

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

**ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600**

UMI[®]

**NOVEL SUPER-RESOLUTION ALGORITHMS AND ENHANCED
NOISE REMOVAL ALGORITHM FOR IMAGE RESTORATION
SYSTEMS AND APPLICATIONS**

by
Ho-Yuen Pang

Copyright © Ho-Yuen Pang 2002

**A Dissertation Submitted to the Faculty of the
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
In Partial Fulfillment of the Requirements
For the Degree of
DOCTOR OF PHILOSOPHY
In the Graduate College
THE UNIVERSITY OF ARIZONA**

2 0 0 2

UMI Number: 3050345

Copyright 2002 by
Pang, Ho-Yuen

All rights reserved.

UMI[®]

UMI Microform 3050345

Copyright 2002 by ProQuest Information and Learning Company.

All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

THE UNIVERSITY OF ARIZONA ©
GRADUATE COLLEGE

As members of the Final Examination Committee, we certify that we have read the dissertation prepared by Ho-Yuen Pang

entitled Novel Super Resolution Algorithms and Enhanced Noise Removal Algorithm For Image Restoration Systems and Applications

and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy

Richard W. Ziolkowski

Richard W. Ziolkowski, Ph.D.

3/21/02
Date

Robert Schowengerdt

Robert Schowengerdt, Ph.D.

3/21/02
Date

John R. Brews

John R. Brews, Ph.D.

Mar 21 / 02
Date

David Mathine

David Mathine, Ph.D.

April 21, 2002
Date

Date

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copy of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

Richard W. Ziolkowski

Dissertation Director Richard W. Ziolkowski; Ph.D.

4/17/02
Date

STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the copyright holder.

SIGNED: Ho-Yuen Pang

ACKNOWLEDGEMENTS

The dissertation committee gave advice and support to make this work possible. Especially, I wish to express my deep gratitude to my dissertation director, Dr. Richard Ziolkowski, for his invaluable guidance and support. I would like to thank Dr. John Brews for his additional mentoring and helping me to become a better teaching assistant. I would also like to thank Dr. Robert Schowengerdt to groom me to discover the beauty of remote sensing. I also thank the minor committee members, Dr. David Mathine and Dr. Jack Gaskill, for their time and input during this effort. I would like to thank my former advisor, Dr. Malur Sundareshan, for his initial guidance and support.

In addition to thanking my committee members, I would like to thank all the professors who taught me at the University of Arizona. Their dedication to teaching helped me to become a better scholar and thinker, and to develop interest in diverse fields. Especially, I would like to thank the professors of Electrical and Computer Engineering, Optical Sciences, Physics, Mathematics, Business and Public Administration, and English.

All of my fellow graduate students and family have been an invaluable source of ideas, assistance, and motivation. I especially thank my friends Yee Chin Wong, Pablo Zegers, Tim Peterson, Ali Bilgin, Joe Green, Edwin Marengo, Candido Pinto, Tracy Smith, and Michael Williams. We have shared ideas and overcome difficulties together in this great research institute. Of course, none of this would have been possible without the love and support of my parents.

Finally, I gratefully acknowledge the *Air Force Office of Scientific Research* and the *Army Research Laboratory* whose sponsorship funded the initial portion of my research effort during my studies.

*To my dear parents
and
all the pioneers
in
science and engineering*

TABLE OF CONTENTS

LIST OF FIGURES	11
LIST OF TABLES	15
ABSTRACT	16
1 INTRODUCTION	18
1.1 Limit of data acquisition	18
1.2 Overview of the dissertation	19
2 IMAGE PROCESSING BASICS	22
2.1 Imaging as Poisson Processing	24
2.2 Inverse Problem	25
2.3 Survey of Common Restoration Techniques	26
2.3.1 Inverse Filter	27
2.3.2 Wiener Filter	28
2.3.3 Maximum a-Posteriori	30
2.3.4 Maximum Likelihood	31
2.4 Image Super-resolution	32

TABLE OF CONTENTS – *Continued*

3 DYNAMIC IMAGING SYSTEM BASED ON	
INFORMATION THEORY	35
3.1 Communication Theory	37
3.2 Similarity and Dissimilarity between Communication Model and Im- age Processing Model	38
3.3 Model Image Processing Based on Communication Theory	39
3.4 Introduction to the Proposed Dynamic Imaging System Model	42
3.4.1 Constraint Imposed By the Application	48
3.5 Components To Be Studied	50
 4 A MARKOV RANDOM FIELD BASED	
CHANNEL DECODER	52
4.1 Introduction	52
4.2 MRF Modeling and Channel Decoder	54
4.3 Limitations of the ICM Algorithm	59
4.4 Relative Potential Function and an Improved ICM Algorithm For Channel Decoding	61
4.5 Application of the Source Decoder to the Channel Decoded Image .	69
4.6 Conclusions	78

TABLE OF CONTENTS – *Continued*

5 OBJECT-BOUNDARY-CONSTRAINED MAXIMUM	
LIKELIHOOD ALGORITHM	84
5.1 Introduction	84
5.2 Importance of Object Extension	84
5.3 Incorporate Object Support Constraint into the Maximum Likelihood Algorithm	85
5.4 1-D Validation Experiments	87
5.5 2-D Validation Experiments	99
5.6 Conclusions	118
 6 SUPER-RESOLUTION ALGORITHM BASED ON A	
SELF-ORGANIZING MAP	120
6.1 Introduction	120
6.2 Neural Networks Basics	121
6.3 The Possible Uses of Local Information for Image Super-resolution .	124
6.4 Self-Organizing Map Training for Image Super-resolution	126
6.5 Validation Experiment	134
6.6 Conclusions	141

TABLE OF CONTENTS – *Continued*

7 SUMMARY AND FUTURE WORK	143
7.1 Closing Remarks	143
7.2 Contribution	144
7.3 Integration and Further Development	147
7.4 Further Development	149
APPENDIX A. PROOF OF GIBBS-MARKOV EQUIVALENCE	
FOR A GIBBS RANDOM FIELD WITH A RELATIVE	
POTENTIAL FUNCTION	
	151
APPENDIX B. MAXIMUM LIKELIHOOD ALGORITHM	
DERIVATION	
	158
B.1 Maximization of the Likelihood of the Image	158
B.2 Poisson Statistical Model of the Object and Image	159
B.3 The Maximum Likelihood Algorithm	160

TABLE OF CONTENTS – *Continued*

APPENDIX C. SOURCE CODE	164
C.1 Disclaimer	164
C.2 Matlab Source Code for a Markov Random Field Based Channel Decoder	164
C.3 Matlab Source Code for Object-Boundary-Constrained Maximum Likelihood Algorithm	179
C.4 C++ Source Code for Super-resolution Algorithm Based on a Self- Organizing Map	180
REFERENCES	211

LIST OF FIGURES

2.1 Imaging system	22
2.2 Upsampling process	34
3.1 Communication system	35
3.2 Imaging system	36
3.3 Imaging model based on deterministic implementation	37
3.4 Image formation	40
3.5 Communication theory based image formation model	41
3.6 Improved image formation model based on communication theory . .	42
3.7 Dynamic Imaging System	43
3.8 Source encoder	45
3.9 Channel encoder	45
3.10 Channel decoder	46
3.11 Source decoder	46
3.12 Post-processing	47
3.13 Dynamic Decision Making System	47
4.1 (a) First order neighborhood. (b) Second order neighborhood. (c) Fifth order neighborhood.	56

LIST OF FIGURES – *Continued*

4.2	Second order neighborhood and 24 cliques.	57
4.3	Typical slides in intensity function. (a) A downward slope. (b) A positive value noise on a flat surface. (c) A negative value noise on a flat surface	60
4.4	Lenna.	65
4.5	(a) A portion of an original Lenna image. (b) A blurred portion by a low pass filter. (c) Adding “Salt and Pepper” noise in the blurred portion of Lenna	66
4.6	(a) A 3×3 median filtered image.(b) ICM filtered image. (c) Proposed filtered image	67
4.7	Tank.	68
4.8	(a) A portion of an original tank image. (b) A blurred portion by a low pass filter. (c) Adding “Salt and Pepper” noise in the blurred portion of the tank image	69
4.9	(a) A 3×3 median filtered image. (b) ICM filtered image. (c) Proposed filtered image	70
4.10	An one-dimensional example of the maximum likelihood algorithm .	72
4.11	Maximum likelihood post-process filter output images. (a) Super-resolved 3×3 median filter output. (b) Super-resolved ICM filter output. (c) Super-resolved the proposed filter output. (d) Super-resolved noisy blurred image. (e) Super-resolved blurred image. . . .	73

LIST OF FIGURES – *Continued*

4.12 Power spectrum of images. (a) Original image. (b) Blurred image. (c) Super-resolved blurred image. (d) Super-Resolved 3×3 median filter output. (e) Super-resolved ICM filter output. (f) Super-resolved proposed filter output.	80
4.13 A portion of Lenna image flipped two times	81
4.14 A portion of Lenna image flipped six times	82
4.15 (a) The correlation between FFT of the original object and FFT of the super-resolved 3×3 median filtered image. (b) The correlation between FFT of the original object and FFT of the super-resolved ICM filtered image. (c) The correlation between FFT of the original object and FFT of the super-resolved proposed filtered image.	83
5.1 An one-dimensional example of the Object-Boundary-Constrained Maximum Likelihood Algorithm	88
5.2 Object-Boundary-Constrained Maximum Likelihood Algorithm	89
5.3 1-D case 1.	90
5.4 1-D case 2.	93
5.5 1-D case 3.	94
5.6 1-D case 4.	95
5.7 1-D case 5	96
5.8 1-D case 6.	98
5.9 2-D case 1.	100

LIST OF FIGURES – *Continued*

5.10 2-D case 2.	101
5.11 2-D case 3.	103
5.12 2-D case 4a.	105
5.13 2-D case 4b.	106
5.14 2-D case 5	107
5.15 2-D case 6.	109
5.16 2-D case 7a.	111
5.17 2-D case 7b.	112
5.18 2-D case 8.	113
5.19 2-D case 9.	114
5.20 2-D case 10a.	116
5.21 2-D case 10b.	117
6.1 The two-dimensional self-organizing map model.	127
6.2 The training procedure of the self-organizing map model.	128
6.3 The construction of the input vector.	132
6.4 The SOM based image super-resolution process.	133
6.5 The clear training image.	136
6.6 The blurred training image.	137
6.7 The original image.	138
6.8 The acquired blurred image.	139
6.9 The restored image.	140

LIST OF TABLES

4.1	Performance record	79
4.2	Proposed filter performance	79
5.1	OBC-ML performance	119

ABSTRACT

This dissertation is concerned with the introduction of a systematic way of modeling image processing. A dynamic imaging system model constructed from an information theory framework is proposed. Unlike an earlier simple model, the proposed dynamic imaging system (DIS) model is suitable for a wide range of applications. This DIS model is inspired by the Shannon communication theory. The Shannon communication theory is credited for the rapid development of the communication industry. Currently, most image processing researchers focus on developing fast algorithms and better hardware. An information theoretic-based approach to image processing could bring as large an impact to the image processing area as Shannon's communication theory had on the communications area.

This proposed DIS model will use the information obtained from the acquired images to provide an estimation of the unknown atmospheric turbulence, vibration, etc. It will also automatically adjust the sampling rate, wavelength band, and algorithms of choice, to produce the best possible restored image with limited information under uncertainty.

This dissertation develops the concept of the DIS model including its basic components. We have implemented three parts of this system. First, we implemented a noise removal algorithm based on the Markov random field (MRF). It is shown that this algorithm achieves better performance than other MRF-based algorithms in noise

removal. Second, we have implemented a hybrid maximum likelihood/projection-on-convex-set image restoration algorithm and demonstrate that it outperforms the maximum likelihood algorithm. Third, we have implemented a self-organized map-based image restoration algorithm and compare its performance to several well-known methods. It can be implemented in parallel processing to achieve super-resolution in real time without performing a time consuming iteration process. The impact of the development of these DIS system critical components is discussed and future research areas are elucidated.

CHAPTER 1

INTRODUCTION

1.1 Limit of data acquisition

As humans, we understand the world by the data we collect through our human sensory system. For imaging sensors, the data we acquire are limited by both known and unknown constraints. The known limitation can be the limit imposed by physics. The aperture size of an image sensor limits the resolution based on diffraction limit. The accuracy of the curvature of the lens also affects the clarity image as shown by the Hubble Space Telescope (HST) [1]. The signal-to-noise ratio in the acquisition system also imposes a limit of how low the intensity of incoming light affects the sampling rate of an imaging array. The unknown limitation can be the atmospheric turbulence, absorption of different wave length in an uncertain air composition of difference molecules, and the vibration of the platform.

Researchers have developed various image restoration algorithms for over thirty years by modeling the limitations and distortions imposed by instruments and the environment. The development of image processing has been assisted by the rapid development of the microprocessors. As computational power doubles every 18 months according to Moore's law, new computationally intense algorithms and new applications become reality.

However, the progress of image processing falls short compared to the progress of communications. O'Sullivan [2] suggested that the possible role of information theory [3] in problems of image processing was to provide a rigid framework for defining the image problem. We extend from this view and try to understanding the information path of image processing from image formation to image restoration. Initially, our model is inspired by the Shannon communication theory. As our research progresses, we extend our model to a highly nonlinear dynamic imaging system model. This proposed imaging model opens up the possibility of developing completely new approaches for image processing. Due to complexity of this proposed imaging model, only three key components of this model are implemented in this dissertation.

1.2 Overview of the dissertation

In Chapter 2, we will go through the basics of optical science that are relevant to the image processing problem. The mathematical and statistical preliminaries of the forward image formation, which are the basics of image science, are reviewed. Subsequently, the notion of the inverse problem of image estimation is reviewed. In the course of this dissertation, we will review the development of some basic linear and nonlinear image restoration algorithms. The present development focuses only on the inverse problem and not upon the information theory aspects of image processing.

In Chapter 3, we propose the notion of a dynamic imaging system based on the Shannon communication theory. The similarity between image formation/restoration and communication theory will be presented. The logic behind the proposed dynamic

imaging system model will be developed. In addition, all the required components will be briefly explained.

In Chapter 4, we extend the iterative conditional mode (ICM) algorithm with Newtonian mechanics to estimate the potential function in a neighborhood. The ICM algorithm is a Markov random field (MRF) based estimation algorithm. We explain a limitation of the ICM algorithm and introduce Newtonian mechanics to assist the estimation of the local probability. Even though the Gibbs-Markov equivalence is not preserved in this new approach, the simulation results show a performance far superior to those obtained with the ICM approach. The computational requirement of this algorithm is high. However, the parallel processing nature of MRF is ideal to be implemented in hardware. The field programmable gate array (FPGA) is a low cost method to accelerate the calculation. The FPGA-based reconfigurable system can also be used to perform this parallel computation. The current trend of moving the imaging array from charge couple devices (CCDs) to CMOS image arrays provides a unique opportunity of building a smart pixel array [4]. We can implement this new ICM algorithm using smart pixels. Therefore, all computations can be performed in a parallel processing mode on a chip. Consequently, the computational time will be greatly reduced.

In Chapter 5, we investigate at the limitations of the maximum likelihood (ML) algorithm. It is proposed that the ML super-resolution performance is limited only by the contour of the estimated object. By applying the projections onto convex sets (POCS) on the contour of the object, we increase the prior knowledge for the

image super-resolution. In return, a high super-resolution performance is achieved in a validation experiment with exact knowledge of the signal contour.

In Chapter 6, we explore the possibility of applying a self-organizing map (SOM) to perform image super-resolution. The SOM, which is one form of a neural network (NN), is a powerful algorithm for data clustering. In this chapter, we apply SOM to cluster the data of the blurred image blocks with clear image pixels to train a SOM weight to map the blurred image blocks to clear image pixels. This SOM-based algorithm, which can be performed in parallel processing, can dramatically reduce the computational time in the image super-resolution process. A high speed optical data base may be used to perform a fast search for all pixels.

Finally, Chapter 7 summarizes our key results and tie together the dynamic imaging system with the above three components. We also propose future directions of this research.

CHAPTER 2

IMAGE PROCESSING BASICS

Despite the rapid developments in the field, the formulation of image formation is still based on a linear shift invariant (LSI) model. The main reason is tractability. The light source is assumed to be incoherent and monochromatic. The light is focused on the focal plane. Moreover, the image is collected in the Fraunhofer region. These constraints yield a linear shift invariant image model with a diffraction limit [5, 6]. The object is assumed to be fixed and is not changing over time. This condition exists when the imaging time is relatively small.

In Fig.(2.1), the blurred image $g(x, y)$ is the convolution of the original object $f(x, y)$ and the point spread function (PSF) $h(x, y)$ of the optics [7]. This is a simpler model assuming the object is flat and can be represented as

$$g(x, y) = \{f(x, y) * h(x, y)\} = \int_{-\infty}^{\infty} f(\xi, \eta)h(\xi - x, \eta - y)d\xi d\eta \quad (2.1)$$

Basically, PSF is a function of the exit pupil $p(x, y)$ of the optical system, where $p(x, y)$ is scaled by the wavelength, λ , of the incoming light and the distance from

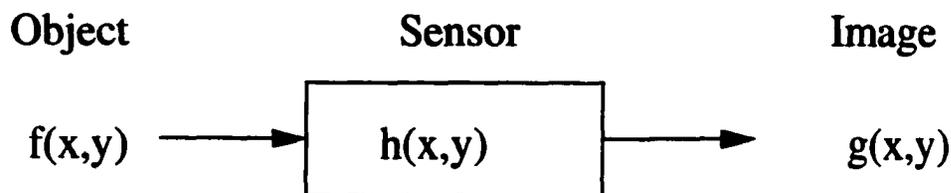


Figure 2.1: Imaging system

the exit pupil to the image plane d_i . The exit pupil represents a finite aperture through which light must pass when it leaves the imaging elements on its way to the image plane. The exit pupil function can be formulated as

$$h(x, y) = \frac{\left(\frac{1}{\lambda d_i}\right)^2 |P\left(\frac{x}{\lambda d_i}, \frac{y}{\lambda d_i}\right)|^2}{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x', y') dx' dy'} \quad (2.2)$$

where $P(\xi, \eta) = FFT(FFT(p(x, y)))$ The frequency characteristic of the system is referred to as the optical transfer function (OTF). When we need to calculate the OTF from physical parameter, we know the OTF is a scaled version of the complex autocorrelation of the exit pupil function. This may look confusing. The OTF is a function in the frequency domain and the exit pupil function is a function in the space domain. The above is a common practice in image processing. Detailed explanation of OTF calculation can be found in Gaskill [8]. With a finite aperture, and thus a finite exit pupil, the OTF must be finite in extent as well. Therefore, the OTF, which is a low pass filter, truncates the object spectrum and only passes a finite range of Fourier information to the image. Due to this finite range, some of the high frequency information is permanently lost.

If we know the point spread function (PSF) of the imaging system, the OTF is simply a Fourier transform of the PSF, *i.e.*,

$$H(\xi, \eta) = FT(h(x, y)) \quad (2.3)$$

where FT stands for a Fourier transform. This behavior in the frequency domain can also be explained in the spatial domain. The higher the frequency information contained in the image, the finer detail we can observe. The resolution in the acquired

image is based on the minimum resolvable angular separation, or angular limit of resolution, which is governed by the Rayleigh criterion for diffusion-limited imaging operations as follow:

$$(\Delta\varphi)_{min} = \Delta\theta = 1.22\frac{\lambda}{D} \quad (2.4)$$

where λ is the wavelength and D is the aperture diameter. The limit on resolution is given by

$$(\delta l)_{min} = 1.22f\lambda/D \quad (2.5)$$

where f is the focal length. This limitation can be measured by injecting a single light pulse into an imaging system. The acquired image in the Fraunhofer region will have an intensity profile that is referred to as the Airy pattern, after G. B. Airy who first derived it. The first zero is given by Eq.(2.5).

2.1 Imaging as Poisson Processing

Equation (2.1) depicts the wave nature of imaging. This imaging model is an ideal case that is based on a deterministic model of imaging. However, in practice, the statistical nature of imaging is more prominent. On a charge couple device (CCD) array, every pixel has to integrate the energy of the captured photons and converts this information to a number of electrons by the photoelectric effect during a given sampling period. The number of photons reflected from an object or emitted by a light source, and the detection of the number of photons can be modeled as a noise process.

A Poisson density function can model emission or reflection process of the object and detection process on image plane. At a particular location in the image plane, the probability of detecting k number of photons can be formulated as

$$Pr(K = k) = \frac{(\eta\phi T)^k \exp(-\eta\phi T)}{k!} \quad (2.6)$$

where ϕ is the average photon emission rate, T is the integration time of the detection element, η is the detection efficiency of the detection element. The probability of a photon being detected is determined by η . Therefore, $n(x, y)$ is the signal dependent Poisson noise (SDPN) and Eq.(2.1) can be rewritten as

$$g(x, y) = \{f(x, y) * h(x, y)\} \odot n(x, y) \quad (2.7)$$

where \odot is a pointwise operator that performs multiplication of elements in a given location. Therefore, each pixel in the imaging array detects a Poisson random variable. Equation (2.1) is a mean of Eq.(2.7). Therefore, an acquired data is a collection of random variables.

2.2 Inverse Problem

Image restoration can be viewed as a mapping problem. A blurred image is mapped to a clear image with a mathematical formula. Furthermore, image restoration is an inverse problem [9]. An inverse problem may be either well-posed or ill-posed. To illustrate this point, domain X and range Y are taken to be matrix spaces that are connected with an unknown mapping function F . The mapping is considered well-posed if the following three conditions are met [10, 11]

1. *Existence.* For every $x \in X$, an output $y = F(x)$ must exist where $y \in Y$.
2. *Uniqueness.* For every pair of input vector x and t , we have $F(x) = F(t)$ if and only if $x=t$.
3. *Continuous.* For any $\epsilon > 0$, there exists some $\delta = \delta(\epsilon)$ such that given distance between two arguments x and t in the matrix space X is less than δ , the condition

$$\rho_X(x, t) < \delta \quad (2.8)$$

implies that the distance between their mappings in the projected space Y $F(x)$ and $F(t)$ is less than ϵ

$$\rho_Y(F(x), F(t)) < \epsilon \quad (2.9)$$

Image restoration is an ill-posed problem for two reasons. First, the OTF eliminates lots of high frequency information permanently. The information loss is severe. Therefore, there is not much information for the unique input-output mapping. The uniqueness requirement for a well-posed mapping is violated. Second, the noise presented in the image and the imprecision in the input data adds uncertainty to the input-output mapping. If the signal to noise level is low, it is possible that the mapping of x will be outside the range of Y . Therefore, the continuity requirement is also violated.

2.3 Survey of Common Restoration Techniques

It would be beneficial to survey some common image restoration techniques before further developing this dissertation. The algorithms we will discuss are well known in

the image processing community. The knowledge of the point spread function (PSF) and positive pixel values is presumed to be known. With only this information, researchers have developed many restoration algorithms.

2.3.1 Inverse Filter

Although linear filters are incapable of producing a super-resolution image, understanding of the linear filter is a prerequisite to understanding more advanced filters. The term super-resolution means the estimation of the lost information beyond the passband. The extrapolation of the unknown or truncated information beyond the passband is possible because of an *analytic continuation* of the information inside and outside the passband. The inverse filter is the most simple form of image restoration filter. Its goal is to recover the original object from the acquired image and the PSF of the imaging system. If we apply a Fourier transform on both sides of Eq.(2.1), we will obtain

$$G(\xi, \eta) = F(\xi, \eta) \times H(\xi, \eta) \quad (2.10)$$

and rearranging the equation, we have

$$F(\xi, \eta) = \frac{G(\xi, \eta)}{H(\xi, \eta)} \quad (2.11)$$

Therefore, an inverse filter, $H_{inverse}(\xi, \eta)$, is

$$H_{inverse}(\xi, \eta) = \frac{1}{H(\xi, \eta)} \quad (2.12)$$

The inverse filter in the frequency domain is the reciprocal of the OTF. However, $H_{inverse}(\xi, \eta) = \infty$ when $H(\xi, \eta) = 0$. The OTF has a finite bandwidth. Therefore,

outside the passband, $H(\xi, \eta) = 0$. We have then a singularity in the filter frequency response. To overcome the problem of a singularity, the pseudoinverse filter, which is a stable version of the inverse filter, is developed. The pseudoinverse filter, $H^{-}(\xi, \eta)$, is defined as

$$H^{-}(\xi, \eta) = \begin{cases} \frac{1}{H(\xi, \eta)} & H \neq 0 \\ 0 & H = 0 \end{cases} \quad (2.13)$$

In practice, $H^{-}(\xi, \eta)$ is set to zero when $H(\xi, \eta) < \epsilon$. When $H(\xi, \eta)$ is small, $H^{-}(\xi, \eta)$ will be a large response in the frequency domain. Both the inverse filter and the pseudoinverse filter are very sensitive to noise. All the noise that is inside the passband and near the cutoff frequency will be magnified by the pseudoinverse filter.

2.3.2 Wiener Filter

The Wiener filter is the optimal linear restoring filter in a minimum mean-square error sense. This filter can restore images in the presence of blur as well as noise. The blurring in the output g is caused by the point spread function (see Fig. 2.1).

$$g(x, y) = \{f(x, y) * h(x, y)\} \quad (2.14)$$

When an object is convolved with a point spread function, the image looks like a blurred version of the object. Noise also can be present. The noise n is caused by the thermal noise of the electronics. Then Eq.(2.14) becomes:

$$g(x, y) = \{f(x, y) * h(x, y)\} + n(x, y) \quad (2.15)$$

Its image restoration capability comes from minimizing the mean square error between f and its estimation \hat{f} :

$$\sigma_e^2 = E[(f(x, y) - \hat{f}(x, y))^2] \quad (2.16)$$

This filter is determined by the power spectrum of the object, the noise, and the OTF. The restored image Fourier spectrum is generated by

$$W(\xi, \eta) = \frac{H^*(\xi, \eta)}{|H(\xi, \eta)|^2 + \frac{S_{nn}(\xi, \eta)}{S_{ff}(\xi, \eta)}} \quad (2.17)$$

where H is the OTF, S_{ff} is a square of the Fourier transform of the auto-correlation of the object, and S_{nn} is a Fourier transform of the auto-correlation of the noise. These two matrices can be calculated by

$$S_{ff}(\xi, \eta) = |F(\xi, \eta)|^2 \quad (2.18)$$

$$S_{nn}(\xi, \eta) = |N(\xi, \eta)|^2 \quad (2.19)$$

where F is a Fourier transform of the object f and N is a Fourier transform of the noise n .

To apply this filter in image restoration, we need to multiply the power spectrum of the blurred image $G(\xi, \eta)^*$ with the Wiener filter.

$$\hat{F}(\xi, \eta) = G(\xi, \eta) \times W(\xi, \eta) \quad (2.20)$$

The above equation looks simple. However, the auto-correlation of the object is unknown. In practice, a smoothed acquired image is used to generate S_{ff} . The

* $G(\xi, \eta)$ is the Fourier transform of the Eq.(2.14).

noise is assumed to be the difference between the acquired image and the smoothed acquired image. While this assumption is overly simplified, the restored image is much better than the one restored by the inverse filter and pseudoinverse filter.

2.3.3 Maximum a-Posteriori

The maximum a-posteriori (MAP) algorithm, which was developed by Hunt [12], optimizes an estimate by maximizing its probability given the PSF and the image. It assumes that each position in the image is statistically independent from any other:

$$\hat{f} = \underset{f}{\operatorname{argmax}} p(g|f)p(f) \quad (2.21)$$

which maximizes the conditional probability of the image distribution given an object distribution. The MAP is an iteration process. The output of a previous step is fed back to the current calculation as the input. The $(n+1)$ -th and n -th step of this process is

$$\hat{f}^{n+1}(x, y) = \hat{f}^n(x, y) \times \exp \left\{ \left[\frac{g(x, y)}{\hat{f}^n(x, y) * h(x, y)} - 1 \right] \odot h(x, y) \right\} \quad (2.22)$$

Since an initial estimation of the object is unknown, the acquired image is often used as an initial estimation.

$$\hat{f}^0(x, y) = g(x, y) \quad (2.23)$$

The maximum a-posteriori algorithm reconstructs information beyond the pass-band. Therefore, it is considered as a super-resolution algorithm that performs non-linear calculations to extend the information beyond the passband. In practice, the

number of iterations are limited because stopping the algorithm early prevents the algorithm from over-fitting the noise. A stopping rule [13] can be applied to prevent noise amplification, but it is not necessary due to the time constraint and the computation resource limit.

2.3.4 Maximum Likelihood

The maximum likelihood algorithm is also known as the Richardson-Lucy algorithm. Richardson [14] developed it in 1970, Lucy [15] developed it in 1974, and the Russian scientist Tarasko [16] published it in 1968 in the U.S.S.R. It maximizes a posteriori density $p(g/f)$

$$\hat{f} = \underset{f}{\operatorname{argmax}} p(g|f) \quad (2.24)$$

that maximizes the conditional probability of the image distribution given an object distribution. It treats the PSF as a probability distribution function, relating the object to the image statistically.

The original proof is based on Bayes [17] rule. If we derive the ML algorithm with a Poisson probability distribution function and assume each position in the image is statistically independent from each other, we will arrive at the same solution as Richardson and Lucy did. The ML is an iterative process as well. The output of the current calculation will be feedback as the input of the future calculation. Its update is given by

$$\hat{f}^{n+1}(x, y) = \hat{f}^n(x, y) \times \left\{ \left[\frac{g(x, y)}{\hat{f}^n(x, y) * h(x, y)} \right] \odot h(x, y) \right\} \quad (2.25)$$

Since an initial estimation of the object is unknown, the acquired image is used as an initial estimation.

$$\hat{f}^0(x, y) = g(x, y) \quad (2.26)$$

Again, this algorithm is also considered to be a super-resolution algorithm. Shepp and Vardi [18] prove that the ML algorithm is guaranteed to converge. In theory, the higher the number of iterations, the better the image is restored. However, the number of iterations is affected by the noise amplification of the iteration process similar to the MAP algorithm. A stopping rule can also be deployed to obtain the optimal result.

2.4 Image Super-resolution

Image super-resolution is defined as the reconstruction of information beyond the passband. It was considered impossible because information beyond the passband was permanently lost. However, due to the success of super-resolution algorithms such as ML and MAP, people reconsidered the possibility of image super-resolution. By knowing the passband information, the lost information may be determined by *analytic continuation* [19]. For a real function, the information can be reconstructed using a Taylor series. For a complex function, such as the object spectrum, the continuation can be calculated by using a Laurent series.

Nonlinear algorithms such as MAP and ML are able to produce super-resolution images. However, algorithms alone cannot produce a super-resolved image. Normally, the acquired data is not suitable for image super-resolution processing. To

maximize the data acquisition, the CCD array is sampled at twice the cutoff frequency, known as the Nyquist frequency. When we look at the data file in frequency, the cutoff frequency is right at the border of the data file. There is no room to construct information beyond the cutoff frequency. Without room for extension, MAP and ML are prevented from recreating the lost information. Therefore, upsampling of the acquired data is needed for pre-processing of the data. For image processing, upsampling is a simple operation of zero padding outside the cutoff frequency. First we need to perform a Fourier transform on the acquired image. If the frequency domain data is $N \times N$, we need to create a $2N \times 2N$ sized zero matrix. After creating the zero matrix, we put the Fourier transform of the image into the zero matrix. We also multiply the new matrix by four to maintain the intensity level in the image because the data is four times bigger than the original. The procedure is shown in Fig. 2.2.

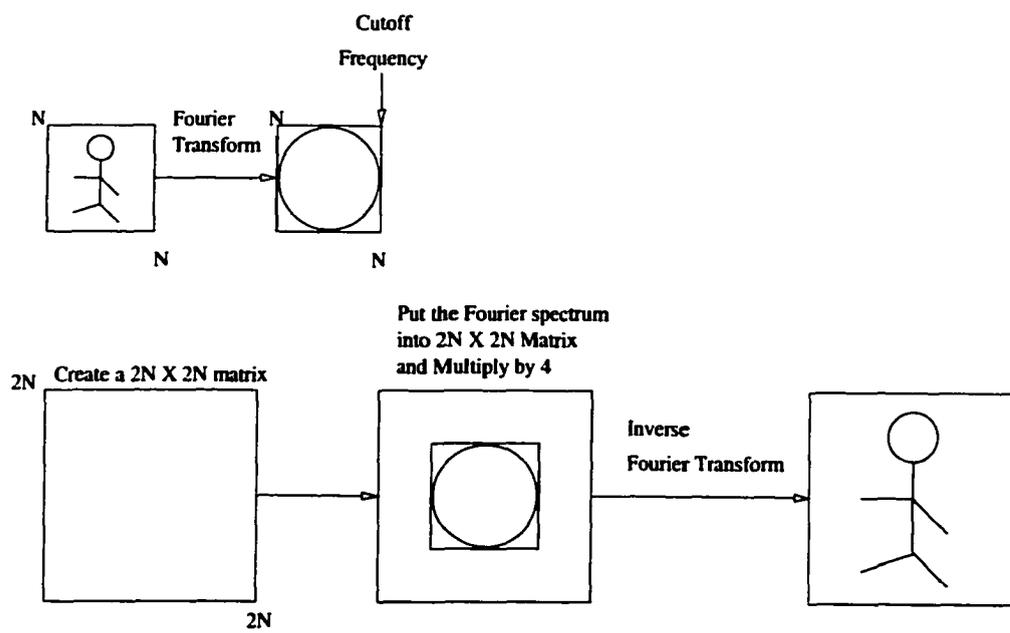


Figure 2.2: Upsampling process

CHAPTER 3

DYNAMIC IMAGING SYSTEM BASED ON INFORMATION THEORY

This chapter focuses on applying information theory to image processing. Information theory may be divided into two fields: communications and image processing. Shannon [20] developed communication theory in 1948. With his clear formalization, communication theory has advanced rapidly since 1948. Shannon had the insight to overlay the subject of communication with a clear partitioning into sources, channels, encoders, and decoders as shown in Fig. 3.1. His vision of communication systems seems obvious now, but it was not obvious in the first half of the twentieth century. All of these components are abstract modules that can be implemented physically. On the other hand, image processing has progressed slowly in the past 30 years. It has been driven by government agencies since the Cold War and has been fueled by

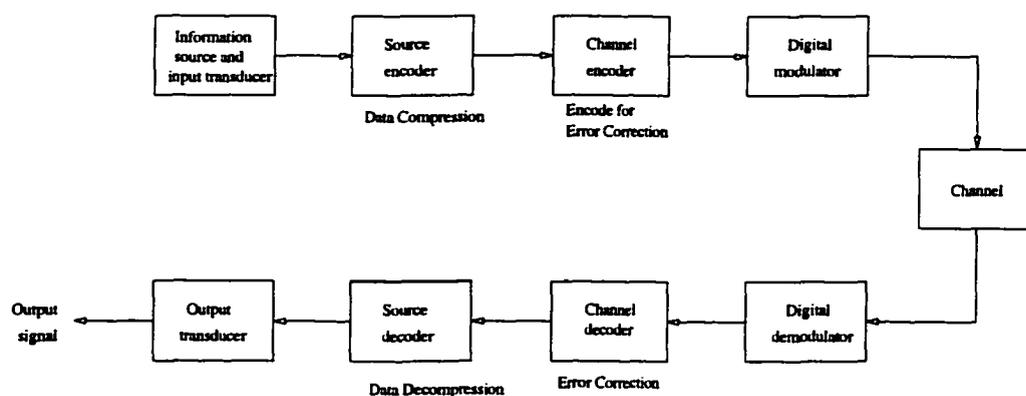


Figure 3.1: Communication system

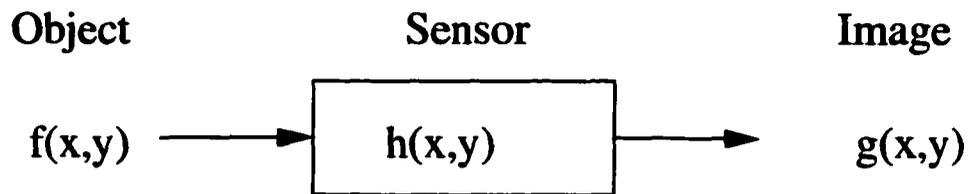


Figure 3.2: Imaging system

the rapid development of computing technologies based on Moore's Law. Image processing has been an integral part of modern technologies. However, the development of image processing has been slow, partly because the image processing is difficult, and perhaps partly because a formal framework for the subject is still emerging. Our understanding of image processing is tied to a particular physical problem as shown in Fig. 3.2. The imaging model can be extended to include electronic components for remote sensing, surveillance, etc., as shown in Fig. 3.3. It is not yet common practice to study problems of image formation and image processing in terms of an abstract formalization that is not connected to a specific physical problem.

Universally, everyone accepts this notion of an image model. Nevertheless, upon closer examination one can find difficulties with this simple view. A physical scene has a richness and complexity well beyond what we can, or may wish to model. In addition, the ultimate goal of image processing is to understand or to categorize the object presented in the field of view. In some extreme examples, the sensor data may contain very little information, but the prior knowledge may be considerable. One can use the sensor data to supplement the prior model to produce images. This is called model-based imaging. In target detection problem, the output of the system

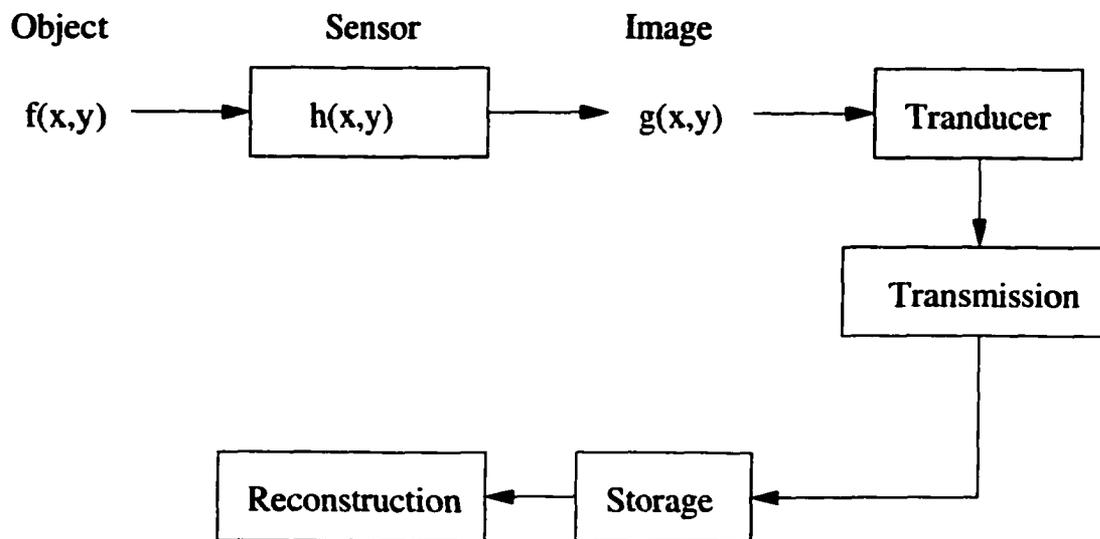


Figure 3.3: Imaging model based on deterministic implementation

may be simply yes or no, meaning only that a target does or does not appear in the scene.

The shortcoming of current image modeling approaches is that the model is totally based on the physical system. It is not possible to take full advantage of all the rich information the physical world provides with the current model. To take advantage of all that the physical world offers, we have to revise the image model. It would be beneficial to study the image formation and the image processing in terms of a high level abstract model. At first, we look at Shannon's communication theory.

3.1 Communication Theory

Figure 3.1 shows the information path of a signal from source to end user. The source output may be either analog or digital. In source encoding, we would like to compress the source as much as possible while resulting in little or no redundancy.

The process is called source encoding or data compression. The sequence of binary digits from the source encoder is called an information sequence. The information sequence is passed to the channel encoder. The channel will introduce redundancy in the binary sequence that can be used in error detection or error correction for the receiver. The digital modulator serves at the interface to the communication channel. The primary purpose of the digital modulator is to map the binary information sequence into a signal waveform. The communication channel is the physical medium that is used to send the signal from the transmitter to the receiver. For a different communication system, a different physical medium will be used as a channel. The communication channel may introduce noise, interference, and distortion into the information sequence. At the receiving end of a digital communications system, the digital demodulator processes the received waveforms and converts them to a sequence of information. The channel decoder will perform error correction and error checking for the converted information sequence. Finally, the source decoder will convert the information sequence to the approximation of its original form. The source decoder cannot guarantee the restored signal is 100% correct. Therefore, the source decoder can only generate a close approximation of the original signal.

3.2 Similarity and Dissimilarity between Communication Model and Image Processing Model

Figure 3.2 shows that the image formation is tied to the physical aspect of the system. However, it is still possible for us to convert the image processing model [7] to a communications model [21]. First, we have to eliminate the ambiguity of the

sensor. How can we define the meaning of a sensor? How much of the processing is part of the sensor? We know that the overall point spread function (PSF) is a two dimensional convolution of platform motion, lens, CCD array, and atmosphere:

$$h_{Overall}(x, y) = h_{Atmosphere}(x, y) * h_{Motion}(x, y) * h_{optics}(x, y) * h_{CCD}(x, y) \quad (3.1)$$

Should all the above elements be part of the channel? We are knowledgeable about some elements, but not others. Obviously, no one has control over atmospheric turbulence. Sometimes, a user may have control over the platform motion. Should the motion or the atmospheric turbulence be considered as a part of the source encoder or a channel? We have sufficient information about the optics and CCD array. Therefore, it may be logical to consider optics as an encoder rather than as a channel. Once we rethink the concept of image formation, opportunities open up for new approaches to image processing.

In the near future, image sensors will provide massive amounts of data, holographic storage devices will store massive amounts of data to provide temporal information to apply image processing with temporal signal processing, and multi-processor parallel computers and quantum computers will provide unthinkable processing power. Therefore, we seek to develop a systematic way to implement image processing using a strong information theoretic approach to improve image processing in the future.

3.3 Model Image Processing Based on Communication Theory

After rethinking the similarity between image processing and communication models, we conclude that the current image formation model relies heavily on the actual

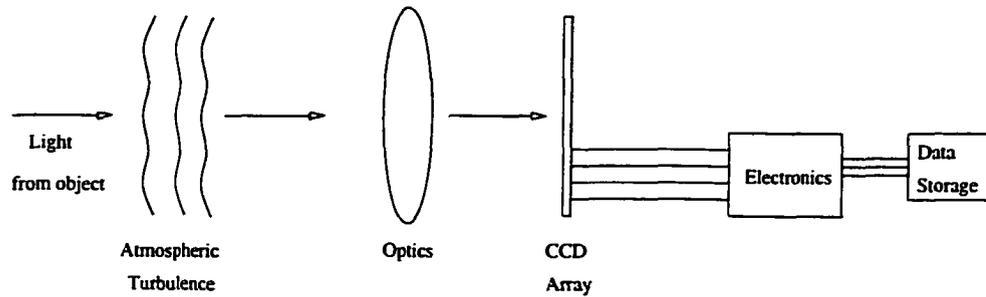


Figure 3.4: Image formation

physical implementation. Researchers fail to recognize that it is possible to organize the image formation and image processing models based on the prospect that a high level conceptual model can be similar to the communication model.

The actual physical image formation model is shown in Fig. 3.4. The light wave passes through the atmosphere and lens and is recorded in the CCD array. The electronic signal will be processed and will be stored in the data storage media. The overall point spread function (PSF) is a series of convolution processes. Equation (3.2) states that the overall PSF is a convolution of the PSF of the atmospheric turbulence, the PSF of lens, the PSF of CCD, and the PSF of the platform motion.

$$h_{Overall}(x, y) = h_{Motion}(x, y) * h_{Atmosphere}(x, y) * h_{lens}(x, y) * h_{CCD}(x, y) \quad (3.2)$$

This result is independent of the order of the convolutions. Therefore, we can rearrange the order to fit into the conceptual communication theory models. In Fig. 3.5, we can treat the CCD array as a communication channel. The communication channel adds noise to the system. On the other hand, in image formation, a sensor array adds noise into the stored image. This is the reason we treat the CCD array as a channel. All the known distortions can be treated as parts of the source encoder.

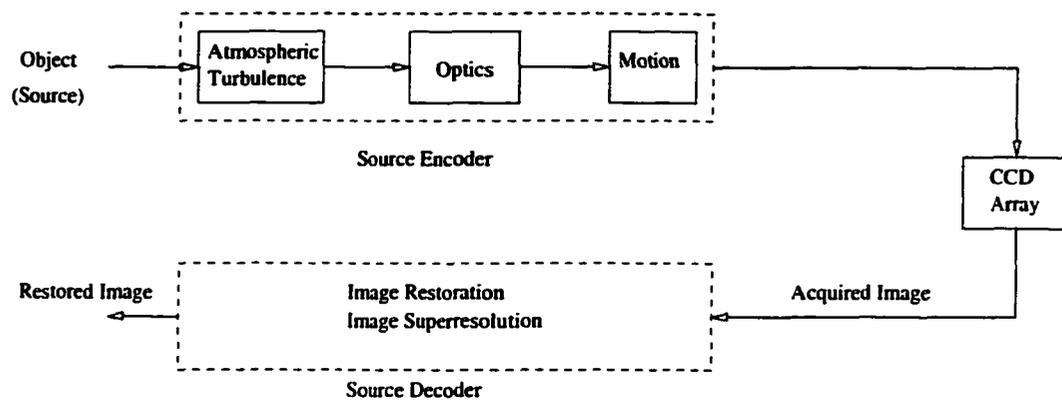


Figure 3.5: Communication theory based image formation model

Therefore, the image restoration/image super-resolution algorithms can be treated as parts of the source decoder.

If we compare Fig. 3.5 and Fig. 3.1, there is no channel encoder or channel decoder in this new model. The question we would like to ask is if a channel encoder exists in image formation.

In communications, the purpose of the channel encoder is to introduce, in a controlled manner, some redundancy in the binary information sequence that can be used at the receiver to overcome the effect of noise and interference encountered in the transmission of the signal through the channel.

It is possible to add a channel encoder in the image formation model. The channel of image formation is the CCD array. In a CCD array, thermal noise, *salt and pepper* noise, and other noises are added to the image. This is similar to adding noise in the communication channel. However, users cannot create redundancy in the image to detect error. Natural objects have special shapes and textures. Therefore, we can use this information as redundancy in the image to help the restoration algorithm

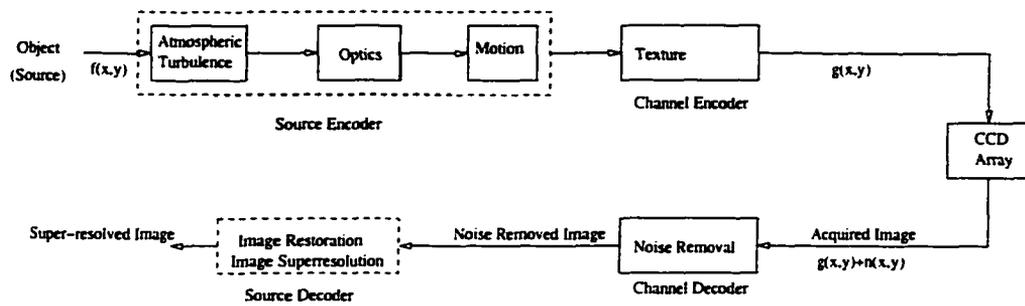


Figure 3.6: Improved image formation model based on communication theory

to remove noise in the image. The texture should be similar to the neighborhood of the position. It is possible to build a system to perform noise removal based on the texture of an object.

With the addition of texture information, a new image formation model is developed and shown in Fig. 3.6. This figure provides a new insight in incorporating the communication theory model into image processing. It also lays the foundation of chapters to come in this dissertation.

3.4 Introduction to the Proposed Dynamic Imaging System Model

The proposed new image processing model is designed in terms of an abstract formalization that is not connected to a specific physical problem. A direct implementation of Fig. 3.6 may not perform the best in a real situation. Atmospheric turbulence is difficult to measure on a surveillance airplane or surveillance satellite even though it is practical for a ground-based telescope. The platform motion can be a factor in some situations, but not all. An intelligent algorithm should be incorporated to optimize the performance of the new information theoretic image formation

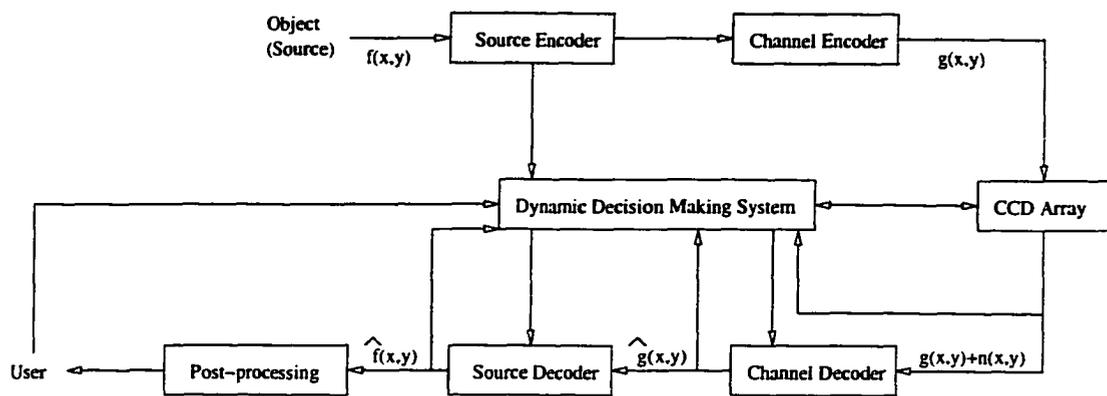


Figure 3.7: Dynamic Imaging System

and image processing model. Therefore, we introduce a new intelligent image model called the dynamic imaging system as shown in Fig. 3.7.

The details of the blocks shown in Fig. 3.7 are conceptualized as follows. The source encoder of the dynamic imaging system consists of atmospheric turbulence, optics, and platform motion as shown in Fig. 3.8. The turbulence is detected by a wavefront sensor. The motion of the imaging platform is detected by a motion sensor. The optics is based on the optical design of the lens. The main idea is that in this source decoder, the information loss is permanent. We treat this operation as a loss compression in the communication model. The channel encoder consists of the texture of the natural object as shown in Fig. 3.9. The texture is not controlled by the system design. However, we treat it as a component block. The channel decoder consists of a noise removal algorithm as shown in Fig. 3.10. The channel decoder removes the noise, n , generated in the CCD array. The source decoder consists of an image super-resolution algorithm as shown in Fig. 3.11. The source decoder performs an estimation of the information loss beyond the passband. The post-processing

consists of psychophysical image enhancement and information visualization as shown in Fig. 3.12. The psychophysical image enhancement helps the user to find the target in the image. The information visualization identifies the useful information for the user. The dynamic decision making system consists of a hybrid rule-based artificial intelligence (AI)/neural network decision making system, an image quality assessment, a feature extraction, and a set of task specific parameters. The hybrid rule-based AI/fuzzy neural network decision making system applies the rule-based AI algorithm to make decisions for all the known scenarios and applies the fuzzy neural network to make decisions for other uncertain scenarios. The image quality assessment algorithm determines the quality of the image and help the hybrid rule-based AI/fuzzy neural network to decide if it is necessary to apply noise removal algorithm. An image quality assessment is also applied to determine the quality of the super-resolved image. The feature extraction algorithm identifies the features in the image. The task specific parameter set is a set of user defined parameters for setting the hybrid rule-based AI/fuzzy neural network for making decision based on a specific task.

Figure 3.7 includes more components than Fig. 3.6. The involvement of the user is critical for the system to determine the optimal strategy to perform image restoration for particular missions. The following subsection will explain how the dynamic imaging system works.

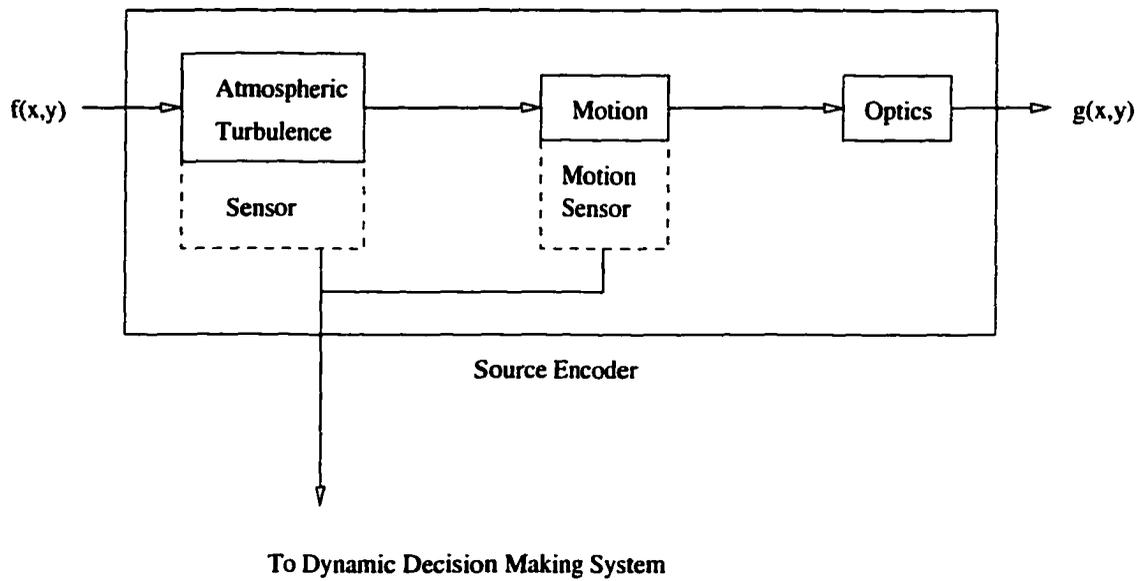
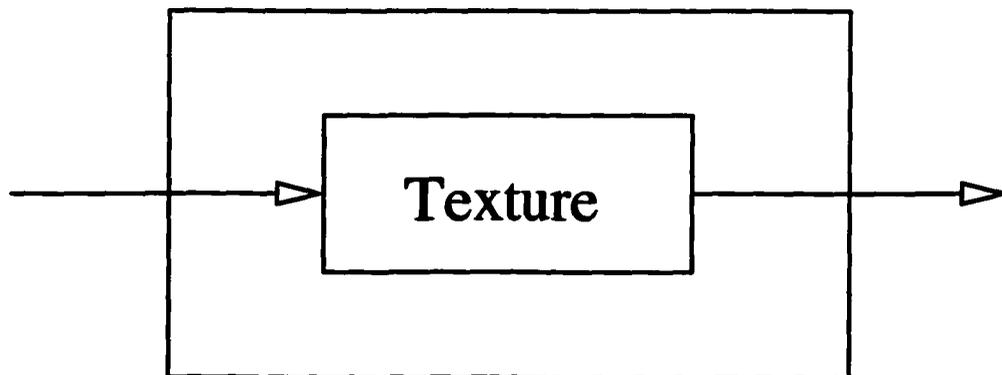


Figure 3.8: Source encoder



Channel Encoder

Figure 3.9: Channel encoder

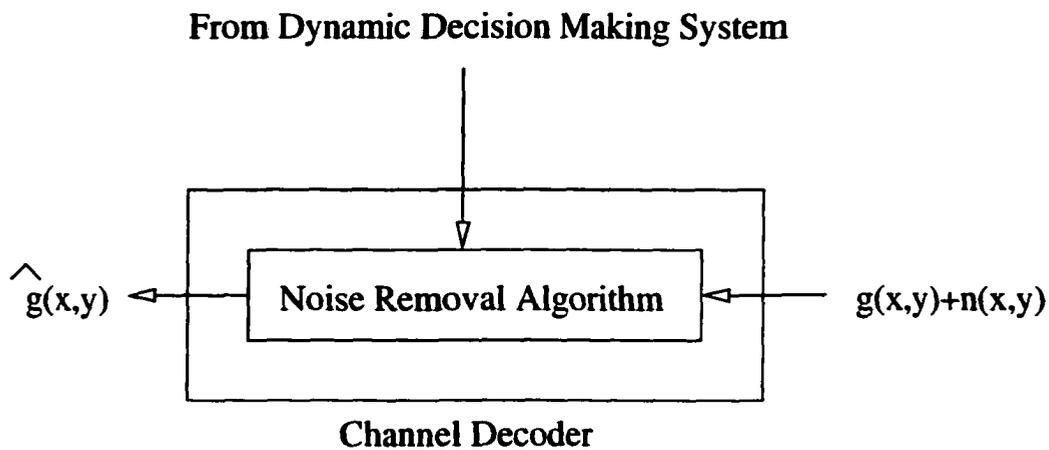


Figure 3.10: Channel decoder

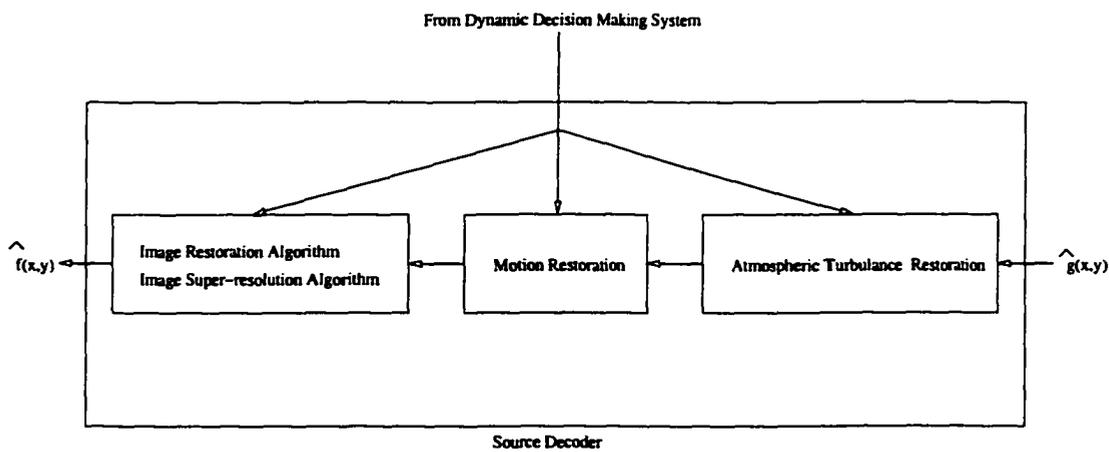


Figure 3.11: Source decoder

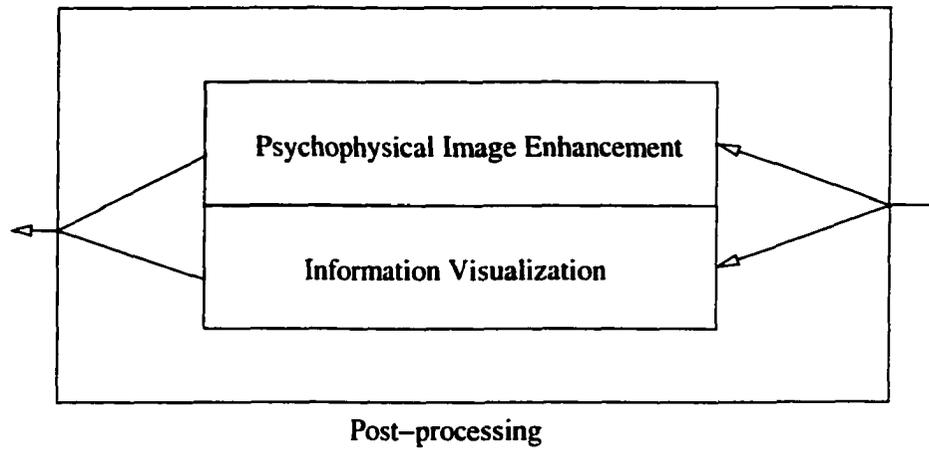


Figure 3.12: Post-processing

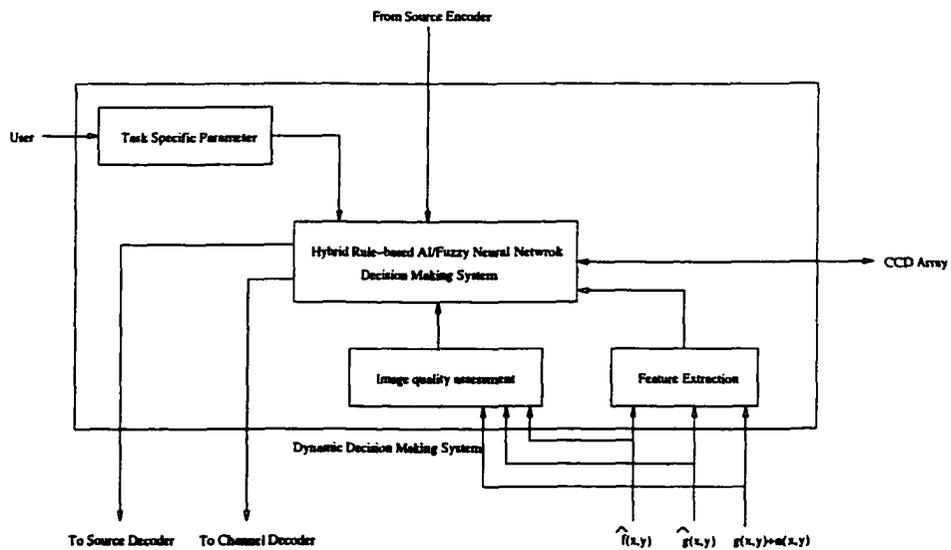


Figure 3.13: Dynamic Decision Making System

3.4.1 Constraint Imposed By the Application

Different applications have different constraints in image restoration. Therefore, the user must be part of the decision-making process. For instance, an astronomer has a fixed target in the sky. Therefore, he can use adaptive optics [22, 23]. If his telescope does not have adaptive optics, he can increase the sampling rate and apply multiframe image restoration algorithms [24, 25] to overcome the atmospheric turbulence. On the other hand, surveillance airplanes cannot take multiple pictures of a fixed target because they are moving at high speeds. Adaptive optics are also out of the question because an airplane's size and its relative high speed. In astronomy, a laser generated guiding star will be generated on top of the atmosphere. Any distortions that are visible to the wavefront sensor correspond directly to distortions somewhere in the atmosphere above the telescope. In order to use this information, the wavefront sensor separates the column of light into many areas or zones and samples each zone to determine how the light was altered by our atmosphere. This is not doable in other environments. We cannot create a point source in most situations. In this case, atmospheric turbulence correction is not possible. The above example clearly shows that an intelligent dynamic system is needed to optimize the result of image processing.

Much useful information can be extracted from an acquired image.

1. Image Quality.
2. Texture of objects.
3. Point Spread Function (assumed corrupted by the atmospheric turbulence).

4. Motion of the sensor.

Based on the user feedback, the dynamic image system can also adjust parameters for the dynamic decision system. To determine the quality of an acquired image, qualitative methods [26] to determine the image quality have to be developed. To assist the system, human user input is needed. The user is the only authority to determine if the output of the system is useful. Therefore, a user must be included into the decision making.

Psychophysical image enhancement and information visualization are not generally considered to be parts of image processing models. They do not increase information in the image. However, we found it was necessary to include them in the model. The goal of image processing is to improve the image quality, that is, to enable users to find useful information easily. For instance, recognizing a potential target is one general purpose of image processing. If psychophysical image enhancement and information visualization can help users find the useful information faster, it would be of great advantage in combat. For a radiologist, it is important to distinguish between tumor and normal tissue. An MRI film has 1024 gray levels which is beyond the capability of normal humans to use. However, a trained radiologist can see the difference when he tilts the film under a light source. An image post-processed by psychophysical image enhancement [27] can greatly increase its usefulness to the radiologists.

Information visualization [28] is also important. A pilot is used to being overloaded with information in the cockpit. A simple useful representation of the super-resolved target is more useful than the data alone. The goal of image super-resolution is

to identify the object from a low resolution data set. If the algorithm can identify the super-resolved object and categorize it, human users can leave the job to the computer and perform higher level operations. For a fighter pilot, it is important to distinguish between enemies and friends in the sky. For example, the F-22 and Joint Strike Fighter (JSF) have softwares to identify potential target and prioritize the potential threats. This is a good demonstration of information visualization.

With increasing computational power, it will soon be possible to develop a dynamic imaging system like the one shown in Fig. 3.7.

3.5 Components To Be Studied

The scope of this dynamic image system model is astonishing. Therefore, only three of its components will be studied in this dissertation. In the next three chapters, we will focus on the channel encoder, channel decoder, and source decoder. In chapter 4, a new Markov random field based noise removal algorithm will be presented. The texture of the blurred image can be used as channel encoded information. We can apply the Markov random field [29, 30] approach to investigate the relationship between the neighborhood pixels and use the information to remove noise in the texture. Therefore, the Markov random field noise removal algorithm functions as the channel decoder. The details of this approach and simulations will be presented. In chapter 5, a constrained limited maximum likelihood (ML) super-resolution technique based on the support of object is developed. The contour constraint of an object greatly increases the super-resolution performance of the ML algorithm. The contour constrained algorithm and validation experiments will be presented. This

algorithm is used as source decoder. In chapter 6, a new self-organizing map (SOM) based image super-resolution algorithm is also developed. A SOM functions as a cluster database to store similar pattern. A large enough database should provide accurate parallel processing image super-resolution in real time. The principle of this algorithm is demonstrated and is used as the source decoder.

CHAPTER 4

A MARKOV RANDOM FIELD BASED CHANNEL DECODER

4.1 Introduction

In this chapter, we will develop the channel decoder of the dynamic imaging system. The algorithm will treat the texture of the acquired image as the channel encoded information to overcome the distortion and noise of the channel. The texture, which is the spatial and visual context of the object, is deterministic in nature. The channel will be assumed to be the CCD array and related electronics in the imaging system.

Source decoding (restoration) of degraded images has been a topic of great interest for a number of years. More recently, a specific form of restoration known as super-resolution has been attracting considerable attention [31]. It is primarily aimed at correcting the degradation caused by diffraction-limited imaging and at providing a degree of spectral extrapolation (*i.e.*, attempting to recreate the spatial frequencies that are removed during the imaging process). Several powerful iterative super-resolution algorithms are being developed utilizing the frameworks of statistical optimization and set-theoretic estimation [31, 32, 33, 34]. The quality of the super-resolved image resulting from these iterative algorithms generally depends on

the initial conditions, *i.e.*, the starting image used to commence the iterative updating process. Since the iterative processing steps are generally computationally intensive, it would be wise to carefully pre-process the acquired image to achieve an optimal super-resolution result.

It is well known that restoration algorithms generally produce undesirable artifacts because of the ill-posed nature of the deconvolution problem. Subsequently, this may limit their usefulness in practical applications. In super-resolution processing, where the goal is the creation of new frequencies that are present in the object imaged but not in the image recorded, the artifacts resulting from the presence of noise need to be given special consideration. Due to the spectral mixing that takes place during the execution of each iteration, it is likely that the high frequency components of the noise become responsible for the expansion of the image bandwidth and appear as artifacts in the processed image. Removal of noise prior to super-resolution processing is therefore an important first step in these applications. Pre-processing for noise removal also helps to obtain a good starting image and, therefore, contributes to desirable convergence rates during further processing.

Selection of a channel decoder (noise removal filter) prior to source decoding (super-resolution processing) needs to be done carefully. Commonly employed noise removal filters [35] tend to smooth the image. They are successful in reducing the level of noise contamination, but they also reduce the resolution in the image. This works counter to the overall resolution enhancement goals of super-resolution processing. A specific approach that performs noise smoothing while attempting to maintain image details and preservation of edge structure is the Iterative Conditional Modes (ICM)

method that exploits characteristics of Markov Random Fields (MRF) in modeling the contextual information present in images [36, 37], as shown by Besag [38, 39] and Moussorius [40]. An elegant algorithm following this approach was recently developed by Park and Kurz [41] by employing a MRF with a second order neighborhood that is capable of extracting contextual information not only from pixel intensity levels but also from the relative position of the pixels. Unfortunately, the specific updating permitted in Park and Kurz [41] can only reduce the intensity value of a given pixel. Hence, it fails to adequately remove the noise effects in strongly corrupted images and leave open the possibility of severe artifacts in any subsequent super-resolution processing.

In this chapter, we shall present a modification to the ICM method that overcomes the limitations stated above. This modification utilizes a novel “relative potential function” to define the Gibbs distribution that underlies the MRF modeling of contextual information present in the image being processed. A new algorithm that is capable of executing both upward and downward adjustments to the pixel intensity levels is developed and is shown to perform a more efficient noise removal in comparison to earlier methods. The strong points of the proposed filter are further emphasized by applying it to a channel decoding, i.e., the filter is used to pre-process images corrupted with strong noise prior to super-resolution processing.

4.2 MRF Modeling and Channel Decoder

The image formation process is described by the convolution model

$$y = g + n = \{f * h\} + n \quad (4.1)$$

where y denotes the observed image of the object f , g denotes the image formed at the sensor through convolution with the sensor point spread function h , n denotes an additive noise, and $*$ denotes the convolution operation. Employing a lexicographic ordering of the pixels, we shall consider y , g , f , h and n as vectors of appropriate dimensions. The problem of interest is to design a processing scheme for the restoration and super-resolution of the image y which involves filtering of the noise and reversal of the blur caused by the convolution operation. As noted about in section 4.1, the present scheme is characterized by a pre-filtering operation executed by a suitably designed ICM filter and is followed by a super-resolution processing of the output of the ICM filter.

In this section we shall briefly outline the basics of MRF modeling and the ICM filtering algorithm (developed by Besag [38], Moussorius [40], Park and Kurz [41] and others) that will be essential to describe the limitations of this method in the next section. The procedure described in this chapter will attempt to overcome these shortcomings. A center pixel and its eight neighbors form a second order neighborhood system, and as depicted in Fig. 4.2, 24 different cliques* can be formed between a center pixel and its neighbors. The specific algorithm developed by Park and Kurz [41] uses the shape of cliques derived from a second order neighborhood system for each pixel to deliver contextual information in terms of the absolute intensity difference between a given pixel and its neighbors.

*A clique is a group of pixels containing the center pixel and a selected number of pixels in the neighborhood.

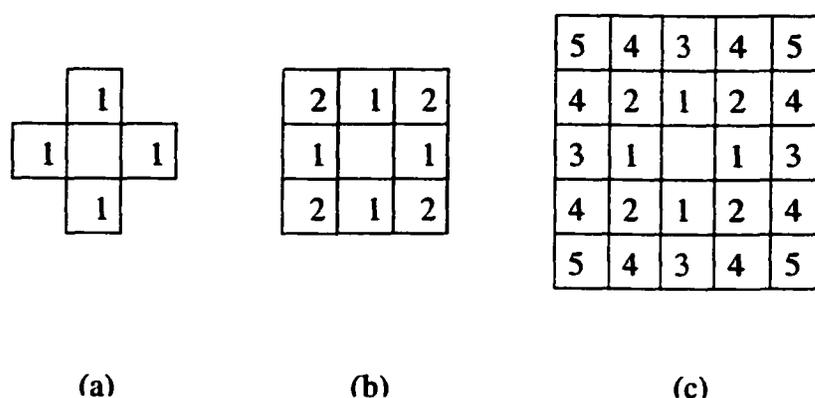


Figure 4.1: (a) First order neighborhood. (b) Second order neighborhood. (c) Fifth order neighborhood.

It is necessary to define the order in a neighborhood. In the first order neighborhood, every center pixel has four neighbor pixels as shown in Fig. 4.1a. The first order neighborhood is also called the 4-neighborhood system. The second order neighborhood, shown in Fig. 4.1b, is also called the 8-neighborhood system. Every center pixel has eight neighbor pixels. The fifth order neighborhood is shown in Fig. 4.1c. The order of neighborhood system is determined by the outermost neighboring pixels from the center pixel as demonstrated in Figs. 4.1a, b, c.

In all of the remaining discussion in this chapter, we shall exclusively consider the cliques formed in a second order neighborhood system.

All images are defined on a $M \times N$ rectangular lattice

$$L = (i, j) : 0 \leq i \leq M - 1, 0 \leq j \leq N - 1 \quad (4.2)$$

We simplify the notation by changing (i, j) to s with $s = Ni + j$. For a concise description of the updating algorithm for noise removal, letting g_s denote the intensity value of a pixel at a site s on the lattice L . Let N_s denote the set of its neighborhood

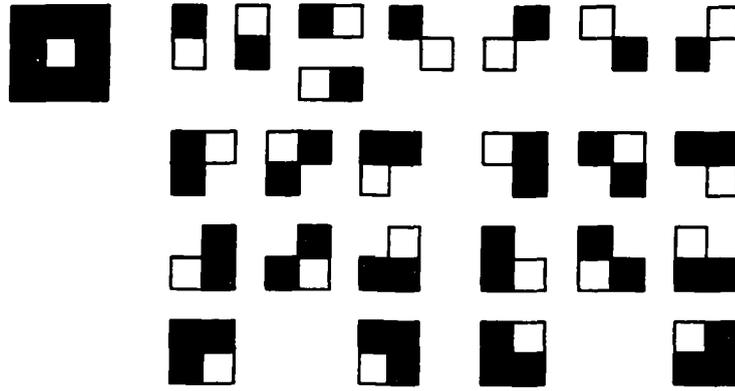


Figure 4.2: Second order neighborhood and 24 cliques.

pixels within a specific clique c , and let Q_s denote the set of cliques for the pixel under consideration. Also, let $g_{\partial s}$ denote the vector of the random variables in N_s . Besag [38] proposed the ICM method as a reasonable solution to the *Maximum a Posteriori* (MAP) estimation for noise removal that maximizes the conditional probability $P_s(g_s|g_{\partial s})$. In other words, the filtered output \hat{g}_s that is a noise removed image is obtained as

$$\hat{g}_s = \arg \max_{g_s} P(y_s|g_s)P_s(g_s|G_{\partial s}) \quad (4.3)$$

which readily leads to the simple updating equation in the iteration process

$$\hat{g}^{k+1}(y_s|g_s) = \hat{g}^k(y_s|g_s) \cdot P_s(g_s|G_{\partial s}) \quad (4.4)$$

In Eq.(4.4), y_s denotes the pixel value in the acquired image y at location s , g_s denotes the true pixel value in the image formed at the sensor (prior to contamination by noise), and \hat{g}^k denotes the estimate formed by the ICM filter at the k -th iteration.

Park and Kurz [41] employ in their algorithm a model for the conditional probability $P_s(g_s|G_{\partial s})$ in the form of a Gibbs distribution [42]

$$P_s(g_s|G_{\partial s}) = \exp\left(-\frac{1}{T}\Phi(s)\right) \quad (4.5)$$

where Φ is an “energy function” and T denotes a “temperature” variable. The energy function Φ is obtained from summing a set of “potential functions” $V_c(\cdot)$ developed for each clique c , *i.e.*

$$\Phi(s) = \sum_{c \in Q_s} V_c(g(s)) \quad (4.6)$$

with the potential function $V_c(\cdot)$ defined as

$$V_c(g(s)) = \left| g(s) - \frac{1}{n_c} \sum_{t \in c} g(t) \right| = |g(s) - \bar{g}(s, c)| \quad (4.7)$$

In Eq.(4.7), $\bar{g}(s, c)$ denotes the mean of the neighboring pixel intensities in clique c .

The temperature T is selected as

$$T = \sigma_s \times n_c \quad (4.8)$$

where σ_s is a standard deviation of $\bar{g}_{s,c}$ in Q_s , Q_s being a set of cliques, and n_c is the number of sites in cliques associated with c . The T is decreasing because σ_s reduces as the number of iterations increase. The noise in the center pixel will be decreased. Therefore, the difference between $g(s)$ and the pixels in the clique will be reduced. Therefore, the simulated annealing method [42, 43, 37] is used in this approach. Even though the Gibbs distribution specified in Eq.(4.5) is rather well known, the novelty in the use of cliques in modeling the potential function V_c given by Eq.(4.7) enables

one to capture the contextual information in terms of the intensity difference between a given pixel and its neighbors. As part of their algorithm, Park and Kurz [41] also gave an outlier rejection scheme based on a predetermined threshold value T_{OL} . In this scheme, an outlier candidate y_k will be rejected if the absolute difference between this candidate and the mean μ_c of four mid-positioned pixels is larger than T_{OL} , i.e.

$$|y_k - \mu_c| \geq T_{OL} \implies \text{reject } y_k.$$

For scanning the lattice, in the x direction, we scan from 1 to $M - 2$ in the lattice. In the y direction, we scan from 1 to $N - 2$ in the lattice. The outermost pixels will not be scanned because they do not have neighborhood pixels on one side. Any clique that contains an outlier will not be counted in the potential function calculation for a given site.

4.3 Limitations of the ICM Algorithm

In our dynamic imaging system, the channel decoder is an algorithm to remove the alternating current (AC) noise accumulated in the channel that consists of the CCD and the electronic components of the imaging system. The ICM filter removes noise while preserving image details. Although the ICM algorithm described above provides a reasonably good channel decoder, it has several shortcomings that may reduce its efficiency in handling strongly corrupted images and in providing a filtered image that is suitable for subjecting the result to further processing steps. For illustration, consider a scenario where some of the noisy pixels have intensity values smaller than most of the surrounding pixels. This situation is quite common in a strongly corrupted image. Since $0 \leq P_s(g(s)|G_{\partial s}) = \exp(-\frac{1}{T}\Phi) \leq 1$, Eq.(4.4) can

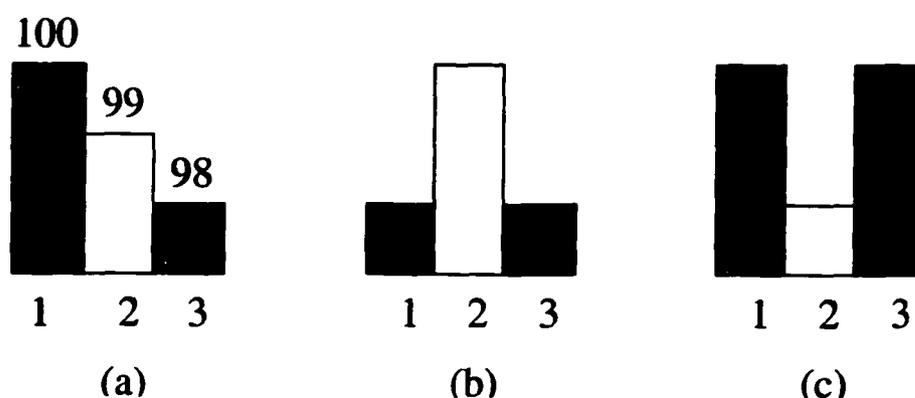


Figure 4.3: Typical slides in intensity function. (a) A downward slope. (b) A positive value noise on a flat surface. (c) A negative value noise on a flat surface

only reduce the values of these pixels at the subsequent updates. Thus, if a given pixel has a lower value than the neighborhood, the intensity value will only keep decreasing until $|y(k) - \mu_c| \geq T_{OL}$. This effect causes that pixel to be rejected and to be replaced by the value μ_c , the average of the four mid-positioned pixels. Thus the Park and Kurz algorithm will not estimate the true pixel value. What is needed to improve the result in this case is the flexibility to revise the intensity value upward, in addition to the downward revision permitted by the algorithm.

The following example more clearly describes the limitations of the algorithm even in a noise-free environment. Figure 4.3a shows a cross section of the intensity values of three adjacent pixels with intensity values decreasing uniformly from left to right, with the value of pixel 2 being the average of the values of pixels 1 and 3. With the potential function $V_c(g_s)$ defined as in Eq.(4.6), the energy function will have a value 2. This leads to a conditional probability value for pixel 2 to be less than 1. The resulting updated pixel value \hat{g}_s will be less than 99 even though the intensity value

99 is correct for this pixel. Consequently, the updating performed by the algorithm (given by Eq.(4.4)) results in a decrease in the pixel value and moves that result away from its true value. This example clearly demonstrates the shortcoming of the ICM algorithm, specifically arising from the requirement of choosing a potential function that can attain only positive values.

Yet another example of the limitations of the ICM algorithm can be given using an image contaminated with a strong "salt and pepper" noise. Consider the cross section plot in the intensity values of adjacent pixels shown in Figs. 4.3b and 4.3c that are caused by noise. The value of pixel 2 should be the same as those of the adjacent pixels, i.e., pixels 1 and 3. However, the pixel 2 value is changed by the "salt and pepper" noise. In processing an image with the cross section shown in Fig. 4.3b, the updating given by Eq.(4.4) works well in reducing the value of pixel 2 progressively until it is the same as the neighboring pixel values. However, in processing an image with the slide shown in Fig. 4.3c, this is not the case as the updating by the algorithm can only reduce the pixel value. Thus the value of pixel 2 is further reduced in comparison with its neighbors, eventually causing its rejection as an outlier.

4.4 Relative Potential Function and an Improved ICM Algorithm For Channel Decoding

In classical mechanics, an object at a lower (higher) altitude is considered to have a lower (higher) potential energy. An observer located at an altitude between these two values will characterize the object at the higher altitude with a positive potential

difference relative to the observer, while the object at the lower altitude will be characterized by a negative potential difference relative to the observer. Using this analogy, the potential function $V_c(\cdot)$ used in defining the conditional probability for the pixel intensity updating in Eq.(4.4) can be treated as a measure of the relative potential difference between a given pixel and its neighborhood cliques. The relative potential difference, of course, can attain both positive and negative values. In other words, one can define a new potential function [44] for each clique c within the neighborhood of a given pixel $g(s)$ as

$$W_c(g(s)) = g(s) - \frac{1}{n_c} \sum_{t \in c} g(t) = g(s) - \bar{g}(s, c) \quad (4.9)$$

Therefore, the energy function ϕ has to be modified as

$$\Phi(s) = \sum_{c \in Q_s} W_c(g(s)) \quad (4.10)$$

We will refer to Eq.(4.9) as the "relative potential function". Use of this $W_c(\cdot)$ to define the energy function Φ as the sum of the relative potential functions developed for each clique creates a problem. Since Φ can attain positive or negative values, the conditional probability $P_s(g(s)|g_{\partial s})$ given by Eq.(4.5) will not be a Gibbs distribution. Hence, we shall modify the definition of the conditional probability $P_s(g(s)|g_{\partial s})$ to accommodate the new relative potential function by setting

$$P_s(g(s)|g_{\partial s}) = \exp\left(-\frac{1}{T}|\Phi|\right) \quad (4.11)$$

A key mathematical step in using a Gibbs distribution function to model an MRF and to further utilize it to tailor an ICM algorithm is to establish the equivalence

between the Gibbs random field defined by the chosen distribution function and the corresponding MRF. The importance of this connection was shown originally by Hammersley and Clifford [45] and was utilized later by researchers in the development of systematic and simple-to-implement algorithms. In particular, the Gibbs-Markov equivalence enables one to establish the equivalence between local properties (correlation between a pixel and its immediate neighbors, for instance) and global properties (correlation between a pixel and all other pixels in a lattice, for instance) and this equivalence permits the use of conditional probabilities to model the MRF. It is desirable to demonstrate the Gibbs-Markov equivalence for this new channel decoder (ICM algorithm) based on the Hammersley-Clifford theorem. However, the best we could find was the upper bound and the lower bound for the Gibbs-Markov equivalence. In some cases, the Gibbs and the Markov distribution are equivalent. In other cases, the Gibbs distribution is within the upper and lower bound. The Gibbs-Markov equivalence exists when $\Phi \geq 0$. The Gibbs random field is bounded between zero and one when $\Phi < 0$. For the relative potential function given by Eq.(4.11), an extension of this theorem establishes the needed Gibbs-Markov equivalence and the upper and the lower bound for the Gibbs random field. *Theorem 1:* A Gibbs random field with the relative potential function described by Eq.(4.11) is an MRF with the same distribution function in some cases. In other cases, a Gibbs random field is still bounded between zero and one. A proof of this theorem, given here for the sake of completeness, is relegated to Appendix A since the detailed steps in the proof are not essential for the further development of the algorithm. It should be emphasized that for a distribution function for which the Gibbs-Markov equivalence cannot be

established, calculation of the MRF is complex. For details on the MRF computation using Gibbs distributions, one may refer to Besag [38] or Moussorius [40].

The benefits of employing the relative potential function in defining the conditional probability can be easily seen by reexamining the intensity slide behavior shown in Fig. 4.3a. Due to the equal but opposite potential differences of pixel 2 relative to its two neighbors, $\Phi = 0$. This results in $P_s(g(s)|g_{\partial s}) = 1$. Hence, the updating achieved by the ICM algorithm will not, as desired in this case, modify the intensity value of pixel 2. Thus, the definition of the relative potential function as given by Eq.(4.9) overcomes a major limitation of the ICM algorithm.

To overcome the other limitations of the ICM procedure that result from the constraint of revising the intensity value of a pixel only downwards, the updating rules need to be appropriately modified. This is facilitated by our definition of Φ as the sum of the relative potential functions $W_c(\cdot)$ and the signal upward or downward adjustment to the intensity value corresponds to whether Φ has a negative or positive value. To implement the needed rule adjustments, we employ the modified updating rules [44] for the ICM algorithm given by

$$\begin{aligned} \hat{g}_{k+1}(y(s)|g(s)) &= \hat{g}_k(y(s)|g(s)) \cdot P_s(g(s)|g_{\partial}(s)) \text{ if } \Phi \geq 0 \\ &= \hat{g}_k(y(s)|g(s)) \cdot \frac{1}{P_s(g(s)|g_{\partial}(s))} \text{ if } \Phi < 0 \end{aligned} \quad (4.12)$$

It may be noted that, since $P_s(g(s)|g_{\partial s}) < 1$, the updating in Eq.(4.12) implements a downward revision of pixel intensity when $\Phi \geq 0$. In this case, the Gibbs-Markov



Figure 4.4: Lenna.

equivalence exists. When $\Phi < 0$, $\frac{1}{P_s(g(s)|g_{\theta s})} \geq 1$. The updating in Eq.(4.12) implements an upward revision of the pixel intensity. In this case, the Gibbs-Markov equivalence does not exist. However, the bound of the Gibbs random field exists; the Gibbs random field is bounded between zero and one. These bounds make $\frac{1}{P_s(g(s)|g_{\theta s})} \geq 1$. The bounds ensure that Eq.(4.12) works.

In order to demonstrate the noise removal performance of the presently developed ICM filter, the results of the following experiments are presented. The test object is the famous Lenna image as shown in Fig. 4.4. A 64×64 portion of the Lenna image (shown in Fig. 4.5a) was blurred by convolving it with a low-pass filter. The resulting blurred image Fig. 4.5b was further degraded by adding “salt and pepper” noise as shown in Fig. 4.5c. This image was used as the starting image for further processing.

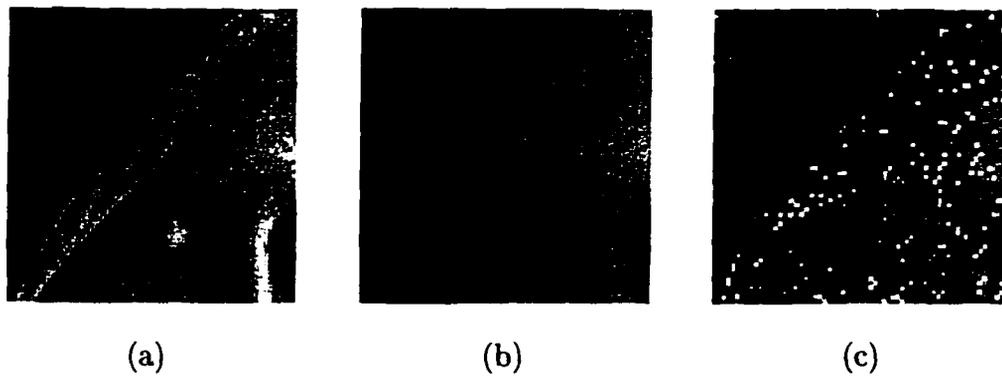


Figure 4.5: (a) A portion of an original Lenna image. (b) A blurred portion by a low pass filter. (c) Adding “Salt and Pepper” noise in the blurred portion of Lenna

Figures 4.6a, 4.6b and 4.6c, respectively, show a comparison of the results of filtering the test image in Fig. 4.5c with a 3×3 median filter, with the Park and Kurz algorithm [41], and with the present algorithm. To obtain a quantitative comparison, each of the filtered images in Figs. 4.6a, 4.6b and 4.6c was compared with the original blurred image shown in Fig. 4.5b and the mean square error (MSE) was computed. The purpose of using the blurred noise free image Fig. 4.5b was to show the noise removal performance of all three filters. These three filters only remove noise and do not perform any image restoration or image super-resolution. Therefore, we cannot compare the processed image with the original image Fig. 4.5a, but rather with the blurred noise free image Fig. 4.5b. The MSE values for the comparisons came out to be 134.42 for the image filtered with the 3×3 median filter, 125.65 for the image filtered with the Park and Kurz algorithm, and 88.44 for the image filtered with the present algorithm. This result clearly demonstrates the superior noise removal performance of the proposed filter.

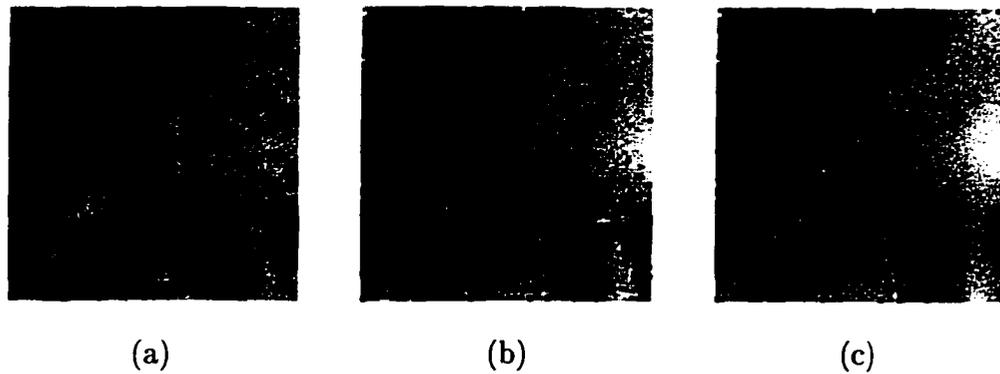


Figure 4.6: (a) A 3×3 median filtered image. (b) ICM filtered image. (c) Proposed filtered image

In order to verify the noise removal performance of the proposed filter, the results of the following experiment are presented in addition to the experiments on the set of Lenna images. The test object is the tank image shown in Fig. 4.7. A 64×64 portion of the tank image (shown in Fig. 4.8a) was blurred by convolving it with a low-pass filter and the resulting blurred image Fig. 4.8b was further degraded by adding “salt and pepper” noise. The blurred test image with “salt and pepper” noise is shown in Fig. 4.8c. This image was used as the starting image for further processing. Figs. 4.9a, 4.9b and 4.9c, respectively, show a comparison of the results of filtering the test image in Fig. 4.9c with a 3×3 median filter, with the Park and Kurz algorithm [41], and with the present algorithm. To obtain a quantitative comparison, each of the filtered images in Figs. 4.9a, 4.9b and 4.9c was compared with the original blurred image shown in Fig. 4.8b and the mean square error (MSE) was computed. The purpose of using the blurred noise free image Fig. 4.8b was again to show the noise removal performance of all three filters. The MSE values came out

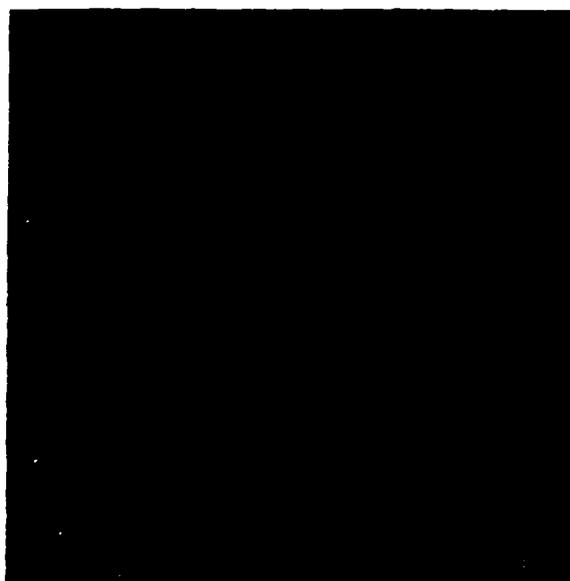


Figure 4.7: Tank.

to be 184.5550 for the image filtered with the 3×3 median filter, 64.2853 for the image filtered with the Park and Kurz algorithm, and 19.2329 for the image filtered with the present algorithm. This result again clearly confirms the superior noise removal performance of the proposed filter.

A further confirmation of the efficacy in noise filtering with the proposed algorithm was obtained by using the three images in Figs. 4.6a, 4.6b and 4.6c as starting points for implementing a super-resolution algorithm aimed at removing the blurring and restoring the original image in Fig. 4.5a. These results will be presented in the next section.

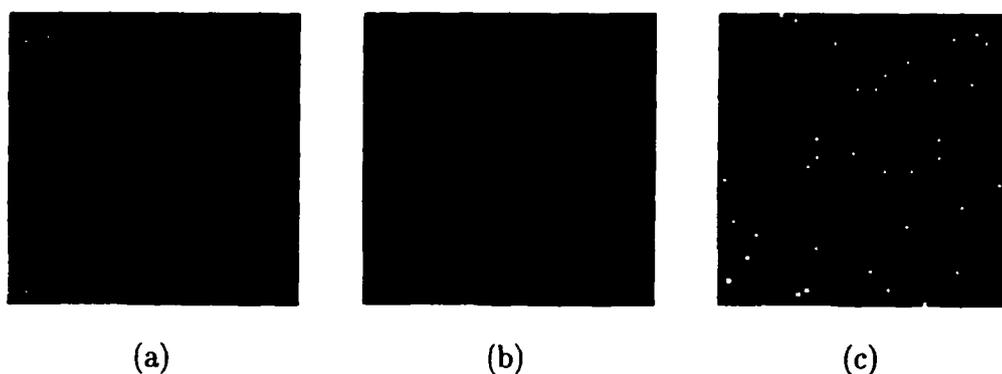


Figure 4.8: (a) A portion of an original tank image. (b) A blurred portion by a low pass filter. (c) Adding “Salt and Pepper” noise in the blurred portion of the tank image

4.5 Application of the Source Decoder to the Channel Decoded Image

In this section, we shall demonstrate the noise removal performance of the presently developed channel decoder (ICM filter) to provide a good starting condition for performing source decoding (super-resolution processing). A specific super-resolution algorithm was selected that has been receiving considerable attention recently. As noted in section 4.1, several powerful iterative restoration algorithms have been developed in a statistical optimization framework. Both the *Maximum a Posteriori* [31] and the *Maximum Likelihood* blind deconvolution [32] algorithms are based on statistical optimization framework. In the *Maximum a Posteriori* [31] algorithm, the prior density of the object f is assumed to be known. In a *Maximum Likelihood* blind deconvolution [32], the prior density of the object f is assumed to be unknown. On the other hand, the projection on Convex Set (POCS) [46] approach is based on a set-theoretic estimation framework.

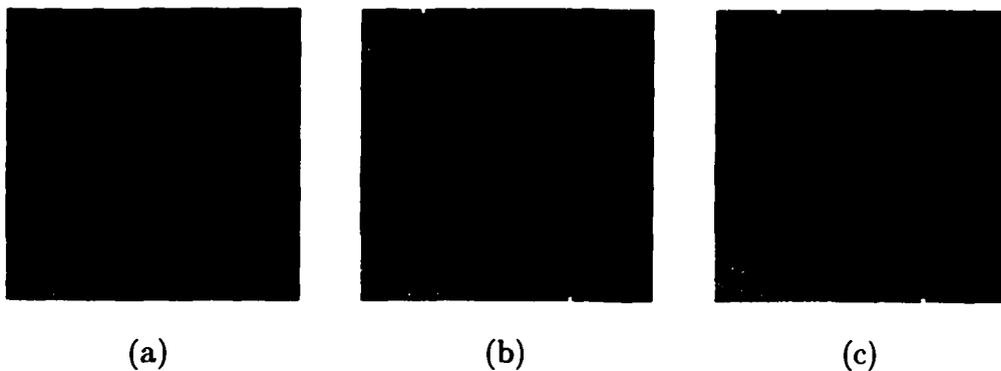


Figure 4.9: (a) A 3×3 median filtered image. (b) ICM filtered image. (c) Proposed filtered image

The fundamental idea behind the various approaches to design restoration algorithms is the attempt to solve the inverse problem of reversing the effects of convolution in the image formation process, as given in Eq.(4.1). This is accomplished by iteratively constructing the estimated image from the observed data using some reasonable assumptions on the point spread function of the sensor, or by intelligently utilizing some known constraints for correcting the estimates obtained at each iteration.

We can construct the super-resolved image \hat{f} by processing the ICM filter output \hat{g} . The iterative restoration algorithm that is generally referred to as the Richardson-Lucy algorithm [14, 15] attempts to maximize the likelihood function $p(g/f)$, modeled as a Poisson distribution function, by implementing the updating rule

$$\hat{f}^{k+1}(i, j) = \hat{f}^k(i, j) \left\{ \left[\frac{\hat{g}(i, j)}{h(i, j) * \hat{f}^k(i, j)} \right] \odot h(i, j) \right\} \quad (4.13)$$

where \hat{f}_k denotes the estimate constructed at the k -th iteration and, h is the point spread function (PSF). For the first iteration, the acquired image \hat{g} was chosen as \hat{f}^0 . A one-dimensional example is shown in Fig. 4.10. The space domain signals are shown above the data flow. The frequency domain signals are shown below the data flow. The plots in this example clearly show that the ML algorithm can super-resolve data and can estimate the lost information outside the passband. Unfortunately, the ML algorithm derivation by Richardson [14] and Lucy [15] is not fundamentally sound for image processing because the whole proof was solely based on a Bayesian approach without considering the detection of the photons. An alternative derivation based on Poisson probability and a Bayesian approach is given in Appendix B. The super-resolution performance of this algorithm is discussed by Pang et. al. [32].

For comparing the efficiency of the presently developed ICM algorithm in filtering noise and providing a good starting image $\hat{f}^0(j)$ to commence the iterations specified by Eq.(4.13), each of the images shown in Figs. 4.6a, 4.6b, and 4.6c were subject to 10 iterations of the above algorithm. The resulting images at the end of 10 iterations are shown in Figs. 4.11a, 4.11b, and 4.11c, respectively. Visually, Fig. 4.11c has the best quality. It is followed by Fig. 4.11b, and then by Fig. 4.11a.

To further provide extreme cases for this comparison, the noisy blurred image Fig.4.6c was processed by the ML algorithm, *i.e.*, Fig.4.6c was used as the starting image $\hat{f}^0(j)$ for the ML algorithm. The results of processing with the original noise-free blurred image, Fig. 4.6b, used as the starting image $\hat{f}^0(j)$, were also obtained. These, resulting at the end of 10 iterations in each case, are shown in Figs. 4.11d

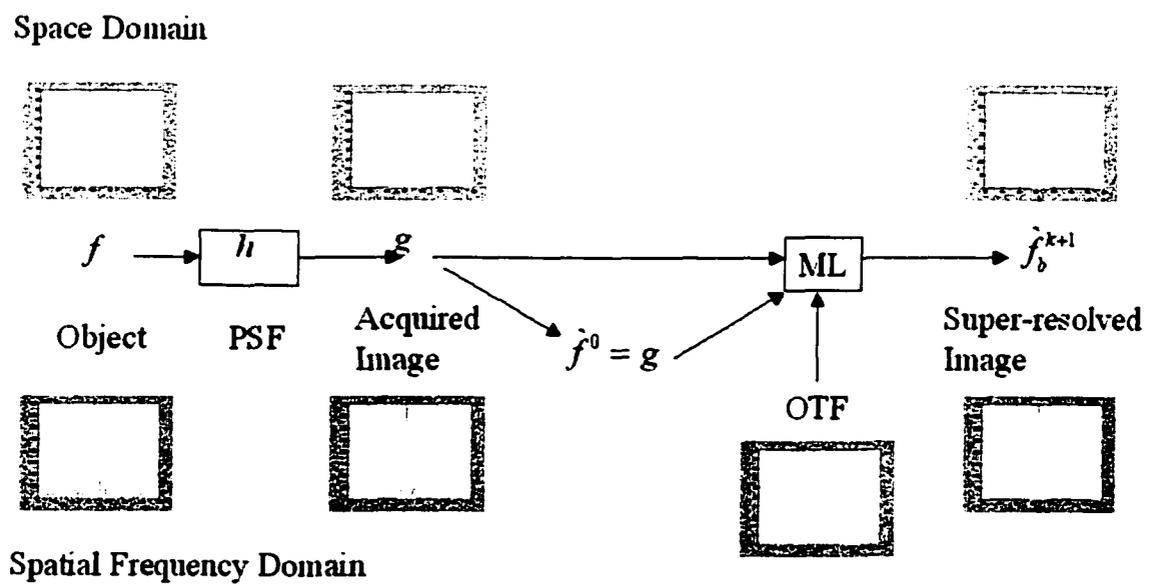


Figure 4.10: An one-dimensional example of the maximum likelihood algorithm

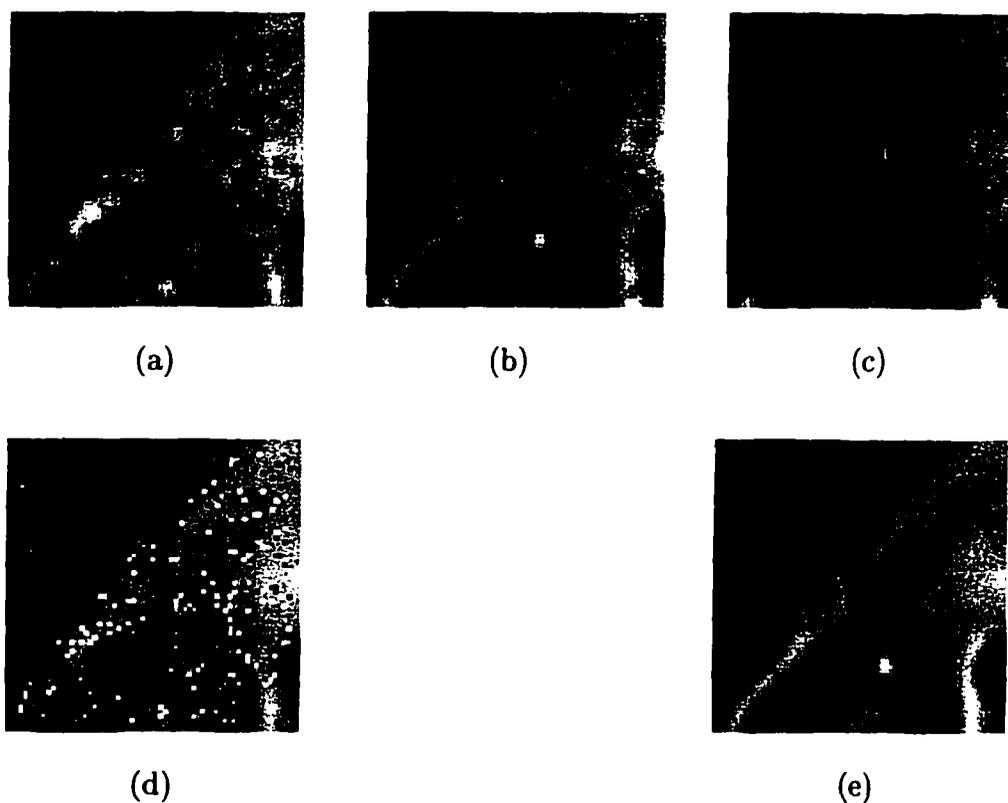


Figure 4.11: Maximum likelihood post-process filter output images. (a) Super-resolved 3×3 median filter output. (b) Super-resolved ICM filter output. (c) Super-resolved the proposed filter output. (d) Super-resolved noisy blurred image. (e) Super-resolved blurred image.

and 4.11e, respectively. For obtaining a quantitative comparison, the MSE was calculated between the original image shown in Fig. 4.5a and each of the images shown in Figs. 4.11a, 4.11b, 4.11c, 4.11d and 4.11e. The MSE values: 729.4, 539.5, 518.3, 1484.4 and 365.8 were obtained, respectively. The reduced MSE value of 518.3 attests to the noise removal efficiency of the present ICM filter. The ML super-resolved result of noise free blurred image Fig. 4.11e as expected, has the lowest MSE value,

365.8. The ML super-resolved result of the proposed algorithm's noise removal image Fig. 4.11c has the second lowest MSE value, 518.3. This means that the noise removed result from the proposed algorithm is very close to the noise free blurred image. This is highly desirable. The ML super-resolved result of the ICM algorithm's noise removal image Fig. 4.11b has the third lowest MSE value, 539.5. This means that the noise removed result from the ICM algorithm is close to the proposed algorithm's performance. However, it is not as good as the performance of the proposed algorithm.

The ML super-resolved result of the 3×3 median filter algorithm's noise removal image (Fig. 4.11a) has a high MSE value, 729.4. When we compare Fig. 4.11a and Fig. 4.6a visually, there is not much difference between them. This implies that the statistical property of the clear object was totally lost. The underlying statistical property of the noise blurred image was totally destroyed by the 3×3 median filter. The ML super-resolved result of the noise blurred image Fig. 4.11d has the highest MSE value, 1484.4. If we examine it visually, we find that it has a high MSE value due to the noise artifact. However, we find theoretically that it has a better image structure than does Fig. 4.11a. This means that most of the statistical property remains despite the presence of "salt and pepper" noise.

To further demonstrate the reliability of the spectrum extrapolation facilitated by the presently developed modified ICM filter, the power spectra of the super-resolved images were computed and compared. The benchmark used in this comparison was obtained from the following experiment. The original "Lenna" image was initially

blurred by convolving it with a known PSF that is defined in Eq.(2.2) and by subjecting the resulting blurred image to 10 iterations of the super-resolution algorithm specified by Eq.(4.13). The power spectra of the original image, the blurred image, and the super-resolved image are shown in Figs. 4.12a, 4.12b, and 4.12c, respectively. The contraction in the spectrum due to blurring is evident by comparing Figs. 4.12a and 4.12b. These two power spectra show how much information is lost during the imaging process. On the other hand, the expansion of the spectrum due to super-resolution processing can be noted by comparing Figs. 4.12b and 4.12c. Since this part of the experiment was conducted in the absence of noise, the spectrum shown in Fig. 4.12c serves as a benchmark for comparing the efficiency of a noise-removal filter if the blurred image to be super-resolved is contaminated with noise. Fig. 4.12c represents how much information can be correctly estimated by the ML algorithm with a noise free blurred image. The super-resolved image's power spectrum should be similar to Fig. 4.12c.

To test this performance of the noise-removal super-resolution algorithm, the blurred image was further degraded by adding "salt and pepper" noise. The resulting image was pre-processed with a 3×3 median filter, an ICM filter following the Park and Kurz algorithm, and the presently developed modified ICM filter prior to executing 10 iterations of the super-resolution algorithm specified by Eq. (4.13). The power spectra of the resulting images are shown in Figs. 4.12d, 4.12e, and 4.12f. Comparing these with the spectrum shown in Fig. 4.12c, one can draw the following conclusions. Random noise exists everywhere in Fig. 4.12d, confirming that the median filter has not performed an efficient noise removal. In fact, it has created

more high frequency, inaccurate information when it removed noise from the blurred image. In return, those inaccuracies gave rise to high frequency noise artifacts. Some random noise is still present in Fig. 4.12e in addition to two prominent lines of noise signals along the x- and y-axes. These effects highlight the shortcomings of the Park and Kurz algorithm. On the other hand, the absence of a high degree of noise in Fig. 4.12f and the similarity of the spectral components to those in Fig. 4.12c attest to the noise removal efficiency of the present algorithm. Even though some high frequency noise still exists in Fig. 4.12f, the noise is much weaker when compared to the noise in Fig. 4.12e. It also illustrates the reliable spectrum extrapolation that would result from super-resolving the pre-processed noisy image with this filter. In Fig. 4.11, we used the MSE to measure the quantitative accuracy of the restored image. We found that it is difficult to use the MSE in a power spectrum measurement as a quantitative measure of goodness. Therefore, we only compared the power spectra visually. It is often acceptable in the image processing community to compare power spectra visually to determine the performance of a super-resolution algorithm.

Since Figs. 4.12d, 4.12e, and 4.12f only show the expansion of the power spectra, they do not provide any measure of how much information is correctly estimated. Clearly, these images cannot have more information than Fig. 4.12c. Since Fig. 4.12c is the ML algorithm super-resolved image of the noise free blurred image, Fig. 4.12c represents the best that the ML algorithm can do with the acquired data. Any more information than Fig. 4.12c shows is just a noise artifact.

To obtain a quantitative measurement, one can calculate the correlation coefficient between the Fourier transform of the original object and the Fourier transform of the

restored image as a way to quantify the accuracy of the restored information. The super-resolved 3×3 filtered image has less accurate information than super-resolved ICM and the proposed image. Both the super-resolved ICM filtered image and the super-resolved proposed filtered image have fairly accurate information estimates. Through careful observation, one can notice that the super-resolved proposed filtered image has a slightly more accurate estimate of the information.

Since it is difficult to quantify the result of these correlation images, the *mean-correlation-coefficient* (MCC) was developed to compare the image similarities of the images. The MCC is given by the expression

$$MCC = \frac{1}{M \times N} \sum_{x=1}^M \sum_{y=1}^N r(x, y) \quad (4.14)$$

where r is the correlation coefficient between an 8×8 block of power spectrum of an original image and an 8×8 block of power spectrum of a blurred image. The original image and the restored image size are 64×64 in size. Therefore, one does not have enough data to calculate the correlation coefficient in the frequency domain. A simple technique was applied to create enough data points. If one flips the image twice in two different directions, the resulting image has 128×128 pixels as shown in Fig. 4.13.

Since the discrete Fourier transform (DFT) performs a circular convolution, the flipped image can avoid the Gibbs artifact in the frequency domain due to the sudden change of the graylevel in the space domain. If one flips the image two more times, the resulting image has 512×512 pixels as shown in Fig. 4.14. There are enough pixels to perform a correlation coefficient.

Although the flipped image has a larger image size, the information content is the same as the original image. The flipping operation preserves all the original information in the original image. The MCC between the super-resolved 3×3 filtered image and the original image is 0.44040. The MCC between the super-resolved ICM filtered image and the original image is 0.46451. The MCC between the super-resolved proposed filtered image and the original image is 0.46533. These results clearly show that the proposed filter assists the ML algorithm to obtain the most from the image super-resolution algorithm. The proposed filter produced slightly better MCC results than the ICM filter did. The 3×3 filter had the worst performance.

4.6 Conclusions

In summary, one can state that the modified ICM filter (channel decoder) developed in this chapter performs efficient noise removal and also optimizes the performance of an iterative super-resolution algorithm (source decoder) by providing a good starting estimate to commence the iterations. The *Theorem 1* guarantees that the Gibbs-Markov equivalence exists in some cases and the Gibbs random field is bound between zero and one in other cases. Hence, it also guarantees that the proposed algorithm will converge in all cases. The numerical experiments demonstrate that the proposed algorithm produces desirable, enhanced results in comparison to the normal ICM approach as shown in Table 4.1 and Table 4.2. It is reasonable to conclude that the proposed algorithm is a useful approach to enhance the performance of a channel decoder in a dynamic imaging system based on information theory.

	3 × 3 Filter	ICM Filter	Proposed Filter
Noise Removal	MSE=134.42	MSE=125.65	MSE=88.44
ML Super-resolution	MSE=729.4	MSE=539.5	MSE=518.3
Fourier Transform	MCC=0.4404	MCC=0.4645	MSE=0.4653

Table 4.1: Performance record

Operation	Compare to 3 × 3 Filter	Compare to ICM Filter
Noise Removal	51.99% better in MSE	42.07% better in MSE
ML Super-resolution	40.73% better in MSE	4.09% better in MSE
Fourier Transform	5.36% better in MCC	0.17% better in MCC

Table 4.2: Proposed filter performance

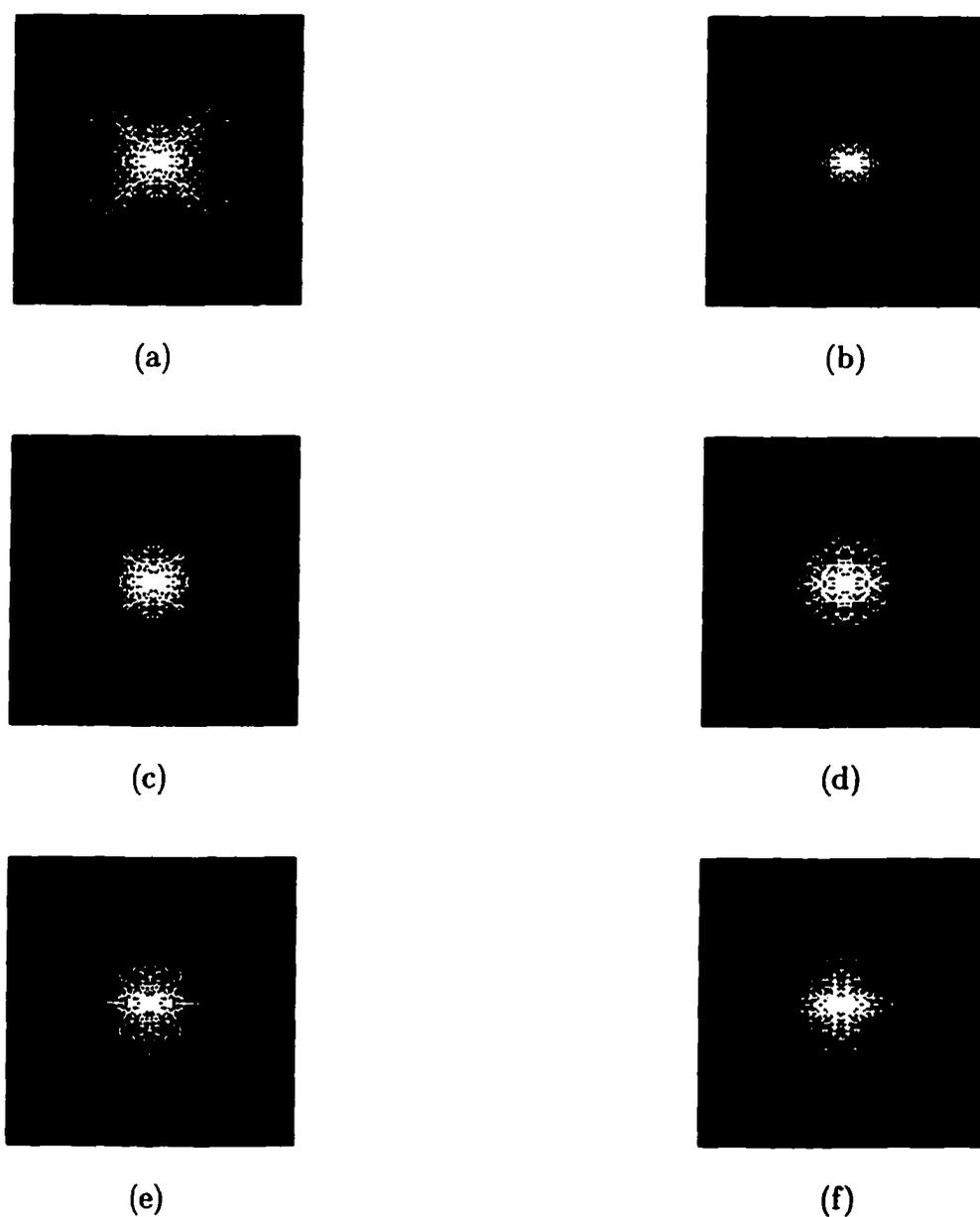


Figure 4.12: Power spectrum of images. (a) Original image. (b) Blurred image. (c) Super-resolved blurred image. (d) Super-Resolved 3×3 median filter output. (e) Super-resolved ICM filter output. (f) Super-resolved proposed filter output.



Figure 4.13: A portion of Lenna image flipped two times

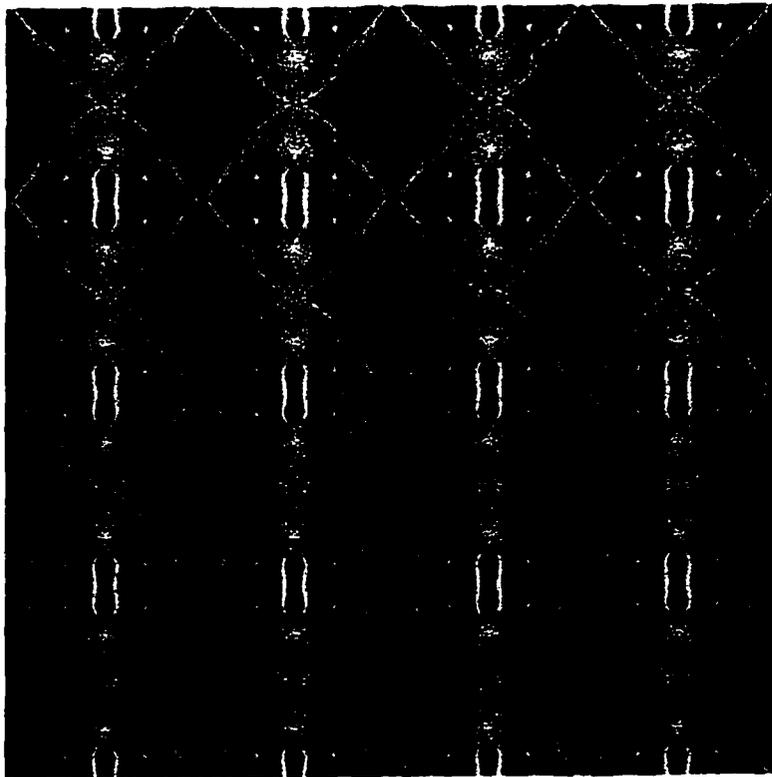


Figure 4.14: A portion of Lenna image flipped six times

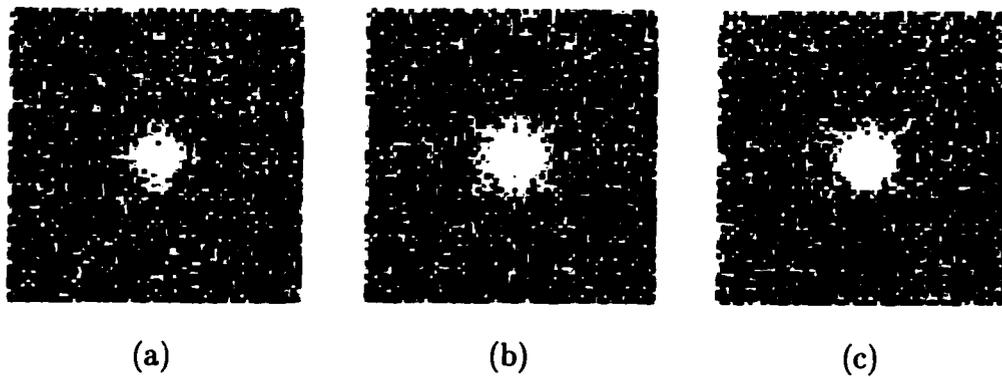


Figure 4.15: (a) The correlation between FFT of the original object and FFT of the super-resolved 3×3 median filtered image. (b) The correlation between FFT of the original object and FFT of the super-resolved ICM filtered image. (c) The correlation between FFT of the original object and FFT of the super-resolved proposed filtered image.

CHAPTER 5

OBJECT-BOUNDARY-CONSTRAINED MAXIMUM LIKELIHOOD ALGORITHM

5.1 Introduction

The source decoder (image restoration) is critical to image processing. Currently, most super-resolution algorithms can restore up to twice the sampling frequency in an operation environment. In a controlled environment, higher super-resolution performance has been achieved. In this chapter, the goal is to increase super-resolution performance within a limited processing time. We will examine the issue of object extension and how the maximum likelihood algorithm can be constrained with the object support. Both 1-D and 2-D validation experiments will be examined to demonstrate the effectiveness of an object-boundary-constrained maximum likelihood (OBC-ML) algorithm.

5.2 Importance of Object Extension

The maximum likelihood (ML) image super-resolution algorithm introduces information beyond the cutoff frequency of the acquired image. However, this algorithm does not show how much information is correctly estimated. An effective limit for the information reconstruction has to be imposed to prevent the algorithm from creating false information. Fortunately, an algorithm to analyze the limit of the effective

frequency reconstruction based on the object's boundary [47, 48] has been developed given an image of an object obtained via some remote sensing method, previously obtained information, or an object boundary detection algorithm. The object boundary is set to be the same as the object's exterior contour as it is projected onto a plane perpendicular to a specified look direction. The limit of effective information restoration [47, 48] in the frequency domain is

$$f_e \approx f_c + \frac{3}{4\pi L} \left[\text{sinc}^{-1}(K) - \text{sinc}^{-1} \left(\frac{KT}{\sigma_n} \right) \right] \quad (5.1)$$

where K is a numerical constant derived from the sinc function, T is a threshold of error by which the reconstructed spatial frequency cannot exceed the actual spatial frequency, σ_n is the standard derivation of image noise, and L is the width of the object contour. The notation sinc^{-1} means to be the inverse of the sinc function restricted to its main lobe.

5.3 Incorporate Object Support Constraint into the Maximum Likelihood Algorithm

If we know the object's contour, we know how much information an algorithm can estimate. However, this approach is most appropriate for an astronomical image. In a dark sky, an object has a finite contour and a zero graylevel outside the object contour. In everyday imaging, the possibility of a zero graylevel background rarely exists. Without a clear finite contour, the maximum likelihood (ML) algorithm can still perform image super-resolution extremely well. If we can create a finite contour for an acquired image, we know how much information the ML algorithm can restore.

Without a clear finite contour, the image restoration algorithm may not reach its highest performance level.

The method of limiting the band-width constraint in the frequency domain has been applied in the projections onto convex sets (POCS) approach [49, 46]. The POCS approach was described by Papoulis [50] in 1975. In his 1975 paper, he limited the bandwidth of the Fourier transform of the acquired signal and performed an inverse Fourier transform to convert the frequency information back to the signal. In addition, he replaced a portion of the known time domain signal back into the restored image. After performing n iterations, he demonstrated that the signal was almost restored.

In our new object-boundary-constrained maximum likelihood (OBC-ML) algorithm, the object contour in the space domain is constrained before the ML restoration is performed and everything outside the contour is set to zero. Consequently, only the object in the image will be processed by the ML algorithm. The $k+1$ iteration of the OBC-ML algorithm is given by the expressions:

$$\hat{f}^{k+1}(i, j) = \hat{f}_B^k(i, j) \left\{ \left[\frac{\hat{g}(i, j)}{h(i, j) * \hat{f}_B^k(i, j)} \right] \odot h(i, j) \right\} \quad (5.2)$$

and

$$\hat{f}_B^k(i, j) = \hat{f}^k(i, j) \times B_f(i, j) \quad (5.3)$$

where \hat{f}_B^k is the constrained object estimation, and B_f is the object boundary constraint that is a template. Pixels inside the boundary are set equal to one. Pixels outside the boundary are set equal to zero. Before every ML iteration, this algorithm

will remove everything beyond the object contour to make the object contour consistent throughout each ML processing stage. After restoration, the restored object replaces the blurred object in the original blurred image. It is useful to see the super-resolved object with the background at the end of the restoration. A one-dimensional example is shown in Fig. 5.1. The space domain signals are shown above the data flow. The frequency domain signals are shown below the data flow. The plots in this example clearly show that the OBC-ML algorithm can super-resolve data with astonishing accuracy and can estimate the lost information outside the passband almost perfectly.

The object contour constraint greatly increases the performance of the ML algorithm. The following validation experiments will support this claim. The flow chart of the object-boundary-constrained maximum likelihood (OBC-ML) method is given in Fig. 5.2 An interested reader can find the proof of the ML approach in the Appendix B.

5.4 1-D Validation Experiments

To test the performance of the OBC-ML method, a one dimensional signal was created and tested. The mean-square-error (MSE) between the original object and the restored image is the measurement of the accuracy of the restored image in this testing.

This set of data is referred to as the 1-D validation experiment case 1. In Fig. 5.3a, we created an object with three pulses on a compact support. The 1-D vector has 128 elements. The compact support has a value of 10 and is from pixel 55 to pixel

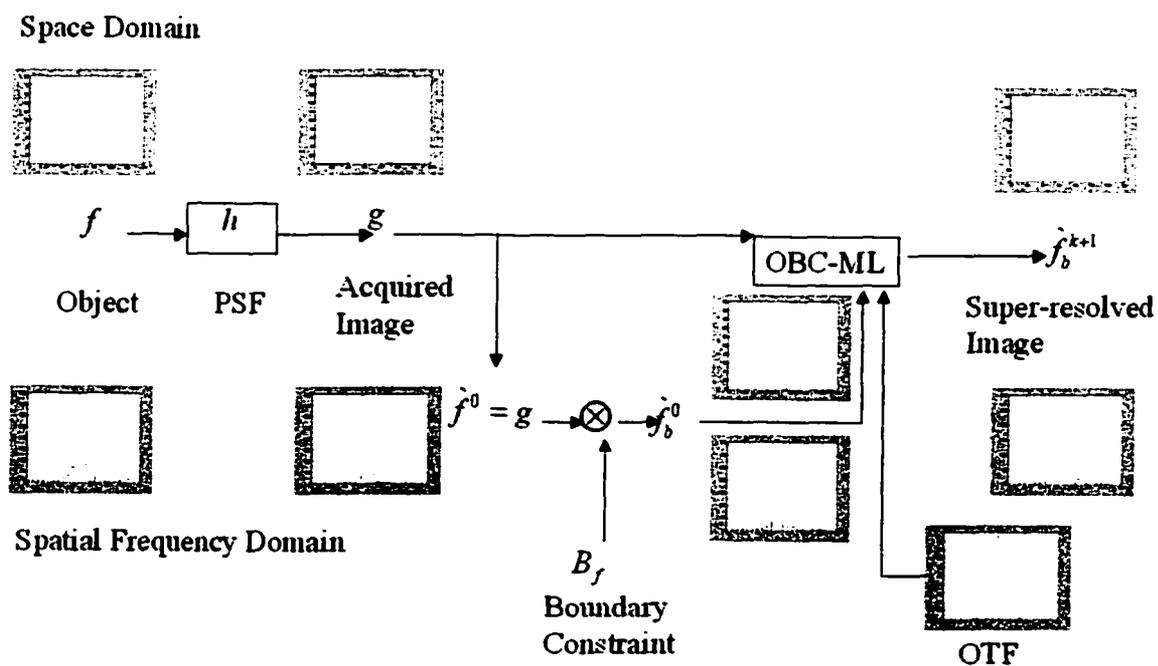


Figure 5.1: An one-dimensional example of the Object-Boundary-Constrained Maximum Likelihood Algorithm

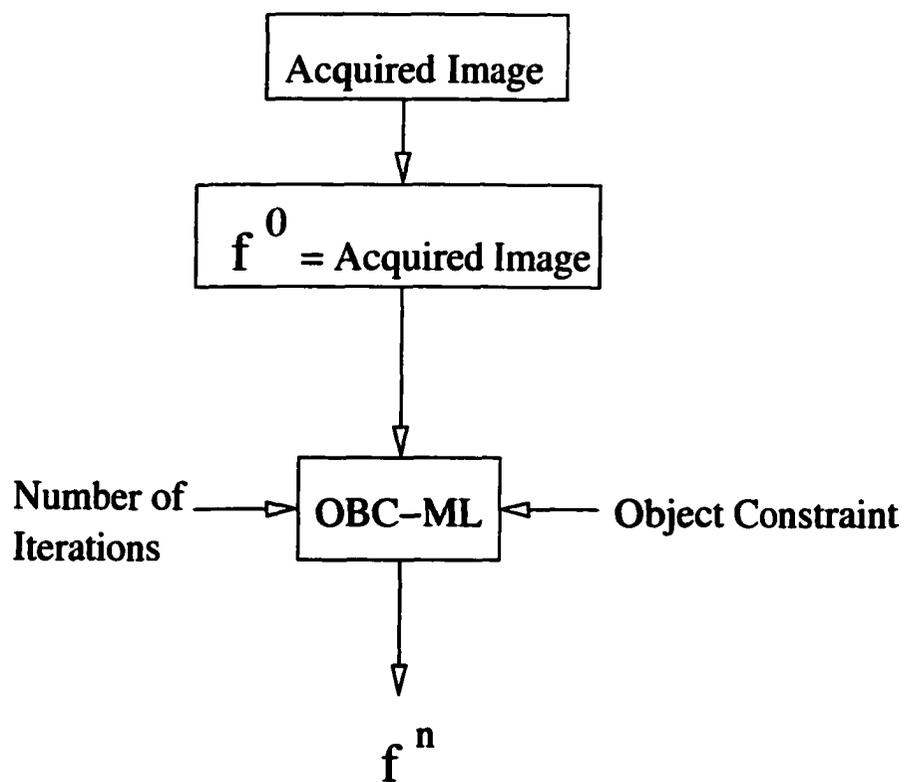


Figure 5.2: Object-Boundary-Constrained Maximum Likelihood Algorithm

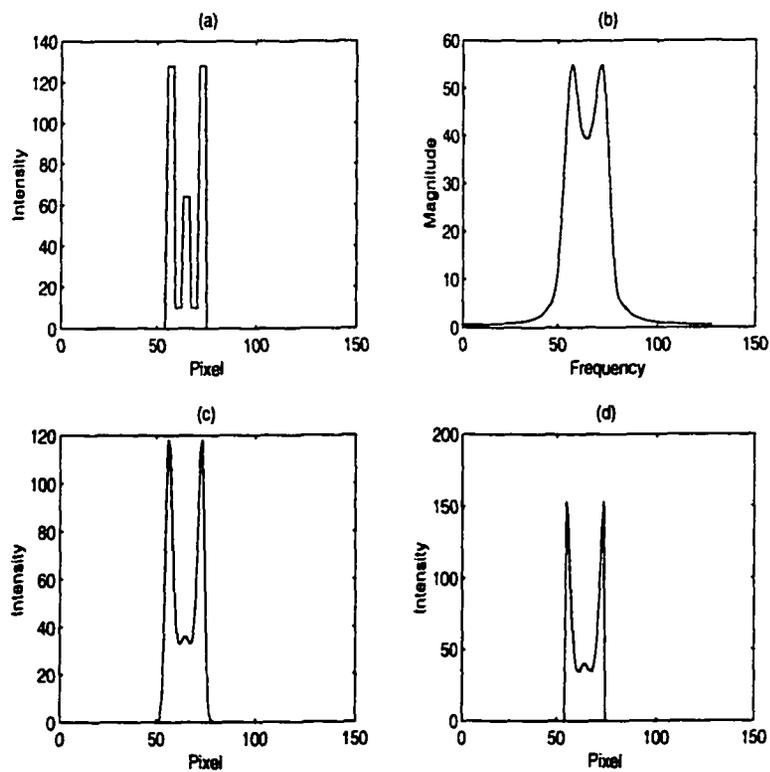


Figure 5.3: 1-D case 1.

74. The peak at the left has a value of 128 and is from pixel 55 to pixel 58. The peak in the middle has a value of 64 and is from pixel 63 and pixel 66. The peak at the right has a value of 128 and is from pixel 71 to pixel 74. The signal is blurred by the imaging system with cutoff frequency at $2\pi \times 20/128$ radians/second. The blurred signal is shown in Fig. 5.3b; it only shows two distinguishable pulses. The MSE of the blurred image is 518.4681. The iteration 100 result of the ML algorithm shown in Fig. 5.3c has an MSE value equal to 181.4611. It shows a little peak between two main peaks. The constraint of the OBC-ML algorithm is a vector consisting of 128 elements of zeros. It has values of one from pixel 55 to pixel 74. The iteration 100 result of the OBC-ML algorithm shown in Fig. 5.3d has an MSE value at 158.1855. It shows a little peak between two main peaks. The OBC-ML algorithm result has a lower MSE value than the usual ML algorithm result does.

To show the performance of the OBC-ML algorithm, an object with two pulses on a compact support was created as shown in Fig. 5.4a. This set of data is referred to as the 1-D validation experiment case 2. The 1-D vector has 128 elements. The compact support has a value of 20 and is from pixel 49 to pixel 80. The peak at the left has a value of 128 and is from pixel 49 to pixel 58. The peak at the right has a value of 128 and is from pixel 71 to pixel 80. The signal is blurred by an imaging system with cutoff frequency at $2\pi \times 6/128$ radians/second. The blurred signal is shown in Fig. 5.4b and the blurred image is a large pulse. The MSE of the blurred image is 1352.9. The 100th iteration of the ML algorithm result shown in Fig. 5.4c has an MSE value equal to 901.0270. Due to the lack of information in Fig. 5.4b, the ML algorithm can only estimate two small peaks on top of a main pulse in the

restored image. The constraint of the OBC-ML is a vector 128 elements of zeros. It has values of one from pixel 49 to pixel 80. With this boundary constraint, the OBC-ML can estimate two clear peaks in the restored image. The 100th iteration of the OBC-ML algorithm result shown in Fig. 5.4d has an MSE value equal to 340.6153. The OBC-ML algorithm result has a lower MSE value than the usual ML algorithm result.

The third 1-D validation experiment, case 3, considers the object shown in Fig. 5.5a. The signal is blurred by an imaging system with cutoff frequency at $2\pi \times 12/128$ radians/second. The blurred signal is shown in Fig. 5.5b and the two pulses are recognizable due to the higher cutoff frequency. The information in the blurred image is more than in the blurred image in Fig. 5.4b. The MSE of the blurred image is 610.7618. The 100-th iteration of the ML algorithm result shown in Fig. 5.5c has an MSE value equal to 246.6081. The result of the ML algorithm shows two peaks that are pointier. The constraint of the OBC-ML algorithm is a vector with 128 elements of zeros. It has values of one from pixel 49 to pixel 80. The 100th iteration of OBC-ML algorithm result shown in Fig. 5.5d has an MSE value equal to 89.8121. The restored image has a wider width and is closer to the original object. The OBC-ML algorithm result has a lower MSE value than the usual ML algorithm result.

In Fig. 5.6a, we created an object with a single pulse. This set of data is referred to as the 1-D validation experiment case 4. The 1-D vector has 128 elements. The pulse has a value of 128 and extends from pixel 33 to pixel 96. The signal is blurred by an imaging system with cutoff frequency at $2\pi \times 32/128$ radians/second. The blurred signal is shown in Fig. 5.6b and is a smooth column structure. The MSE of

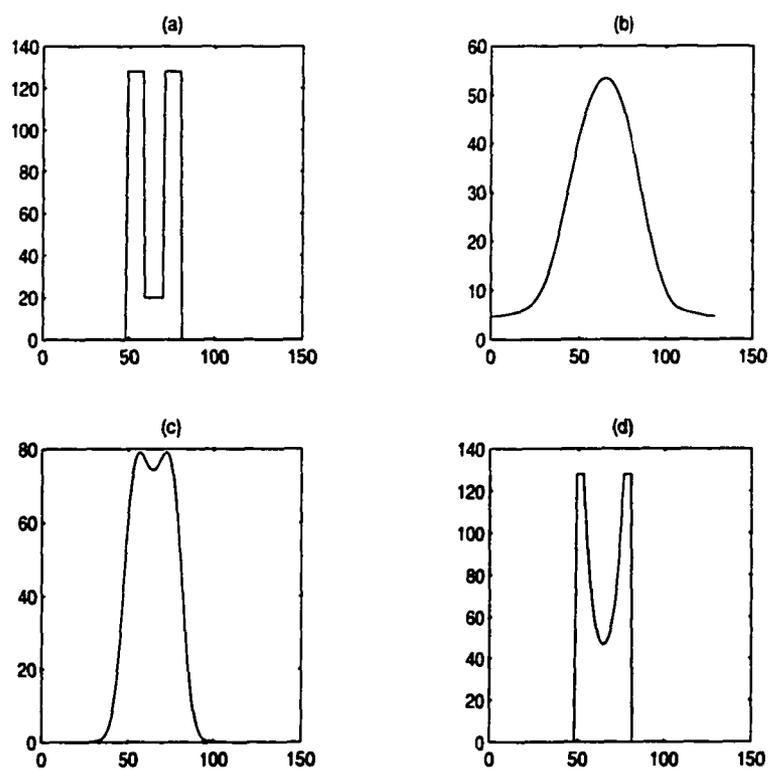


Figure 5.4: 1-D case 2.

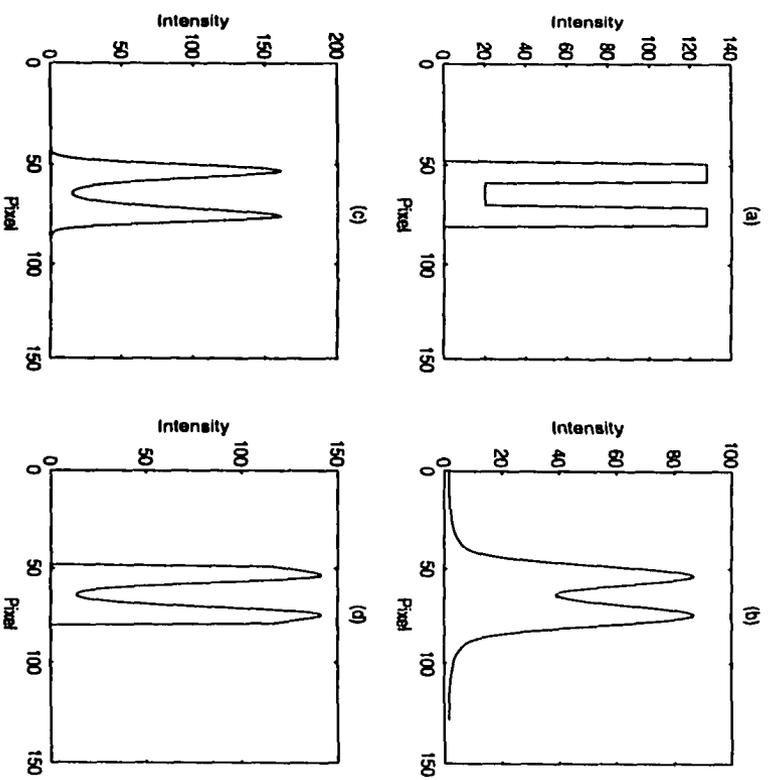


Figure 5.5: 1-D case 3.

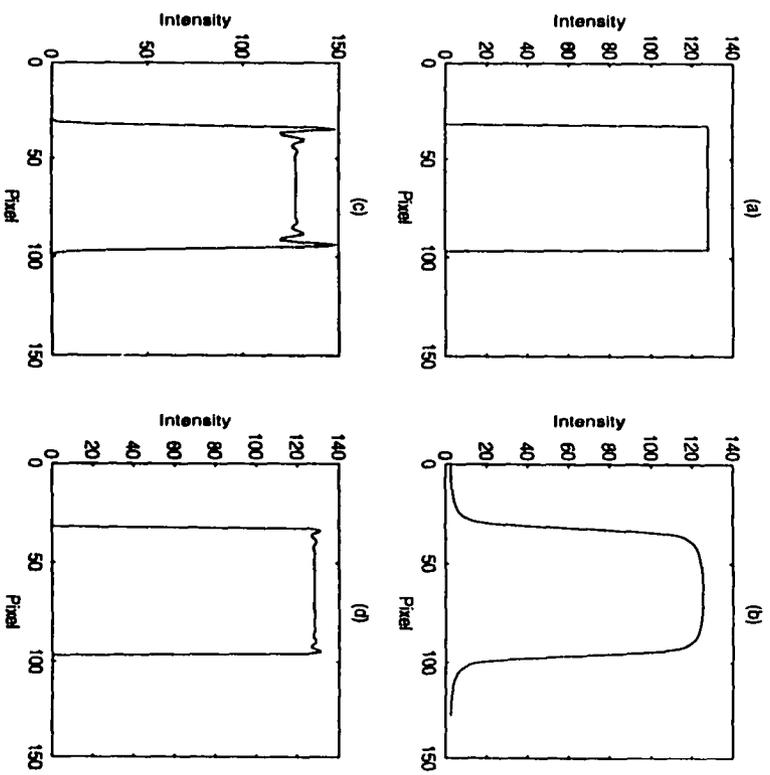


Figure 5.6: 1-D case 4.

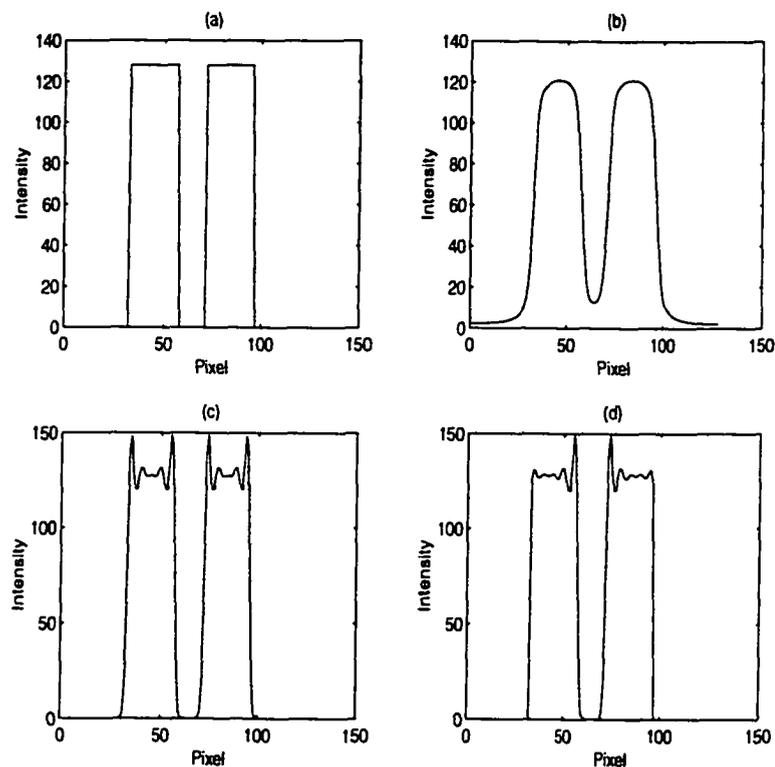


Figure 5.7: 1-D case 5

the blurred image is 173.5698. The iteration 100 result of the ML algorithm shown in Fig. 5.6c has an MSE value equal to 57.2103. The ML algorithm creates a large ringing artifact caused by Gibbs phenomenon as shown in Fig. 5.6c. The constraint of the OBC-ML algorithm is a vector consisting of 128 elements of zeros. It has values of one from pixel 33 to pixel 96. The iteration 100 result of the OBC-ML algorithm shown in Fig. 5.6d has an MSE value equal to 0.3566. The restored signal has a weak ringing artifact at the edge. The OBC-ML algorithm result has a lower MSE value than the usual ML algorithm result. Due to the amount of information in the blurred image, the OBC-ML algorithm result in Fig. 5.6d has a lower MSE value than the OBC-ML algorithm result in Fig. 5.5d.

In Fig. 5.7a, we created an object with two pulses. This set of data is referred to as the 1-D validation experiment case 5. The 1-D vector has 128 elements. The left pulse has a value of 128 and extends from pixel 33 to pixel 57. The right pulse has a value of 128 and extends from pixel 72 to pixel 96. The signal is blurred by an imaging system with cutoff frequency at $2\pi \times 32/128$ radians/second. The blurred signal is shown in Fig. 5.7b as two smooth pulses. The MSE of the blurred image is 346.5147. The iteration 100 result of the ML algorithm shown in Fig. 5.7c has an MSE value equal to 114.7538. The ML algorithm creates a large ringing artifact at the edges of the pulses caused by Gibbs phenomenon as shown in Fig. 5.7c. The constraint of the OBC-ML algorithm is a vector consisting of 128 elements of zeros. It has values of one from pixel 33 to pixel 96. The iteration 100 result of the OBC-ML algorithm shown in Fig. 5.7d has an MSE value equal to 58.6566. The restored signal has large ringing artifacts at the inner edges of the two pulses and decreases the ringing artifacts at the outer edges. The OBC-ML algorithm result has a lower MSE value than the usual ML algorithm result.

The set of data referred to as 1-D validation experiment, case 6, is the same object in Fig. 5.8. The imaging system has the same cutoff frequency as in case 5. The blurred signal is shown in Fig. 5.8b. The MSE of the blurred image is 346.5147. The iteration 100 result of the ML algorithm shown in Fig. 5.8c has an MSE value equal to 114.7538. The ML algorithm creates a large ringing artifact in the edges of the pulses caused by Gibbs phenomenon as shown in Fig. 5.8c. The constraint of the OBC-ML algorithm is a vector consisting of 128 elements of zeros. It has values of one from pixel 33 to pixel 57 and from pixel 72 to pixel 96. Basically, the boundary

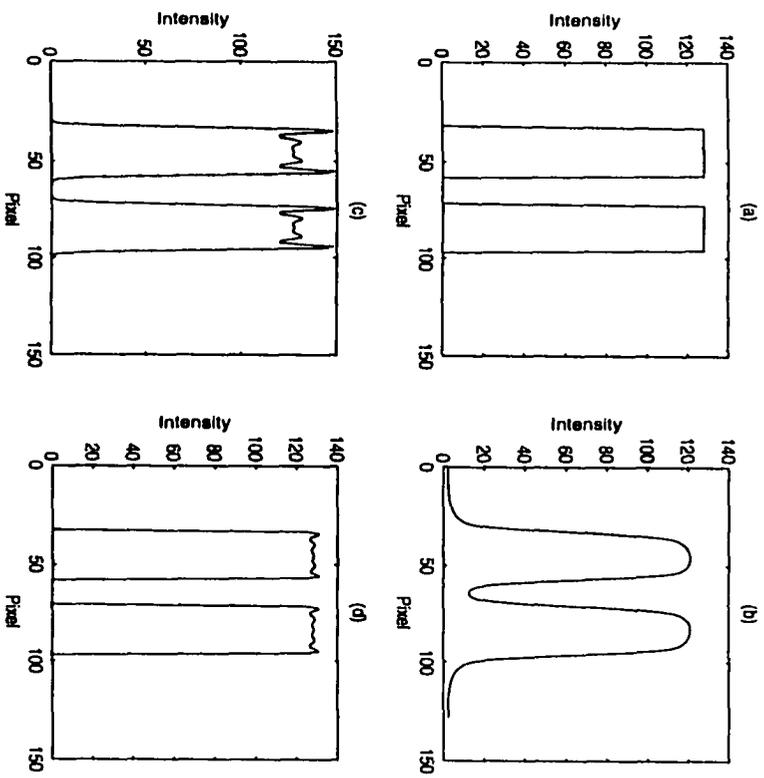


Figure 5.8: 1-D case 6.

constraint has two isolated boundaries. The iteration 100 result of the OBC-ML algorithm shown in Fig. 5.8d has an MSE value equal to 0.6914. The ringing artifact at the inner edges of the two pulses of the restored signal is decreased. The OBC-ML algorithm result has a lower MSE value than the usual ML algorithm result does.

The one dimensional validation experiments show that the OBC-ML approach generates more accurate super-resolved images than the standard ML algorithm. It also reduces the ringing artifact that is caused by the Gibbs phenomenon.

5.5 2-D Validation Experiments

To further test the OBC-ML algorithm, 2-D validation experiments were developed. To standardize the experiment, a common object was used in all of the experiments. The object is built on a 128×128 array filled with zeros. It consists of three rectangular structures. The rectangular structure on the left has a value of 128 from pixel 25 to pixel 40 along the x-axis and from pixel 21 to pixel 108 along the y-axis. The rectangular structure in the middle has a value of 128 from pixel 57 to pixel 72 along the x-axis and from pixel 21 to pixel 108 along the y-axis. The rectangular structure on the right has a value of 128 from pixel 89 to pixel 104 along the x-axis and from pixel 21 to pixel 108 along the y-axis.

To make the analysis more convincing, we introduce a residue quantity to measure the performance of the OBC-ML algorithm in the following experiments. The residue is defined as the following difference between the blurred real image and the restored

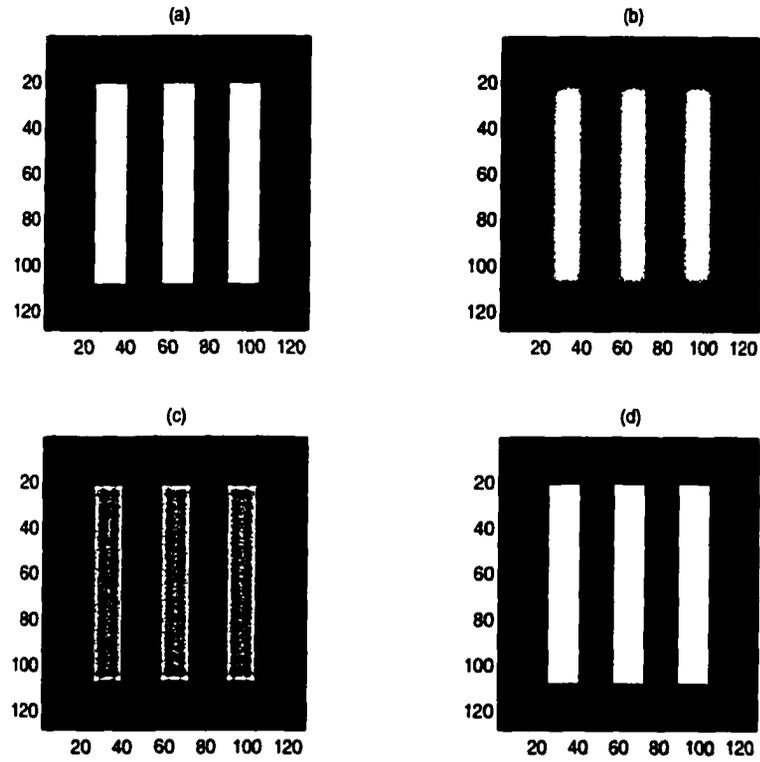


Figure 5.9: 2-D case 1.

blurred image:

$$Residue = \frac{1}{M \times N} \sum_{x=1}^M \sum_{y=1}^N [g(x, y) - \hat{x}(x, y) * h(x, y)] \quad (5.4)$$

where g is the acquired image, \hat{x} is the corresponding restored image, and h is the point spread function for the imaging system. The residue decreases when the accuracy of the restored image increases.

The test object, three rectangular structures, is shown in Fig. 5.9a. The structures are blurred by an imaging system with cutoff frequency at $2\pi \times 32/128$ radians/second. The blurred image is shown in Fig. 5.9b and has an MSE value of 402.5682. This set of data is referred to as the 2-D validation experiment case 1. The resolution

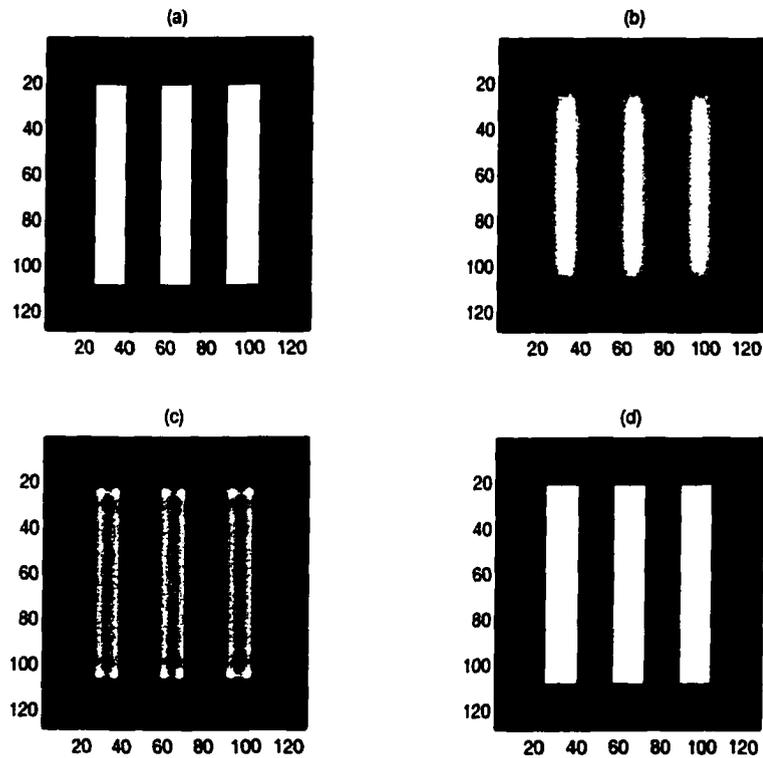


Figure 5.10: 2-D case 2.

is high enough to show all three structures. The iteration 100 ML restored image has an MSE value of 135.4259 and a residue value of 0.1386. The ringing artifacts created by Gibbs phenomenon near the edges are showed in Fig. 5.9c. The object boundary constraint of the OBC-ML algorithm is exactly the same as the object's boundary. The iteration 100 OBC-ML algorithm restored image has an MSE value of 0.7571 and a residue value of 0.0033. The restored image is shown in Fig. 5.9d and is a more accurate estimate of the original object. The object boundary constraint (OBC) significantly reduces the Gibbs phenomenon.

The test object is next blurred by an imaging system with cutoff frequency at $2\pi \times 16/128$ radians/second. This set of data is referred to as the 2-D validation

experiment case 2. The blurred image is shown in Fig. 5.10b and has a MSE value equal to 806.7981. The resolution is high enough to show all three structures. The iteration 100 ML restored image has an MSE value equal to 263.2145 and a residue value of 0.1960. The ringing artifacts created by Gibbs phenomenon near the edges are shown in Fig. 5.10c. The amplitude of the ringing reduces but the width increases. This is due to the decrease of the cutoff frequency. For the ML algorithm, the ringing artifact appears at the sharp edge. If there is not high enough frequency information present in the blurred image, no hard edge will be defined in the restored image. Sharp edges are created by high frequency information. Gibbs phenomenon happens when the Fourier transform recreates the sharp edge in the space domain. The sharp edge is a summation of a large number of terms in the Fourier series. The sharper the edge is, the more Fourier components are needed and the stronger the Gibbs phenomenon will be. The object boundary constraint of the OBC-ML algorithm is exactly the same as the object's boundary. The iteration 100 OBC-ML algorithm restored image has an MSE value of 2.3605 and a residue value of 0.0037. The restored image is shown in Fig. 5.10d and is not as accurate as the 2-D case 1 due to the lower cutoff frequency in the imaging system.

The test object is blurred by an imaging system with cutoff frequency at $2\pi \times 8/128$ radians/second for the 2-D validation experiment, case 3. The blurred image is shown in Fig. 5.11b and has an MSE value equal to 1543.1. The resolution is low and barely show all three structures. The iteration 100 ML restored image has an MSE value equal to 759.0785 and a residue value equal to 0.0802. The ML restored image is blurred as shown in Fig. 5.11c and is narrower than the original object. The object

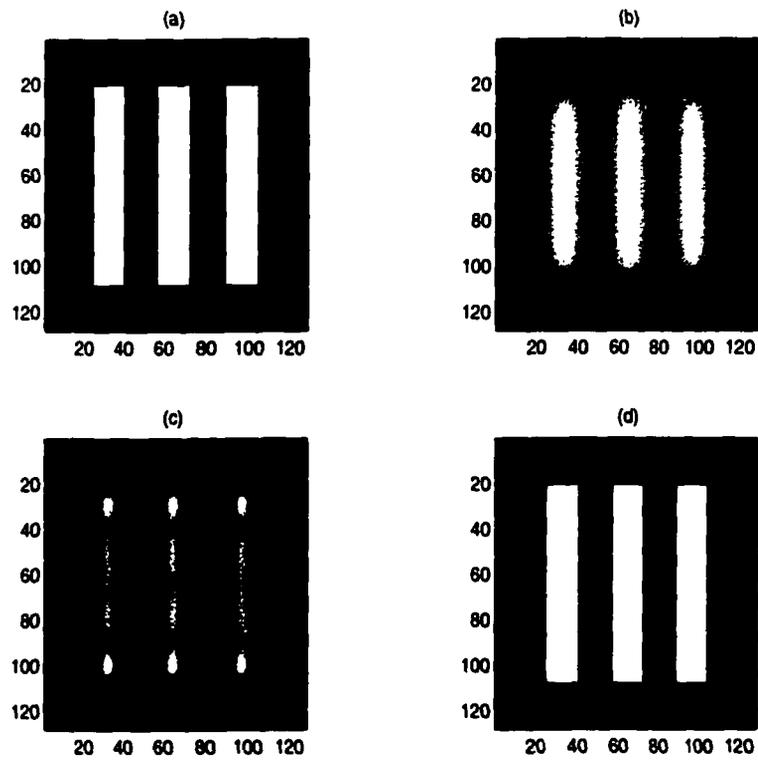


Figure 5.11: 2-D case 3.

boundary constraint of the OBC-ML algorithm is exactly the same as the object's boundary. The iteration 100 OBC-ML algorithm restored image has an MSE value equal to 3.0188 and a residue value equal to 0.0126. The restored image is shown in Fig. 5.11d and is not as accurate as the 2-D case 2 due to the lower cutoff frequency in the imaging system.

In the following experiment, the 2-D case 4, we studied the effect of the inaccuracy of the object contour constraint in the OBC-ML process. The size of the object boundary constraint was increased and made less accurate gradually by increasing the width on both sides of each rectangular structure by 2 pixels at a time.

The object under investigation is the same as the one in 2-D cases 1, 2, and 3. It is blurred by a point spread function with cutoff frequency at $2\pi \times 8/128$ radians/second. As the width of the constraint increases in Figs. 5.12a, 5.12c, 5.13a, and 5.13c, the MSE and residue increase as shown, respectively in Figs. 5.12b, 5.12d, 5.13b, and 5.13d. However, the MSEs and residues in these OBC-ML generated images are still lower than the MSE and residue values obtained from the corresponding ML algorithm restored images. The results of the above experiments show that the OBC-ML algorithm outperforms the standard ML algorithm well even with an incorrect object contour constraint.

In the 2-D case 4a, the OBC is shown in Figs. 5.12a. The iteration 100 OBC-ML algorithm restored image has an MSE value equal to 356.6329 and a residue value equal 0.014. The restored image is shown in Fig. 5.12b. The OBC keeps increasing as shown in Fig. 5.12c. The iteration 100 OBC-ML algorithm restored image has an

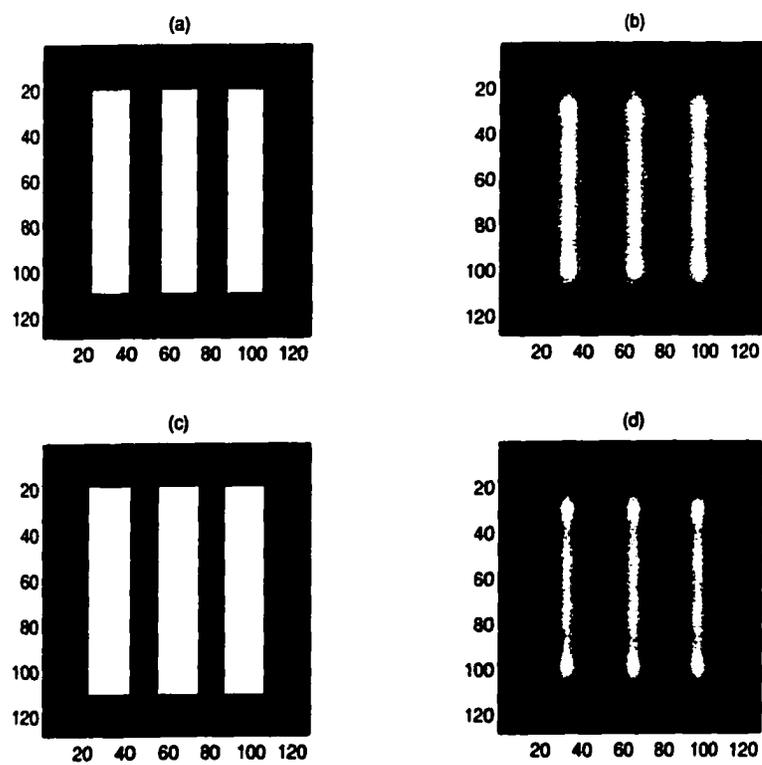


Figure 5.12: 2-D case 4a.

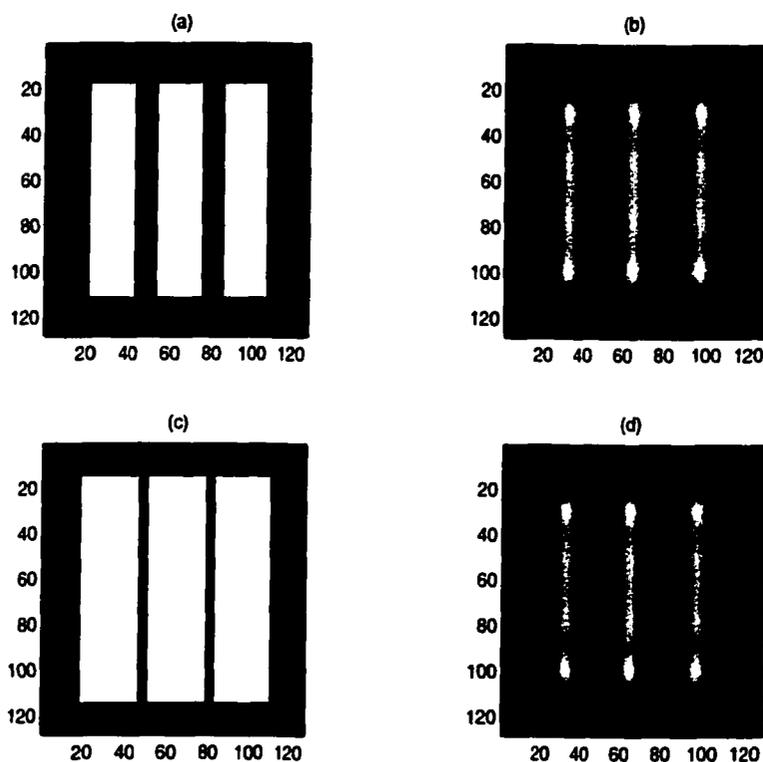


Figure 5.13: 2-D case 4b.

MSE value equal to 503.2826 and a residue value equal 0.0434. The restored image is shown in Fig. 5.12d.

In the 2-D case 4b, the OBC is shown in Figs. 5.13a. The iteration 100 OBC-ML algorithm restored image has an MSE value equal to 589.1381 and a residue value equal to 0.0601. The restored image is shown in Fig. 5.13b. The OBC keeps increasing as shown in Fig. 5.13c. The iteration 100 OBC-ML algorithm restored image has an MSE value equal to 721.4939 and a residue value equal to 0.0774. The restored image is shown in Fig. 5.13d.

In the following experiment, the limit of the ML is challenged to see what difference the object boundary constraint makes in the ML super-resolution results. This set of

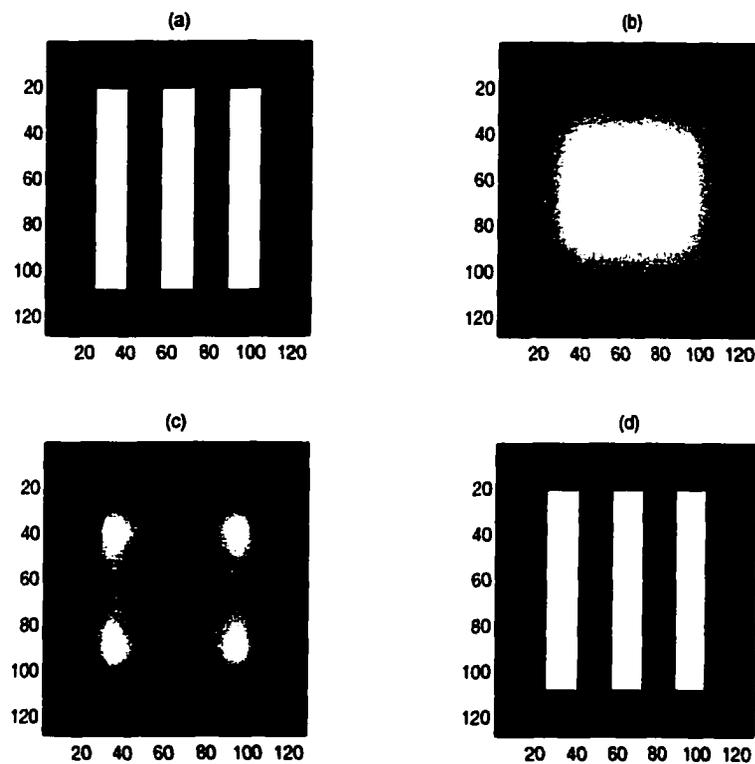


Figure 5.14: 2-D case 5

data is referred to as the 2-D case 5. The three rectangular structure test object was imaged with an imaging system with a cutoff frequency of $2\pi \times 4/128$ radians/second. The blurred image, shown in Fig. 5.14b, is unrecognizable. The blurred image has an MSE equal to 2460.4. The ML restored image Fig. 5.14c is not showing any useful aspects of the original object. The iteration 100 ML restored image has an MSE value equal to 1824.6 and a residue value equal to 1.0539. The boundary constraint used in the OBC-ML algorithm is the exact boundary of the object. The iteration 100 OBC-ML algorithm restored image Fig. 5.14d has an MSE value equal to 1.0821 and a residue value equal to 0.0026. This is an extremely accurate restoration considering the extremely blurred image to which the OBC-ML algorithm was applied.

In this extremely low resolution example, no one can be sure what the object contour should be in Fig. 5.14b. It is interesting to study the effect of the rotation of the OBC. To see what the effect of an incorrect object boundary constraint on the restored image, the OBC was rotated 90 degrees. This new set of test data is referred to as the 2-D case 6. In Fig. 5.15d, the restored image is estimated with a 90 degree rotated object contour constraint. The iteration 100 OBC-ML restored image has an MSE value equal to 3679.5 and a residue value equal to 0.8550. This means that the incorrect constraint generates more MSE error than the blurred image. This shows the importance of a correct OBC. Even without the MSE and residue measurements, an observer can easily recognize that the restored object in Fig. 5.15d is not correct. Visually, no one can tell there is hidden information in Fig. 5.15b. However, with the help of a correct object contour constraint, the hidden information can be used to recreate a correct estimation as shown in Fig. 5.14d. With an object contour constraint that is totally wrong, the hidden information becomes useless in the restoration as shown in Fig. 5.15d.

In the next two experiments, we studied the effect of the inaccuracy of the object contour constraint in the OBC-ML process with an unrecognizable acquired image. This set of data will be referred to as the 2-D case 7a and 7b. The object is the same three rectangular structure test object. The size of the object boundary constraint was increased to become less accurate gradually by increasing the slit width by 2 pixels at a time. The object was blurred by a point spread function with a cutoff frequency at $2\pi \times 4/128$ radians/second. The blurred image was unrecognizable as is shown in Fig. 5.14b. As the width of the constraint increases in Figs. 5.16a, 5.16c,

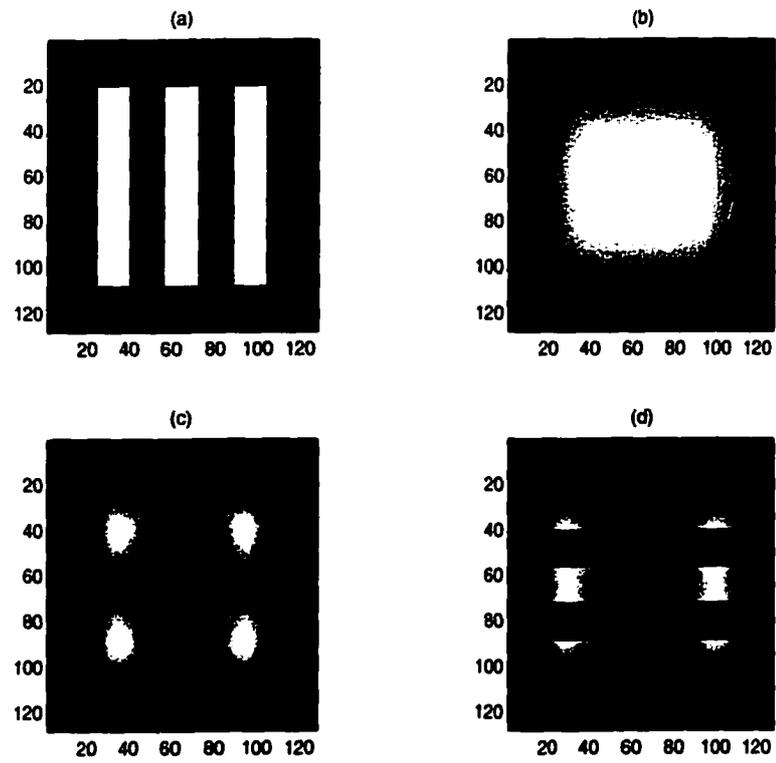


Figure 5.15: 2-D case 6.

5.17a, and 5.17c, the MSE and residue increase as shown in in Figs. 5.16b, 5.16d, 5.17b, and 5.17d, respectively. In Fig. 5.16b, the restored image has an MSE value equal to 11.5778 and a residue value equal to 0.0134. In Fig. 5.16d, the restored image has an MSE value equal to 880.0374 and a residue value equal to 0.0519. In Fig. 5.17b, the restored image has an MSE value equal to 1153.6 and a residue value equal to 0.1171. In Fig. 5.17d, the restored image has an MSE value equal to 1642.2 and a residue value equal to 0.4359. However, the MSEs and the residues associated with these OBC-ML restored images are still lower than the MSEs and the residues exhibited by the ML restored image. The results of these experiments show that the OBC-ML algorithm can perform well with an incorrect object boundary constraint and that the prior knowledge of an image, such as the object's boundary, can be used to restore the low resolution image. Nonetheless, the experiments also demonstrated that the orientation of the constraint cannot be extremely far from accurate for the OBC-ML algorithm to be effective.

To confirm this observation, a blurred image was created with a low pass filter that has a cutoff frequency at $2\pi \times 2/128$ radians/second. This blurred image is shown in Fig. 5.18b. The previous OBC-ML algorithm experiments were repeated with this new blurred image. The resolution is even lower than in the previous case, and the resulting blurred image is not recognizable.

The 2-D case 8 proved the effectiveness of the OBC-ML just as the 2-D case 5 did. The original object is shown in Fig. 5.18a. The blurred image is shown in Fig. 5.18b. It has a MSE value equal to 2882.0. The ML restored image Fig. 5.18c does not show any useful structure of the original object; the appearance of the restored image is

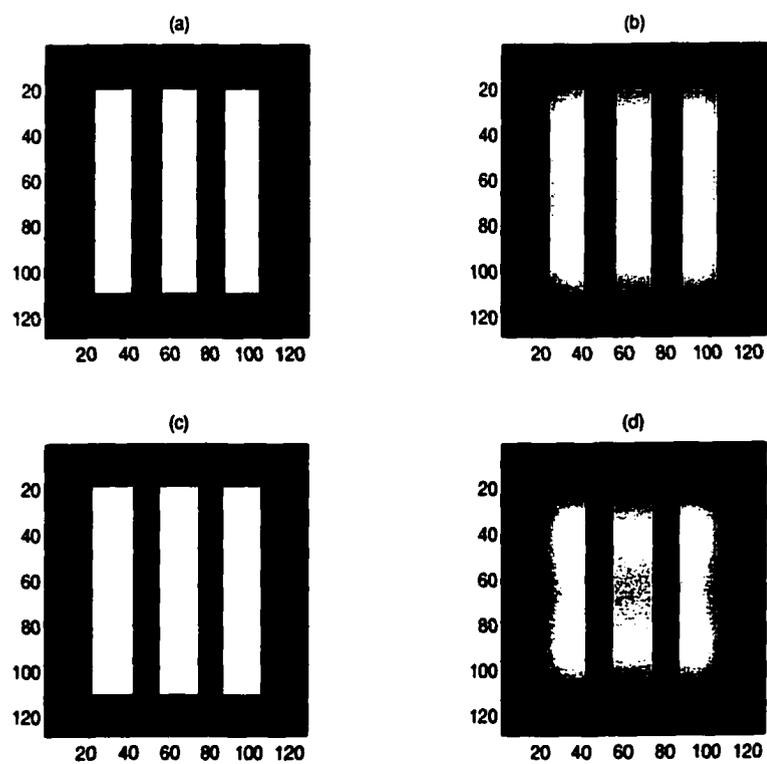


Figure 5.16: 2-D case 7a.

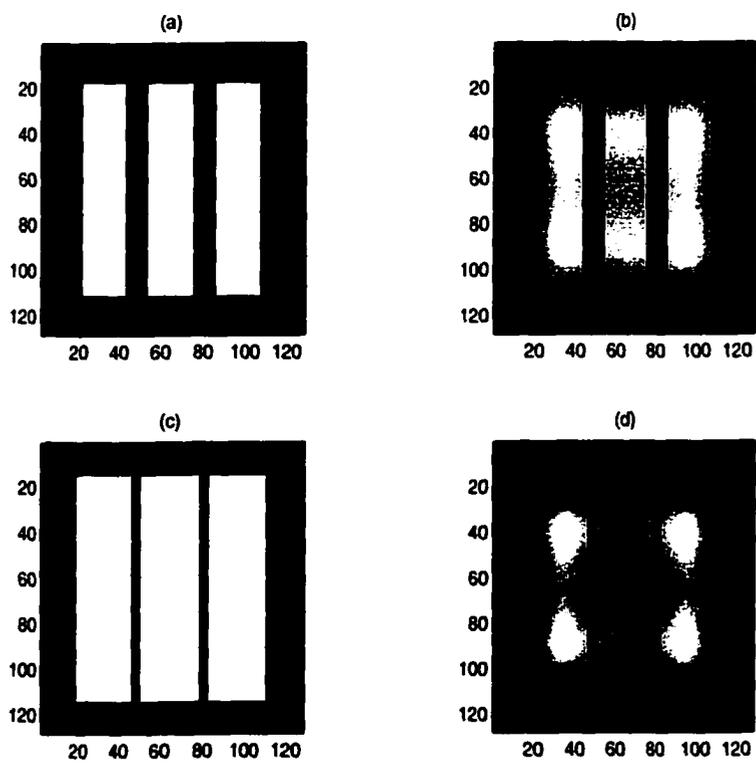


Figure 5.17: 2-D case 7b.

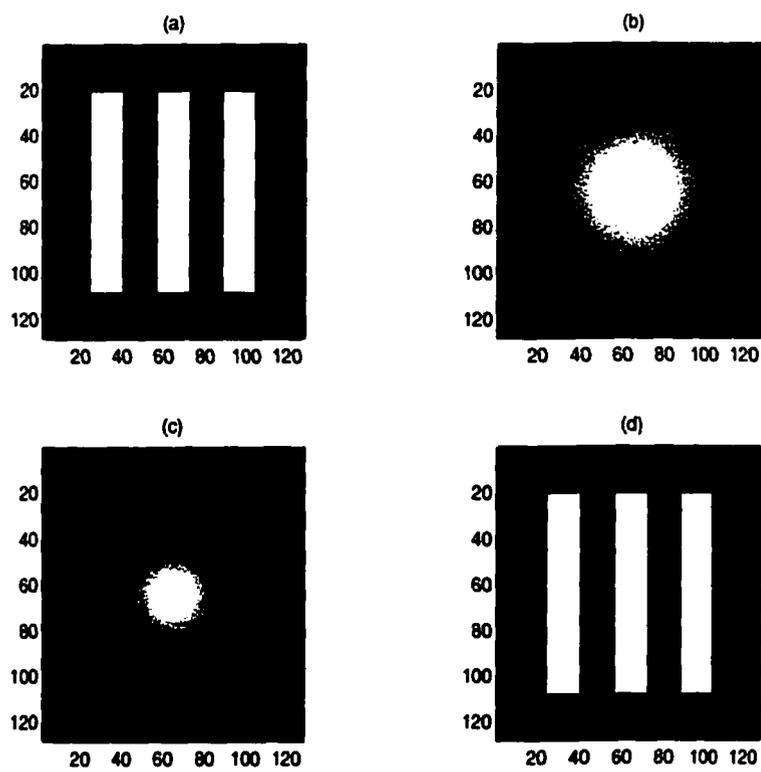


Figure 5.18: 2-D case 8.

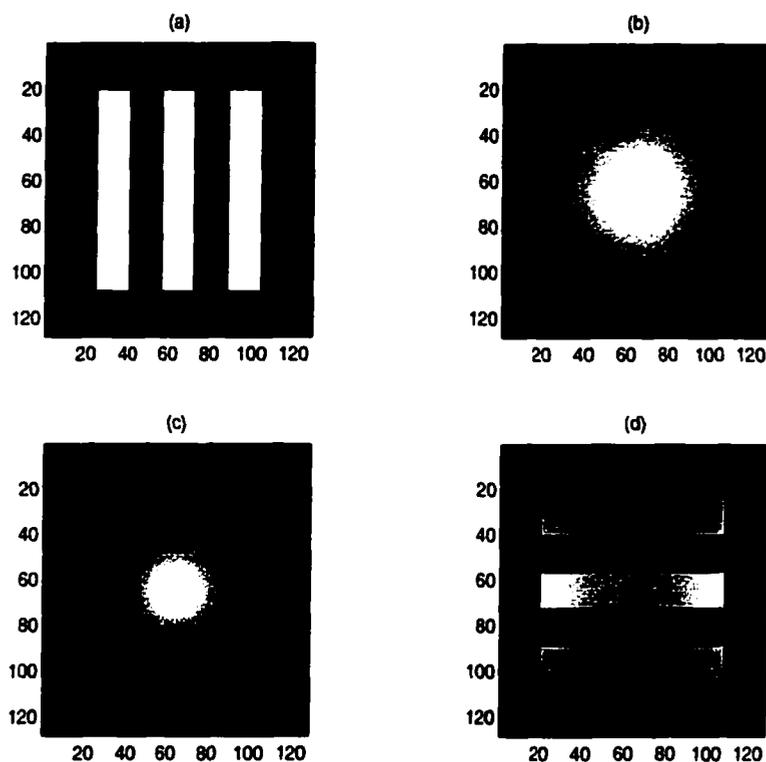


Figure 5.19: 2-D case 9.

similar to a cone. The iteration 100 ML restored image has an MSE value equal to 2692.8 and a residue value equal to 0.00064093. The boundary constraint used in the OBC-ML algorithm was the exact boundary of the object, and the iteration 100 OBC-ML algorithm restored image shown in Fig. 5.18d has an MSE value equal to 0.1743 and a residue value equal to 0.00088727. This is extremely accurate considering the extremely blurred image.

The 2-D case 9 was used to show the effect of a rotated object boundary constraint in the OBC-ML algorithm as the 2-D case 6 did. The original object is shown in Fig. 5.19a. The blurred image is shown in Fig. 5.19b. It has an MSE value equal to 2882.0. The ML restored image Fig. 5.19c is not showing any useful structure of

the original object. The appearance of the restored image is similar to a cone. The iteration 100 ML restored image has an MSE value equal to 2692.8 and a residue value equal to 0.00064093. The boundary constraint used in the OBC-ML algorithm was the exact boundary of the object; the iteration 100 OBC-ML algorithm restored image Fig. 5.19d has an MSE value equal to 3892.9 and a residue value equal to 0.0021. With an incorrect object contour constraint that is totally out of place, the hidden information in the blurred image becomes useless in the restoration process as shown in Fig. 5.19d. This demonstrates the importance of a correctly oriented contour constraint.

In the following experiment, the effect of the inaccuracy of the object contour constraint in the OBC-ML process was studied. The size of the object boundary constraint was increased and made less accurate gradually by increasing the width on both sides by 2 pixels at a time for the experiment.

The object under investigation is the same as the one used in the 2-D case 7a, 7b. It was blurred by a point spread function with cutoff frequency at $2\pi \times 2/128$ radians/second. As the width of the constraint increases in Figs. 5.20a, 5.20c, 5.21a, and 5.21c, the MSE and residue increase as shown, respectively in Figs. 5.20b, 5.20d, 5.21b, and 5.21d. However, the MSEs and residues in these OBC-ML generated images are still lower than the MSE and residue values obtained from the corresponding ML algorithm restored images. The results of the above experiments show that the OBC-ML algorithm can perform well and outperforms the standard ML algorithm even with an incorrect object contour constraint.

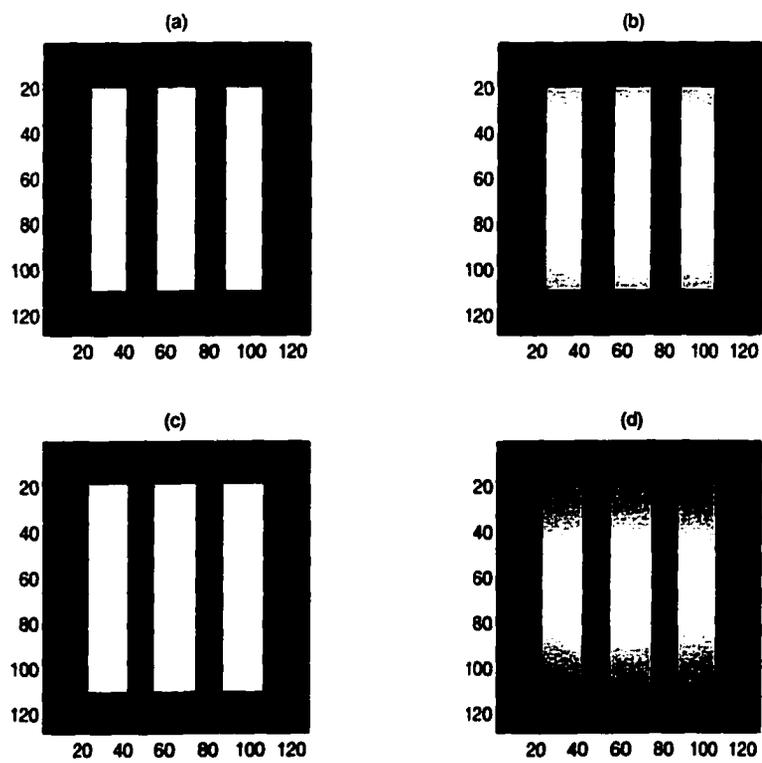


Figure 5.20: 2-D case 10a.

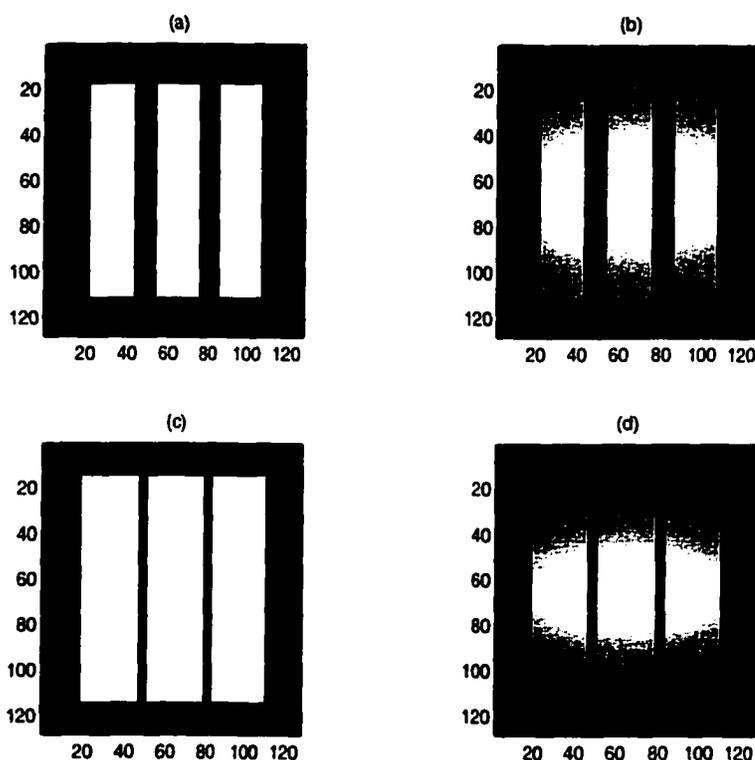


Figure 5.21: 2-D case 10b.

In the 2-D case 10a, the OBC is shown in Figs. 5.20a. The iteration 100 OBC-ML algorithm restored image has an MSE value equal to 544.6345 and a residue value equal to 0.0004336. The restored image is shown in Fig. 5.20b. The OBC keeps increasing as shown in Fig. 5.20c. The iteration 100 OBC-ML algorithm restored image has an MSE value equal to 968.2964 and a residue value equal to 0.00024605. The restored image is shown in Fig. 5.20d.

In 2-D case 10b, the OBC is shown in Figs. 5.21a. The iteration 100 OBC-ML algorithm restored image has an MSE value equal to 1306.7 and a residue value equal to 0.00019235. The restored image is shown in Fig. 5.21b. The OBC keeps increasing as shown in Fig. 5.21c. The iteration 100 OBC-ML algorithm restored image has an

MSE value equal to 1306.7 and a residue value equal to 0.00019235. The restored image is shown in Fig. 5.21d. The results of the 2-D cases 10a and 10b confirm our observation in the 2-D cases 7a and 7b that the OBC-ML algorithm can perform well even with an incorrect object boundary constraint. Moreover, they confirm that the prior knowledge of an image, such as the object's boundary, can be used to restore an image with extremely low resolution. Nonetheless, the experiments also demonstrate that the orientation of the constraint cannot be extremely inaccurate for the OBC-ML algorithm to be effective.

5.6 Conclusions

A series of 1-D and 2-D validation experiments has been presented that demonstrated the robustness of the OBC-ML algorithm. This algorithm can be used in realistic imaging situation. A contour detection algorithm can be applied to detect the contour of an object in any image, and the detected contour can be used as the object contour constraint for the image reconstruction process. The OBC-ML super-resolved image has been shown in general to have a higher resolution than the corresponding ML super-resolved image as shown in Table 5.1. This research was a precursor to the research that was done by Bhattacharjee [33]. His dissertation efforts extended this research and showed that the OBC-ML algorithm could be used in a real operation environment. It is also possible to apply the OBC-ML algorithm in remote sensing situations. For a hyperspectrum image, certain bands of wavelengths will have problems due to absorption by water molecules, carbon dioxide, and etc. The window of absorption by molecule is different from molecule to molecule. The

image contour obtained by other wavelength bands can be used to create contour information for applying the OBC-ML algorithm to the images collected in the wavelength that is strongly absorbed by the molecules. The OBC-ML super-resolved images can then be used in clustering to produce superior remote sensing images.

Acquired Image	Performance with Accurate Boundary Constraint
Blurred Image	25,045% better than ML in MSE
Extreme Blurred Image	168,516% better than ML in MSE

Table 5.1: OBC-ML performance

CHAPTER 6

SUPER-RESOLUTION ALGORITHM BASED ON A SELF-ORGANIZING MAP

6.1 Introduction

We explored the possibility of applying a self-organizing map (SOM) to perform image super-resolution. Currently, every super-resolution algorithm is constructed as an iterative algorithm. The most common ones are *Maximum a Posteriori*, *Maximum Likelihood*, *Maximum Entropy* [51], *Expectation Maximization* [52], and *Pixon* [53]. These algorithms are ideal for use in astronomy because it is not necessary to process images in real time. However, it would be difficult to justify the applications of the above algorithms in real time applications.

The artificial neural network is inspired by the human brain [54, 55, 56, 57]. The brain computes in an entirely different way than the digital computer. The advantage of a neural network over traditional computing is that the fundamental unit of the neural network, the neuron, is a highly parallel processing device. The brain operates at 30Hz. However, the brain overcomes the extremely slow operation frequency by taking advantage of massive interconnections between neurons. The *energetic efficiency* of the brain is also superior to the VLSI circuit [58]. Moreover, the brain is optimized by millions of years of evolutionary processes. Eventually, consciousness and free will have arisen from the human brain [59]. The brain is a

highly complex, nonlinear, and parallel information processing system. The following quotation gives an excellent definition of a neural network [60]:

A neural network is a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:

1. *Knowledge is acquired by the network through a learning process.*
2. *Interneuron connection strengths known as synaptic weights are used to store the knowledge.*

The use of neural networks offers the following useful properties and capabilities: nonlinearity, input-output mapping, adaptivity, evidential response, contextual information, fault tolerance, uniformity of analysis and design, and neurobiological analogy. For image processing, neural networks can perform one-pass operations and generate a closed approximation of the solution. Neural networks are universal approximators; they cannot generate perfect results. However, they generate close approximations in short periods of time. This feature is ideal for real time operations. Due to the massively parallel nature of neural networks, developing image super-resolution with local operations can take advantage of the parallel processing afforded by neural networks.

6.2 Neural Networks Basics

Neural networks can be classified into two major categories: supervised learning and unsupervised learning. The supervised learning paradigm requires the presence

of an external teacher. By having a prior knowledge of the environment, the teacher can extract a training vector from the environment and provide the neural network with a desired or target response for that training vector. The neural network response is adjusted by fine tuning the synaptic weight (parameters) of the neurons to generate an output close to the desired response. The most popular supervised learning method is back-propagation. It was developed by Paul J. Werbos [61] in his Applied Mathematics Ph.D. dissertation [62] in 1974. Professor Shun-ichi Amari [63] of the University of Tokyo developed a version of it in 1967. However, Professor Amari's work is less known. Back-propagation has been used in machine learning, adaptive control, communications, and signal processing.

On the other hand, the unsupervised learning paradigm [64] does not require a teacher. The unsupervised learning approach is also known as self-organized learning. The purpose of the unsupervised learning mode is to discover significant patterns or features in the input data without a teacher. The algorithm is provided with a set of rules of a local nature that enables it to perform an input-output mapping. The self-organizing map (SOM) is an unsupervised neural network. The SOM was developed by Kohonen [65] and has been one of the most successful clustering algorithms in the areas of information retrieval, data mining, digital library, internet search, multimedia data mining, semantic search, multilingual systems, medical informatics, information analysis and visualization [66]. It is a very powerful algorithm to find hidden information.

It would be appropriate to introduce the basics of clustering algorithms [67]. Clustering of data is a method by which large sets of data are grouped into clusters of

smaller sets of similar data. A clustering algorithm attempts to find natural groups of components (or data) based on some similarity. The clustering algorithm also finds the centroid of a group of data sets. To determine cluster membership, most algorithms evaluate the distance between a point and the cluster centroids. The output from a clustering algorithm is basically a statistical description of the cluster centroids with the number of components in each cluster. The centroid of a cluster is a point whose parameter values are the mean of the parameter values of all the points in the clusters. Generally, the distance between two points is taken as a common metric to assess the similarity among the components of a population. The most commonly used distance measure of clustering algorithms is the Euclidean metric. The clustering algorithms operate on the raw data set. The various clustering concepts available can be grouped into two broad categories: hierarchical algorithms (pairwise clustering algorithms) and non-hierarchical algorithms (central clustering algorithms). Hierarchical algorithms include those techniques where the input data is not partitioned into the desired number of classes in a single step. Instead, a series of successive fusions of data are performed until the final number of clusters is obtained. The most popular hierarchical clustering algorithm is the minimal spanning tree method [67].

Non-hierarchical algorithms include those techniques in which a desired number of clusters is assumed at the start. Points are allocated among clusters so that a particular clustering criterion is optimized. A possible criterion is the minimization of the variability within clusters, as measured by the sum of the variance of each

parameter that characterizes a point. The most common non-hierarchical algorithm is the K-means clustering algorithm [68].

In general, the SOM algorithm is a non-hierarchical clustering algorithm. The weight vectors of the SOM are cluster centers. A special class of SOM is hierarchical. Usually a hierarchical SOM refers to a tree of maps, the lower of which acts as a preprocessing stage to higher ones. As the hierarchy is traversed upwards, the information becomes more abstract. Although adding extra layers to a vector quantizer yields a higher distortion in reconstruction, it also effectively reduces the complexity of the task. Another advantage is that different kinds of representations are available from different levels of the hierarchy.

6.3 The Possible Uses of Local Information for Image Super-resolution

In iterative image super-resolution algorithms such as the *Maximum a Posteriori*, *Maximum Likelihood*, *Maximum Entropy* [51], *Expectation Maximization* [52], and *Pixon* [53] algorithms, a Fourier transform is required. The 2-D Fourier transform is a global operation that concerns all pixel values in an image. However, the 2-D Fourier transform becomes more expensive as the size of the image increases. In contrast, it is proposed that a local operation can be used to achieve image super-resolution. A local operation concerns only the pixel values in an immediate neighborhood of a specified pixel. Changing from a global operation to a local operation is particularly well suited for large images because it can be implemented as an algorithm in a massively parallel fashion. Indeed, a massively parallel operation can be implemented in hardware [4] to provide real time performance.

A Markov random field (MRF) is characterized by its local property, whereas a Gibbs random field (GRF) is characterized by its global property. The Hammersley-Clifford theorem [45] establishes the equivalence of the local property and the global property. Equivalence implies that a change in the local level causes an equivalent change in the global level. If one can perform super-resolution with a MRF at the local level, super-resolution will also happen at the global level.

In this chapter, a possible MRF-based image super-resolution algorithm is discussed. Because many 3×3 blocks can be extracted from a blurred image, it is possible to extract the super-resolved value of a center pixel by computing the relationship between this center pixel and the pixels in the cliques in its 3×3 neighborhood. It is possible to correctly estimate the true value of the center pixel in a noisy blurred image as shown in chapter 4. In chapter 4, a MRF algorithm was applied to remove the noise. With careful fine tuning of the parameters, the relative potential function based MRF algorithm should be able to perform image super-resolution. However, it would be difficult to estimate the necessary parameters to achieve the image super-resolution.

To avoid the difficulty of finding the optimal parameters for the MRF to perform image super-resolution, an alternative method was sought to achieve the same goal. The SOM approach appears to be a good alternative in this case. In SOM the spatial location of an output neuron corresponds to a particular domain or feature of the input data. The unsupervised learning clusters the redundant domains or features of the the input data. The redundancy of information represents knowledge. Completely nonredundant stimuli are indistinguishable from random noise. In training, the SOM

can learn the redundant information or knowledge. With enough data, an SOM image super-resolution algorithm can learn the image structure and the corresponding super-resolved image pixel value through unsupervised learning.

6.4 Self-Organizing Map Training for Image Super-resolution

In a self-organizing map, the neurons are placed at the nodes of a lattice that is usually one- or two-dimensional. A two-dimensional array of neurons is shown in Fig. 6.1 Higher-dimensional maps are also possible, but are not as common. The neurons become selectively tuned to various input patterns or class of patterns in the course of a competitive learning process. In the two dimensional lattice, the neurons having similar vector values tend to concentrate to form their own cluster during the training. It is believed that the SOM can learn the relationship between the blurred image pixels in the random field and the center pixel in the corresponding clear image. Therefore, one can avoid the problem of finding the optimal parameter set for the MRF algorithm in the associated image super-resolution algorithm.

The training of the SOM is shown in Fig. 6.2. The procedure is as follows:

1. *Initialization.* First, a two-dimensional lattice of neurons is created. Every neuron contains a randomized weight vector. The synaptic weight vector of neuron j is denoted by

$$\mathbf{w}_j = [\mathbf{x}_{j1}, \mathbf{x}_{j2}, \dots, \mathbf{x}_{jp}]^T, \quad j = 1, 2, \dots, N \quad (6.1)$$

2. *Sampling.* The input vector, representing the set of input signals, is denoted by

$$\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p]^T \quad (6.2)$$

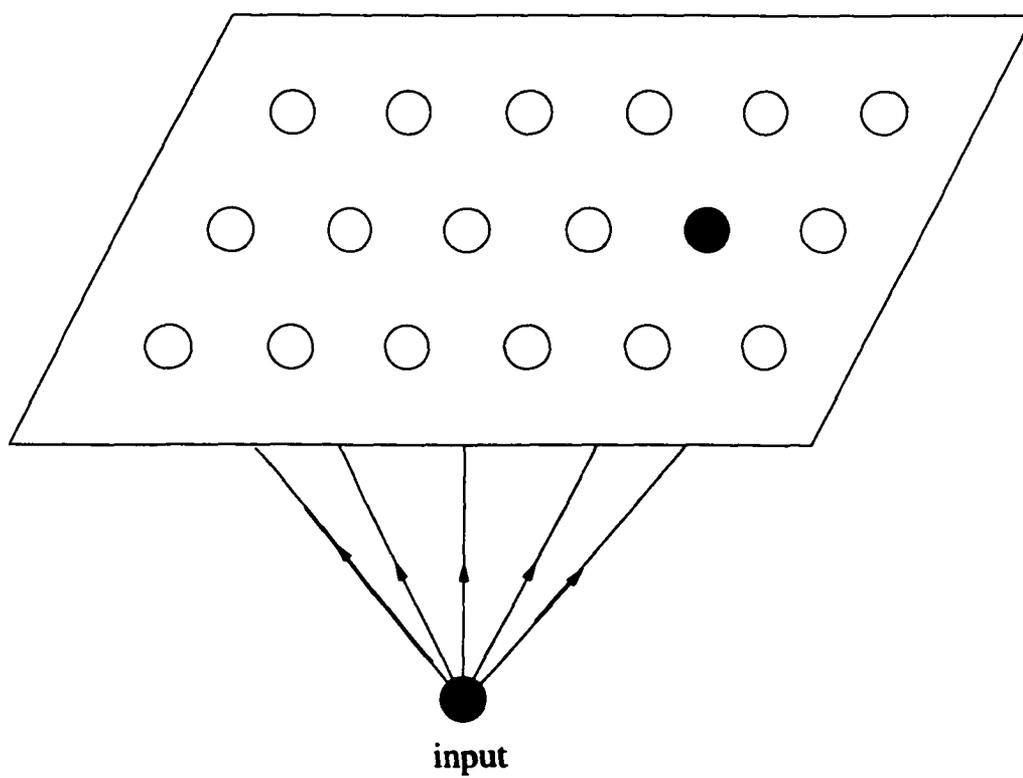


Figure 6.1: The two-dimensional self-organizing map model.

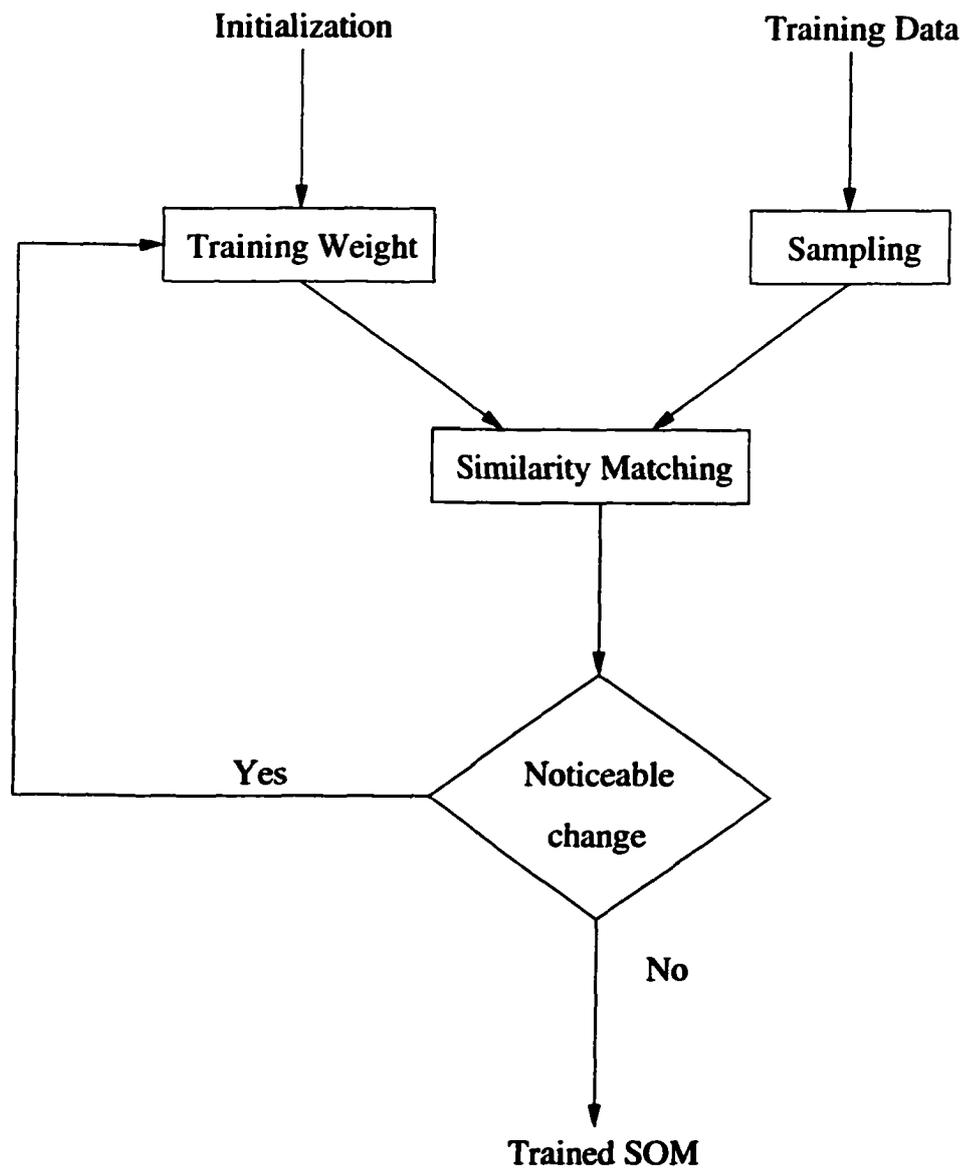


Figure 6.2: The training procedure of the self-organizing map model.

3. *Similarity Matching.* The index $i(\mathbf{x})$ identifies the neuron that best matches the input vector \mathbf{x} . The Euclidean norm is calculated to find the closest match between the neuron weight and the input vector, *i.e.*,

$$i(\mathbf{x}) = \arg \min_j \|\mathbf{x} - \mathbf{w}_j\|, \quad j = 1, 2, \dots, N \quad (6.3)$$

where $\|\cdot\|$ denotes the Euclidean norm of the argument vector. The particular neuron i that satisfies this condition is called the *best-matching* or *winning neuron* for the input vector \mathbf{x} . Depending on the application of interest, the response of the network could be either the index of the winning neuron, or the synaptic weight vector that is closest to the input vector in a Euclidean sense. In the proposed algorithm, the response of the network is the index of the winning neuron.

4. *Updating.* In the update process, the neurons in the neighborhood $\Lambda_{i(\mathbf{x})}(n)$ of the winning neuron will be updated. The neighborhood function $\Lambda_{i(\mathbf{x})}(n)$ is selected fairly wide at first and then is permitted to shrink with time n .

The update formula for the weights in the neighborhood of the winning neuron is

$$\begin{aligned} \mathbf{w}_j(\mathbf{n} + 1) &= \mathbf{w}_j(\mathbf{n}) + \eta(\mathbf{n})\pi_{j,i(\mathbf{x})}(\mathbf{n})[\mathbf{x} - \mathbf{w}_j(\mathbf{n})], \mathbf{j} \in \Lambda_{i(\mathbf{x})} \\ &= \mathbf{w}_j(\mathbf{n}), \text{ otherwise} \end{aligned} \quad (6.4)$$

where $\eta(n)$ is the corresponding value of the learning-rate parameter at a given time n and $\pi_{j,i}$ is a Gaussian-type function used to calculate the neighborhood magnitude. The effect of the update equation 6.4 is to move the synaptic weight vector \mathbf{w}_j of the winning neuron i towards the input vector \mathbf{x} . The Gaussian function $\pi_{j,i}$ is defined as

$$\pi_{j,i} = \exp\left(-\frac{d_{j,i}^2}{2\sigma^2}\right) \quad (6.5)$$

where $\sigma(n)$ is defined as

$$\sigma(n) = \sigma_0 \exp\left(-\frac{n}{\tau_1}\right) \quad (6.6)$$

and where $\eta(n)$ is defined as

$$\eta(n) = \eta_0 \exp\left(-\frac{n}{\tau_2}\right) \quad (6.7)$$

the constants σ_0 and η_0 being, respectively, the values of $\sigma(n)$ and $\eta(n)$ at the initiation of the SOM algorithm, and τ_1 and τ_2 being their respective time constants.

5. *Continuation.* Continue with step 2 until no noticeable changes in the SOM weights are observed.

The algorithm leads to a topological ordering of the SOM in the input space in the sense that the neurons that are adjacent in the lattice will tend to have similar synaptic weights.

To implement the SOM-based image super-resolution algorithm, we try to mimic the MRF approach. A blurred image and clear image pair of the same scene are used as a training set. A corresponding pair of 3×3 blocks are extracted from both images and are used to build the training vector. However, the MRF only uses the surrounding eight pixels and the center pixel to estimate the center pixel. Thus, it is logical that only the center pixel from a clear image is needed to train the SOM. In fact, it is the one that must be estimated. The construction of the input vector is shown in Fig. 6.3. The input vector to the SOM is a 1×10 vector. The first nine values are from the blurred 3×3 image block. The tenth value of the input vector is the center pixel of the clear image as shown in Fig. 6.3.

In the image super-resolution processing, only the blurred 3×3 block from the blurred image is available. Thus, the input vector does not contain the center pixel of the clear image. This blurred 3×3 block will be converted to a 1×9 input vector. Since the SOM weight is a $N \times 10$ matrix, the SOM weight that contains the information of the blurred image has to be extracted as a partial weight and is a $N \times 9$ matrix. The partial input vector must be used to find the match of the partial SOM weight that contains only the information of the blurred image data as shown in Fig. 6.4.

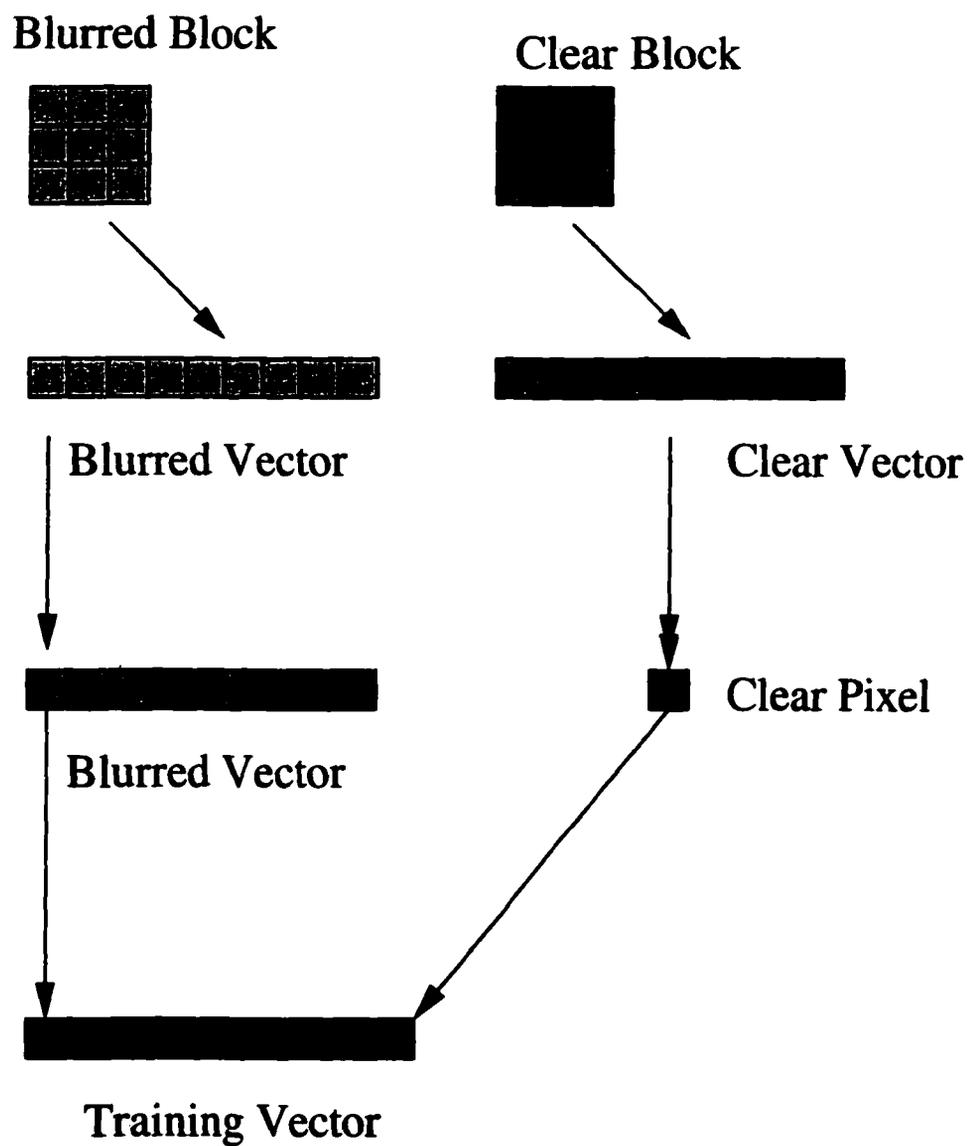


Figure 6.3: The construction of the input vector.

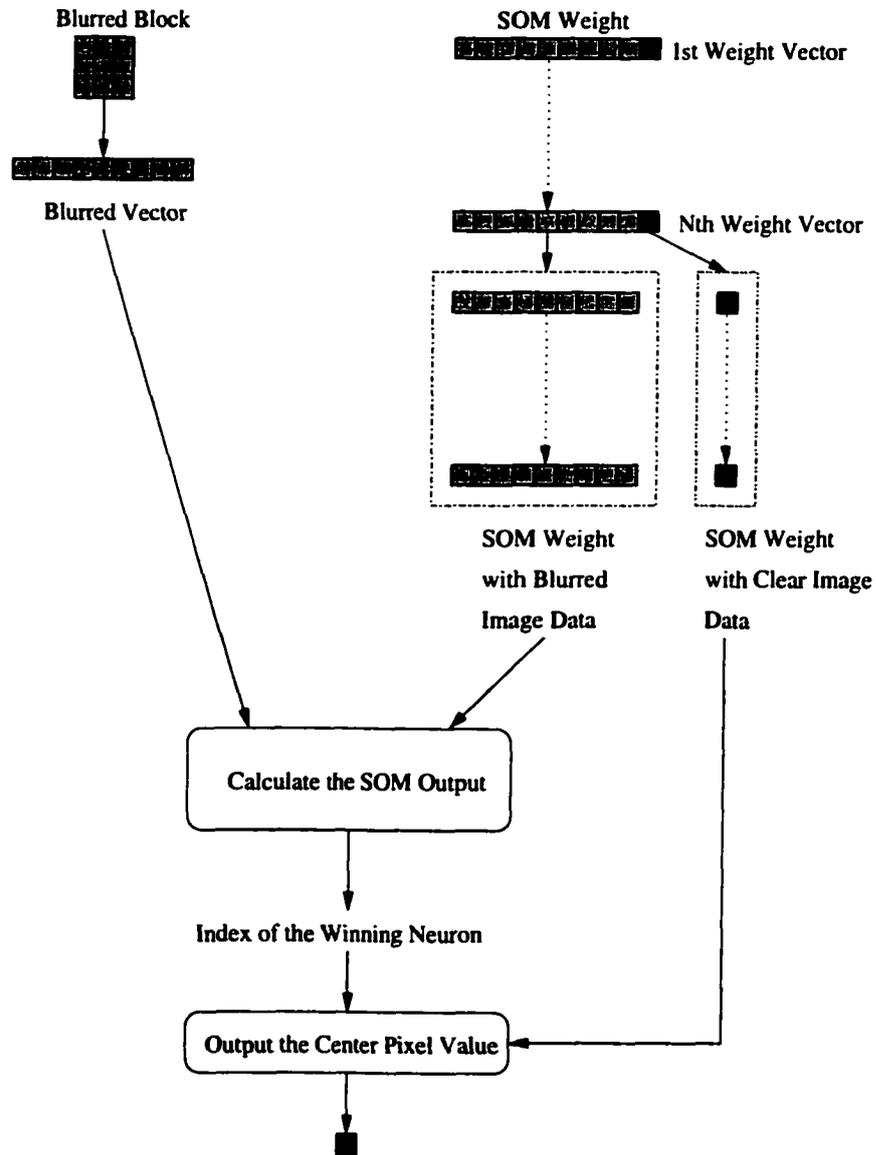


Figure 6.4: The SOM based image super-resolution process.

The search for the match between the partial SOM weight and the blurred image vector is based on the Euclidean distance.

$$i(\mathbf{x}) = \arg \min_j \|\mathbf{x} - \mathbf{w}_j^{\text{Partial Weight}}\|, j = 1, 2, \dots, N \quad (6.8)$$

where $\mathbf{w}_j^{\text{Partial Weight}}$ is the partial weight that contained the information of the blurred image structure as shown in Fig. 6.4. Once the winning neuron has been selected. The index of the winning neuron is used to extract the corresponding partial SOM weight that contained the clear image pixel. The corresponding center pixel of the clear image block will be used to replace the center pixel of the blurred image block.

6.5 Validation Experiment

In this experiment, the 2-D SOM lattice of neurons is a 46×46 lattice and contains 2116 neurons. Each neuron has a 1×10 vector. A 10 megabyte remote sensing image is used as the object. However, due to the size of the image, we have to extract 74 512×512 images as the training images to reduce the memory usage in the workstation. In one iteration cycle, the SOM has to upload all 74 images in sequence. The SOM will upload a 512×512 blurred image and its corresponding clear image. The SOM will start scanning at the second row and will scan the images from the second column to the 511 column. After completing the scanning of the second row, the SOM will move down one row and repeat the scanning. When the SOM scan occurs, a 3×3 block in the blurred and a single center pixel in the clear image are scanned at the same time. The 3×3 blurred block is rearranged into a

1×9 vector. The value of the clear pixel will be added to the end of the vector to form a 1×10 vector. The construction of this SOM training vector is shown in Fig. 6.3. After the scanning of one pair of images is finished and the SOM training is completed, the 2-D SOM algorithm will upload the second pair.

In section 6.4, the training process of the SOM weight was presented. The five step process includes *initialization*, *sampling*, *similarity matching*, *updating*, and *continuation*. In one iteration cycle, $(N - 2) \times (N - 2)$ pixels would be updated if the image size was $N \times N$. In the beginning, the clear pixel of the clear image that would be scanned is located at pixel (2, 2) in the traditional (y, x) coordinate system used in matrix computations. The blurred 3×3 image block of the blurred image occupied nine pixels including, (1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3). Once all nine pixels values are uploaded to the first nine elements of the training vector, the clear pixel value will be uploaded to the 10th element of the training vector as shown in Fig. 6.3. The above process built one training vector. This completes step two, *sampling*, of the training process introduced in section 6.4. Step three, *similarity matching*, of the training process, would then be performed. The index of the winning neuron would be returned to the algorithm. In step four of the training process, *updating*, the winning neuron and the neurons in the winning neuron's neighborhood will be updated by Eq.(6.4). The process will repeat until all of the image pairs are uploaded. This is counted as one iteration cycle. In step five of the training process, *continuation*, one continues step two until no noticeable changes are observed in the SOM weight. Normally, the training cycle will continue until the change of the winning neuron is smaller than some small value ϵ .

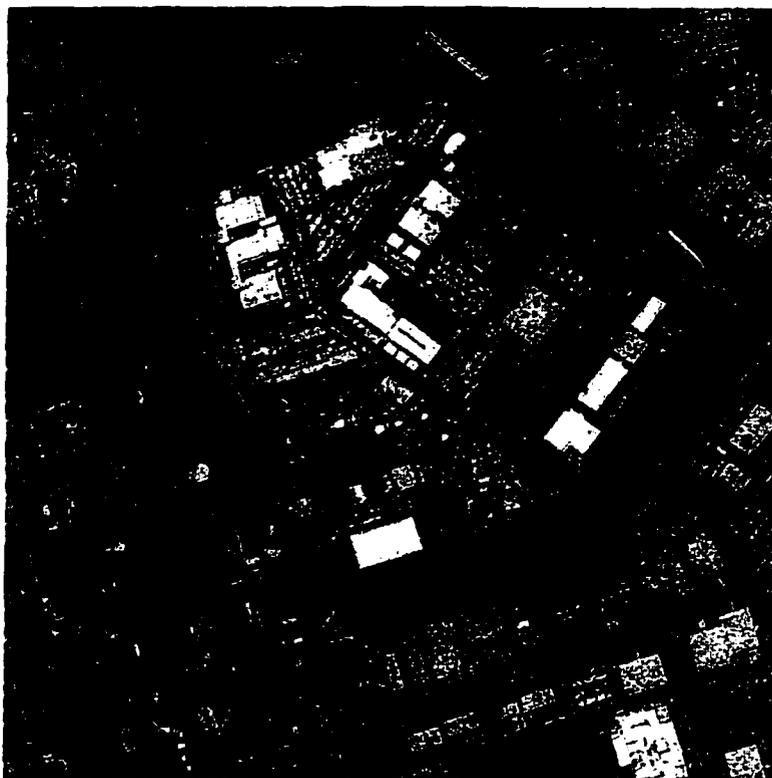


Figure 6.5: The clear training image.

The original 10 MB image is too large to show in this document. Instead, we will show one pair of clear and blurred images. The clear image is shown in Fig. 6.5. The blurred image is shown in Fig. 6.6.

The SOM was trained for 100 cycles. After the training was finished, a blurred test image, Fig. 6.8, was used to test the performance of this new method. Visually, the acquired blurred image and the SOM restored image, Fig. 6.9, look very similar. However, after careful scrutiny it was found that the SOM restored image was a little sharper than the acquired image. The clear image is shown in Fig. 6.7.

Visually, there is not much difference between the restored image and the blurred image. By careful observation, better restoration of the edges can be observed. A



Figure 6.6: The blurred training image.

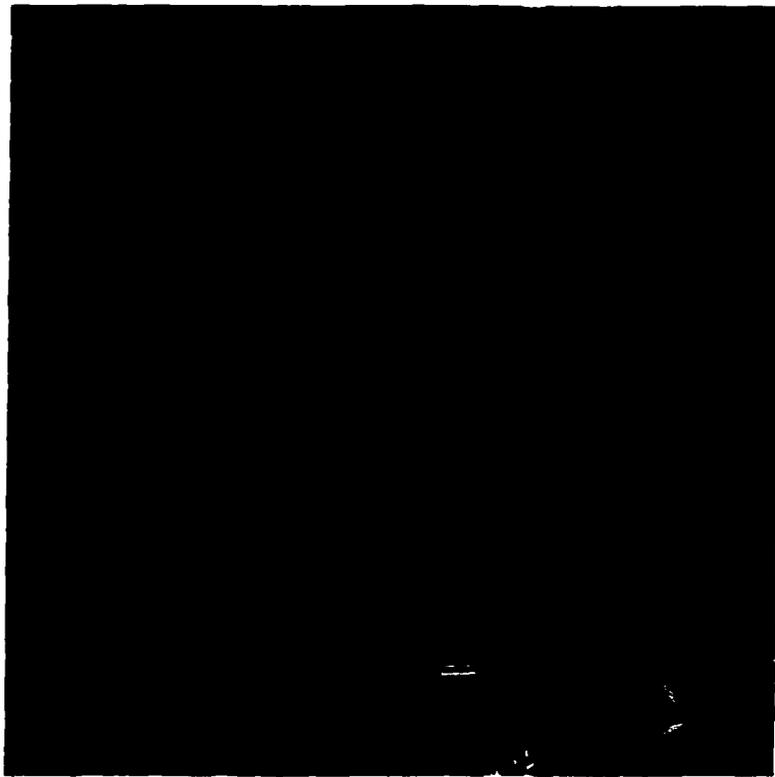


Figure 6.7: The original image.

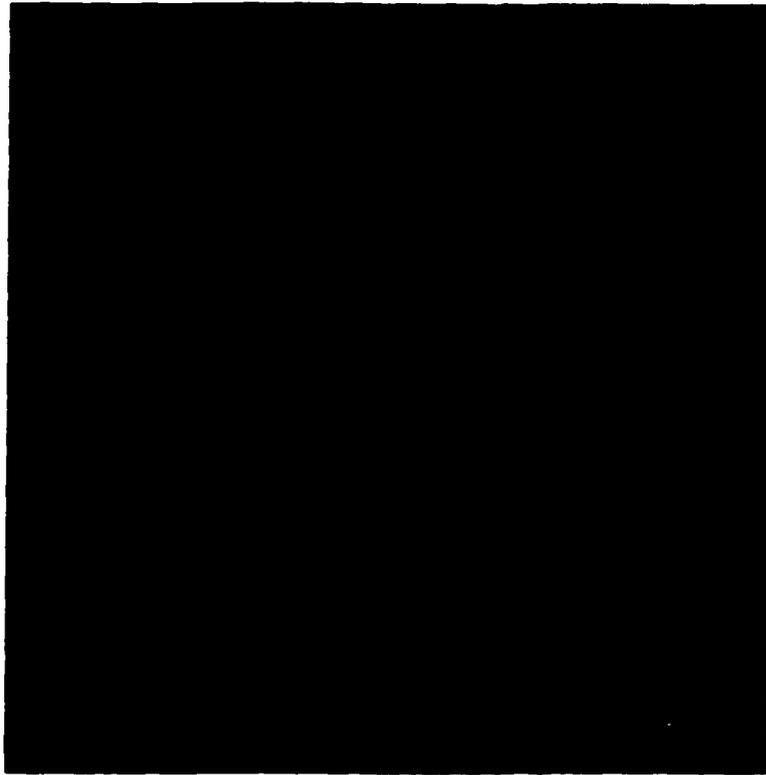


Figure 6.8: The acquired blurred image.

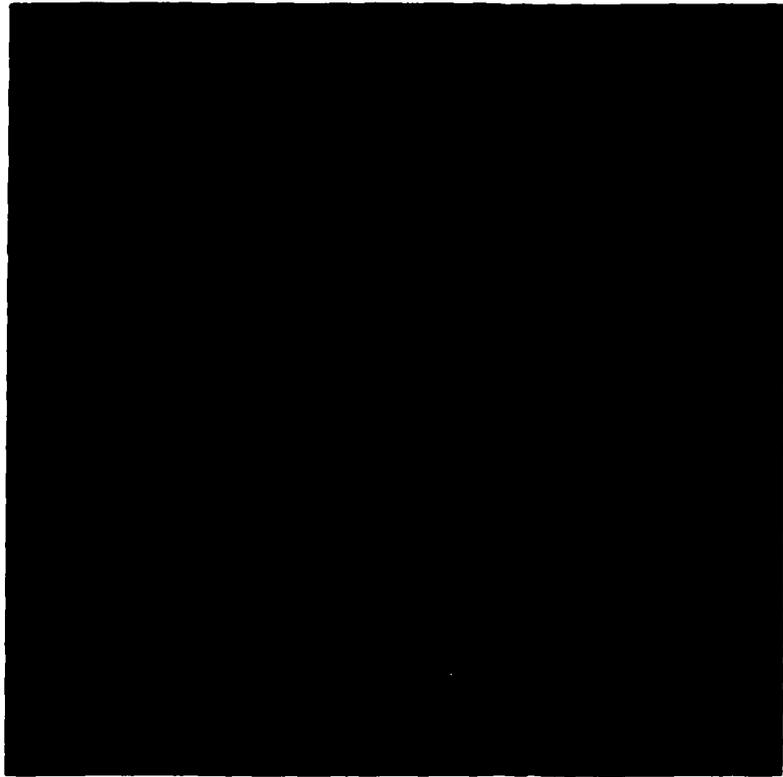


Figure 6.9: The restored image.

quantification of the quality of the image is needed for these comparisons. The mean-square-error is chosen for its simplicity. The mean-square-error (MSE) between the restored image and the clear image, and between the blurred image and the clear image are compared. The MSE between the blurred image and the clear image is 185.4979 while the MSE between the restored image and the clear image is 78.3746. The MSE value of the restored image had lower than half of the MSE value that the blurred image did. In principle, this experimental result demonstrates that the SOM can be used in image restoration. The only draw back of this experiment is our relatively small SOM weight matrix. A better restoration result is expected if the SOM matrix is larger. More can be done for this SOM approach to achieve a real image super-resolution performance.

6.6 Conclusions

The SOM image super-resolution algorithm did not perform as well as was anticipated. The super-resolved image is only 136.68% better than blurred image when MSE values are compared. However, the above results show that this SOM approach can restore an image, in principle. There are only 2116 neurons in the trained weight and the image has 256 gray levels. Therefore, on average, eight neurons handle a given gray level. In reality, the number of gray level differences in the neighborhood pixels must be more than eight. The SOM algorithm performs data compression to store the information. However, the compression rate is high. The compression rate can be reduced by increasing the number of neurons. Therefore, increasing the

number of neurons will generate better results. Other possible improvements will be discussed in Chapter 7.

CHAPTER 7

SUMMARY AND FUTURE WORK

7.1 Closing Remarks

Image processing has been more art than science for last thirty years. The most common image processing problem is the inverse problem. Unfortunately, the inverse problem is ill-posed. The ambiguity brought forth by the imaging operators are in effect a many-to-one mapping. Current approaches to this ill-posed problem deal with the estimation process as an optimization problem, often with constraints and often with some statistical intuition. Many times, researchers just change parameters and develop new algorithms. Most new algorithms may only work in certain situations after the parameters have been altered. A new breakthrough rarely happens in the field of image processing. Maybe the image processing problem is just too difficult? Maybe the lack of a systematic approach is the real cause of the slow development of the image processing? A new way to approach image processing problems has to be developed.

7.2 Contribution

In this dissertation, a new approach to image processing, a dynamic imaging system based on information theory, was proposed. The success of Shannon communications theory has its roots in applying abstract/statistical model to the communications process. On the other hand, image processing is always based on the physical/deterministic models of the imaging systems under consideration. To incorporate Shannon communications theory into image processing, it was suggested in Chapter 2 that a new systematic approach could be developed to accelerate progress in image processing research. Researchers can rearrange the abstract system blocks and think about image processing at an abstract level. The proposed conceptual dynamic imaging system was developed based on this Shannon communications theory approach. This proposed system incorporates information theory and a dynamic decision making system to optimize the image restoration/image super-resolution strategy to achieve better image restoration/image super-resolution. The importance of the information visualization and the psychophysical image enhancement to the human user were also discussed. A Markov random field based noise removal algorithm, a hybrid maximum likelihood and constraint based image super-resolution algorithm, and a self organizing-map based image super-resolution algorithm were introduced in this dissertation as components of this dynamic imaging system.

In Chapter 4, a Markov random field based channel decoder for image processing was developed. The texture of the scene was used as the channel encoded information to assist the image noise removal operation. The pattern of the texture was assumed to be deterministic and was used to determine a pixel value based on the surrounding

pixels' values. With the encoded information from the texture operation, a Markov random field based noise removal algorithm was developed as the channel decoder. A new relative potential function was developed to improve the noise removal performance of the iterative conditional mode (ICM) algorithm. The sign of the relative potential function assists the MRF algorithm to determine the effect of the distortion of the noise to the center pixel in its 3×3 neighborhood. With the above information, an intelligent adjustment of the pixel value updating assisted the algorithm to avoid magnifying noise with a relative negative value against the surrounding pixels. The proposed filter was used as a channel decoder in the dynamic imaging system as described in Chapter 3. The proposed filter outperforms the traditional ICM algorithm and a 3×3 median filter in the noise removal process. The proposed filtered image was super-resolved with the Maximum Likelihood (ML) algorithm as discussed in Chapter 3. The validity of the relative potential function approach was proved mathematically based on a proof of convergence using Gibbs-Markov equivalence arguments. This proof of the Markov-Gibbs equivalence for a Gibbs random field with a relative potential function was given in appendix A.

In Chapter 5, an object-boundary-constrained maximum likelihood (OBC-ML) algorithm was developed. The width of the object boundary was viewed as prior information for the image restoration/image super-resolution algorithm. After extracting the object from the image based on the detected object contour, the maximum likelihood (ML) algorithm was applied to perform image super-resolution on the extracted object. It was demonstrated that the OBC-ML super-resolved image had a higher resolution than the ML super-resolved image did. Based on several 1-D and

2-D validation experiment results, mean-square-error measurements and residue measurements show that the OBC-ML algorithm outperforms the ML approach in image super-resolution applications when a reasonable object contour is detected. Detailed control experiments were performed to show that even without an absolutely correct object contour, the OBC-ML algorithm still outperformed the ML algorithm. In some extreme cases, with prior knowledge of the object contour, it was shown that the OBC-ML can super-resolve an unrecognizable image. The OBC-ML approach has potential for applications in remote sensing cases in which GPS and accurate mapping information can be obtained to assist the OBC-ML algorithm. Thus, it is expected that the OBC-ML algorithm can be used to increase the image resolution of a remote sensing image.

In Chapter 6, a super-resolution algorithm based on a self-organizing map (SOM) was developed. The SOM applied unsupervised training to learn the mapping between a 3×3 blurred image block and a corresponding clear image pixel. In the training process, pairs of blurred image blocks and a clear pixel were extracted from the corresponding blurred image and clear image. This SOM approach mimics the Markov random field (MRF) approach in pixel value estimation without fine tuning the parameters in the MRF algorithm. It was demonstrated that the SOM restored image had lower MSE values than the blurred image. In principle, it was shown that this SOM approach works. The performance of the SOM approach is limited by the size of the SOM weight lattice. Due to the limit of the SOM size in the experiments, the demonstrated enhancement of the image super-resolution performance was minimal. With a larger SOM size, the performance is expected to increase.

Many potential enhancements in image super-resolution could be achieved with this proposed SOM algorithm.

7.3 Integration and Further Development

In this dissertation, we proposed a new dynamic imaging system (DIS) and developed three related components. The Markov random field based channel decoder is used to remove noise generated in the CCD array. It is one of the noise removal algorithms that can be used for noise removal. However, it is especially suitable for removing strong noise such as "salt and pepper" noise that is generated by saturated pixels and damaged pixels. For weak noise, a simple ideal low pass filter is suitable. The development of the channel decoder of the DIS is thus complete.

For a source decoder, the OBC-ML algorithm can be applied in a situation that presents a clear background. In this case, the target contour can be easily extracted and can be incorporated for use in the OBC-ML algorithm. A scene with targets in a blue sky background is ideal for the OBC-ML algorithm. Another candidate category would be astronomical images, *i.e.*, a black background with stars is also ideal for the OBC-ML algorithm. For the super-resolution algorithm based on a self-organizing map (SOM), a suitable application case would be a large image with targets that are hard to detect. The proposed SOM-based super-resolution algorithm can perform parallel processing in a large scale image in a fraction of the time required by Fourier transform based super-resolution algorithms such as *ML*, *MAP*, and *Pixon* when parallel processing computer hardware is available. In this case, the detection of the targets is dependent on the user. A psychophysical image enhancement algorithm

can be deployed to assist the user in target searching. For a small scale image, the *ML*, *MAP*, and *Pixon* super-resolution algorithms can be applied. For a single image, the development of the source decoder of the DIS is thus complete.

The next step is the development of an effective video sequence super-resolution algorithm for the source decoder of the DIS. Temporal information can be used to extract sub-pixel information and assist the reconstruction of the object in a video sequence. A video sequence super-resolution algorithm would thus make the source decoder more effective.

The remaining components needed to complete the DIS are a dynamic decision making system and a post-processing system.

For the post-processing stage, two items need to be developed.

- A psychophysical image enhancement algorithm is needed to aid the user to see targets or significant features in the image.
- An information visualization algorithm is needed to represent the priority of the targets in the scene.

For the dynamic decision making system, several items need to be developed:

- A set of task specific parameters to use in a real environment.
- An image quality assessment algorithm to analyze the quality of the acquired image from the CCD array. This algorithm would assist the hybrid rule-based artificial intelligence (AI)/fuzzy neural network decision making system to decide if a noise removal process is needed. It would also help determine the

quality of the super-resolved image. The image quality assessment algorithm would assist the hybrid rule-based artificial intelligence (AI)/fuzzy neural network decision making system to find the optimal super-resolved image during the iterative processing stage.

- A feature extraction algorithm to find features of targets in a scene. It would assist the hybrid rule-based artificial intelligence (AI)/fuzzy neural network decision making system to locate the targets.
- A hybrid rule-based artificial intelligence (AI)/fuzzy neural network decision making system to make optimal decisions for the acquired image. The rule-based artificial intelligence (AI) algorithm would handle well defined scenarios. The fuzzy neural network would handle uncertain scenarios.

7.4 Further Development

If the proposed dynamic imaging system were to be completed, one could move on to another phase of this research. The following items are additional, closely related topics suggested for future research efforts.

- The fuzzy neural network can be replaced by a dynamic neural network [69]. The dynamic neural network is ideal for handling temporal information. The hybrid rule-based artificial intelligence (AI)/dynamic neural network decision making system can handle video sequences from the CCD array; hence, it can perform tracking of the region of interest (ROI) and control the super-resolution algorithm to super-resolve a video sequence. The dynamic neural

network combines supervised training and unsupervised training to reduce the overall training complexity. It should also outperform other neural networks in its ability to track targets.

- The pixel value updating for the Markov random field based channel decoder is slow and requires several iterations to converge. However, the simplex [70] algorithm can provide a fast non-iterative process in the pixel value update routine to improve the accuracy of the Markov random field based channel decoding.
- The difficulty of applying the Markov random field (MRF) approach to image super-resolution is related to finding the optimal parameters for the MRF. The genetic algorithm [71] could be used to search for the optimal parameters for the MRF based image super-resolution algorithm.
- The time to train the SOM and to search the SOM for the clear pixel will be increased as the SOM weight lattice increases. The hierarchical SOM approach could be applied to build a hierarchical scheme for a specific image structure to reduce the training time and the searching time.
- Hybrid video sequence image super-resolution based on well known super-resolution techniques such as *ML*, *MAP*, and *Pixon* could be combined with a dynamic neural network [69] to extract sub-pixel resolution from the temporal information of the video sequence.

Several of these suggested research directions are currently under consideration and may lead to future efforts.

Appendix A

PROOF OF GIBBS-MARKOV EQUIVALENCE FOR A GIBBS RANDOM FIELD WITH A RELATIVE POTENTIAL FUNCTION

Proof of Theorem 1:

It is necessary to define a neighborhood system in the beginning of this appendix. The site i in S is related to other sites in S through a neighborhood system. A neighborhood system for S is defined as

$$N = \{N_i | \forall i \in S\} \quad (\text{A.1})$$

where N_i is the set of sites neighboring i , with the following restrictions

1. a site is not a neighbor to itself: $i \notin N_i$
2. the neighboring relationship is mutual; $i \in N_{i'} \iff i' \in N_i$

The neighbor set of i is defined as the set of nearby sites within a radius of r for a lattice S .

$$N_i = \{i' \in S | [EuclideanDist(pixel_{i'}, pixel_i)]^2 \leq r, i' \neq i\} \quad (\text{A.2})$$

Note that the Euclidean distance, *EuclideanDist*, and the radius r are integer valued for a discrete grid.

Let $G = G_1, G_2, \dots, G_n$ be a family of random variables defined on the set S ; each random variable G_i takes a value g_i in a discrete label set \mathcal{L} . The family G is a

random field. The relation $G_i = g_i$ denotes the event that G_i takes the value g_i . The set of relations $(G_1 = g_1, \dots, G_n = g_n)$ denotes the joint event. A joint event can be expressed as $G = g$ where $g = g_1, \dots, g_n$ is a configuration of G . The probability that the random variable G_i takes the value g_i is denoted $P(G_i = g_i)$. Its abbreviation is $P(g_i)$. The joint probability can be written as $P(G = g) \Leftrightarrow P(G_1 = g_1, \dots, G_n = g_n)$ and its abbreviation is $P(g)$.

The field F is a MRF if and only if the following two conditions are satisfied:

1. The positivity condition

$$P(g) > 0, \forall g \in G \quad (\text{A.3})$$

2. The Markovianity condition

$$P(f_i | f_{S-\{i\}}) = P(f_i | f_{N_i}) \quad (\text{A.4})$$

where $S - i$ is the set difference, $f_{S-\{i\}}$ denotes the set of labels at the sites in $S - \{i\}$, and

$$f_{N_i} = \{f_{i'} | i' \in N_i\} \quad (\text{A.5})$$

is the set of labels at the sites neighboring i .

It is useful to show the original Hammersley-Clifford proof of the Gibbs-Markov equivalence [38, 40, 45, 72]. One can employ the Gibbs distribution representation of the joint probability $P(g)$:

$$P(g) = Z^{-1} \times e^{(-\frac{1}{T}\Phi)} \quad (\text{A.6})$$

where

$$Z = \sum_{g \in F} e^{(-\frac{1}{T}\Phi)} \quad (\text{A.7})$$

is a normalizing constant called the partition function, Φ is an “energy function” and T denotes a “temperature” variable. The energy function Φ is obtained from summing a set of “potential functions” $V_c(\cdot)$ developed for each clique c , *i.e.*

$$\Phi = \sum_{c \in Q_s} V_c(g(s)) \quad (\text{A.8})$$

where Q_s is a set of cliques. The potential function $V_c(\cdot)$ is defined as

$$V_c(g(s)) = \left| g(s) - \frac{1}{n_c} \sum_{t \in c} g(t) \right| = |g(s) - \bar{g}(s, c)| \quad (\text{A.9})$$

In Eq.(A.9), $\bar{g}(s, c)$ denotes the mean of the neighboring pixel intensities in clique c .

The temperature T is selected as

$$T = \sigma_s \times n_c \quad (\text{A.10})$$

where σ_s is a standard deviation of $\bar{g}_{s,c}$ in Q_s , and n_c is the number of sites in cliques c .

It is simple to see that the conditional probability $P(g_i | g_{\partial s - i})$ can be expressed as

$$P(g_i | g_{\partial s - i}) = \frac{P(g_i, g_{\partial s - i})}{P(g_{\partial s - i})} \quad (\text{A.11})$$

where $P(g_i, g_{\partial s - i})$ is a Gibbs distribution on S with respect to the neighborhood system N . Therefore,

$$P(g_i, g_{\partial s - i}) = P(g) \quad (\text{A.12})$$

and $P(g_{\partial s-i})$ is the independent Gibbs distribution on S with respect to the neighborhood system $N_1, N_2, \dots, N_{i-1}, N_i, N_{i+1}, \dots, N_n$, given by the expression

$$P(g_{\partial s-i}) = \sum_{g'_i \in \mathcal{C}} P(g') \quad (\text{A.13})$$

Therefore,

$$P(g_i | g_{\partial s-i}) = \frac{P(g)}{P(g_{\partial s-i})} = \frac{P(g)}{\sum_{g'_i \in \mathcal{C}} P(g')} \quad (\text{A.14})$$

where $g' = g_1, \dots, g_{i-1}, g_{i+1}, \dots, g_m$ is any configuration that agrees with g at all sites except i . Now, replacing $P(g)$ with a Gibbs distribution $P(g) = Z^{-1} \times e^{-\sum_{c \in \mathcal{C}} V_c(g)}$, where Z is a normalization constant, we have

$$P(g_i | g_{\partial s-i}) = \frac{e^{-\sum_{c \in \mathcal{C}} V_c(g)}}{\sum_{g'_i} e^{-\sum_{c \in \mathcal{C}} V_c(g')}} \quad (\text{A.15})$$

Let \mathcal{C} consist of two sets \mathcal{A} and \mathcal{B} . The set \mathcal{A} consists of cliques where i resides and the set \mathcal{B} consists of cliques where i does not reside. Eq.(A.15) can be rewritten as

$$P(g_i | G_{\partial s-i}) = \frac{[e^{-\sum_{c \in \mathcal{A}} V_c(g)}] [e^{-\sum_{c \in \mathcal{B}} V_c(g)}]}{\sum_{g'_i} [e^{-\sum_{c \in \mathcal{A}} V_c(g')}] [e^{-\sum_{c \in \mathcal{B}} V_c(g')}]} \quad (\text{A.16})$$

Note that $V_c(g) = V_c(g')$ for any clique that does not contain i . Therefore,

$$e^{-\sum_{c \in \mathcal{B}} V_c(g)} = e^{-\sum_{c \in \mathcal{B}} V_c(g')} \quad (\text{A.17})$$

and the Gibbs distribution associated with the set \mathcal{B} can be cancelled from both the numerator and the denominator. Thus, the conditional probability (A.16) involves

only the potentials of cliques containing i , and

$$P(g_i | g_{\partial s - i}) = \frac{e^{-\sum_{c \in \mathcal{A}} V_c(g)}}{\sum_{g'_i} e^{-\sum_{c \in \mathcal{A}} V_c(g')}} \quad (\text{A.18})$$

The expression (A.18) establishes the Gibbs-Markov equivalence.

For finding the proof of Gibbs-Markov equivalence for a Gibbs random field defined by a relative potential function, replace $P(g)$ with a Gibbs distribution with a relative potential function expression. To make the proof clear, several symbols have to change. The new relative potential function W_c is defined as:

$$W_c(g(s)) = g(s) - \frac{1}{n_c} \sum_{t \in c} g(t) = g(s) - \bar{g}(s, c) \quad (\text{A.19})$$

rather than with the absolute value in Eq.(4.7). The Gibbs distribution with relative potential function is $P(g) = Z^{-1} \times e^{-|\sum_{c \in \mathcal{C}} W_c(g)|}$, where Z is a normalization constant. Then the conditional probability becomes

$$P(g_i | g_{\partial s - i}) = \frac{e^{-|\sum_{c \in \mathcal{C}} W_c(g)|}}{\sum_{g'_i} e^{-|\sum_{c \in \mathcal{C}} W_c(g')|}} \quad (\text{A.20})$$

Let \mathcal{C} consist of two sets \mathcal{A} and \mathcal{B} . The set \mathcal{A} consists of cliques where i resides and the set \mathcal{B} consists of cliques where i does not reside. Eq.(A.20) can then be rewritten as

$$P(g_i | g_{\partial s - i}) = \frac{\left[e^{-|\sum_{c \in \mathcal{A}} W_c(g) + \sum_{c \in \mathcal{B}} W_c(g)|} \right]}{\sum_{g'_i} \left[e^{-|\sum_{c \in \mathcal{A}} W_c(g') + \sum_{c \in \mathcal{B}} W_c(g')|} \right]} \quad (\text{A.21})$$

By the triangle inequality [73], $|x+y| \leq |x|+|y|$ and $-|x+y| \geq -|x|-|y|$. Therefore, $e^{-|x+y|} \geq e^{-|x|}e^{-|y|}$, and

$$P(g_i|g_{\partial s-i}) \leq \frac{\left[e^{-|\sum_{c \in \mathcal{A}} W_c(g)|} \right] \left[e^{-|\sum_{c \in \mathcal{B}} W_c(g)|} \right]}{\sum_{g'_i} \left[e^{-|\sum_{c \in \mathcal{A}} W_c(g')|} \right] \left[e^{-|\sum_{c \in \mathcal{B}} W_c(g')|} \right]} \quad (\text{A.22})$$

Again, one knows that $W_c(g) = W_c(g')$ for any clique that does not contain i . Therefore,

$$e^{-|\sum_{c \in \mathcal{B}} W_c(g)|} = e^{-|\sum_{c \in \mathcal{B}} W_c(g')|} \quad (\text{A.23})$$

The Gibbs distribution with a relative potential function associated with the set \mathcal{B} can then be cancelled from both the numerator and the denominator. Thus, this conditional probability involves only the potentials of cliques containing i , and

$$P(g_i|g_{\partial s-i}) \leq \frac{e^{-|\sum_{c \in \mathcal{A}} W_c(g)|}}{\sum_{g'_i} e^{-|\sum_{c \in \mathcal{A}} W_c(g')|}} \quad (\text{A.24})$$

The Eq.(A.24) establishes the upper bound of the Gibbs-Markov equivalence for Gibbs distribution with relative potential function.

Equation(A.18) can be applied to find the lower bound of the Gibbs-Markov equivalence for Gibbs distribution with relative potential function. One knows

$|\sum_{c \in \mathcal{C}} W_c(g')| \leq \sum_{c \in \mathcal{C}} V_c(g')$. Therefore, $e^{-|\sum_{c \in \mathcal{C}} W_c(g')|} \geq e^{-\sum_{c \in \mathcal{C}} V_c(g')}$; and one can write

$$0 \leq \frac{e^{-\sum_{c \in \mathcal{C}} V_c(g)}}{\sum_{g'_i} e^{-\sum_{c \in \mathcal{C}} V_c(g')}} \leq P(g_i|g_{\partial s-i}) \leq \frac{e^{-|\sum_{c \in \mathcal{C}} W_c(g)|}}{\sum_{g'_i} e^{-|\sum_{c \in \mathcal{C}} W_c(g')|}} \leq 1 \quad (\text{A.25})$$

The bound expression(A.25) sets the range of $P(g_i|g_{\partial s-i})$. Note that when $g(s) - \frac{1}{n_c} \sum_{t \in c} g(t) = g(s) - \bar{g}(s, c) > 0$, $V_c = |W_c|$. The upper bound and the lower bound will converge. This shows one special case for the convergence of the Gibbs-Markov equivalence for Gibbs distribution with relative potential function. This shows that the proposed algorithm is correct when the noise on a given pixel has positive value. When the noise on a given pixel has a negative value, the Gibbs random field is bounded between zero and one. This proof demonstrates that the proposed algorithm will converge.

Appendix B

MAXIMUM LIKELIHOOD ALGORITHM DERIVATION

B.1 Maximization of the Likelihood of the Image

The maximum likelihood (ML) algorithm can be derived for image restoration by applying a Bayesian approach with Poisson Probability [74]. Given the likelihood law $p(f|g)$ and fixed data f, g must have the property that it maximizes the likelihood of the data,

$$\hat{f} = \underset{f}{\operatorname{argmax}} J_{ML}(f, g) \quad (\text{B.1})$$

where

$$J_{ML}(f, g) = p(f|g) = \frac{p(g|f)p(f)}{p(g)} \quad (\text{B.2})$$

The likelihood can be simplified when its logarithm is taken. The logarithm of a quantity monotonically increases with the quantity. Therefore, Eq.(B.2) is equivalent to

$$\ln(J_{ML}(f, g)) = \ln(p(g|f)) + \ln(p(f)) - \ln(p(g)) \quad (\text{B.3})$$

The first term in Eq.(B.3) states the maximum likelihood objective. The second term is the *Maximum a Posteriori* (MAP) objective which is the prior knowledge of the object. In the ML estimation, the MAP objective does not exist. Therefore, the second term is dropped. The last term is not a function of the estimate. Therefore,

its first derivative does not exist and it is subsequently dropped from the formulation. One finds that statistical models for the images are needed for the estimation.

B.2 Poisson Statistical Model of the Object and Image

The statistical model of the object is not required in the ML estimation. However, it would be useful to explain it. The object under observation is an unknown deterministic entity. The spatial statistics in intensity f due to the emission or reflection from the object surface can be modeled as a Poisson distribution. The object can be treated as a combination of points in three dimensional space. Therefore, the photons emitted or reflected by the object can be modeled as a collection of independent point sources in three dimensional space.

To simplify the model, we can treat all of the independent point sources as being projected onto a two dimensional space. Therefore, the object's probability density function (PDF) is described as

$$p(f) = \prod_{x,y} \frac{[\bar{f}(x,y)]^{f(x,y)} e^{-\bar{f}(x,y)}}{f(x,y)!} \quad (\text{B.4})$$

The mean photon emission rate at a particular (x, y) location is represented by the function $\bar{f}(x, y)$. When the number of photons collected is large, the Poisson distribution can be converted to a Gaussian distribution. In this appendix, the mathematical derivative is based on the Poisson distribution.

The photon detection process can also be modeled as a Poisson distribution. The emitted or reflected photons from the object propagate through the medium and are

detected by the imaging system. In the simplest form, the detection of an object can be modeled based on the assumption that the detections of photons in the CCD array are statistically independent. Thus,

$$p(g|f) = \prod_{x,y} \frac{[(f * h)(x, y)]^{g(x,y)} e^{-(f * h)(x,y)}}{g(x, y)!} \quad (\text{B.5})$$

where the mean detection rate is defined by the object convolved with the specific PSF, $(f * h)(x, y)$.

B.3 The Maximum Likelihood Algorithm

To maximize the objective, the classic gradient approach is applied. The first derivative of the object is zero at a maximum as stated below:

$$\frac{\partial \ln(J_{MAP}(f, g))}{\partial f} \Big|_{x,y} = 0, \forall x, y \quad (\text{B.6})$$

$$\frac{\partial \ln(p(g|f))}{\partial f} \Big|_{x,y} = 0, \forall x, y \quad (\text{B.7})$$

The first derivate of $\ln(g!)$ is complicated. Therefore, the Sterling's approximation is employed to simplify the factorial term in Eq.(B.5). Sterling's approximation is:

$$\ln(g!) \approx g \times \ln(g) - g \quad (\text{B.8})$$

Substitute the properties of the natural logarithm with Sterling's approximation.

Equation (B.7) becomes

$$\begin{aligned} \frac{\partial \ln(p(g|f))}{\partial f} \Big|_{x,y} &= \frac{\partial}{\partial f(x,y)} \left\{ \sum_{x',y'} g(x',y') \ln[(f * h)(x',y')] \right. \\ &\quad \left. - (f * h)(x',y') - \ln[g(x',y')!] \right\} \Big|_{x'=x,y'=y} \end{aligned} \quad (\text{B.9})$$

$$\begin{aligned} \frac{\partial \ln(p(g|f))}{\partial f} \Big|_{x,y} &= \sum_{x',y'} g(x',y') \\ &\quad \times \frac{\partial}{\partial f(x,y)} \ln \left[\sum_{x'',y''} f(x'',y'') h(x' - x'', y' - y'') \right] \\ &\quad - \frac{\partial}{\partial f(x,y)} \sum_{x'',y''} f(x'',y'') h(x' - x'', y' - y'') \Big|_{x'=x,y'=y} \\ &\quad \sum_{x',y'} g(x',y') \frac{h(x' - x, y' - y)}{(f * h)(x',y')} - h(x' - x, y' - y) \Big|_{x'=x,y'=y} \\ &= \sum_{x,y} \left(\frac{g(x,y)}{(f * h)(x,y)} - 1 \right) * h(-x, -y) \end{aligned} \quad (\text{B.10})$$

Substituting Eq.(B.10) into the objective Eq.(B.7), one finds

$$\left[\frac{g(x,y)}{(f * h)(x,y)} - 1 \right] * h(-x, -y) = 0, \forall x, y \quad (\text{B.11})$$

The left hand side of Eq.(B.11) can be rewritten as

$$\left[\frac{g(x,y)}{(f * h)(x,y)} \right] * h(-x, -y) = 1 * h^+(x, y), \forall x, y \quad (\text{B.12})$$

An appropriate object $f(x, y)$ can be found satisfying Eq.(B.12) on a pixel by pixel basis.

It is difficult to solve Eq.(B.12). However, a *functional iteration* can be applied to solve such an optimization problem. The $(n+1)$ -th and n -th step of the process satisfy

$$\hat{f}^{n+1}(x, y) = F[\hat{f}^n(x, y)] \quad (\text{B.13})$$

The function $F[\cdot]$ is the estimate update function providing a *contractive mapping*. The *contractive mapping* will converge to a fixed point in the domain of feasible solutions. By applying a Picard derivative, Eq.(B.13) can be rewritten as

$$\hat{f}^{n+1} \times (1 * h(-x, -y)) = \hat{f}^n \left\{ \left[\frac{g(x, y)}{(\hat{f}^n * h)(x, y)} \right] * h(-x, -y) \right\}, \forall x, y \quad (\text{B.14})$$

The convolution $1 * h(-x, -y)$ can be simplified

$$\begin{aligned} FT[1 * h(-x, -y)] &= \delta(x, y) \times H(-x, -y) \\ &= H(0, 0) \times \delta(-x, -y) \end{aligned} \quad (\text{B.15})$$

where FT denotes the Fourier transform. We know $\delta(-x, -y) = \delta(x, y)$. Therefore,

$$FT^{-1}[H(0, 0) \times \delta(x, y)] = H(0, 0) \times FT^{-1}[\delta(x, y)] = H(0, 0) = 1 \quad (\text{B.16})$$

where $H(0, 0) = 1$ and FT^{-1} denotes the inverse Fourier transform. We thus arrive at the ML estimate update equation:

$$\hat{f}^{n+1} = \hat{f}^n \left\{ \left[\frac{g(x, y)}{(\hat{f}^n * h)(x, y)} \right] * h(-x, -y) \right\}, \forall x, y \quad (\text{B.17})$$

A clearer formulation is:

$$\hat{f}^{n+1} = \hat{f}^n \left\{ \left[\frac{g(x, y)}{(\hat{f}^n * h)(x, y)} \right] \odot h(x, y) \right\}, \forall x, y \quad (\text{B.18})$$

where the symbol \odot represents the correlation operator.

Appendix C

SOURCE CODE

C.1 Disclaimer

The source code published in this appendix is for reference only. Due to the complexity of the algorithms, the amount of open source code used, and the time limit of completion of this dissertation, the author cannot guarantee the completeness of the code published in this appendix. It is possible that some non-trivial components of the codes may be missing. We have tried to be as complete as possible to have all the code segments we use in this dissertation published in this appendix. The flowcharts for the code set are also presented in previous chapters in case a reader wants to recreate the program even if code fragments are missing by mistake.

C.2 Matlab Source Code for a Markov Random Field Based Channel

Decoder

The code for the proposed Markov random field based channel decoder is algorithm:

```
function y = nicm(a, o, Tol, n)
```

```
% This is the proposed algorithm for noise removal
```

```
% a is the image.
% Tol is a predetermined outliner threshold
% n is the number of iterations
% o is the original image for comparison

count = 0; % This is for counting the number of iteration.
m1 = mse2(o, a);
m = [count m1];

for i = 1:+1:n
    a1 = nmicm2(a, Tol);

        % This prevent any zero value pixel in the image.
    if min(min(a1)) <= 0
        a1 = thre2l(a1, 0);
    end

        count = count + 1;
        m2 = mse2(o, a1);

        m = [m; count m2];

        % Search for the optimal results
```

```

[sy, sx] = size(m);
if m(sy, sx) > m(sy-1, sx) & m(sy-2, sx) > m(sy-1, sx)

    sm = m(sy-1, sx);
    sc = m(sy-1, 1);
    save goodnicm2result sc sm a1;

end

% This save the result of the last iteration
a = a1;
save nicm2record m a1;

end

y = a1;

```

The following is a modulus of the proposed Markov random field channel decoder that performs the actual calculation.

```

function y = nmicm2(a, Tol)

% This is a modulus of the proposed algorithm that
% perform calculation
% a is the image.
% Tol is a predetermined outlier threshold

```

```
[sy, sx] = size(a);

mu = 0;
di = 0;

re = [0 0 0];
a1 = a;
% This for-loop is for outlier rejection, it finds outlier
for i = 2:sy-1
for j = 2:sx-1
mu = (a(i+1, j) + a(i-1, j) + a(i, j+1) + a(i, j-1))/4;
di = abs(a(i,j) - mu);
if di > Tol;
re = [re; i j 0];
a1(i, j) = mu;
end
end
end

[ry, rx] = size(re);
re = re(2:ry, :);
```

```

% This creates a matrix for image update
b = ones(sy, sx);

% This calculates the probability of a pixel in MRF
for i = 3:sy-2
for j = 3:sx-2
b(i,j) = svc(a1, re, i, j);
end
end

y = b.*a1;

```

The following program calculates the conditional probability and performs a check of the energy function.

```

function y = svc(a, re, i, j)
% This calculates conditional probability and perform the checking
% of the energy function function
% a is the image, re is the outliner pixes coordinate, i is the
% y coordinate of desired pixel, j is the x-coordinate of desired
% pixel
% clique? is the clique without the center
% vc? is the potential function

% re must be modified before going to vc calculation

```

```
[jy, jx] = size(re);  
ol = 0;  
% This operation prevent considering the center as a outlier.  
% Without this operation, all the outlier will remain the same.  
for k = 1:jy  
    if re(k, 1) == i & re(k, 2) == j  
        re(k, 3) = 1;  
    end  
end  
  
[jy, jx] = size(re);  
  
c1 = clique1(i, j);  
vc1 = vc(a, re, c1, i, j);  
  
c2 = clique2(i, j);  
vc2 = vc(a, re, c2, i, j);  
  
c3 = clique3(i, j);  
vc3 = vc(a, re, c3, i, j);
```

```
c4 = clique4(i, j);  
vc4 = vc(a, re, c4, i, j);  
  
c5 = clique5(i, j);  
vc5 = vc(a, re, c5, i, j);  
  
c6 = clique6(i, j);  
vc6 = vc(a, re, c6, i, j);  
  
c7 = clique7(i, j);  
vc7 = vc(a, re, c7, i, j);  
  
c8 = clique8(i, j);  
vc8 = vc(a, re, c8, i, j);  
  
c9 = clique9(i, j);  
vc9 = vc(a, re, c9, i, j);  
  
c10 = clique10(i, j);  
vc10 = vc(a, re, c10, i, j);  
  
c11 = clique11(i, j);  
vc11 = vc(a, re, c11, i, j);
```

c12 = clique12(i, j);
vc12 = vc(a, re, c12, i, j);

c13 = clique13(i, j);
vc13 = vc(a, re, c13, i, j);

c14 = clique14(i, j);
vc14 = vc(a, re, c14, i, j);

c15 = clique15(i, j);
vc15 = vc(a, re, c15, i, j);

c16 = clique16(i, j);
vc16 = vc(a, re, c16, i, j);

c17 = clique17(i, j);
vc17 = vc(a, re, c17, i, j);

c18 = clique18(i, j);
vc18 = vc(a, re, c18, i, j);

c19 = clique19(i, j);

```
vc19 = vc(a, re, c19, i, j);
```

```
c20 = clique20(i, j);
```

```
vc20 = vc(a, re, c20, i, j);
```

```
c21 = clique21(i, j);
```

```
vc21 = vc(a, re, c21, i, j);
```

```
c22 = clique22(i, j);
```

```
vc22 = vc(a, re, c22, i, j);
```

```
c23 = clique23(i, j);
```

```
vc23 = vc(a, re, c23, i, j);
```

```
c24 = clique24(i, j);
```

```
vc24 = vc(a, re, c24, i, j);
```

```
v = [vc1;vc2;vc3;vc4;vc5;vc6;vc7;vc8;vc9;vc10;vc11;vc12;vc13;vc14;
```

```
vc15;vc16;vc17;vc18;vc19;vc20;vc21;vc22;vc23;vc24];
```

```
% This is for putting the pixel back as outliner.
```

```
if ol ~= 0
```

```
re(ol, 3) = 0;
```

```
end

xcs = [0 0];

[sy, sx] = size(v);
nq = 0;

for k = 1:sy
    if v(k, 1) == 0
        nq = nq+1;
        xcs = [xcs; v(k, 2:3)];
    end
end

% c is a constant to constraint the value
c = 100;

[ry, rx] = size(xcs);
xcs = xcs(2:ry, :);
[ry, rx] = size(xcs);
if ry > 0
    sv = sum(xcs(:, 1));
    % ds is a derivation of  $\bar{x}$  s,c in eq. 4 of Park
```

```
ds = std(xcs(:, 2));
t = ds*nq*c;
if t < 0.005 % assume the difference is too smaller in the clique
y = 1;
else
if sv > 0
x = (-1/t)*sv;
y = exp(x);
else
% perform the inversion when the energy function < 0
x = (-1/t)*abs(sv);
y = 1/exp(x);
end
end
else
y = 1;
end
```

This is the relative potential function

```
function y = vc(a, re, c, i, j)
```

```
% This is a relative potential function program, a is the image,
```

```
% re is the outliner
```

```
% pixel coordinate, i is the y coordinate of desired pixel,
```

```
% j is the x-coordinate of desired pixel,  
% c are in the clique without the center  
  
[sy, sx] = size(re);  
% t are coordinates of sites in the clique with the center  
t = [i, j; c];  
  
[ry, rx] = size(t);  
  
% f is the flag  
f = 0;  
  
for k = 1:ry  
for l = 1:sy  
if t(k, 1) == re(l, 1) & t(k, 2) == re(l, 2) & re(l, 3) == 0  
f = 1;  
end  
end  
end  
  
st = 0;
```

```

s = 0;
v = 0;

%s is the sum of pixel value in a clique
if f == 0
for m = 1:ry
s = s + a(t(m, 1), t(m, 2));
end

% st is the mean of pixel value in the clique
st = s/ry;
v = a(i,j) - st;
end

% f is flag, v is relative potential function (VC),
% st is mean of pixels in clique
y = [f v st];

```

The following is the program used to measure the mean-square-error.

```

function y = mse2(a, b)

% This is the program to measure the mean-square-error
% between two images

% a is the image one
% b is the image two

```

```
s = size(a);
s1 = s(1, 1);
s2 = s(1, 2);

c = a - b;
d = c .* c;

e = sum(d);
f = sum(e');
g = f/(s1*s2);

y = g;
```

This is the maximum likelihood algorithm.

```
function y = ml2(g, f, o, n)
```

```
% This is a Maximum Likelihood Superresolution Algorithm for
% 2-D image
% g is the input image
% n is the number of iterations
% f is the object estimate obtained by ML algorithm
% o is the OTF to be used for object estimation
```

```
for i = 1:1:n
a1 = fft2(f).*o;
a2 = real(ifft2(a1));
a3 = 1 ./ a2;
a4 = g.*a3;
a5 = fft2(a4) .* o;
a6 = real(ifft2(a5));
a7 = f .* a6;
f = thre2l(a7, 0);
end;
y = f;
```

This function thresholds an image.

```
function y = thre2l(a, th2)

% This program thresholds an image

% The size of the input signal
% s1 is row.
s = size(a);
s1 = s(1, 1);

% s2 is col
s2 = s(1, 2);
b = real(a);
```

```
for i = 1:s1
for j = 1:s2
if b(i, j) <= th2
b(i, j) = 0;
else
b(i, j) = b(i,j);
end
end
end

y = b;
```

C.3 Matlab Source Code for Object-Boundary-Constrained Maximum Likelihood Algorithm

```
function y = obcml2(g, f, t, o, n)

% This is a Object-boundary-Constrained Maximum Likelihood
% Superresolution Algorithm for 2-D image
% g is the input image
% n is the number of iterations
% f is the object estimate obtained by ML algorithm
% o is the OTF to be used for object estimation
```

```
% t is the mask

for i = 1:+1:n
f = f.*t;
a1 = fft2(f).*o;
a2 = ifft2(a1);
a3 = 1 ./ a2;
a4 = g.*a3;
a5 = fft2(a4) .* o;
a6 = abs(ifft2(a5));
a7 = f .* a6;
f = thre2l(a7, 0);
end;
y = f;
```

C.4 C++ Source Code for Super-resolution Algorithm Based on a Self-Organizing Map

To simplify the programming effort, as many public domain source codes as possible were used in this dissertation. The `tmatrix.h`, `tvector.h`, `tmatrix.cpp`, and `tvector.cpp` code snippets were from the second edition of *A Computer Science Tapestry: Exploring Computer Science and Programming with C++* [75].

The following code fragments represent the SOM weight training program.

```
#include <iostream>
#include <fstream>
#include <sstream>
#include <string>
#include <math.h>

using namespace std;

// Library from "A Computer Science Tapestry" by Owen L. Astrachan
#include "tmatrix.h"
#include "tvector.h"

// This program must be compiled with g++ compiler, not gcc
// This program is a weight training program for my SOM based image
// superresolution algorithm.
// The feature extraction is modified the one to find the lowest
// pixel value for the data training.
// We have to start running this program from the beginning
// I name this algorithm SOM1 for image superresolution training
// Henry Pang 07-13-2001

// Prototype for generating input file name with number
void name_num(string & name, int number, string head);
```

```
// Prototype for reading the image file header
void input_info(istream & filename, int & cols, int & rows,
int & max);

// Prototype for reading the image file data
void input_data(istream & filename, tmatrix<int> & array,
int cols, int rows, int max);

// Prototype for reading weight header
void weight_info(istream & filename, int & s_cols, int & s_rows,
int & cols, int & rows);

// Prototype for reading weight data
void weight_data(istream & filename, tmatrix<double> & array,
int cols, int rows);

// Prototype for storing the number of iteration
void weight_update_num(ofstream & filename, int it_no);

// Extract 3X3 block feature from the image
void extract_feature(tmatrix<int> array_1, tmatrix<int> array_2,
tvector<int> & feature, int f_x, int f_y);
```

```
// Prototype for writing weight data file
void output_weight(ofstream & filename, tmatrix<double> array,
int s_cols, int s_rows, int cols, int rows, int it_no,
int neighbor_width);

// Find the neuron which has the minimum Euclidean distance between
// neuron in the weight matrix and the input vector
int winning_neuron(tvector<int> feature, tmatrix<double>
weight_array, int w_rows, int w_cols, int f_dim);

// This is a function for weight update
void weight_update(int neuron_num, tvector<int> feature,
tmatrix<double> & weight_array, int w_rows, int s_rows,
int s_cols, int f_dim, int n, int & neighbor_width);

int main() {

    int b_rows, b_cols; //blur image dimension
    int b_max; // blur image max value
    int c_rows, c_cols; //clear image dimension
    int c_max; // clear image max value
```

```
int r_rows, r_cols; //random pixel data dimension
int r_max; // random pixel max value which is not correct

int s_rows, s_cols; //SOM dimension
int w_rows, w_cols; //weight dimension
int t_row = 0;
int f_dim = 10; // feature dimension

// This part need human intervention
int neuron_num = 0, neighbor_width = 46;
int begin_iteration = 1; // The beginning iteration number
int max_iteraton = 100; // maximum iteration number
int begin_image_num = 11, // beginning image file number
int max_image_num = 74; // end image file numbers

double result;

int i; // iteration number
int j; // image number
int k; // row number
int l; // column number

// This makes the names of blur and clear images
string b_head = "clr";
```

```
string b_name;

string c_head = "brn";

string c_name;

// This starts the weight matrix input

ifstream input_weight("initial_weight_amp_20.som", ios::in);

// Open an weight file and store its header information to the
// computer

weight_info(input_weight, s_rows, s_cols, w_cols, w_rows);

// Initialize the weight matrix

tmatrix<double> weight_array(w_rows, w_cols);

// Input the weight data to the matrix

weight_data(input_weight, weight_array, w_cols, w_rows);

// Initialize the feature vector

tvector<int> feature(f_dim);

// This start the iteration

for (i=begin_iteration; i<max_iteraton+1; i++){

    for (j=begin_image_num; j<max_image_num+1; j++){

        // This generates the file names for input

        name_num(b_name, j, b_head);
```

```
name_num(c_name, j, c_head);

const char* b_image = b_name.data();
const char* c_image = c_name.data();
ifstream blur_image(b_image, ios::in);
ifstream clear_image(c_image, ios::in);

// This inputs the blur image
// Open the blur image file and store its header information
input_info(blur_image, b_cols, b_rows, b_max);
// Initialize the blur image matrix
tmatrix<int> blur_array(b_rows, b_cols);
// Input the blur image data to the matrix
input_data(blur_image, blur_array, b_cols, b_rows, b_max);

// This input the clear image
// Open the clear image file and store its header information
input_info(clear_image, c_cols, c_rows, c_max);
// Initialize the matrix
tmatrix<int> clear_array(c_rows, c_cols);
// Input the image data to the matrix
input_data(clear_image, clear_array, c_cols, c_rows, c_max);

// This starts the training of a given picture
```

```
        for (k = 240; k<272; k++){
for (l = 240; l<272; l++){

// Extract feature from the blur image and clear image
extract_feature(blur_array, clear_array, feature, l, k);

// This program finds the winning neuron
neuron_num = winning_neuron(feature, weight_array, w_rows,
w_cols, f_dim);

// This program update the SOM weight
weight_update(neuron_num, feature, weight_array, w_rows, s_rows,
s_cols, f_dim, i, neighbor_width);

}

}

// Write the iteration number
ofstream weight_num("num_iteration_1_t1.log", ios::out);
weight_update_num(weight_num, i);

// Write the weight to the weight.dat file for testing
ofstream final_weight("weight_1_t1.som", ios::out);
output_weight(final_weight, weight_array, s_cols, s_rows,
w_cols, w_rows, i, neighbor_width);
}
```

```
    return 0;
}

void name_num(string & name, int number, string head)
{
    int bufferSize = 3;
    char buf[bufferSize];
    ostream output(buf, bufferSize);
    string local;
    string ending = ".pgm";

    output << number << ends;
    local = output.str();
    name = head + local + ending;
}

void weight_update(int neuron_num, tvector<int> feature,
tmatrix<double> & weight_array, int w_rows, int s_rows,
int s_cols, int f_dim, int n, int & neighbor_width)
{
    int i, j, k;
    int neuron_num_x, neuron_num_y;
```

```
int n_n_up, n_n_down, n_n_left, n_n_right;

double dist;

int sigma_o = 5, eta_o = 1;

double sigma, pi, eta;

int tau_1= 100;

int tau_2 = 100;

neuron_num_y = neuron_num / s_rows;

neuron_num_x = neuron_num % s_rows;

n_n_up = neuron_num_y - neighbor_width;

n_n_down = neuron_num_y + neighbor_width;

n_n_left = neuron_num_x - neighbor_width;

n_n_right = neuron_num_x + neighbor_width;

if (n_n_up < 0){

    n_n_up = 0;

}

if (n_n_down > s_rows - 1){

    n_n_down = s_rows - 1;

}

if (n_n_left < 0){

    n_n_left = 0;
```

```
}

if (n_n_right > s_cols - 1){
    n_n_right = s_cols - 1;
}

sigma = sigma_o * exp(-n/tau_1);

eta = eta_o * exp(-n/tau_2);

// This calculate the distance between the winning neuron and the
// neighbor in 2-D SOM
// and the pi value
// It also update the SOM weight
for (i=n_n_up; i<n_n_down+1; i++){
    for (j=n_n_left; j<n_n_right+1; j++){
        dist = sqrt((j-neuron_num_x)*(j-neuron_num_x) +
            (i-neuron_num_y)*(i-neuron_num_y));
        pi = exp(-(dist*dist)/(2*sigma*sigma));
        // cout << pi << " ";

        // update the weight
        for (k=0; k<f_dim; k++){
            weight_array[i*s_cols+j][k] = weight_array[i*s_cols+j][k] +
```

```
eta * pi* (double(feature[k]) -
weight_array[i*s_cols+j][k]);
    }

    // This function is used to recalculate the neighborhood width
    // to save computational time, the new width only affects the
    // next iterations
    if (pi < 0.001) {
if (dist < neighbor_width) {
    neighbor_width = (int)dist;
    // cout << "new width is " << neighbor_width << endl;
}
    }
}
// cout << endl;
}
}

void input_info(ifstream & filename, int & cols, int & rows,
int & max)
{
    char dummy[2];
    // char *dummy = new char[2];
```

```
// The section open a clear file image and store it in the
// memory
filename.read(&dummy,2);
if(strncmp("P5",dummy, 2) != 0) {
    cerr << "\nCan't read magic marker P5";
    exit(0);
}
//filename.read(&dummy,1);
filename >> cols;
filename >> rows;
filename >> max;
filename.read(&dummy,1);
}

void input_data(istream & filename, tmatrix<int> & array,
int cols, int rows, int max)
{
    int i, j, data;
    unsigned char buffer;
    // Read the image into the matrix, the data in the file is
    // in binary
    long dummy = filename.tellg();
    for (i=0; i<rows; i++) {
```

```
    for (j=0; j<cols; j++) {
        if(filename.read((char *)&buffer, 1)) {
            array[i][j] = (int)buffer;
        } else {
cerr << "\nFile shorter than expected.";
exit(0);
        }
    }
}

dummy = filename.tellg();
filename.read((char *)&buffer,1);
if(filename.gcount() != 0) {
    cerr << "\nFile longer than expected.";
    exit(0);
}

filename.close();
}

void weight_info(ifstream & filename, int & s_cols, int & s_rows,
int & cols, int & rows)
{
    // The section open a clear file image and store it in the memory
    filename >> s_cols; // s_cols is the som column number
```

```
filename >> s_rows; // s_rows is the som row number
filename >> cols;
filename >> rows;
}

void weight_data(ifstream & filename, tmatrix<double> & array,
int cols, int rows)
{
    int i, j;
    double data;
    // Read the image into the matrix
    for (i=0; i<rows; i++) {
        for (j=0; j<cols; j++) {
            filename >> data;
            array[i][j] = (double)data;
        }
    }
    filename.close();
}

void weight_update_num(ofstream & filename, int it_no)
{
    filename << "Number of iteration = " << it_no << endl;
```

```
    filename.close();
}

void output_weight(ofstream & filename, tmatrix<double> array,
int s_cols, int s_rows, int cols, int rows, int it_no,
int neighbor_width)
{
    int i, j;
    filename << s_cols << " ";
    filename << s_rows << endl;
    filename << cols << " ";
    filename << rows << endl;
    filename << it_no << " ";
    filename << neighbor_width << endl;
    for (i=0; i<rows; i++) {
        for (j=0; j<cols; j++) {
            filename << array[i][j] << " ";
        }
        filename << endl;
    }
    filename.close();
}
```

```
void extract_feature(tmatrix<int> array_1, tmatrix<int> array_2,
tvector<int> & feature, int f_x, int f_y)
{
    int i, j;
    int min = 100000;

    feature[0] = array_1[f_y - 1][f_x - 1];
    feature[1] = array_1[f_y - 1][f_x];
    feature[2] = array_1[f_y - 1][f_x + 1];
    feature[3] = array_1[f_y][f_x - 1];
    feature[4] = array_1[f_y][f_x + 1];
    feature[5] = array_1[f_y + 1][f_x - 1];
    feature[6] = array_1[f_y + 1][f_x];
    feature[7] = array_1[f_y + 1][f_x + 1];

    feature[8] = array_1[f_y][f_x];

    feature[9] = array_2[f_y][f_x];

    for (i=0; i<8; i++) {
        if (feature[i] < min) {
            min = feature[i];
        }
    }
}
```

```
    }

    for (j=0; j<10; j++){
        feature[j] = feature[j] - min;
    }

}

int winning_neuron(tvector<int> feature,
tmatrix<double> weight_array, int w_rows,
int w_cols, int f_dim)
{
    int i, j, o = -1;
    double n = 99999999;
    for (i = 0; i<w_rows; i++){
        double k, l = 0, m;
        for (j = 0; j<f_dim; j++){
            k = double(feature[j]) - weight_array[i][j];
            l = l + k*k;
        }
        m = sqrt(l);
        if (m < n){
            n = m;
        }
    }
}
```

```
        o = i;
    }
}
return o;
}
```

The following represents the superresolution algorithm based on the self-organizing map that was developed in this dissertation.

```
#include <iostream>
#include <fstream>
#include <sstream>
#include <string>
#include <math.h>

using namespace std;

// Library from "A Computer Science Tapestry" by Owen L. Astrachan
#include "tmatrix.h"
#include "tvector.h"

// This program is used to super-resolve blur image with the
// trained SOM data. The training algorithm is som1.cpp
// This program must be compiled with g++ compiler, not gcc
// Henry Pang 06-07-01
```

```
// Prototype for reading the image file header
void input_info(istream & filename, int & cols, int & rows,
int & max);

// Prototype for reading the image file data
void input_data(istream & filename, tmatrix<int> & array,
int cols, int rows, int max);

// Prototype for reading weight header
void trained_weight_info(istream & filename, int & s_cols,
int & s_rows, int & cols, int & rows, int & num, int & neighbor);

// Prototype for reading weight data
void weight_data(istream & filename, tmatrix<double> & array,
int cols, int rows);

// Prototype for writing image file
void output(ofstream & filename, tmatrix<int> array, int cols,
int rows, int max);

// Extract 3X3 block feature from the image
void extract_blur_feature(tmatrix<int> array_1,
```

```
tvector<int> & feature, int f_x, int f_y);

// Find the neuron which has the minimum Euclidean distance
// between neuron in the weight matrix and the input vector
double winning_neuron_value(tvector<int> feature,
tmatrix<double> weight_array, int w_rows, int w_cols, int f_dim);

int main() {

    int b_rows, b_cols; //blur image dimension
    int b_max; // blur image max value

    int c_rows, c_cols; //clear image dimension
    int c_max; // clear image max value

    int s_rows, s_cols; //SOM dimension
    int w_rows, w_cols; //weight dimension
    int num_iteration; // number of iteration
    int neighborhood; // size of the neighborhood

    int f_dim = 9; // feature dimension

    double neuron_value;
```

```
int i; // iteration number
int j; // image number
int k; // row number
int l; // column number

// This starts the weight matrix input
ifstream trained_weight("weight_1_t1.som", ios::in);
// Open an weight file and store its header information to
// the computer
trained_weight_info(trained_weight, s_rows, s_cols, w_cols,
w_rows, num_iteration, neighborhood);
// Initialize the weight matrix
tmatrix<double> weight_array(w_rows, w_cols);
// Input the weight data to the matrix
weight_data(trained_weight, weight_array, w_cols, w_rows);

// Initialize the feature vector
tvector<int> feature(f_dim);

// This input the blur image
ifstream blur_image("brn11.pgm", ios::in);
input_info(blur_image, b_cols, b_rows, b_max);
// cout << blur_image << " " << b_cols << " " << b_rows <<
```

```
" " << b_max << endl;

// Initialize the blur image matrix
tmatrix<int> blur_array(b_rows, b_cols);

// Input the blur image data to the matrix
input_data(blur_image, blur_array, b_cols, b_rows, b_max);

// This input the clear image
ifstream clear_image("brn11.pgm", ios::in);

// Open the clear image file and store its header information
input_info(clear_image, c_cols, c_rows, c_max);

// Initialize the matrix
tmatrix<int> clear_array(c_rows, c_cols);

// Input the image data to the matrix
input_data(clear_image, clear_array, c_cols, c_rows, c_max);

// This starts the restoration of a blur image
for (k = 1; k<255; k++){
    for (l = 1; l<255; l++){

        // Extract feature from the blur image and clear image
        extract_blur_feature(blur_array, feature, l, k);

        // This part begin the table look up from the SOM weight
```

```
// This program finds the winning neuron value
neuron_value = winning_neuron_value(feature, weight_array,
w_rows, w_cols, f_dim);

// This program update the super-resolved image pixel value
clear_array[i][j] = (int)(neuron_value+0.5);

}

}

// Write the restored image to a pgm file
ofstream restored_image("restored_som1_11.pgm", ios::out);
output(restored_image, clear_array, c_cols, c_rows, c_max);

return 0;
}

void input_info(ifstream & filename, int & cols, int & rows,
int & max)
{
char dummy[2];

// char *dummy = new char[2];

// The section open a clear file image and store it in
// the memory
```

```
filename.read(&dummy,2);
if(strncmp("P5",dummy, 2) != 0) {
    cerr << "\nCan't read magic marker P5";
    exit(0);
}

//filename.read(&dummy,1);

filename >> cols;
filename >> rows;
filename >> max;

filename.read(&dummy,1);
}

void input_data(istream & filename, tmatrix<int> & array,
int cols, int rows, int max)
{
    int i, j, data;
    unsigned char buffer;

    // Read the image into the matrix, the data in the file
    // is in binary

    long dummy = filename.tellg();
    for (i=0; i<rows; i++) {
        for (j=0; j<cols; j++) {
            if(filename.read((char *)&buffer, 1)) {
```

```
        array[i][j] = (int)buffer;
    } else {
cerr << "\nFile shorter than expected.";
exit(0);
    }
}
}

dummy = filename.tellg();
filename.read((char *)&buffer,1);
if(filename.gcount() != 0) {
    cerr << "\nFile longer than expected.";
    exit(0);
}
filename.close();
}

void trained_weight_info(ifstream & filename, int & s_cols,
int & s_rows, int & cols, int & rows, int & num, int & neighbor)
{
    // The section open a clear file image and store it in the
    // memory
    filename >> s_cols; // s_cols is the som column number
    filename >> s_rows; // s_rows is the som row number
```

```
filename >> cols;
filename >> rows;
filename >> num;
filename >> neighbor;
}

void weight_data(ifstream & filename, tmatrix<double> & array,
int cols, int rows)
{
int i, j;
double data;
// Read the image into the matrix
for (i=0; i<rows; i++) {
for (j=0; j<cols; j++) {
filename >> data;
array[i][j] = (double)data;
}
}
filename.close();
}

void extract_blur_feature(tmatrix<int> array_1,
tvector<int> & feature, int f_x, int f_y)
```

```
{  
    int i, j;  
    int min = 100000;  
  
    feature[0] = array_1[f_y - 1][f_x - 1];  
    feature[1] = array_1[f_y - 1][f_x];  
    feature[2] = array_1[f_y - 1][f_x + 1];  
    feature[3] = array_1[f_y][f_x - 1];  
    feature[4] = array_1[f_y][f_x + 1];  
    feature[5] = array_1[f_y + 1][f_x - 1];  
    feature[6] = array_1[f_y + 1][f_x];  
    feature[7] = array_1[f_y + 1][f_x + 1];  
  
    feature[8] = array_1[f_y][f_x];  
  
}  
  
double winning_neuron_value(tvector<int> feature,  
tmatrix<double> weight_array, int w_rows, int w_cols, int f_dim)  
{  
    int i, j, k, l, o = -1;  
    int min = 100000;
```

```
for (k=0; k<8; i++) {
    if (feature[k] < min) {
        min = feature[k];
    }
}

for (l=0; l<f_dim; l++){
    feature[l] = feature[l] - min;
}

double n = 99999999, p;
for (i = 0; i<w_rows; i++){
    double k, l = 0, m;
    for (j = 0; j<f_dim; j++){
        k = feature[j] - weight_array[i][j];
        l = l + k*k;
    }
    m = sqrt(l);
    if (m < n){
        n = m;
        o = i;
    }
}
```

```
}  
  
p = weight_array[o][9] + min;  
  
// cout << "Min E-D is " << n << endl;  
  
return p;  
  
}
```

```
void output(ofstream & filename, tmatrix<int> array, int cols,  
int rows, int max)  
{  
  
    int i, j;  
  
    unsigned char dummy;  
  
    filename << "P5\n";  
  
    filename << cols << " ";  
  
    filename << rows << endl;  
  
    filename << max << endl;  
  
    for (i=0; i<rows; i++) {  
        for (j=0; j<cols; j++) {  
            if(array[i][j] > 255 ) {  
                dummy = 255;  
  
                filename.write((char *)&dummy,1);  
            } else if(array[i][j] < 0) {  
                dummy = 0;  
  
                filename.write((char *)&dummy,1);  
            }  
        }  
    }  
}
```

```
    } else {  
dummy = (unsigned char)array[i][j];  
    filename.write((char *)&dummy,1);  
    }  
    }  
    }  
filename.close();  
}
```

REFERENCES

- [1] H. M. Adorf, "Hubble space telescope image restoration in its fourth year," *Inverse Problems*, vol. 11, no. 4, pp. 639–653, August 1995.
- [2] J. A. O'Sullivan, R. E. Blahut, and D. L. Snyder, "Information-theoretic image processing," *IEEE Transactions on Information Theory*, vol. 44, pp. 2094–2123, 1998.
- [3] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley Series in Telecommunications. John Wiley and Sons, Inc., 1991.
- [4] L. O. Chua, *CNN: A Paradigm for Complexity*, World Scientific Series on Nonlinear Science. World Scientific, 1998.
- [5] W. J. Smith, *Modern Optical Engineering: The Design of Optical Systems*, Optical and Electro-Optical Engineering Series. McGraw-Hill, 1990.
- [6] R. Kingslake, *Optical System Design*, Graduate Texts in Mathematics. Academic Press, 1983.
- [7] J. W. Goodman, *Introduction to Fourier Optics*, Electronic Engineering Series. McGraw-Hill, 1968.
- [8] J. D. Gaskill, *Linear Systems, Fourier Transforms, and Optics*, Wiley Series in Pure and Applied Optics. John Wiley and Sons, Inc., 1978.

- [9] P. C. Sabatier, "Past and future of inverse problems," *Journal of Mathematical Physics*, vol. 41, no. 6, pp. 4082–4124, 2000.
- [10] V. A. Morozov, *Regularization Methods for Ill-posed Problems*, CRC Press, 1993.
- [11] R. J. Watt, *Visual Processing: Computational, Psychophysical, and Cognitive Research*, Lawrence Erlbaum Associates, 1988.
- [12] B. R. Hunt and P. Sementilli, "Description of a poisson image super-resolution algorithm," *Astronomical Data Analysis Software and Systems I*, vol. 25, 1992.
- [13] S. J. Reeves, "Generalized cross-validation as a stopping rule for the richardson-lucy algorithm," *International Journal of Imaging Systems and Technology*, vol. 6, no. 4, pp. 387–391, 1995.
- [14] W. H. Richardson, "Bayesian-based iterative method of image restoration," *Journal of Optical Society of America, A*, vol. 62, pp. 55–60, 1972.
- [15] L. B. Lucy, "An iterative technique for the rectification of observed distributions," *Astron. J.*, vol. 79, no. 6, pp. 745–759, June 1974.
- [16] M. Z. Tarasko, "Preprint fei-156 (in russian)," *Obninsk*, 1968.
- [17] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, Electrical and Electronic Engineering Series. McGraw-Hill, 1991.

- [18] L. A. Shepp and Y. Vardi, "Maximum likelihood reconstruction for emission tomography," *IEEE Transactions on Medical Imaging*, vol. 1, pp. 113–122, 1982.
- [19] R. V. Churchill and J. W. Brown, *Complex Variables and Applications*, McGraw-Hill, 1990.
- [20] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech.*, vol. 44, pp. 379–423, 1948.
- [21] J. G. Proakis, *Digital Communications*, Electrical Engineering Series. McGraw-Hill, 1995.
- [22] F. Roddier, *Adaptive Optics in Astronomy*, Cambridge University Press, 1999.
- [23] M. C. Roggemann, *Imaging through Turbulence*, CRC Press Laser and Optical Science and Technology Series. CRC Press, 1996.
- [24] J. J. Green, *Multiframe Restoration Methods for Image Synthesis and Recovery*, Ph.D. dissertation, The University of Arizona, Tucson, AZ, 2000.
- [25] D. G. Sheppard, *Image Super-resolution: Iterative Multiframe Algorithms and Training of a Nonlinear Vector Quantizer*, Ph.D. dissertation, The University of Arizona, Tucson, AZ, 1997.
- [26] S. Parsons, *Qualitative Methods for Reasoning under Uncertainty*, The MIT Press, 2001.

- [27] M. S. Landy and J. A. Movshon, *Computational Models of Visual Processing*, MIT Press, 1991.
- [28] R. Spence, *Information Visualization*, ACM Press, 2001.
- [29] H. Derin and P. A. Kelly, "Discrete-index markov-type random processes," *Proceedings of the IEEE*, vol. 77, no. 10, pp. 1485–1510, October 1989.
- [30] R. C. Dubes and A. K. Jain, "Random field models in image analysis," *Journal of Applied Statistics*, vol. 16, no. 2, pp. 131–164, 1989.
- [31] B. R. Hunt, "Super-resolution of images: Algorithms, principles, performance," *International Journal of Imaging Systems and Technology*, vol. 6, pp. 297–304, 1995.
- [32] H. Y. Pang, M. K. Sundareshan, and S. V. Amphay, "Optimized maximum likelihood algorithms for super-resolution of passive millimeter-wave imagery," in *Proc. of SPIE Conf. on Passive Millimeter-wave Imaging Technology (Aerosense'98)*, Orlando, FL, April 1998.
- [33] S. Bhattacharjee and M. K. Sundareshan, "Modeling and extrapolation of prior scene information for set-theoretic restoration and super-resolution of diffraction-limited images," in *Proc. of the IEEE International Conf. On Image Processing (ICIP 2000)*, 2000.

- [34] M. K. Sundareshan and S. Bhattacharjee, "Super-resolution of sar images using bayesian and convex set-theoretic methods," in *Proc. of SPIE Conf. on Algorithms for Synthetic Aperture Radar Imagery , Aerosense'2000*, April 2000.
- [35] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall Information and System Sciences Series. Prentice Hall, 1989.
- [36] R. Chellappa and A. Jain, *Markov Random Fields: Theory and Application*, Academic Press, 1991.
- [37] L. B. Lucy, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721–741, November 1984.
- [38] J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *Journal of the Royal Statistical Society, Series B*, vol. 36, pp. 192–236, 1974.
- [39] J. Besag, "On the statistical analysis of dirty pictures," *Journal of the Royal Statistical Society, Series B*, vol. 48, no. 3, pp. 259–302, 1986.
- [40] J. Moussorius, "Gibbs and Markov random field systems with constraints," *Journal of Statistical Physics*, vol. 10, pp. 11–33, 1974.
- [41] J. Park and L. Kurz, "Image enhancement using the modified ICM method," *IEEE Transactions on Image Processing*, vol. 5, no. 5, pp. 765–771, 1996.

- [42] S. Kirkpatrick, C. D. Jr. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.
- [43] P. J. M. van Laarhoven and E. H. L. Aarts, *Simulated Annealing: Theory and Applications*, Kluwer Academic Publishers, 1988.
- [44] H. Y. Pang and M. K. Sundareshan, "A modified iterative conditional modes algorithm with relative potential function for pre-filtering in image super-resolution," submitted to the *ICIP-2001 International Conference on Image Processing*, Thessaloniki, Greece, October 2001.
- [45] J. M. Hammersley and P. Clifford, "Markov field on finite graphs and lattices," *Unpublished*, 1971.
- [46] P. L. Combettes, "Convex set theoretic image recovery: History, current status, and new directions," *Journal of Visual Communication and Image Presentation*, vol. 3, no. 4, pp. 307–315, December 1992.
- [47] P. J. Sementilli, B. R. Hunt, and M. S. Nadar, "Analysis of the limit to superresolution in incoherent imaging," *Journal of Optical Society of America, A*, vol. 10, no. 11, pp. 2265–2276, 1993.
- [48] P. J. Sementilli, M. S. Nadar, and B. R. Hunt, "Empirical evaluation of a bound on image superresolution performance," in *Proc. SPIE*, 1994, vol. 2302, pp. 178–187.

- [49] H. Stark and Y. Yang, *Vector Space Projections: A Numerical Approach to Signal and Image Processing, Neural Nets, and Optics*, Wiley Series in Telecommunications and Signal Processing. John Wiley and Sons, Inc., 1998.
- [50] A. Papoulis, "A new algorithm in spectral analysis and band-limited extrapolation," *IEEE Transactions on Circuits and Systems*, vol. 22, no. 9, pp. 735–742, 1975.
- [51] N. Wu, *The Maximum Entropy Method*, Springer series in information sciences. Springer, 1997.
- [52] T. Krishnan, *The EM Algorithm and Extensions*, Wiley Series in Probability and Statistics. Applied probability and Statistics. John Wiley and Son, Inc, 1997.
- [53] R. C. Puetter, "Pixion-based multiresolution image reconstruction and the quantification of picture information content," *International Journal of Imaging Systems and Technology*, vol. 6, pp. 314–331, 1995.
- [54] J. Von Neumann, *The Computer and the Brain.*, Mrs. Hepsa Ely Silliman memorial lectures. Yale Nota Bene, 2000.
- [55] P. S. Churchland, *The Computational Brain*, MIT Press, 1992.
- [56] P. S. Churchland, *The Engine of Reason, the Seat of the Soul: a Philosophical Journey into the Brain*, MIT Press, 1995.

- [57] D. E. Rumelhart, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1 of *Computational Models of Cognition and Perception*, MIT Press, 1986.
- [58] F. Faggin, "Vlsi implementation of neural networks," in *International Joint Conference on Neural Network*, 1990.
- [59] J. Satinover, *The Quantum Brain: The Search for Freedom and the Next Generation of Man*, John Wiley and Sons, 2001.
- [60] I. Aleksander and H. Morton, *An Introduction to Neural Computing*, Chapman and Hall, 1990.
- [61] P. J. Werbos, *The Roots of Backpropagation: from Ordered Derivatives to Neural Networks and Political Forecasting*, Adaptive and Learning Systems for Signal Processing, Communications, and Control. John Wiley and Son, 1994.
- [62] P. J. Werbos, *Beyond Regression: New Tools for Predictions and Analysis in the behavioral Sciences*, Ph.D. dissertation, Harvard University, Cambridge, MA, 1974.
- [63] S. Amari, "Theory of adaptive pattern classifiers," *IEEE Transactions on Electronic Computers*, vol. 16, pp. 299–307, June 1967.
- [64] G. Hinton and T. J. Sejnowski, *Unsupervised Learning: Foundations of Neural Computation*, The MIT Press, 1999.

- [65] T. Kohonen, *Self-organizing maps*, Springer series in information sciences. Springer, 1995.
- [66] H. Chen, *Knowledge Management Systems: A Text Mining Perspective*, Knowledge Computing Corporation, 2001.
- [67] P. Arabie, L. J. Hubert, and G. De Soete, *Clustering and Classification*, World Scientific, 1996.
- [68] R. A. Schowengerdt, *Remote Sensing, Models, and Methods for Image Processing*, Academic Press, 1997.
- [69] P. Zegers, *Some New Results on the Architecture, Training Process, and Estimation Error Bounds for Learning Machine*, Ph.D. dissertation, The University of Arizona, Tucson, AZ, 2002.
- [70] F. H. Walters, L. R. Parker, S. L. Morgan, and S. N. Deming, *Sequential Simplex Optimization: A Technique for Improving Quality and Productivity in Research, Development, and Manufacturing*, Chemometrics Series. CRC Press, 1991.
- [71] D. B. Fogel, *Evolutionary Computation: The Fossil Record*, Selected Readings on the History of Evolutionary Algorithms. IEEE Press, 1998.
- [72] J. G. Kemeny, J. L. Snell, and A. W. Knapp, *Denumerable Markov Chains*, Graduate Texts in Mathematics. Springer-Verlag, 1976.

- [73] D. G. Dudley, *Mathematical Foundations for Electromagnetic Theory*, The IEEE/OUP Series on Electromagnetic Wave Theory. Oxford University Press, 1994.
- [74] B. R. Frieden, *Probability, Statistical Optics, and Data Testing*, Springer Series in Information Sciences. Springer-Verlag, 1991.
- [75] O. L. Astrachan, *A Computer Science Tapestry: Exploring Computer Science and Programming with C++*, McGraw-Hill, 1999.