

THE UNIVERSITY OF CALGARY

**Adaptive-Neighborhood and Iterative Methods for Image
Restoration**

by

Tamer F. Rabie

A THESIS

**SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE**

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

CALGARY, ALBERTA

DECEMBER, 1992

© Tamer F. Rabie 1992



National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-83244-4

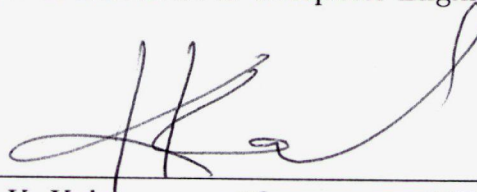
Canada

THE UNIVERSITY OF CALGARY
FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled, " Adaptive-Neighborhood and Iterative Methods for Image Restoration ", submitted by Tamer F. Rabie in partial fulfillment of the requirements for the degree of Master of Science.



Dr. R. M. Rangayyan, Supervisor
Dept. of Electrical & Computer Engineering



Dr. K. Kaler
Dept. of Electrical & Computer Engineering



Dr. P. Gong
Dept. of Geomatics Engineering

Date: November 23, 1992

ABSTRACT

As physical imaging systems are imperfect, and as the conditions under which images are obtained are frequently less than ideal, recorded images often represents a degraded version of the original scenes. Image restoration is concerned with filtering the recorded image to minimize the effect of such degradation. The majority of natural scenes consist of different features and edges. As such, most images cannot be considered to be statistically stationary. The application of spatially-invariant filters that assume stationary image models to the restoration of nonstationary images often produces unsatisfactory results. This thesis presents original image restoration techniques based on adaptive-neighborhood regions that are more stationary in nature. It will be shown that filters that are based on stationary regions perform better than those based on fixed, nonstationary regions. A new iterative method for blind deconvolution which does not require either the original image or the blur function will also be presented.

ACKNOWLEDGEMENTS

I would like to acknowledge the guidance and encouragement of my supervisor, Dr. Raj Rangayyan, who provided a research topic which was of intellectual interest. His continued enthusiasm and suggestions during the course of this work were instrumental in the results we have obtained.

Other members of our research group were also of great assistance. In particular, Dr. Raman Paranjape has made substantial contributions to the adaptive-neighborhood noise subtraction algorithm and the iterative blind deconvolution algorithm described herein.

Dedicated to
my parents and mentors,
for the talent, example, and knowledge

CONTENTS

APPROVAL PAGE	ii
ABSTRACT	iii
ACKNOWLEDGEMENTS	iv
DEDICATION	v
TABLE OF CONTENTS	vi
LIST OF TABLES	ix
LIST OF FIGURES	x

CHAPTERS

1. INTRODUCTION	1
1.1 Motivation for Research on Image Restoration	1
1.2 Classical Image Restoration Techniques	3
1.3 Thesis Layout	5
2. IMAGE RESTORATION TECHNIQUES	7
2.1 Introduction	7
2.2 Restoration of Noisy Images	8
2.2.1 Space Domain Filters	9
2.2.2 Frequency Domain Filters	12
2.3 Restoration of Blurred Images	15
2.4 Discussion	21
3. ADAPTIVE RESTORATION METHODS	23
3.1 Introduction	23
3.2 The Adaptive Two-dimensional LMS Algorithm	24
3.2.1 Derivation of the ATD-LMS Filter	25

3.2.2	Implementation of the ATD-LMS Filter	26
3.3	The Adaptive Rectangular Window LMS Algorithm	27
3.3.1	Derivation of the ARW-LMS Filter	28
3.3.2	Implementation of the ARW-LMS Filter	29
3.4	Experimental Results and Discussion	33
3.5	Summary	39
4.	ADAPTIVE-NEIGHBORHOOD NOISE SUBTRACTION	41
4.1	Introduction	41
4.2	Region Growing	42
4.3	The Adaptive-Neighborhood Noise Subtraction Method	46
4.3.1	Derivation of the ANNS filter	46
4.3.2	Implementation of the ANNS filter	48
4.4	Experimental Results and Discussion	50
4.5	Summary	58
5.	SECTIONED IMAGE DEBLURRING	59
5.1	Introduction	59
5.2	Fixed-Neighborhood Sectioned Deblurring	62
5.3	Experimental Results and Discussion	66
5.4	Summary	75
6.	ADAPTIVE-NEIGHBORHOOD IMAGE DEBLURRING	76
6.1	Introduction	76
6.2	The Adaptive-Neighborhood Deblurring Method	77
6.3	Experimental Results and Discussion	81
6.4	Summary	88
7.	AN ITERATIVE METHOD FOR BLIND DECONVOLUTION	89
7.1	Introduction	89
7.2	The Iterative Blind Deconvolution Method	91
7.3	Experimental Results and Discussion	95
7.4	Summary	99
8.	DISCUSSION AND CONCLUSIONS	100

REFERENCES	104
------------------	-----

LIST OF TABLES

3.1	Results of noise suppression on the synthesized, 128×128 pixel image with noise variance of 256.	36
3.2	Results of noise suppression on the Lena image for various sizes and noise levels.	39
4.1	Results of noise suppression on the synthesized, 128×128 pixel image with noise variance of 256. Approximate computer processing time for the various filters using a SUN/Sparc-2 workstation are also listed.	52
4.2	Results of noise suppression on the Lena image for various sizes and noise levels.	55
5.1	Results of fixed-neighborhood sectioned deblurring on the Lena and Camera Man images of size 128×128 pixels and 256 gray levels for various section sizes and two different blurring functions.	74
6.1	Results of fixed- and adaptive-neighborhood deblurring on the Lena and Camera Man images of size 128×128 pixels and 256 gray levels for various neighborhood sizes and two different blurring functions.	84

LIST OF FIGURES

2.1	A model of the image degradation process.	7
3.1	Noise Suppression with a synthesized image of size 128×128 pixels and a gray level range of 0 – 255. (a) original image, (b) noise corrupted image; $\sigma_n^2 = 256$, (c) ATD-LMS output image, (d) ARW-LMS output image.	34
3.2	Noise Suppression with the Lena image of size 128×128 pixels and a gray level range of 0 – 255. (a) original image, (b) noise corrupted image; $\sigma_n^2 = 256$, (c) ATD-LMS output image, (d) ARW-LMS output image.	37
4.1	Adaptive-neighborhood (AN) region growing. (a) From a seed pixel inside an object or feature, an AN is formed. The first layer of the AN defines the object (foreground) and the second layer defines its background. (b) The foreground and background regions grown from a seed pixel at location (96,17) inside the image shown in figure 3.2(a).	45
4.2	Noise Suppression with a synthesized image of size 128×128 pixels and a gray level range of 0 – 255. (a) original image, (b) noise corrupted image; $\sigma_n^2 = 256$, (c) ANNS output image, (d) ARW-LMS output image, (e) ATD-LMS output image.	51
4.3	Noise Suppression results obtained with the Lena image of size 128×128 pixels and a gray level range of 0 – 255. (a) original image, (b) noise corrupted image; $\sigma_n^2 = 256$, (c) ANNS output image, (d) ARW-LMS output image, (e) ATD-LMS output image.	54
4.4	Noise Suppression with the synthesized image in figure 4.2(b) using two passes through (a) the ANNS filter, and (b) the ARW-LMS filter. . .	57
5.1	Sectioning of the Lena image of size 128×128 pixels and a gray level range of 0 – 255. (a) original image. (b) blurred image with a Gaussian-shaped blur function; $MSE = 607$. (c), (d), and (e) show three 32×32 sections mean-padded to 128×128 pixels created for sectioned deblurring. (f) the windowed equivalent of the region in (e).	68

5.2	Sectioned deblurring results for the Lena image of size 128×128 pixels and a gray level range of 0–255. (a) deblurred image using non-overlapping sections of size 32×32 pixels; MSE = 1255. (b) Wiener-deblurred image using overlapped sections of size 32×32 pixels; MSE = 501. (c) Wiener-deblurred image using overlapped sections of size 16×16 pixels; MSE = 783. (d) Wiener-deblurred image using the full image frame; MSE = 634. The overlap in all cases is equal to half the section size in both the horizontal and vertical directions.	69
5.3	Sectioned deblurring results for the Camera Man image of size 128×128 pixels and a gray level range of 0 – 255. (a) original image. (b) 9-pixel horizontal motion blurred image; MSE = 1247. (c) a sectioned mean-padded region created for sectioned deblurring. (d) the windowed equivalent of the region in (c). (e) Wiener-deblurred image using overlapped sections of size 32×32 pixels; MSE = 424. (f) Wiener-deblurred image using overlapped sections of size 64×64 pixels; MSE = 463. (g) Wiener-deblurred image using overlapped sections of size 16×16 pixels; MSE = 539. (h) Wiener-deblurred image using the full image frame; MSE = 217.	73
6.1	Adaptive-neighborhood segmentation of the Lena image of size 128×128 pixels and gray level range of 0 – 255. (a) original image. (b) blurred image with a Gaussian-shaped blur function and noise to 35 dB SNR; MSE = 607. (c), (d), and (e) show three AN mean-padded regions created for AND. (f) the windowed equivalent of the region in (e). . .	83
6.2	Adaptive-neighborhood deblurring (AND) of the Lena image of size 128×128 pixels and gray level range of 0 – 255. (a) deblurred image using the AND filter; MSE = 292. (b) deblurred image using FNSD with non-overlapping sections of size 32×32 pixels; MSE = 1255. (c) FNSD Wiener-deblurred image using overlapped sections of size 16×16 pixels; MSE = 783. (d) FNSD Wiener-deblurred image using overlapped sections of size 32×32 pixels; MSE = 501.	85
6.3	Adaptive-neighborhood deblurring (AND) of the Camera Man image of size 128×128 pixels and a gray level range of 0 – 255. (a) Original image. (b) Image blurred by 9-pixel horizontal motion and degraded by additive Gaussian noise to 35 dB SNR; MSE = 1247. (c) An AN mean-padded region created for AND. (d) A windowed version of another stationary AN region. (e) Deblurred image using the AND filter; MSE = 181. (f) FNSD Wiener-deblurred image using overlapped sections of size 32×32 pixels; MSE = 424.	87

- 7.1 Iterative blind deconvolution with the Lena image of size 128×128 pixels. (a) original image, (b) blurred image with Gaussian-shaped blur function of radial standard deviation $\sigma_r = 3$ pixels; MSE = 606, (c) enhanced phase image, (d) initial estimate image; MSE = 877, (e) deblurred image after first iteration; MSE = 128, (f) deblurred image after 4 iterations; MSE = 110. 97
- 7.2 Iterative blind deconvolution results for the slightly blurred text image of size 64×64 pixels and 256 gray levels. (a) original image, (b) slightly blurred image with Gaussian-shaped blur function of radial standard deviation $\sigma_r = 3$ pixels; MSE = 1041, (c) enhanced phase image, (d) initial estimate image; MSE = 385, (e) final restored image after 8 iterations; MSE = 156. 98
- 7.3 Iterative blind deconvolution with severely blurred text image of size 64×64 pixels and 256 gray levels. (a) highly blurred image of figure 7.2(a) with Gaussian-shaped blur function of radial standard deviation $\sigma_r = 5$ pixels; MSE = 3275, (b) enhanced phase image, (c) initial estimate image; MSE = 690, (d) final restored image after one iteration; MSE = 618. 98

CHAPTER 1

INTRODUCTION

1.1 Motivation for Research on Image Restoration

Millions of images are created every day. Most of them are of very high quality. However, some images are of poorer quality. Of the images that are of lesser quality, a certain subset are of such importance, or are so unique, that it becomes necessary to consider techniques by which their quality may be enhanced. Image restoration represents the category of computer techniques by which the effects of image degradation phenomena may be removed and the undegraded image estimated from the observed image.

The areas of space imaging, biomedical imagery, industrial radiography, photo-reconnaissance imaging, television, infrared imaging, radar, and several multispectral or other forms of mapping scenes or objects onto a two-dimensional image format are all likely candidates for digital image processing, due to the complexity of their acquisition systems and/or the volume of the associated data.

Restoration techniques require some form of knowledge concerning the degradation phenomenon if an attempt at inversion of that phenomenon is to be made. This knowledge may come in the form of analytic models, or other a priori information coupled with the knowledge (or assumption) of the physical system that provided the

imaging process in the first place. Thus considerable emphasis must be placed on the sources of the images and their models of degradation.

Digital computer techniques in image restoration and enhancement had their first fruitful application at the Jet Propulsion Laboratory (JPL) of the California Institute of Technology. As part of the program to land a man on the moon, it was decided to land an unmanned spacecraft initially, which would televise back images of the moon's surface and test the soil for later manned landings. Unfortunately, the limitations on weight and power supply made it impossible to launch a perfect TV camera system on the unmanned craft. Consequently, JPL measured the degradation properties of the cameras before they were launched and then used computer processing to remove, to the extent possible, the degradations from the received lunar images [1]. Since that beginning, interest in digital image restoration has continued to grow. JPL's first work was conducted in the early 1960's. Today, digital image restoration is being applied in a number of areas, most of them bearing no relation at all to JPL's original work, except for the common desire to improve image quality or remove degradations from an image.

Although the number of applications of digital image restoration may be large and diverse, the problem of restoration itself is not so nebulous. Image restoration in general can be understood in terms of the specific nature of the problem, that of reducing additive noise and/or finding the most appropriate filter for deblurring a blurred image, using the terminology of functional analysis from mathematics. Furthermore, the mathematical procedures that govern image processing make it possible to describe restoration filters in specific detail. Thus, all image restoration methods, whether applied to images from medicine or aerial reconnaissance, can be discussed with a straightforward emphasis on mathematics and not upon specific applications.

1.2 Classical Image Restoration Techniques

This section presents a sampling of the most commonly known image restoration methods that have received considerable attention in the literature [1,2,3,4,5,6,7]. One of the most basic of all image restoration filters is the inverse filter. The inverse filter was implemented as a solution to the problem of deconvolving two signals. It is a linear filter whose point spread function (PSF) is the inverse of the blurring function. This filter assumes no additive noise after convolution, and thus works only when the noise level is minimal, and fails if the blur function is of the form of an ideal lowpass filter or if the blur function is zero at some frequencies. Even when the blurring frequency response does not actually go to zero, there are usually problems caused by excessive noise amplification at high frequencies. This is because the power spectrum of the blurred image is typically highest at low frequencies and rolls off significantly for higher frequencies. The spectrum of the additive noise, on the other hand, typically contains more high-frequency components, especially if the noise is white. Thus, at high frequencies, the restored image is dominated by the inverse-filtered noise, which yields useless solutions.

To overcome the noise sensitivity of the inverse filter, a number of restoration filters have been developed, which are collectively called *least-squares* filters. The Wiener filter is the most common of these least-squares filters. It is a linear space-invariant filter which makes use of power spectrum models of both the image and the noise to prevent excessive noise amplification. The frequency response of this restoration filter is chosen such that the mean-squared error (MSE) between the original undegraded image and the restored image is minimum. The Wiener filter requires a priori knowledge of the degradation function and models of the power spectra of the noise and original undegraded image. The power spectrum of the

additive noise is usually easy to estimate (it is equal to the noise variance for Gaussian white noise). To obtain an estimate of the power spectrum of the original image, an ensemble of many samples of the original image is required. Spatial averages (over prototype images) are often used in place of ensemble averages in implementation. When prototype images that closely resemble the original image are not available, as is the case in most practical situations, all the required a priori knowledge about the original image will have to be estimated from a single copy of the degraded image. Power spectral estimates so derived are far from accurate estimates of the original image. For these reasons, practical Wiener restoration filters may not be optimal. One other problem with the Wiener filter is that it is based on a (second-order) stationary image model. For most images that contain large fluctuations such as edges, the statistics will vary with pixel position and thus the image is nonstationary, or at best quasi-stationary. Image spectral estimates based on regional averages cannot follow rapid statistical changes within any given region. For such images, the Wiener filter, which uses a common estimate of the power spectrum for regions containing edges and flat regions, produces a noisy effect in flat intensity regions due to insufficient smoothing, and blurring around the edges and other information-bearing regions due to over-smoothing.

Several attempts have been made [8,9,10,11,12] to use fixed or adaptive square windows to approximate locally-stationary regions; the original pixel values are estimated by these methods inside each window using the local statistics. For the case of deblurring, more accurate power spectral estimates have been obtained by sectioning the given blurred image into square regions of some fixed size and averaging the power spectra calculated from the various regions. This procedure assumes that each region better represents a locally-stationary portion of the whole image.

Although the use of square or rectangular regions to approximate stationarity is a valid option, it still has limitations: regions cannot be reduced beyond a minimum size for accurate statistical estimation, and they should be larger than the region of support of the blurring PSF for proper deblurring. Another disadvantage of using fixed regions is the fact that each region cannot distinguish between relative pixel-to-pixel gray level differences, and thus, even small fixed regions are likely to contain flat as well as busy areas of the image, rendering the region nonstationary.

This thesis presents various image restoration techniques using regions that have the ability to adapt to pixel variations and grow into such areas of the image where relative pixel gray level differences lie within specified limits of tolerance. Thus, the new adaptive-neighborhood regions are more stationary in nature, and, as will be shown later, filters based on adaptive regions perform better than those based on fixed or adaptive *square* or *rectangular* windows. The main part of this thesis is centered around using adaptive region-based techniques for image noise suppression and deblurring. A new iterative method for blind-deconvolution which requires neither the original image nor the blur function for restoration will also be presented.

1.3 Thesis Layout

This thesis is presented in 8 chapters. Chapters 2 to 7 present theoretical aspects and results, while the last chapter gives a discussion of the work presented and some concluding remarks.

Chapter 2 surveys different image restoration techniques available in the literature, and gives a detailed technical review of these methods. Restoration of images degraded by additive white noise is first described, and then restoration of blurred noisy images is discussed. Finally, a discussion which summarizes the methods avail-

able and the relative contribution of this work is presented.

Chapter 3 discusses some well known space domain adaptive restoration methods that have been used as a comparison with our adaptive region-based techniques. Then, our space domain adaptive neighborhood noise reduction method is presented in chapter 4, with a detailed description of the adaptive neighborhood region growing algorithm used to define stationary regions. The results of applying our new filter are presented, and a comparative analysis with the methods introduced in chapter 3 is provided.

Chapter 5 presents fixed-neighborhood sectioned methods for image deblurring (deconvolution) with a detailed discussion of the methods used. Chapter 6 gives our new adaptive-neighborhood image deblurring method which is formulated in the frequency domain, and experimental results are then discussed using a comparative analysis with the fixed-neighborhood methods of chapter 5.

Chapter 7 presents the second part of the thesis which deals with a new iterative algorithm for blind deconvolution. This method is different from the other methods in the thesis that deal with image deblurring in that no knowledge of the blurring function is required, and the only information used is the blurred image. Restoration takes place using an iterative method that uses the information preserved in the phase of the blurred image to recover, as much as possible, the high-frequency components that were lost due to blurring. The final part of this chapter gives some experimental results, and a discussion of the merits of this new technique.

Chapter 8, the final chapter, gives a discussion on the methods presented in the thesis and draws conclusions based on the work completed.

CHAPTER 2

IMAGE RESTORATION TECHNIQUES

2.1 Introduction

Any image acquired by optical, electro-optical, or electronic means is likely to be degraded by the sensing environment. The degradation may be in the form of sensor noise, blur due to camera misfocus, relative object-camera motion, random atmospheric turbulence, and so on. Image restoration is concerned with filtering the observed image to minimize the effect of such degradation. This includes deblurring of images degraded by the limitations of a sensor or its environment, noise filtering, and correction of geometric distortion or nonlinearities due to sensors. Figure 2.1 shows a typical image degradation model. The imaging system, having a degradation PSF $h(k, l)$, is assumed to be linear and spatially invariant (LSI). For LSI systems, the observed image $y(k, l)$ can be expressed as

$$y(k, l) = x(k, l) * h(k, l) + n(k, l), \quad (2.1)$$

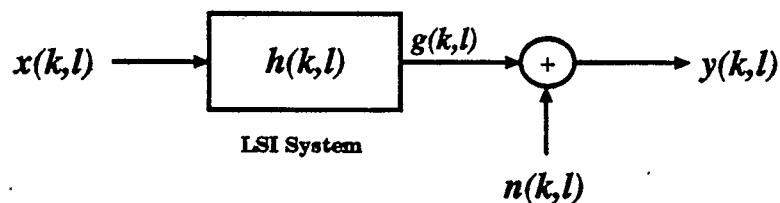


Figure 2.1 A model of the image degradation process.

where $x(k, l)$ is the original undegraded image which is unknown, and $n(k, l)$ is the additive noise function, which is assumed to be a zero-mean, white, Gaussian function of variance σ_n^2 throughout this thesis. A typical image restoration problem is to find an estimate of $x(k, l)$ given the PSF, the observed image, and some statistical properties of the noise process.

The effectiveness of image restoration filters depends on the extent and the accuracy of the knowledge of the degradation process as well as on the filter design criterion. A frequently used criterion is the MSE. Although the validity of the MSE as a global measure of visual fidelity is questionable, it is a reasonable local measure and is mathematically tractable. Other criteria such as weighted mean-squared error and maximum entropy are also used, although less frequently.

2.2 Restoration of Noisy Images

In the absence of a blurring PSF ($h(k, l) = \delta(k, l)$), the only degrading element is the additive noise $n(k, l)$, in which case the restoration problem becomes that of reducing the effects of noise as much as possible. From equation (2.1) the observed image $y(k, l)$ is now expressed as

$$y(k, l) = x(k, l) + n(k, l). \quad (2.2)$$

Many heuristic as well as mathematically optimal techniques have been proposed [1,2,4] for restoring noisy images. Noise reduction techniques can be mainly classified as *space domain* and *frequency domain* methods.

2.2.1 Space Domain Filters

In space domain filtering, restoration takes place on a pixel-by-pixel basis. The most widely used filters are those which make use of statistics (mean, variance, etc.) derived from a certain neighborhood of every pixel in the given image. Such filters are sampled below:

- *Neighborhood Averaging:* In general, image data contain superimposed broad-band noise (quantization errors, sensor noise, etc). In order to reduce these finely structured erroneous intensity fluctuations, spatial averaging operations are frequently performed on the distorted image signals [4]. Neighborhood averaging is a straightforward space domain technique for image smoothing. Given an $N \times N$ noise-degraded image $y(k, l)$, the procedure is to generate a smoothed image $\hat{x}(k, l)$ which is an estimate of the original undegraded image $x(k, l)$, and whose gray level at every point (k, l) is obtained by averaging the gray-level values of the pixels of $y(k, l)$ contained in a predefined neighborhood of (k, l) . In other words, the smoothed image is obtained by using the relation

$$\hat{x}(k, l) = \frac{1}{M} \sum_{(i,j) \in S} y(i, j), \quad (2.3)$$

for $k, l = 0, 1, \dots, N - 1$. In this equation S is the set of coordinates of points in the neighborhood of the point (k, l) , including (k, l) itself, and M is the total number of points in the neighborhood. The most commonly used neighborhoods are 3×3 windows, in which case $M = 9$ [2]. Of course, we are not limited to fixed neighborhoods in equation (2.3). Adaptive neighborhood algorithms that adjust the neighborhood size to local variations in the statistical properties of the data have been given considerable attention in the literature [8,9,10,11,13].

To explain how the neighborhood averaging filter reduces noise effects, we use equation (2.2) in equation (2.3);

$$\hat{x}(k, l) = \frac{1}{M} \sum_{(i,j) \in S} [x(i, j)] + \bar{n}(k, l), \quad (2.4)$$

where $\bar{n}(k, l)$ is the local spatial average of $n(k, l)$ inside the neighborhood. Since the noise is assumed to have zero mean, then $\bar{n}(k, l)$ has zero mean and variance $\bar{\sigma}_n^2 = \sigma_n^2/M$, that is, the noise power is reduced by a factor equal to the number of pixels in the neighborhood. If the noiseless image $x(k, l)$ is constant over the neighborhood, then spatial averaging results in an improvement in the output signal-to-noise ratio (SNR) by a factor of M . In practice, the size of the square window is limited due to the fact that $x(k, l)$ is most probably nonstationary, and therefore cannot be constant over the window. Regardless, spatial averaging introduces distortion in the form of edge blurring [3].

- *Directional Smoothing:* To protect the edges from blurring while smoothing, a directional averaging filter can be useful. Spatial averages $\hat{x}(k, l : \theta)$ are calculated in several directions θ as

$$\hat{x}(k, l : \theta) = \frac{1}{M_\theta} \sum_{(i,j) \in S_\theta} y(i, j), \quad (2.5)$$

and a direction θ^* is found such that $|y(k, l) - \hat{x}(k, l : \theta^*)|$ is minimum. Since replacing an edge pixel with the average of its neighborhood would blur the edge unless the neighborhood belongs to the same statistical ensemble as the edge pixel, this condition ensures that averaging across edges does not occur. Then

$$\hat{x}(k, l) = \hat{x}(k, l : \theta^*) \quad (2.6)$$

gives the desired result [3].

- *Median Filtering:* One of the principal difficulties of neighborhood averaging discussed above is that it blurs edges and other sharp details. An alternative approach is to use *median filters*, in which the input noisy pixel is replaced by the median of the pixels contained in a neighborhood window around the pixel, that is,

$$\hat{x}(k, l) = \text{median}\{y(i, j), (i, j) \in S\}, \quad (2.7)$$

where S is a suitably chosen window. The algorithm for median filtering requires arranging the pixel values in the window in increasing or decreasing order and picking the middle value. Generally, the window size is chosen so that the number of pixels M inside the window is odd. If M is even, then the median is taken as the average of the two values in the middle. Typical window sizes are 3×3 , 5×5 , or 7×7 [2,3].

The median filter has the following properties:

1. It is a nonlinear filter.
 2. It is useful for removing isolated lines or pixels while preserving spatial resolution. It has been shown [2,3,4] that the median filter performs very well on images containing binary noise, but performs poorly when the noise is Gaussian.
 3. Its performance is poor when the number of noise pixels in the window is greater than half the number of pixels in the window.
- *Averaging of Multiple Images:* If we had an ensemble of noisy images $\{y_i(k, l), i = 1, 2, \dots, M\}$ such that

$$y_i(k, l) = x_i(k, l) + n_i(k, l), \quad (2.8)$$

where $x_i(k, l) \simeq x(k, l)$ for all i , and $n_i(k, l)$ satisfies the same assumptions as those stated for $n(k, l)$ above (i.e., $n_i(k, l)$ are zero-mean, with variances $\sigma_{n_i}^2 = \sigma_n^2$ for all i), then the objective of this procedure is to obtain a noise-free result by taking the ensemble average of the set of noisy images. Therefore, for a set of M different noisy images

$$\hat{x}(k, l) = \frac{1}{M} \sum_{i=1}^M y_i(k, l), \quad (2.9)$$

and we have

$$\bar{\sigma}_n^2 = \frac{1}{M} \sigma_n^2, \quad (2.10)$$

where $\bar{\sigma}_n^2$ is the variance (power) of the noise in the restored image $\hat{x}(k, l)$, which has been reduced by a factor of M . This indicates that as M increases, the noise power in the restored image decreases, and $\hat{x}(k, l)$ approaches the original image $x(k, l)$. In practice, the principal difficulty in using this method lies in being able to register the images so that corresponding pixels line up correctly [2].

2.2.2 Frequency Domain Filters

In frequency domain filtering, restoration takes place on the spectrum of the image. Several frequency domain methods for noise suppression have been considered in the literature [1,2,3,4,8,10,12], and are sampled below:

- *Lowpass Filtering:* Noise and other sharp transitions in the gray levels of an image contribute heavily to the high-frequency content of its Fourier transform. Lowpass filtering is sometimes used to smooth out the noise and other unwanted gradients. Taking the Fourier transform of equation (2.2), we have

$$Y(u, v) = X(u, v) + N(u, v), \quad (2.11)$$

where upper case letters denote the frequency domain equivalents of the space domain functions named with the corresponding lower case letters. Since the noise $n(k, l)$ is white (uncorrelated), has a zero mean, and has a variance of σ_n^2 , its power spectral density $P_n(u, v)$ is constant over all frequencies of interest and equal to σ_n^2 . From equation (2.11), the additive noise $N(u, v)$ would thus result in raising the high-frequency components of the original image $X(u, v)$. Lowpass filtering reduces the noise level in the degraded image $Y(u, v)$ by attenuating a specified range of high-frequency components in the transform of the degraded image. For a specified lowpass filter transfer function $H_{lpf}(u, v)$ we obtain the noise-reduced image as

$$\hat{X}(u, v) = H_{lpf}(u, v) \cdot Y(u, v). \quad (2.12)$$

The inverse Fourier transform of $\hat{X}(u, v)$ will yield the desired smoothed image $\hat{x}(k, l)$. Clearly, this method will not produce the best results, because of the fact that a lowpass filter will not discriminate between high-frequency noise and important high-frequency image information such as edges. Also, the lowpass filter chosen will not make use of the information present in the noisy image, such as the noise power spectral density, which is easily determined from its variance. Thus, we would either have insufficient noise suppression due to a wide filter bandwidth, or too much smoothing and edge blurring due to a narrow filter bandwidth.

- *Wiener Smoothing Filter*: The Wiener filter is derived by minimizing the MSE between the original undegraded image $x(k, l)$ and the estimated image $\hat{x}(k, l)$ [1,2,3,4,5]. This would yield a filter having the transfer function

$$H_w(u, v) = \frac{P_x(u, v)}{P_x(u, v) + P_n(u, v)}, \quad (2.13)$$

where $P_x(u, v)$ and $P_n(u, v)$ are the power spectra of the original image and noise respectively. It is clear that the Wiener smoothing filter $H_w(u, v)$ is a zero-phase filter that depends only on the SNR $S_{nr} = \frac{P_x(u, v)}{P_n(u, v)}$. For frequencies where $S_{nr} \gg 1$ (i.e., high signal power), $H_w(u, v)$ becomes nearly equal to unity, which means that all these frequency components are in the passband of the filter. When $S_{nr} \ll 1$, $H_w(u, v) = S_{nr}$; that is, all frequency components where $S_{nr} \ll 1$ are attenuated in proportion to their SNR. For images, S_{nr} is usually high at low spatial frequencies (flat gray level regions). Therefore, the Wiener smoothing filter is one form of a lowpass filter. This fact attributes to the oversmoothing problem of the Wiener filter, where the edges suffer from too much smoothing due to the lowpass action of the filter. In return, the Wiener filter removes more noise than the power-spectrum equalization filter, described below.

- *Power-Spectrum Equalization (PSE) Filter*: The PSE filter transfer function is derived on the basis of a simple constraint, much less stringent than the minimum-MSE constraint of the Wiener filter. The filter is derived by seeking a linear estimate, $\hat{x}(k, l)$, such that the power spectrum of the estimate is equal to the power spectrum of the original image. The resulting filter response is [5]

$$H_{PSE}(u, v) = \left[\frac{P_x(u, v)}{P_x(u, v) + P_n(u, v)} \right]^{1/2}, \quad (2.14)$$

and the restored image is

$$\hat{X}(u, v) = H_{PSE}(u, v) \cdot Y(u, v), \quad (2.15)$$

which gives the frequency-domain value of the noise-reduced image. Taking the inverse Fourier transform of $\hat{X}(u, v)$ would generate the actual space domain restored image. Like the Wiener filter, the PSE filter is a zero-phase filter that

also depends only on the SNR $S_{nr} = \frac{P_s(u,v)}{P_n(u,v)}$. For frequencies where $S_{nr} \gg 1$ (i.e., high signal power), $H_{PSE}(u, v)$ becomes nearly equal to unity, which means that all these frequency components are in the passband of the filter. When $S_{nr} \ll 1$, $H_{PSE}(u, v) = \sqrt{S_{nr}}$; that is, all frequency components where $S_{nr} \ll 1$, are attenuated in proportion to the square root of their SNR. However, the PSE filter differs from the Wiener filter in that in between these two extremes, the filter gain is greater than $H_w(u, v)$ and less than unity. This is because of the square-root operation. This leads to more high frequencies passing through the PSE filter, which would have otherwise been cut off by the Wiener filter; thus images restored using PSE filtering tend to be visually sharper because of less edge blurring, at the expense of allowing more high-frequency noise to pass through the filter. Usually, this is not an issue of concern due to the fact that the PSE-filtered images are visually pleasing.

2.3 Restoration of Blurred Images

The objective of any system that forms an image is to obtain the best image possible, one that is sharp, clear, and free from degradations. This is not always possible. First, every real image formation system has inherent limitations; the impulse response of a real system is of finite width and causes an inevitable loss of resolution in the image. If important details of size comparable to the width of the impulse response are sought in the image, the loss of resolution becomes a matter of concern. The process of attempting to correct for degradations due to blurring is the problem of image deblurring.

In the frequency domain, equation (2.1) becomes

$$Y(u, v) = X(u, v) \cdot H(u, v) + N(u, v). \quad (2.16)$$

Several frequency domain methods for image deblurring have been considered in the literature [1,2,3,4,8,10,12], and are sampled below:

- *Inverse Filter*: Inverse filtering is the process of recovering the input of a system from its output. For example, in the absence of noise (from Fig. 2.1) the inverse filter would be a system that recovers $x(k, l)$ from the observation $y(k, l)$. This requires that the inverse filter transfer function be the reciprocal of the blur modulation transfer function (MTF), that is

$$H_{inv}(u, v) = \frac{1}{H(u, v)}, \quad (2.17)$$

and the deblurred image becomes (in the presence of noise)

$$\hat{X}(u, v) = H_{inv}(u, v) \cdot Y(u, v) = X(u, v) + \frac{N(u, v)}{H(u, v)}. \quad (2.18)$$

Inverse filters are useful for precorrecting an input signal in anticipation of the degradations caused by the system, such as correcting the nonlinearity of a display [3]. Being the simplest approach, this is also the one fraught with the most difficulties:

1. Many blur PSFs are such that their Fourier transforms have zeros. For example, the Fourier transform of a one-dimensional motion blur along the horizontal direction is given by

$$H(u, v) = \frac{\sin(\pi au)}{\pi au}, \quad (2.19)$$

where a is the blur distance in pixels, and u is the horizontal frequency variable. If the blur is severe enough (a is large enough) so that zeros of the sinc function lie within the Nyquist frequency, the inverse filter would be singular. A similar problem is encountered with out-of-focus blur, which

convolves the aperture shape with the image. For most regular aperture shapes (circle, square, etc.), the corresponding Fourier transform has zeros, and the zeros will result in singularities if they lie within the Nyquist frequency. Unfortunately, this is usually the case.

2. Even if the PSF is not singular, it is usually *ill conditioned*. That is, the magnitude of the MTF goes to zero so rapidly for some values of u and v (typically for high-frequency indexes, since blurring implies lowpass behavior) that the noise term $\frac{N(u,v)}{H(u,v)}$ in equation (2.18) is drastically magnified, obliterating the restored image.
- *Wiener Filter*: The sensitivity of the inverse filter to noise and its singularity is avoided by another deblurring technique, the Wiener filter. As discussed in section 2.2.2, the Wiener filter is derived by minimizing the MSE between the original undegraded image $x(k,l)$ and the estimated image $\hat{x}(k,l)$ [1,2,3,4,5]. This would yield a filter having the transfer function

$$H_w(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + \frac{P_n(u,v)}{P_x(u,v)}}, \quad (2.20)$$

where the asterisk superscript indicates complex conjugation. The deblurred image is thus given by

$$\hat{X}(u,v) = H_w(u,v) \cdot Y(u,v). \quad (2.21)$$

It is possible to observe the following properties of the Wiener filter from a quick inspection of equation (2.20):

1. If the noise is very small or zero, so that $P_n(u,v) \rightarrow 0$, the Wiener filter reduces to the inverse filter. Thus, in low-noise frequency regions (typically

the low-frequency regions of the image), the Wiener filter has inverse filter behavior.

2. If the signal power becomes small, so that $P_x(u, v) \rightarrow 0$, the Wiener filter has a very low gain. This solves the problem of singular behavior in the PSF, and also controls ill-conditioned behavior even in the absence of singularities [3,5].

In general, the Wiener filter reduces the visual noise associated with the inverse filter, but usually at the expense of some detail and sharpness. This can be explained in two ways:

1. The minimum MSE constraint is a very powerful requirement and could be relaxed (may not always be minimum);
 2. The nonlinear and adaptive properties of the human visual system may not be “matched” to the minimum MSE criterion.
- *Power-Spectrum Equalization and Blind Deblurring:* As discussed in section 2.2.2, the PSE filter transfer function is derived by seeking a linear estimate, $\hat{x}(k, l)$, such that the power spectrum of the estimate is equal to the power spectrum of the original image. The resulting filter response is

$$H_{PSE}(u, v) = \left[\frac{1}{|H(u, v)|^2 + \frac{P_n(u, v)}{P_x(u, v)}} \right]^{1/2}, \quad (2.22)$$

and

$$\hat{X}(u, v) = H_{PSE}(u, v) \cdot Y(u, v) \quad (2.23)$$

gives the frequency domain values of the deblurred image, which are inverse Fourier transformed to generate the actual space domain restored image.

The following properties of the PSE filter are to be noted:

1. For low noise, $P_n(u, v) \rightarrow 0$, the filter reduces to the magnitude response of the inverse filter.
2. For low signal, $P_x(u, v) \rightarrow 0$, the filter gain goes to zero.
3. In between these two extremes, the filter gain is greater than the Wiener filter gain $H_w(u, v)$ and less than the inverse filter gain $H_{inv}(u, v)$. This is because of the absence of the $H^*(u, v)$ term seen in the numerator of the Wiener filter equation (2.20) and the square root operation. It is, in fact, possible to show that the PSE filter is the geometric mean between the inverse filter and the Wiener filter [5].

Since the PSE filter has greater gain than the Wiener filter, but without the ill-conditioned behavior of the inverse filter, the result is a filter that admits into the deblurred image more of the detailed information associated with high frequency, such as spatial edges, which is where the Wiener filter usually cuts off. There is also an increase in visual noise, but the human eye is usually willing to accept the increased noise in return for the additional fine structure that is meaningful [5].

Although the transfer function in equation (2.22) is a magnitude-only response, for many known image degradations (e.g., motion blur and out-of-focus blur), the phase response of the blur is zero [5,14], and consequently the phase of the blurred image is very similar to that of the original undegraded image. In this case, only magnitude deblurring is required, and using equation (2.22) is sufficient for deblurring.

The importance of the PSE method is manifested in the case where the blurring PSF is unknown. In this case, the blurred image itself could be used to

estimate the parameters necessary to carry out the deblurring operation. The Wiener filter requires that values of $P_n(u, v)$, $P_x(u, v)$, and $H(u, v)$ be known a priori, and a similar observation might be made about the PSE filter. Explicit concentration upon power spectrum relations in equation (2.22) leads to the realization that the relevant information can be estimated from the blurred image $y(k, l)$. Consider dividing $y(k, l)$ into segments of size $P \times P$, where P is large compared to the region of support of the blur PSF, but small compared to the actual image dimensions $M \times M$; $P = 32$ is typical for $M = 128$. The sections need not be overlapping. Neglecting edge effects, each section of the image can be expressed as the convolution of the PSF with an equivalent section from the original undegraded image $x(k, l)$. Thus, from equation (2.1),

$$y_i(k, l) \simeq x_i(k, l) * h(k, l) + n_i(k, l) \quad (2.24)$$

is the approximate description of the image formation law for each section $y_i(k, l)$. The power spectrum of each $y_i(k, l)$ can be computed as

$$P_y^i(u, v) \simeq |H(u, v)|^2 \cdot P_x^i(u, v) + P_n^i(u, v), \quad (2.25)$$

where the superscripts denote the i^{th} section. Assuming that the image and noise can be approximated by stationary random processes, $P_x^i(u, v)$ and $P_n^i(u, v)$ are samples from the same power spectral functions of the original and noise processes respectively. Summing over a number of sections will, therefore, average out the statistical fluctuations in these power spectra that would be seen in any one section [3,5,6,12]. We thus have

$$\hat{P}_y(u, v) = |H(u, v)|^2 \cdot \hat{P}_x(u, v) + \hat{P}_n(u, v), \quad (2.26)$$

where the hats denote the averaged power spectra over all the sections. This quantity can be evaluated from the blurred image. Equation (2.22) may then

be rewritten as

$$\begin{aligned}
 H_{PSE}(u, v) &= \left[\frac{P_x(u, v)}{|H(u, v)|^2 \cdot P_x(u, v) + P_n(u, v)} \right]^{1/2} \\
 &= \left[\frac{P_x(u, v)}{\hat{P}_y(u, v)} \right]^{1/2} \quad (2.27)
 \end{aligned}$$

The quantity in equation (2.26) gives the denominator in equation (2.27) for the PSE filter. Stockham *et al.* [15] argued that $P_x(u, v)$ for the numerator in equation (2.27) could be approximated by an average power spectrum evaluated over a wide variety of images. Since this filter estimates most of the required information from the given blurred image itself, it is called a *blind* deconvolution filter.

2.4 Discussion

In this chapter, we discussed various image restoration techniques and their mathematical formulas. First we started by giving an account of the most commonly used methods for restoring images degraded by additive white Gaussian noise. We divided noise reduction methods into space domain and frequency domain methods. Four space domain methods were discussed. These methods mainly smoothed out the noise in the degraded image, such as in neighborhood averaging, where square windows are used to approximate stationary regions and obtain estimates of the signal and noise statistics. One of the problems faced by most of the noise reducing techniques described above is the absence of information about the noise. The most common solution to this problem is to estimate the variance of a uniform section in the noisy image $y(k, l)$ that contains only noise. This variance would give a good approximation of the noise variance σ_n^2 in the image. We then discussed three very common frequency domain noise reduction filters, the most popular of which are the

Wiener smoothing filter and the PSE filter. These two filters make use of the power spectra of the original image and noise to control smoothing so that the problem of oversmoothing of information-bearing regions, encountered in the simple lowpass smoothing filter, is reduced. It was shown that the frequency response of the PSE filter has a gain that is higher than that of the Wiener filter, thus allowing for the passage of more information at the expense of allowing high-frequency noise as well.

We next discussed three popular methods of image deblurring for restoring images degraded by both convolutional blur and additive noise. It was explained that the inverse filter, besides its simplicity, has many problems regarding singularity and ill-conditioned behavior at high frequencies. The Wiener filter and PSE filter solve the problems encountered in the inverse filter by cutting off at the singularities of the inverse filter. It was again explained that the PSE filter produces visually pleasing restorations in spite of the increased noise level. Finally, the concept of blind deconvolution was introduced, where it was shown how blurred images could be restored without the prior knowledge necessary for the other filters.

From this discussion, it is clear that nearly all restoration techniques are based on the assumption that images are statistically stationary. For most images this is not true. Images are at best locally stationary, and then only if the windows or neighborhoods used contain pixels that belong to the same statistical ensemble. For most fixed neighborhoods stationarity is a very crude approximation, which results in poor statistical estimates and hence poor restoration. In the next few chapters of the thesis, we will present some adaptive-neighborhood techniques, both old and original, that attempt to create neighborhoods that are better approximations to stationary regions. Results from the application of these methods to images are then evaluated.

CHAPTER 3

ADAPTIVE RESTORATION METHODS

3.1 Introduction

Many heuristic as well as mathematically optimal techniques have been proposed [8-13,16-21] for restoring noisy images. Earlier approaches to image restoration mostly used linear filters based on the assumption of a stationary image model. Such an assumption leads to space-invariant filters which smooth out edges because of the unavoidable compromise between noise and resolution. Many adaptive restoration systems have been proposed recently to overcome this problem. Although adaptive systems are computationally more expensive in both design and implementation, they perform significantly better than nonadaptive methods. Some adaptive systems partition the image into regions or subimages in which different stationary models are assumed [16,17,18]. Others assume a simple image model and use a moving two-dimensional window to continuously estimate the model parameters and adjust a nonlinear two-dimensional filter [19,20].

In this chapter we present two different examples of adaptive Wiener filtering for noise reduction which have been recently reported in the literature [8-10,19-21]. Some experimental results are then shown with a comparative discussion regarding filter performance. This chapter serves as the background for the development of our original adaptive-neighborhood filter, which will be introduced in chapter 4.

3.2 The Adaptive Two-dimensional LMS Algorithm

Noise reduction is usually accomplished by a filtering procedure optimized with respect to some error measure. The most widely used error measure is the MSE. The Wiener filter is a classical solution to this problem. Unfortunately, the Wiener filter is designed under the assumption of stationary signal and noise models. While for additive white noise the noise model can very well be assumed to be stationary, it is unlikely to be true for the signal model, especially for signals that contain large fluctuations such as edges. The Wiener filter has been used for image restoration with limited success because its lowpass characteristics give rise to unacceptable blurring of lines and edges in the image. Recent methods of circumventing this problem have taken into account the nonstationarity of the given image. One example is the method developed by Chan and Lim [11] which takes into account the image nonstationarity and varies the filter parameters according to changes in image characteristics/statistics. Another example is the adaptive two-dimensional least-mean-squares (ATD-LMS) algorithm developed by Hadhoud and Thomas [19], which will be described here. Again we use the same noise-corrupted image model with no blurring as in the previous chapter, that is

$$y(k, l) = x(k, l) + n(k, l), \quad (3.1)$$

where

$x(k, l)$ is the original, noise-free image,

$n(k, l)$ is the additive white Gaussian noise, uncorrelated to the original image $x(k, l)$,

and

$y(k, l)$ is the noise-corrupted input image.

The ATD-LMS method is an efficient and practical algorithm for noise reduction.

It is an example of a fixed-window Wiener filter in which the filter coefficient values change depending on the image characteristics. The algorithm is based on the method of steepest descent, and can track the variations in the local statistics of the given image, thus adapting to different image features. The advantage of this algorithm is that it does not require any a priori information about the image, the noise statistics, or their correlation properties. Also, it does not require any averaging, differentiation, or matrix operations.

3.2.1 Derivation of the ATD-LMS Filter

The ATD-LMS algorithm is derived by defining a causal finite impulse response (FIR) restoration filter $W_p(i, j)$ whose region of support is $N \times N$ (N is typically 3) such that

$$\hat{x}(k, l) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} W_p(i, j) \cdot y(k-i, l-j), \quad (3.2)$$

where

$\hat{x}(k, l)$ is the estimate of the original pixel value $x(k, l)$,

$y(k, l)$ is the noise-corrupted input image, and

p marks the current position of the filter in the image, which is given by

$$p = k \cdot M + l \quad (3.3)$$

for the pixel position (k, l) in an $M \times M$ image, and will take values from 0 to $M^2 - 1$.

The filter coefficients $W_{p+1}(i, j)$ are determined at the next pixel position, $p+1$, by minimizing the MSE between the desired pixel value $x(k, l)$ and the estimated pixel value $\hat{x}(k, l)$ at the present pixel location, p , using the method of steepest descent. In this method, the next filter coefficients $W_{p+1}(i, j)$ are equal to the present coefficients plus a change proportional to the negative gradient of the error power (MSE). This

is expressed as

$$W_{p+1}(i, j) = W_p(i, j) - \mu \nabla_p [e_p^2], \quad (3.4)$$

where

μ is a scalar multiplier controlling the rate of convergence and filter stability,

e_p is the error signal, which is the difference between the desired signal $x(k, l)$ and the estimate $\hat{x}(k, l)$, and

$\nabla_p[\cdot]$ is a gradient operator applied to the error power, e_p^2 , at p .

Since the original image $x(k, l)$ is unknown, and the only image in hand is the noise-corrupted image $y(k, l)$, an approximation to the original image is used, and the error becomes

$$e_p = d(k, l) - \hat{x}(k, l), \quad (3.5)$$

where $d(k, l)$ is the approximation to $x(k, l)$. The technique used by Hadhoud and Thomas [19] to obtain $d(k, l)$ was to generate $d(k, l)$ from the input image $y(k, l)$ by decorrelation. The decorrelation operator used by Hadhoud and Thomas [19] and used here is a two-dimensional delay operator of $(1, 1)$ samples. This would allow the correlation between $d(k, l)$ and $y(k, l)$ to be similar to the correlation between $x(k, l)$ and $y(k, l)$, and, in turn, would make $d(k, l)$ correlated to $x(k, l)$ to some extent. Evaluation of equation (3.4) using equation (3.2) and equation (3.5) gives

$$W_{p+1}(i, j) = W_p(i, j) + 2\mu \cdot e_p \cdot y(k - i, l - j), \quad (3.6)$$

which is a recursive equation defining the filter coefficients at any pixel position p .

3.2.2 Implementation of the ATD-LMS Filter

Equation (3.2) and equation (3.6) give the ATD-LMS restoration filter and the filter weight updating algorithm respectively. The algorithm convergence does not

depend on the initial conditions; it will converge for any arbitrary initial value, and hence, provides good nonstationary performance. In comparing the ATD-LMS filter with the non-adaptive LMS algorithm, the second term of equation (3.6) would not be included in the latter, and the filter coefficients would not change from pixel to pixel under the assumption that the image is stationary. This would put a constraint on the initial coefficient values since these would be the values used for the whole image, and thus, different initial values would result in different filtered outputs. Although the initial conditions do not affect the convergence of the ATD-LMS filter, the design choice of the convergence factor μ depends on the particular application, and involves a tradeoff between convergence speed, tracking ability, and steady-state MSE.

In implementing equation (3.2) and equation (3.6), the initial weights of the filter, $W_0(i, j)$, were estimated by processing 10 lines of the given image starting with zero weights. The weights obtained after processing these 10 lines were then used as initial conditions, and processing started again at the beginning of the image. The convergence factor, μ , was determined by trial and error for different images as suggested by Hadhoud and Thomas [19], and the region of support used for the filter was 3×3 .

3.3 The Adaptive Rectangular Window LMS Algorithm

As discussed earlier, the general Wiener estimate makes use of the image power spectrum which is estimated over a combination of flat and rapidly changing regions of the image. When the Wiener filter is implemented in the space domain, the variance and mean of the data are used. These are calculated over the entire image, and the same mean and variance are used for the restoration operation. However, not all pixels in the image belong to the same statistical ensemble. Different ensembles have different means and variances. Thus there is a certain amount of error in the

evaluation of these quantities, leading to restored images that are noisy in flat regions due to insufficient smoothing, and fuzzy around edge regions due to over-smoothing [8-13].

3.3.1 Derivation of the ARW-LMS Filter

To overcome the problems due to image nonstationarity, Wiener filtering using an adaptive-size rectangular window (ARW) to estimate the filter coefficients was recently proposed by Song and Pearlman [8,9,21] and refined by Mahesh *et al.* [10]. In their method, the same noisy image model as that in equation (3.1) was used, where the noise $n(k, l)$ is of zero mean and variance σ_n^2 , and is uncorrelated to the original image $x(k, l)$, which is assumed to be of zero mean and variance σ_x^2 . The estimate is of the form

$$\hat{x}(k, l) = \alpha \cdot y(k, l), \quad (3.7)$$

and the problem reduces to that of finding the constant α at each pixel location using the minimum MSE criterion of the standard Wiener filter. The error is given by

$$e = x(k, l) - \hat{x}(k, l) = x(k, l) - \alpha \cdot y(k, l). \quad (3.8)$$

Minimization of the MSE requires the error signal e to be orthogonal to the data $y(k, l)$ [1,2,3,10]. That is

$$E\{[x(k, l) - \alpha \cdot y(k, l)] \cdot y(k, l)\} = 0, \quad (3.9)$$

where $E\{\cdot\}$ is the expectation operator. Solving for α we obtain

$$\alpha = \frac{\sigma_x^2}{\sigma_x^2 + \sigma_n^2}, \quad (3.10)$$

which is the LMS estimator that reduces noise from $y(k, l)$. If the original image is not of zero mean, equation (3.7) can still be used by first subtracting the mean from

both the images $x(k, l)$ and $y(k, l)$. Since the noise is of zero mean for all pixels, the a posteriori mean m_y of the image $y(k, l)$ at pixel position (k, l) is equal to the a priori mean m_x of the original image $x(k, l)$, and the estimate of equation (3.7) thus becomes

$$\hat{x}(k, l) = m_y + \frac{\sigma_x^2}{\sigma_x^2 + \sigma_n^2} \cdot [y(k, l) - m_y]. \quad (3.11)$$

The estimate $\hat{x}(k, l)$ given by equation (3.11) is considered to be the best estimate of the nonstationary image $x(k, l)$, if we are able to obtain accurate values for the estimator parameters m_y and σ_x^2 .

3.3.2 Implementation of the ARW-LMS Filter

In implementing equation (3.11), it is important to make the assumption that the signal values in the immediate neighborhood of a data point (k, l) are the samples from the same ensemble as $x(k, l)$. That is, globally nonstationary sequences can be considered to be locally stationary and ergodic over a small region. Thus, if we can accurately determine the size of an interval in which the signal values have the same statistical parameters, the sample statistics can approximate the a posteriori parameters needed for the estimate in equation (3.11).

The view taken by Song and Pearlman [8,9,21] was to identify the size of a stationary square region for each pixel in the image and to calculate the local statistics of the image within that region. The size of the window changes according to a measure of signal activity; an effective algorithm was proposed for the window size that improved the performance of various point estimators. The manifestation of the improved performance was greater smoothing in the flat intensity regions and less smoothing across edges. This results in a sharp restoration, and the lack of noise

smoothing across edges is masked by the human visual system.

In deriving the local (sample) statistics, \hat{m}_y and $\hat{\sigma}_x^2$, Mahesh *et al.* [10] make use of adaptive rectangular windows of length L_r in the row direction and L_c in the column direction. Since the window is required to be centered about the pixel being evaluated, the ARW lengths will be odd. Except near the borders of the image, the ARW dimensions can be given by

$$L_r = 2N_r + 1, \quad (3.12)$$

$$\text{and } L_c = 2N_c + 1, \quad (3.13)$$

where N_r and N_c are the dimensions of the one-sided neighborhood. Within this window, the local mean and variance are calculated as

$$\hat{m}_y = \frac{1}{L_r L_c} \sum_{i=k-N_r}^{k+N_r} \sum_{j=l-N_c}^{l+N_c} y(i, j), \quad (3.14)$$

and

$$\hat{\sigma}_y^2 = \frac{1}{L_r L_c} \sum_{i=k-N_r}^{k+N_r} \sum_{j=l-N_c}^{l+N_c} [y(i, j) - \hat{m}_y]^2, \quad (3.15)$$

where (k, l) is the current pixel location.

The local variance of the original signal $\hat{\sigma}_x^2$ is thus given by

$$\hat{\sigma}_x^2 = \begin{cases} \hat{\sigma}_y^2 - \sigma_n^2 & \text{if } \hat{\sigma}_y^2 > \sigma_n^2 \\ 0 & \text{otherwise.} \end{cases} \quad (3.16)$$

Using the local statistics, \hat{m}_y and $\hat{\sigma}_x^2$, in place of the ensemble statistics, m_y and σ_x^2 , equation (3.11) becomes

$$\hat{x}(k, l) = \hat{m}_y + \frac{\hat{\sigma}_x^2}{\hat{\sigma}_x^2 + \sigma_n^2} \cdot [y(k, l) - \hat{m}_y]. \quad (3.17)$$

It should be noted that the parameters L_r , L_c , N_r , N_c , \hat{m}_y , $\hat{\sigma}_y^2$, and $\hat{\sigma}_x^2$, and the other parameters that follow depend on the point (k, l) , and should in fact be denoted by

$L_r(k, l)$ and so on. This has been suppressed for convenience of notation, but should not be forgotten. It should also be noted that although the noise variance, σ_n^2 , is not known a priori, it is easily estimated from a window in a flat area of the degraded image [11].

As mentioned earlier, if we are to be accurate in determining m_y and σ_x^2 , the pixels in the ARW have to belong to the same ensemble as the central pixel. If relatively large windows are used, the windows may cross over boundaries of different regions and include pixels from other ensembles when the sample statistics are generated. In such a case, some smoothing will occur across edges within the windows. On the other hand, if the windows are too small, then the lack of samples would result in poor estimates of the mean and variance, and consequently insufficient noise suppression would occur over flat regions. Thus we should use small windows where the image intensity changes rapidly, and large windows where the image contrast is relatively flat.

The ARW lengths, L_r and L_c , are varied depending on a signal activity parameter defined by Mahesh *et al.* [10] and Song and Pearlman [21] as

$$S_r = \frac{1}{L_r L_c} \sum_{i=k-N_r}^{k+N_r} \sum_{j=l-N_c}^{l+N_c} [y(i, j) - \hat{m}_r]^2 - \sigma_n^2, \quad (3.18)$$

where \hat{m}_r is the mean evaluated in the row direction as

$$\hat{m}_r = \frac{1}{L_r} \sum_{i=k-N_r}^{k+N_r} y(i, l), \quad (3.19)$$

and S_r is a measure of local roughness of the signal in the row direction, being equal to the variance of the original signal in the same direction. Analogously, the expression for the signal activity parameter in the column direction is

$$S_c = \frac{1}{L_r L_c} \sum_{i=k-N_r}^{k+N_r} \sum_{j=l-N_c}^{l+N_c} [y(i, j) - \hat{m}_c]^2 - \sigma_n^2, \quad (3.20)$$

where \hat{m}_c is the mean evaluated in the column direction as

$$\hat{m}_c = \frac{1}{L_c} \sum_{j=l-N_c}^{l+N_c} y(k, j). \quad (3.21)$$

If the signal activity parameter S_r in the row direction is large, indicating the presence of an edge or other information, then the window size in the row direction, N_r , is decremented so that points from other ensembles are not included. If S_r is small, indicating that the current pixel lies in a flat contrast region, N_r is incremented so that a better estimate of the mean and variance will be obtained. In order to make this decision, the signal activity parameter in the row direction is compared with a threshold T_r , and that in the column direction with T_c . The rules are therefore

$$N_r \leftarrow N_r - 1, \text{ if } S_r \geq T_r, \quad (3.22)$$

$$\text{or } N_r \leftarrow N_r + 1, \text{ if } S_r < T_r, \quad (3.23)$$

and

$$N_c \leftarrow N_c - 1, \text{ if } S_c \geq T_c, \quad (3.24)$$

$$\text{or } N_c \leftarrow N_c + 1, \text{ if } S_c < T_c. \quad (3.25)$$

User-specified minimum and maximum values for N_r and N_c are used such that if either N_r or N_c goes beyond these limits, it will be assigned the value of the corresponding limit. The threshold is defined by [8,10,21]

$$T_r = \frac{\eta \sigma_n^2}{L_r}, \quad (3.26)$$

$$\text{and } T_c = \frac{\eta \sigma_n^2}{L_c}, \quad (3.27)$$

where η is a weighting constant that affects the rate at which the window size changes. The threshold is point-dependent and varies directly as the noise variance and inversely as the ARW dimension. Thus if the noise variance is high, the threshold will

be high, and the window length is more likely to be incremented than decremented. This will lead to a large window and effective smoothing of the noise. If the window size is large, the threshold will be small. Thus the window size is likely to be decremented for the next pixel. This helps the ARW length to converge to a certain range.

3.4 Experimental Results and Discussion

In this section some experimental results which illustrate the performance of the preceding algorithms are presented. The results were evaluated by direct visual examination and by calculation of the MSE between the original and restored images.

The ATD-LMS and the ARW-LMS algorithms were first applied to a synthesized image after it was corrupted with additive white Gaussian noise. Figure 3.1(a) shows the original image of size 128×128 pixels and having gray levels between 0 and 255. It is composed of a uniform background of gray level 60, and circles, squares and triangles of various gray levels placed at randomly selected positions. The original image was corrupted with additive white Gaussian noise of variance 256, as shown in figure 3.1(b). This synthesized image is very useful for observing the effects of the algorithms as it contains almost ideal edges.

The ATD-LMS-restored image is shown in figure 3.1(c). The ATD-LMS algorithm was applied to the corrupted image with the convergence factor μ set to 0.4×10^{-7} . This value was arrived at after testing with values ranging between 1.0×10^{-9} and 0.7×10^{-7} . The final value used is close to the value of 0.35×10^{-7} chosen by Hadhoud and Thomas [19] in their original paper for a similar type of image. The ATD-LMS algorithm applies a gradually changing filter which tends to suppress noise in a relatively uniform way over the image. Although this typically results in lower values

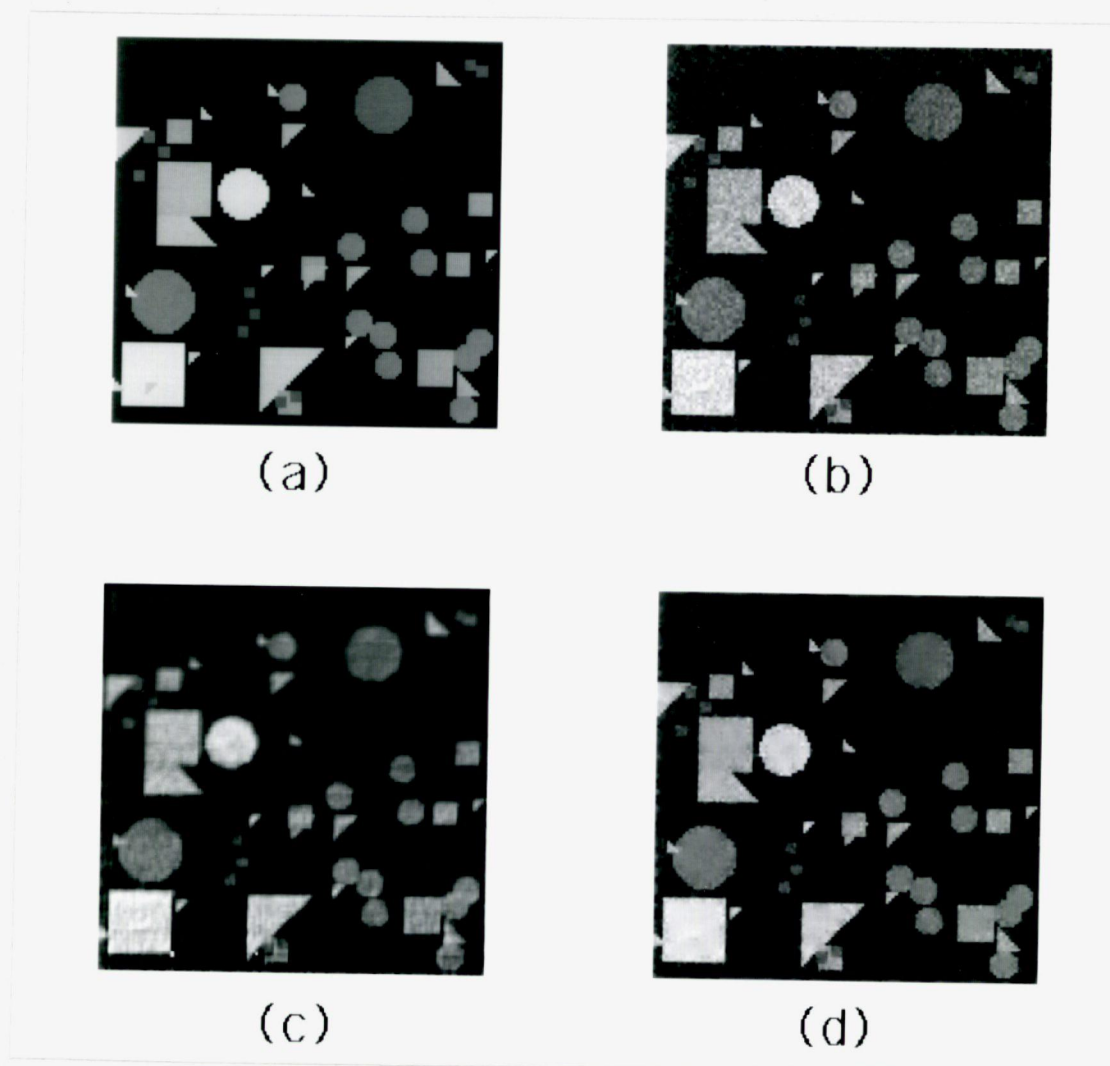


Figure 3.1 Noise Suppression with a synthesized image of size 128×128 pixels and a gray level range of $0 - 255$. (a) original image, (b) noise corrupted image; $\sigma_n^2 = 256$, (c) ATD-LMS output image, (d) ARW-LMS output image.

of the MSE, it also tends to smooth or blur edges and other structured features in the image, and to leave excessive noise in uniform regions of the image. One explanation to this could be the fact that the adaptive weights of the filter, $W_p(i, j)$, depend on the model of the original image which is approximated in the above algorithm by the decorrelated image $d(k, l)$. Since $d(k, l)$ is far from an accurate approximation of the original image $x(k, l)$, the updated filter weights are far from optimal, and thus the algorithm does not give optimum MSE values. The ATD-LMS algorithm did not perform particularly well on the synthesized image. The final output image of figure 3.1(c) appears significantly blurred with the additive noise largely intact.

The ARW-LMS-restored image is presented in figure 3.1(d). The ARW-LMS algorithm is designed to attempt to reduce the visually disturbing attributes of the ATD-LMS algorithm by tuning the filter more closely to the characteristics of the image. As such, it is a highly non-linear filter. For the synthesized image, the ARW size was restricted to be a minimum of 1×1 and a maximum of 5×5 . This maximum size was chosen because of the relatively small size of the synthesized image. In their original work, Song and Pearlman [8,21], and Mahesh *et al.* [10] use a maximum size of 11×11 for a 256×256 pixel input image. This upper limit on the ARW was used since beyond a certain limit it would be unreasonable to believe that all the pixels within the window are from the same ensemble. As the image in our example is smaller, a smaller ARW size is appropriate. For our examples, the value of the weighting constant, η , was fixed to be 7, which is the same as that used by Song and Pearlman [8,21], and Mahesh *et al.* [10]. The ARW-LMS algorithm typically produces much less smoothing at the edges than in uniform regions. As a result, a layer of noise remains in the restored image surrounding each of the structures. Although this is clearly objectionable in the synthesized images, as seen in figure

3.1(d), this effect is not as pronounced in more natural scenes since ideal edges are not common in natural scenes. The ARW-LMS output image appears clearer and sharper than the ATD-LMS restored image, because the ARW-LMS algorithm tends to concentrate the error around the edges where the human visual system masks it. On the other hand, the ATD-LMS algorithm tends to reduce the error uniformly, which also leads to smoothing across the edges. The decreased sharpness of the edges makes the pictures less pleasing to the viewer.

Table 3.1 presents MSE values between the original and restored images for the synthesized image. Both methods suppress the noise in the corrupted image. The ARW-LMS algorithm, however, removes more of the noise as depicted by an MSE value of 58 compared to an MSE value of 237 for the image restored using the ATD-LMS algorithm.

<i>Image / Filter</i>	<i>MSE</i>
Test + Noise	256
ATD-LMS (μ)	237 ($\mu = 0.4 \times 10^{-7}$)
ARW-LMS (max. window size)	58 (5×5)

Table 3.1 Results of noise suppression on the synthesized, 128×128 pixel image with noise variance of 256.

Figure 3.2 presents results using the *Lena* image. The original image, shown in figure 3.2(a), is of size 128×128 pixels with 256 gray levels. Figure 3.2(b) presents the *Lena* image after it has been corrupted with additive white Gaussian noise of variance 256 and some added impulse noise. The ATD-LMS restored image is presented in figure 3.2(c) and the ARW-LMS restored image is presented in figure 3.2(d). The convergence factor, μ , used to minimize the MSE for the ATD-LMS algorithm



(a)



(b)



(c)



(d)

Figure 3.2 Noise Suppression with the Lena image of size 128×128 pixels and a gray level range of $0 - 255$. (a) original image, (b) noise corrupted image; $\sigma_n^2 = 256$, (c) ATD-LMS output image, (d) ARW-LMS output image.

was 1.0×10^{-8} . The ATD-LMS algorithm uniformly suppressed the noise (both the Gaussian and the impulse noise), but still blurred the edges in the image. As a result, the ATD-LMS output image is only marginally better than the input image from a visual perspective.

For the ARW-LMS-restored image, the ARW size was restricted to a lower limit of 1×1 pixel, and an upper limit of 5×5 pixels as before. It is very clear that the ARW-LMS restoration is visually better than the ATD-LMS-restored image, although it was unable to suppress the impulse noise. The ARW-LMS algorithm reduces the noise significantly in flat regions while leaving edge regions (including impulse noise effects which are regarded as edges) virtually unchanged. This accounts for the layer of noise apparent around very distinct edges in the image. Noise is particularly visible as a halo around the girl's hat. In addition, for areas in which there are rapid spatial variations in the image, such as near the girl's mouth and nose, the noise is not significantly reduced. The masking by the human visual system of noise near edges accounts for the visually pleasing effect produced by the ARW-LMS restorations as compared to the blurred results produced by the ATD-LMS restorations shown in figures 3.1(c) and 3.2(c).

Table 3.2 presents the MSE values between the original and restored images for the Lena image corrupted with two levels of noise and for two sizes of the image. Both the ATD-LMS and ARW-LMS methods suppress the noise in the corrupted image in a mean-squared sense, except the ATD-LMS algorithm with the 128×128 image. This failure may be due to the small size of the image, resulting in the filter not being able to converge and track the variations in the image rapidly enough. For the larger Lena image, the ATD-LMS algorithm was successful in bringing the MSE values down and significant noise suppression has been achieved. The ARW-LMS

algorithm again produced better results for both the small and the large image sizes as depicted by the MSE values, which are much lower than the values produced using the ATD-LMS algorithm. For the 512×512 size images, an upper limit of 11×11 was used for the ARW size. This window size was used by Song and Pearlman [8,21], and Mahesh *et al.* [10] for similar image sizes.

<i>Image / Filter</i>	<i>MSE for various image sizes</i>		
	128×128	512×512	512×512
Lena + Noise	256	256	438
ATD-LMS (μ)	476 (1.0×10^{-8})	118 (0.3×10^{-8})	158 (0.3×10^{-8})
ARW-LMS (max. window size)	135 (5×5)	80 (11×11)	131 (11×11)

Table 3.2 Results of noise suppression on the Lena image for various sizes and noise levels.

The overall impression after visually examining the restored images is that the ARW-LMS algorithm produces restorations that are visually more pleasing than those produced by the ATD-LMS algorithm. Yet both algorithms have shortcomings that are manifested in noise left unsuppressed (by the ARW-LMS algorithm) around edges (including spike noise) due to the reduction in window size near edges, and blurring of information-bearing regions by the ATD-LMS algorithm as explained above. The next chapter describes a new algorithm that attempts to overcome the shortcomings in the algorithms discussed here.

3.5 Summary

In this chapter, two different types of adaptive noise-reduction filters were presented. Mathematical derivations and implementation of both filters were discussed in detail, and experimental results were then presented with a comparative analysis of

the performance and shortcomings of these adaptive filters. In the next chapter, the new Adaptive-Neighborhood Noise Subtraction filter will be presented in full detail, including experimental results. Its performance will be evaluated in comparison with that of the adaptive filters discussed in this chapter.

CHAPTER 4

ADAPTIVE-NEIGHBORHOOD NOISE SUBTRACTION

4.1 Introduction

Statistical characteristics of images are of fundamental importance in many areas of image processing. Incorporation of a priori statistical knowledge of spatial correlation in an image, in essence, can lead to considerable improvement in many image processing algorithms. In image restoration, the well-known Wiener filter for minimum mean-squared error (MMSE) estimation is derived from a measure or an estimate of the power spectrum of the image, as well as the MTF of the spatial degradation phenomenon and the noise power spectrum. Unfortunately, the Wiener filter is designed under the assumption of wide-sense stationary (WSS) signal and noise. Although the stationarity assumption for additive, zero-mean, white Gaussian noise is valid for most cases, it is not reasonable for most realistic images, apart from the uninteresting case of uniformly gray image fields. What this means in the case of the Wiener filter is that we will experience uniform filtering throughout the image, with no allowance for changes between edges and flat regions, resulting in unacceptable blurring of high-frequency detail across edges and inadequate filtering of noise in relatively flat areas. Recent methods of circumventing this problem have taken into account the nonstationarity of image signals, resulting in the development of signal-dependent, space-variant algorithms [8-13,16-26]. Two of these algorithms

were discussed in the previous chapter.

In this chapter, we propose a new region-based noise-reduction filter that incorporates adaptive neighborhoods to isolate statistically stationary regions with similar gray levels. For each pixel in the image, a new region is grown. Each region grows from a seed pixel (the current pixel being processed) and includes neighboring pixels having gray-level values that lie within a specified tolerance limit. The shape of the region closely approximates a stationary area, and thus the statistics calculated inside each region will more accurately represent the local feature than those calculated using fixed or adaptive square or rectangular windows, as discussed in previous chapters. Another important advantage of using this type of region growing is that regions are now comparably large on the average (they do not decrease as they approach edges as in the ARW-LMS method described in chapter 3), and contain minimal features. It would thus be logical to estimate the noise using the adaptive neighborhood and subtract it from the region for restoration. Doing this for all regions of the noisy image would result in a restored image. We call this algorithm *Adaptive-Neighborhood Noise Subtraction* (ANNS). The method is a new algorithm in the adaptive-neighborhood image processing paradigm which was first proposed by Gordon and Rangayyan in 1984 [28] and which was developed and expanded upon over the past decade [13,29-34].

Before we get into the derivation of the ANNS filter, details of the adaptive region growing method will be discussed in the next section.

4.2 Region Growing

In fixed-neighborhood image processing (FNIP), the fixed neighborhood is normally defined as an arbitrary (usually square) region which is centered about the

pixel of interest (or the seed pixel). On the other hand, in adaptive-neighborhood image processing (ANIP), the following approach is adopted. The image is treated as being made up of a collection of regions (features or objects) of relatively uniform gray levels. An adaptive neighborhood (AN) is determined for each pixel in the image, being defined as the set of pixels 8-connected to the seed pixel, and having a difference in gray level from that of the seed which is within some limits of tolerance. The tolerance can be set as an additive factor t_1 or a multiplicative factor t_2 , as

$$|y(i, j) - y(k, l)| \leq t_1, \quad (4.1)$$

or

$$\frac{|y(i, j) - y(k, l)|}{y(k, l)} \leq t_2, \quad (4.2)$$

where $y(k, l)$ is the seed pixel, and $y(i, j)$ represents pixels 8-connected to the seed pixel. Thus t_1 or t_2 determines the maximum allowed deviation in the gray level from the seed pixel value within each AN, and any deviation less than this is considered to be an intrinsic property of the AN region. The number of pixels in any AN is limited by a pre-determined number Q . The tolerance used for growing regions in this thesis is the additive tolerance t_1 , and is made adaptive depending on the signal activity in the region and the features surrounding it.

From the definition above, it should be clear that an AN is formed for each pixel $y(k, l)$ in the degraded image. Furthermore, the various ANs over an image field are allowed to overlap. Ideally, therefore, ANs within a distinct uniform object in an image will totally overlap for all seed pixels which lie within that object, and will be identical in the absence of noise above the threshold t_1 . On the other hand, the ANs for all seed pixels which do not lie directly within the object will exclude the object.

A second region is grown molded to the outline of the 8-connected first layer of

the AN with a predefined width, and is called the background layer.

The program developed to grow the AN [29,30,31] also calculates the number of pixels making the foreground and background of each AN region, and stores them together with their values in the variables $fgCount(k,l)$, $bgCount(k,l)$, $fgValue(i)$, and $bgValue(j)$, where $i = 0, 1, 2, \dots, fgCount - 1$, and $j = 0, 1, 2, \dots, bgCount - 1$. Thus, a $fgCount$ and a $bgCount$ exist for every seed pixel. This makes it easy to calculate statistics inside the neighborhood, as will be shown later. Figure 4.1(a) shows a schematic of the region growing procedure, and figure 4.1(b) shows an actual AN region grown from the seed pixel at location (96,17) inside the Lena image shown in figure 3.2(a).

In developing the ANNS algorithm, we used a background layer which was three pixels wide. The advantage of growing a background is to aid in removing spikes that may appear in the noisy image. If the seed pixel is a spike with a large value that does not lie within the tolerance level specified, the foreground will only be that seed pixel and will not grow further, but in any case a background layer will grow around the spike and will aid in removal of the spike. This can be accomplished by replacing the seed value (the spike) with the mean of the background layer. Alternatively, we may replace the seed value with the total mean of both the foreground and the background layers combined as one region. It should be noted that the background is only used when the foreground is a single pixel representing a spike.

Now that a foreground is defined for each seed pixel, the restoration problem reduces to one of estimating the noise in the foreground region and then subtracting it from the region to obtain an estimate of the undegraded image.

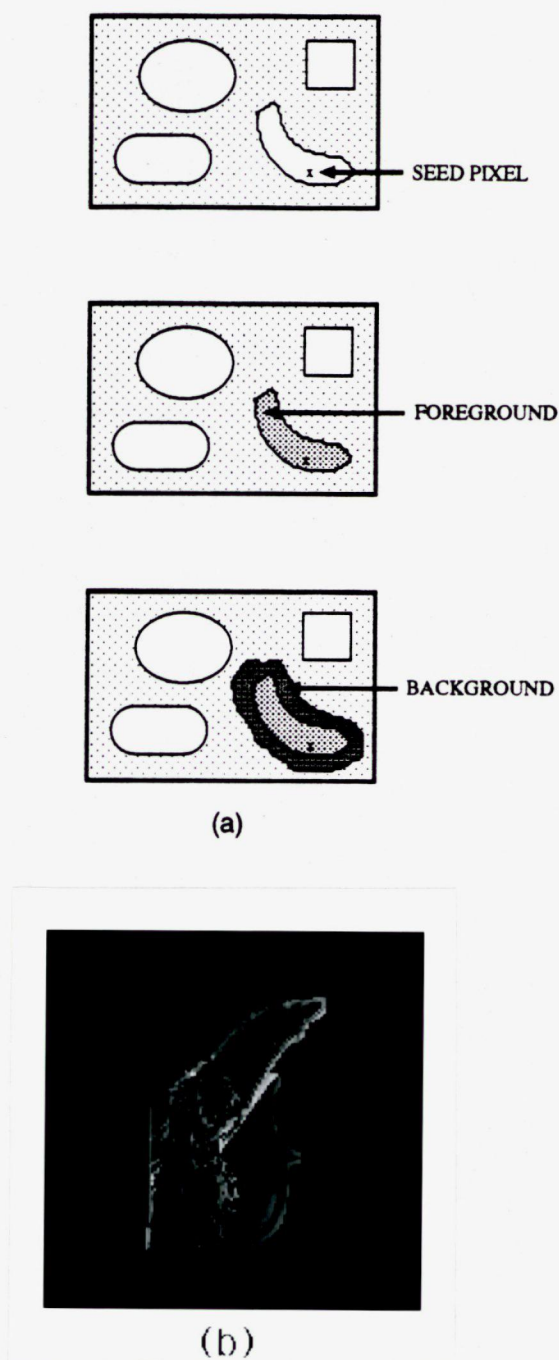


Figure 4.1 Adaptive-neighborhood (AN) region growing. (a) From a seed pixel inside an object or feature, an AN is formed. The first layer of the AN defines the object (foreground) and the second layer defines its background. (b) The foreground and background regions grown from a seed pixel at location (96, 17) inside the image shown in figure 3.2(a).

4.3 The Adaptive-Neighborhood Noise Subtraction Method

The main theme of this restoration algorithm is to estimate the noise in the seed pixel $y(k, l)$ which lies inside the AN region, and then to subtract the noise from the seed pixel to obtain an estimate of the original undegraded seed pixel $x(k, l)$.

4.3.1 Derivation of the ANNS filter

The strategy used in deriving the ANNS filter is based on the same principles as those of the ARW-LMS algorithm; that of a zero-mean random variable x of variance σ_x^2 in the presence of additive white Gaussian noise. The degraded random variable is $y = x + n$, where the noise n is of zero mean, of variance σ_n^2 , and is uncorrelated to x . The variables (k, l) have been suppressed for convenience of notation. It can easily be shown that

$$\sigma_y^2 = \sigma_x^2 + \sigma_n^2. \quad (4.3)$$

An estimate of the additive noise within an AN can be computed from the corrupted image as

$$\hat{n}(k, l) = \alpha \cdot y(k, l), \quad (4.4)$$

where α is a scale factor which depends on the characteristics of the AN region grown.

The estimate of $x(k, l)$ is thus,

$$\hat{x}(k, l) = y(k, l) - \hat{n}(k, l), \quad (4.5)$$

which reduces to

$$\hat{x}(k, l) = \beta \cdot y(k, l), \quad (4.6)$$

where $\beta = 1 - \alpha$.

As described in chapter 3 for the ARW-LMS algorithm, if the images used are of non-zero mean, the estimate of equation (4.6) can be used by first subtracting the mean of each image from both sides of the equation; thus

$$\hat{x}(k, l) = m_y + (1 - \alpha) \cdot [y(k, l) - m_y], \quad (4.7)$$

where m_y is the a posteriori mean of the degraded image $y(k, l)$, which is also equal to the a priori mean m_x of the original image $x(k, l)$ for zero-mean noise.

The problem now is to find the noise estimator α , based on the criterion that the estimated noise variance σ_n^2 be equal to the original noise variance σ_n^2 . The solution is given as follows:

$$\begin{aligned} \sigma_n^2 &= E\{\hat{n}^2\} \\ &= E\{(\alpha \cdot [y(k, l) - m_y])^2\}. \end{aligned} \quad (4.8)$$

Now, σ_n^2 can be expressed as

$$\sigma_n^2 = \alpha^2 \cdot \sigma_y^2, \quad (4.9)$$

and from equation (4.3) we have

$$\sigma_n^2 = \alpha^2 \cdot (\sigma_x^2 + \sigma_n^2). \quad (4.10)$$

Then, the noise estimator α becomes

$$\alpha = \sqrt{\frac{\sigma_n^2}{\sigma_x^2 + \sigma_n^2}}. \quad (4.11)$$

Thus, the estimate of equation (4.7) becomes

$$\hat{x}(k, l) = m_y + \left(1 - \sqrt{\frac{\sigma_n^2}{\sigma_x^2 + \sigma_n^2}}\right) \cdot [y(k, l) - m_y]. \quad (4.12)$$

The estimate $\hat{x}(k, l)$ given by equation (4.12) can be considered to be an approximation to the original image $x(k, l)$ if we are able to obtain accurate values for the statistical parameters m_y and σ_x^2 . The noise variance, σ_n^2 , is obtained in a similar manner as that given in chapter 3.

4.3.2 Implementation of the ANNS filter

In implementing equation (4.12), we must first derive the local (sample) statistics, \hat{m}_y and $\hat{\sigma}_x^2$, from each AN region grown at every seed pixel location (k, l) in the degraded image. Since the AN regions grow into areas with similar gray levels, there should be no problem with assuming that the sample statistics in the regions are stationary and ergodic.

The local mean and variance of the AN foreground region grown for the seed pixel $y(k, l)$ are calculated as

$$\hat{m}_y = \frac{1}{fgCount} \sum_{i=0}^{fgCount-1} fgValue(i), \quad (4.13)$$

and

$$\hat{\sigma}_y^2 = \frac{1}{fgCount} \sum_{i=0}^{fgCount-1} [fgValue(i) - \hat{m}_y]^2. \quad (4.14)$$

The local variance of the original signal is thus given by

$$\hat{\sigma}_x^2 = \begin{cases} \hat{\sigma}_y^2 - \sigma_n^2 & \text{if } \hat{\sigma}_y^2 > \sigma_n^2 \\ 0 & \text{otherwise.} \end{cases} \quad (4.15)$$

Using the local statistics, \hat{m}_y and $\hat{\sigma}_x^2$, in place of the ensemble statistics, m_y and σ_x^2 , equation (4.12) thus becomes

$$\hat{x}(k, l) = \hat{m}_y + \left(1 - \sqrt{\frac{\sigma_n^2}{\hat{\sigma}_x^2 + \sigma_n^2}} \right) \cdot [y(k, l) - \hat{m}_y]. \quad (4.16)$$

It should be noted that the parameters $fgCount$, \hat{m}_y , $\hat{\sigma}_y^2$, and $\hat{\sigma}_x^2$, depend on the point (k, l) , and should in fact be denoted by $fgCount(k, l)$ and so on. This has been suppressed for convenience of notation, but should not be forgotten.

As mentioned in section 4.2 above, the tolerance used for growing regions is the additive tolerance t_1 , and is made adaptive depending on the signal activity in the region and the features surrounding it. The AN is first grown with the tolerance set to the full dynamic range of the input image (i.e., initially $t_1 = 256$). This results in a square region of size Q pixels initially being formed, where Q is defined in section 4.2 above. Using the foreground pixels in the AN, a measure of the uncorrupted signal activity in the AN of the seed pixel is given by the local signal variance $\hat{\sigma}_x^2$. This signal variance in the current AN is then compared with the noise variance σ_n^2 , and if $\hat{\sigma}_x^2 > 2\sigma_n^2$, it is assumed that the AN has identified a region in the original image with significant structural characteristics such as an edge, or other distinct objects. This is contrary to the desired characteristics of an AN. An AN is to be formed such that it includes relatively uniform structures in the original image, so that the primary source of variance in the AN is the additive noise. Therefore, the gray level tolerance used to define the AN is modified to $t_1 = 2\hat{\sigma}_x$, with the idea that the signal standard deviation $\hat{\sigma}_x$ be used to define the new AN. The AN is grown again using this new tolerance. As the tolerance has been reduced, the new AN developed presumably will not contain edges or structural features in the image, but will rather grow up to but not including these features. Using this approach to define the AN, the statistics of the AN, \hat{m}_y and $\hat{\sigma}_x^2$, are used in equation (4.16) to estimate the uncorrupted image at each pixel location. In the special situation that the foreground is only one pixel, the AN is enlarged to include the background layer of the pixels. This approach is particularly useful in the presence of impulse noise in the corrupted image.

The advantage of the ANNS method is the fact that in flat or slowly-varying regions, the signal variance will be small compared to $2\sigma_n^2$, and the AN foreground will grow to the maximum foreground bound Q . On the other hand, in busy regions where the signal variance is high compared to $2\sigma_n^2$, the tolerance will be reduced and the foreground will grow right up to any edge present but not across it. This will have the effect of removing the noise in the foreground region, and will, therefore, remove noise that is very close to edges, which the ARW-LMS filter discussed in chapter 3 fails to do.

4.4 Experimental Results and Discussion

We applied the ANNS algorithm to the same corrupted images as those used for the ATD-LMS and ARW-LMS algorithms given in chapter 3. The results were evaluated by direct visual examination and by comparing the MSE values obtained from the results of the ANNS algorithm to those obtained by using the ATD-LMS and ARW-LMS algorithms.

The ANNS algorithm was first applied to the synthesized image after it was corrupted with additive white Gaussian noise. Figure 4.2(a) shows the original image, which was shown previously in figure 3.1(a). The noise-corrupted image is shown in figure 4.2(b) (same as that in figure 3.1(b)). The same noise variance of 256 used in section 3.4 was also used here to degrade the original image. The ANNS-restored image is shown in figure 4.2(c), and looks similar to the ARW-LMS output image of figure 3.1(d) shown again in figure 4.2(d) for convenience. The ATD-LMS output image of figure 3.1(c) is also shown again in figure 4.2(e) for comparison.

For the ANNS filter, the maximum AN size Q was assigned a value of 25 pixels. This allows us to directly compare the performance of the ANNS method against

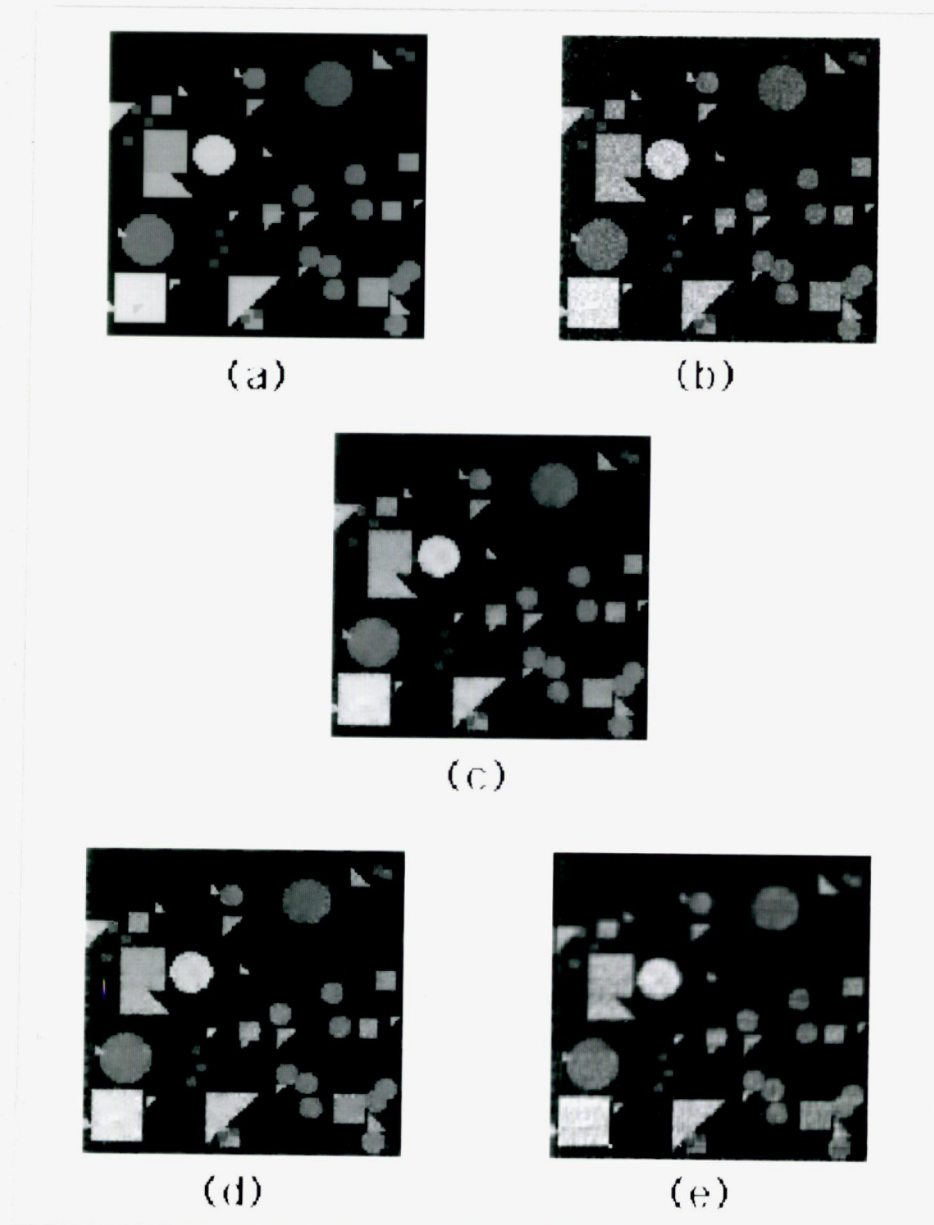


Figure 4.2 Noise Suppression with a synthesized image of size 128×128 pixels and a gray level range of $0 - 255$. (a) original image, (b) noise corrupted image; $\sigma_n^2 = 256$, (c) ANNS output image, (d) ARW-LMS output image, (e) ATD-LMS output image.

<i>Image / Filter</i>	<i>Time</i>	<i>MSE</i>
Test + Noise	–	256
ATD-LMS (μ)	1 min	237 ($\mu = 0.4 \times 10^{-7}$)
ARW-LMS (max. window size)	2 min	58 (5×5)
ANNS (AN size)	3 min	44 ($Q = 25$)
Two-pass ARW-LMS (max. window size)	4 min	50 (5×5)
Two-pass ANNS (AN size)	6 min	25 ($Q = 25$)

Table 4.1 Results of noise suppression on the synthesized, 128×128 pixel image with noise variance of 256. Approximate computer processing time for the various filters using a SUN/Sparc-2 workstation are also listed.

the ARW-LMS method for equal-sized regions of support. However, the ANNS algorithm uses a variable-shape window (adaptive neighborhood) in order to compute the restored image. Unlike the ARW-LMS filter window, the size of the AN is not compromised near edges in the image; rather, its shape changes according to the contextual details in the image. This allows better estimation of the noise near edges, and thus permits greater noise suppression in such areas. Although a very small layer of noise pixels is still visible around the objects in the ANNS image of figure 4.2(c), the noise is at a much lower level than that observed in the ARW-LMS output image of figure 4.2(d). Both the ANNS and ARW-LMS methods appear to reduce the noise equally well in relatively uniform regions of the image. However, there does appear to be some patchiness in the uniform regions of both these output images. The ANNS-restored image is clearly the best of the three restoration methods based on direct visual examination of the three output images.

Table 4.1 presents the approximate computer processing time for each algorithm, and MSE values between the original and restored images for the synthesized image. The ARW-LMS and ANNS methods have significantly reduced the noise in the cor-

rupted image. The ANNS algorithm, however, has the lowest MSE (44), followed by the ARW-LMS algorithm (58), and the largest MSE is in the ATD-LMS output image (237). Thus, both qualitatively and quantitatively, the ANNS method is the best of the three methods tried.

Figure 4.3 presents results using the Lena image. The original image is shown in figure 4.3(a), and the noise corrupted image is shown in figure 4.3(b). The same noise variance of 256 used in section 3.4 was used here again. However, we added some impulse noise to the noisy image to demonstrate the ability of the ANNS filter to suppress spikes as well as Gaussian noise. The ANNS-restored image is shown in figure 4.3(c); the ARW-LMS-restored image of figure 3.2(d) is shown again in figure 4.3(d), and the ATD-LMS-restored image of figure 3.2(c) is shown again in figure 4.3(e). For the ANNS filter, the maximum AN size Q was assigned a value of 25 pixels, allowing us to directly compare the ANNS method and the ARW-LMS method. The ANNS and the ARW-LMS output image appear to be quite similar, and both are significantly better than the ATD-LMS-restored image. However, by inspection of figure 4.3(d), it is clear that the ARW-LMS method failed to suppress the spikes in the noisy image. This is because spikes appear as edges to the ARW-LMS algorithm, and thus the window size is reduced as it approaches a spike pixel such that when the pixel being filtered is the spike, the window is too small and insufficient smoothing of the spike occurs. On the other hand, the ANNS filter was successful in filtering out all the spikes as explained in section 4.2 above. As discussed previously in section 3.4, closer examination of the ARW-LMS output image shows a layer of noise that is apparent around edges in the image. Again, the ANNS-restored image is clearly the best output image based on direct visual examination of the three images.

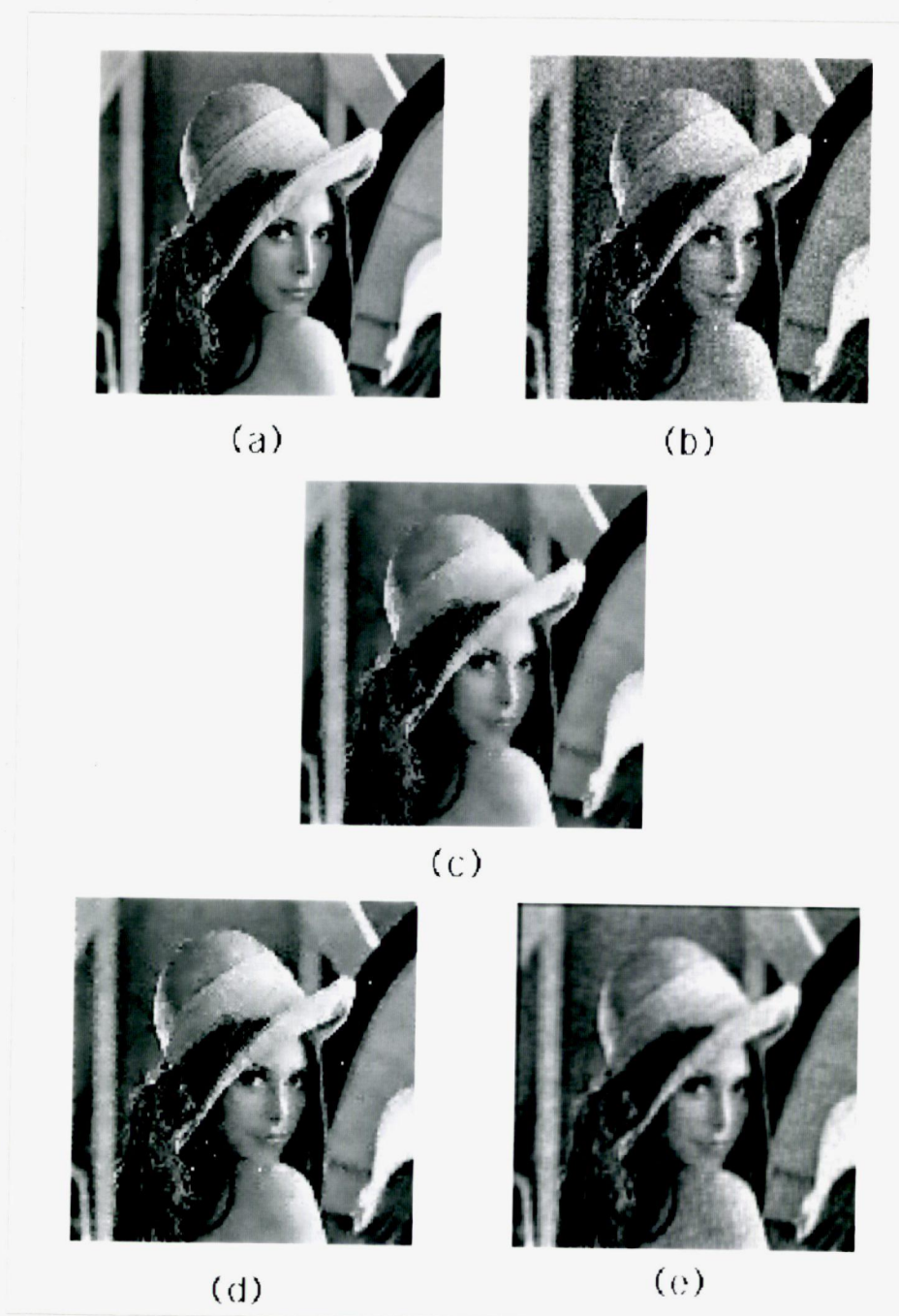


Figure 4.3 Noise Suppression results obtained with the Lena image of size 128×128 pixels and a gray level range of 0 – 255. (a) original image, (b) noise corrupted image; $\sigma_n^2 = 256$, (c) ANNS output image, (d) ARW-LMS output image, (e) ATD-LMS output image.

Image / Filter	MSE for various image sizes		
	128 × 128	512 × 512	512 × 512
Lena + Noise	256	256	438
ATD-LMS (μ)	476 (1.0×10^{-8})	118 (0.3×10^{-8})	158 (0.3×10^{-8})
ARW-LMS (max. window size)	135 (5×5)	80 (11×11)	131 (11×11)
ANNS (AN size)	122 ($Q = 25$)	66 ($Q = 121$)	87 ($Q = 121$)

Table 4.2 Results of noise suppression on the Lena image for various sizes and noise levels.

Table 4.2 presents the MSE values between the original and restored images for the Lena image corrupted with two levels of noise and for two sizes of the image. All three methods suppress the noise in the corrupted image in a mean-squared sense, except the ATD-LMS algorithm with the 128×128 image as discussed in section 3.4. For the images presented in figure 4.3, the ANNS algorithm again has the lowest MSE (122), followed by the ARW-LMS algorithm (135), and the largest MSE is in the ATD-LMS output image (476). The next two columns of table 4.2 present results using the Lena image of size 512×512 pixels and noise levels of 256 and 438. For the ARW-LMS and the ANNS methods, window limits of 11×11 and $Q = 121$ were used respectively. This is the same window size used by Song and Pearlman [8,21], and Mahesh *et al.* [10] for similar image sizes. The ANNS method consistently produces output images with lower MSE than those for the other two algorithms. The ARW-LMS and the ANNS images are visually very similar, and the only difference between them is apparently the layer of noise that is left unsuppressed in the ARW-LMS output image, as compared to less overall noise, even around edges, in the ANNS output image.

In figure 4.4, we present results from the repeated application of the ANNS and ARW-LMS algorithms to the synthesized image in figure 4.2(b). Repeated applica-

tion is a very powerful and useful attribute of these adaptive-window methods. Figure 4.4(a) shows the output image obtained after passing the noise-corrupted image in figure 4.2(b) twice through the ANNS filter. Figure 4.4(b) is the output image obtained by passing the image in figure 4.2(b) twice through the ARW-LMS filter. For both of these algorithms, knowledge of the additive noise variance is required for the second pass through the filter. The second-pass noise variance was estimated by calculating the variance of the output image after the first pass, and subtracting from it an estimate of the variance of the noise-free image $x(k, l)$. Thus, no additional information is required for multiple passes through the ANNS or ARW-LMS algorithms. The output images show significant improvement over the results from a single pass through the algorithms. The major artifact after the first pass through the algorithm was the retention of noise around distinct edges in the image. For both algorithms, this layer of noise is greatly reduced by the second application; however, the noise is much more reduced by the ANNS method than by the ARW-LMS method. A second artifact was the patchy appearance of the background region in the image as demonstrated in figures 4.2(c) and 4.2(d). This patchy appearance too is reduced after the second pass for both algorithms.

The MSE between the images in figure 4.4 and the original image are presented in table 4.1 on the bottom two rows. The ANNS-restored image is seen to have a lower MSE (25) than the ARW-LMS-restored image (50). Thus, the ANNS-restored image is again qualitatively and quantitatively superior. While the two-pass method was successful with the synthesized image, it produced too much smoothing when applied to the natural Lena image scene. Further work will be required in order to investigate the practical use of repeated application of these adaptive region-based algorithms to natural scenes.

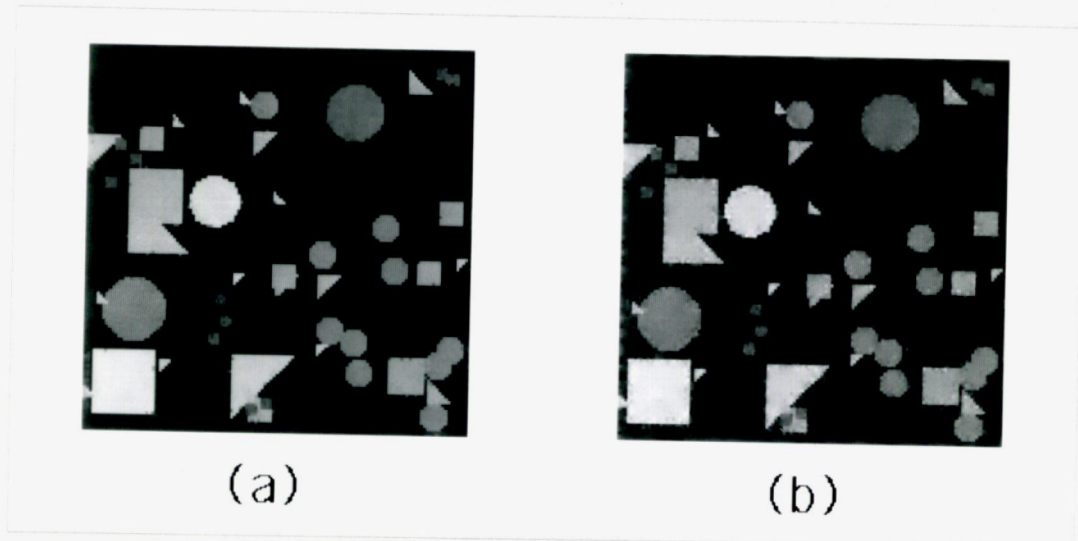


Figure 4.4 Noise Suppression with the synthesized image in figure 4.2(b) using two passes through (a) the ANNS filter, and (b) the ARW-LMS filter.

4.5 Summary

In this chapter, the new ANNS filter was presented. This new algorithm uses adaptive neighborhoods to define stationary regions, and obtains an estimate of the noise in each region as a function of the input image region. This noise is then subtracted from the seed pixel of the corresponding region to obtain the restored seed pixel. Growing AN regions for all the pixels in the input image and applying the adaptive filter, we eventually obtain the restored output image. Experimental results were then presented, with a comparative analysis between the performance of the ANNS algorithm and the performance of the ARW-LMS and ATD-LMS algorithms discussed in chapter 3. The ANNS algorithm was shown to produce better output images than the other two algorithms in terms of both visual quality and MSE. The adaptive window techniques (ANNS and ARW-LMS) were further applied in a two-pass procedure, and the ANNS method was again shown to be superior with extremely low MSE values. The ANNS method is capable of noise filtering without blurring and without leaving noise around edges and sharp features in the given image. The method is also adaptive, and takes into account nonstationarities across the image field.

CHAPTER 5

SECTIONED IMAGE DEBLURRING

5.1 Introduction

In a number of different applied problems in signal processing (e.g., speech, data acquisition, and image restoration) it becomes necessary to carry out deconvolution. Deconvolution is the process of removing the effects of the impulse response of a linear system from a signal which passed through the system. This problem is commonly made difficult by the ill-conditioned or singular behavior of the impulse response of the linear system, coupled with the noise inherent in any real-world system.

Many methods have been proposed for image deconvolution in the presence of noise [1-7,12,15,17,18,26,35-45]. In many cases the restoration techniques (such as the Wiener and PSE filters) are based on the assumption that the image can be modeled by a stationary random field, and restoration is achieved by filtering the degraded image with a linear space-invariant restoration filter, the frequency response of which is a function of the ideal image power spectral density. There are at least two difficulties in this approach. For a typical image, any given part of the image generally differs sufficiently from the other parts so that the stationarity assumption over the entire image is not generally valid. The second difficulty is that, in practice, the image power spectral density is not given and has to be estimated. A common procedure to estimate the power spectral density of a signal involves sectioning the one-dimensional

signal into smaller (presumably) stationary segments, taking modified periodograms of these sections, and averaging these modified periodograms [46,47,48] in an attempt to reduce the variance in the power spectral estimate and thus create a more accurate spectral estimate. This basic technique has been applied to two-dimensional signals (images) [1,3,4,5,6,12,15,35] by dividing the available image into a number of subimages and then averaging the log of their spectral magnitudes. In order for the subdivision to be valid, the blurring PSF must have a region of support much smaller than the size of the subimages. As a result, the size of the subimages cannot be made arbitrarily small. Thus, the number of subimages used to form the ensemble average is limited, and consequently the variance of the spectral estimate from the subimages can be relatively high. This leads to a poor estimate of the original power spectral density.

Another consequence of the stationarity assumption and the use of space-invariant filtering is the fact that deblurred images suffer from edge-effects at the boundaries of the image [5]. Since the original image is, theoretically, of indefinite extent compared to the PSF of the blurring system, the edges of a *finite extent* blurred image have contributions from outside the image boundaries convolved into the image by the PSF. The effects at the edges from deconvolution with incomplete information (scene information beyond the image's boundary) causes different contributions from outside the image boundaries during deblurring than those contributions that were convolved into the image boundaries during blurring. This leads to a layer of boundary pixels taking incorrect values during deblurring, and consequently, edge artifacts at the image boundaries become a source of degradation, especially if boundary information is of importance.

Recent attempts to overcome the problems caused by the inherent nonstationarity of images have resulted in various adaptive techniques based on sectioning the given image into smaller subsections and assuming different stationary models for each section [12,17,18,26]. Other techniques enhance the performance of nonadaptive filters by using radiometric and geometric transforms to generate nearly stationary (block stationary) images in the first and second moments [16]. The radiometric transform generates stationary mean and variance, while the geometric transform gives stationary autocorrelation.

One method that has not been fully exploited is sectioned *deconvolution*. In the iterative sectioned maximum a posteriori (MAP) restoration technique proposed by Trussell and Hunt [17,18], the input image is broken into smaller $P \times P$ sections and the MAP estimate of the original image section is developed (in the space domain using computationally expensive matrix operations) and then iterated upon for refinement. This procedure is carried out on each section using an overlap-save technique to reduce edge effects.

Since sectioning the image presumably causes each individual section to better approximate a stationary section of the whole image, a simpler approach to sectioned deblurring would be to use a conventional, space-invariant, frequency domain filter to deblur each section individually and then to combine the deblurred sections to form the final deblurred image. This method is discussed in the next section, and its shortcomings are analyzed. In chapter 6, a new adaptive-neighborhood deblurring (AND) method is introduced, which attempts to solve the problems of the fixed-neighborhood sectioned deblurring (FNSD) technique described in the next section, and its performance is evaluated in comparison with the FNSD method.

5.2 Fixed-Neighborhood Sectioned Deblurring

The problem which we consider is to restore an image degraded by a blur PSF and additive noise. The degradation model was shown in figure 2.1, and can be represented by

$$y(k, l) = x(k, l) * h(k, l) + n(k, l), \quad (5.1)$$

where $x(k, l)$ is the original undegraded image which is unknown, and $n(k, l)$ is the additive noise component, which is assumed to be a zero-mean, white, Gaussian function of variance σ_n^2 . The restoration problem to be considered is to find an estimate of $x(k, l)$ given the PSF $h(k, l)$, the observed image $y(k, l)$, and some statistical properties of the noise process. In the frequency domain, equation (5.1) becomes

$$Y(u, v) = X(u, v) \cdot H(u, v) + N(u, v). \quad (5.2)$$

A typical solution is to deblur using a standard filter such as the Wiener (MMSE) filter or the PSE filter. The Wiener solution is given by

$$H_w(u, v) = \frac{H^*(u, v) \cdot P_x(u, v)}{|H(u, v)|^2 \cdot P_x(u, v) + P_n(u, v)}, \quad (5.3)$$

where the asterisk superscript indicates complex conjugation, $P_x(u, v)$ represents the power spectrum of the original image, and $P_n(u, v)$ represents the power spectrum of the noise process. The deblurred image is thus given by

$$\hat{X}(u, v) = H_w(u, v) \cdot Y(u, v). \quad (5.4)$$

The PSE solution is given by

$$H_{PSE}(u, v) = \left[\frac{P_x(u, v)}{|H(u, v)|^2 \cdot P_x(u, v) + P_n(u, v)} \right]^{1/2}, \quad (5.5)$$

and

$$\hat{X}(u, v) = H_{PSE}(u, v) \cdot Y(u, v), \quad (5.6)$$

gives the frequency domain values of the deblurred image, which are inverse Fourier transformed to generate the actual space domain restored image.

It is important to note that the solutions given by equation (5.3) and equation (5.5) are based on the assumption that $y(k, l)$ can be approximately modeled by a stationary random field. For a typical image, this stationarity assumption is not valid, as explained in section 5.1 above. To reduce the effects of the nonstationarity, a reasonable approach is to implement the deblurring systems on a short-space basis in which the degraded image is sectioned into many subimages, each subimage is deblurred separately, and then the subimages are combined. Assuming that each section (or subimage) now approximates a stationary random field, one technique that would theoretically suppress edge effects and obtain more accurate power spectral estimates is to center each subimage in a square region of size comparable to that of the input image, and then pad the region surrounding the centered subimage with its mean value. If the subimage were really stationary, then its mean value would be very close to its pixel values. This argument is a direct result of the definition of stationarity: the statistics of a stationary region are independent of pixel position. Mean-padding would thus cause the larger square region to also be stationary. Therefore, when deblurring each section, contributions from outside the section boundaries that are convolved into the section are not so different from the pixel values inside that section, thus (theoretically) the edge effects are suppressed. Another advantage of mean-padding is that more accurate power spectral estimates (used in the Wiener and PSE equations) are now possible because of the stationarity of the whole region, and this would improve the performance of the Wiener and PSE filters. To further reduce edge effects and obtain a more accurate frequency domain representation, the whole mean-padded region is multiplied in the space domain with a smooth window func-

tion of the same size. A good choice for a smooth window function is the Hamming window, because of its small frequency domain side lobes and narrow bandwidth. It is important to note that due to the finite extent of the regions, not using a window is equivalent to multiplying the region, in the space domain, with a unity-amplitude rectangular window, which has relatively large frequency domain side lobes. This causes the frequency domain representation of the region to exhibit large under- and overshoots, thus giving an incorrect frequency domain approximation [49].

Using the above argument and assuming that each section (of size $P \times P$ pixels) is large compared to the region of support of the blur PSF, but small compared to the actual image dimensions $M \times M$ ($P = 32$ is typical for $M = 128$), each section of the image can be expressed as the convolution of the PSF with an equivalent section from the original undegraded image $x(k, l)$. Thus, from equation (5.1),

$$y_i(k, l) \simeq x_i(k, l) * h(k, l) + n_i(k, l), \quad (5.7)$$

is the approximate description of the image formation law for each mean-padded region $y_i(k, l)$. In the frequency domain, equation (5.7) becomes

$$Y_i(u, v) = X_i(u, v) \cdot H(u, v) + N_i(u, v). \quad (5.8)$$

Now, applying the two-dimensional Hamming window of size $M \times M$, given by

$$w_H(k, l) = \left[0.54 - 0.46 \cdot \cos \left(\frac{2 \cdot \pi \cdot k}{M-1} \right) \right] \cdot \left[0.54 - 0.46 \cdot \cos \left(\frac{2 \cdot \pi \cdot l}{M-1} \right) \right], \quad (5.9)$$

to each region we obtain

$$y_i(k, l) \cdot w_H(k, l) = [x_i(k, l) * h(k, l)] \cdot w_H(k, l) + n_i(k, l) \cdot w_H(k, l), \quad (5.10)$$

or

$$\tilde{y}_i(k, l) \simeq \tilde{x}_i(k, l) * h(k, l) + \tilde{n}_i(k, l), \quad (5.11)$$

where the tilde (\sim) represents the corresponding windowed regions. The power spectrum of each region $y_i(k, l)$ can be derived from equation (5.11) as

$$P_y^i(u, v) \simeq |H(u, v)|^2 \cdot P_x^i(u, v) + P_n^i(u, v), \quad (5.12)$$

where the superscripts denote the i^{th} region. The power spectral density $P_y^i(u, v)$ is computed from the Fourier transform of the windowed region $\tilde{y}_i(k, l)$ using [46]

$$P_y^i(u, v) = \frac{|\tilde{Y}_i(u, v)|^2}{\sum_{i=0}^{M-1} \sum_{j=0}^{M-1} w_H^2(i, j)}. \quad (5.13)$$

The Wiener filter of equation (5.3) is then applied to each region given by equation (5.8) to recover $X_i(u, v)$ from $Y_i(u, v)$. This is expressed as

$$\hat{X}_i(u, v) = Y_i(u, v) \cdot \frac{H^*(u, v) \cdot P_x^i(u, v)}{|H(u, v)|^2 \cdot P_x^i(u, v) + P_n(u, v)}, \quad (5.14)$$

where $P_n^i(u, v) = P_n(u, v) = \sigma_n^2$ for Gaussian white noise. A similar solution can be obtained using the PSE filter by applying equation (5.5) to each region as

$$\hat{X}_i(u, v) = Y_i(u, v) \cdot \left[\frac{P_x^i(u, v)}{|H(u, v)|^2 \cdot P_x^i(u, v) + P_n(u, v)} \right]^{1/2}, \quad (5.15)$$

and the space domain deblurred region, $\hat{x}_i(k, l)$, is obtained from the inverse Fourier transform of $\hat{X}_i(u, v)$.

The quantity in equation (5.12) gives the denominator in equation (5.14) and equation (5.15) for the Wiener and PSE filters respectively. Stockham *et al.* [15] argued that $P_x^i(u, v)$ for the numerator in equation (5.14) and equation (5.15) could be approximated by an average power spectrum evaluated over a wide variety of similar regions.

The final restored image $\hat{x}(k, l)$ is obtained by translating the individual restored sections from the center of the corresponding restored mean-padded regions $\hat{x}_i(k, l)$ to their original locations, thus filling the whole image space of $M \times M$ pixels.

It will be shown experimentally in the next section that the stationarity assumption for the square sections used above is not really an accurate assumption because of the inability of square sections to discriminate between flat and busy regions of any given image. Furthermore, because of the limitations on section size (each section of size $P \times P$ pixels must be large compared to the region of support of the blur PSF), sections cannot be made arbitrarily small. Thus, the mean value of a section is quite different from its pixel values, and consequently, edge artifacts at section boundaries do occur. To partially solve the problem of edge artifacts, the sections could be overlapped by one-half the section size in each dimension. This, however, does not reduce the edge effects at the image boundaries.

5.3 Experimental Results and Discussion

In this section, some experimental results which illustrate the performance of the preceding algorithm are presented. The results are evaluated by direct visual examination and by calculation of the MSE between the known original and deblurred images.

The FNSD method was first applied to the 128×128 pixel, 256 gray level "Lena" image after it was degraded by a Gaussian-shaped blur PSF with a radial standard deviation $\sigma_r = 3$ pixels and noise to 35 dB signal-to-noise ratio (SNR). The original image is shown in figure 5.1(a) and its blurred noisy version is shown in figure 5.1(b). Figures 5.1(c,d,e) represent three different fixed-neighborhood sections of the blurred image in figure 5.1(b). Each section of size 32×32 pixels is centered in a square region of the same size as the full image (128×128) and the surrounding area is padded with the mean value of the section. It is very clear from figures 5.1(c,d,e) that the stationarity assumption for each section is not really an accurate assumption

because of the fact that each section still contains various image features such as high-variance edges. This is also obvious from the values of the mean-padded areas which are very much different from the pixel values of the corresponding centered sections. Figure 5.1(f) shows the windowed version of the region in figure 5.1(e) where a Hamming window was used as described in the previous section. All the regions that were created for FNSD were windowed in the same manner as that of figure 5.1(f) in an attempt to reduce to some extent the edge effects between the borders of the centered sections and their mean-padded surroundings. It is clear from figure 5.1(f) that windowing does not do a good job of reducing the edges, mainly because the sections are not as stationary as theoretically required.

Figure 5.2(a) shows the restored image after using the FNSD algorithm with the sectioned Wiener filter of equation (5.14) and no overlapping of the sections. It is clear that severe edge artifacts have occurred at the four borders of each section due to the nonstationarity of the sections and the large difference between their mean values and pixel values. To partially overcome the edge effect problems, adjacent sections were overlapped by one-half the section size in both dimensions. The deblurred image using overlapping sections of size 32×32 pixels and the sectioned Wiener filter is shown in figure 5.2(b). Overlapping by half the section size effectively suppresses the edge artifacts associated with the inner sections of the image. It does not, however, reduce edge effects at the boundaries of the image because of the lack of information beyond the image boundaries. The image in figure 5.2(b) is, however, a significant improvement over the deblurred image in figure 5.2(a). Figure 5.2(c) shows the deblurred image obtained by using smaller overlapping sections of size 16×16 pixels and the same Wiener filter. It would be expected that the smaller the section size the more stationary it would become. This is very true, and is obvious from the fact that more

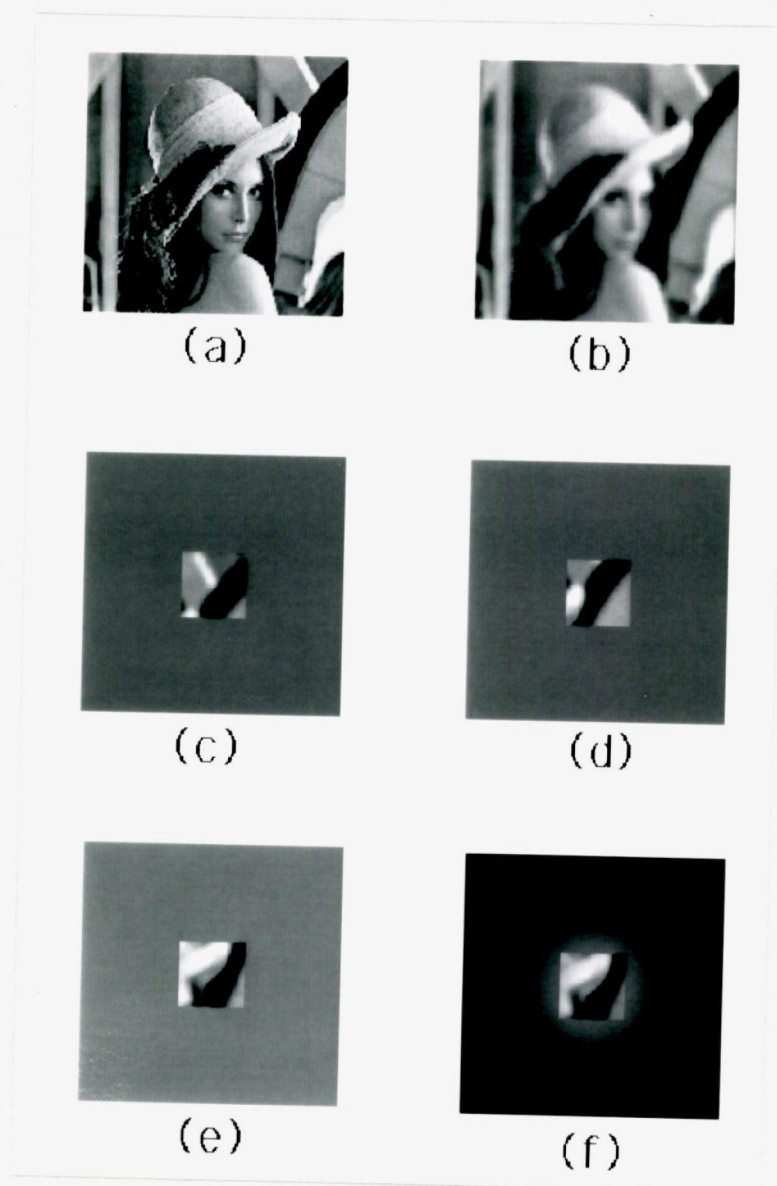


Figure 5.1 Sectioning of the Lena image of size 128×128 pixels and a gray level range of $0 - 255$. (a) original image. (b) blurred image with a Gaussian-shaped blur function; $MSE = 607$. (c), (d), and (e) show three 32×32 sections mean-padded to 128×128 pixels created for sectioned deblurring. (f) the windowed equivalent of the region in (e).

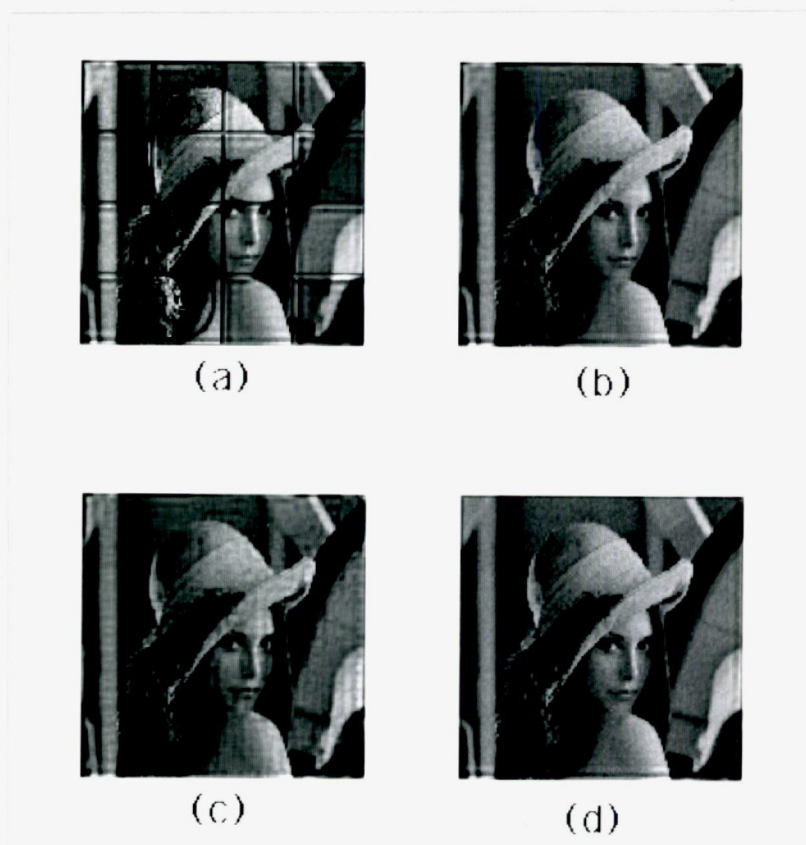


Figure 5.2 Sectioned deblurring results for the Lena image of size 128×128 pixels and a gray level range of 0 – 255. (a) deblurred image using non-overlapping sections of size 32×32 pixels; $\text{MSE} = 1255$. (b) Wiener-deblurred image using overlapped sections of size 32×32 pixels; $\text{MSE} = 501$. (c) Wiener-deblurred image using overlapped sections of size 16×16 pixels; $\text{MSE} = 783$. (d) Wiener-deblurred image using the full image frame; $\text{MSE} = 634$. The overlap in all cases is equal to half the section size in both the horizontal and vertical directions.

noise smoothing has occurred inside each section (see figure 5.2(c)). This is explained as follows: since the smaller regions are more stationary, their power spectral density estimates (required by the Wiener filter) are more accurate than those obtained from larger, less stationary sections, thereby leading to a more optimal Wiener filter. This, however, does not compensate for the edge artifacts that still exist after overlapping. The edge artifacts that degrade this version of the deblurred image are mainly due to the small size of the sections relative to the size of the region of support of the blurring PSF, which is a circular blur of diameter 7 pixels; i.e., the PSF is only 9 pixels smaller than the section size. This small section size means that most of the pixels in the section have contributions from outside the section convolved into the section by the PSF. Thus, deconvolving with incomplete information causes edge artifacts in a larger layer of border pixels, and even overlapping the sections cannot suppress them.

Figure 5.2(d) shows the image deconvolved using the full image frame (i.e., one section of size 128×128 pixels) and the Wiener filter. It is clear that edge effects do not exist inside the image boundaries because of the use of one large section (the full frame) having a size much larger than the region of support of the PSF. They do, however, exist at the boundary pixels for the same reasons as discussed above. By inspection of the restored image in figure 5.2(d), it is clear that less noise smoothing has occurred as compared to the restored image using 32×32 or 16×16 section sizes. This is because of the nonstationarity of the full frame image used to calculate the power spectral density required in the Wiener filter equation. Thus, the power spectral estimates are far from accurate and do not correctly represent the various image features, thereby causing the Wiener filter to deviate from optimality.

The FNSD method was also applied to the 128×128 pixel, 256 gray level "Camera Man" image after it was degraded by a 9-pixel horizontal motion blur PSF and noise

to 35 dB SNR. The original image is shown in figure 5.3(a) and its blurred noisy version in figure 5.3(b). Figure 5.3(c) shows one of the regions created by centering a 32×32 section of the blurred noisy image in a mean-padded region of the same size as the input image. Figure 5.3(d) is the windowed region corresponding to the section in figure 5.3(c). Figure 5.3(e) shows the restored image after using the FNSD algorithm with the sectioned Wiener filter of equation (5.14) and overlapping of the 32×32 sections by one-half the section size in each direction. Edge artifacts are apparent in this deblurred image. They are more pronounced around the vertical boundaries of the image than around the horizontal boundaries. This is due to the shape of the blur PSF, which is a one-dimensional function along the horizontal axis. Thus, convolution occurs in the horizontal direction only and the blurred image only has contributions from outside the vertical boundaries of the image, causing vertical edge artifacts during restoration. Edge artifacts are also more pronounced near edges than in flat regions of the image. This may be due to the small size of the sections and the large error due to the nonstationarity of the sections near edges.

The use of larger 64×64 overlapping sections was found to give slightly better visual results with less edge artifacts inside the image, as shown in figure 5.3(f). It is, however, clear that less noise smoothing is taking place due to the inaccurate power spectral estimates obtained when larger, less stationary sections are involved. Figure 5.3(g) shows the deblurred image obtained by using smaller overlapping sections of size 16×16 pixels and the same Wiener filter. The edge artifacts near the image edges are now more severe. This is directly related to the small size of the sections used with respect to the size of the PSF region of support, and the explanation given for the image in figure 5.2(c) also holds for this image. However, more noise smoothing in flat regions of the image is achieved with the 16×16 sections as compared to the

results with 32×32 and 64×64 sections. This is due to a more accurate stationary model obtained in flat regions when the section size is small enough to exclude sharp edges. Consequently, the power spectral estimates obtained in flat regions are more accurate and the Wiener filter performance is thus enhanced. Another explanation for the severe edge artifacts near sharp edges is the fact that at sharp edges even the smaller sections would include the edges, which would render the section less stationary and also cause the section pixels to have large differences with their mean value.

Figure 5.3(h) shows the image deconvolved using the full image frame and the Wiener filter. It is clear that edge effects do not exist inside the image boundaries because of the use of one large section (the full frame) having a size much larger than the region of support of the PSF. They do, however, exist at the boundary pixels for the same reasons discussed above. It is also clear that less noise smoothing has occurred as compared to the restored image using 32×32 or 16×16 section sizes, and the same explanation provided for the image in figure 5.2(d) holds for this image.

Similar results were obtained when using the FNSD algorithm with the sectioned PSE filter of equation (5.15) in place of the sectioned Wiener filter. Table 5.1 presents the MSE values between the original and deblurred images for both the Lena image and the Camera Man image and for various section sizes. The first column of the table gives the section size used and the next two columns give the MSE results for the Lena and Camera Man images deblurred using both the Wiener and the PSE filters. The MSE values are quite consistent with the visual results seen in the figures and discussed above. For the Lena image, which is blurred by the 7-pixel Gaussian blur PSF, using the full image frame for deblurring resulted in the highest MSE values (634 and 605), disregarding the results of the 16×16 sectioned case. This is mainly

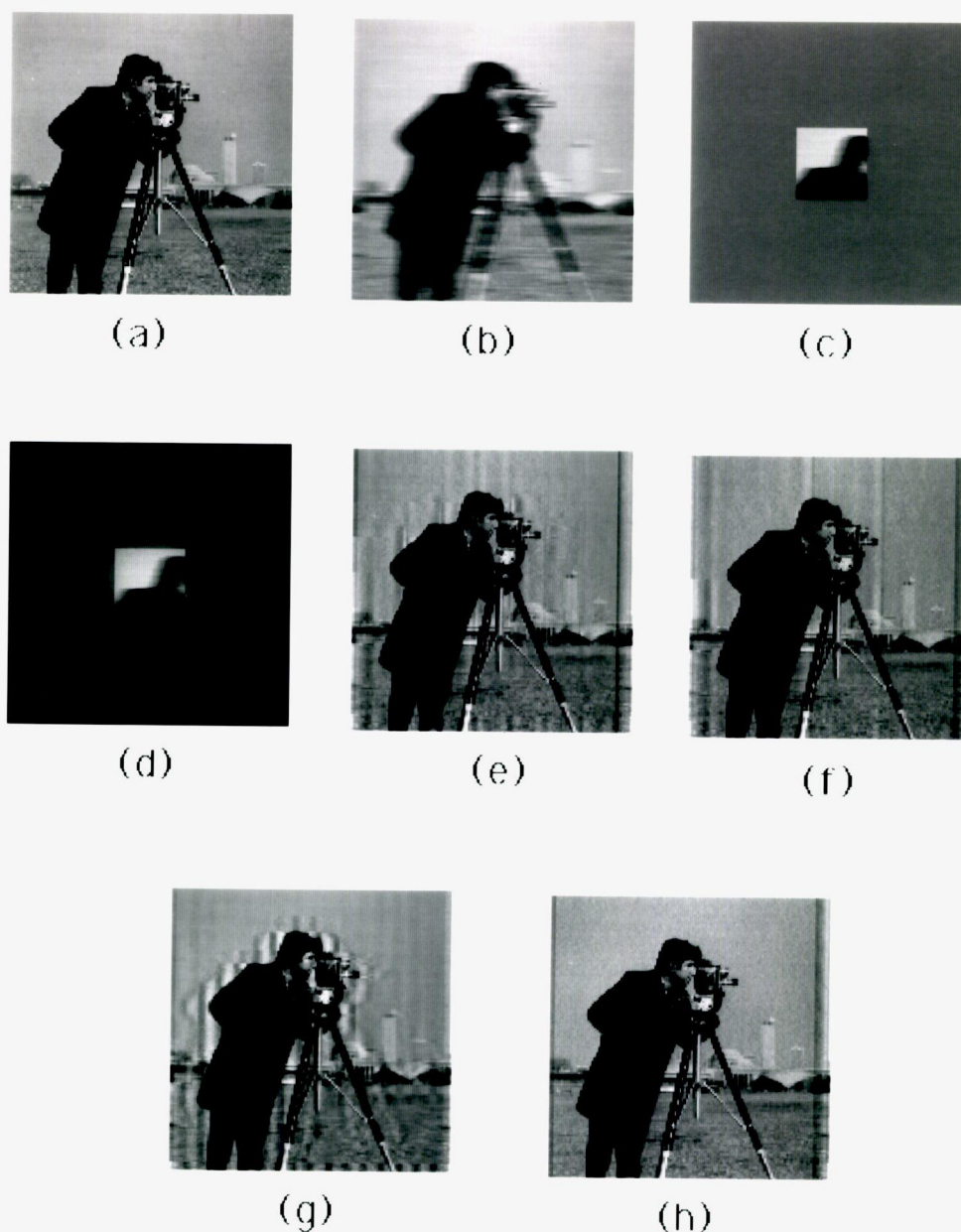


Figure 5.3 Sectioned deblurring results for the Camera Man image of size 128×128 pixels and a gray level range of $0 - 255$. (a) original image. (b) 9-pixel horizontal motion blurred image; $\text{MSE} = 1247$. (c) a sectioned mean-padded region created for sectioned deblurring. (d) the windowed equivalent of the region in (c). (e) Wiener-deblurred image using overlapped sections of size 32×32 pixels; $\text{MSE} = 424$. (f) Wiener-deblurred image using overlapped sections of size 64×64 pixels; $\text{MSE} = 463$. (g) Wiener-deblurred image using overlapped sections of size 16×16 pixels; $\text{MSE} = 539$. (h) Wiener-deblurred image using the full image frame; $\text{MSE} = 217$.

<i>Section Size Used in Deblurring</i>	<i>MSE</i>			
	<i>Lena</i>		<i>Camera Man</i>	
	Gaussian Blurred: 607		Motion Blurred: 1247	
	Wiener	PSE	Wiener	PSE
16 × 16	783	751	539	538
32 × 32	501	513	424	425
64 × 64	483	488	463	460
Full Frame	634	605	217	220

Table 5.1 Results of fixed-neighborhood sectioned deblurring on the Lena and Camera Man images of size 128×128 pixels and 256 gray levels for various section sizes and two different blurring functions.

due to the large errors in estimating the spectral density from the nonstationary full frame image, thereby causing less noise smoothing than the other smaller sections used. This situation is, however, reversed for the Camera Man image blurred by a 9-pixel horizontal motion blur PSF. The MSE values, when using the full frame image for deblurring, were the lowest (217 and 220). This is because of the severe vertical edge artifacts that accompany deblurring a 9-pixel horizontal, motion blurred image, using small sections. When using the full image frame for deblurring, no edge artifacts exist inside the image boundaries (they exist only around the outer image boundaries) as discussed earlier, and this compensates for the increased noise due to insufficient noise smoothing, which is caused by the nonstationarity of the full frame image. For this reason, deblurring horizontal motion blurs using the full image frame gives lower MSE values and visually better restored images although less noise is removed.

5.4 Summary

In this chapter, the fixed-neighborhood sectioned deblurring (FNSD) method was presented. Mathematical derivations and implementation of the algorithm with both the Wiener and the PSE filters were discussed in detail, and experimental results were presented with a comparative analysis of the performance and shortcomings of the sectioned filters. In the next chapter, the new adaptive-neighborhood deblurring (AND) filter will be presented in full detail, including experimental results. The performance of the AND filter will be evaluated in comparison with that of the sectioned filters discussed in this chapter.

CHAPTER 6

ADAPTIVE-NEIGHBORHOOD IMAGE DEBLURRING

6.1 Introduction

The main problems that face conventional, linear, space-invariant, deblurring techniques [1-5,38] such as the Wiener and PSE filters are the inherent nonstationarity of natural images, and edge effects that accompany the deblurring process. In an attempt to solve these problems, a fixed-neighborhood sectioned deblurring (FNSD) algorithm was presented in chapter 5. The algorithm was implemented by sectioning the input blurred noisy image into square sections of a fixed size $P \times P$ pixels and assuming the sections to be more accurate approximations to stationary sections of the image. This assumption allowed us to carry out the deblurring process using the standard, space-invariant, frequency domain Wiener and PSE filters. Each section was deblurred individually, and all restored sections were combined to form the final deblurred image. It was shown experimentally that the stationarity assumption for square sections is not really a reasonable assumption due to the fact that fixed-neighborhood sections are unable to distinguish between different image features, and thus tend to include edges as well as flat areas, which renders them effectively nonstationary.

In this chapter, we present a new approach to sectioned deblurring in which the relative pixel-to-pixel difference is used to grow relatively stationary regions in the

given image. Each region grows from a seed pixel (the current pixel being processed) and includes neighboring pixels having gray-level values that lie within a specified tolerance limit. The same region growing technique described in chapter 4 is also used here, with the same definition for the additive tolerance t_1 given by

$$t_1 \geq |y(i, j) - y(k, l)|, \quad (6.1)$$

where $y(k, l)$ is the seed pixel, and $y(i, j)$ represents pixels 8-connected to the seed pixel.

A new adaptive filter is derived in the next section, which is well suited to this type of region growing. The filter is applied separately to each input pixel using power spectral estimates calculated from the adaptive-neighborhood (AN) region grown from the seed pixel in an attempt to deblur that pixel. We call this new technique adaptive-neighborhood deblurring (AND).

To directly compare our new method with the FNSD method discussed in the previous chapter, we adopt the same technique of centering the ANs in the middle of a square region having the same size as that of the input image ($M \times M$), and padding the area surrounding the AN with its mean value. This helps reduce edge effects at the AN boundaries, and allows us to apply the two-dimensional fast Fourier transform (2DFFT) algorithm to these variable-shape, variable-size ANs.

6.2 The Adaptive-Neighborhood Deblurring Method

As discussed above, the AND algorithm which we introduce in this work is based on an AN region which is determined for each pixel in the input image. The degradation model for the input image $y(k, l)$ that was used in chapter 5 and is also used

here can be expressed as

$$y(k, l) = x(k, l) * h(k, l) + n(k, l), \quad (6.2)$$

where $x(k, l)$ is the original undegraded image which is unknown, and $n(k, l)$ is the additive noise process, which is assumed to be a zero-mean, white, Gaussian function of variance σ_n^2 . The restoration problem to be considered is to find an estimate of $x(k, l)$ given the PSF $h(k, l)$, the observed image $y(k, l)$, and the statistical properties of the noise process. In the frequency domain, equation (6.2) becomes

$$Y(u, v) = X(u, v) \cdot H(u, v) + N(u, v). \quad (6.3)$$

Assuming that each AN region grown is large compared to the region of support of the PSF, each AN region can be expressed as the convolution of the PSF with an equivalent AN region grown inside the original undegraded image $x(k, l)$. Thus, from equation (6.2),

$$y_{kl}(i, j) \simeq x_{kl}(i, j) * h(i, j) + n_{kl}(i, j), \quad (6.4)$$

where k and l are the seed pixel coordinates, and (i, j) give the locations of pixels that lie inside the current AN region. Equation (6.4) is the approximate description of the image formation law for each AN region $y_{kl}(i, j)$.

We now center each AN, $y_{kl}(i, j)$, inside a square region of the same size as the input image ($M \times M$), and pad the area surrounding the AN with its mean value to reduce edge effects and to enable us to use the TDFFT on each AN region grown. Thus, in the frequency domain, equation (6.4) becomes

$$Y_{kl}(u, v) = X_{kl}(u, v) \cdot H(u, v) + N_{kl}(u, v), \quad (6.5)$$

where $Y_{kl}(u, v)$ gives the Fourier transform of the AN region grown from the seed pixel at location (k, l) in the degraded image $y(k, l)$.

Then, applying the same two-dimensional Hamming window function $w_H(i, j)$ used with the FNSD method of chapter 5 to each mean-padded AN region, we obtain

$$y_{kl}(i, j) \cdot w_H(i, j) = [x_{kl}(i, j) * h(i, j)] \cdot w_H(i, j) + n_{kl}(i, j) \cdot w_H(i, j), \quad (6.6)$$

or

$$\tilde{y}_{kl}(i, j) \simeq \tilde{x}_{kl}(i, j) * h(i, j) + \tilde{n}_{kl}(i, j), \quad (6.7)$$

where the tilde (\sim) represents the corresponding windowed regions. The power spectrum of each region $y_{kl}(i, j)$ can be derived from equation (6.7) as

$$P_y^{kl}(u, v) \simeq |H(u, v)|^2 \cdot P_x^{kl}(u, v) + P_n^{kl}(u, v), \quad (6.8)$$

where the superscripts kl denote the AN region grown from the seed pixel at (k, l) . This power spectral density $P_y^{kl}(u, v)$ is computed from the Fourier transform of the windowed region $\tilde{y}_{kl}(i, j)$ using [46]

$$P_y^{kl}(u, v) = \frac{|\tilde{Y}_{kl}(u, v)|^2}{\sum_{i=0}^{M-1} \sum_{j=0}^{M-1} w_H^2(i, j)}. \quad (6.9)$$

In deriving the AND filter, we take into account the stationarity of the AN regions grown, and estimate the Fourier transform of the noise $N_{kl}(u, v)$, within the current AN region grown from the seed pixel at (k, l) , as a function of the Fourier transform of the AN region $Y_{kl}(u, v)$ as

$$\hat{N}_{kl}(u, v) = A_{kl}(u, v) \cdot Y_{kl}(u, v), \quad (6.10)$$

where $A_{kl}(u, v)$ is a frequency-domain, magnitude-only scale factor which depends on the spectral characteristics of the AN region grown. The estimate of $X_{kl}(u, v)$ is obtained from equation (6.5) by using the spectral estimate of the noise, $\hat{N}_{kl}(u, v)$, in place of $N_{kl}(u, v)$. We thus have

$$\hat{X}_{kl}(u, v) = \frac{Y_{kl}(u, v) - \hat{N}_{kl}(u, v)}{H(u, v)}, \quad (6.11)$$

which reduces to

$$\hat{X}_{kl}(u, v) = \frac{Y_{kl}(u, v)}{H(u, v)} \cdot [1 - A_{kl}(u, v)]. \quad (6.12)$$

The problem now is to find the spectral noise estimator $A_{kl}(u, v)$, based on the criterion that the power spectral density of the estimated noise $P_n^{kl}(u, v)$ be equal to the original noise power spectral density $P_n^{kl}(u, v)$ for the current frequency-domain AN region $Y_{kl}(u, v)$. The solution is derived from equation (6.10) as

$$P_n^{kl}(u, v) = A_{kl}^2(u, v) \cdot P_y^{kl}(u, v). \quad (6.13)$$

From equation (6.8) and equation (6.13), the spectral noise estimator $A_{kl}(u, v)$ is thus given by

$$A_{kl}(u, v) = \left(\frac{P_n(u, v)}{|H(u, v)|^2 \cdot P_x^{kl}(u, v) + P_n(u, v)} \right)^{1/2}, \quad (6.14)$$

where $P_n^{kl}(u, v) = P_n(u, v) = \sigma_n^2$ for Gaussian white noise. The quantity in equation (6.8) gives the denominator in equation (6.14). Therefore, no additional information is required about the power spectral density of the original undegraded image.

The frequency-domain estimate of the uncorrupted AN region is obtained by using the value of $A_{kl}(u, v)$, computed from equation (6.14), in equation (6.12). The spectral estimate of the original undegraded AN region is thus given by

$$\hat{X}_{kl}(u, v) = \frac{Y_{kl}(u, v)}{H(u, v)} \cdot \left[1 - \left(\frac{P_n(u, v)}{|H(u, v)|^2 \cdot P_x^{kl}(u, v) + P_n(u, v)} \right)^{1/2} \right], \quad (6.15)$$

and the space-domain estimate of the uncorrupted AN region $\hat{x}_{kl}(i, j)$ is obtained from the inverse Fourier transform of the frequency-domain estimate of the AN region $\hat{X}_{kl}(u, v)$ given in equation (6.15).

Now, by replacing the seed pixel at (k, l) with the deblurred pixel $\hat{x}_{kl}(k, l)$ (which is the pixel obtained from the space-domain estimate of the uncorrupted AN region

$\hat{x}_{kl}(i, j)$, by replacing (i, j) with the seed pixel location (k, l) , and running the above algorithm for every pixel in the input image, we will eventually obtain a deblurred image based on stationary AN regions.

The only disadvantage of the above algorithm is the fact that it requires three 128×128 TDFFT operations per pixel, which is about 10 seconds per pixel on the fast SUN/Sparc-2 workstations that were used for processing the images in this work. For 128×128 pixel images, at 10 seconds per pixel, we require about 45.5 hours to process a single image, which is impractical. A solution to this problem is made possible by the nature of the AN regions grown: Since most of the pixels inside an AN region will have similar regions when they become seed pixels (because they lie within similar limits of tolerance), instead of growing AN regions for every single pixel in the input image, we grow AN regions only from those pixels that do not already belong to a previously grown region. Note that the AN regions so grown would overlap in an unspecified manner. This method reduces the processing time to about 10 to 15 minutes per image (of size 128×128 pixels).

6.3 Experimental Results and Discussion

In this section, some experimental results which illustrate the performance of the AND algorithm are presented. The results are evaluated by direct visual examination and by calculation of the MSE between the known original and deblurred images.

The AND algorithm was first applied to the 128×128 pixel, 256 gray level Lena image after it was degraded by a Gaussian-shaped blur PSF with a radial standard deviation $\sigma_r = 3$ pixels and noise to 35 dB SNR. The original image is shown in figure 6.1(a) and its blurred, noisy version is shown in figure 6.1(b). Figures 6.1(c,d,e) show three different AN foreground regions of the blurred image in figure 6.1(b). Each

AN region was allowed to grow to any size as long as the pixel values were within the limits of the tolerance t_1 defined in equation (6.1). This means that the value of the foreground bound, Q , (described in chapter 4) was set to the total number of pixels in the given image (i.e., $Q = 16384$ for a 128×128 image). The tolerance used to grow AN regions for the AND algorithm was set to $\frac{\sigma_g}{\sqrt{2}}$, where σ_g is an estimate of the standard deviation of the noise-free blurred image $g(k, l) = x(k, l) * h(k, l)$. This value was arrived at using trial-and-error, and gave optimum results. Each AN region was centered in a square region of the same size as the full image (128×128) and the surrounding area was padded with its mean value as before. It is very clear from figures 6.1(c,d,e) that the stationarity assumption holds for the mean-padded regions. Also, the mean value is very similar to the AN region pixel values, and thus the mean also lies within the same limits of the tolerance used to grow the AN regions.

A closer look at figures 6.1(c) and 6.1(d) will indicate that overlapping of the two AN regions has occurred although the AN region in figure 6.1(c) was grown with the seed pixel at location (0,16) in the input image and the AN region in figure 6.1(d) was grown with the seed pixel at (0,106). Such overlap between AN regions will aid in suppressing edge artifacts which may arise at AN region borders. Figure 6.1(f) is the windowed version of the region in figure 6.1(e) where a Hamming window was used as described in the previous section. All the regions that were created for AND were windowed in the same manner as that of figure 6.1(f).

Figure 6.2(a) shows the restored image after applying the AND filter of equation (6.15) to the blurred noisy image of figure 6.1(b). The FNSD-restored image of figure 5.2(a) using non-overlapping sections of size 32×32 pixels is shown again in figure 6.2(b); the FNSD-restored image of figure 5.2(c) using overlapping sections of size 16×16 pixels is shown again in figure 6.2(c); and the FNSD-restored image of

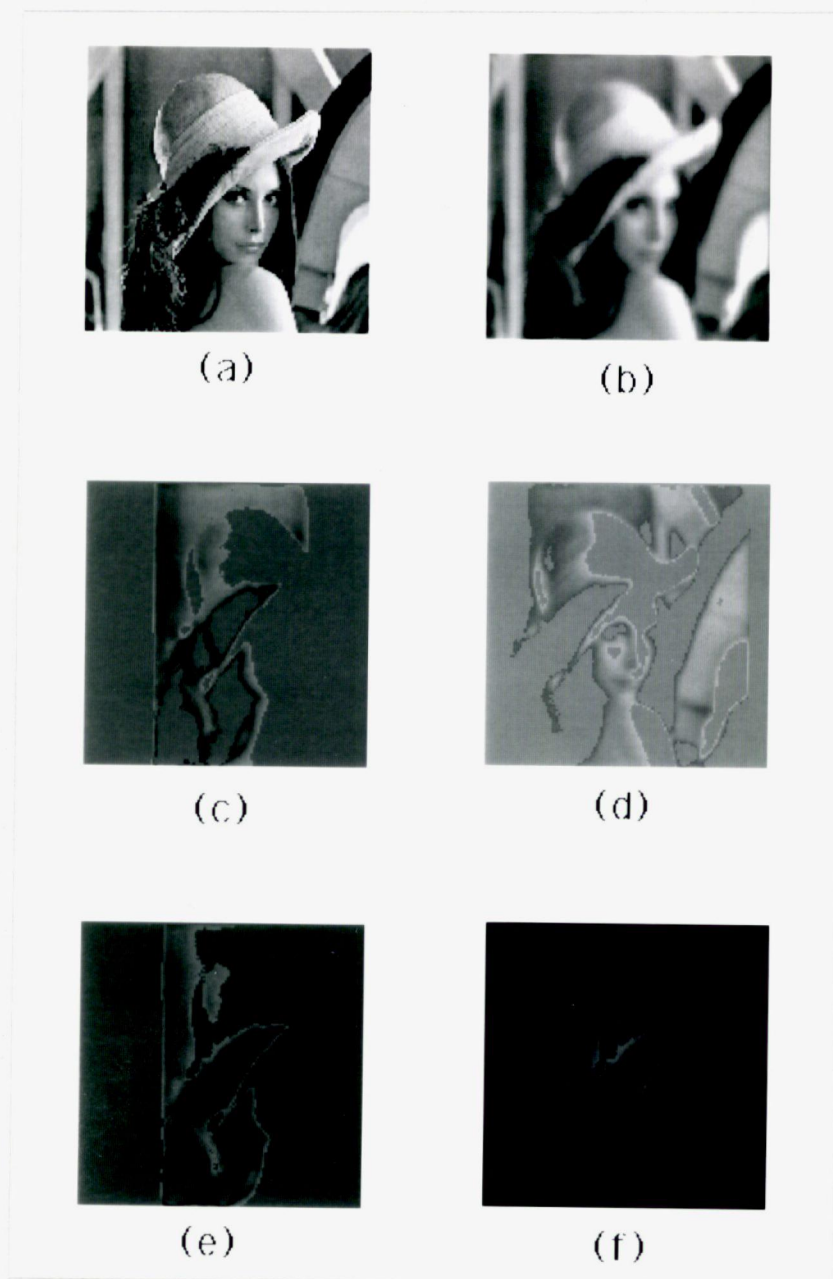


Figure 6.1 Adaptive-neighborhood segmentation of the Lena image of size 128×128 pixels and gray level range of 0 – 255. (a) original image. (b) blurred image with a Gaussian-shaped blur function and noise to 35 dB SNR; MSE = 607. (c), (d), and (e) show three AN mean-padded regions created for AND. (f) the windowed equivalent of the region in (e).

<i>Filter</i>	<i>Neighborhood Size Used</i>	<i>Time</i>	<i>MSE</i>	
			<i>Lena, SNR = 35dB</i> Gaussian Blurred: 607	<i>Camera Man, SNR = 35dB</i> Motion Blurred: 1247
Wiener	16×16	25 min	783	539
PSE	16×16		751	538
Wiener	32×32	10 min	501	424
PSE	32×32		513	425
Wiener	64×64	5 min	483	463
PSE	64×64		488	460
Wiener	Full Frame	2 min	634	217
PSE	Full Frame		605	220
AND	$Q = 16384$	15 min	292	181

Table 6.1 Results of fixed- and adaptive-neighborhood deblurring on the Lena and Camera Man images of size 128×128 pixels and 256 gray levels for various neighborhood sizes and two different blurring functions.

figure 5.2(b) using overlapping sections of size 32×32 pixels is shown again in figure 6.2(d) for comparison. From the AND-restored image shown in figure 6.2(a), we can see that almost no edge artifacts have occurred. This is directly related to the AN regions being used, and is also aided by the inherent overlapping of the AN regions grown inside the input image. Comparing the AND-restored image of figure 6.2(a) with the FNSD-restored images of figures 6.2(b,c,d) shows the superiority of the AND restoration technique to the FNSD technique. Also, by inspection of the MSE values (shown in table 6.1) between the original undegraded image of figure 6.1(a) and the restored images, we find that the AND-restored image has an MSE value of only 292, which is much lower than the MSE values of the FNSD-restored images (483, 501, 634, 783).

The AND filter was also applied to the 128×128 pixel, 256 gray level Camera Man image after it was degraded by a 9-pixel horizontal motion blur PSF and noise to 35 dB SNR. The original image is shown in figure 6.3(a) and its blurred noisy

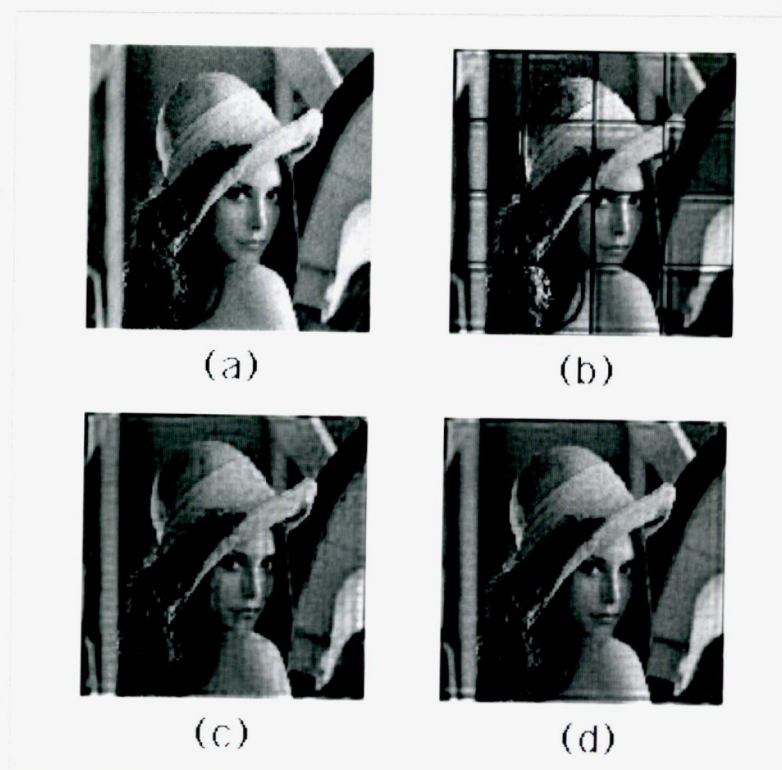


Figure 6.2 Adaptive-neighborhood deblurring (AND) of the Lena image of size 128×128 pixels and gray level range of $0 - 255$. (a) deblurred image using the AND filter; $MSE = 292$. (b) deblurred image using FNSD with non-overlapping sections of size 32×32 pixels; $MSE = 1255$. (c) FNSD Wiener-deblurred image using overlapped sections of size 16×16 pixels; $MSE = 783$. (d) FNSD Wiener-deblurred image using overlapped sections of size 32×32 pixels; $MSE = 501$.

version is shown in figure 6.3(b). Figure 6.3(c) shows one of the regions created by centering an AN region (grown from a seed pixel in the blurred noisy image) in a mean-padded region of the same size as that of the input image. It is very clear that the AN region approximates a stationary region in the blurred image. It is also clear that the mean value is similar to the pixel values in the AN region, thus causing the whole region to be stationary. Figure 6.3(d) is the Hamming-windowed version of another AN region grown from a seed pixel outside the AN region of figure 6.3(c). Figure 6.3(e) shows the image restored by applying the AND filter of equation (6.15) to the blurred noisy image of figure 6.3(b). The FNSD-restored image of figure 5.3(e) using overlapping sections of size 32×32 pixels is shown again in figure 6.3(f) for comparison. By inspection of the AND-restored image shown in figure 6.3(e), it is evident that practically no edge artifacts have occurred during deblurring with the AN regions. The image is very sharp and visually more pleasing as compared to the FNSD-restored image shown in figure 6.3(f). The same explanation used for the Lena AND-restored image of figure 6.2(a) also holds for the Camera Man image. Also, by inspection of the MSE values shown in table 6.1, we find that the Camera Man AND-restored image has an MSE value of only 181, which is significantly lower than the MSE values of the FNSD-restored images (217, 424, 463, 539).

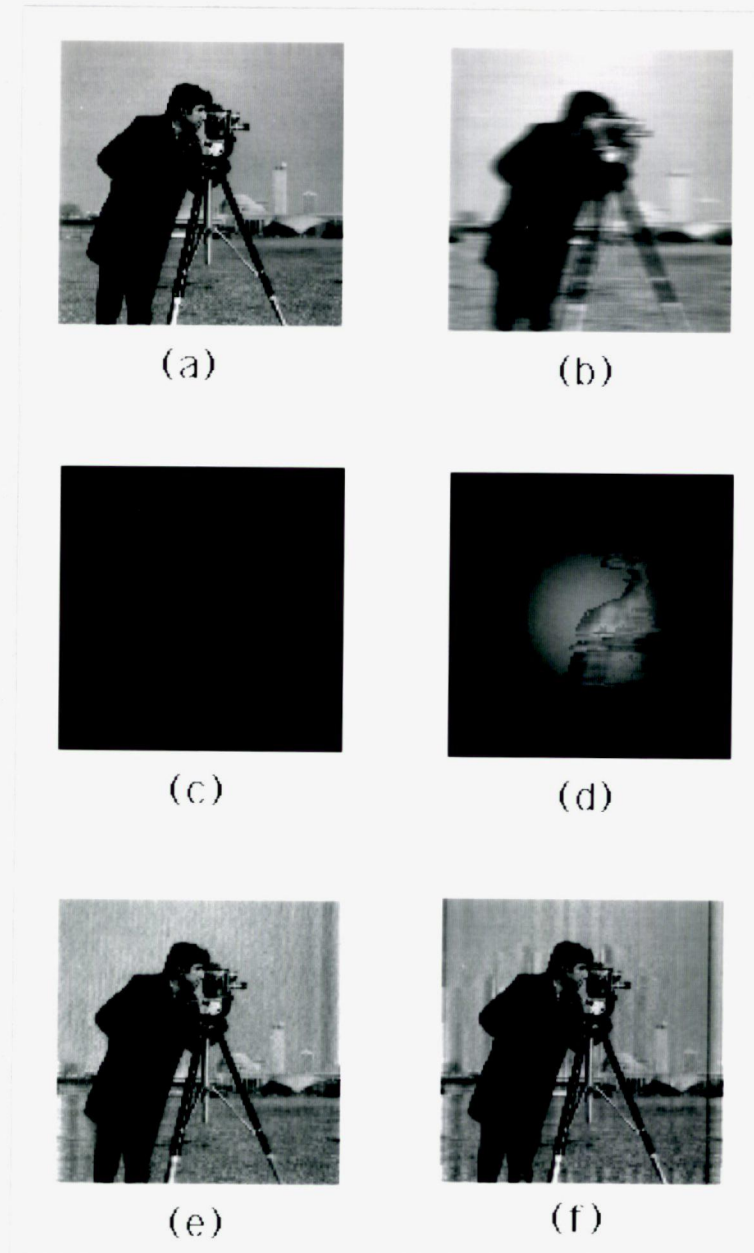


Figure 6.3 Adaptive-neighborhood deblurring (AND) of the Camera Man image of size 128×128 pixels and a gray level range of 0 – 255. (a) Original image. (b) Image blurred by 9-pixel horizontal motion and degraded by additive Gaussian noise to 35 dB SNR; MSE = 1247. (c) An AN mean-padded region created for AND. (d) A windowed version of another stationary AN region. (e) Deblurred image using the AND filter; MSE = 181. (f) FNSD Wiener-deblurred image using overlapped sections of size 32×32 pixels; MSE = 424.

6.4 Summary

In this chapter, the new AND filter was presented. This new algorithm uses ANs to define stationary regions in the input image and obtains a spectral estimate of the noise in the AN region. This estimate is then used to obtain a spectral estimate of the original undegraded AN region, which is inverse Fourier transformed to obtain the space-domain deblurred AN region. The regions are then combined to form the final restored image. Mathematical derivation and implementation of the AND filter were discussed, and experimental results were presented with an analysis of the performance of the AND filter as compared to that of the FNSD Wiener and PSE filters discussed previously in chapter 5. Using the AND algorithm for image deblurring, we were able to create fairly stationary regions. This improved the restoration process and produced results that were superior to those obtained using the FNSD method both in terms of qualitative (visual) and quantitative (MSE) measures.

CHAPTER 7

AN ITERATIVE METHOD FOR BLIND DECONVOLUTION

7.1 Introduction

The problem of separating two signals that have been convolved has received considerable attention in the literature [1-7,12,15,17,18,26,35-45,50]. It is important to distinguish between the two different forms of this problem. The first and simpler of these forms assumes that one of the two signals is known. The problem then reduces to estimating the other signal by some form of inverse filtering. The second, and more complex problem, considers that both signals are unknown, and that the only data available are those from the convolution itself. The task of estimating the unknown signals in the second case is called blind deconvolution.

A problem frequently encountered in image processing is the restoration of an image after it has been degraded by a linear, shift-invariant (LSI) system, without any knowledge about the degrading phenomenon. LSI blurring can occur in many ways in an actual image acquisition system, including defocused camera lenses, relative motion between the object and the recording plane, and imaging through turbulent media. Previous attempts to address this problem [1,5,15,35] estimate the blur function by first attempting to ascertain some statistical properties of the image by averaging over an ensemble of images. Since, however, only one blurred image is

usually available, an artificial ensemble must be created by subdividing the available image into a number of subimages and then averaging the log of their spectral magnitudes in an attempt to reinforce the shift-invariant property of the blur PSF, while averaging out the nonstationary frequency content of the original image. In order for the subdivision to be valid, the blur PSF must have a region of support much smaller than the size of the subimages. As a result, the size of the subimages cannot be made arbitrarily small. Thus, the number of subimages used to form the ensemble average is limited, and consequently the variance of the spectral estimate from the subimages can be relatively high. This leads to a poor estimate of the original image.

In this chapter, we present a new technique for blind deconvolution under the assumption that the MTF of the LSI system is zero-phase and its magnitude is smooth. This assumption is valid for many types of degradation, such as motion blur and out-of-focus blur [5,14]. As discussed in the literature [14,51-58], the spectral magnitude in the Fourier representation of a signal is affected by the blur function, while many of the important features of the signal, such as edge locations, are preserved in the phase. For example, it has been shown that the intelligibility of a sentence is retained if the phase of the Fourier transform of a long segment of the speech signal is combined with unity magnitude [14]. In our work, we make use of the image characteristics (edge information) preserved in the phase of the blurred image, and try to recover the original magnitude spectrum that was altered by the blur function.

The blind deconvolution algorithm presented here differs from earlier work [5,15,35] in that the averaging is achieved by smoothing in the frequency plane using the entire image, which relaxes the assumption on the region of support of the PSF. Another key feature of our algorithm is that further enhancement on the initial estimate of the image is achieved through an iterative approach.

In the next section, we present an iterative method for recovering the magnitude spectrum of a blurred image using an enhanced phase-based estimate. Next, we present experimental results from the application of this technique to different types of images. Finally, we evaluate these results and the different implications of this new technique.

7.2 The Iterative Blind Deconvolution Method

A commonly used model for image degradation is

$$y(k, l) = x(k, l) * h(k, l) + n(k, l), \quad (7.1)$$

where $x(k, l)$ represents the (unknown) original image, $h(k, l)$ represents the PSF (unknown in the situation considered here), $n(k, l)$ represents the additive noise component, and $y(k, l)$ is the given blurred image.

In this work, we neglect the additive noise component. Therefore, in the frequency domain, equation (7.1) can be expressed as

$$Y(u, v) = X(u, v) \cdot H(u, v). \quad (7.2)$$

Representing equation (7.2) in the form of a magnitude and phase function, and taking into account that $H(u, v)$ is assumed to be zero-phase, we can write equation (7.2) as

$$M_y(u, v) = M_x(u, v) \cdot M_h(u, v), \quad (7.3)$$

and

$$\theta_y(u, v) = \theta_x(u, v), \quad (7.4)$$

where $M_x(u, v)$ represents the spectral magnitude of the original image, $M_h(u, v)$ represents the degradation MTF and has the property that it is a smooth magnitude function, and $\theta_i(u, v)$ represents the spectral phase of the image i .

The blur model is thus defined in terms of magnitude and phase. The original image spectral magnitude $M_x(u, v)$ can be recovered from the blurred image spectral magnitude $M_y(u, v)$ as follows. The initial estimate of $M_x(u, v)$ is based on smoothing the spectral magnitude of the blurred image, $M_y(u, v)$, and using the assumption that $M_h(u, v)$ is smooth. If we let $S[\cdot]$ denote a two-dimensional linear, separable, smoothing operator, then a smoothed $M_y(u, v)$ is given by

$$S[M_y(u, v)] = S[M_x(u, v) \cdot M_h(u, v)]. \quad (7.5)$$

Since $M_h(u, v)$ is a smooth function, and $S[\cdot]$ is separable, $S[M_x(u, v) \cdot M_h(u, v)]$ may be approximated by $S[M_x(u, v)] \cdot M_h(u, v)$ [14,50], and therefore

$$S[M_y(u, v)] \simeq S[M_x(u, v)] \cdot M_h(u, v). \quad (7.6)$$

Combining equation (7.3) and equation (7.6), we obtain

$$M_x(u, v) \simeq M_y(u, v) \cdot \frac{S[M_x(u, v)]}{S[M_y(u, v)]}. \quad (7.7)$$

Equation (7.7) suggests that if we can obtain an initial approximation to $M_x(u, v)$ we can rewrite equation (7.7) in an iterative form and use it to refine the initial magnitude estimate. Equation (7.7) can thus be written in an iterative form as

$$\widehat{M}_x^{i+1} = M_y \cdot \frac{S[\widehat{M}_x^i]}{S[M_y]}, \quad (7.8)$$

where i is the iteration number, and (u, v) have been dropped for compact notation.

The initial estimate \widehat{M}_x^0 is derived from the phase of the blurred image $e^{j\theta_y(u, v)}$ which retains most of the high-frequency information (spatial edges) in the image. The initial estimate of the original image is defined in the frequency domain to be the sum of the phase of the blurred image and the Fourier transform of the blurred image itself. We thus have

$$\widehat{M}_x^0 \cdot e^{j\theta_y} = M_y \cdot e^{j\theta_y} + e^{j\theta_y}, \quad (7.9)$$

and in terms of magnitudes,

$$\widehat{M}_x^0 = M_y + 1. \quad (7.10)$$

As we can see from equation (7.10), we are only adding a unity constant to the spectral magnitude at all frequencies, which would only raise the whole spectral magnitude response by the amount of that constant. This is equivalent to adding a high-pass filtered version of the image to the blurred image, which would have the effect of amplifying the high-frequency components in the image. This would, however, produce a noisy initial approximation to M_x due to some added high-frequency noise. Rather than simply adding a unity magnitude to recover high frequencies, it would be more appropriate to add those high-frequency components in M_x that were lost due to blurring. Although we do not have M_x , we do have a ratio between M_x and its smooth version derived from equation (7.7), giving

$$\frac{M_x}{S[M_x]} \simeq \frac{M_y}{S[M_y]}. \quad (7.11)$$

Adding this ratio to the blurred magnitude spectrum gives

$$\widehat{M}_x^0 = M_y + \frac{M_x}{S[M_x]}. \quad (7.12)$$

We can expect this estimate to be a more accurate approximation to M_x since we are now adding the ratio $\frac{M_x}{S[M_x]}$ which contains more information about the original spectral magnitude than a simple constant (1 in equation (7.10)).

The advantages of adding $\frac{M_x}{S[M_x]}$ to the blurred magnitude spectrum are:

- *At low frequencies*, $M_x \simeq S[M_x]$. Therefore, $\widehat{M}_x^0 \simeq M_y + 1$, which would not affect the low-frequency magnitude response much.
- *At higher frequencies*, $S[M_x] < M_x$ at high amplitudes of M_x , because any variance in M_x would be averaged out in $S[M_x]$. Thus $\frac{M_x}{S[M_x]} > 1$ at those high-

frequency components representing spatial edges. Adding this to the blurred magnitude would have the effect of amplifying the high-frequency edges more than the high-frequency noise. Therefore, the high-frequency components of the phase signal $e^{j\theta_y}$ are emphasized in $\frac{M_x}{S[M_x]} \cdot e^{j\theta_y}$, which we call the *enhanced phase* image. Thus the operation in equation (7.12) tends to add to M_y the (normalized) high-frequency components that were lost because of blurring.

Using equation (7.12) as the initial estimate of M_x and iterating using equation (7.8) to improve this estimate until the MSE between the n^{th} estimate and the $(n+1)^{th}$ estimate is less than a certain error ϵ , we would eventually recover most of the original spectral magnitude of M_x .

From equation (7.4), we can now combine the best estimate of $M_x(u, v)$ with the given phase function $\theta_x(u, v) = \theta_y(u, v)$ so that the Fourier transform of the restored image $\hat{X}(u, v)$ is

$$\hat{X}(u, v) = \widehat{M_x}(u, v) \cdot e^{j\theta_x(u, v)}. \quad (7.13)$$

Taking the inverse Fourier transform of equation (7.13) we obtain the deblurred image $\hat{x}(k, l)$.

Although the method described above neglects noise, it can still be used for deblurring images corrupted by blur and additive noise after first reducing the noise in the blurred image [12,50] using a noise reduction filter such as the Wiener smoothing filter. Further work will be required in order to investigate the most effective approach to noise reduction before applying the iterative procedure.

7.3 Experimental Results and Discussion

In this section we present some experimental results which illustrate the performance of the new iterative blind deblurring method described above. The results will be evaluated by direct visual examination and by calculation of the MSE between the original image (known in the experiments) and the restored images.

Figure 7.1 presents results using a natural scene. Figure 7.1(a) shows a 128×128 , 256 gray level image of the original Lena face. Figure 7.1(b) is the image in figure 7.1(a) blurred by a Gaussian-shaped PSF with a radial standard deviation $\sigma_r = 3$ pixels. The MSE between the original image and the blurred image is 606. After applying the iterative method to the blurred image of figure 7.1(b), the images in figures 1(c,d,e,f) were generated. Figure 7.1(c) shows the magnitude of the inverse Fourier transform of the enhanced phase $\frac{M_x}{S[M_x]} \cdot e^{j\theta_y}$. It is very clear that this enhanced phase image retains most of the edges in the original image that were destroyed by blurring. Figure 7.1(d) is the initial estimate of $\hat{x}(k, l)$ used in the iterative algorithm. This image was formed by addition of the image in figure 7.1(b) to the image in figure 7.1(c) as described in equation (7.12). The addition emphasizes the edges of the blurred image thus giving a sharper image (the MSE of this initial estimate, at 877, is however higher than that of the blurred image). One can also notice from figure 7.1(d) that the dynamic range of the image is different from that of the original image in figure 7.1(a), and that it could be improved by iteration. Figure 7.1(e) shows the image generated by combining the first estimate \widehat{M}_x^1 obtained by using equation (7.8) with the phase of the blurred image $e^{j\theta_y}$. It is very clear that even after just one iteration, the deblurred image closely resembles the original without any blurring of the edges. The MSE for the image in figure 7.1(e) is only 128, which is a significant improvement over the previous estimate. After only four iterations

through the filter, we obtain the image in figure 7.1(f), with an MSE of 110. Further iterations did not improve the quality of the restored image beyond that in figure 7.1(f). The restored image in figure 7.1(f) is very sharp, with most of the high-frequency components recovered by only using the information in the blurred image and its phase, and applying the iterative algorithm described in the previous section.

Figure 7.2 is a different set of images used to experiment with the restoration of blurred text. The images are of size 64×64 pixels with 256 gray levels. Figure 7.2(a) is the original image representing a portion of an address. The text is bright on a dark background. Figure 7.2(b) is the image in figure 7.2(a) blurred by the same blur function used in figure 7.1. The MSE between the original image and the blurred image is 1041, which gives an indication of how much the blurring has altered the original image. It is clear from figure 7.2(b) that all of the sharp edges in figure 7.2(a) have been smoothed out, corresponding to a loss of some of the high frequencies from the Fourier transform of the original image. Figure 7.2(c) shows the magnitude of the inverse Fourier transform of the enhanced phase, described above, which shows most of the important information, namely the edges. Adding the images in figures 7.2(c) and 7.2(b) produces the initial estimate shown in figure 7.2(d) with an MSE of 385. Figure 7.2(e) is the final restored image reached after 8 iterations, with the initial estimate as shown in figure 7.2(d). The MSE between the restored image and the original further dropped to only 156, and by visual examination, it is clear that most of the lost information (including edges) has been recovered.

In figure 7.3 we show a set of images similar to those in figure 7.2, but with the original image more severely blurred by a Gaussian-shaped PSF of radial standard deviation $\sigma_r = 5$ pixels, as shown in figure 7.3(a), and as indicated by the much higher MSE of 3275. Clearly, this blurred image is totally illegible with all the words



Figure 7.1 Iterative blind deconvolution with the Lena image of size 128×128 pixels. (a) original image, (b) blurred image with Gaussian-shaped blur function of radial standard deviation $\sigma_r = 3$ pixels; $\text{MSE} = 606$, (c) enhanced phase image, (d) initial estimate image; $\text{MSE} = 877$, (e) deblurred image after first iteration; $\text{MSE} = 128$, (f) deblurred image after 4 iterations; $\text{MSE} = 110$.



Figure 7.2 Iterative blind deconvolution results for the slightly blurred text image of size 64×64 pixels and 256 gray levels. (a) original image, (b) slightly blurred image with Gaussian-shaped blur function of radial standard deviation $\sigma_r = 3$ pixels; MSE = 1041, (c) enhanced phase image, (d) initial estimate image; MSE = 385, (e) final restored image after 8 iterations; MSE = 156.

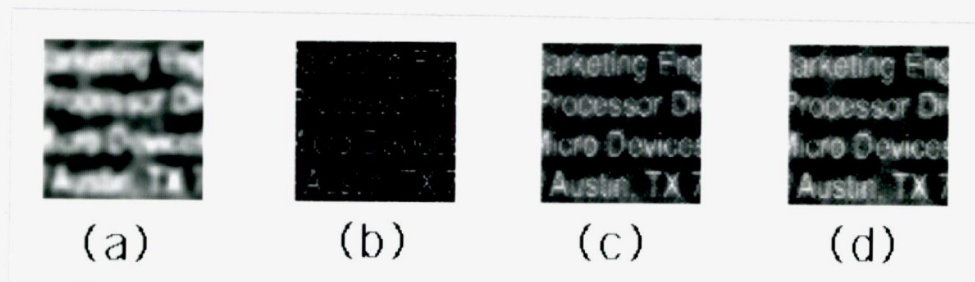


Figure 7.3 Iterative blind deconvolution with severely blurred text image of size 64×64 pixels and 256 gray levels. (a) highly blurred image of figure 7.2(a) with Gaussian-shaped blur function of radial standard deviation $\sigma_r = 5$ pixels; MSE = 3275, (b) enhanced phase image, (c) initial estimate image; MSE = 690, (d) final restored image after one iteration; MSE = 618.

extremely blurred. Figure 7.3(b) is the corresponding space domain enhanced phase image, and figure 7.3(c) is the initial estimate image, which is much more legible than the blurred image in figure 7.3(a). The MSE between the original image and the initial estimate is 690. Finally, after only one iteration, the restored image shown in figure 7.3(d) was obtained with an MSE of 618. Although the MSE value of the final restored image has not decreased much from that of the initial estimate, the image is satisfactory, and very legible compared to the blurred image in figure 7.3(a).

It is very clear from the results shown that this iterative method does not cause edge-effects during deconvolution, as did the other methods described in previous chapters. This is mainly because deconvolution is carried out by detecting the high-frequency information lost due to blurring, and adding it back to the spectrum of the blurred image. In essence, we are not deblurring by convolving the image (or subimages as in the previous chapters) with the inverse of the blur PSF, but instead, we are refining the initial estimate through iteration over the whole image.

7.4 Summary

In this chapter, a new blind deconvolution algorithm was presented. The new method creates an initial estimate using an enhanced phase signal, and an iterative algorithm is used to refine the initial estimate making use of the fact that for zero-phase blur functions, the edge information is preserved in the phase of the blurred image and only the magnitude of the original image is altered. Finally, we presented some experimental results by applying our new technique to successfully deblur two types of images degraded by Gaussian-shaped blur functions.

CHAPTER 8

DISCUSSION AND CONCLUSIONS

This thesis presented various image restoration techniques, both conventional and original, that deal with the restoration of nonstationary images. It first started by giving an account of the most commonly used methods for restoring images degraded by additive white Gaussian noise. Noise reduction methods were divided into space domain and frequency domain methods. Four space domain methods were discussed. These methods mainly smooth out the noise in the degraded image, such as in neighborhood averaging discussed in chapter 2, where square windows are used to approximate stationary regions and obtain estimates of the signal and noise statistics. It was concluded from chapter 2 that using fixed neighborhoods to approximate stationary statistics does not give accurate statistical estimates, and causes smoothing of high-frequency information in the degraded image.

Images are at best locally stationary, and then only if the windows or neighborhoods used contain pixels that belong to the same statistical ensemble. For most fixed neighborhoods stationarity is still a very crude approximation, which results in poor statistical estimates and hence poor restoration. Based on these observations, we presented three different adaptive noise-reduction filters in an attempt to overcome problems due to the inherent nonstationarity of images. In chapter 3, two adaptive noise-reduction filters (ATD-LMS and ARW-LMS) were presented, and mathematical

derivations and implementation of both filters were discussed in detail. The ATD-LMS filter has the problem of too much edge smoothing causing the restored images to appear blurred. The ARW-LMS filter performs much better from a visual perspective because flat regions are smoothed more than edge regions, thus preserving the sharpness of the image. This however, leaves much of the noise unsuppressed near edges. Experimental results were also presented with a comparative analysis of the performance and shortcomings of these adaptive filters.

In chapter 4, the new ANNS filter was presented in an attempt to overcome the problems of the previous two filters discussed in chapter 3. This new algorithm uses adaptive neighborhoods (ANs) to define stationary regions, and obtains an estimate of the noise in each region as a function of the input image region. This noise is then subtracted to obtain the restored image. The advantage of using these variable-shape, variable-size ANs over the adaptive rectangular windows (ARWs) is the fact that the AN size is not compromised near edges; thus a more accurate statistical estimate is possible, and more noise reduction takes place. Experimental results were presented with a comparative analysis between the performance of the ANNS algorithm and the performance of the ARW-LMS and ATD-LMS algorithms. The ANNS algorithm was shown to produce better output images than the other two algorithms in terms of both visual quality and MSE. The adaptive window techniques (ANNS and ARW-LMS) were further applied in a two-pass procedure, and the ANNS method was again shown to be superior with extremely low MSE values. The ANNS method is capable of noise filtering without blurring and without leaving noise around edges and sharp features in the given image. The method is also adaptive, and takes into account nonstationarities across the image field.

The thesis then shifted the focus to restoring images degraded by both a linear shift-invariant blur PSF, and additive, white Gaussian noise. Chapter 5 discussed conventional deblurring techniques, and then a fixed-neighborhood sectioned deblurring (FNSD) algorithm was derived and implemented. In this method, the input image is sectioned into square subimages, and each subimage is deblurred individually. In doing so, we make the assumption that each section is presumably stationary. We later showed, however, that this is not an accurate assumption, and consequently, the spectral estimates of the sections are inaccurate; the deblurred images using this method suffered from edge effects as shown by the experimental results. To solve this problem, we presented a new adaptive-neighborhood deblurring (AND) algorithm in chapter 6. This method uses AN region growing techniques to define stationary regions in the input image, and obtains a spectral estimate of the noise in the AN region. This estimate is then used to obtain a spectral estimate of the original undegraded AN region, which is inverse Fourier transformed to obtain the space-domain deblurred AN region. The regions are then combined to form the final restored image. Mathematical derivation and implementation of the AND filter were discussed, and experimental results showed an improvement over the FNSD method from a visual perspective, as well as in MSE.

The last part of this thesis was presented in chapter 7, and dealt with a new iterative algorithm for blind deconvolution. The new filter uses the information preserved in the phase of the blurred image to recover, as much as possible, the high-frequency components that were lost due to blurring. Experimental results of applying this new technique to two types of images degraded by Gaussian-shaped blur PSFs were presented.

From this discussion, one can conclude that image stationarity plays an important role in the overall performance of restoration filters. It was shown that the use of stationary parameters computed from AN regions improved the filter performance and produced output results that were superior to those obtained using fixed or adaptive square neighborhoods. Although this thesis gave an extensive analysis of AN image restoration techniques, and showed their relative merit as compared to other methods, one cannot boldly say that the problems of image restoration are over. The area of AN image restoration is still open for further research. The application of AN region growing techniques to adaptive, as well as iterative, image restoration methods that were implemented to improve the performance of conventional filters still needs to be investigated. Since image stationarity is an important issue in image restoration, the use of AN methods can be advantageous to most image restoration techniques as discussed in this thesis.

REFERENCES

- [1] H.C. Andrews and B.R. Hunt, *Digital Image Restoration*, Englewood Cliffs, NJ: Prentice-Hall, 1977.
- [2] R.C. Gonzalez and P. Wintz, *Digital Image Processing*, Addison-Wesley, Reading, MA, 2nd Edition, 1987.
- [3] A.K. Jain, *Fundamentals of Digital Image Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [4] F.M. Wahl, *Digital Image Signal Processing*, Artech House, Inc., Norwood, MA, 1987.
- [5] B.R. Hunt, Ch. 4, "Digital Image Processing", in A.V. Oppenheim, Editor, *Applications of Digital Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, pp. 169-237, 1978.
- [6] J. Biemond, R.L. Lagendijk and R.M. Mersereau, "Iterative Methods for Image Deblurring", *Proc. IEEE*, vol. 78, No. 5, pp. 856-883, May 1990.
- [7] A.D. Hillery and R.T. Chin, "Iterative Wiener Filters for Image Restoration", *IEEE Trans. on Signal Processing*, vol. 39, No. 8, pp. 1892-1899, August 1991.
- [8] W.J. Song and W.A. Pearlman, "Restoration of Noisy Images With Adaptive Windowing and Non-linear Filtering.", *Proc. SPIE*, vol. 707, Visual Communications & Image Processing, pp. 198-206, 1986.
- [9] W.J. Song and W.A. Pearlman, "A Minimum-Error, Minimum-Correlation Filter for Images", *Proc. SPIE*, vol. 697, Applications of Digital Image Processing IX, pp. 225-232, 1986.

- [10] B. Mahesh, W.J. Song and W.A. Pearlman, "Adaptive Estimators for Filtering Noisy Images.", *Optical Engineering*, vol. 29, No.5, pp. 488-494, May 1990.
- [11] P. Chan and J.S. Lim, "One-Dimensional Processing for Adaptive Image Restoration", *IEEE Trans. on ASSP*, vol. 33, No.1, pp. 117-129, Feb. 1985.
- [12] J.S. Lim, "Image Restoration by Short Space Spectral Subtraction", *IEEE Trans. on ASSP*, vol. 28, No. 2, pp. 191-197, April 1980.
- [13] R.B. Paranjape, T.F. Rabie and R.M. Rangayyan, "Image Restoration by Adaptive-Neighborhood Noise Subtraction", *Applied Optics*, (submitted, July 13, 1992).
- [14] A.V. Oppenheim and J.S. Lim, "The Importance of Phase in Signals", *Proc. IEEE*, vol. 69, No. 5, pp. 529-541, May 1981.
- [15] T.G. Stockham, T.M. Cannon and R.B. Ingebretsen, "Blind Deconvolution Through Digital Signal Processing", *Proc. IEEE*, vol. 63, No. 4, pp. 678-692, April 1975.
- [16] R.N. Strickland, "Transforming Images into Block Stationary Behavior", *Applied Optics*, vol. 22, No. 10, pp. 1462-1473, May 1983.
- [17] H.J. Trussell and B.R. Hunt, "Sectioned Methods for Image Restoration", *IEEE Trans. on ASSP*, vol. 26, No. 2, pp. 157-164, April 1978.
- [18] H.J. Trussell and B.R. Hunt, "Image Restoration of Space-Variant Blurs by Sectioned Methods", *IEEE Trans. on ASSP*, vol. 26, No. 6, pp. 608-609, Dec. 1978.
- [19] M.M. Hadhoud and D.W. Thomas, "The Two-Dimensional Adaptive LMS (TDLMS) Algorithm", *IEEE Trans. Circuits and Systems*, vol. 35, No. 5, pp. 485-494, May 1988.

- [20] W.B. Mikhael and S.M. Ghosh, "Two-Dimensional Variable Step-Size Sequential Adaptive Gradient Algorithms with Applications", IEEE Trans. on Circuits and Systems, vol. 38, No. 12, pp. 1577-1580, Dec. 1991.
- [21] W.J. Song and W.A. Pearlman, "Edge-Preserving Noise Filtering Based on Adaptive Windowing", IEEE Trans. on Circuits and Systems, vol. 35, pp. 1046-1055, 1988.
- [22] H.S. Mallikarjuna and L.F. Chaparro, "Iterative Composite Filtering for Image Restoration", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 14, No. 6, pp.674-678, June 1992.
- [23] K. Rank and R. Unbehauen, "An Adaptive Recursive 2-D Filter for Removal of Gaussian Noise in Images", IEEE Trans. on Image Processing, vol. 1, No. 3, pp. 431-436, July 1992.
- [24] C.A. Pomalaza-Raez and C.D. McGillem, "An Adaptive, Nonlinear Edge-Preserving Filter", IEEE Trans. on ASSP, vol. 32, No. 3, pp. 571-576, June 1984.
- [25] L. Sica, "Use of a Sharpness Criterion in Correcting Images Degraded by Random Distortion", Journal of the Optical Society of America, vol. 5, No. 9, pp. 1492-1501, Sept. 1988.
- [26] S.A. Rajala and R.J.P. De Figueiredo, "Adaptive Nonlinear Image Restoration by a Modified Kalman Filtering Approach", IEEE Trans. on ASSP, vol. 29, No. 5, pp. 1033-1042, Oct. 1981.
- [27] R. Gordon and R.M. Rangayyan, "Radiographic Feature Enhancement, Information Content, and Dose Reduction in Mammography and Cardiac Angiography", Proceedings of the IEEE (EMBS) Frontiers of Engineering and Computing in Health Care Conference, pp. 161-165, 1983.

- [28] R. Gordon and R.M. Rangayyan, "Feature Enhancement of Film Mammograms Using Fixed and Adaptive Neighborhoods", *Applied Optics*, vol. 23, pp. 560-564, Feb. 1984.
- [29] W.M. Morrow, "Region-based Image Processing with Application to Mammography", M.Sc. Thesis, The University of Calgary, Calgary, Canada, Dec. 1990.
- [30] R.B. Paranjape, R.M. Rangayyan, W.M. Morrow and H.N. Nguyen, "Adaptive Neighborhood Image Processing", *Proc. SPIE, Visual Communications & Image Processing*, Boston, Nov. 18-20, 1992, in press.
- [31] W.M. Morrow and R.M. Rangayyan, "Implementation of Adaptive Neighborhood Image Processing Algorithms on a Parallel Supercomputer", *Proc. Supercomputing Symposium '90*, University of Montreal, pp. 329-334, June 4-6, 1990.
- [32] W.M. Morrow and R.M. Rangayyan, "Feature-Adaptive Enhancement and Analysis of High-Resolution Digitized Mammograms", *Proceedings of the 12th IEEE Engineering in Medicine and Biology Conference*, pp. 165-166, 1990.
- [33] R.B. Paranjape, W.M. Morrow and R.M. Rangayyan, "Adaptive-Neighborhood Histogram Equalization for Image Enhancement", *Computer Vision, Graphics, and Image Processing: GMIP*, vol. 54, No. 3, pp. 259-267, May 1992.
- [34] W.M. Morrow, R.B. Paranjape, R.M. Rangayyan and J.E.L. Desautels, "Region-Based Contrast Enhancement of Mammograms", *IEEE Trans. on Medical Imaging*, vol. 11, No. 3, pp. 392-406, Sept. 1992.
- [35] M. Cannon, "Blind Deconvolution of Spatially Invariant Image Blurs with Phase", *IEEE Trans. on ASSP*, vol. 24, No. 1, pp. 58-63, Feb. 1976.
- [36] C.V. Jakowatz, P.H. Eichel and D.C. Ghiglia, "Image Restoration Via Iterative Improvement of the Wiener Filter", *Proc. SPIE*, vol. 974, *Applications of Digital Image Processing XI*, pp. 46-52, 1988.

- [37] L. Guan and R.K. Ward, "Restoration of Randomly Blurred Images Via the Maximum A Posteriori Criterion", IEEE Trans. on Image Processing, vol. 1, No. 2, pp. 256-262, April 1992.
- [38] M.M. Sondhi, "Image Restoration: The Removal of Spatially Invariant Degradations", Proc. IEEE, vol. 60, No. 7, pp. 842-853, July 1972.
- [39] A.A. Sawchuk, "Space-Variant Image Motion Degradations and Restoration", Proc. IEEE, vol. 60, No. 7, pp. 854-861, July 1972.
- [40] G.M. Robbins and T.S. Huang, "Inverse Filtering for Linear Shift-Variant Imaging Systems", Proc. IEEE, vol. 60, No. 7, pp. 862-872, July 1972.
- [41] R.A. Gonsalves and H.M. Kao, "Maximum Entropy Image Restoration: Formulated as a Correction Term to the Inverse Filter Solution", Proc. SPIE, vol. 697, Applications of Digital Image Processing IX, pp. 357-363, 1986.
- [42] T. Katayama, "Restoration of Images Degraded by Motion Blur and Noise", IEEE Trans. on Automatic Control, vol. 27, No. 5, pp. 1024-1033, Oct. 1982.
- [43] S.J. Reeves and R.M. Mersereau, "Blur Identification by the Method of Generalized Cross-Validation", IEEE Trans. on Image Processing, vol. 1, No. 3, pp. 301-311, July 1992.
- [44] N.B. Karayiannis and A.N. Venetsanopoulos, "Regularization Theory in Image Restoration - The Stabilizing Functional Approach", IEEE Trans. on ASSP, vol. 38, No. 7, pp. 1155-1179, July 1990.
- [45] R.L. Lagendijk, J. Biemond and D.E. Boeke, "Identification and Restoration of Noisy Blurred Images Using the Expectation-Maximization Algorithm", IEEE Trans. on ASSP, vol. 38, No. 7, pp. 1180-1191, July 1990.

- [46] P.D. Welch, "The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodograms", IEEE Trans. on Audio and Electroacoustics, vol. 15, No. 2, pp. 70-73, June 1967.
- [47] C. Bingham, M.D. Godfrey and J.W. Tukey, "Modern Techniques of Power Spectrum Estimation", IEEE Trans. on Audio and Electroacoustics, vol. 15, No. 2, pp. 56-66, June 1967.
- [48] C.M. Rader, "An Improved Algorithm for High Speed Autocorrelation with Applications to Spectral Estimation", IEEE Trans. on Audio and Electroacoustics, vol. 18, No. 4, pp.439-441, Dec. 1970.
- [49] R.A. Gabel and R.A. Roberts, *Signals and Linear Systems*, John Wiley & Sons, Inc., New York, 3rd Edition, pp. 401-408, 1987.
- [50] S.C. Pohlig, J.S. Lim, A.V. Oppenheim, D.E. Dudgeon and A.E. Filip, "New Technique for Blind Deconvolution", Proc. SPIE, vol. 207, Appl. of Digital Image Processing III, pp. 119-124, 1979.
- [51] M.H. Hayes, J.S. Lim and A.V. Oppenheim, "Signal Reconstruction from Phase or Magnitude", IEEE Trans. on ASSP, vol. 28, No. 6, pp. 672-680, Dec. 1980.
- [52] T.S. Huang, J.W. Burnett, and A.D. Deczky, "The Importance of Phase in Image Processing Filters", IEEE Trans. on ASSP, vol. 23, No. 6, pp. 529-542, Dec. 1975.
- [53] J. Behar, M. Porat and Y.Y. Zeevi, "Image Reconstruction from Localized Phase", IEEE Trans. on Signal Processing, vol. 40, No. 4, pp. 736-743, April 1992.
- [54] A.V. Oppenheim, M.H. Hayes and J.S. Lim, "Iterative Procedures for Signal Reconstruction from Phase", Proc. SPIE, vol. 231, Inter. Optical Computing Conf., pp. 121-129, 1980.

- [55] M.H. Hayes, "The Reconstruction of a Multidimensional Sequence from the Phase or Magnitude of its Fourier Transform", IEEE Trans. on ASSP, vol. 30, No. 2, pp. 140-154, April 1982.
- [56] D. Kermisch, "Image Reconstruction from Phase Information Only", Journal of the Optical Society of America, vol. 60, No. 1, pp. 15-17, Jan. 1970.
- [57] C.Y. Espy and J.S. Lim, "Effects of Additive Noise on Signal Reconstruction from Fourier Transform Phase", IEEE Trans. on ASSP, vol. 31, No. 4, pp. 894-898, August 1983.
- [58] T.F. Rabie, R.B. Paranjape and R.M. Rangayyan, "An Iterative Method for Blind Deconvolution", IEEE Trans. on Image Processing, (submitted, Oct. 1, 1992).