detail. On the other hand, application of Lee's algorithm resulted in edge blurring and loss of detail despite of the fact that variance distribution graphs such as the one shown in Figure 3.13 were used to help choose a reasonable value for threshold variance.

Edge blurring caused by Lee's method can be explained by the fact that the noise variance is computed for each pixel by averaging the five smallest local variances corresponding to that pixel and its neighbors (within a 7 x 7 window). Consequently, σ^2 at coordinates (x, y) is very close to the pixel variance at those coordinates. From Equation 2.13 it is seen that as $\sigma^2 \longrightarrow V(x,y) \Longrightarrow \hat{f}(x,y) \longrightarrow \overline{g}(x,y)$. Automated Noise Filtering avoids edge blurring by choosing a σ^2 which is significantly smaller than the variance in the regions where an edge is present.

ANF can also be used to process images that are noise free, e.g., TV images. Presence of regions of uniform intensity in such images helps this method to estimate a very low value for σ^2 ,and therefore, generate a resultant image which is very close to the original one. It is also evident from Equation 3.3, that as $\sigma^2 \longrightarrow 0 \Longrightarrow \hat{f}(x,y) \longrightarrow g(x,y)$. On the other hand, due to the blurring caused by Lee's technique, processing of TV images by this method introduces undesirable changes in them. Illustrations of such effects are shown in Figure 3.14.

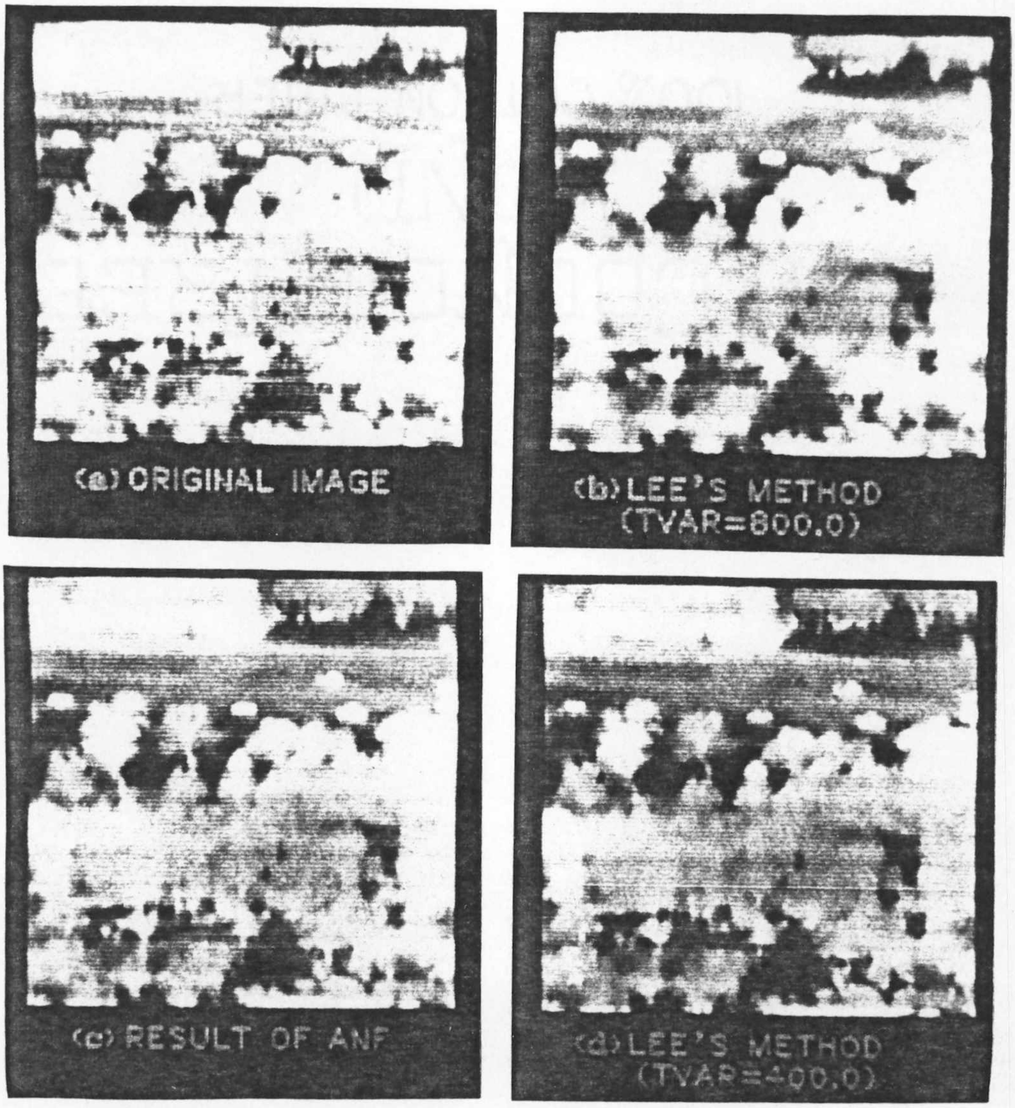

Figure 3.11: Comparison of Obtained Results.

Figure 3.12: Edge Preservation Ability.

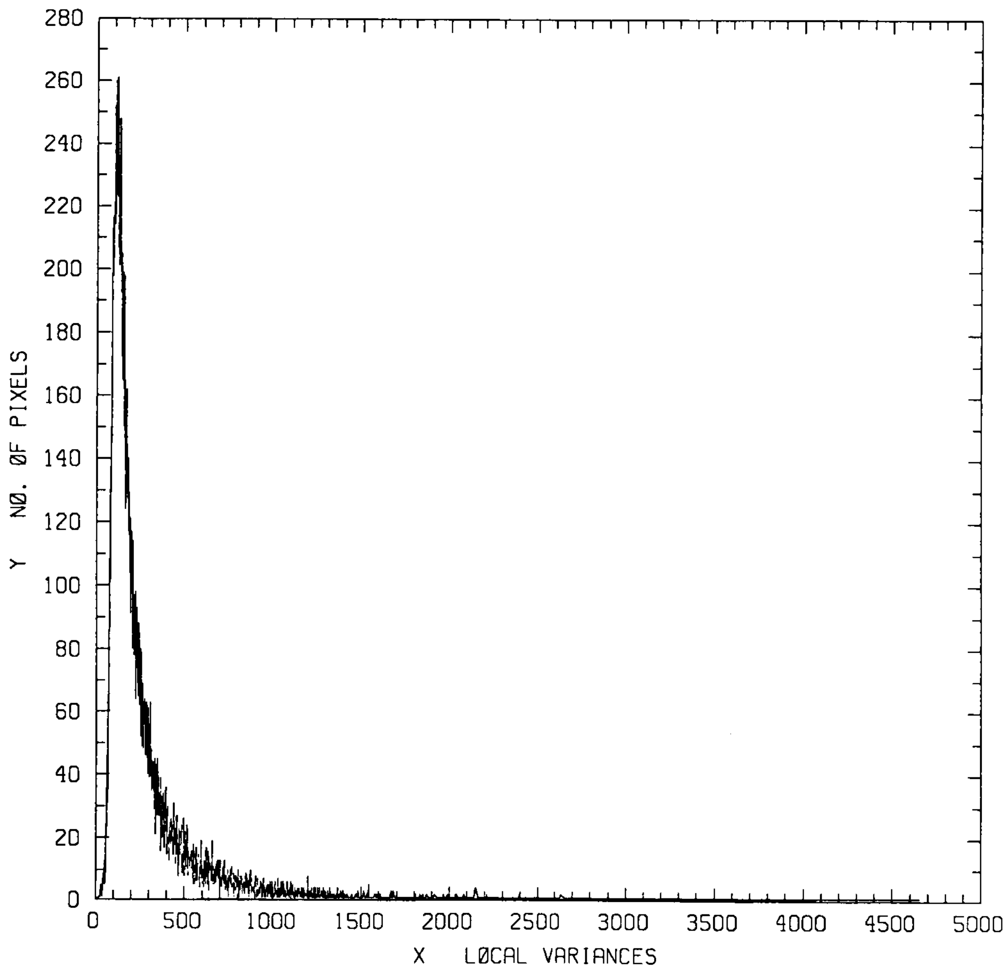

Figure 3.13: Distribution of Local Variances.

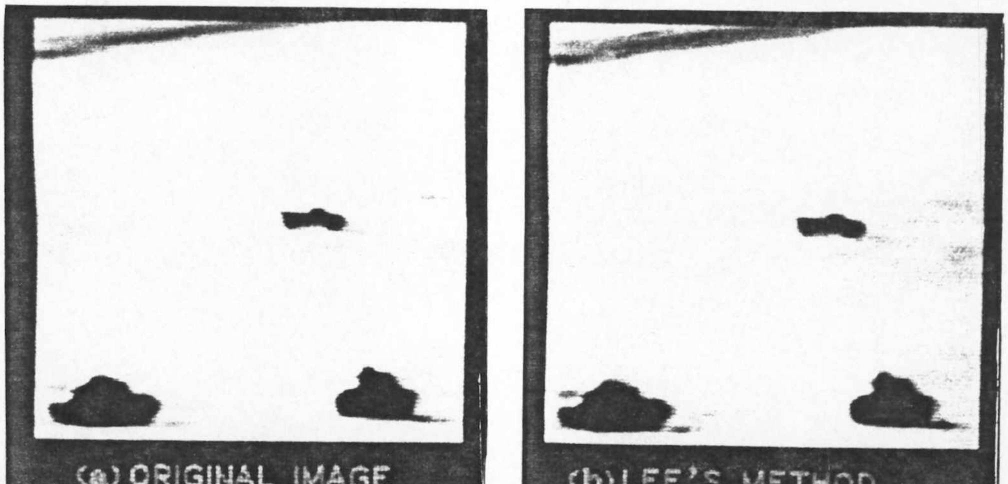

Figure 3.14: Processing Images with Uniform Regions.

3.4.2 Computational Efficiency

The method of Automated Noise Filtering is computationally more efficient than Lee's algorithm because of the following:

1. While both methods compute all the local variances, Lee's technique requires their modification afterwards.

2. Lee's algorithm computes a noise variance corresponding to each pixel. In contrast, ANF calculates σ^2 only once and it remains constant for the whole image.

Eventhough results obtained by applying ANF to degraded images are satisfactory, presence of noise can still be detected visually in some of the output images. Since it is desired to remove the additive noise completely from the original image, the need for further removal of noise has brought about the idea of developing an iterative filter as an extension to ANF. Derivation of an iterative filter is the subject of Chapter 4.

CHAPTER 4

ITERATIVE NOISE FILTERING

Automated Noise Filtering does not remove the additive noise completely in cases where images are degraded by high levels of noise. In order to satisfy the criterion of *complete removal of noise*, described in Chapter 1, the method of Iterative Noise Filtering is proposed in this work. This iterative algorithm and its performance are discussed in detail in this Chapter.

In general, development of an iterative filter as an extension to Automated Noise Filtering, requires:

1. Formulation of the iterative process.

2. Development of a method for selection of the window size within which the local statistics are calculated during each iteration.

3. Development of the procedure for automatic estimation of the additive noise variance during each iteration.

4. Establishment of a criterion that terminates the iterative process automatically.

Formulation of the iterative process, and considerations given to the selection of the window size are presented in Section 4.1. Development of the procedure for automatic estimation of the additive noise variance as well as establishment of the criterion for automatic termination of the iterative process, because of their crucial role in the method of Iterative Noise Filtering, are discussed separately in Sections 4.2 and 4.3 respectively.

4.1 FORMULATION OF INF AND CHOICE OF WINDOW SIZE

Formulation of the iterative process is fairly straightforward, and can be achieved by introducing iterative notation into Equation 3.3, i.e.,

$$\hat{f}_q(x,y) = \begin{cases} \overline{g}_{q-1}(x,y) + \dfrac{V_{q-1}(x,y) - \sigma_{q-1}^2}{V_{q-1}(x,y)} \left[\hat{f}_{q-1}(x,y) - \overline{g}_{q-1}(x,y)\right] & \\ \qquad\qquad \text{if} \qquad V_{q-1}(x,y) \geq \sigma_{q-1}^2 & \\ \hat{f}_{q-1}(x,y) \qquad \text{if} \qquad V_{q-1}(x,y) = 0 & \\ \hat{f}_{q-1}(x,y) \qquad \text{if} \qquad V_{q-1}(x,y) < \sigma_{q-1}^2 & \end{cases} \tag{4.1}$$

where:

$q = 1, 2, 3, ...$	:	Number of iteration.
$\hat{f}_k(x,y)$	:	Output of the iterative filter after (k-1)
	:	iterations.
$\hat{f}_0(x,y) = g(x,y)$	:	Original noisy image.
$\overline{g}_k(x,y)$ and $V_k(x,y)$	:	Local mean and variance calculated from
	:	$\hat{f}_k(x,y)$ within a $[(2n+1) \text{ x } (2m+1)]_k$
	:	window in the $(k+1)^{th}$ iteration.
σ_k^2	:	Additive noise variance in the $(k+1)^{th}$
	:	iteration.

It is seen from Equation 4.1 that in the first iteration, i.e., $q = 1$, equation of Iterative Noise Filtering becomes identical to that of ANF. In the subsequent iterations, output image of the $(q-1)^{th}$ iteration becomes the input image in the q^{th} iteration.

The next step in formulating INF is to consider the choice of the window size within which the local statistics are computed during each iteration. Since the first iteration of the iterative process is intended to be identical to the method of Automated Noise Filtering, a 7 x 7 window size is used. Experimental results showed that if the window size is not decreased on the second iteration or thereafter, edges and small details contained in input images can no longer be preserved. Local statistics are therefore computed over a $[(2n+1) \text{ x } (2m+1)]_q$ window in the q^{th} iteration, such that:

$$[(2n+1) \text{ x } (2m+1)]_q = \begin{cases} 7 \text{ x } 7 \text{ if } q = 1 \\ 3 \text{ x } 3 \text{ if } q > 1 \end{cases} \tag{4.2}$$

This choice is based on experimental results as well as considerations regarding the computational complexity.

4.2 ADDITIVE NOISE VARIANCE ESTIMATION IN INF

As in Automated Noise Filtering, estimation of the additive noise variance is the most crucial step in the iterative filtering process. Underestimation or overestimation of this parameter can result in no noise removal or image blurring, respectively. Furthermore, the negative effects of determining an incorrect value for noise variance magnify (during each iteration) in an iterative process, resulting in undesirable changes in the final output. Keeping this in mind, based on theoretical considerations and experimental evidence, the following procedure for automatic estimation of the additive noise variance in iterative filtering is proposed.

Analogous to ANF, in the first iteration, the noise variance is estimated by averaging K% (K = 5, refer to Section 3.2) of the smallest local variances (calculated within a 7 x 7 window), i.e., in iterative notation

$$\sigma_0^2 = \frac{1}{N \times M \times \frac{5}{100}} \sum_{i=1}^{N \times M \times \frac{5}{100}} Var_0(i) \tag{4.3}$$

where:

$Var_0(i)$	:	Local variances ($V_0(x, y)$) arranged in ascending order.
σ_0^2	:	Additive noise variance in the first iteration.
$M \times N$	:	Dimensions of the noisy image.

In the second iteration, noise variance is estimated in a different manner. This is necessary since the local variances calculated within a 3 x 3 window, correspond to smaller variations than those calculated within a 7 x 7 window. Furthermore, the input image in the second iteration, $\hat{f}_1(x, y)$, contains significantly smaller amount of noise than $\hat{f}_0(x, y)$. Consequently, in the second iteration there exists an overall drop in the values of the local variances computed from the input image, causing the percentage of the smallest local variances corresponding to uniform regions to increase at this step. Thus, a different method of obtaining the smallest local variances which represent regions of uniform intensity is required. In this work the noise variance obtained from the first iteration, σ_0^2, is chosen to be the upper limit of local variances to be considered (in the second iteration) when determining σ_1^2. Local variances smaller than σ_0^2 are averaged to give the additive noise variance used in the second iteration, i.e.,

$$\sigma_1^2 = \frac{1}{l-1} \sum_{i=1}^{l-1} Var_1(i) \tag{4.4}$$

where:

$Var_1(i)$	:	Local variances ($V_1(x,y)$) arranged in ascending order.
σ_1^2	:	Additive noise variance in the second iteration.
l	:	Smallest number for which $Var_1(l) = \sigma_0^2$.

The same procedure for automatic determination of noise variance is followed in the subsequent iterations. For example in the fourth iteration, noise variance (σ_3^2) is estimated by averaging those local variances that are found to be less than the noise variance obtained in the third iteration (σ_2^2). Consequently, after the first iteration the additive noise variance is estimated as

$$\sigma_q^2 = \frac{1}{l-1} \sum_{i=1}^{l-1} Var_q(i) \tag{4.5}$$

where:

$Var_k(i)$	:	Local variances ($V_k(x,y)$) arranged in ascending order.
σ_k^2	:	Additive noise variance in the $(k+1)^{th}$ iteration.
l	:	Smallest number for which $Var_k(l) = \sigma_{k-1}^2$.

As will be proven in the following Section, the noise variance estimated in this way is ensured to decrease consistently during each consecutive iteration, i.e., $\sigma_k^2 < \sigma_{k-1}^2 < \ldots < \sigma_2^2 < \sigma_1^2$. Intuitively, a decrease in the value of noise variance is an indicator of noise removal during each iteration.

4.3 AUTOMATIC TERMINATION OF INF

This criterion for automatic termination of the iterative process is developed based on:

Theorem 4.3.1 The noise variance estimated in Iterative Noise Filtering decreases consistently during each consecutive iteration, and

$$\lim_{q \to \infty} \sigma_q^2 = 0.$$

Proof:

In the k^{th} *iteration largest local variance corresponding to regions of uniform intensity,* $Var_k(l)$*, is set equal to the noise variance obtained from the previous iteration, that is*

$$Var_k(l) = \sigma_{k-1}^2 \tag{4.6}$$

Where:

$Var_k(i)$: *Local variances computed from the* M x N *input image and arranged in ascending order so that* $Var_k(0) < Var_k(1) < \ldots < Var_k(\mathrm{M} \times \mathrm{N})$.

The noise variance in this iteration is obtained by Equation 4.5, i.e.,

$$\sigma_k^2 = \frac{1}{l-1} \sum_{i=1}^{l-1} Var_k(i). \tag{4.7}$$

It follows from Equation 4.7 that

$$\sigma_k^2 \leq Var_k(l-1). \tag{4.8}$$

Furthermore,

$$Var_k(l-1) < Var_k(l). \tag{4.9}$$

It follows from Equations 4.8 and 4.9 that

$$\sigma_k^2 < Var_k(l). \tag{4.10}$$

Finally, by considering Equations 4.6 and 4.10 it is seen that

$$\sigma_k^2 < \sigma_{k-1}^2 \tag{4.11}$$

and therefore

$$\lim_{q \to \infty} \sigma_q^2 = 0. \tag{4.12}$$

Theorem 4.3.1 and the graph shown in Figure 4.1, prove that regardless of how noisy the image is in the first iteration, the noise variance estimated during each iteration approaches zero as the number of iterations increases. It is also evident by inspection of Equation 4.1 that as noise variance approaches zero, the output image ($\hat{f}_k(x,y)$) becomes identical to the input image ($\hat{f}_{k-1}(x,y)$). As a consequence of Theorem 4.3.1 and Equation 4.1, the iterative process of the method of Iterative Noise Filtering is convergent.

The iterative process is thus terminted automatically only when the output image in the k^{th} iteration is identical to the input image in that iteration. Computational procedure shown in Figure 4.2 illustrates the steps involved in the method of Iterative Noise Filtering.

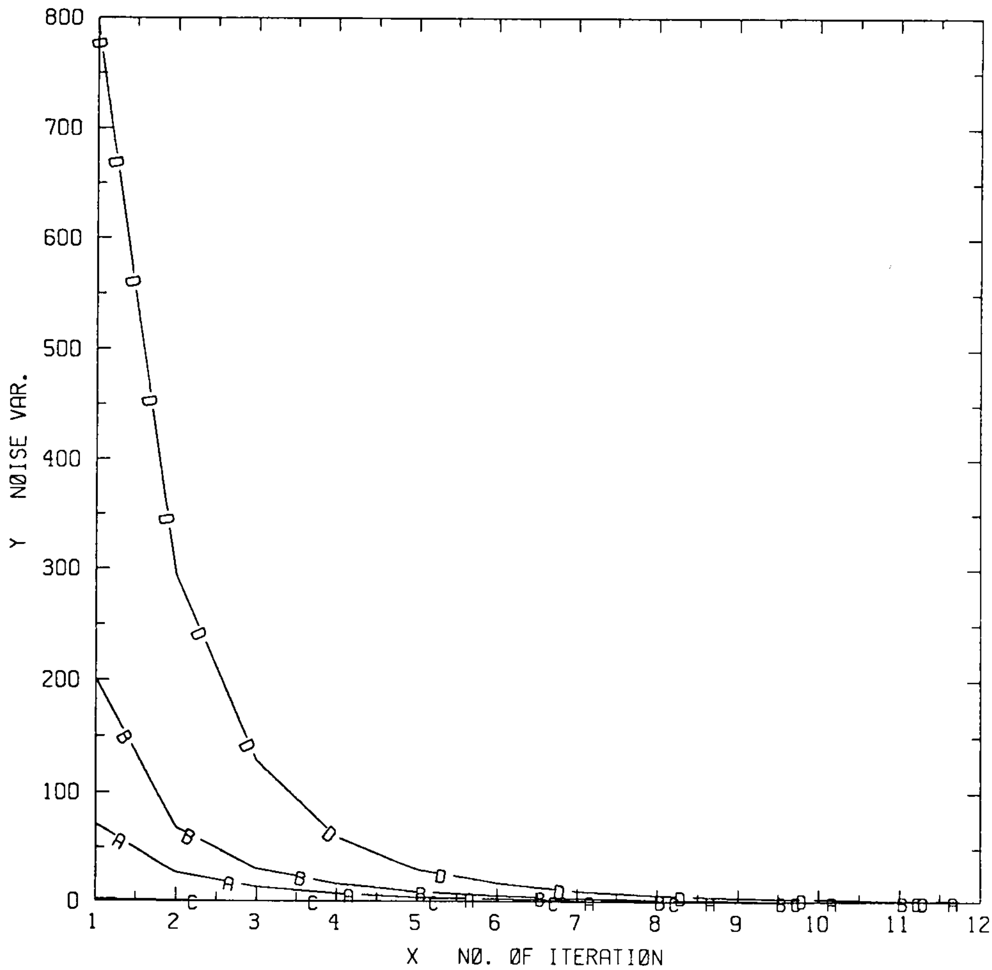

Figure 4.1: Behavior of Noise Variance *vs.* Number of Iteration.

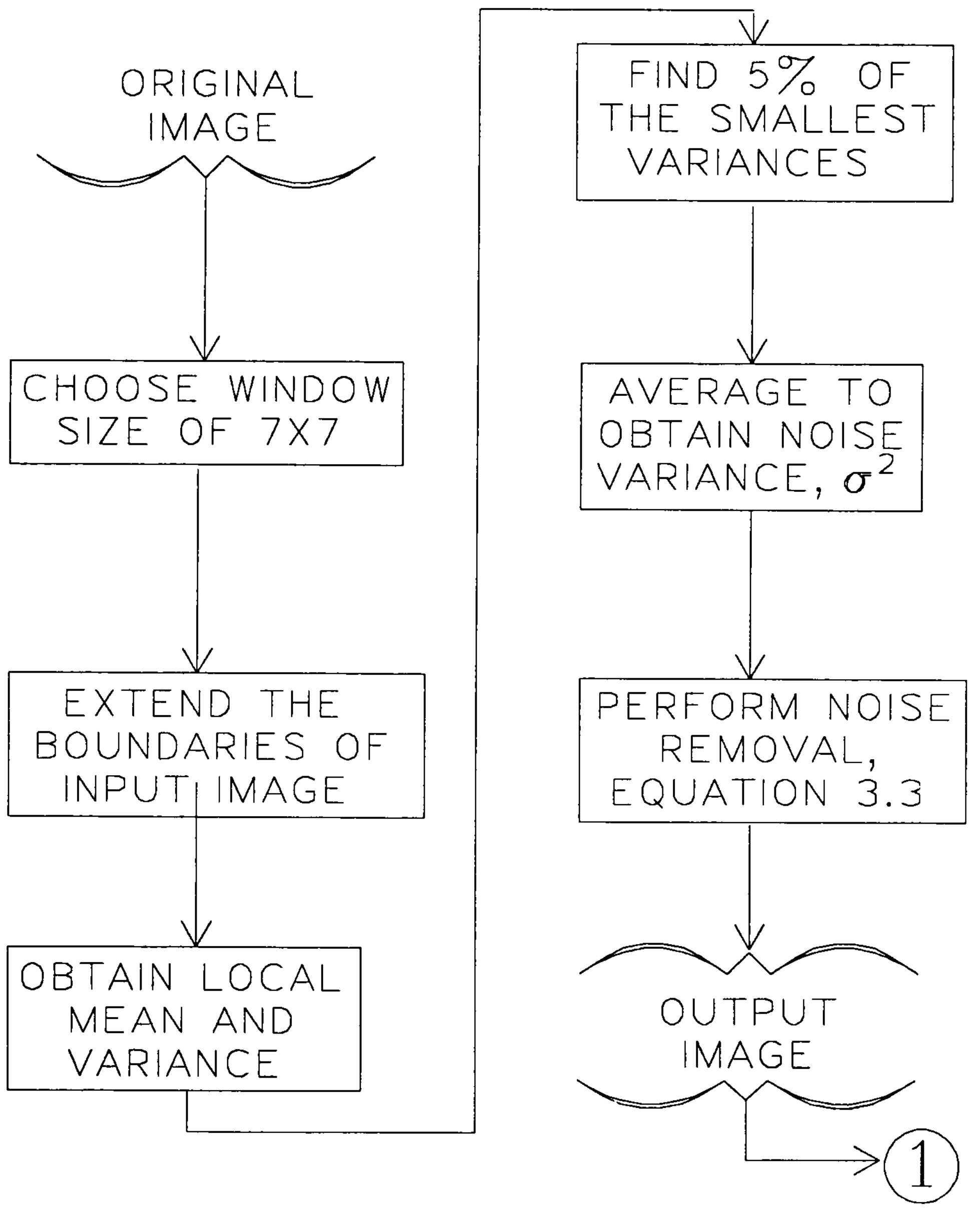

Figure 4.2: Computational Procedure for Iterative Noise Filtering.

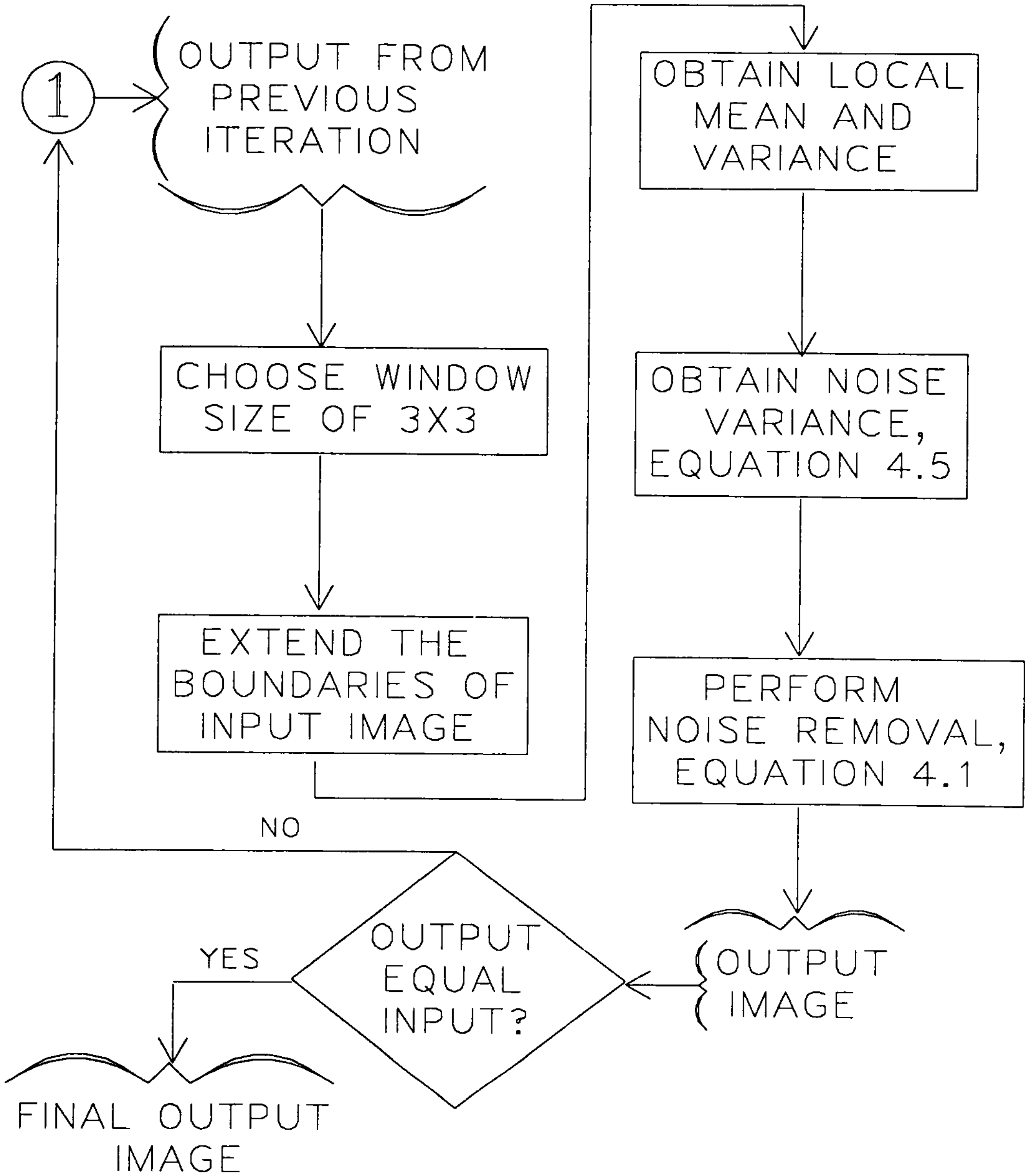

Figure 4.2: Continued.

4.4 PERFORMANCE EVALUATION OF INF

The method of Iterative Noise Filtering has been tested on numerous images. In addition, its performance has been compared to an alternative technique of additive noise removal, called Iterative Enhancement of Noisy Images (IENI) [13]. These two methods are evaluated based on:

1. The ability to remove noise without destroying vital signal information when applied to:
   (a) An artificial image specially generated to illustrate how and when either of the two methods is effective.
   (b) FLIR and TV images.
2. Computational efficiency.

The method of IENI assumes that the images under consideration are approximately piecewise constant, and estimates a neighbor-weighted average for each pixel centered within a 3 x 3 neighborhood of the noisy image. The iterative application of this method, described by Equation 4.13, compensates for the small size of the neighborhood within which the averaging takes place, and causes further removal of noise,

$$f^{(k)}(i,j) = \sum\sum D^{(k)}(u,v) f^{(k-1)}(u,v) \tag{4.13}$$

where:

$f^{(k)}(i,j)$	:	Estimated gray level of point (i,j) at k^{th} iteration.
$f^{(0)}(i,j)$	:	Originally observed gray level of (i,j).
$D^{(k)}(u,v)$	:	Nonnegative weights that sum to 1 at k^{th} iteration.
(u,v)	:	Neighbors of (i,j) within a 3 x 3 window.

An obvious shortcoming of the method of IENI is the fact that, unlike Iterative Noise Filtering, the iterative process is not terminated automatically, instead, it is done by visually inspecting the results at the end of each iteration.

Lev et al. [13] consider two methods of constructing the weights $D^{(k)}(u,v)$. They report that the first method called the *Edge and Line Weights*, yields poor results, thus, this method is not used in this work. The second method, called the *Contrast-Sensitive Weights*, computes the final weight, $D^{(k)}(u,v)$, as a product of four intermediate weight patterns $D_i^{(k)}$, $0 \leq i \leq 3$, each corresponding to one of the directions $0°, 45°, 90°$, and $135°$. The comparison between Iterative Noise Filtering and IENI is performed by using *Contrast-Sensitive Weights.*

4.4.1 Ability to Remove Noise and Preserve Vital Signal Information

Processing an Artificial Image. The method of Iterative Noise Filtering has the ability to preserve edges and small details in the input image. This ability is due to estimating a value for σ_q^2 that is significantly smaller than the local variances computed in the regions where an edge is present. The method of IENI, on the other hand, preserves only those edges that can be modeled by the intermediate weight patterns. Thus, any edge with an orientation other than those modeled by $D_i^{(k)}$ is blurred.

To illustrate this, the artificial image of Figure 4.3a was generated. A 3 x 3 subregion within the encircled region of this image has the following intensities:

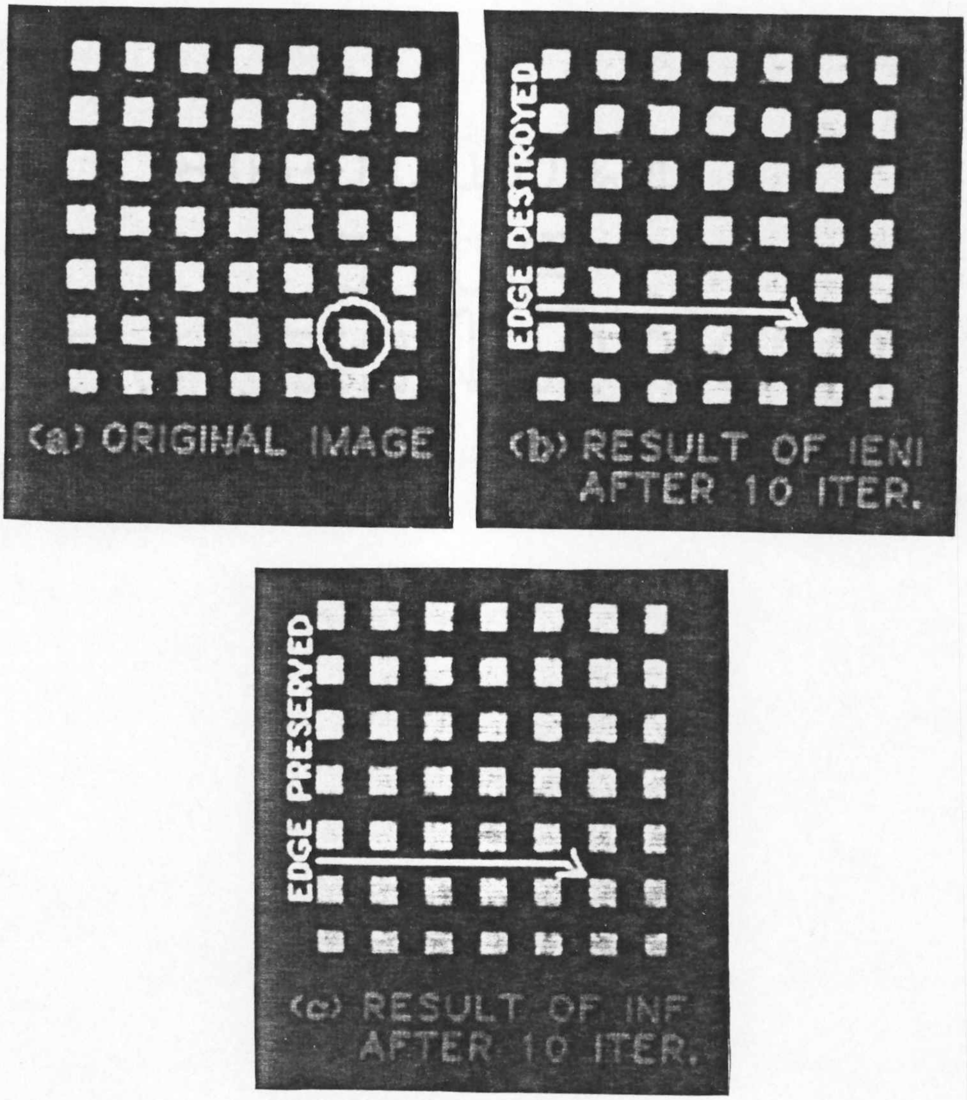

Figure 4.3: Artificial image generated to test the ability of the two methods in preserving edges.

57	53	165
48	45	135
32	45	128

It is evident that there is a vertical edge present in this subregion. The final weight pattern, $D^{(1)}(u,v)$, corresponding to this subregion is computed as:

0.31	2.2×10^{-11}	7.5×10^{-29}
2.9×10^{-6}	0.11	1.7×10^{-30}
0.57	3.5×10^{-6}	8.2×10^{-22}

As illustrated by the pattern of these weights, pixels that do not belong to the same neighborhood as the center pixel (45) are rightfully weighted less than those that do.

Consider another 3 x 3 subregion of the same encircled region of Figure 4.3a :

92	81	47
53	165	194
45	135	142

It is obvious that there is a *corner edge* present in this subregion. The computed weight pattern for this particular subregion is

3.3×10^{-13}	2.6×10^{-3}	0.87
6.0×10^{-11}	0.11	1.1×10^{-2}
4.5×10^{-16}	4.5×10^{-18}	8.0×10^{-4}

As the pattern of weights show, the top right pixel in the given subregion (47), that does not belong to the edge, is weighted more than the two edge pixels (135) and (194). This distribution of weights causes a tremendous drop in the value of the center pixel (from 165 to 62), and therefore, destroys the edge which was present in the subregion.

In contrast, Iterative Noise Filtering takes in the 7 x 7 neighborhood of the pixel from which the noise is to be removed (165), namely

33	70	55	24	87	60	61
30	19	40	41	57	77	43
45	60	92	81	47	49	72
20	57	53	165	194	173	170
113	48	45	135	142	133	178
47	32	45	128	148	165	136
19	31	53	164	153	152	142

and computes a local mean of $\overline{g}(x, y) = 85.4$ (Equation 2.2), a local variance of $V(x, y) = 2759.4$ (Equation 2.3), and estimates the additive noise variance for the encircled region of Figure 4.3a to be $\sigma_0^2 = 227.0$ (Equation 4.5). This is significantly less than the value computed for the local variance. Substitution of all these values in Equation 4.1 yields an estimate of $\hat{f}_1(x, y) = 158$ for the center pixel and thus the existing edge is preserved.

Images of Figures 4.3b and 4.3c are the results obtained after ten iterations of the methods of IENI and Iterative Noise Filtering, respectively. Note that because of the problems discussed, corners of the squares in Figure 4.3b are destroyed. Tables 4.1 and 4.2 show how much noise was removed by each method as well as the maximum values by which a pixel intensity was increased or decreased at the end of each iteration. Signs of instability are apparent for the method of IENI when looking at the numbers of Table 4.2. On the other hand, it is evident from Table 4.1 that noise is removed consistantly by the method of Iterative Noise Filtering during each iteration (the amount of removed noise decreases as the number of iterations increases) untill the tenth iteration when the process finally stops.

Processing Real Images. The ability of the methods INF and IENI to preserve edges was further tested on real images. The image shown in Figure 4.4a is an example of a FLIR image which contains delicate edges (represented by branches of the tree) as well as small details embedded in a region of constant intensity (represented by objects appearing on the road). Result of applying Iterative Noise Filtering to this image, Figure 4.4b, is generated after eight iterations (when the iterative process is terminated automatically), while noise is removed completely leaving the edges and details contained in the input image unchanged. Application of IENI however, has caused severe damage to the input image after only five iterations, Figure 4.4c.

Image of Figure 4.5a is an example of another FLIR image. It is obvious by inspection that this image is degraded by very low levels of noise. As a result process of Iterative Noise Filtering stopped after only four iterations giving the result shown in Figure 4.5b. Method of IENI has also produced a

Table 4.1: Data Obtained by Applying INF to the Image in Figure 4.3a.

Iteration number	Estimated noise variance	% of pixels from which noise was removed	Max. intensity by which pixels increased in value	Max. intensity by which pixels decreased in value
1	252.5	87.46	41	50
2	102.5	53.29	26	25
3	44.52	37.52	17	17
4	21.57	26.16	12	12
5	11.46	17.74	9	9
6	6.05	9.29	6	6
7	3.50	7.81	5	5
8	1.60	1.44	3	3
9	0.94	1.42	2	2
10	0.00	0.00	0	0

Table 4.2: Data Obtained by Applying IENI to the Image in Figure 4.3a.

Iteration number	% of pixels from which noise was removed	Max. intensity by which pixels increased in value	Max. intensity by which pixels decreased in value
1	97.09	69	103
2	95.49	78	86
3	93.49	50	72
4	90.11	51	54
5	84.54	68	60
6	78.05	56	47
7	69.74	32	52
8	61.78	41	41
9	53.63	37	35
10	46.86	42	44

Figure 4.4: A FLIR Image Processed by both Methods.

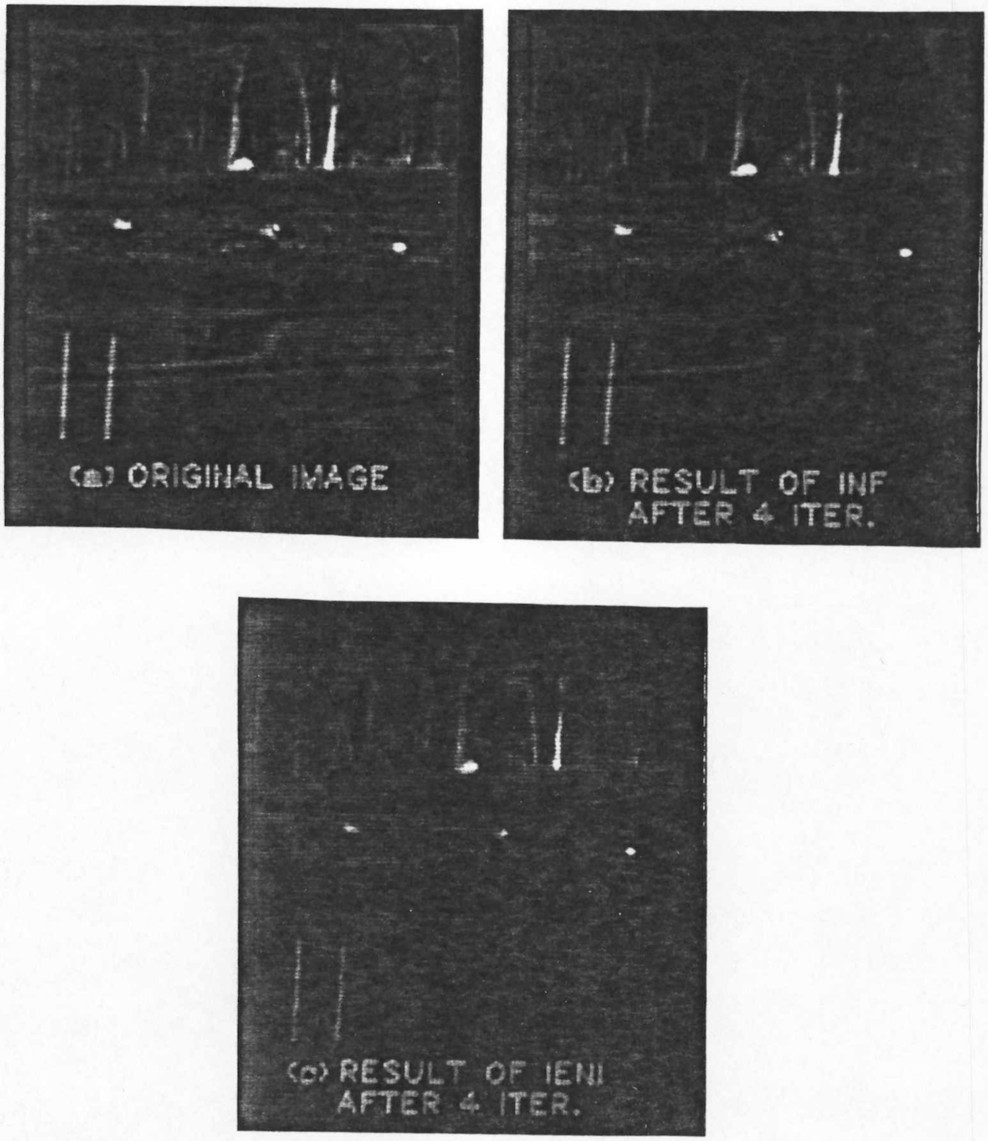

Figure 4.5: A FLIR Image Processed by both Methods.

satisfactory result after four iterations, Figure 4.5c.

The TV image of Figure 4.6a is also degraded by very low levels of noise. Iterative Noise Filtering was applied to this image producing the image of Figure 4.6b after three iterations. Process of IENI was stopped manually after three iterations leaving the image shown in Figure 4.6c. Note that edges of the dark objects are blurred in this image, whereas, the image of Figure 4.6b is almost identical to the original image.

4.4.2 Computational Efficiency

The method of Iterative Noise Filtering is computationally more efficient than IENI because:

1. The number of pixels processed by Iterative Noise Filtering decreases consistantly after each iteration. The method of IENI on the other hand, processes every pixel during each iteration.

2. In the process of noise removal, the method of IENI generates a series of very small *real* numbers which not only take a long time to be manipulated, but they also occupy a huge amount of the computer memory. Iterative Noise Filtering, on the other hand, only uses *integer* values which makes it fast and efficient.

In conclusion, Iterative Noise Filtering is a fully automated method which processes input images without requiring any a priori knowledge about the signal or the noise parameters. Furthermore, it terminates the iterative process automatically upon complete removal of noise. On the other hand, the method of Iterative Enhancement of Noisy Images does not terminate automatically, instead, the decision of when to terminate the process is done by

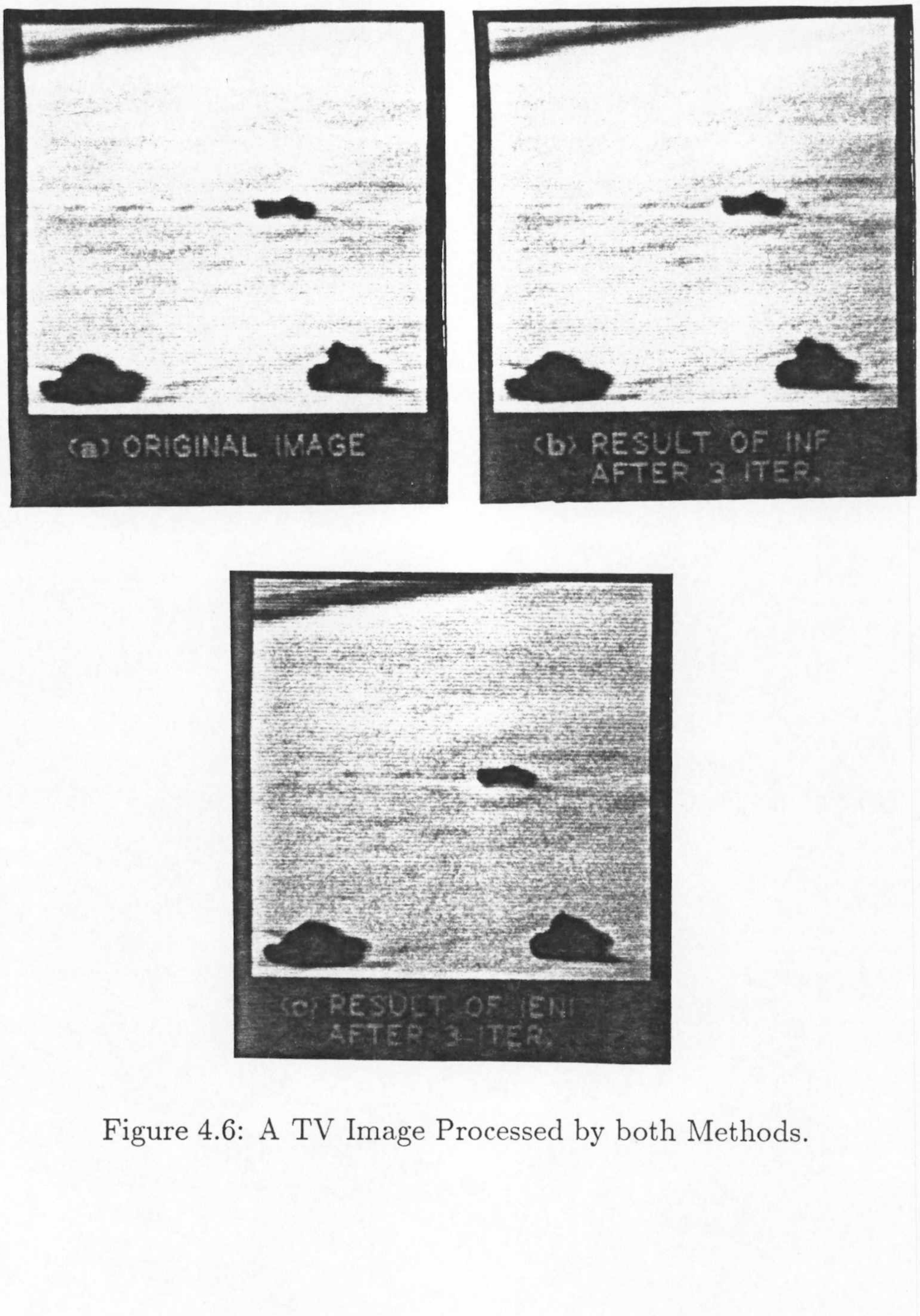

Figure 4.6: A TV Image Processed by both Methods.

visual inspection of the results. Iterative Noise Filtering preserves edges and small details without the use of any edge detection methods, and finally, it is computationally more efficient in comparison with the method of Iterative Enhancement of Noisy Images.

CHAPTER 5

CONCLUSIONS

As discussed in Chapter 1, the study described in this thesis is aimed towards developing a noise filtering technique which is:

1. Adaptive (because of conceptual simplicity and computational efficiency associated with adaptive filters [14]).
2. Suited for parallel processing.
3. Able to remove noise from images that are degraded by different levels of noise.
4. Able to preserve vital signal information, i.e., edges and details.
5. Automated and general, i.e., requires no a priori knowledge of signal or noise parameters.
6. Able to remove noise completely from highly degraded images.

The method of Noise Filtering by use of Local Statistics is investigated theoretically, and experimentally in Chapter 2. The advantages of this method are: (1) adaptivity, and (2) readiness for parallel processing. The disadvantages of NFLS are: (1) signal removal while processing images containing regions of uniform or semi-uniform intensity, and (2) requirement of a priori knowledge of additive noise variance.

The method of Automated Noise Filtering is developed and discussed in detail in Chapter 3. This method is the modified and automated version of NFLS. Therefore, in addition to possessing all the advantages of NFLS, ANF also overcomes its shortcomings. ANF is capable of preserving the fine detail and edges embedded in the degraded image. An evaluation of the performance of ANF and a comparison of this method to an alternative technique of noise removal proves its effectiveness. The method of Automated Noise Filtering meets all the requirements stated at the beginning of this chapter. However, when dealing with highly degraded images, ANF lacks the ability to remove the additive noise completely.

Iterative Noise Filtering is thus developed in Chapter 4 to satisfy the last requirement. INF is an extension to Automated Noise Filtering, and it removes additive noise completely from the input images (regardless of the level of degradation), while preserving the vital signal information. The iterative process terminates automatically after the additive noise is entirely removed. Performance of INF is compared to an alternative iterative method. The obtained results and computational efficiency were the deciding factors in choosing INF as the more effective method.

LIST OF REFERENCES

LIST OF REFERENCES

[1] Anderson, C.L. and Netravali, A.N., "Image Restoration Based on Subjective Criterion," *IEEE Trans. Syst. Man Cybern.*, Vol. SMC–6, pp. 845–853, Dec. 1976.

[2] Andrews, H.C., "Digital Image Restoration: A Survey," *IEEE Trans. Comput.*, Vol. 7, No. 5, pp. 36–45, 1974.

[3] Andrews, H.C. and Hunt, B.R., *Digital Image Restoration*, Prentice-Hall Inc., Englewood Cliffs, N.J., 1977.

[4] Bryson, A.E., Jr. and Ho, Y.C., *Applied Optimal Control*, Chapter 12, Waltham, MA: Blaisdell, 1969.

[5] Davis, L.S. and Rosenfeld, A., "Noise Cleaning by Iterated Local Averaging," *IEEE Trans. Syst. Man Cybern.*, Vol. SMC–8, pp. 705–710, 1978.

[6] Gonzalez, R.C. and Wintz, P., *Digital Image Processing*, New York: Wiley and Sons, 1977.

[7] Jain, A.K., "A Semicausal Model for Recursive Filtering of Two-Dimensional Images," *IEEE Trans. Comput.*, Vol. C–26, pp. 734-738, 1972.

[8] Kalman, R.E., "A New Approach to Linear Filtering and Prediction Problems," *Trans. ASME, J. Basic Eng.*, Vol. 82, pp. 35–45, 1960.

[9] Kuan, D.T., Sawchuk, A.A., Strand, T.C. and Chavel, P., "Adaptive Noise Smoothing Filter for Image with Signal-Dependent Noise," *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. PAMI–7, No. 2, pp. 165–177, March 1985.

[10] Lee, J.S., "Digital Image Enhancement and Noise Filtering by Use of Local Statistics," *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. PAMI–2, no.2, pp. 165–168, 1980.

[11] Lee, J.S., "Refined Noise Filtering Using Local Statistics," *Comput. Graphics Image Process.*, Vol. 15, pp. 380–389, 1981.

[12] Lee, J.S., "Digital Image Smoothing and the Sigma Filter," *Comput. Graphics Image Process.*, Vol. 24, pp. 255–269, 1983.

[13] Lev, A., Zucker, S.W. and Rosenfeld, A., "Iterative Enhancement of Noisy Images," *IEEE Trans. Syst. Man Cybern.*, Vol. SMC–7, pp. 435–442, 1977.

[14] Mastin, G.A., "Adaptive Filters for Digital Image Noise Smoothing: An Evaluation," *Comput. Vision Graphics Image Process.*, Vol. 31, pp. 103–121, 1985.

[15] Nagao, M. and Matsuyama, T., "Edge Preserving Smoothing," *Comput. Graphics Image Process.*, Vol. 9, pp. 394–407, 1979.

[16] Nahi, N.E. and Assefi, T., "Baysian Recursive Image Estimation," *IEEE Trans. Comput.*, Vol. C–12, pp. 734–738, 1972.

[17] Robinson, G.S., "Edge Detection by Compass Gradient Masks," *Comput. Graphics Image Process.*, Vol. 6, pp. 492–501, 1977.

[18] Sari–Sarraf, H.,"Automated Noise Filtering," *Proceedings of the Eighteenth Southeastern Symposium on System Theory*, IEEE Computer Society Press, pp. 531–535, 1986.

[19] Tomita, F. and Tsuji, S., "Extraction of Multiple Regions by Smoothing in Selected Neighborhoods," *IEEE Trans. Syst. Man Cybern.*, Vol. SMC–7, pp. 107–109, 1977.

[20] Wang, D.C.C., Vagnucci, A.H. and Li, C.C., "Gradient Inverse Weighted Smoothing Scheme and the Evaluation of its performance," *Comput. Graphics Image Process.*, Vol. 15, pp. 167–181, 1981.

VITA

Hamed Sari–Sarraf was born in Tabriz, Iran on January 5, 1962. He was graduated from the Tennessee Military Institute in Sweetwater, Tennessee in August 1979 and then entered Knoxville College in September of the same year. He entered the University of Tennessee in June 1981 and received his Bachelor of Science degree in Electrical Engineering in March 1984. He was a teaching assistant starting in June 1984 and a research assistant starting in September 1984 and received the Master of Science degree in Electrical Engineering in December 1986.