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

Signal Detection in Medical Imaging

by
Hongbin Zhang

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

2001

THE UNIVERSITY OF ARIZONA ®
GRADUATE COLLEGE

As members of the Final Examination Committee, we certify that we have read the dissertation prepared by Hongbin Zhang

entitled Signal Detection In Medical Imaging

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

R N Strickland	6-7-01
Robin N. Strickland, Ph.D.	Date
H. H. Barrett	6-7-01
Harrison H. Barrett, Ph.D.	Date
Micheal W. Marcellin	6/7/01
	Date
Mark A. Neifeld, Ph.D.	
	Date
Kathleen L. Virga	6/7/01
Kathleen L. Virga, Ph.D.	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.

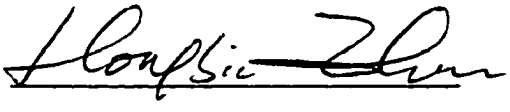
I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

R N Strickland	6-7-01
Co-Dissertation Director Robin N. Strickland, Ph.D.	Date
H. H. Barrett	6-7-01
Co-Dissertation Director Harrison H. Barrett, Ph.D.	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 head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: 

ACKNOWLEDGMENTS

I would like to express my deep appreciation to all who have given me guidance and support towards my education and especially for this research. I would like to give sincere thanks to both my advisors, Dr. Harrison H. Barrett and Dr. Robin N. Strickland, for introducing me into medical imaging processing and signal detection, and for giving me the opportunity to work by their sides and for creating such an excellent research environment to work in. I do not know how to express my appreciation to them in words, but I will always miss them in my life. I would also like to thank Dr. Eric Clarkson who is actually my *unofficial advisor*. He is such a smart person that I can always get help and advice from him. I would like to extend my appreciation to the other members of the radiology research group for their cooperation and friendship. Special thanks go to Brandon Gallas for his long-term help when I was in this group. In addition, I would like to thank Dr. Michael W. Marcellin, Dr. Kathleen Virga, Dr. Mark A. Neifeld, Dr. Dickson Lang and Dr. Bruce A. Thomas in the Department of Electrical & Computer Engineering for their discussions and help. Finally, I would like to thank my parents and my wife with whom I can share the joy, sorrow and pain in my life. It is their consistent support and love that help me to overcome all the difficulties and loneliness. I know that they will truly feel happy in this moment for their son and husband.

Dedication

To my Mam and Dad.

Table of Contents

TABLE OF CONTENTS—*Continued*

TABLE OF CONTENTS—*Continued*

LIST OF FIGURES

LIST OF FIGURES—*Continued*

LIST OF FIGURES—*Continued*

LIST OF FIGURES—*Continued*

List of Tables

ABSTRACT

The goal of this research is to develop computational methods for predicting how a given medical imaging system and reconstruction algorithm will perform when the resulting images are used by mathematical observers for tumor detection. Here the mathematical observer is the ideal observer which sets an upper limit to the performance as measured by the Bayesian risk or receiver operating characteristic analysis. This dissertation concentrates on constructing the ideal observer in complex detection problems and estimating its performance. Thus the methods reported in this dissertation can be used to approximate the ideal observer for real medical images.

We define our detection problem as a two-hypothesis detection task, where a known signal is superimposed on a random background with complicated distributions, and embedded in independent Poisson noise. The first challenge of this detection problem is that the distribution of the random background is usually unknown and difficult to estimate. The second challenge is that the calculation of the ideal observer is computationally intensive for non-stylized problems. In order to solve these two problems, our work relies on multiresolution analysis of images. The multiresolution analysis is achieved by decomposing an image into a set of spatial frequency bandpass images, so each bandpass image represents information about a particular fineness of detail or scale. Connected with this method, we will use three types of image representations by invertible linear transforms. They are the orthogonal wavelet transform, pyramid transform (or dyadic wavelet transform) and independent component analysis.

Based on the findings from human and mammalian vision, we can model textures by using marginal densities of a set of spatial frequency bandpass images. In order to estimate the distribution of an ensemble of images given the empirical marginal distributions of filter responses, we can use the maximum entropy principle and get

a unique solution.

We find that the ideal observer calculates a posterior mean of the ratio of conditional density functions, or the posterior mean of the ratio of two prior density functions, both of which are high-dimensional integrals and have no analytic solution usually. But there are two ways to approximate the ideal observer. The first one is a classic decision process; that is, we construct a classifier following feature-extraction steps. We use the integrand of the posterior mean as features, which are calculated at the estimated background close to the posterior mode. The classifier combines these features to approximate the integral (or the ideal observer). Finally, if we know both the conditional density function and the prior density function, then we can also approximate the high-dimensional integral by Monte Carlo integration methods. Since the calculation of the posterior mean is usually a very high-dimensional integration problem (the number of integrations equals the number of pixels), we must construct a Markov chain which can explore the posterior distribution efficiently. We will give two proposal functions. The first proposal function is the likelihood function of random backgrounds. The second method makes use of the multiresolution representation of the image by decomposing the image into a set of spatial frequency bands. Sampling one pixel in each band (in the same spatial location) equivalently updates a cluster of pixels in the neighborhood of the pixel location in the original image.

Chapter 1
INTRODUCTION

The purpose of medical imaging systems is to map the unknown attributes of objects (soft tissue, bones, trace elements etc.) to an accessible, measurable image. The basic clinical imaging techniques are the measurement of x-ray and ultrasound attenuation through the body, the number of gamma rays emitted by radioactive tracers which have been injected in the body, and the spatial density of spins using nuclear magnetic resonance imaging (MRI).

The values of the image yield information about the object, but the image itself need not have any resemblance to the object. As a matter of fact, many types of medical imaging are indirect imaging like tomography and its varied forms, including x-ray computed tomography (CT), emission tomography such as single-photon emission computed tomography (SPECT) and positron emission tomography (PET) as well as MRI. In all of these methods, the data consist of a set of line integrals or plane integrals of the object, and a reconstruction step is necessary to obtain the final image.

The imaging system is usually designed to maximize the amount of information about the object that can be obtained. Therefore, the quality of an imaging system is defined by how well inferences about an underlying object can be made using its image as input. More importantly, image quality must be assessed on the basis of average performance of some inference task; thus, image quality assessment is a process of statistical inference by some observers, for example, human and machine vision systems. The statistical inference is that, given an observed image $\mathbf{g}$, both human and machine vision systems compute *the most probable interpretations or causes of the observed images.* In statistics, this is to compute some functional of the

posterior probability density which is

$$p(\mathbf{f}|\mathbf{g}) \sim p(\mathbf{g}|\mathbf{f})\, p(\mathbf{f})\,, \tag{1.1}$$

where the conditional probability density $p(\mathbf{g}|\mathbf{f})$ is referred to as the likelihood function which tells how likely it is that a given data set is obtained when some underlying state of nature is true. The likelihood function depends on both the deterministic characteristics of the imaging system, specified by $\mathcal{H}$ which contains the system parameters to be optimized, and the measurement noise. The prior density $p(\mathbf{f})$ specifies the randomness of the object. Unfortunately, it is very difficult to be precise about the meaning of $p(\mathbf{f})$ for realistic problems.

We usually divide statistical inference into two groups: classification and parameter estimation. In this dissertation, we will consider one of the statistical inferences, the classification tasks. In the imaging literature, the terms pattern recognition, signal detection, discriminant analysis and hypothesis testing fall under this category. In medical imaging, more specifically, a signal-detection task is to find lesions on anatomical backgrounds which are the images of biological tissues, and measurement noise; thus the lesion is the signal which characterizes the abnormality of the object. The observer in a lesion-detection task on real clinical images might be a radiologist. Alternatively, many research efforts are developing mathematical observers for computer aided diagnosis (CAD) [Chan 1988], [Davies 1990], [Karssemeijer 1991], [Strickland 1996]. However, the overall goal of this research is to develop computational methods for predicting how a given medical imaging system and reconstruction algorithm will perform when the resulting images are used by mathematical observers for tumor detection. These methods will then be used to compare imaging systems or optimize the parameters of a given system. They will also be used to compare reconstruction algorithms for a given imaging system and to find optimal values for the free parameters that such algorithms normally contain. During the past decade, research in our group had been focusing on the performance of the ideal linear dis-

criminant function, i.e., the Hotelling trace [Smith 1986], [Fiete 1987], [Barrett 1990], [Barrett 1992], [Barrett 1993]. The current thrust of our research has been in constructing the *ideal observer* or *Bayesian observer* who utilizes all statistical information available regarding the task to maximize task performance as measured by Bayes risk or some other related measure of performance [Barrett 1997], [Barrett 1998a], [Clarkson 2000].

In order to develop effective methods for computing the ideal observer on realistic tumor detection tasks, we will introduce several linear transform methods in chapter 2. These linear transforms are the necessary tools for analyzing images. The information about the object is uniformly distributed over all the pixels in images, and the number of pixels is usually so large that we even cannot estimate the covariance matrix of the image. For example, if the image size is $10^3 \times 10^3$ then there are 10^6 pixels and the covariance matrix has a size of $10^6 \times 10^6$. In order to let the sample covariance matrix be invertible, we need at least one million samples of the training images; moreover, in order to get a stable estimate of the covariance matrix we need 10 times larger than the number of pixels, so we need 10 million samples of the training images. Note that the we have just considered the second-order statistics of the images; the number of samples increases exponentially with the order of statistics. A clever way to solve this problem is to transform the image into another representation such that the information about the object is not uniformly distributed over the pixels.

Since the beginning of 1980's there has been a great deal of interest in representations that retain spatial localization as well as localization in the spatial frequency domain. This is achieved by decomposing the image into a set of spatial frequency bandpass component images. Each bandpass image represents information about a particular fineness of detail or scale. There is evidence that the human visual system uses such a representation, and multiresolution schemes are becoming increasingly popular in machine vision and image processing. Most of the work in the 1980's revolves around a representation known as a pyramid, which is a data structure de-

signed to support efficient scaled convolution through reduced image representation [Cross 1983], [Burt 1983], [Adelson 1984]. It consists of a sequence of copies of an original image in which both sample density and resolution are decreased in regular steps. During the same time, another representation has been proposed, but for geophysical signals rather than images. Morlet, a geophysical engineer, invented a set of windowed cosine waves, called Morlet wavelets, for the integral transform of his seismic data, and Grossmann, who was a physicist, gave the inversion formula for the transform. Their work was linked with harmonic analysis by Meyer, who was a pure mathematician. Finally, it is Mallat who connected the wavelet decomposition with the pyramid transform in image processings and gave the multiresolution analysis method to construct the orthonormal wavelet bases and a recursive filtering algorithm to compute the wavelet coefficients [Mallat 1989b], [Mallat 1989c]. A good description of the history of wavelets can be found in [Daubechies 1996]. Throughout this dissertation, multiresolution analysis methods are repeatedly recalled and used in almost each chapter. Another invertible linear transform, called independent component analysis (ICA), was proposed to let the components of the transformed vector be independent of each other. The design of the ICA transform matrix requires higher-order statistics of the data set, either explicitly or implicitly, so we often use it within feature spaces which have smaller dimensions. The pyramid transform, wavelet transform and independent component analysis are invertible linear transforms; thus the performance of the ideal observer in terms of the Bayesian error is invariant under these transforms.

In chapter 3, we will discuss methods to synthesize the random background and learn the statistical model from a set of training images. We will give two types of method to generate the random background. The first type of random background is called the lumpy background proposed by Rolland and Barrett for the simulation of anatomical variation in nuclear medicine images [Rolland 1992], [Rolland 1997]. The second type of random background is the clustered-blob lumpy background for the

simulation of mammograms [Bochud 1998].

In realistic problems of decision or inference, we often have prior information about the object and background, but how to translate prior information into a definite prior probability density assignment? We will discuss learning the statistical model from a set of training images. In order to simplify the task of learning the statistical model of the random background, the stationary assumption and the Markov assumption are made. We will introduce briefly the Markov random field and the Gibbs distribution which have been used often to model texture images. Finally, we will use multiresolution analysis techniques to get a set of bandpass image representations and use the maximum entropy principle to derive the probability density of the random background using the marginal histograms of the bandpass images [Zhu 1997], [Zhu 1998]. This technique can be used to get the prior density of the random background from a set of training images. We will use it in chapter 5 for calculating the discriminant function of the ideal observer and in chapter 6 for Markov-chain Monte Carlo integration.

In chapter 4, we will discuss one special ideal observer, called the optimal linear discriminant function or the Hotelling observer, when the random background and the measurement noise have Gaussian distributions. When the measurement noise has a Poisson distribution, the Hotelling observer is still a good approximation to the ideal observer since the Poisson density function is very close to the Gaussian when the mean value is reasonably large. The template of the Hotelling observer includes the inverse of a large covariance matrix. We will discuss several methods to estimate this template.

Chapter 5 describes the computational methods for the ideal observer. We begin with a discussion about the strategies of the ideal observer. The ideal observer calculates the ratio of two probability density functions of the image data under different hypotheses. The probability density ratio can be expressed as a functional of the posterior density (or the posterior mean). Based on this strategy, we can design the

feature extraction followed by a classifier. The combination of feature extraction and classifiers approximate the posterior mean, which is a high-dimensional integral. We propose two feature extractions, Λ_{bke} and $\Lambda_{\text{noisefree}}$. The first one is the ratio of two conditional densities of the image data, and the second one is the ratio of two prior densities. Since the conditional density of data is usually modeled as a Poisson density function, we can easily calculate Λ_{bke}. In order to get the ratio of prior densities by using a set of training images, we propose two methods to approximate this ratio. The first method is to apply an invertible transform to "gaussianize" the image, then calculate the ratio of two Gaussian densities. The second method is to make use of the statistical model of the texture images as discussed in chapter 3.

In chapter 6, we will discuss another computational method for the ideal observer; that is, we will calculate the Monte Carlo integration of the posterior mean mentioned in the above paragraph. Unfortunately, direct independent sampling from the posterior distribution is difficult in high-dimensional problems. Thus we use Markov chain Monte Carlo (MCMC) methods for dependent sampling from the posterior distribution. In MCMC the objective is to generate a sequence of samples, called the Markov chain, with a specified equilibrium distribution. There are many ways of constructing these chains, but they all can be classified into the general framework of the Metropolis-Hastings algorithm. At each iteration, the Metropolis-Hastings algorithm selects a candidate sample from a proposal function, then accepts it according to an acceptance probability.

Since the calculation of the posterior mean is usually a very high-dimensional integration problem (the number of integrations equals the number of pixels), we must construct a Markov chain which can explore the posterior distribution efficiently. We will give two proposal functions. The first proposal function is the likelihood function of random backgrounds. If the image data has an independent Poisson conditional distribution then the likelihood function is an independent Gamma density function, so it is easy to generate the candidate sample by using 1D Gamma random variate

generator. The second method makes use of the multiresolution representation of the image by decomposing the image into a set of spatial frequency bands. Sampling one pixel in each band (in the same spatial location) equivalently updates a cluster of pixels in the neighborhood of the pixel location in the original image. Simulation results show that the second method mixes faster than the first method.

Chapter 2

WAVELET TRANSFORM AND INDEPENDENT COMPONENT ANALYSIS

2.1 Introduction

In this chapter we will discuss several techniques for the characterization of texture properties. They are 1) orthogonal discrete wavelet transform 2) pyramid transform 3) independent component analysis. Traditional statistical approaches to texture analysis include co-occurrence matrices [Haralick 1973], second-order statistics [Chen 1983], Gauss-Markov random fields [Kashyap 1982]. These methods are restricted to the analysis of spatial interactions over relatively small neighborhoods, so they are suitable only for the analysis of the class of microtextures. Recently, people have found from studies of human and mammalian vision that localized spatial and frequency representation of the natural images is capable of preserving both local and global information. Motivated by this finding, the pyramid transform of the image using Gabor-like filter banks with different scales and orientations is used in texture analysis and synthesis. Mathematically, the pyramid transform method is based on frame theory, which analyzes the completeness, stability and redundancy of linear discrete signal representations. The analysis part is the calculation of the expansion coefficients using an inner product between the input signal and the projection function. The synthesis part is the calculation of the signal from the expansion coefficients and the basis functions. Since the early work in the 1980's by Morlet, Grossmann, Meyer, Mallat, Daubechies etc., wavelets and wavelet transforms have caught the attention of the applied mathematics communities in signal processing, statistics, and numerical analysis. The goal of most wavelet research is to create a set of basis functions and transforms that will give an efficient description of a func-

tion or signal; thus wavelets are rooted in functional analysis, Fourier transforms, harmonic analysis and frame theory, all of which have been studied long before the 1980's. We will introduce the wavelet systems and the orthogonal discrete wavelet transform (DWT) by multiresolution formulation. The acronym DWT is sometimes used to denote wavelet series decomposition of continuous-time signals. Here DWT refers to a type of wavelet transform which decomposes discrete-time signals. Note that the wavelet basis functions generally have a spatial-frequency localization; therefore, we can also directly use the wavelet basis functions in the pyramid transform, which is called the discrete wavelet frame (DWF). Different from the DWT, the DWF uses an overcomplete wavelet decomposition in which the output of the filter banks is not subsampled. Although the DWF introduces redundancy, it has a desirable property which the DWT does not have; that is, the texture description is invariant with respect to translations of the input signal.

The requirements of spatial-frequency localization for the basis functions in the pyramid transforms and the DWTs are not uniquely borrowed from human and mammalian vision studies. Recently, both the information theory community and the statistics community have proposed independent component analysis, which searches for an invertible linear transformation to minimize the statistical dependence between the elements of a random vector. A.J. Bell and T. J. Sejnowski [Bell 1995] used an unsupervised learning algorithm based on information maximization for the independent component analysis of an ensemble of natural scenes, and they found that the final filters are also localized and oriented edge filters.

The orthogonal wavelet transform, pyramid transform (or wavelet frame) and independent component analysis are all invertible transforms. Therefore if we define the class separability in terms of receiver operating characteristic curve or the Bayes error, then it is invariant under these transforms. All of these transforms share the same characteristics of the basis functions, that is, they are localized in spatial and frequency domain. Among them, the ICA is preferable because the components of

the transformed vector are as independent as possible, but, the ICA bases are not fixed. In order to get the ICA bases, we need a large number of training samples and long computational time. The orthogonal wavelet transform works well in image compression and data analysis and other tasks for which the Fourier transform has been used traditionally. One of its drawbacks is the lack of translation invariance. On the other hand, the pyramid transform and wavelet frame overcome this difficulty and have been used intensively in texture synthesis and feature detection.

2.2 Multiresolution Formulation of Wavelet Systems

The wavelet transform is usually described as a multiresolution decomposition for the functions in L_2 space [Mallat 1998], [Burrus 1998], the space of all functions with a well defined integral of the square of the modulus of the function. However, it is more appropriate to consider wavelet representations for discrete signals in l_2 (the space of square summable sequences) since we often have the discrete signals rather than the continuous ones [Shensa 1992], [Rioul 1993]. We consider the sequence of nested subspaces $l_2 = V_0 \supset V_1 \supset ... \supset V_I$, where V_i =span$\{\varphi_{i,l}\}_{l\in Z}$ is the approximation space at resolution i, and $\varphi_{i,l}$ is the discrete normalized basis function in V_i. We also introduce the detail subspace W_i =span$\{\psi_{i,l}\}_{l\in Z}$ at resolution i , which is defined as the orthogonal component of V_i with respect to V_{i-1}, i.e., $V_{i-1} = V_i \oplus W_i$ and $V_i \perp W_i$. From the above discussions we see that functions in the l_2 space are expanded by a set of two-dimensional orthonormal basis functions $\psi_{i,l}, i = 1, ..., I$ and $\varphi_{I,l}$, i.e., $l^2 = V_I \oplus W_1 \oplus W_2 \oplus ... \oplus W_I$. The full discrete wavelet expansion of a signal $x \in l_2$ is

$$x(k) = \sum_l c_{(I)}(l)\,\varphi_{I,l}(k) + \sum_{i=1}^{I}\sum_l d_{(i)}(l)\,\psi_{i,l}(k), \tag{2.1}$$

where $c_{(I)}(l) = \langle x, \varphi_{I,l}\rangle$ and $d_{(i)}(l) = \langle x, \psi_{I,l}\rangle$, and $\langle .,. \rangle$ denotes the standard l_2 inner product.

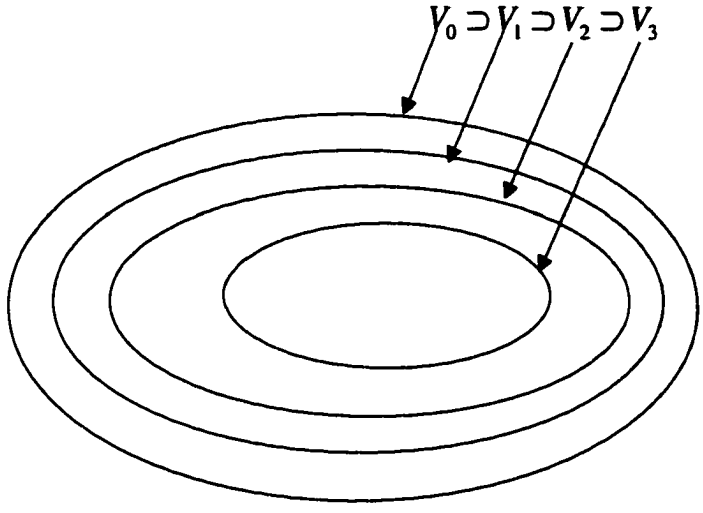


FIGURE 2.1. Nested spaces in l_2

We define a set of normalized basis functions of V_i and W_i in terms of the integer translations of the discrete filters [Unser 1995],

$$\varphi_{i,l}(k) = 2^{i/2} h_i\left(k - 2^i l\right), \tag{2.2}$$

$$\psi_{i,l}(k) = 2^{i/2} g_i\left(k - 2^i l\right), \tag{2.3}$$

where i and l are the scale and translation indices and the factor $2^{i/2}$ is an inner product normalization. Since $V_i = V_{i+1} \oplus W_{i+1}$, there exists an important relationship between the basis functions of V_{i+1}, W_{i+1} and V_i; that is, we can express the basis functions of V_{i+1} and W_{i+1} in terms of the linear combination of the basis functions of V_i by

$$\varphi_{i+1,l}(k) = \sum_n \sqrt{2} h(n) \varphi_{i,2l+n}(k), \tag{2.4}$$

$$\psi_{i+1,l}(k) = \sum_n \sqrt{2} g(n) \varphi_{i,2l+n}(k). \tag{2.5}$$

Equivalently, the discrete filter at scale 2^{i+1} is related to the discrete filter at scale 2^i

by

$$h_{i+1}(k) = \sum_l h(l)\, h_i\left(k - 2^i l\right), \tag{2.6}$$

$$g_{i+1}(k) = \sum_l g(l)\, h_i\left(k - 2^i l\right), \tag{2.7}$$

with the initial condition $h_0(k) = \delta(k)$ as the basis function of the space V_0; $h(l)$ and $g(l)$ are the low-pass filter and the high-pass filter. We can see from the above equations that h_{i+1} and g_{i+1} are dilated by a factor of two from h_i and g_i. In practice, we can generate $h_{i+1}(k)$ and $g_{i+1}(k)$ by the following equations:

$$h_{i+1}(k) = [h]_{\uparrow 2^i} * h_i(k), \tag{2.8}$$

$$g_{i+1}(k) = [g]_{\uparrow 2^i} * h_i(k), \tag{2.9}$$

where the notation $[.]_{\uparrow m}$ denotes upsampling by a factor of m, and the notation $*$ denotes linear convolution. Such a sequence of filters (a filter bank) can be used to decompose a signal in subbands of approximately one octave each. Figs. 2.2, 2.3 and 2.4 illustrate the frequency characteristics of the underlying filter bank for different wavelets, and we can verify that they satisfy the identity

$$|H_I(w)|^2 + \sum_{i=1}^{I} |G_i(w)|^2 = 1, \tag{2.10}$$

where $H_I(w)$ and $G_i(w)$ are the discrete-time Fourier transform (DTFT) of $h_I(k)$ and $g_i(k)$.

Definition The DTFT of a discrete-time signal $x(k)$ is the complex-valued function of the continuous (frequency) variable, w, defined by

$$X(w) = \sum_{k=-\infty}^{\infty} x(k) \exp(-jkw). \tag{2.11}$$

For a finite length of discrete-time signal, the DTFT may be easily approximated using a discrete Fourier transform (DFT) as follows:

$$\widetilde{X}(n) = \sum_{k=0}^{N-1} x(k) \exp\left(\frac{-j2\pi kn}{N}\right). \tag{2.12}$$

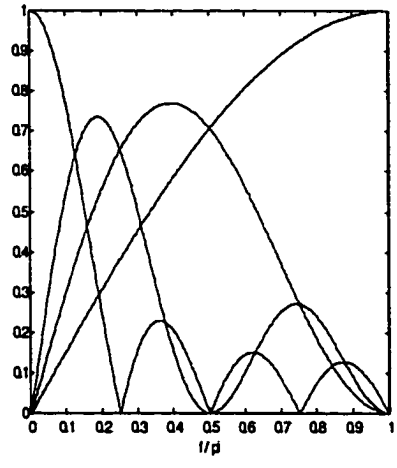


FIGURE 2.2. Frequency magnitude $|H_3(w)|$ and $|G_i(w)|, i = 1, 2, 3$ of the Haar wavelet transform.

Comparing (2.11) and (2.12), we see that the DFT is the sampled DTFT of the finite sequence $x(k)$ extended by zeros outside the interval $[0,\ N-1]$. If the length of the sequence is small, then we can use zero padding to increase the frequency resolution. For example, the frequency magnitudes of the underlying filter bank for different wavelets in Figs. 2.2, 2.3 and 2.4 are calculated by the DFT, with $N = 1024$. If the filter length is less than N, then we use zero padding to fill it up to 1024 samples.

Next we will give some properties of the filter coefficients h and g in order to satisfy the orthogonality of the basis functions.

2.2.1 Necessary conditions

Theorem 1 If integer translates of $\varphi_{i,l}(k)$ are orthogonal, as defined by

$$\langle \varphi_{i,l}, \varphi_{i,m} \rangle = \delta(l-m), \tag{2.13}$$

then

$$\sum_n h(n)\, h(n-2k) = \frac{1}{2}\delta(k). \tag{2.14}$$

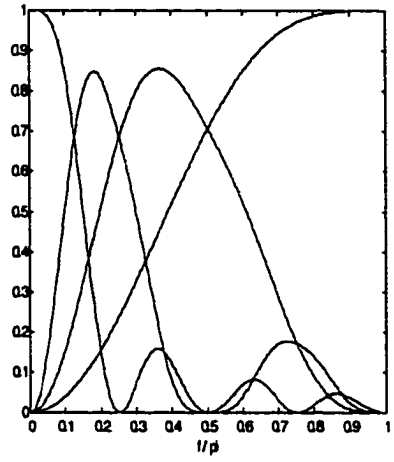


FIGURE 2.3. Frequency magnitude $|H_3(w)|$ and $|G_i(w)|, i = 1, 2, 3$ of the Daubechies 4 wavelet transform.

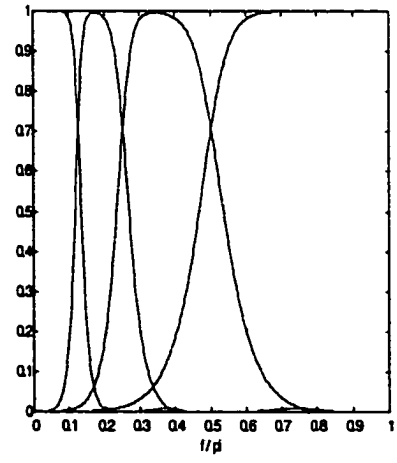


FIGURE 2.4. Frequency magnitude $|H_3(w)|$ and $|G_i(w)|, i = 1, 2, 3$ of the Battle-Lemarie cubic spline wavelet transform.

Proof: By combining Eq. (2.2) and Eq. (2.6), we have the basis function for a subspace V_{i+1} as

$$\varphi_{i+1,l}(k) = 2^{(i+1)/2} \sum_n h(n)\, h_i\left[k - 2^i(2l+n)\right]. \tag{2.15}$$

Thus the inner product of the basis function $\varphi_{i+1,l}$ and $\varphi_{i+1,m}$, $\langle \varphi_{i+1,l}, \varphi_{i+1,m} \rangle$, is

$$\begin{aligned} & \sum_k 2^{i+1} \sum_n \sum_{n'} h(n)\, h(n')\, h_i\left[k - 2^i(2l+n)\right] h_i\left[k - 2^i(2m+n')\right] \\ = & \sum_n \sum_{n'} 2h(n)\, h(n') \sum_k \varphi_{i,2l+n}(k)\, \varphi_{i,2m+n'}(k). \end{aligned} \tag{2.16}$$

The third summation inside the last equation is the inner product of the basis function $\varphi_{i,2l+n}$ and $\varphi_{i,2m+n'}$, and

$$\langle \varphi_{i,2l+n}, \varphi_{i,2m+n'} \rangle = \delta(2l+n-2m-n'). \tag{2.17}$$

Therefore,

$$\begin{aligned} \langle \varphi_{i+1,l}, \varphi_{i+1,m} \rangle &= \sum_n \sum_{n'} 2h(n)\, h(n')\, \delta(2l+n-2m-n') \\ &= \sum_n 2h(n)\, h[n+2(l-m)]. \end{aligned} \tag{2.18}$$

Letting $k = m - l$, we have

$$\sum_n h(n)\, h[n-2k] = \frac{1}{2}\delta(k). \tag{2.19}$$

This theorem shows that in order to let the set of basis functions $\varphi_{i,l}$ be orthogonal under integer translations, it is necessary that the filter coefficients be orthogonal themselves after decimating, i.e.,

$$[h(k) * h(-k)]_{\downarrow 2} = \frac{1}{2}\delta(k). \tag{2.20}$$

The filter coefficients $h(k)$ that satisfy Eq. (2.14) are called the quadrature mirror filter (QMF). If we apply the discrete-time Fourier transform (DTFT) to both sides of Eq. (2.20), then we get

$$|H(w)|^2 + |H(w+\pi)|^2 = 1. \tag{2.21}$$

Theorem 2 If integer translates of $\varphi_{i,l}(k)$ and $\psi_{i,m}(k)$ are orthogonal as defined by

$$\langle \varphi_{i,l}, \psi_{i,m} \rangle = 0, \tag{2.22}$$

then

$$\sum_n h(n)\, g(n-2k) = 0. \tag{2.23}$$

and

$$g(n) = (-1)^n h(1-n) \tag{2.24}$$

Proof: The proof for (2.23) is similar to the previous method. If we apply the discrete-time Fourier transform (DTFT) to both sides of (2.23), then we get

$$H(w)\overline{G(w)} + H(w+\pi)\overline{G(w+\pi)} = 0. \tag{2.25}$$

where $\overline{G(w)}$ is the complex conjugate of $G(w)$.

In order to prove (2.24), we see from (2.21) that $H(w)$ and $H(w+\pi)$ cannot be zero at the same time, thus, in order to let (2.25) exist, we need a 2π periodic function $\lambda(w)$ such that

$$G(w) = \overline{\lambda(w)\, H(w+\pi)}, \tag{2.26}$$

and

$$\lambda(w) + \lambda(w+\pi) = 0. \tag{2.27}$$

One special solution is

$$\lambda(w) = \exp(jw), \tag{2.28}$$

thus

$$G(w) = \exp(-jw)\,\overline{H(w+\pi)}. \tag{2.29}$$

By the inverse DTFT, we have

$$g(n) = (-1)^n h(1-n). \tag{2.30}$$

From the above equation, we see that the highpass filter may be constructed from the lowpass filter by (1) modulating (multiplying) by $(-1)^n$ (equivalent to shifting by π in the Fourier domain), (2) flipping (i.e., reversing the order of the filter taps), (3) spatially shifting by one sample. Note that the third operation cannot be seen from the tables of the highpass filters, but is built into the convolution and downsampling code.

2.2.2 Filter-bank implementation

Due to the work by S. Mallat [Mallat 1998], we need not calculate the wavelet expansion coefficients by the inner product between the input signal and the basis functions; instead, we can use a filter-bank implementation to calculate the expansion coefficients iteratively.

Analysis Recall that the basis functions of V_{i+1} and W_{i+1} are expressed in terms of linear combinations of the basis functions of V_i

$$\varphi_{i+1,l}(k) = \sqrt{2}\sum_n h(n)\,\varphi_{i,2l+n}(k), \tag{2.31}$$

$$\psi_{i+1,l}(k) = \sqrt{2}\sum_n g(n)\,\varphi_{i,2l+n}(k). \tag{2.32}$$

The expansion coefficients in V_{i+1} are

$$\begin{aligned} c_{i+1}(l) &= \langle x(k), \varphi_{i+1,l}(k)\rangle \\ &= \sum_k x(k)\sqrt{2}\sum_n h(n)\,\varphi_{i,2l+n}(k) \\ &= \sum_n h(n)\sqrt{2}\sum_k x(k)\,\varphi_{i,2l+n}(k) \\ &= \sqrt{2}\sum_n h(n)\,c_i(2l+n), \end{aligned} \tag{2.33}$$

and the expansion coefficients in W_{i+1} are

$$\begin{aligned} d_{i+1}(l) &= \langle x(k), \psi_{i+1,l}(k) \rangle \\ &= \sum_k x(k) \sqrt{2} \sum_n g(n) \varphi_{i,2l+n}(k) \\ &= \sum_n g(n) \sqrt{2} \sum_k x(k) \varphi_{i,2l+n}(k) \\ &= \sqrt{2} \sum_n g(n) c_i(2l+n). \end{aligned} \tag{2.34}$$

By using the downsampling notation $[.]_{\downarrow 2}$,we can rewrite (2.33) and (2.34) as

$$c_{i+1}(n) = \sqrt{2} [h(-n) * c_i(n)]_{\downarrow 2}, \tag{2.35}$$

$$d_{i+1}(n) = \sqrt{2} [g(-n) * c_i(n)]_{\downarrow 2}. \tag{2.36}$$

Since the highpass filter has one pixel shift from the lowpass filter, the highpass and lowpass bands are subsampled on different pixels: the lowpass band retains the odd-numbered samples and the highpass band retains the even-numbered samples.

Synthesis Next we will give an expression for the expansion coefficients in V_i in terms of the expansion coefficients in V_{i+1} and W_{i+1}. Recall that $V_i = V_{i+1} \oplus W_{i+1}$, so any function $x(k) \in V_i$ can be expanded by the basis functions of V_{i+1} and W_{i+1}

$$x(k) = \sum_l c_i(l) \varphi_{i,l}(k) \tag{2.37}$$

$$= \sum_l c_{i+1}(l) \varphi_{i+1,l}(k) + \sum_l d_{i+1}(l) \psi_{i+1,l}(k). \tag{2.38}$$

Also note that $\varphi_{i+1,l}(k)$ and $\psi_{i+1,l}(k)$ can be expressed as linear combinations of $\varphi_{i,l}(k)$, so we have

$$x(k) = \sum_l c_{i+1}(l) \sqrt{2} \sum_n h(n) \varphi_{i,2l+n}(k) + \sum_l d_{i+1}(l) \sqrt{2} \sum_n g(n) \varphi_{i,2l+n}(k). \tag{2.39}$$

Changing the index variable $m = 2l + n$, we obtain

$$\begin{aligned} x(k) &= \sum_l c_{i+1}(l)\sqrt{2}\sum_m h(m-2l)\,\varphi_{i,m}(k) + \sum_l d_{i+1}(l)\sqrt{2}\sum_n g(m-2l)\,\varphi_{i,m}(k) \\ &= \sum_m \left\{ \sqrt{2}\sum_l c_{i+1}(l)\,h(m-2l) + \sqrt{2}\sum_l d_{i+1}(l)\,g(m-2l) \right\} \varphi_{i,m}(k). \quad (2.40) \end{aligned}$$

Comparing (2.37) with (2.40), we have

$$c_i(m) = \sqrt{2}\sum_l c_{i+1}(l)\,h(m-2l) + \sqrt{2}\sum_l d_{i+1}(l)\,g(m-2l). \quad (2.41)$$

By using the upsampling notation $[.]_{\uparrow 2}$,we can express (2.41) as

$$c_i(m) = [c_{i+1}(m)]_{\uparrow 2} * \sqrt{2}h(m) + [d_{i+1}(m)]_{\uparrow 2} * \sqrt{2}g(m). \quad (2.42)$$

We illustrate this filter bank algorithm in Fig. 2.5, in which $h'(n) = \sqrt{2}h(n)$ and $g'(n) = \sqrt{2}g(n)$.

Computational cost In practice, the input signal has a finite size of N samples. The convolutions in the filter-bank algorithm are replaced by circular convolutions. This is equivalent to applying periodic extensions of the signal. But the periodic extension has the disadvantage of creating large wavelet coefficients at the borders. If the wavelet function is symmetric or antisymmetric, we can use symmetric extensions, which create smaller wavelet coefficients at the borders. However, the Haar wavelet is the only symmetric orthonormal wavelet with a compact support. Thus people often use biorthogonal wavelets which are compactly supported and can be either symmetric or antisymmetric [Mallat 1998].

Suppose h and g have K non-zero coefficients. With appropriate border calculations, each c_i and d_i has $2^{-i}N$ samples, (2.33) and (2.34) compute c_{i+1} and d_{i+1} from c_i with $2^{-i}NK$ additions and multiplications. The wavelet representation is therefore calculated with at most $2KN$ additions and multiplications. Similarly, the original signal is also recovered from the wavelet representation with at most $2KN$ additions and multiplications.

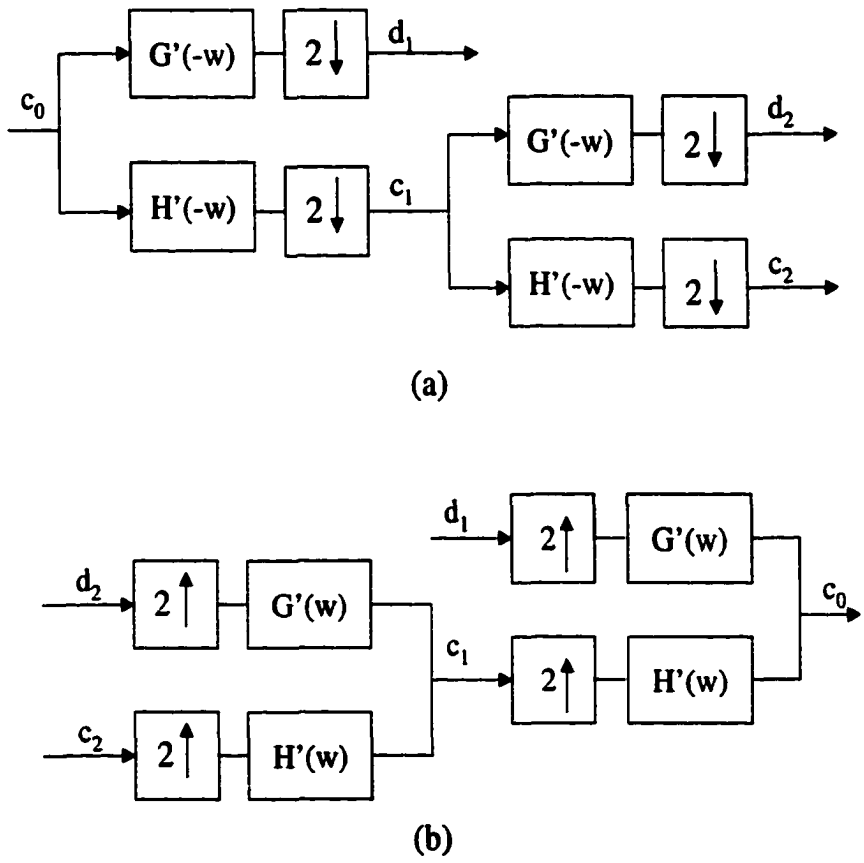


FIGURE 2.5. (a) Analysis filter bank of DWT. (b) Synthesis filter bank of DWT.

2.2.3 Two-Dimensional Wavelet Bases

We have discussed the one-dimensional wavelet bases in $l_2(Z)$ space. In order to apply the DWT to 2D images, we need to generalize the wavelet bases to two-dimensional $l_2(Z^2)$ space. As in one dimension, we begin with the multiresolution decomposition for $l_2(Z^2)$ space.

Recall that $V_i = \text{span}\{\varphi_{i,l}\}_{l \in Z}$ is the approximation space of $l_2(Z)$ at resolution i. A separable two-dimensional approximation space is composed of the tensor product spaces

$$V_i^2 = V_i \otimes V_i, \tag{2.43}$$

where V_i^2 is the approximation space of $l_2(Z^2)$ at resolution i. By the properties of the tensor product, the basis functions of V_i^2 are

$$\{\varphi_{i,l,m}(x,y) = \varphi_{i,l}(x)\,\varphi_{i,m}(x)\}_{(l,m) \in Z}. \tag{2.44}$$

As in one dimensional multiresolution decomposition, we let W_i^2 be the detail space equal to the orthogonal complement of the lower-resolution approximation space V_i^2 in V_{i-1}^2

$$V_{i-1}^2 = V_i^2 \oplus W_i^2. \tag{2.45}$$

Now if we substitute (2.43) into (2.45), we can relate the 2D W_i^2 space with 1D V_i and V_{i-1} spaces

$$(V_{i-1} \otimes V_{i-1}) = (V_i \otimes V_i) \oplus W_i^2. \tag{2.46}$$

Substituting $V_{i-1} = V_i \oplus W_i$ in (2.46), we have

$$W_i^2 = (V_i \otimes W_i) \oplus (W_i \otimes V_i) \oplus (W_i \otimes W_i). \tag{2.47}$$

Since $\{\varphi_{i,l}\}_{l \in Z}$ and $\{\psi_{i,l}\}_{l \in Z}$ are orthonormal bases of V_i and W_i, we derive that

$$\{\varphi_{i,l}(x)\,\psi_{i,m}(y),\ \psi_{i,l}(x)\,\varphi_{i,m}(y),\ \psi_{i,l}(x)\,\psi_{i,m}(y)\}_{(l,m) \in Z^2}$$

is an orthonormal basis of W_i^2.

The overall $l_2(Z^2)$ space is

$$l_2(Z^2) = V_I^2 \oplus W_I^2 \oplus W_{I-1}^2 \oplus ... \oplus W_1^2. \tag{2.48}$$

Hence

$$\left\{\varphi_{I,l}(x)\varphi_{I,m}(x),\ \varphi_{i,l}(x)\psi_{i,m}(y),\ \psi_{i,l}(x)\varphi_{i,m}(y),\ \psi_{i,l}(x)\psi_{i,m}(y)\right\}_{(l,m)\in Z^2, 1\le i\le I}$$

is an orthonormal basis of $l_2(Z^2)$.

The fast filter-bank implementation can be extended into two dimensions. Since the two-dimensional wavelet bases are separable, the two-dimensional wavelet transform for image analysis can be implemented using a one-dimensional transform. Each row of the input image is separately filtered by the same filters used in the one-dimensional transform. The resulting pair of row-transformed images are likewise filtered in the column direction, yielding three detail images: HL, LH, HH and one smooth version of the original image, LL. We illustrate the analysis and synthesis filter banks in Fig. 2.6.

In this section, we consider constructing the orthogonal expansion of the signal in both one dimensional and two dimensional spaces. Next we will introduce another signal representation called the frame which has no orthogonal requirements for the expansion functions.

2.3 Frames

Frame theory was originally developed by Duffin and Schaeffer [Duffin 1952] to reconstruct band-limited signals in a Hilbert space from its inner products with a family of vectors $\{\psi_n\}_{n\in z}$, which we call the frame. Unlike orthogonal basis functions, functions that form a frame are not necessarily linearly independent, thus a frame may be an overcomplete set.

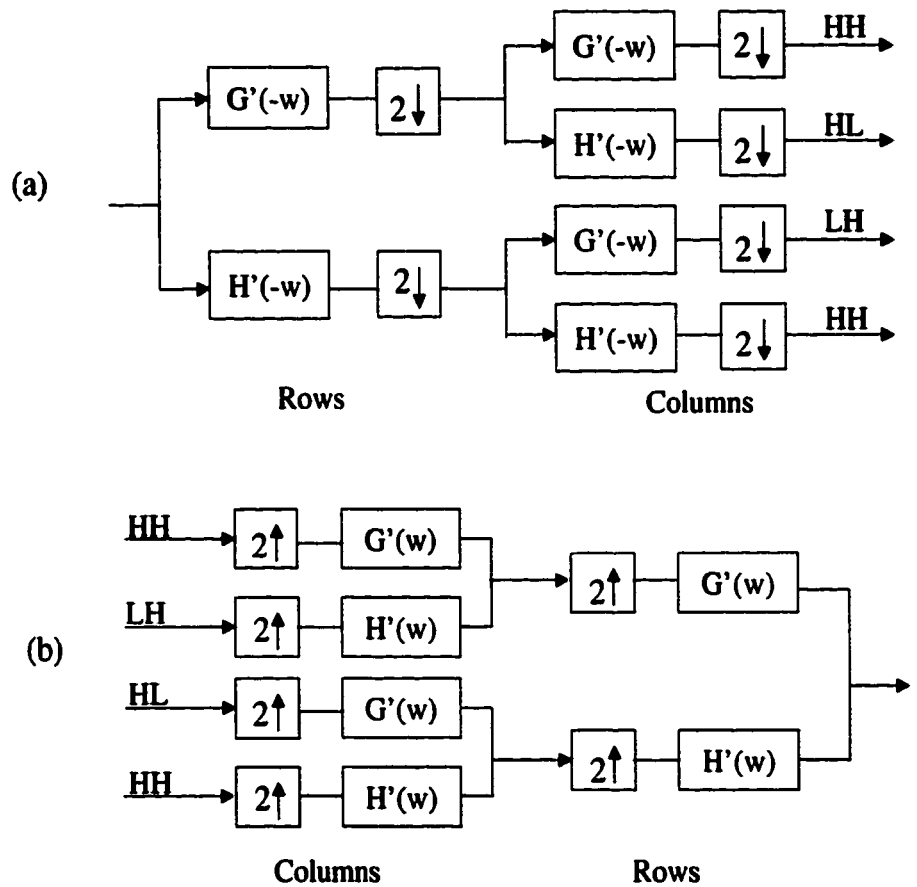


FIGURE 2.6. (a) Two-dimensional analysis filter bank, (b) Two-dimensional synthesis filter bank.

Definition 1 The family $\{\psi_n\}_{n\in z}$ is a frame if there exists two constants $0 < A \leq B < \infty$ such that for any function $x(k)$

$$A\|x\|^2 \leq \sum_n |\langle \psi_n, x\rangle|^2 \leq B\|x\|^2 . \tag{2.49}$$

Definition 2 The linear operator U is defined by

$$\forall n \in Z, \quad Ux(k) = \langle \psi_n, x\rangle . \tag{2.50}$$

If (2.49) is satisfied, then U is called a frame operator.

Mallat [Mallat 1998] proved that (2.49) is a necessary and sufficient condition guaranteeing that U is invertible with a bounded inverse. A frame thus defines a complete and stable signal representation; there exists a dual frame of functions $\{\overline{\psi}_n\}_{n\in z}$ that are biorthogonal to $\{\psi_n\}_{n\in z}$, i.e.,

$$\langle \overline{\psi}_n, \psi_m \rangle = \delta_{nm}, \tag{2.51}$$

and

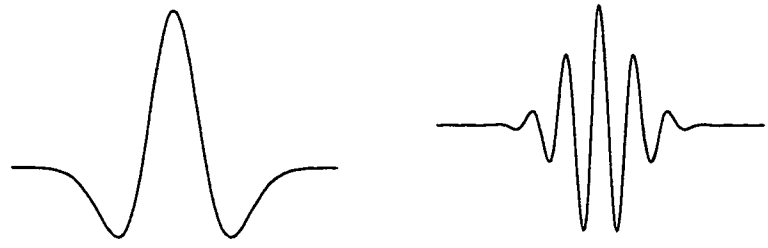
$$x(k) = \sum_n \langle \psi_n, x\rangle \, \overline{\psi}_n(k) . \tag{2.52}$$

From now on, we will call ψ_n the *projection* function, which is used to calculate an inner product, and $\overline{\psi}_n$ the *basis* function, which is used to reconstruct a signal. Daubechies [Daubechies 1992] showed that the tighter the frame bounds in (2.49) were, the better the analysis and synthesis system was conditioned. In other words, if $\frac{B}{A}$ is large, then there will be numerical problems in the analysis and synthesis calculations. If $A = B$ then the frame is said to be tight and

$$\overline{\psi}_n = \frac{1}{A}\psi_n, \tag{2.53}$$

so $x(k)$ can be expressed by

$$x(k) = \frac{1}{A}\sum_n \langle \psi_n, x\rangle \, \psi_n(k) , \tag{2.54}$$



Marr Wavelets (Mexican hat function) Morlet Wavelets

FIGURE 2.7. "Good" wavelets

which is the same as the expansion using an orthonormal basis except for the $\frac{1}{A}$ factor. If $A > 1$, then the frame is redundant, so A can be interpreted as a minimum-redundancy factor. If $A = B = 1$, then the tight frame becomes an orthonormal basis.

The concept of the frame is important because some functions which we want to use as the wavelets in image analysis and edge detection are not orthonormal, but they have good properties such as spatial and frequency localization, symmetry, compactness and regularity. These functions include the Marr wavelets and the Morlet wavelets as shown in Fig. 2.7. Based on frame theory, we can use these nonorthonormal basis functions to construct the pyramid transform, which is an overcomplete decomposition of the image. Another important reason is that the DWT is not shift invariant; a simple integer shift of the input signal will usually result in a nontrivial modification of the discrete wavelet transform. In order to overcome this limitation, the dyadic wavelet transform uses the same wavelet basis function as in the DWT, but these wavelet basis functions lie in all spatial locations. Therefore, the dyadic wavelet transform has a redundancy in representing the signal, and we call it the discrete wavelet frame (DWF). Next I will give three examples of the frame: (1) the Laplacian pyramid (2) the steerable pyramid and (3) the dyadic wavelet transform.

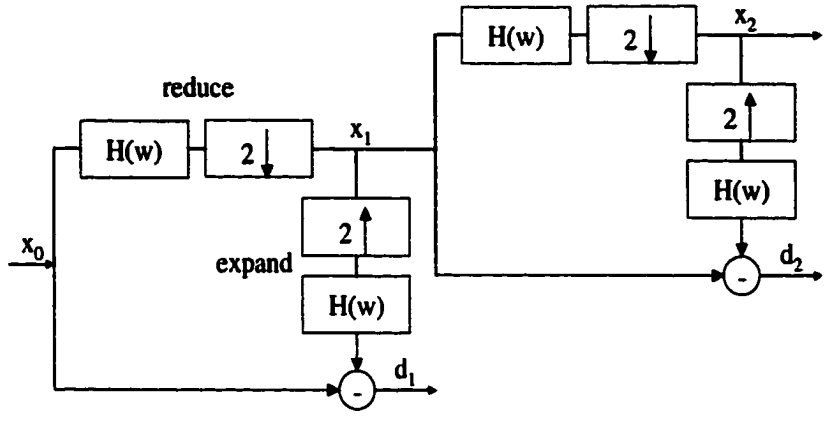


FIGURE 2.8. Laplacian pyramid

2.3.1 Laplacian pyramid

The Laplacian pyramid was first developed by Burt [Burt 1983] in order to find a fast calculation algorithm in performing convolutions with a set of projection functions which have many scales but identical shape. The projection functions of the Laplacian pyramid are Laplacian of Gaussian functions, which are the second derivatives of Gaussian functions (or Marr wavelets). However the pyramid is not computed by convolving the image directly with the projection functions. Instead the Laplacian pyramid is computed using two basic operations: *reduce* and *expand*. The reduce operation applies a low-pass filter and then subsamples by a factor of two in each dimension. The expand operation upsamples by a factor of two (zero padding) and then applies the same low-pass filter. A commonly used low-pass filter kernel is: $h = \frac{1}{16}(1, 4, 6, 4, 1)$. Finally, we get a collection of pyramid subband images consisting of several bandpass images and one leftover lowpass image.

There are two branches of the Laplacian pyramid in Fig. 2.8. The upper branch has a sequence of signals $\{x_0, x_1, ...x_N\}$ in which x_j is the blurred and downsampled version of x_{j-1} by the reduce operation, i.e., $x_j = \text{reduce}(x_{j-1})$. We next prove that

x_j can be expressed by a convolution between the input signal x_0 and a new low-pass filter ϕ_j, followed by downsampling by a factor of 2^j, i,e.,

$$x_j = \left[x_0 * \phi_j\right]_{\downarrow 2^j}, \tag{2.55}$$

where $\phi_1 = h$ and

$$\phi_j = [h]_{\uparrow 2^{j-1}} * \phi_{j-1} \text{ for } j > 1. \tag{2.56}$$

Proof: We first introduce two identities for multirate systems

$$[x]_{\downarrow 2} * h = \left[x * [h]_{\uparrow 2}\right]_{\downarrow 2}, \tag{2.57}$$

$$[x * h]_{\uparrow 2} = [x]_{\uparrow 2} * [h]_{\uparrow 2}. \tag{2.58}$$

These identities are very valuable in applications for efficient implementation of filters and filter banks, thus called the "noble identities" in [Vaidyanathan 1990]. Applying the "noble identities" in the *reduce* operation of x_{j+1} in the Laplacian pyramid, we have

$$\begin{aligned} x_{j+1} &= [x_j * h]_{\downarrow 2} \\ &= \left[[x_{j-1} * h]_{\downarrow 2} * h\right]_{\downarrow 2} \\ &= \left[\left[x_{j-1} * h * [h]_{\uparrow 2}\right]_{\downarrow 2}\right]_{\downarrow 2} \\ &= \left[x_{j-1} * h * [h]_{\uparrow 2}\right]_{\downarrow 2^2}. \end{aligned} \tag{2.59}$$

We can substitute the *reduce* operation of x_{j-1}, $x_{j-1} = [x_{j-2} * h]_{\downarrow 2}$, into the above equation and get

$$\begin{aligned} x_{j+1} &= \left[[x_{j-2} * h]_{\downarrow 2} * h * [h]_{\uparrow 2}\right]_{\downarrow 2^2} \\ &= \left[\left[x_{j-2} * h * \left[h * [h]_{\uparrow 2}\right]_{\uparrow 2}\right]_{\downarrow 2}\right]_{\downarrow 2^2} \\ &= \left[x_{j-2} * h * \left[h * [h]_{\uparrow 2}\right]_{\uparrow 2}\right]_{\downarrow 2^3} \\ &= \left[x_{j-2} * h * [h]_{\uparrow 2} * [h]_{\uparrow 2^2}\right]_{\downarrow 2^3}. \end{aligned} \tag{2.60}$$

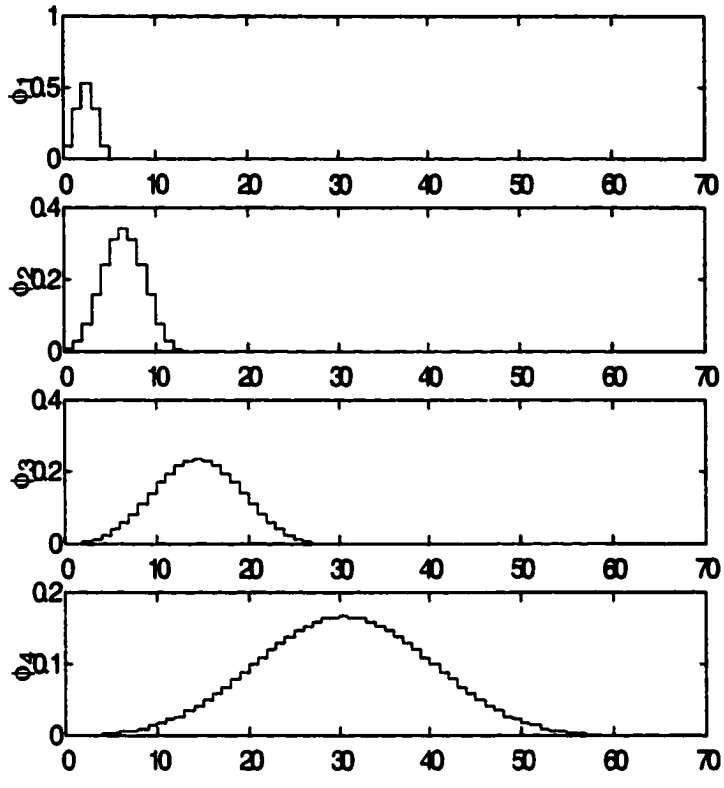


FIGURE 2.9. The equivalent low-pass filters ϕ_j for nodes in the upper branch of the Lalacian pyramid. Note that ϕ_j resemble the Gaussian functions and ϕ_j is nearly dialated by a factor of 2 from ϕ_{j-1}.

Repeating the above process from $j-3$ to 0, we have

$$x_{j+1} = \left[x_0 * h * [h]_{\uparrow 2} * [h]_{\uparrow 2^2} * \ldots * [h]_{\uparrow 2^j}\right]_{\downarrow 2^{j+1}}. \tag{2.61}$$

Defining $\phi_1 = h$ and $\phi_j = [h]_{\uparrow 2^{j-1}} * \phi_{j-1}$ for $j > 1$, it is not difficult to see that

$$x_j = \left[x_0 * \phi_j\right]_{\downarrow 2^j}. \tag{2.62}$$

In Fig. 2.9 we illustrate several low-pass filters of ϕ_j , which shows that these filters have the same Gaussian shapes but different scales.

The lower branch of the Laplacian pyramid in Fig. 2.8 has a collection of detail signals $\{d_1, ...d_N\}$, and

$$d_{j+1} = x_j - \text{expand}\,(x_{j+1}) \tag{2.63}$$

$$= \left[x_0 * \phi_j\right]_{\downarrow 2^j} - \left[\left[x_0 * \phi_{j+1}\right]_{\downarrow 2^{j+1}}\right]_{\uparrow 2} * h. \tag{2.64}$$

We cannot express d_{j+1} in terms of filtering and down-sampling operations exactly. However, since the low-pass filter h is used to interpolate between sample points after zero padding, we can approximate

$$\left[\left[x_0 * \phi_{j+1}\right]_{\downarrow 2^{j+1}}\right]_{\uparrow 2} * h \approx \left[x_0 * \phi_{j+1}\right]_{\downarrow 2^j}.$$

Hence

$$d_{j+1} \approx \left[x_0 * \left(\phi_j - \phi_{j+1}\right)\right]_{\downarrow 2^j}. \tag{2.65}$$

Therefore we can explain d_{j+1} as being calculated approximately by the convolution of the input signal x_0 and the difference of two adjacent low-pass filters, then down-sampled by 2^j. The difference of two Gaussian functions approximates the Laplacian of the Gaussian. Eq. (2.63) also gives the reconstruction method from $\{x_{j+1},\ d_{j+1}\}$ to x_j by

$$x_j = d_{j+1} + \text{expand}\,(x_{j+1}). \tag{2.66}$$

Now we have discussed the projection functions of the Laplacian pyramid, which are (1) the low-pass filters ϕ_j, and (2) the difference of two adjacent low-pass filters centered at some spatial locations. We can express the coefficients of the Laplacian pyramid as inner products of the input signal and these projection functions. The basis functions of the Laplacian pyramid are generated by setting one sample of one subband image equal to 1.0 (and all others to zero) and reconstructing. In Fig. 2.10 we illustrate the basis functions of the Laplacian pyramid, which are not the same as its projection functions, so the Laplacian pyramid is not a tight frame. In Fig. 2.11 we show an example of the Laplacian pyramid for a disc image.

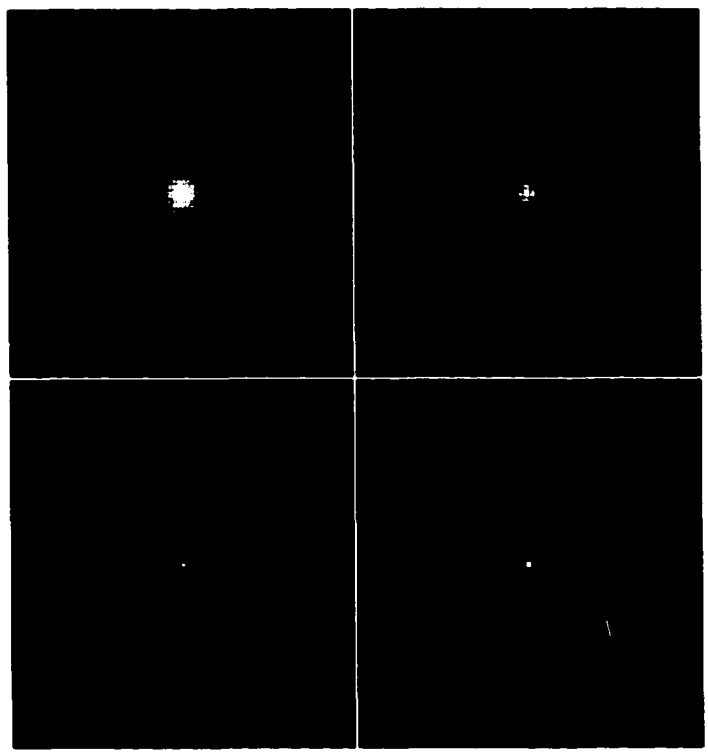


FIGURE 2.10. Basis functions of the Laplacian pyramid resemble Gaussian functions.

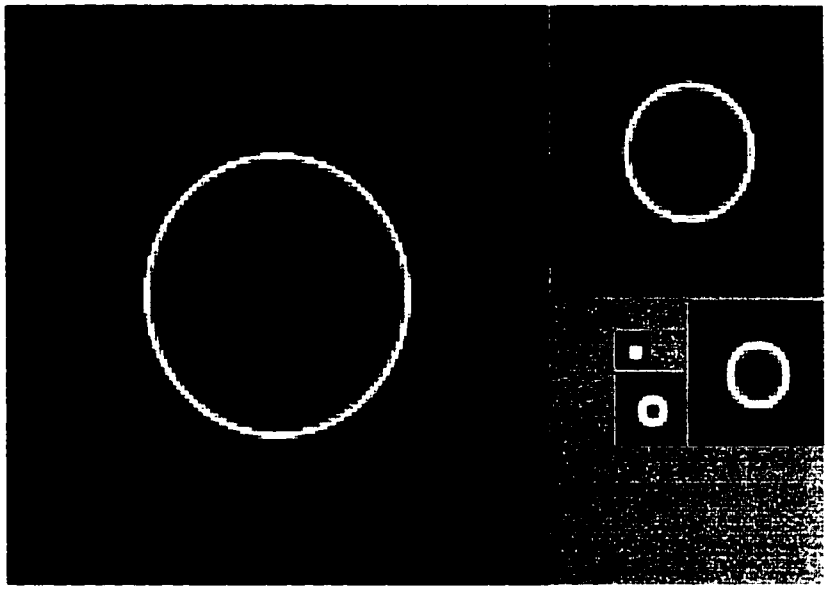


FIGURE 2.11. The 4-level Laplacian pyramid of a disc. The sequence of subband images d_1, d_2, d_3, d_4, x_4 is arranged by size.

2.3.2 Steerable pyramid

Since the Laplacian pyramid has radially symmetric basis functions, it cannot capture oriented and elongated structures in textures. Therefore, we may use the steerable pyramid transform [Simoncelli 1995] for anisotropic textures. Like the Laplacian pyramid, this transform decomposes the image into several spatial-frequency bands. Then, it further divides each frequency band into a set of orientation bands. In addition to having steerable orientation subbands, the transform is designed to be "self-inverting" (the transform is a tight frame), and the basis functions are localized in space and spatial-frequency.

The block diagram for the decomposition is shown in Fig. 2.12. Initially, the image is separated into low-pass and high-pass subbands, using filters L_0 and H_0. The low-pass subband is then divided into a set of oriented bandpass subbands and a lower-pass subband, using filters B_k, $k = 0, ..., K-1$ and L_1. This lower-pass subband is downsampled by a factor of 2 in the X and Y axes. The oriented bandpass filters

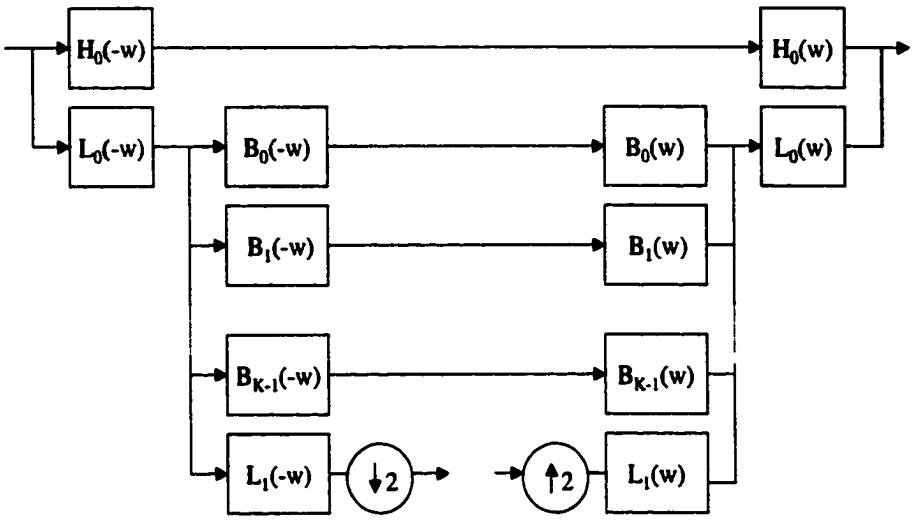


FIGURE 2.12. Steerable pyramid

used in this transformation are polar-separable in the Fourier domain, where they may be written as

$$B_k(r,\theta) = H(r)\,G_k(\theta)\,,\ k = 0, ..., K-1, \tag{2.67}$$

where r and θ are polar frequency coordinates, i.e., $w_x = r\cos\theta$ and $w_y = r\sin\theta$. Fig. 2.13 contains a diagram of the idealized frequency response of the subbands, for $K = 4$. The radial and angular parts of the oriented bandpass filters are

$$H(r) = \begin{cases} \cos\left(\frac{\pi}{2}\log_2\left(\frac{2r}{\pi}\right)\right) & \text{for } \frac{\pi}{4} < r < \frac{\pi}{2} \\ 1 & \text{for } r > \frac{\pi}{2} \\ 0 & \text{for } r \le \frac{\pi}{4} \end{cases}, \tag{2.68}$$

$$G_k(\theta) = \begin{cases} \left[\cos\left(\theta - \frac{\pi k}{K}\right)\right]^{K-1} & \text{for } \left|\theta - \frac{\pi k}{K}\right| < \frac{\pi}{2} \\ 0 & \text{for } \left|\theta - \frac{\pi k}{K}\right| \ge \frac{\pi}{2} \end{cases}. \tag{2.69}$$

The lower-pass filter is

$$L_1(r) = \begin{cases} 2\cos\left(\frac{\pi}{2}\log_2\left(\frac{4r}{\pi}\right)\right) & \text{for } \frac{\pi}{4} < r < \frac{\pi}{2} \\ 2 & \text{for } r \le \frac{\pi}{4} \\ 0 & \text{for } r > \frac{\pi}{2} \end{cases}. \tag{2.70}$$

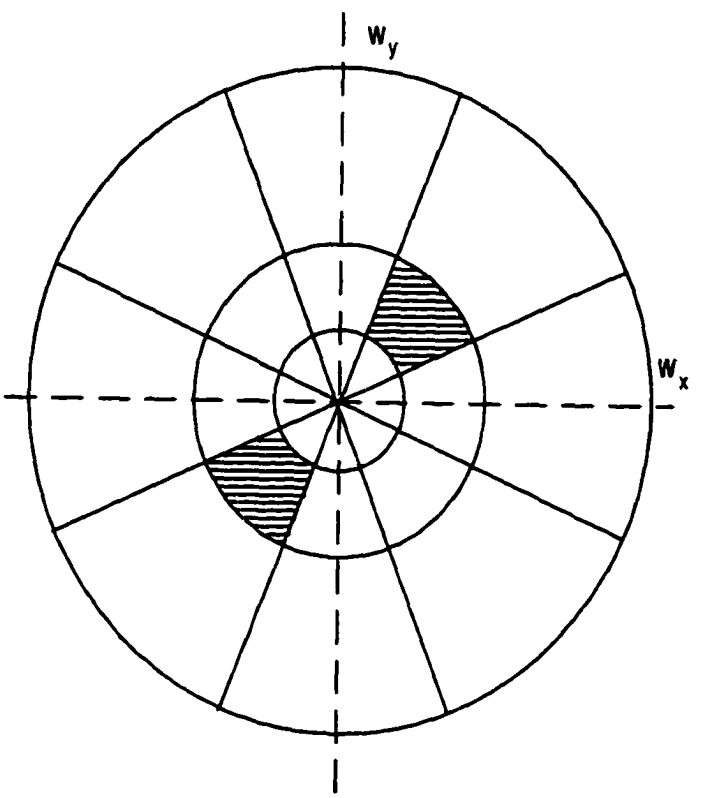


FIGURE 2.13. Illustration of the spectral decomposition performed by a steerable pyramid with $K = 4$ orientation bands. Frequency axes range from $-\pi$ to π. The shaded region corresponds to the spectral support of a single subband.

The lower-pass filter is bandlimited within $\frac{\pi}{2}$ in order to prevent aliasing after downsampling by a factor of 2. The recursive procedure is initialized by splitting the input image into low-pass and high-pass portions, using the following filters:

$$L_0(r) = \frac{1}{2}L_1\left(\frac{r}{2}\right), \tag{2.71}$$

$$H_0(r) = H(\frac{r}{2}). \tag{2.72}$$

If we use $N = 3$ pyramid levels and $K = 4$ orientation bands then we have 12 bandpass oriented filters, one low-pass filter and one high-pass filter. It is not difficult to verify that the filter bank satisfies

$$|H_0(\mathbf{w})|^2 + |L_0(\mathbf{w})|^2 \left[\sum_{k=0}^{K-1} |B_k(\mathbf{w})|^2 + |L_1(\mathbf{w})|^2\right] = 1, \tag{2.73}$$

where $\mathbf{w} = (w_x, w_y)$ are continuous two-dimensional frequency variables. Therefore the filter bank provides a full coverage of the frequency domain.

The above discrete filters are designed using weighted least-square techniques in the Fourier domain to approximately fit the constraints detailed above. The resulting filters are fairly compact (typically 9×9 taps), thus spatially localized. Since the steerable pyramid is self-inverting, i.e, a tight frame, the projection functions are the same as the basis functions. We can generate the basis functions by setting one sample of one subband equal to 1.0 (and all others to zero) and reconstructing. We illustrate the basis functions of the steerable pyramid in Fig. 2.14, and an example of the steerable pyramid for a disc image in Fig. 2.15.

2.3.3 Dyadic wavelet transform

In some applications, we require all of the subbands to have the same sampling rate as the input image. For example, we want to get the set of subband images of the steerable pyramid without downsampling. It is not clear if we can still reconstruct the input image from these subbands without downsampling. To answer this question, we

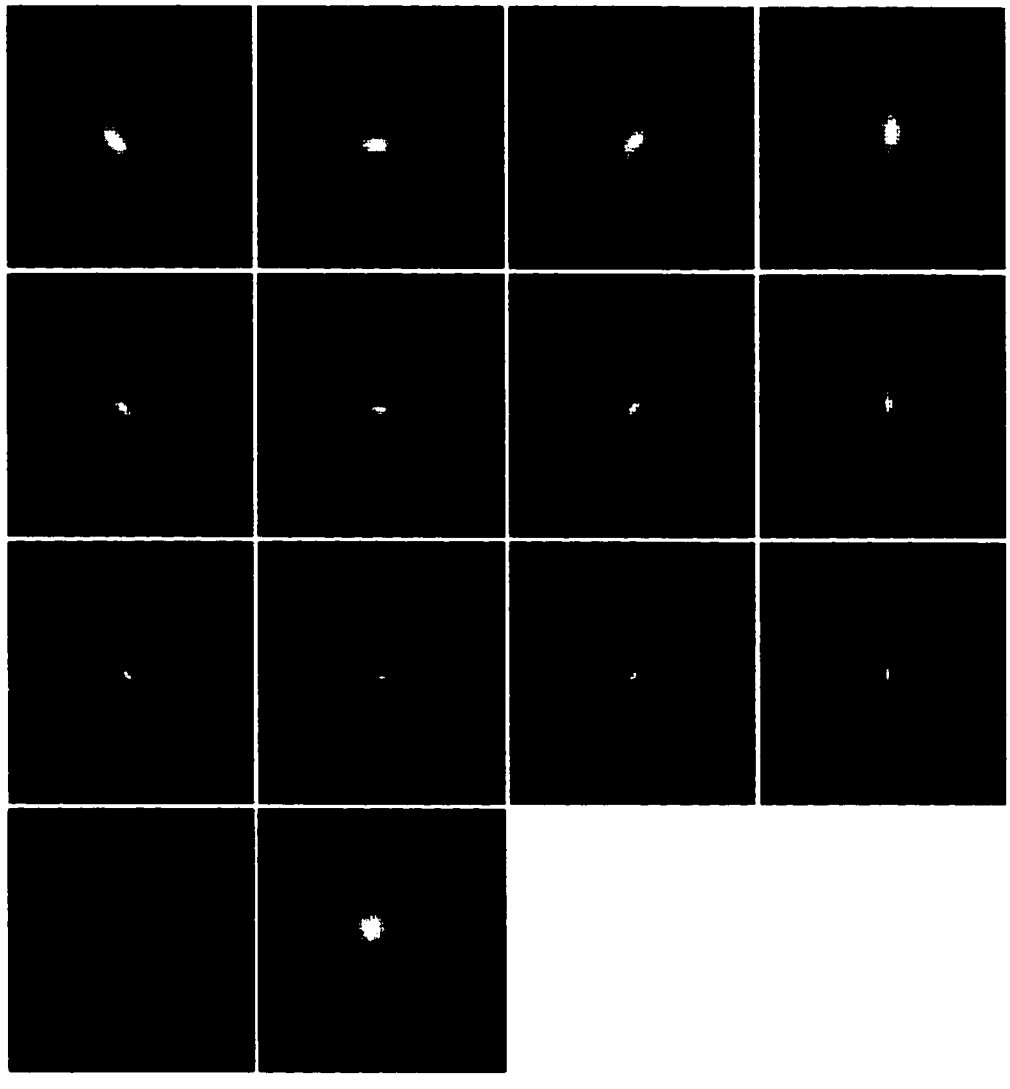


FIGURE 2.14. Basis and projection functions of the steerable pyramid.

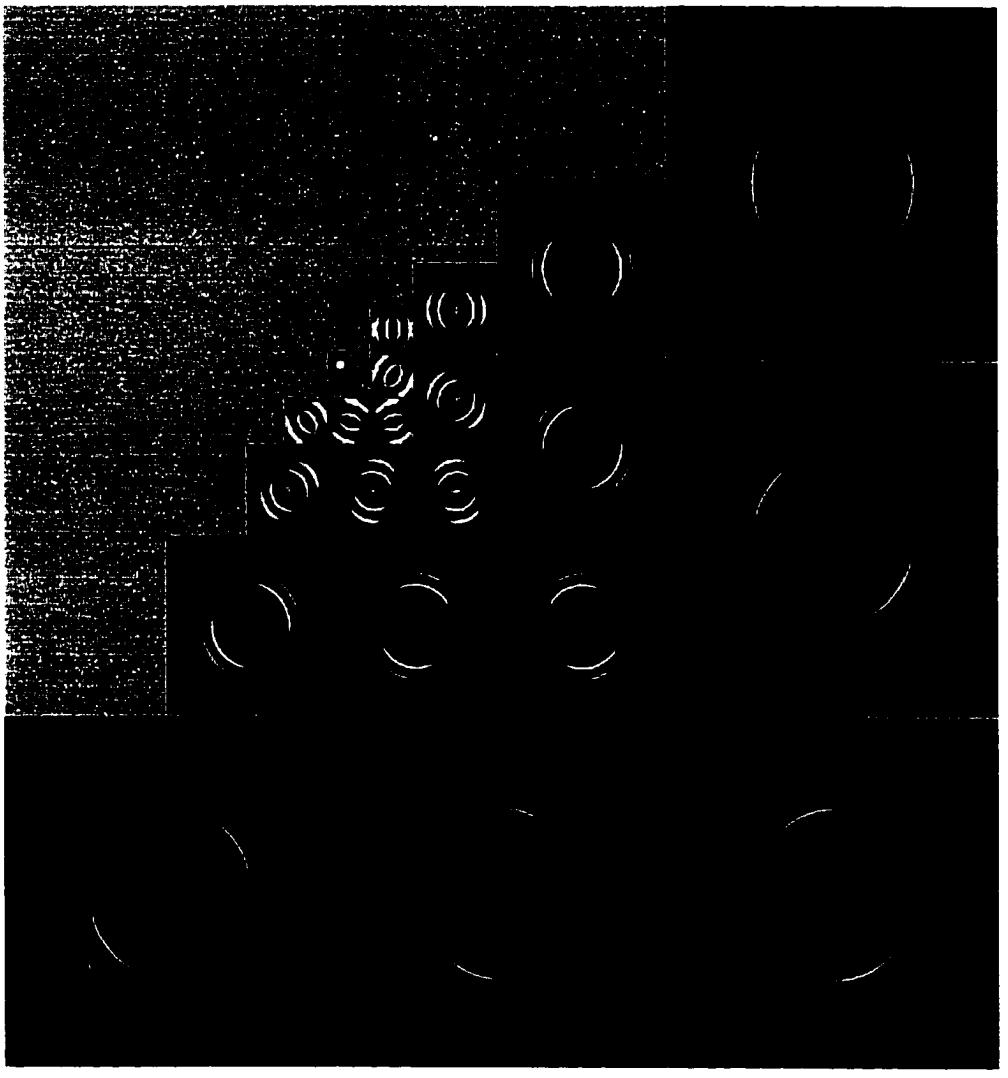


FIGURE 2.15. The steerable pyramid of a disc. We use $N = 4$ pyramid levels and $K = 5$ orientation bands.

will discuss the dyadic wavelet transform, which is another type of frame. However, the dyadic wavelet transform was originally introduced for overcoming a limitation of the DWT; that is, the DWT is not translation-invariant [Mallat 1998]. If the signal is shifted by just one pixel, then the discrete wavelet coefficients will change significantly. This behavior causes difficulty in texture analysis and synthesis since textures usually have a translation-invariant or stationary property.

Different from the orthonormal wavelet basis functions in (2.2) and (2.3), the dyadic wavelets are defined by discrete filters at all of the spatial locations [Unser 1995], which leads to the simple decomposition algorithm as follows:

$$c_I(k) = h_I(-k) * x(k), \tag{2.74}$$

$$d_i(k) = g_i(-k) * x(k), \ i = 1, ..., I. \tag{2.75}$$

Thus each subband image has the same sampling rate as the input image. It has been proved that if the frequency axis is completely covered by the dyadic wavelets, i.e.,

$$0 < A \leq |H_I(w)|^2 + \sum_{i=1}^{I} |G_i(w)|^2 \leq B < \infty, \tag{2.76}$$

then the frame condition (2.49) is satisfied and the dyadic wavelet transform is a complete and stable representation of the signal [Mallat 1998]. Here $H_I(w)$ and $G_i(w)$ are the DTFTs of the filter sequences. Moreover, if

$$|H_I(w)|^2 + \sum_{i=1}^{I} |G_i(w)|^2 = 1, \tag{2.77}$$

then it is a tight frame, and the Parseval's theorem (energy conservation) exists:

$$\|c_I(k)\|^2 + \sum_{i=1}^{I} \|d_i(k)\|^2 = \|x(k)\|^2.$$

The signal in l_2 space can be expressed by

$$x(k) = h_I(k) * c_I(k) + \sum_{i=1}^{I} g_i(k) * d_i(k). \tag{2.78}$$

We know from (2.73) and (2.10) that both the steerable pyramid basis functions and the orthonormal wavelet basis functions can be used as the dyadic wavelets. It should be noted that the dyadic wavelets are not orthonormal bases (but a tight frame) even though $A = B = 1$. This disagreement with the discussion in the beginning of section 2.3 is due to the fact that the dyadic wavelets are not normalized.

Eqs. (2.74), (2.75) and (2.78) show that the decomposition and synthesis of signals can be implemented by linear convolutions. In practice, the input signal $x(k)$ has a finite size of N samples, hence the linear convolutions are replaced by circular convolutions. We may use the the FFT to calculate the circular convolution in order to speed up the computation. The FFT algorithm has a complexity of $O(N \log_2 N)$. In this case the following condition needs to be satisfied

$$\left|\widetilde{H}_I(n)\right|^2 + \sum_{i=1}^{I} \left|\widetilde{G}_i(n)\right|^2 = 1, \; n = 0, 1, ..., N-1, \tag{2.79}$$

where $\widetilde{H}_I(n)$ and $\widetilde{G}_i(n)$ are the DFTs of the filter sequences, and the length of filter sequences is usually increased to the length of signals, N, by zero padding. We illustrate the DWF coefficients of Lena using Daubechies-4 wavelet basis functions in Fig. 2.16 and the DWF coefficients of a disc using the steerable pyramid basis functions in Fig. 2.17.

If the two-scale relations for the discrete filters, i.e., (2.6) and (2.7) still exist (for example, the orthonormal wavelets), we will have a fast filter-bank algorithm for the dyadic wavelet transform as follows:

$$\begin{aligned} c_{i+1}(l) &= \sum_k x(k)\, h_{i+1}(k-l) \\ &= \sum_k x(k) \left\{ [h]_{\uparrow 2^i} * h_i(k-l) \right\} \\ &= [h]_{\uparrow 2^i} * \sum_k x(k)\, h_i(k-l) \\ &= [h]_{\uparrow 2^i} * c_i(l), \; l = 0, 1, ..., N-1. \end{aligned} \tag{2.80}$$

$$\begin{aligned} d_{i+1}(l) &= \sum_k x(k)\, g_{i+1}(k-l) \\ &= \sum_k x(k) \left\{ [g]_{\uparrow 2^i} * h_i(k-l) \right\} \\ &= [g]_{\uparrow 2^i} * \sum_k x(k)\, h_i(k-l) \\ &= [g]_{\uparrow 2^i} * c_i(l)\,, \quad l = 0, 1, ..., N-1. \end{aligned} \tag{2.81}$$

We can also use a similar procedure in the synthesis step as shown in the following

$$c_i(l) = [h]_{\uparrow 2^i} * c_{i+1}(l) + [g]_{\uparrow 2^i} * d_{i+1}(l)\,. \tag{2.82}$$

Applying the DTFT, we can express (2.80) and (2.81) in the frequency domain by

$$c_{i+1}(w) = H\left(2^i w\right) c_i(w)\,, \tag{2.83}$$

$$d_{i+1}(w) = G\left(2^i w\right) c_i(w)\,, \tag{2.84}$$

and express (2.82) in the frequency domain by

$$c_i(w) = H\left(2^i w\right) c_{i+1}(w) + G\left(2^i w\right) d_{i+1}(w)\,. \tag{2.85}$$

When the signal has a finite size of samples, say N, we will use the circular convolution in implementing the filter-bank algorithm. This means we will use periodic extensions for dealing with the border problem. Suppose that h and g have respectively K_h and K_g non-zero filter coefficients, we can see from ( ) and (2.81) the number of multiplications needed to compute c_{i+1} and d_{i+1} from c_i is equal to $(K_h + K_g)\, N$. If K_h and K_g are smaller than $\log_2 N$ when N is large, then the filter-bank algorithm is faster than the FFT.

Such an overcomplete analysis can be extended to the biorthogonal case. In this more general situation, we lose the energy conservation property but have more freedom in the choice of h and g.

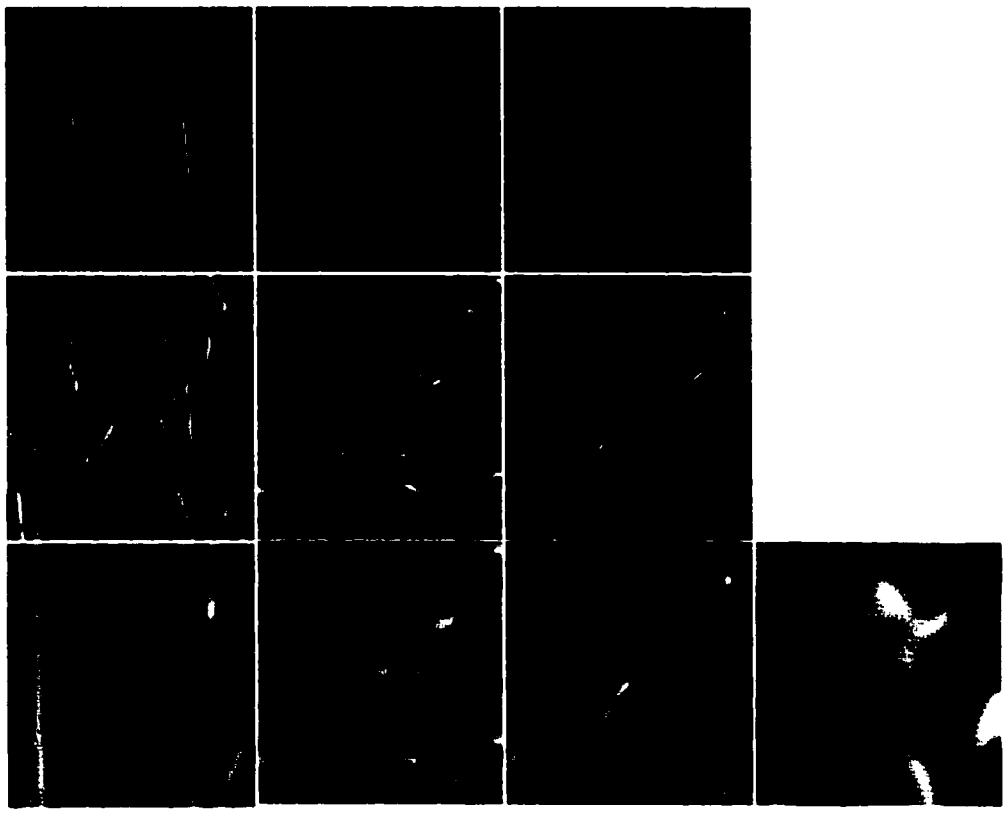


FIGURE 2.16. DWF of Lenna using Daubechies-4 wavelet basis functions. There are three octave bandpass images (LH, HL, HH) with the same resolution in each row and one smooth version of the image (LL) in the right-bottom corner.

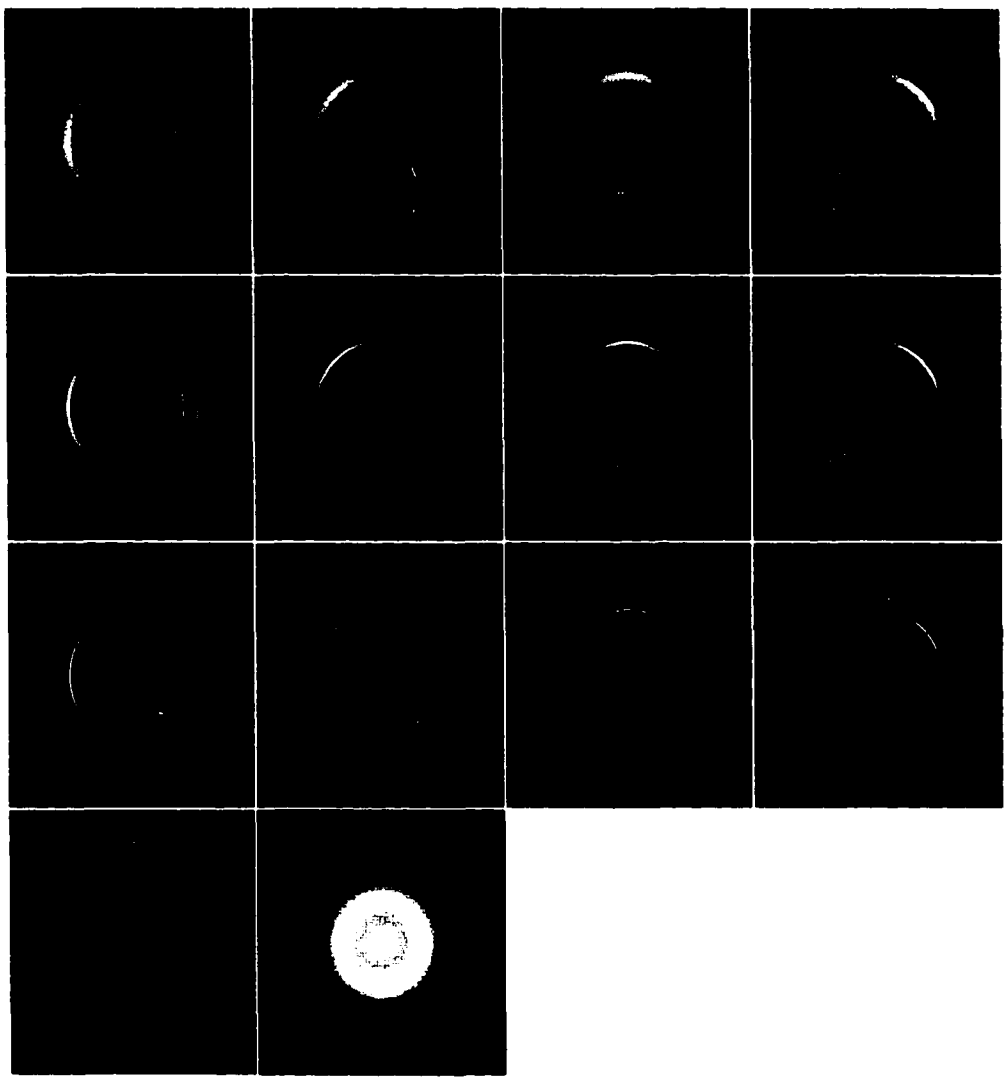


FIGURE 2.17. DWF of a disc using the steerable pyramid basis functions. There are four octave bandpass images with the same resolution in each row. The fourth row shows one smooth version of the image and one highe-passed image.

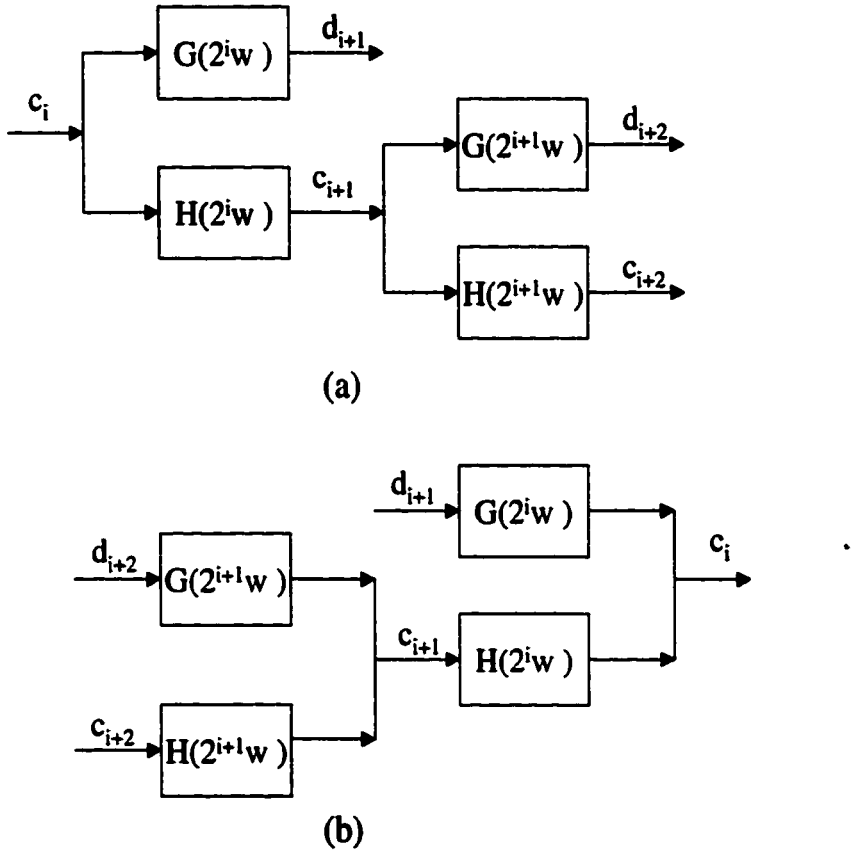


FIGURE 2.18. (a) Analysis filter bank of DWF, (b) Synthesis filter bank of DWF

2.4 Independent component analysis

Now we introduce another type of invertible linear transform, called independent component analysis (ICA). The goal of ICA is to recover independent sources given only observations that are unknown linear mixtures of the unobserved independent source signals. In contrast to correlation-based transformations such as principal component analysis (PCA), ICA not only decorrelates the signals but also reduces the high-order statistical dependences, attempting to make the signals as independent of each other as possible. In short, the goal of ICA is to find an invertible linear transform $\mathbf{W}$ of the random vector $\mathbf{x}$ such that

$$\mathbf{u} = \mathbf{W}\mathbf{x}, \tag{2.86}$$

and the components of $\mathbf{u}$ are independent of each other [Hyvarinen 1999], [Lee 1999], [Nandi 1999]. Different from the previous methods, $\mathbf{W}$ is not fixed and depends on the data sets. Thus one may see ICA as a way of choosing a basis which is custom-tailored to the sampling data.

A natural measure of the independence between the components of the random vector $\mathbf{u}$ is the Kullback divergence between $p(\mathbf{u})$ and $\prod p(u_i)$, which is also called the mutual information of $\mathbf{u}$:

$$I(\mathbf{u}) = \int p(\mathbf{u}) \log \frac{p(\mathbf{u})}{\prod p(u_i)} d\mathbf{u}. \tag{2.87}$$

The mutual information is always non-negative, and the zero value is reached when the components of $\mathbf{u}$ are statistically independent. The matrix $\mathbf{W}$ is determined so that the mutual information of the transformed components are minimized. From (2.87), $I(\mathbf{u})$ can also be expressed as

$$\begin{aligned} I(\mathbf{u}) &= \int p(\mathbf{u}) \log p(\mathbf{u})\, d\mathbf{u} - \sum_{i=1}^{N} \int p(\mathbf{u}) \log p(u_i)\, du_i \\ &= \sum_{i=1}^{N} H(u_i) - H(\mathbf{u}), \end{aligned} \tag{2.88}$$

where $H(\mathbf{u})$ is called the differential entropy, and defined by

$$H(\mathbf{u}) = -\int p(\mathbf{u}) \log p(\mathbf{u})\, d\mathbf{u}. \tag{2.89}$$

By the above definition of differential entropy, we can see that

$$H(\mathbf{Au}) = H(\mathbf{u}) + \log|\det(\mathbf{A})|. \tag{2.90}$$

To prove this property, we first define a new random vector $\mathbf{y} = \mathbf{Au}$. If $\mathbf{A}$ is an invertible transform, then we have

$$p(\mathbf{y}) = \frac{1}{|\det(\mathbf{A})|} p(\mathbf{u}). \tag{2.91}$$

The differential entropy of $\mathbf{y}$ is

$$H(\mathbf{y}) = -\int p(\mathbf{y}) \log p(\mathbf{y})\, d\mathbf{y}. \tag{2.92}$$

Changing the variable of the above integral, we get

$$\begin{aligned} H(\mathbf{Au}) &= -\int p(\mathbf{u}) \left[-\log|\det(\mathbf{A})| + \log p(\mathbf{u})\right] d\mathbf{u} \\ &= -\int p(\mathbf{u}) \log p(\mathbf{u})\, d\mathbf{u} + \log|\det(\mathbf{A})|. \end{aligned} \tag{2.93}$$

Therefore, $H(\mathbf{u})$ is not invariant by invertible transforms, but only by orthonormal transforms since $|\det(\mathbf{A})| = 1$ when $\mathbf{A}$ is an orthonormal matrix.

We here introduce two ICA methods. One is based on the higher-order cumulants (usually $3^{th}$and 4^{th}-order statistics), and the other directly minimizes the mutual information by using the stochastic gradient method. Both methods drive the distribution of the transformed vector away from Gaussian distributions, but the direct minimization of the mutual information provides statistics higher than fourth-order necessary to make the data as independent as possible. On the other hand, the direct minimization method requires the prior distribution of $\mathbf{u}$, and the higher-order cumulants method does not need this requirement.

2.4.1 Higher-order cumulants method

We divide the procedure for finding $\mathbf{W}$ into two steps. The first step is standardization, that is, transform a random vector $\mathbf{x}$ into another random vector $\tilde{\mathbf{x}}$ by a decorrelating matrix $\tilde{\mathbf{x}} = \mathbf{V}\mathbf{x}$ such that its components are uncorrelated and have unit variances. That is,

$$\mathbf{K}_{\tilde{\mathbf{x}}} = \mathbf{I} = \mathbf{V}\mathbf{K}_{\mathbf{x}}\mathbf{V}^T, \tag{2.94}$$

where $\mathbf{I}$ is an identity matrix. In the second step, a further transformation $\mathbf{u} = \mathbf{Q}\tilde{\mathbf{x}}$ using high-order correlations is required to reduce the remaining redundancy within the vector for non-Gaussian sources. In order to let the components of $\mathbf{u}$ be uncorrelated and have unit variances,

$$\mathbf{K}_{\mathbf{u}} = \mathbf{Q}\mathbf{K}_{\tilde{\mathbf{x}}}\mathbf{Q}^T = \mathbf{Q}\mathbf{Q}^T = \mathbf{I}, \tag{2.95}$$

$\mathbf{Q}$ must be an orthonormal matrix; and is determined so that $I(\mathbf{u})$ is minimized. Finally, the transform matrix is $\mathbf{W} = \mathbf{Q}\mathbf{V}$.

Since the differential entropy is invariant by orthonormal transforms, (2.88) shows that minimizing the mutual information $I(\mathbf{u})$ will also minimize the sum of the marginal differential entropy $\sum_{i=1}^{N} H(u_i)$. Thus we have simplified the N-dimensional problem of minimizing mutual information to the separate minimization of N 1-D marginal differential entropies.

In order to estimate the marginal differential entropy, $H(u_i)$, from samples of u_i, Comon [Comon 1994] used the Edgeworth expansion for the non-Gaussian marginal density in terms of the Gaussian density (with the same mean and variance) and the Hermite polynomials. This construction leads to the use of higher-order cumulants,

like kurtosis. The Edgeworth expansion for $p(u_i)$ is [Kendall 1977]

$$\begin{aligned} p(u_i) \approx\ & p_G(u_i)\,[1 + \frac{1}{3!}k_3 h_3(u_i) + \frac{1}{4!}k_4 h_4(u_i) + \frac{10}{6!}k_4^2 h_6(u_i) \\ & + \frac{1}{5!}k_5 h_5(u_i) + \frac{35}{7!}k_3 k_4 h_7(u_i) + \frac{280}{9!}k_3^3 h_9(u_i) \\ & + \frac{1}{6!}k_6 h_6(u_i) + \frac{56}{8!}k_3 k_5 h_8(u_i) + \frac{35}{8!}k_4^2 h_8(u_i) \\ & + \frac{2100}{10!}k_3^2 k_4 h_{10}(u_i) + \frac{15400}{12!}k_3^4 h_{12}(u_i), \end{aligned} \tag{2.96}$$

where $p_G(u_i)$ denotes the Gaussian density with the same mean and variance as $p(u_i)$. The cumulants k_n are coefficients related to the form of the p.d.f. of u_i as

$$\int_{-\infty}^{\infty} \exp(\beta u)\, p(u)\, du = \exp\left(\sum_{n=1}^{\infty} \frac{1}{n!} k_n \beta^n\right) \text{ for } \forall \beta. \tag{2.97}$$

The terms $h_k(u_i)$ are the orthogonal Hermite polynomials defined by the recursion

$$\begin{aligned} h_0(u_i) &= 1, \qquad h_1(u_i) = u, \\ h_{k+1}(u_i) &= u h_k(u_i) - \frac{\partial}{\partial u_i} h_k(u_i). \end{aligned} \tag{2.98}$$

After substituting the Edgeworth expansion for $p(u_i)$ into the marginal entropy and making use of the properties of the orthogonal Hermite polynomials, $H(u_i)$ becomes

$$H(u_i) = H_G(u_i) - \left(\frac{1}{12}k_3^2(i) + \frac{1}{48}k_4^2(i) + \frac{7}{48}k_3^4(i) + \frac{1}{12}k_3^2(i)\,k_4(i)\right). \tag{2.99}$$

The third-order cumulant $k_3(i) = E\{u_i^3\}$ is called the skewness and the fourth-order cumulant $k_4(i) = E\{u_i^4\} - 3$ is called the kurtosis. We can easily estimate the third-order and the fourth-order cumulants from samples of u_i. If we make the assumption that the p.d.f. of u_i is approximately symmetric, then the third-order cumulant will be approximately zero.

The mutual information is now approximated by

$$I(\mathbf{u}) = \sum_{i=1}^{N} H_G(u_i) - \frac{1}{48}\sum_{i=1}^{N} k_4^2(i) - H(\mathbf{u}). \tag{2.100}$$

The first term $\sum_{i=1}^{N} H_G(u_i)$ is the sum of the marginal entropies for the standardized Gaussian random variables, and $\sum_{i=1}^{N} H_G(u_i) = \frac{N}{2}[1 + \log(2\pi)]$. The last term $H(\mathbf{u})$ equals $H(\tilde{\mathbf{x}})$ since the differential entropy is invariant by orthonormal transforms. Therefore, minimizing the mutual information $I(\mathbf{u})$ is equivalent to maximizing $\sum_{i=1}^{N} k_4^2(i)$, and Comon proposed the following contrast function

$$\Phi_{\max} = \sum_{i=1}^{N} k_4^2(i). \tag{2.101}$$

Like the Jacobi algorithm in the diagonalization of symmetric real matrices, Comon's algorithm processes each pair of $\mathbf{u}$ components in turn. For any pair of random variables u_i and u_j in each iteration, a rotational matrix $\mathbf{Q}_{ij}$ is determined to maximize the contrast function $k_4^2(i) + k_4^2(j)$, and the whole orthonormal transform is given by

$$\mathbf{Q} = \prod_{i,j} \mathbf{Q}_{ij}. \tag{2.102}$$

2.4.2 Direct mutual information minimization

The mutual information can be minimized directly by using the stochastic gradient ascent method. The gradient of $-I(\mathbf{u})$ is

$$\begin{aligned} \frac{\partial}{\partial \mathbf{W}}(-I(\mathbf{u})) &= \frac{\partial}{\partial \mathbf{W}}\left(\int p(\mathbf{u}) \log\left(\frac{\prod_{i=1}^{N} p(u_i)}{p(\mathbf{u})}\right) d\mathbf{u}\right) \\ &= \frac{\partial}{\partial \mathbf{W}}\left(E\left\{\log\left(\prod_{i=1}^{N} p(u_i)\right)\right\} + H(\mathbf{u})\right). \end{aligned} \tag{2.103}$$

Since $\mathbf{u} = \mathbf{W}\mathbf{x}$, we have

$$H(\mathbf{u}) = H(\mathbf{x}) + \log|\det(\mathbf{W})|. \tag{2.104}$$

Substituting $H(\mathbf{u})$ into (2.103), we have

$$\frac{\partial}{\partial \mathbf{W}}(-I(\mathbf{u})) = \frac{\partial}{\partial \mathbf{W}}\left(E\left\{\log\left(\prod_{i=1}^{N} p(u_i)\right)\right\} + \log|\det(\mathbf{W})| + H(\mathbf{x})\right). \tag{2.105}$$

Since the differential entropy of the input vector, $H(\mathbf{x})$, is not a function of $\mathbf{W}$, $\frac{\partial}{\partial \mathbf{W}}(H(\mathbf{x})) = 0$, the stochastic gradient is

$$\frac{\partial}{\partial \mathbf{W}}(-I(\mathbf{u})) = \frac{\partial}{\partial \mathbf{W}}\left(\sum_{i=1}^{N} \log(p(u_i))\right) + \frac{\partial}{\partial \mathbf{W}}(\log|\det(\mathbf{W})|). \tag{2.106}$$

For the first term in (2.106),

$$\frac{\partial}{\partial \mathbf{W}}\left(\sum_{i=1}^{N} \log(p(u_i))\right) = -\varphi(\mathbf{u})\,\mathbf{x}^T, \tag{2.107}$$

where $\varphi(\mathbf{u})$ is the gradient vector of the log-likelihood

$$\varphi(\mathbf{u}) = -\frac{\frac{\partial p(\mathbf{u})}{\partial \mathbf{u}}}{p(\mathbf{u})} = \left[-\frac{\frac{\partial p(u_1)}{\partial u_1}}{p(u_1)}, \ldots, -\frac{\frac{\partial p(u_N)}{\partial u_N}}{p(u_N)}\right]^T. \tag{2.108}$$

For the second term in (2.106),

$$\frac{\partial}{\partial \mathbf{W}}(\log|\det(\mathbf{W})|) = \left(\mathbf{W}^{-1}\right)^T. \tag{2.109}$$

Therefore, the general learning rule is

$$\frac{\partial}{\partial \mathbf{W}}(-I(\mathbf{u})) = \left[\mathbf{W}^{-T} - \varphi(\mathbf{u})\,\mathbf{x}^T\right]. \tag{2.110}$$

In order to calculate the stochastic gradient in an efficient way, Amari et al. proposed a modification of this rule, which utilizes the *natural* gradient rather than the *absolute* gradient [Amari 1996]. It amounts to multiplying the absolute gradient by $\mathbf{W}^T\mathbf{W}$, giving the following rule

$$\begin{aligned}\frac{\partial}{\partial \mathbf{W}}(-I(\mathbf{u})) &= \left[\mathbf{W}^{-T} - \varphi(\mathbf{u})\,\mathbf{x}^T\right]\mathbf{W}^T\mathbf{W} \\ &= \left[\mathbf{I} - \varphi(\mathbf{u})\,\mathbf{u}^T\right]\mathbf{W}.\end{aligned} \tag{2.111}$$

This learning rule can be derived from several theoretical view points such as maximum likelihood estimation [Cardoso 1997], [Pearlmutter 1996], entropy maximization [Bell 1995], [Bell 1997] and negentropy maximization [Girolami 1997a]. In (2.111),

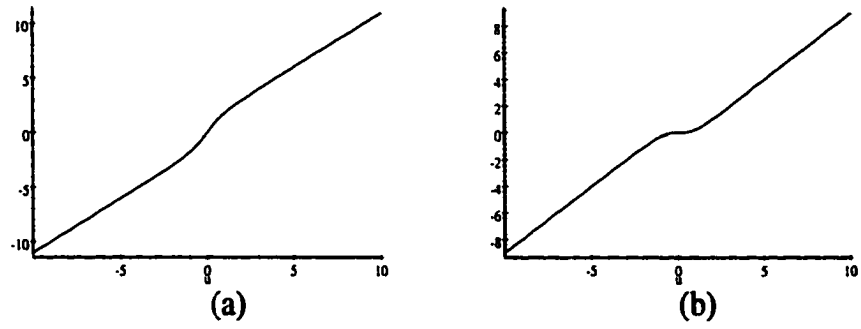


FIGURE 2.19. (a) $\varphi(u) = u + \tanh(u)$ (b) $\varphi(u) = u - \tanh(u)$

$\varphi(\mathbf{u})$ is selected according to a prior distribution of sources. In [Girolami 1997b] a parametric density model is employed for sub- and super- Gaussian sources, resulting in

$$\varphi(u) = u + \tanh(u) \text{ for super- Gaussian,} \tag{2.112}$$

$$= u - \tanh(u) \text{ for sub- Gaussian.} \tag{2.113}$$

Now we can estimate the ICA matrix by using the stochastic gradient ascent method. Assuming that we have a set of training vectors $\{\mathbf{x}(1), \mathbf{x}(2), ..., \mathbf{x}(N)\}$, we denote the $\mathbf{W}$ matrix at the n^{th} iteration by $\mathbf{W}(n)$ and $\mathbf{u}(n) = \mathbf{W}(n)\mathbf{x}(n)$. Then the ICA matrix is updated by the following algorithm

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \mu \left[\mathbf{I} - \varphi(\mathbf{u}(n))\,\mathbf{u}(n)^T\right] \mathbf{W}(n), \tag{2.114}$$

where μ is the learning rate. Bell and Sejnowski [Bell 1997] used this learning rule for a set of natural images involving trees, leaves and so on. The training set, $\{\mathbf{x}\}$, was then generated from 17,595 12×12 samples from the images. They found that the ICA filters (or the columns of $\mathbf{W}$) were localized and oriented, resembling the edge filters or the wavelets. Fig. 2.20 shows a typical example of such a basis.

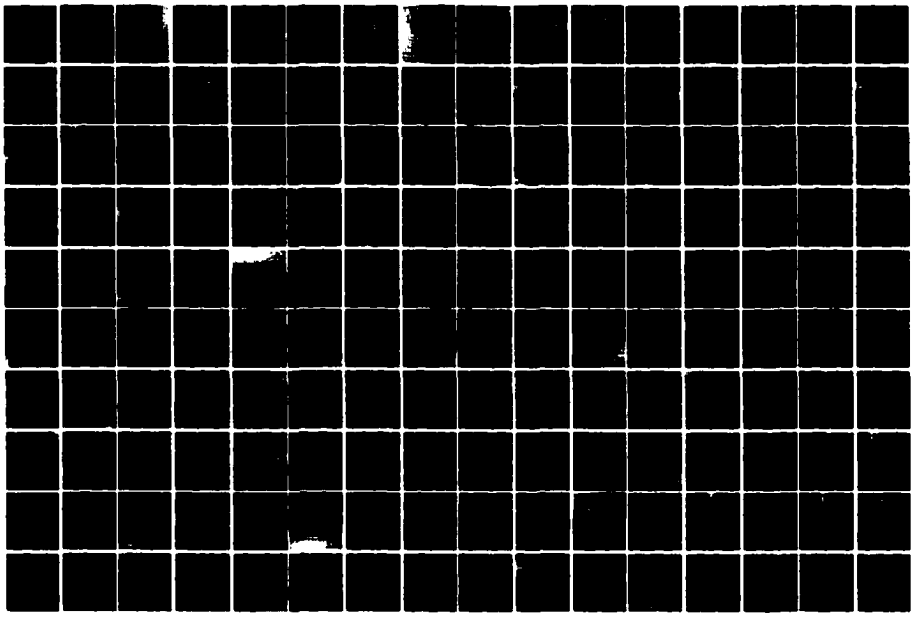


FIGURE 2.20. ICA filters of the natural images.

Chapter 3

Statistical Texture Synthesis and Analysis

3.1 Introduction

Performance for human visual detection and computer model observers are specific to the statistical properties of the images. It is for this reason that most observer studies and development and testing of computed-aided diagnosis algorithms are done using samples of real medical images pertinent to the imaging modality and anatomy. However, one limitation of using real medical images is that there might be access to only a limited number of samples, and the statistical properties of the images might be difficult to fully characterize. Within this context, developing a method to model and synthesize realistic textures will facilitate observer studies. On the other hand, the fact that we can synthesize texture images does not mean that we know the full statistics of them; therefore, developing a method to analyze the statistics of a set of training images with the same distribution is necessary in designing computer model observers. This chapter considers the modeling, synthesis and analysis of the texture images.

Textures have often been classified into two categories, deterministic textures and stochastic textures [Heeger 1995]. A deterministic texture is characterized by a set of primitives and a placement rule (e.g., a tile floor). A stochastic texture, on the other hand, does not have easily identifiable primitives (e.g., granite, bark, sand). This chapter focuses on the analysis and synthesis of stochastic textures. There are basically three kinds of techniques for texture analysis and synthesis. The first technique models the texture image by its formation mechanism. For example, the lumpy

background models assume the anatomical images are composed of many blobs, and simulate the anatomical images by adding the blobs onto a constant background. The second technique models the texture image by some probability distributions. For simplicity, a Markov assumption is usually made; that is, the probability density of a pixel, when conditioned on a set of pixels in a small spatial neighborhood, is independent of the pixels beyond the neighborhood. This assumption is, however, inadequate for image modeling when images contain large structures which extend beyond a small neighborhood of pixels. The third technique is based on the assumption that most of the spatial information characterizing a texture image can be captured in the statistics of a small set of filter outputs [Heeger 1995], [Zhu 1997], [Zhu 1998]. Initially, people believed that the marginal distributions of the filter outputs were enough to characterize the statistics of the texture image. Then given the marginal statistics, we can use the maximum-entropy principle to get a Markov random field model as in the second technique, but the new model is not restricted to a small neighborhood and has stronger descriptive ability. Recently, people have realized that we can also explore the joint statistics of the filter outputs and give a more powerful prior model for the texture images.

3.2 Texture formation model

The first approach is to model the texture formation process. A successful medical texture synthesis has been proposed by Rolland and Barrett [Rolland 1990], [Rolland 1992]. The images, called lumpy backgrounds, consist of a random number of single structures called "blobs" located at random locations. The lumpy backgrounds have the advantage of being statistically tractable (within the second-order statistics) and stationary. Lumpy backgrounds are mainly used for the simulation of anatomical variation in nuclear medicine images as part of an investigation of the

resolution-noise trade off for different aperture sizes in a pinhole imaging system.

One generalization of the lumpy backgrounds, called the clustered-blob lumpy background was proposed by Bochud et al. [Bochud 1998]. The clustered-blob lumpy background contains asymmetric blobs with similar properties pooled into larger spatial structures. The main application for the method has been the simulation of mammograms.

3.2.1 Lumpy background

In order to model and simulate the randomness in the background of the object to be imaged, Rolland and Barrett give two approaches to simulate the lumpy background. We here refer to images as functions of two dimensional discrete variables.

Lumpy Background of Type 1 The first approach randomly superimposes Gaussian functions on a constant background. We often refer to these Gaussian functions as Gaussian blobs or simply blobs. If we denote the lumpy component of the background by b, then b is generated by filtering a Poisson point process through a Gaussian kernel, and is given by

$$\begin{aligned} b(\mathbf{r}) &= \sum_{j=1}^{K} \delta(\mathbf{r}-\mathbf{r}_j) * \left[\frac{b_0}{\pi r_b^2}\exp\left(-|\mathbf{r}|^2/r_b^2\right)\right] \\ &= \sum_{j=1}^{K} \frac{b_0}{\pi r_b^2}\exp\left(-|\mathbf{r}-\mathbf{r}_j|^2/r_b^2\right), \end{aligned} \tag{3.1}$$

where $\mathbf{r}$ is a a *discrete* (two-dimensional spatial) variable, $\mathbf{r}_j$ is a *discrete* random variable uniformly distributed over the image; K is the number of blobs in the background. Note K is a Poisson random variable.

The autocorrelation function is

$$R(\mathbf{r}) = \frac{W(0)}{\pi r_b^2}\exp\left(-|\mathbf{r}|^2/2r_b^2\right), \tag{3.2}$$

in which $W(0)$ is a measure of lumpiness and is given by

$$W(0) = \frac{\overline{K}}{N^2} b_0^2, \tag{3.3}$$

where $\frac{\overline{K}}{N^2}$ is the mean number of blobs per pixels and b_0 is the strength of the blobs.

The mean level $\overline{B}$ in the object is the sum of two terms, and is expressed by

$$\overline{B} = B_0 + \frac{\overline{K}}{N^2} b_0, \tag{3.4}$$

where the first term is the mean level of the constant background and the second term is the average value of the Gaussian blobs.

Lumpy Background of Type 2 The second approach to lumpy backgrounds with a Gaussian autocorrelation is to filter uncorrelated Gaussian noise. The filter function is chosen to be a Gaussian function of correlation length r_b

$$\begin{aligned} b(\mathbf{r}) &= a(\mathbf{r}) * \left[\frac{H(0)}{\pi r_b^2} \exp\left(-|\mathbf{r}|^2 / r_b^2\right) \right] \\ &= \frac{H(0)}{\pi r_b^2} \sum_j a(\mathbf{r}_j) \exp\left(-|\mathbf{r} - \mathbf{r}_j|^2 / r_b^2\right), \end{aligned} \tag{3.5}$$

where $a(\mathbf{r})$ is a uncorrelated Gaussian point process with mean value A_0 and standard deviation σ. Note that $\mathbf{r}_j$ is a nonrandom variable but the amplitude of the Gaussian blobs is a Gaussian random variable. This is quite a different approach from the lumpy backgrounds of type 1 where the location of the blobs was the random variable and their amplitude was constant.

The power spectrum of uncorrelated noise is constant with a value of σ^2, and if we denote by $H(\boldsymbol{\rho})$ the DTFT of the Gaussian filter, the resulting power spectrum of the lumpy background is given by

$$W(\boldsymbol{\rho}) = \sigma^2 |H(\boldsymbol{\rho})|^2 = \sigma^2 [H(0)]^2 \exp\left(-2\pi^2 r_b^2 \ |\boldsymbol{\rho}|^2\right), \tag{3.6}$$

where $\boldsymbol{\rho}$ is a *continuous* (two-dimensional frequency) variable. The lumpiness in the autocorrelation function is

$$W(0) = \sigma^2 [H(0)]^2. \tag{3.7}$$

The mean level $\overline{B}$ in the object is expressed by

$$\overline{B} = A_0 H(0). \tag{3.8}$$

In practice images have a finite size, we use the DFT (or FFT) to calculate a circular convolution between the uncorrelated Gaussian point process and the Gaussian filter function, so the generated lumpy background is wrapped around; that is, there exist correlations between both sides of the image. In order to learn about the covariance matrix of the lumpy background, we use a matrix-vector form as follows:

$$\mathbf{b} = \mathbf{H}_c \mathbf{a}, \tag{3.9}$$

where $\mathbf{H}_c$ is a circulant matrix, formed by cyclic shifts of the filter coefficients. The covariance matrix of $\mathbf{a}$ is an identity matrix, and the covariance matrix of $\mathbf{b}$ is

$$\mathbf{K_b} = \mathbf{H_c K_a H_c^\dagger} = \mathbf{H_c H_c^\dagger}. \tag{3.10}$$

We know that a circulant matrix can be diagonalized by the DFT (see Appendix A), i.e., $\mathbf{H_c} = \mathbf{F^\dagger \Lambda F}$, where $\mathbf{F}$ is the unitary DFT matrix and Λ is a diagonal matrix. The diagonal elements of Λ are samples of $H(\boldsymbol{\rho})$. Hence,

$$\mathbf{K_b} = \mathbf{F^\dagger \Lambda F F^\dagger \Lambda F}. \tag{3.11}$$

Since $\mathbf{F}^{-1} = \mathbf{F}^\dagger$, we have

$$\mathbf{K_b} = \mathbf{F^\dagger \Lambda^2 F}. \tag{3.12}$$

Now we see that the covariance matrix of $\mathbf{b}$ can also be diagonalized by the DFT, so it is a circulant matrix. The eigenvalues of the covariance matrix are samples of the power spectrum $W(\boldsymbol{\rho})$:

$$\widetilde{W}(m,n) = \sigma^2 [H(0)]^2 \exp\left(-2\pi^2 r_b^2 \left(\left|\frac{2\pi m}{N}\right|^2 + \left|\frac{2\pi n}{N}\right|^2\right)\right), \; 0 \le m,n \le N. \tag{3.13}$$

Comparison between both types of lumpy background By the central limit theorem, the Poisson process $\sum_{j=1}^{K} \delta(\mathbf{r} - \mathbf{r}_j)$ converges to the uncorrelated Gaussian process, thus type 1 lumpy background converges to the type 2 lumpy background. We illustrate both types of lumpy background in Fig. 3.1 which shows that when the mean number of blobs is small, the type 1 lumpy background is non-Gaussian, and the increase of the mean number of blobs let the type 1 lumpy background be close to the type 2 lumpy background which has a Gaussian distribution.

3.2.2 Clustered-blob lumpy background

Bochud et al. [Bochud 1998] give a method to generalize the lumpy background of type 1, called the clustered-blob lumpy background (CBLB) technique. This technique groups together the asymmetric blobs oriented at a given direction into a "super-blob". A random number of super-blobs are positioned in the image at random locations. The number of super-blobs as well as the number of blobs is Poisson distributed and the random location has an uniform distribution. The clustered-blob lumpy components are expressed by

$$b(\mathbf{r}) = \sum_{i=1}^{K} \sum_{j=1}^{N_k} b_{\theta_i}(\mathbf{r} - \mathbf{r}_i - \mathbf{r}_{ij}). \tag{3.14}$$

Instead of the symmetric Gaussian blobs, Bochud et al. use an asymmetrical exponential blob

$$b_\theta(\mathbf{r}) = \exp\left[-\alpha \frac{\|R_\theta \mathbf{r}\|^\beta}{L(R_\theta \mathbf{r})}\right], \tag{3.15}$$

where R_θ is the rotation matrix corresponding to an angle of θ, $L(\mathbf{r})$ is the characteristic length of the exponential, and α, β are adjustable parameters. The blob asymmetry allows us to take into account the local anisotropy observed in real tissues. We can rewrite the asymmetrical blob function by letting the 2D spatial coordinates

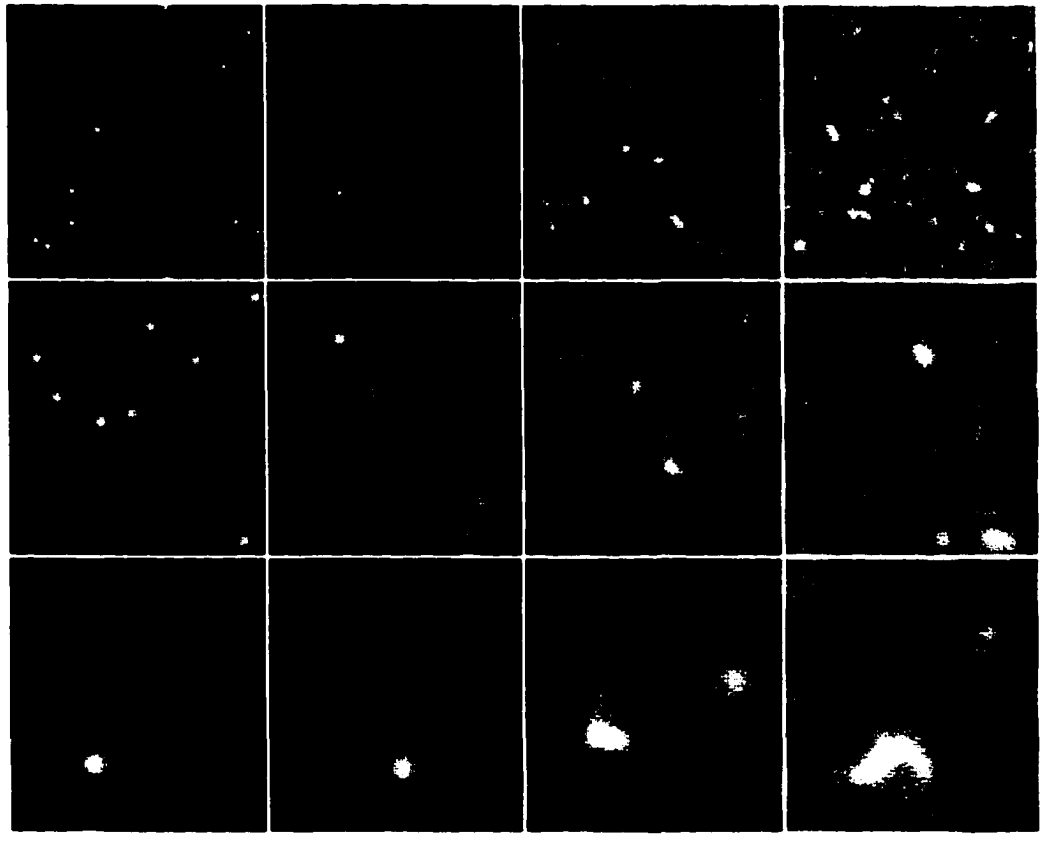


FIGURE 3.1. The correlation length increases from top to bottom with r_b equal 3, 6, and 10 pixels. The first three columns show the type 1 lumpy background: the mean number of blobs increases from left to right with K equal 10, 100, and 10^4. The fourth column shows the lumpy background of type 2. All of the images have the same mean background value $\overline{B}$ and the lumpiness $W(0)$. When the mean number of blobs increases, both type 1 and type 2 lumpy backgrounds converge to each other.

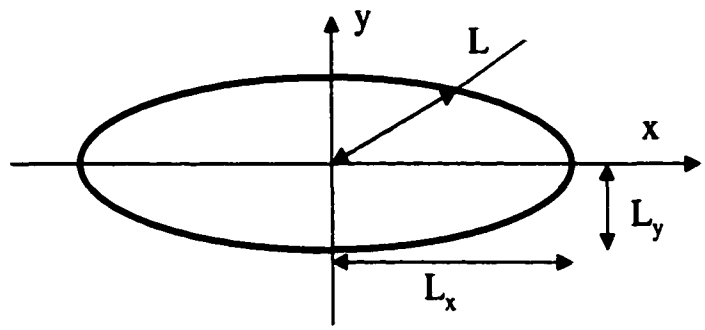


FIGURE 3.2. The characteristic length L is equal to the radius of the ellipse having half-axes Lx and Ly.

$\mathbf{r} = (x, y)$ as follows:

$$b_\theta(x,y) = \exp\left[-\alpha \frac{\left(\sqrt{x^2+y^2}\right)^\beta}{\frac{L_x L_y}{\sqrt{L_x^2\cos^2(\theta-\arctan(y/x))+L_x^2\sin^2(\theta-\arctan(y/x))}}}\right], \qquad (3.16)$$

where L_x and L_y are the characteristic length in the x and y direction, respectively.

Both lumpy background and clustered-blob lumpy background models are successful in medical image synthesis, but it is difficult to know the full density function for complex texture formation models.

3.3 Statistical model (Markov random field)

The second approach is statistical modeling, which characterizes texture images as arising from probability distributions on random fields. The Gibbs distribution and Markov random field (MRF) models have shown to be appropriate tools for modeling spatial context [Cross 1983], [Li 1995]. MRF models were popularized by Besag for modeling spatial interactions on lattice system and were used by Cross and Jain [Cross 1983] for texture modeling. An important characteristic of MRF modeling is that the global patterns are formed via stochastic propagation of local interactions.

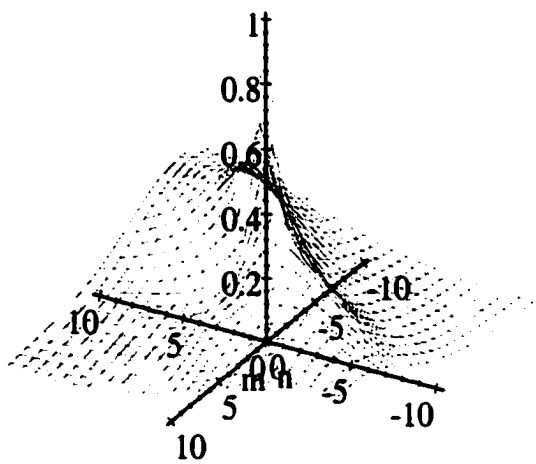


FIGURE 3.3. Asymmetrical exponential blob $\theta = \frac{\pi}{4}, \alpha = 2, \beta = 0.5, L_x = 5, L_y = 2$

We denote $b(\mathbf{r})$ as the random variable at location $\mathbf{r}$ and $\mathcal{N}_\mathbf{r}$ as the neighborhood of a pixel at location $\mathbf{r}$. We let $\mathcal{N} = \{\mathcal{N}_\mathbf{r}\}$ be a neighborhood system of an image, which satisfies $\mathbf{r} \notin \mathcal{N}_\mathbf{r}$ and $\mathbf{r}' \in \mathcal{N}_\mathbf{r} \Longleftrightarrow \mathbf{r} \in \mathcal{N}_{\mathbf{r}'}$. The pixels in $\mathcal{N}_\mathbf{r}$ are called neighbors of $\mathbf{r}$. A clique consists of a set of pixels that are neighbors to each other.

Definition. $p(b)$ is an MRF distribution with respect to $\mathcal{N}$ if

$$p(b(\mathbf{r}) \,|\, b(\overline{\mathbf{r}})) = p(b(\mathbf{r}) \,|\, b(\mathcal{N}_\mathbf{r})), \tag{3.17}$$

where $b(\overline{\mathbf{r}})$ denotes the value of pixels other than $\mathbf{r}$.

Definition. $p(b)$ is a Gibbs distribution with respect to $\mathcal{N}$ if

$$p(b) = \frac{1}{Z} \exp\left\{ -\sum_c \phi_c(b) \right\}, \tag{3.18}$$

where Z is the normalizing constant and $\phi_c(b)$ is a potential function of pixels in clique c.

The Hammersley-Clifford theorem establishes the equivalence between MRF and the Gibbs distribution. Next we will give three kinds of MRF models: 1) Potts model 2) pairwise difference model 3) Gaussian MRF model. The pairwise difference models are generalizations of the Potts model, and both of them presume that surfaces of objects are smooth and adjacent pixels in images have similar intensity values unless separated by edges. Therefore the potential function of pixels in clique c is a U-shaped cost function of the difference between neighboring pixels. The Gaussian MRF models are the linear regression models for the pixel at $\mathbf{r}$ by its neighboring pixels in $\mathcal{N}_\mathbf{r}$.

3.3.1 Potts model

The simplest example of a Markov random field or Gibbs distribution is the Potts model

$$p(b) = \frac{1}{Z} \exp\left(-\sum_{i \sim j} \beta \, I[b(\mathbf{r}_i) \neq b(\mathbf{r}_j)] \right). \tag{3.19}$$

Here $i \sim j$ means that i and j are neighbors in some prescribed undirected graph with the pixels as vertices; in other words, the pixels i and j are in the same clique. The sum in (3.19) is over all neighboring pairs, and $I[\,]$ is the indicator function, taking the value 1 if $b(\mathbf{r}_i)$ and $b(\mathbf{r}_j)$ are different (assuming that $b(\mathbf{r})$ can take only a finite discrete set of values), and 0 otherwise.

3.3.2 Pairwise difference model

More generally, we can use a pairwise difference prior defined by

$$p(b) = \frac{1}{Z} \exp\left(-\sum_{i \sim j} \phi(b(\mathbf{r}_i) - b(\mathbf{r}_j)) \right), \tag{3.20}$$

for some symmetric U-shaped potential functions ϕ.

Green [Green 1990] gave a flexible family of prior distributions in (3.21) ranging from absolute value to Gaussian as a shape parameter $w \to 0$ and ∞.

$$p(b) = \frac{1}{Z} \exp\left(-\sum_{i \sim j} \beta \log\left(\cosh\left(\frac{b(\mathbf{r}_i) - b(\mathbf{r}_j)}{w} \right) \right) \right). \tag{3.21}$$

We illustrate the shape of ϕ in Figs. 3.4 and 3.5 respectively for $w = 0.01$ and $w = 10^4$.

In order to preserve edges and object boundaries, Zhu [Zhu 1997] gave two types of potential functions in (3.22) and (3.23)

$$p(b) = \frac{1}{Z} \exp\left(-\sum_{i \sim j} \beta \min\left(\theta^2, (b(\mathbf{r}_i) - b(\mathbf{r}_j))^2\right) \right), \tag{3.22}$$

$$p(b) = \frac{1}{Z} \exp\left(-\sum_{i \sim j} \beta \left(1 - \frac{1}{1 + c(b(\mathbf{r}_i) - b(\mathbf{r}_j))^2} \right) \right), \tag{3.23}$$

which have flat tails as shown in Figs. 3.6 and 3.7.

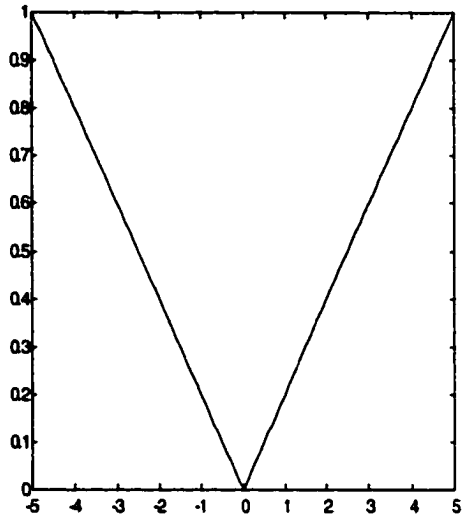


FIGURE 3.4. $\phi(\xi) = \beta \log\left(\cosh\left(\frac{\xi}{w}\right)\right)$ when $w = 0.01$

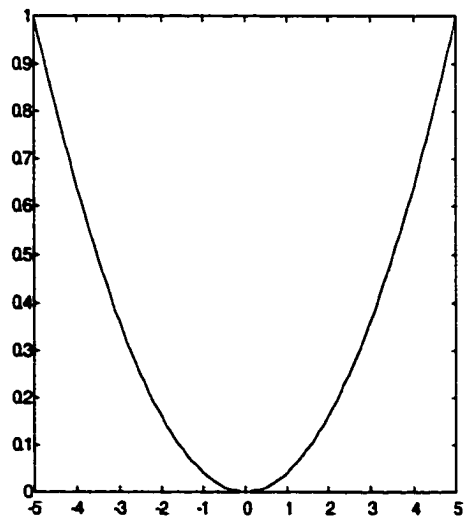


FIGURE 3.5. $\phi(\xi) = \beta \log\left(\cosh\left(\frac{\xi}{w}\right)\right)$ when $w = 10^4$

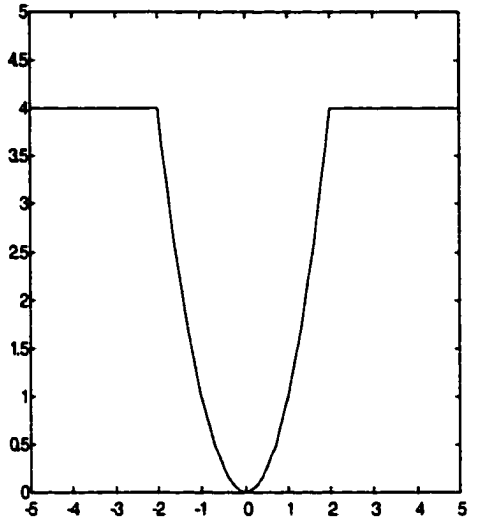


FIGURE 3.6. Line process $\phi(\xi) = \beta \min(\theta^2, \xi^2)$

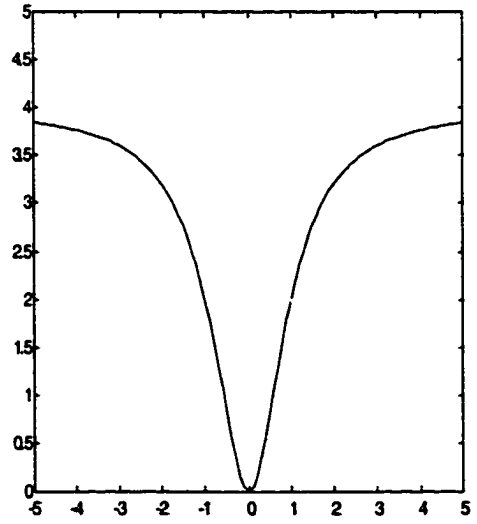


FIGURE 3.7. T-function $\phi(\xi) = \beta\left(1 - \frac{1}{1+c\xi^2}\right)$

3.3.3 Gaussian MRF (GMRF) model

Another MRF model for texture is the auto-normal model, also called a Gaussian MRF (GMRF). It is a generalization of an auto-regressive model in image space:

$$b(\mathbf{r}_i) = u(\mathbf{r}_i) + \sum_{j \in N_{\mathbf{r}_i}} \beta_{ij} \left(b(\mathbf{r}_j) - u(\mathbf{r}_j) \right) + v_i, \tag{3.24}$$

where v_i is a Gaussian random variable, $N(0, \sigma^2)$, and $u(\mathbf{r}_i)$ is the mean value of the pixel at location $\mathbf{r}_i$. The conditional distribution is a normal regression,

$$p\left(b(\mathbf{r}_i) \,|\, b(\mathbf{r}_j)\right) \sim N\left(u(\mathbf{r}_i) + \sum \beta_{ij} \left(b(\mathbf{r}_j) - u(\mathbf{r}_j) \right), \sigma^2\right). \tag{3.25}$$

The probability density function of the texture is the multivariate Gaussian model, $N(u, \sigma^2 B^{-1})$, and $p(b)$ is of the form

$$p(b) = \frac{1}{(2\pi\sigma^2)^{M/2}} |B|^{1/2} \exp\left(-\frac{1}{2\sigma^2} \left(b(\mathbf{r}) - u(\mathbf{r}) \right)^T B \left(b(\mathbf{r}) - u(\mathbf{r}) \right) \right), \tag{3.26}$$

where the diagonal element of B is unity and off-diagonal (i, j) element is $-\beta_{ij}$.

The Gibbs distribution and the MRF distribution can be sampled by using the Gibbs sampler or the Metropolis algorithms which will be discussed in chapter 6. However the MRF models are severely limited by small cliques, i.e., the neighborhood is usually of order less than or equal to three pixels, which is too small to capture features of texture.

3.4 Visual filter model

The third approach is inspired by the multi-channel filtering mechanism discovered and accepted in neurophysiology [Bergen 1991], [Chubb 1991]. This mechanism suggests that the visual system decomposes the retinal image into a set of sub-bands, which are computed by convolving the image with an appropriately chosen set of linear filters, and most of the spatial information characterizing a texture image can

be captured in the statistics of this set of filter outputs. Recently many researchers have used this principle for texture segmentation, classification and synthesis. Heeger and Bergen [Heeger 1995] make use of the marginal distribution of the filter outputs for image synthesis. Unser [Unser 1995] characterizes the texture by a set of channel variances estimated at the output of the corresponding filter bank in texture classification task. Simoncelli [Simoncelli 1998], [Simoncelli 1999] models the joint statistics of images in the wavelet domain by (1) the local spatial correlation of coefficients within each subband, (2) the cross-correlation between coefficients at adjacent scales and all orientations and (3) the first few moments of the pixel histogram. Simoncelli and his colleagues use the joint statistical characterization in the wavelet domain for image compression and synthesis. In this section we will introduce the pyramid-based texture analysis and synthesis method proposed by Heeger and Bergen and the prior model of the texture images based on the minimax entropy principle, which is constrained by the estimated marginal distributions from subbands of an example image. We will also give our prior model based on the joint statistics of the filter outputs.

3.4.1 Pyramid-based texture analysis and synthesis

Heeger and Bergen propose the pyramid-based texture synthesis method, which synthesizes textures by matching distributions (or histograms) of a bank of (orientation and spatial-frequency selective) linear filters. The pyramid-based texture analysis and synthesis technique starts with an input texture image and a white noise image. The algorithm modifies the noise to make it look like the input texture by the pyramid transform and histogram matching.

In order to see whether the marginals of a set of filter outputs can characterize textures, we simulate lumpy backgrounds of type 2 as the reference image, and use the Laplacian pyramid or both the Laplacian and steerable pyramids as linear filter banks. We can see from Fig. 3.8 that the synthesized images resemble the reference

image, and the result is improved by adding more filters. We also simulate clustered-blob lumpy backgrounds as the reference image, and use the steerable pyramid as linear filter banks. The synthesized images are illustrated in Fig. 3.9. The quality of the synthesized images is not as good as the synthesized lumpy backgrounds in Fig. 3.8 subjectively.

Algorithm 3.1: Synthesize texture images by the pyramid-based histogram matching method

1. Given a reference image b^{obs}
2. Decompose b^{obs} into a set of subband images
3. Compute the histograms of both b^{obs} and the subband images
4. Initialize a synthesized image b by a white noise process
5. Repeat for 5 or 6 iterations:
   - 5.1 Apply histogram matching to b
   - 5.2 Decompose b into a set of subband images
   - 5.3 Apply histogram matching to the subband images
   - 5.4 Synthesize new b from the subband images

3.4.2 Learning prior models by minimax entropy

The pyramid-based texture analysis and synthesis method can synthesize textures that look like the reference image. However, this method does not give us the statistical distribution of the texture images. If we want to study the statistical properties of training images $\{b_n^{obs}, n = 1, ..., N\}$, we can start from exploring a set of linear filters $\{F^{(\alpha)}, \alpha = 1, ..., K\}$ which are characteristic of the observed images, and get the empirical marginal distributions (or histograms) of filter outputs. In order to estimate the distribution of an ensemble of images given the empirical marginal distributions of filter outputs, Zhu et al. [Zhu 1997] [Zhu 1998] use the maximum entropy principle and get a unique solution.

Definition Given a probability distribution $p(b)$, the marginal distribution of

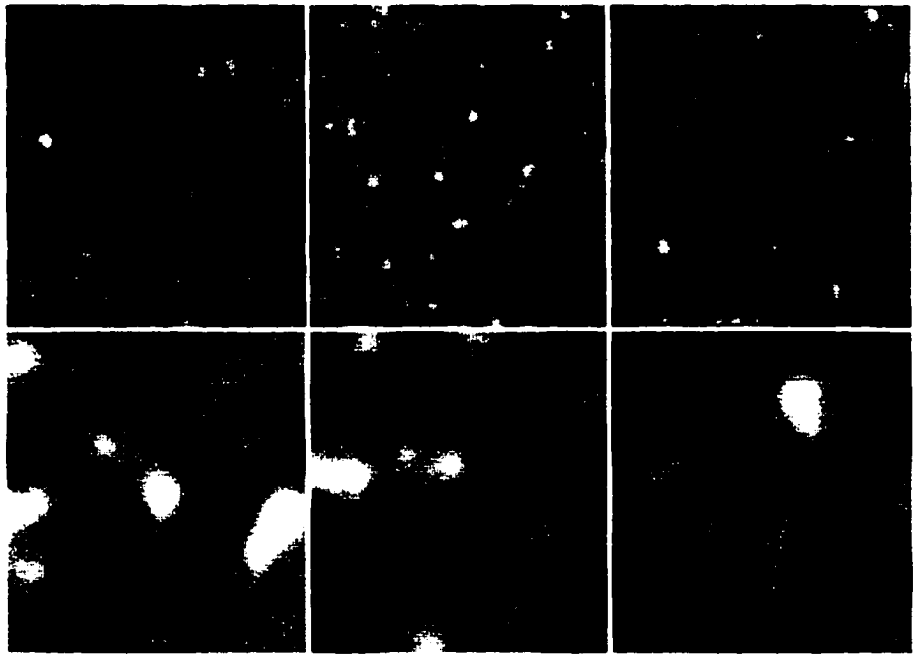


FIGURE 3.8. The first column shows two reference lumpy backgrounds with the correlation length $r_b = 3$ and $r_b = 10$. The second column shows the synthesized images using the Laplacian pyramids as the linear filters. The third column shows the synthesized images using both the Laplacian and steerable pyramids as the linear filters.

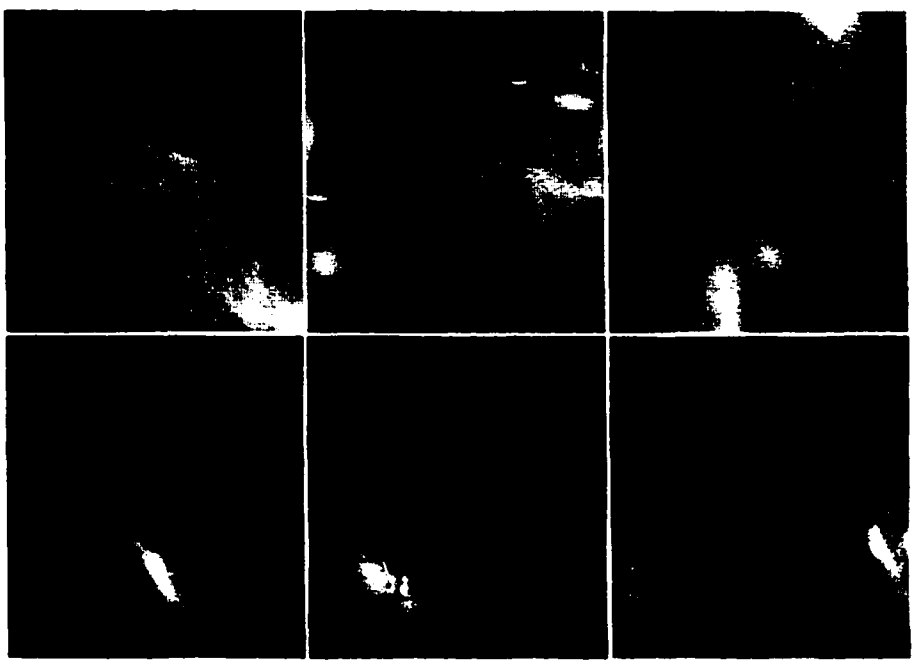


FIGURE 3.9. The first column shows two reference clustered-blob lumpy backgrounds with different number of blobs. The second column shows the synthesized images using the steerable pyramid as the linear filters. The third column shows the synthesized images using both the steerable pyramid and the Laplacian pyramid as the linear filters.

$p(b)$ with respect to the linear filter $F^{(\alpha)}$ is

$$f^{(\alpha)}(z) = \int_{z=F^{(\alpha)}*b(\mathbf{r})} p(b)\ db = E\left[\delta\left(z - F^{(\alpha)} * b(\mathbf{r})\right)\right], \tag{3.27}$$

where $F^{(\alpha)} * b(\mathbf{r})$ is the filter response at location $\mathbf{r}$. We see that the marginal distribution of the filter output is the Radon transform of an "object" $p(b)$ along the line $z = F^{(\alpha)} * b(\mathbf{r})$. In practical implementations we can have at most K projection lines if we have K filters, so the task of reconstructing the full density function from K empirical marginal distributions is an ill-posed inverse problem. In order to avoid the ill-posedness, we can use the maximum entropy principle

$$p(b) = \arg\max\left\{-\int p(b)\log p(b)\,db\right\},$$

$$\text{subject to } E\left[\delta\left(z - F^{(\alpha)} * b(\mathbf{r})\right)\right] = f^{(\alpha)}(z),\ \alpha = 1,...,K,$$

$$\text{and } \int p(b)\ db = 1.$$

The reason to choose the maximum entropy principle is that while $p(b)$ satisfies the constraints along some dimensions, it is made as random as possible in other unconstrained dimensions. In other words, $p(b)$ should represent information no more than what is available. Solving this maximization problem by Lagrange multipliers yields

$$\begin{aligned} p(b) &= \frac{1}{Z}\exp\left\{-\sum_{\mathbf{r}}\sum_{\alpha=1}^{K}\int \phi^{(\alpha)}(z)\,\delta\left(z - F^{(\alpha)} * b(\mathbf{r})\right)dz\right\} \\ &= \frac{1}{Z}\exp\left\{-\sum_{\mathbf{r}}\sum_{\alpha=1}^{K}\phi^{(\alpha)}\left(F^{(\alpha)} * b(\mathbf{r})\right)\right\}, \end{aligned} \tag{3.28}$$

where the Lagrange parameter $\phi^{(\alpha)}$ takes the form of a continuous function of the filter response $F^{(\alpha)} * b(\mathbf{r})$. The resulting model is a Markov random field (MRF) model, but has much stronger descriptive ability than the previous MRF models used

for texture modeling because the size of neighborhood is determined by the size of the linear filters which may lie in a wide spatial range.

To compute $\phi^{(\alpha)}, \alpha = 1, ..., K$, Zhu et al. approximate them by piecewise constant functions, i.e., vectors, and adopt the Gibbs sampler, which samples from the distribution $p(b)$ in (3.28), and then compute the histogram of the filter responses for the sample. The values of $\phi^{(\alpha)}$ are updated to bring the histograms closer to the observed ones. Since it involves Gibbs sampling, this procedure is computationally intensive and slow.

The task of computing $\phi^{(\alpha)}$ may not be as hard as it appears; in fact, there is an easier way than Zhu's method. We see from (3.28) that the maximum entropy solution implies that the filter outputs are independent of each other, since

$$p(b) = \frac{1}{Z} \prod_{\alpha=1}^{K} \exp\left\{ -\sum_{\mathbf{r}} \phi^{(\alpha)} \left(F^{(\alpha)} * b(\mathbf{r}) \right) \right\}. \tag{3.29}$$

Also, since the constraints are the same for all locations, $\phi^{(\alpha)}$ should be independent of $\mathbf{r}$. Therefore we can design $\phi^{(\alpha)}$ by looking at the single filter output of the whole image. If $f^{(\alpha)}(z)$ is the observed histogram of the response from $F^{(\alpha)}$, then we can transform the histogram to a known density function by histogram matching. For example, if the desired distribution is a Gaussian density function $N(0,1)$ the histogram matching maps z monotonically onto a new variable y by

$$y(z) = \text{erf inv}\left(2 \int_{-\infty}^{z} f^{(\alpha)}(x)\ dx - 1 \right). \tag{3.30}$$

Note y is a random variable $\sim N(0,1)$. The marginal distribution of the filter response is

$$\begin{aligned} p_Z(z) &= p_Y(y(z)) \frac{dy}{dz} \\ &= \frac{1}{2\pi} \exp\left(-\frac{1}{2} \left(\text{erf inv}(2 \int_{-\infty}^{z} f^{(\alpha)}(x)\ dx - 1) \right)^2 + \log\left(\frac{dy}{dz} \right) \right) \end{aligned} \tag{3.31}$$

Hence

$$\phi^{(\alpha)}(z) = \frac{1}{2}\left(\text{erf inv}(2\int_{-\infty}^{z} f^{(\alpha)}(x)\ dx - 1)\right)^2 - \log\left(\frac{dy}{dz}\right). \tag{3.32}$$

If we choose the another distribution, we can also get the same solution of $\phi^{(\alpha)}$ by the histogram matching.

If we just calculate a ratio of the prior models, then Jacobian terms $\log\left(\frac{dy}{dz}\right)$ in both the denominator and numerator are cancelled out, so we can simplify the potential function into

$$\phi^{(\alpha)}(z) = \frac{1}{2}\left(\text{erf inv}(2\int_{-\infty}^{z} f^{(\alpha)}(x)\ dx - 1)\right)^2. \tag{3.33}$$

Example If z is a random variable with the Student t distribution, $t(3)$, Fig. 3.10 illustrates the histogram $f(z)$ and Fig. 3.11 illustrates the piecewise approximate function $\phi^{(\alpha)}(z)$ by using the above method.

We have not yet discussed how to design the subband filters $F^{(\alpha)}$. The assumption of independence between filter responses is not valid unless the subband filters implement the independent component analysis (ICA). In order to solve this problem, Zhu et al. construct a library of the subband filters, which includes

1. The intensity filter $\delta()$, which captures the DC component
2. Isotropic center-surround filters, i.e., the Laplacian of Gaussian filters
3. Gabor filters with both sine and cosine components
4. Spectrum analyzers, whose response are powers of the Gabor filters

Then they use a stepwise algorithm to choose the subband filters from the library to minimize the entropy of $p(b)$ in (3.29), which is the product of K marginal density functions. As we have seen in chapter 2, the ICA minimizes the marginal entropy, so, Zhu et al. actually approximate the independent component analysis of the input image. However, this method does not implement the "true" ICA since the subband filters are restricted within the constructed library; furthermore, the computational cost is excessive.

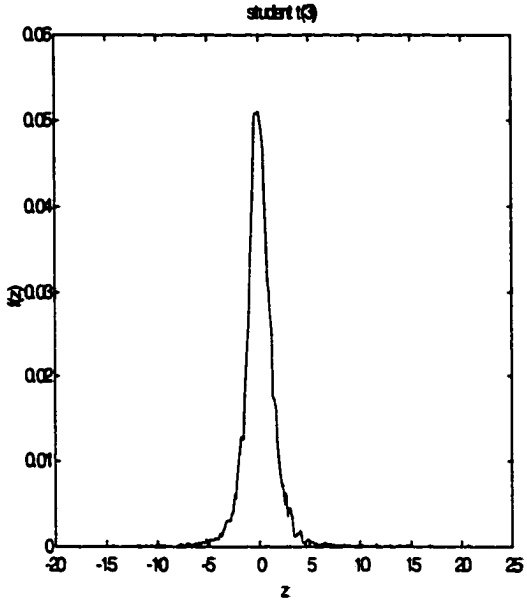


FIGURE 3.10. Illustration of the estimated histogram from samples of student t distribution.

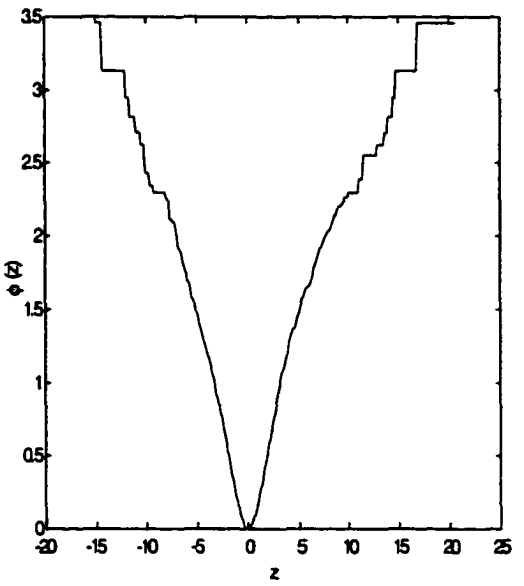


FIGURE 3.11. Illustration of the estimated ϕ from samples of the student t distribution.

Finally, given the prior model of texture images in (3.28), we can use the Metropolis algorithm to synthesize new texture samples. We illustrate both the reference images and the synthesized images Fig. 3.12.

Algorithm 3.2: Synthesize texture images by the Metropolis algorithm

1. Given a reference image b^{obs}
2. Select a set of linear filters $\{F^{(\alpha)},\ \alpha = 1, ..., K\}$
3. Compute the histograms of the filter responses
4. Initialize a synthesized image b_0 by white noise
5. Repeat until enough samples are collected:
   - 5.1 At iteration n, generate the filter responses $I^{(\alpha)} = F^{(\alpha)} * b_n$
   - 5.2 Randomly pick a location $\mathbf{r}_0$
   - 5.3 Sample a scalar random variable w_α from the marginal density of $I^{(\alpha)}$
   - 5.4 Update $I^{(\alpha)}(\mathbf{r}_0) = I^{(\alpha)}(\mathbf{r}_0) + \lambda w_\alpha$, where λ is a controlling parameter
   - 5.5 Reconstruct the synthesized image $\{I^{(\alpha)}(z), \alpha = 1, ..., K\} \rightarrow \widehat{b}$
   - 5.6 Accept the synthesized image $\widehat{b}$ with the probability as $\min\left(1, \frac{p(\widehat{b})}{p(b_n)}\right)$

3.4.3 Joint statistics of filter response

The marginal histograms of the filter outputs are sufficient statistics of $p(b)$ in (3.28). However, as we can see from the synthesized images of the clustered-blob lumpy background, the marginals of a fixed finite linear basis are often insufficient. In particular, long-range structures (such as straight or curved contours), pseudo-periodic patterns, and second-order textures are not well represented in typical bases [Simoncelli 1998]. One of the reasons is that the information in each subband is processed independently. When we use pyramid transforms for image decomposition, the image is represented as a weighted sum of basis functions, and the weight coefficients are computed by projecting onto a set of projection functions which are translated copies of the convolution kernels (i.e., $F^{(\alpha)}, \alpha = 1, ..., K$). Since we process each subband independently, only spatial characteristics represented in the basis functions can be conveyed to the reconstructed image.

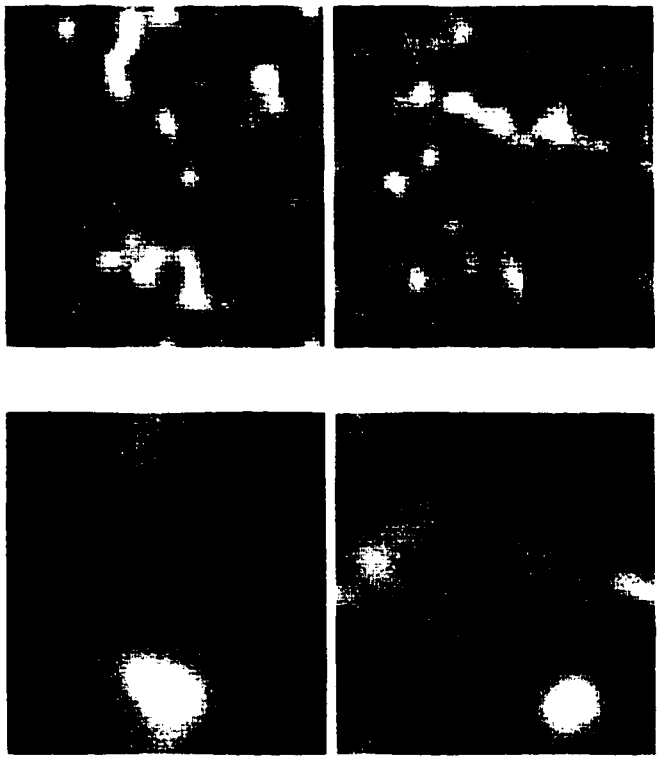


FIGURE 3.12. Synthesize texture images by the Metropolis algorithm. The first column shows two reference lumpy backgrounds with the correlation length r_b = 3 and r_b = 10. The second column shows images synthesized using the steerable pyramids.

In order to solve this problem, we can apply ICA for the filter responses. ICA estimates an invertible $K \times K$ matrix $\mathbf{W}$, which combines the given linear filters in order to yield the most informative marginals such that the product over the marginals most closely approximates the joint probability density function of the filter outputs. Since the number of filters is usually small, we can easily use ICA to explore the joint statistics of the filter outputs. By combining ICA with the previous prior models in (3.28), we give a new probability density function

$$p(b) = \frac{1}{Z} \prod_{\alpha=1}^{K} \exp\left\{ -\sum_{\mathbf{r}} \phi^{(\alpha)} \left(\sum_{\beta=1}^{K} W_{\alpha\beta} \left(F^{(\beta)} * b(\mathbf{r}) \right) \right) \right\}, \tag{3.34}$$

where $W_{\alpha\beta}$ is the (α, β) element of the ICA matrix. If we define a new set of linear filters by

$$F_{new}^{(\alpha)} = \sum_{\beta=1}^{K} W_{\alpha\beta} F^{(\beta)}, \tag{3.35}$$

then we have the same expression of the previous prior density function except using the new filters

$$p(b) = \frac{1}{Z} \prod_{\alpha=1}^{K} \exp\left\{ -\sum_{\mathbf{r}} \phi^{(\alpha)} \left(F_{new}^{(\alpha)} * b(\mathbf{r}) \right) \right\}. \tag{3.36}$$

However, we need to let the filtered images have the same sampling rate since the ICA matrix is estimated by samples of each filtered image in the same spatial locations. Thus we cannot use the steerable pyramid in which filtered images have different sampling rates, but we can use the dyadic wavelet frame which uses the basis functions of the steerable pyramid.

Chapter 4

HOTELLING OBSERVER

4.1 Introduction

After discussing the modeling, synthesis and analysis of the texture images and related techniques in the previous chapters, we are in a position to detect lesions in medical images. In this chapter we will give an introduction to the mathematical model of a medical imaging system and signal detection theory. Then we will focus on computing the template of the Hotelling observer. In chapter 5 and 6, we will discuss approximating the ideal observer for a lesion detection task.

A medical image consisting of M pixels can be represented as an $M \times 1$ column vector $\mathbf{g}$. This vector is related to the object being imaged, denoted $\mathbf{f}$, by a relation of the form [Barrett 1990]

$$\mathbf{g} = \mathcal{H}\mathbf{f} + \mathbf{n}, \tag{4.1}$$

where $\mathbf{n}$ is a vector representing the measurement noise (e.g. Poisson noise), and $\mathcal{H}$ is an operator representing the imaging system. It is important to realize that this equation does not imply that $\mathbf{n}$ is independent of the object $\mathbf{f}$.

Note that $\mathbf{g}$ is a random vector due to the measurement noise and the fact that there are random variations in the objects being imaged. We shall consider both sources of randomness. We must therefore define several kinds of statistical averages. We define a conditional average over all realizations of $\mathbf{n}$ for a fixed $\mathbf{f}$ as

$$\langle \cdots \rangle_{\mathbf{n}|\mathbf{f}} = \int d\mathbf{n}\, p(\mathbf{n}|\mathbf{f}) \cdots, \tag{4.2}$$

where the ellipsis denotes the quantity to be averaged. The function $p(\mathbf{n}|\mathbf{f})$ is the conditional probability density for the noise given the object $\mathbf{f}$, and this function is usually known from our prior knowledge of the imaging system and measurement noise. An average over the objects is given by

$$\langle \cdots \rangle_{\mathbf{f}} = \int d\mathbf{f}\; p(\mathbf{f}) \cdots . \tag{4.3}$$

The function $p(\mathbf{f})$ is the probability density function for the random objects. This prior probability is usually not known, so these averages are usually estimated by samples. We will also use an overbar to denote averages over objects, e.g., $\bar{\mathbf{f}} = \langle \mathbf{f} \rangle_{\mathbf{f}}$.

If there were no object variability, and the only noise was due to the discrete nature of the radiation, then the components of the data vector could be modeled as independent Poisson random variables. The Poisson model is routinely assumed for medical imaging with x rays or gamma rays. In this situation, for a single object $\mathbf{f}$, we say that $\mathbf{g}$ is conditionally Poisson, with the conditional mean given by $\langle \mathbf{g} \rangle_{\mathbf{n}|\mathbf{f}} = \mathcal{H}\mathbf{f}$. The conditional covariance matrix of the noise vector $\mathbf{n}$ in this model is $\mathbf{K}_{\mathbf{n}|\mathbf{f}}$ =diag($\mathcal{H}\mathbf{f}$). The overall covariance matrix of $\mathbf{n}$, denoted by $\mathbf{K}_{\mathbf{n}}$, is obtained by averaging the conditional covariance matrix over $\mathbf{f}$

$$K_{\mathbf{n}} = \left\langle K_{\mathbf{n}|\mathbf{f}} \right\rangle_{\mathbf{f}} = \mathrm{diag}(\mathcal{H}\bar{\mathbf{f}}). \tag{4.4}$$

The covariance matrix of $\mathbf{g}$ is related to the covariance matrix of the noise vectors $\mathbf{n}$ and the covariance operator of the objects $\mathbf{f}$ by

$$\begin{aligned} \mathbf{K} &= \langle [\mathbf{g}-\mathcal{H}\bar{\mathbf{f}}][\mathbf{g}-\mathcal{H}\bar{\mathbf{f}}]^t \rangle \\ &= \left\langle \langle [\mathcal{H}(\mathbf{f}-\bar{\mathbf{f}}) + \mathbf{n}][\mathcal{H}(\mathbf{f}-\bar{\mathbf{f}}) + \mathbf{n}]^t \rangle_{\mathbf{n}|\mathbf{f}} \right\rangle_{\mathbf{f}} \\ &= \mathbf{K}_{\mathbf{n}} + \mathcal{H}\mathbf{K}_{\mathbf{f}}\mathcal{H}^t. \end{aligned} \tag{4.5}$$

The task is to observe a particular image $\mathbf{g}$ and use it to classify the corresponding $\mathbf{f}$ that produced the image into one of two classes. (e.g. normal versus abnormal or lesion-present versus lesion-absent). This is a binary hypothesis test where the null

hypothesis H_0 is that the signal is absent and the alternative hypothesis H_1 is that it is present. In general, this binary detection task can be performed by computing a discriminant function of the data, λ, also called a test statistic. The classification is performed by comparing this test statistic to a threshold λ_{th}; if $\lambda > \lambda_{th}$, $\mathbf{f}$ is said to belong to class 1, while otherwise it is classified into class 0.

The Hotelling observer is an ideal linear classifier, which was presented by Harold Hotelling [Hotelling 1931], but is often referred to as the Fisher linear discriminant [Fisher 1936] in the pattern recognition literature. The distinction among names is frequently based on whether the means and the variances of the classes are sample estimates or population values. If the covariance matrix is estimated by a set of training samples then we call it the Fisher linear discriminant; on the other hand, the Hotelling observer uses the ensemble covariance matrix. As a matter of fact, the name of Fisher-Hotelling observer is more appropriate for our model in Eq. (4.5) because the covariance matrix is the sum of two matrices: the first matrix is a diagonal matrix with the diagonal elements being the mean pixel values; and the second matrix is the sample covariance matrix of the noise-free samples of the background.

For deciding between H_0 and H_1, the Hotelling observer computes the test statistic as a linear function of $\mathbf{g}$:

$$\lambda = \mathbf{w}^T \mathbf{g}, \tag{4.6}$$

where $\mathbf{w}$ is an $M \times 1$ vector of weights $\{w_m\}$, and $\mathbf{g}$ is drawn from one of the two classes. If signal-present and signal-absent images are equally likely to occur, the Hoteling template $\mathbf{w}$ is given by

$$\mathbf{w} = \left[\frac{1}{2}\mathbf{K}_1 + \frac{1}{2}\mathbf{K}_0\right]^{-1} (\bar{\mathbf{g}}_1 - \bar{\mathbf{g}}_0), \tag{4.7}$$

where $\mathbf{K}_j$ is the covariance matrix of $\mathbf{g}$ under hypothesis j and $\bar{\mathbf{g}}_j$ is the corresponding mean vector.

This expression can be further simplified if we consider a SKE detection problem in which a specified signal is added to a random background, where the background

is a random vector $\mathbf{b} = \mathcal{H}\mathbf{f}$ when $\mathbf{f}$ is drawn from the signal-absent ensemble. So the signal-present hypothesis H_1 is that $\mathbf{g} = \mathbf{b} + \mathbf{s} + \mathbf{n}$, and the signal-absent hypothesis H_0 is that $\mathbf{g} = \mathbf{b} + \mathbf{n}$. For this problem, the signal is a deterministic vector, $\mathbf{s} = \bar{\mathbf{g}}_1 - \bar{\mathbf{g}}_0$ and $\mathbf{K}_1 = \mathbf{K}_0 = \mathbf{K}$; thus

$$\mathbf{w} = \mathbf{K}^{-1}\mathbf{s}. \tag{4.8}$$

When the random background and the measurement noise have Gaussian distribution, the true ideal observer performs only linear operations on the data, so the Hotelling observer is an ideal observer. If the measurement noise has a Poisson distribution then the Hotelling observer is not precisely the ideal observer but the difference is very small since the Poisson distribution is very close to the Gaussian distribution when the mean value of pixels is larger than 30 counts.

The performance of the Hotelling observer is specified by the signal-to-noise ratio (SNR), defined as

$$\begin{aligned} \mathrm{SNR}^2 &= \frac{\left[\langle\lambda\rangle_1 - \langle\lambda\rangle_0\right]^2}{\frac{1}{2}var_1(\lambda) + \frac{1}{2}var_0(\lambda)} && (4.9) \\ &= \mathbf{s}\mathbf{K}^{-1}\mathbf{s} = \mathbf{w}^T\mathbf{s}. && (4.10) \end{aligned}$$

It is well known [Barrett 1998a] that if the test statistic is normally distributed, then the SNR is related to the area under receiver operating characteristic curve (AUC) by

$$\mathrm{AUC} = \frac{1}{2} + \frac{1}{2}\,\mathrm{erf}\left(\frac{\mathrm{SNR}}{2}\right), \tag{4.11}$$

where $\mathrm{erf}(\cdot)$ is the error function. Since the linear test statistics are linear combinations of random components of $\mathbf{g}$, they tend to have a Gaussian distribution as a result of the central limit theorem; thus the SNR is a good predictor for the AUC of the Hotelling observer.

The computation of SNR^2 is not a trivial problem since we need to calculate the inverse of the covariance matrix. Unfortunately the size of the covariance matrix is

$M \times M$, where M is the number of image pixels. Direct inversion of the sample covariance matrix $\mathbf{K}$ is very difficult since it requires at least M samples to be invertible and 10 to 100 times M to get a stable estimate. Thus, even for small 64×64 images, we need about $40,000 - 400,000$ images to compute the SNR^2. If the random backgrounds are stationary, $\mathbf{K}$ is a Toeplitz (or block Toeplitz) matrix, often well approximated by a circulant matrix and hence diagonalizable by a discrete Fourier transform. The following discussions give some methods to solve this problem without assuming stationarity.

4.2 The Matrix-inversion lemma method

Suppose we want to invert an overall covariance matrix of the form

$$\begin{aligned} \mathbf{K} &= \mathbf{K_n} + \mathcal{H}\mathbf{K_f}\mathcal{H}^t \\ &= \mathbf{K_n} + \widehat{\mathbf{K}}_\mathbf{b}, \end{aligned} \tag{4.12}$$

where $\widehat{\mathbf{K}}_\mathbf{b}$ is the sample covariance matrix of backgrounds. Given a set of noise-free (simulated) sample images from the class 0 $\{\mathbf{g}_j,\ j = 1, ..., J\}$,we can subtract the sample mean from each image to form the set $\{\delta\mathbf{g}_j, j = 1, ..., J\}$, and the sample covariance matrix can be estimated by

$$\widehat{\mathbf{K}}_\mathbf{b} = \mathbf{W}\mathbf{W}^T, \tag{4.13}$$

where $\mathbf{W}$ is the $M \times J$ matrix,

$$\mathbf{W} = \frac{1}{\sqrt{J}} \left[\delta\mathbf{g}_1, \delta\mathbf{g}_2, ..., \delta\mathbf{g}_J\right]. \tag{4.14}$$

By the Woodbury matrix-inversion lemma [Harville 1997],

$$\left[\mathbf{K_n} + \mathbf{W}\mathbf{W}^T\right]^{-1} = \mathbf{K_n^{-1}} - \mathbf{K_n^{-1}}\mathbf{W}\left[\mathbf{I} + \mathbf{W}^T\mathbf{K_n^{-1}}\mathbf{W}\right]^{-1}\mathbf{W}^T\mathbf{K_n^{-1}}. \tag{4.15}$$

Hence, the weight vector is

$$\mathbf{w} = \mathbf{K}_\mathbf{n}^{-1}\mathbf{s} - \mathbf{K}_\mathbf{n}^{-1}\mathbf{W}\left[\mathbf{I} + \mathbf{W}^T\mathbf{K}_\mathbf{n}^{-1}\mathbf{W}\right]^{-1}\mathbf{W}^T\mathbf{K}_\mathbf{n}^{-1}\mathbf{s}. \tag{4.16}$$

The advantage of this form is that $\left[\mathbf{I} + \mathbf{W}^T\mathbf{K}_\mathbf{n}^{-1}\mathbf{W}\right]$ is a $J \times J$ matrix, where $J \ll M$ in practice. Thus, the direct inverse of $\left[\mathbf{I} + \mathbf{W}^T\mathbf{K}_\mathbf{n}^{-1}\mathbf{W}\right]$ can be implemented easily. The inverse of $\mathbf{K}_\mathbf{n}$ is trivially solved since $\mathbf{K}_\mathbf{n}$ is a diagonal matrix.

4.3 Image compression method

The above method is easily implemented if J is small. However this method works only if $\mathbf{K}$ is diagonally dominated; otherwise, there is a large error in estimating the covariance matrix and its inverse even though it can be inverted. So we can use this method only if the measurement Poisson noise dominates over the randomness of the background.

Now we consider the situation in which J is so large that direct inversion of the $J \times J$ matrix is not feasible (J is still much less than M in practice). We can use image compression techniques to reduce the dimension. Given a set of simulated backgrounds $\{\mathbf{g}_j,\ j = 1, ..., J\}$, we can find an $M \times K$ transform matrix $\mathbf{E}$ such that

$$\mathbf{g}_j \approx \mathbf{E}\mathbf{v}_j, \tag{4.17}$$

where $\mathbf{v}_j$ is a $K \times 1$ vector, and $K < J \ll M$. After subtracting the sample mean from each $\mathbf{v}_j$ we have a new data set, $\mathbf{V} = \frac{1}{\sqrt{J}}\left[\delta\mathbf{v}_2, ..., \delta\mathbf{v}_J\right]$. The sample covariance matrix is approximated by

$$\widehat{\mathbf{K}}_\mathbf{b} \approx \mathbf{E}\left[\mathbf{V}\mathbf{V}^T\right]\mathbf{E}^T = \mathbf{E}\widehat{\mathbf{K}}_\mathbf{v}\mathbf{E}^T. \tag{4.18}$$

This expression resembles the eigenvalue-decomposition (EVD) of the covariance matrix. By the Woodbury matrix-inversion lemma,

$$\left[\mathbf{K}_\mathbf{n} + \mathbf{E}\widehat{\mathbf{K}}_\mathbf{v}\mathbf{E}^T\right]^{-1} = \mathbf{K}_\mathbf{n}^{-1} - \mathbf{K}_\mathbf{n}^{-1}\mathbf{E}\left[\widehat{\mathbf{K}}_\mathbf{v} + \mathbf{E}^T\mathbf{K}_\mathbf{n}^{-1}\mathbf{E}\right]^{-1}\mathbf{E}^T\mathbf{K}_\mathbf{n}^{-1}. \tag{4.19}$$

The advantage of this form is that $\left[\widehat{\mathbf{K}}_{\mathbf{v}}+\mathbf{E}^T\mathbf{K}_{\mathbf{n}}^{-1}\mathbf{E}\right]$ is a $K \times K$ matrix and K does not depend on the number of samples, but rather on the compression ratio.

Next we will discuss how to design the transform matrix $\mathbf{E}$. Recently, the discrete wavelet transform (DWT) has been used in image compression. For example, the JPEG 2000 standard (http://www.jpeg.org/JPEG2000.htm) which uses the DWT shows significant advantages over the traditional JPEG version which uses the block cosine transform. The energy of the wavelet coefficients concentrates in large scale space. Thus we may select the wavelet coefficients as the components of $\mathbf{v}$ according to the scale parameters. If we want to further reduce the size of $\mathbf{v}$, we can apply the eigenvalue-decomposition (EVD) on the covariance matrix of the selected wavelet coefficients. Assuming the number of the selected wavelet coefficients is N, we define an $M \times N$ matrix $\mathbf{B}$ whose columns are the N wavelet basis functions for the selected coefficients, and we also define a $N \times K$ matrix $\mathbf{U}$ whose columns are the K eigenvectors corresponding to the largest eigenvalues of the EVD. The transform matrix is

$$\mathbf{E} = \mathbf{B}_{M\times N}\mathbf{U}_{N\times K}. \tag{4.20}$$

Example Given a set of 64×64 images , we can apply the DWT on these images up to the second scale level, resulting in LL_2, LH_2, HL_2, HH_2, LH_1, HL_1, HH_1 subband images in the wavelet domain, where the subscripts stand for the scale parameters. The subband images in the second scale level are 16×16 pixels and the subband images in the first scale level are 32×32 pixels. If we keep LL_2, LH_2, HL_2, HH_2 then we have $N = 1024$ coefficients whose covariance matrix is 1024×1024, which is smaller than the 4096×4096 element covariance matrix of the input image. We can implement the EVD and keep $K = 100$ eigenvectors (or eigenimages reshaped from the eigenvectors). We illustrate these eigenimages in Fig. 4.1 for the set of training images in the following simulation example.

We have simulated 1000 type 2 lumpy backgrounds as the training images. If

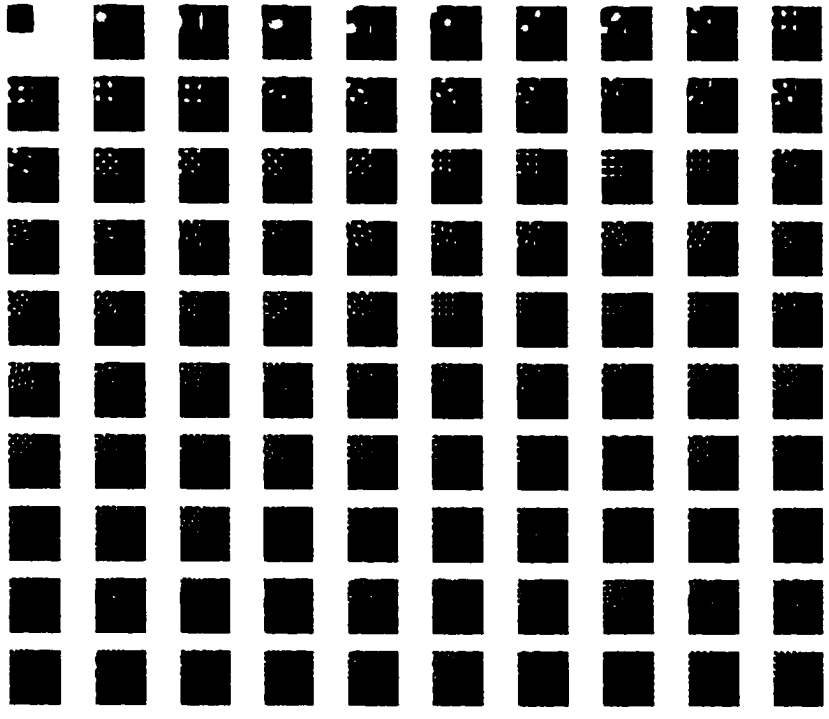


FIGURE 4.1. Eigenimages of the wavelet covariance matrix estimated by 1000 simulated type 2 lumpy backgrounds.

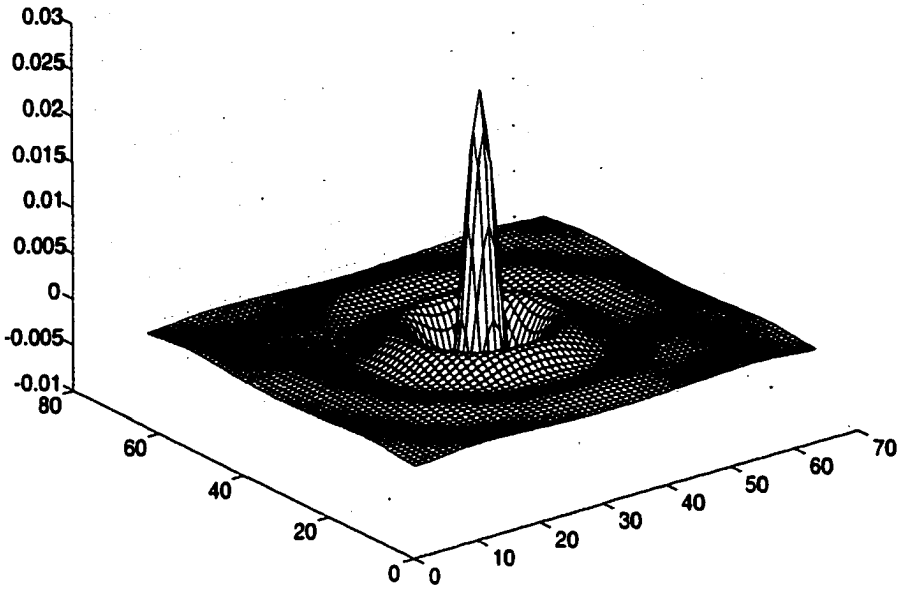


FIGURE 4.2. The Hotelling template estimated by DFT, estimated $\mathrm{SNR}^2 = 10.374$.

we use the matrix-inversion lemma method then we need to calculate the inverse of $\left[\mathbf{I} + \mathbf{W}^T\mathbf{K}_n^{-1}\mathbf{W}\right]$, which is 1000×1000; but if we use the image compression method we can approximate the image with a much smaller number of wavelet coefficients, say, $K = 100$. Then we only need to calculate the inverse of $\left[\widehat{\mathbf{K}}_v + \mathbf{E}^T\mathbf{K}_n^{-1}\mathbf{E}\right]$, which is 100×100. We set the parameters of the type 2 lumpy background as : $W(0) = 10^8, r_b = 10, \overline{B} = 10^3$. For the type 2 lumpy background, the DFT is the eigenvalue-decomposition of the covariance matrix, thus we can get the ideal template by the DFT method. We estimate the Hotelling templates $\widehat{\mathbf{w}}$ by both the DFT and the image compression method, and illustrate them in Figs. 4.2 and 4.3. We also estimate the SNR^2 by $\mathrm{SNR}^2 = \widehat{\mathbf{w}}^T\mathbf{s}$. The result shows that the estimated SNR^2 is approximately the same as the true value.

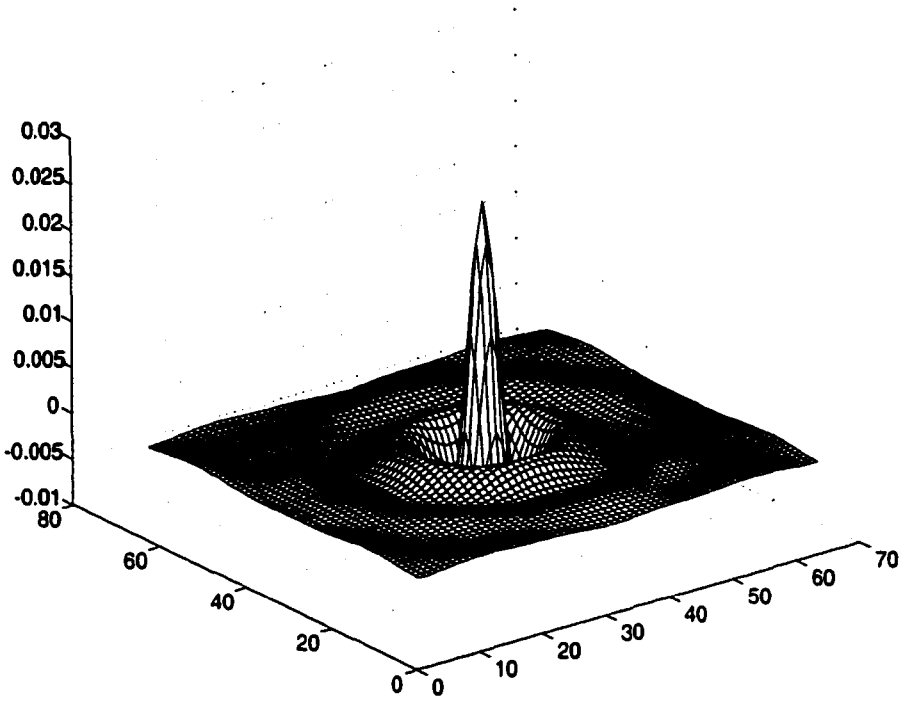


FIGURE 4.3. The Hotelling template estimated using the image compression method, estimated $\mathrm{SNR}^2 = 10.459$.

4.4 Channelized Hotelling observer

The above two methods give the estimated $\mathbf{K}^{-1}$ first, then calculate the Hotelling template by $\mathbf{w} = \mathbf{K}^{-1}\mathbf{s}$. Here we consider how to estimate the Hotelling template directly. The strategy is to transform the image into a feature vector by

$$\mathbf{v} = \mathbf{F}^T\mathbf{g}, \tag{4.21}$$

where $\mathbf{F} = \{\mathbf{f}_1, ..., \mathbf{f}_K\}$ is an $M \times K$ matrix whose columns are of the various channel filters, thus the k^{th} component of $\mathbf{v}$ is $v_k = \mathbf{f}_k^T\mathbf{g}$. The test statistic is a scalar value, and is calculated by the linear function of the feature vector, i.e., the inner product of the weight vector $\mathbf{u}$ and $\mathbf{v}$

$$\lambda = \mathbf{u}^T\mathbf{v} = (\mathbf{F}\mathbf{u})^T\mathbf{g}. \tag{4.22}$$

Comparing (4.22) to (4.6), we actually estimate the Hotelling template by

$$\mathbf{w}_c = \mathbf{F}\mathbf{u}. \tag{4.23}$$

In the sense of maximizing the signal-to-noise ratio, the optimal linear combination rule is to apply the Hotelling observer strategy to the channel responses

$$\mathbf{u} = \mathbf{K}_c^{-1}\mathbf{s}_c, \tag{4.24}$$

where $\mathbf{s}_c = \mathbf{F}^T\mathbf{s}$ and $\mathbf{K}_c = \mathbf{F}^T\mathbf{K}\mathbf{F}$. The size of $\mathbf{K}_c$ is $K \times K$, so we avoid the computational difficulties of the Hotelling observer. However the information is lost inevitably in the formation of the channel responses and leads to suboptimal performance.

The Hotelling template is a linear combination of the channel filters, so $\mathbf{w}_c$ lies in the range space of $\mathbf{F}$. The selection of the channel filters is important for the success of this method. We know from Rolland's work [Rolland 1990] on lumpy backgrounds that the Hotelling observer will look at rather high spatial frequencies, especially if the measurement noise is low, so bandwidth restriction may be dangerous, and

we need to let the channel filters cover the whole spatial frequencies. We also note that humans process visual information through spatial-frequency-selective channels, which are localized in the spatial and frequency domain as being discussed in the previous chapters. These kinds of channel filters have been used for feature extraction by researchers [Mallat 1989a], [Myers 1987], [Strickland 1996]. The localization in the spatial domain is very attractive for the SKE problem because the Hotelling template is always centered around the signal position, so we want these channel filters also to concentrate around the signal. There are other kinds of channel filters. For example, Barrett et al. [Barrett 1998b] suggested the expansion of the Hotelling template by Laguerre-Gauss functions when the background and signal are rotationally symmetric. This suggestion is based on the fact that the template is a rotationally symmetric function in this case, and the Laguerre-Gauss functions are orthonormal on the radial axis. The spatial support of the Laguerre-Gauss functions is decided by the standard deviation of the Gaussian. Next we will give several examples of the channel filters; they are (1) the difference-of-Gaussian (DoG) functions, (2) Battle-Lemarie wavelet functions (3) Laguerre-Gauss functions. Note that we are not using these channel filters to emulate the human visual system, but to approximate the Hotelling observer.

4.4.1 Difference-of-Gaussian (DoG) filters

The DoG filters are created by subtracting two Gaussian functions of different widths

$$DoG\,(r,\theta|\sigma_1,\sigma_2) = \frac{\sqrt{2}}{\sigma_1}\exp\left(-\frac{\pi r^2}{\sigma_1^2}\right) - \frac{\sqrt{2}\lambda}{\sigma_2}\exp\left(-\frac{\pi r^2}{\sigma_2^2}\right),\ \sigma_1 < \sigma_2 \qquad (4.25)$$

where λ is adjusted to let these two Gaussin functions have the same average values, so that their difference has zero DC value. There are usually two ways to alter the DoG filter characteristics by changing (σ_1,σ_2). One is to hold the larger Gaussian fixed, and vary the size of the smaller Gaussian, the resulting DoG filters have more bandwidth

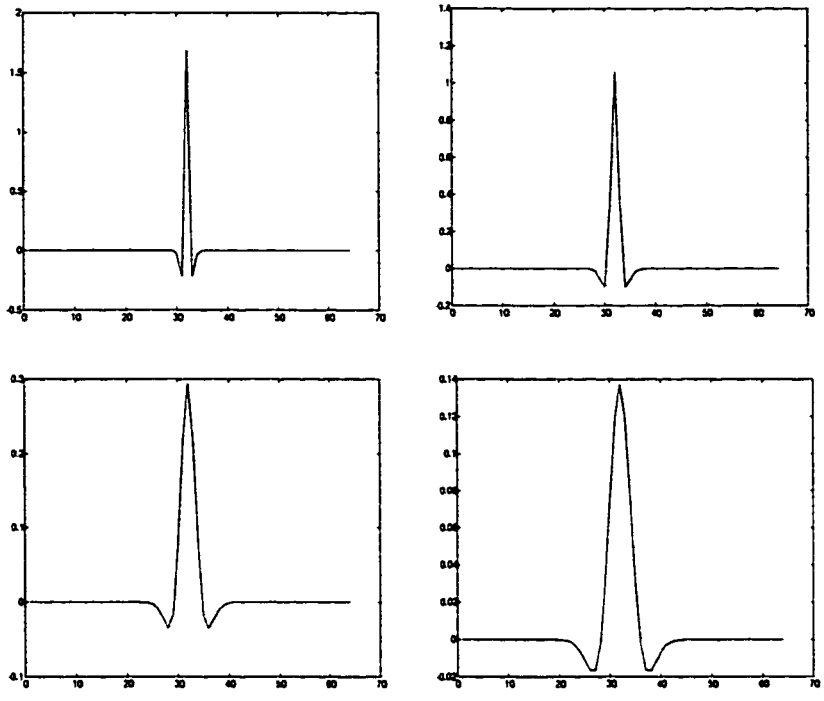


FIGURE 4.4. Spatial profiles of DoG filters

as the ratio $\frac{\sigma_2}{\sigma_1}$ increases. The second way to change the DoG filter characteristics is to hold $\frac{\sigma_2}{\sigma_1}$ constant, while allowing the overall size of the filter to vary as shown in Fig. 4.4. In this case the resulting DoG filters occupy the lower frequency region as the filter size increases. The ratio $\frac{\sigma_2}{\sigma_1}$ is often selected as 2, resulting in a logarithmic set of bandwidths as illustrated in the top graph of Fig. 4.5. However, we have found that the Hotelling template requires more frequency resolution in the mid-frequency band. So we add more channels in the mid-frequency region as illustrated in the bottom graph of Fig. 4.5. This change results in a significant difference in the Hotelling template and SNR.

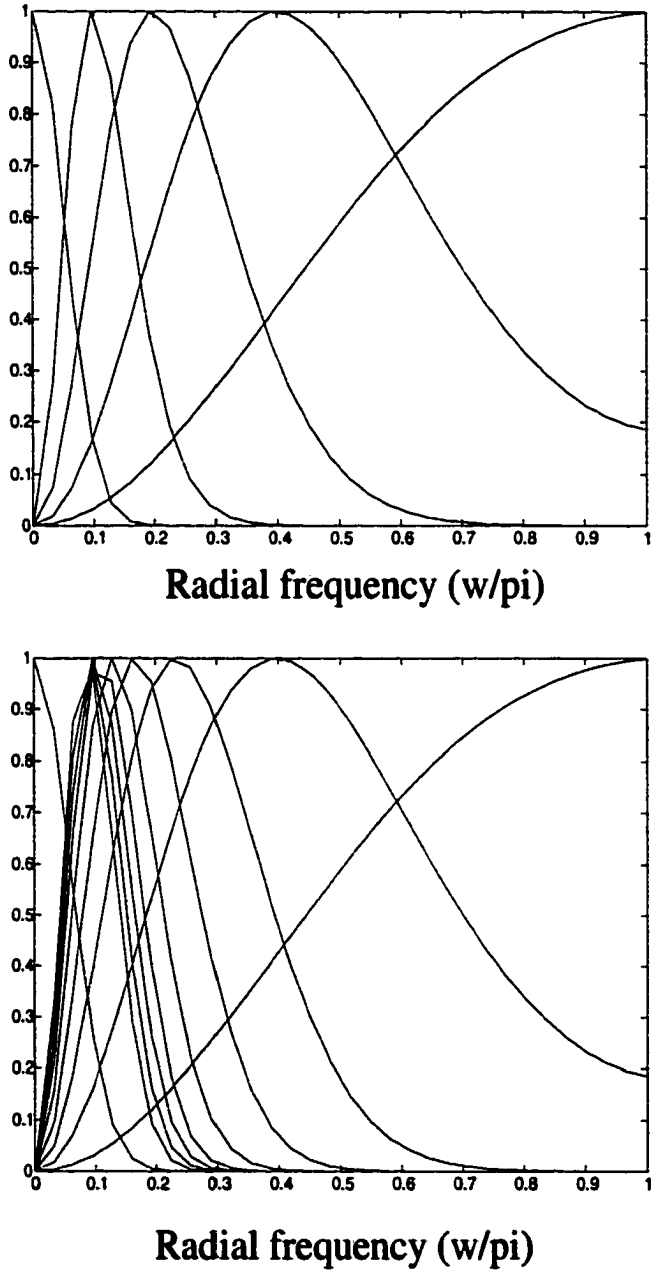


FIGURE 4.5. Frequency profiles of DoG filters.

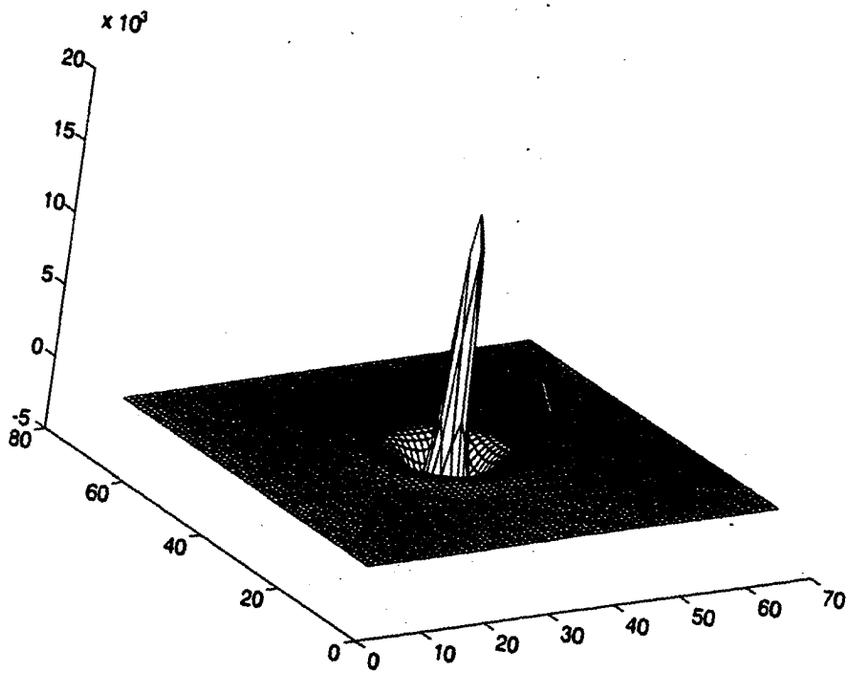


FIGURE 4.6. Hotelling template constructed from 5 DoG filters in the top graph of Fig.4.5, with estimated $SNR^2 = 3.814$.

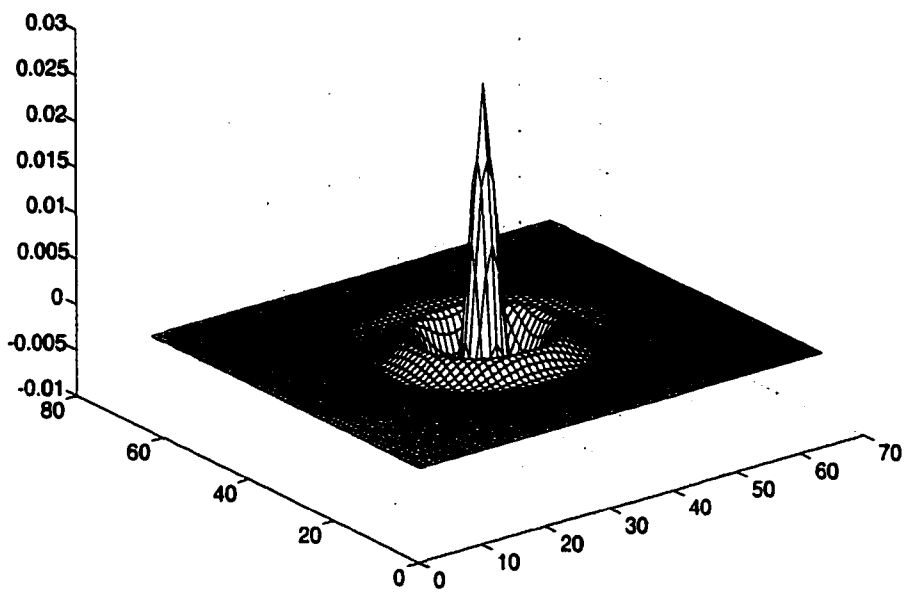


FIGURE 4.7. Hotelling template constructed from the DoG filters with additional bands in the bottom graph of Fig. 4.5. The estimated $SNR^2 = 10.189$ which is much higher than the estimated SNR^2 without additional bands.

4.4.2 Orthonormal wavelet filters

We have discussed orthonormal wavelet bases in chapter 2. Now we can use them as the channel filters in estimating the Hotelling template. Due to their attractive properties such as the localization in the spatial and frequency domains, many researchers have used them for edge detection and feature extraction. For example, Strickland and Hahn [Strickland 1996] used the biorthogonal spline wavelets for multiscale matched filtering in mammograms.

We here select the polynomial spline wavelets, also called the Battle-Lemarie wavelets after their inventors [Battle 1987], [Lemarie 1988]. Figs. 4.8 and 4.9 show examples of the basis functions in V_i and W_i. These figures are generated by using the two-scale relations in chapter 2 for given the filter coefficients. We also give the two-dimensional basis functions in Fig. 4.10. Since the two-dimensional wavelet bases are not rotationally symmetric, the estimated Hotelling template is also not rotationally symmetric and has less SNR^2.

4.4.3 Laguerre-Gauss (LG) filters

Now we consider a special case where both the signal and the background covariance are rotationally symmetric. In this case the template of the Hotelling observer is rotationally symmetric, so rotationally symmetric orthonormal functions are the best selection for the channel filters. Barrett et al. [Barrett 1998b] used the Laguerre-Gauss functions as the channel filters, which are the products of Laguerre polynomials and Gaussians.

The n-order Laguerre polynomial is defined by

$$L_n(x) = \sum_{m=0}^{n} (-1)^m \binom{n}{m} \frac{x^m}{m!}. \tag{4.26}$$

The Laguerre polynomials are orthonormal on $(0, \infty)$ with respect to an exponential

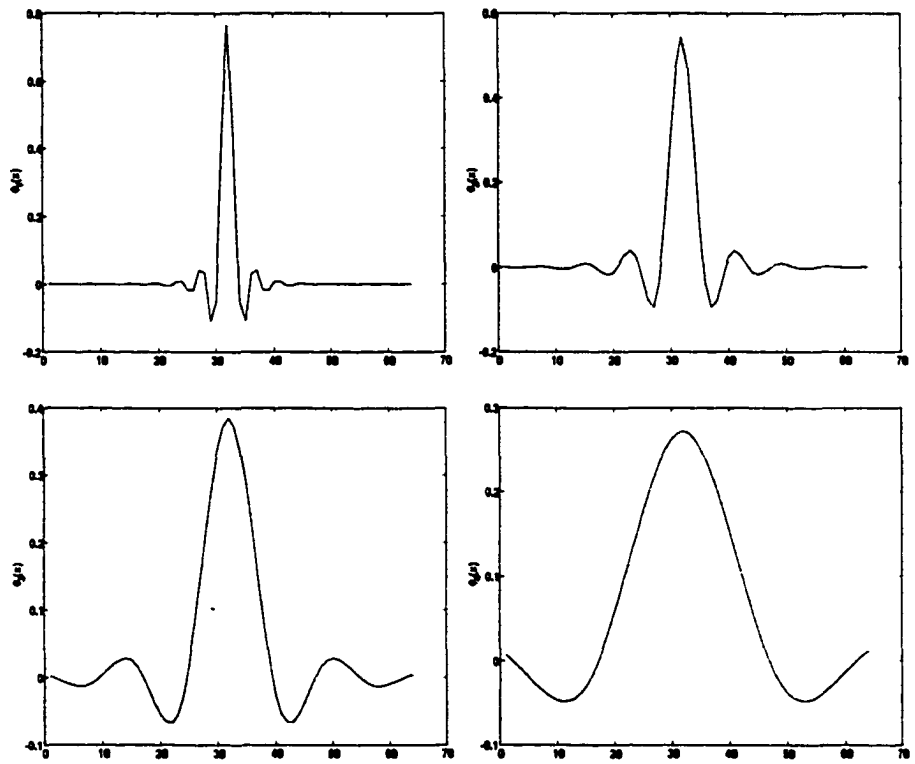


FIGURE 4.8. The Battle-Lemarie basis functions in V_i =span$\{\phi_{i,l}\}$.

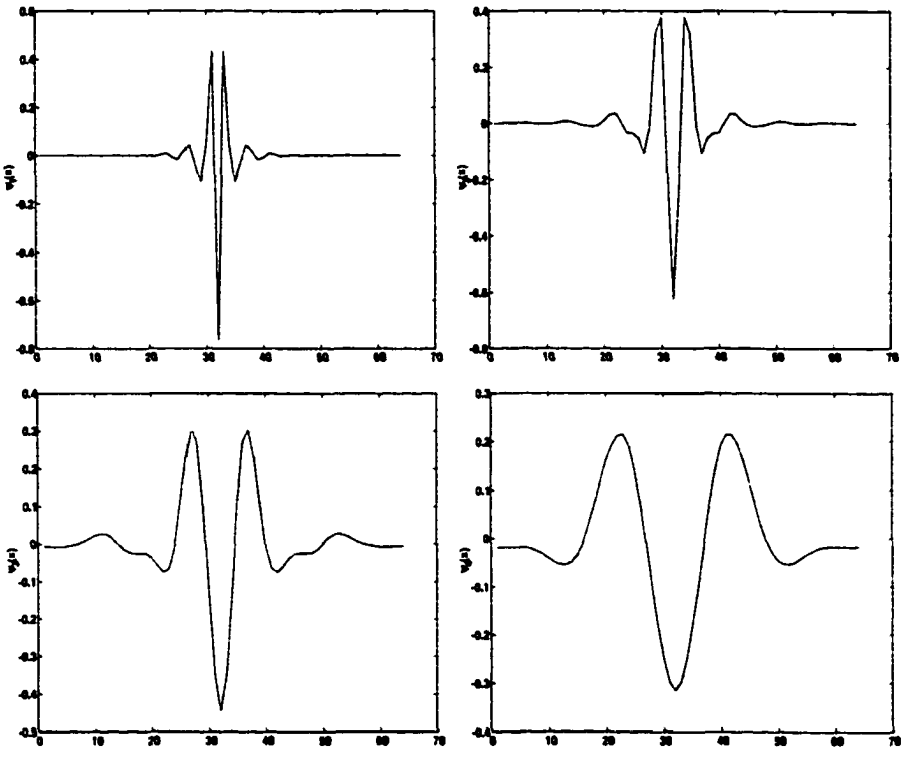


FIGURE 4.9. The Battle-Lemarie basis functions in W_i =span$\{\psi_{i,l}\}$.

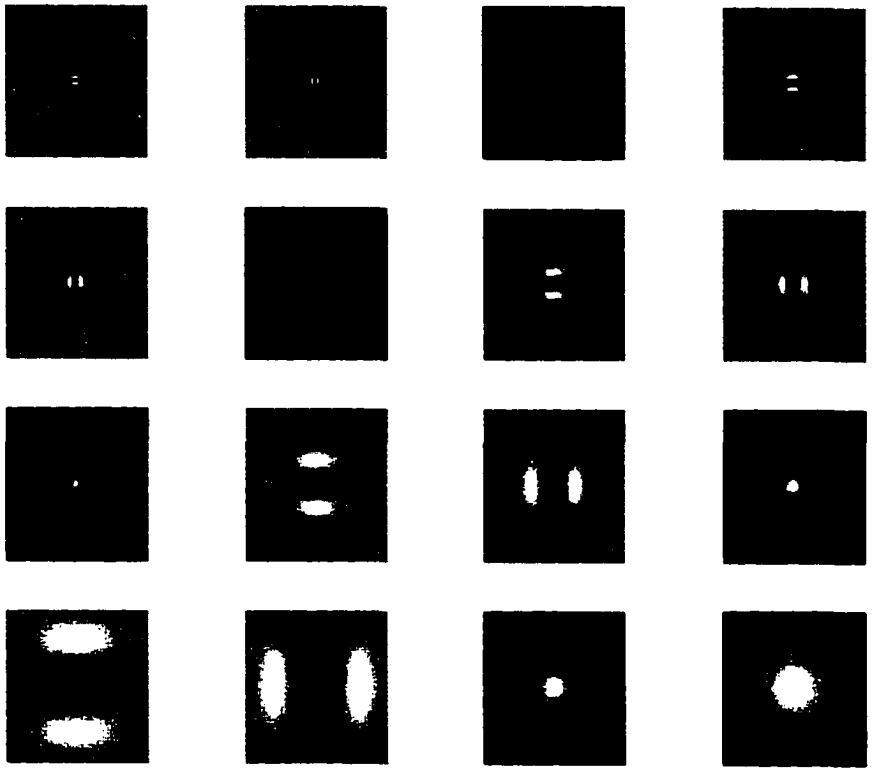


FIGURE 4.10. Two-dimensional Battle-Lemarie wavelet bases.

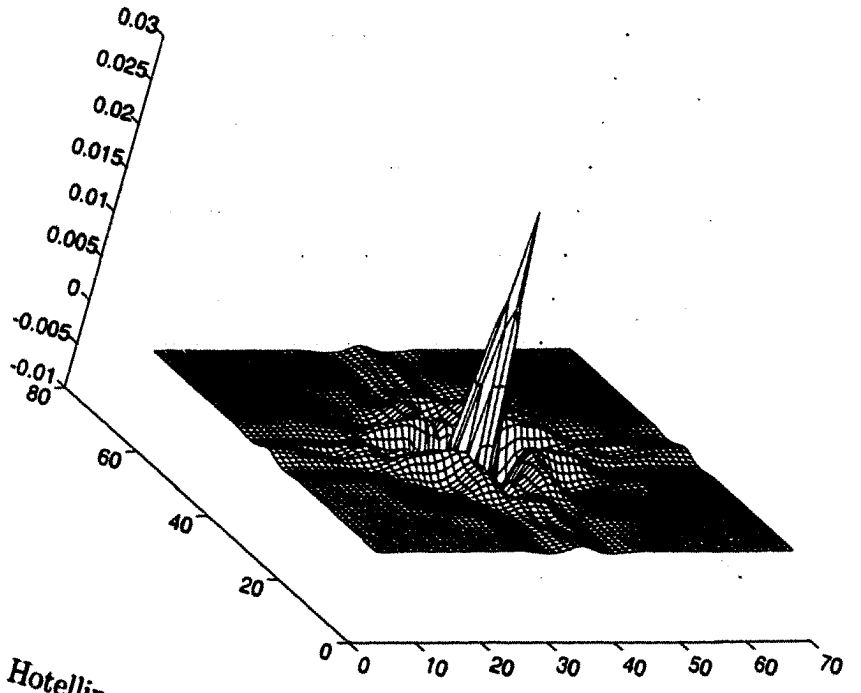


FIGURE 4.11. Hotelling template constructed using 16 Battle-Lemarie wavelet bases, with estimated $SNR^2 = 7.64$.

weight factor

$$\int_0^\infty dx e^{-x} L_n(x) L_m(x) = \delta_{nm}. \tag{4.27}$$

The change of variables $x = 2\pi r^2/a^2$ yields a new orthonormal family, called Laguerre-Gauss functions, satisfying

$$\int_0^\infty r dr \; LG_n(r) LG_m(x) = \delta_{nm}, \tag{4.28}$$

where the n-order Laguerre-Gauss function is defined by

$$LG_n(r) = \frac{2\sqrt{\pi}}{a} \exp\left(-\pi r^2/a^2\right) L_n\left(2\pi r^2/a^2\right). \tag{4.29}$$

We illustrate the first four Laguerre-Gauss functions in Fig. 4.12. We can see from this figure that these functions have almost the same spatial support, which is decided mostly by the parameter a in the Gaussian functions, but the oscillations increase with the order of the Laguerre polynomials.

The Hotelling template is expressed as the expansion of the Laguerre-Gauss functions with the origin at the signal location

$$w(r) = \sum_n u_n \exp\left(-\frac{\pi r^2}{a^2}\right) L_n\left(\frac{2\pi r^2}{a^2}\right), \tag{4.30}$$

where the expansion coefficients $\mathbf{u} = [u_0, u_1, ..., u_{K-1}]$ are estimated by $\mathbf{u} = \mathbf{K}_c^{-1}\mathbf{s}_c$. We give the estimated Hotelling template in Fig. 4.13 by using the first 10 Laguerre-Gauss functions.

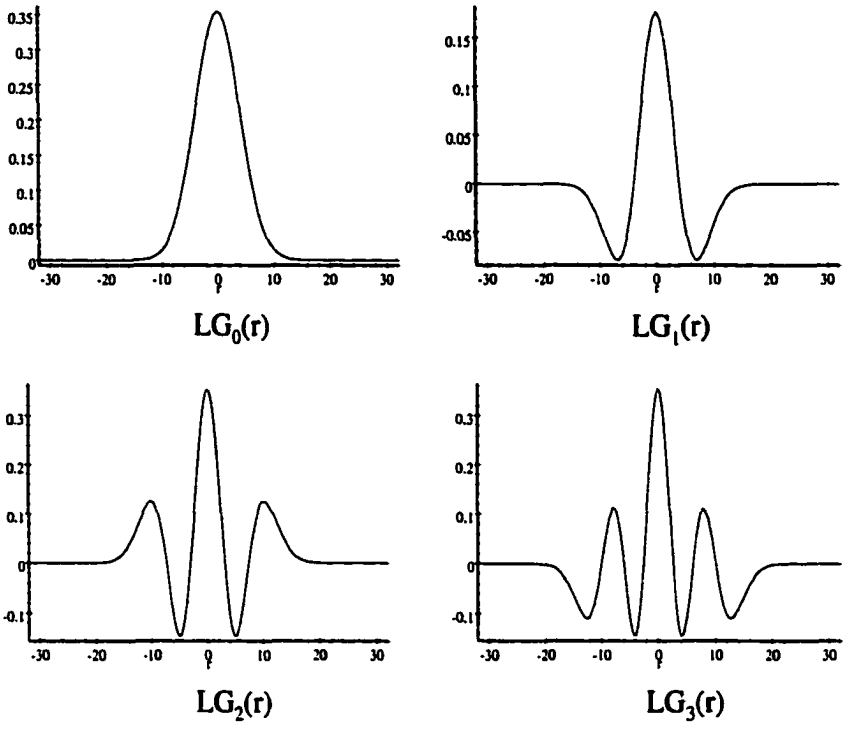


FIGURE 4.12. The first four Laguerre-Gauss functions.

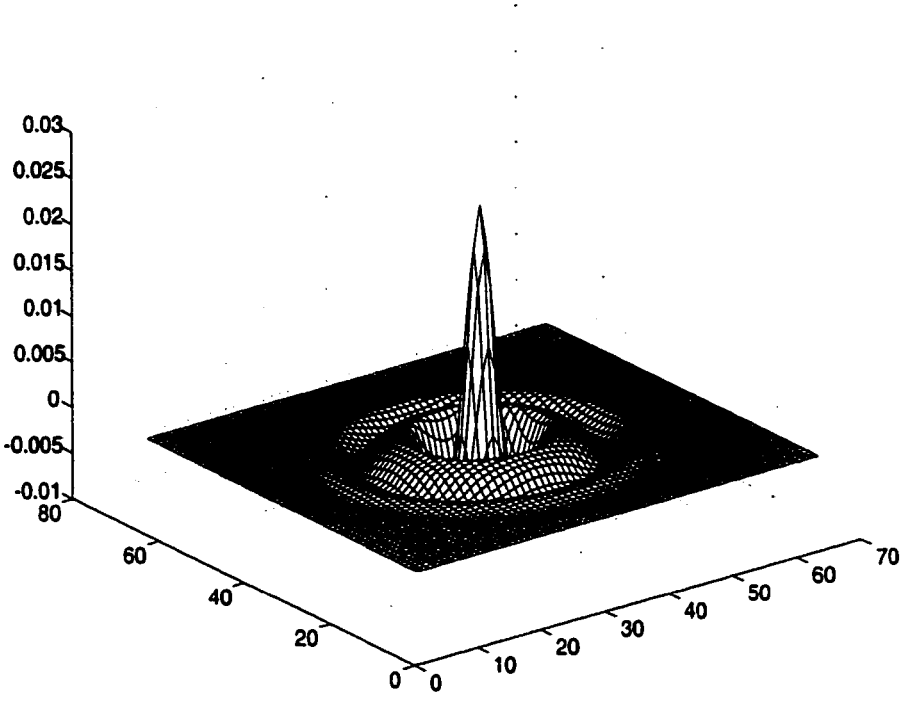


FIGURE 4.13. Hotelling template constructed from10 Laguerre-Gauss functions, with estimated $\mathrm{SNR}^2 = 10.266$.

Chapter 5

IDEAL OBSERVER

5.1 Introduction

We are interested in developing mathematical and computational methods for predicting how a given medical imaging system will perform when the resulting images are used for tumor detection. These computational methods will give us the ability to design and optimize medical imaging systems for maximum performance on tumor-detection tasks, and we define the image quality as the performance of a model observer for the given detection tasks. Among all model observers, the ideal observer sets an upper limit to the performance as measured by the Bayesian risk or receiver operating characteristic (ROC) analysis. Therefore, the current thrust of our research has been in constructing ideal observers and computing their performance for increasingly complex detection tasks [Barrett 1997], [Barrett 1998a], [Clarkson 2000] [Zhang 2001a], [Zhang 2001b]. In this chapter, we present several computational methods to calculate the decision statistic for the ideal observer and estimate its performance.

The classical strategy in the decision process includes three steps. The first step is to choose the features that are most effective for preserving class separability. The objective of the feature-extraction step is dimensionality reduction. Then the second step is to construct a classifier that combines the features into a scalar test statistic. Finally, the third step is to compare this test statistic with a threshold value and make a decision. In the second step of the decision process, conventional *probabilistic classifiers* characterize classes by their probability density functions on the input

features and use Bayes' decision theory to form decision regions from these densities. The form of input distributions is assumed to be known, and parameters are estimated using training data. Adaptive non-parametric *neural-net classifiers* do not estimate or even assume probability density functions, but directly estimate discriminant functions to form decision regions. Richard and Lippmann [Richard 1991] have demonstrated that neural-net classifiers can accurately estimate posterior probabilities (see Appendix B) and provide reduced error rates when compared to conventional probabilistic classifiers, so we adopt using the neural-net classifier following feature extraction.

Since the feature-extraction step reduces the dimensionality, all feature-extraction methods but the ideal observer will lose information in terms of class separability or Bayes error. Therefore we need to study the strategy of the ideal observer and give feature-extraction methods based on the strategy of the ideal observer; then we will train a neural-net classifier to approximate the ideal observer [Zhang 2001a]. We propose two kinds of features. One of the features is the ratio of two conditional density functions of data which are known from the imaging process; another feature is the ratio of two prior density functions of random backgrounds which are unknown. We will give two ways to approximate the second ratio by using a set of training background images. The first method is to apply an invertible transform to "gaussianize" the background image, then calculate the ratio of two Gaussian density functions. The second method is to use maximum-entropy estimates of the prior density functions, then calculate the ratio of these estimates.

5.2 Background

5.2.1 Figures of merit

We have discussed one figure of merit for a binary decision problem, called SNR^2, which is defined by

$$\mathrm{SNR}^2 = \frac{[\langle\lambda\rangle_1 - \langle\lambda\rangle_0]^2}{\frac{1}{2}var_1(\lambda) + \frac{1}{2}var_0(\lambda)}, \tag{5.1}$$

where λ is a test statistic of an observer. However the SNR^2 is usually used to specify the performance of the Hotelling observer, and it is defined only by the mean and variance of the test statistic. Thus it implicitly assumes that the test statistic has a Gaussian distribution. In complex detection tasks, the discriminant function of the ideal observer might be a nonlinear function of the data, and the test statistics of the ideal observer can have non-Gaussian distributions. Therefore, in general the SNR^2 cannot adequately predict the performance of the ideal observer. In this chapter we will give another figure of merit, called the area under the ROC curve (AUC). The maximum value of the AUC, among all observers performing a given detection task and with a given noise model, is set by the ideal observer for that particular task and noise model [Clarkson 2000].

There are four possible outcomes for each individual decision [Van Trees 1968], [Barrett 1998a]. If the decision is signal-present and it really is present, the decision is a true positive (TP), while a decision of signal-present when there is no signal is a false positive (FP). The conditional probability of a positive decision, given that the signal is actually present, is called the true-positive fraction (TPF). Similarly, we have the false-positive fraction (FPF). True negative (TN) and false negative (FN) decisions and their associated fractions (TNF and FNF) are defined in a similar fashion. From basic properties of conditional probabilities, TPF=1-FNF and TNF=1-FPF, so only two of the four fractions are needed to specify the test performance; it is conventional

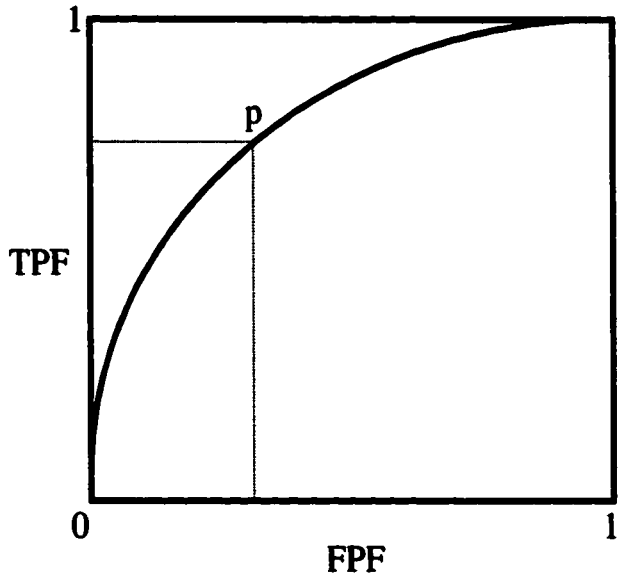


FIGURE 5.1. ROC curve.

to choose TPF and FPF. The TPF and FPF at threshold x are given by

$$\text{TPF}(x) = \int_x^\infty d\lambda\, p(\lambda|H_1), \tag{5.2}$$

$$\text{FPF}(x) = \int_x^\infty d\lambda\, p(\lambda|H_0), \tag{5.3}$$

where $p(\lambda|H_i)$ is the probability density function of the test statistic under H_i. The threshold value controls the trade-off between TPF and FPF. Graphically, this trade-off is portrayed by the ROC curve, which is a plot of TPF(x) vs. FPF(x) as illustrated in Fig. 5.1.

The operating points of the ROC curve are the values of the TPF at some specified FPF (or the probability of detection at a specified false-alarm rate). Since the separation between two classes does not depend on the chosen operating points, many researchers in signal detection and image quality advocate using the entire ROC curve as the quality metric. A common scalar figure of merit is the area under the ROC

curve, denoted by AUC [Barrett 1998a]:

$$\mathrm{AUC} = \int_0^1 \mathrm{TPF}\, d(\mathrm{FPF}), \tag{5.4}$$

where $\mathrm{TPF}(x)$ and $\mathrm{FPF}(x)$ are given by (5.2) and (5.3), respectively. Since FPF is a monotonic function of x, we can exchange the variable of integration from $\mathrm{FPF}(x)$ to x, thus obtaining

$$\mathrm{AUC} = -\int_{-\infty}^{\infty} dx \mathrm{TPF}(x)\, \frac{d}{dx}\mathrm{FPF}(x). \tag{5.5}$$

Substituting (5.2) and (5.3) into the above integral, we have

$$\begin{aligned} \mathrm{AUC} &= \int_{-\infty}^{\infty} dx\, p(x|H_0) \int_x^{\infty} d\lambda\, p(\lambda|H_1) \\ &= \int_{-\infty}^{\infty} dx \int_{-\infty}^{\infty} d\lambda\, p(x|H_0)\, p(\lambda|H_1)\, \mathrm{step}(\lambda - x). \end{aligned} \tag{5.6}$$

From (5.6), we see that the AUC of a test statistic is the expected value of step $(\lambda - x)$, where λ and x are the test statistic under H_1 and H_0 respectively. Given samples of the test statistic, we can estimate the AUC by Monte Carlo integration. Eq. (5.6) also shows that the AUC is invariant under a monotonic transformation of the test statistics since $\mathrm{step}(\lambda - x)$ does not change, which is a useful property for a figure of merit because such a transformation produces an observer that is equivalent to the original one [Clarkson 2000].

5.2.2 Ideal observer

The ideal observer, also called the Bayesian observer, for a signal-detection task, makes optimal use of all of the information in the data and any prior knowledge about the images or the imaging process. The performance, as measured by Bayesian risk or the AUC, of the ideal observer sets an upper limit to the performance obtainable by any observer. For a binary detection task, the ideal observer calculates a test statistic called the likelihood ratio or a monotonic function of this statistic, e.g., the

logarithm of the likelihood ratio (log-likelihood). The likelihood ratio is the ratio of two probability densities for the data, one under the hypothesis that the signal is present and the other that the signal is absent

$$\Lambda = \frac{p(\mathbf{g}|H_1)}{p(\mathbf{g}|H_0)}. \tag{5.7}$$

The two hypotheses are specified as

$$H_0 : \mathbf{g} = \mathbf{b} + \mathbf{n}, \qquad H_1 : \mathbf{g} = \mathbf{b} + \mathbf{n} + \mathbf{s}, \tag{5.8}$$

where the background vector $\mathbf{b} = \mathcal{H}\mathbf{f}$ if there is no signal and $\mathbf{b} + \mathbf{s} = \mathcal{H}\mathbf{f}$ if the signal is present. Under H_0, the probability density of the data vector, $p(\mathbf{g}|H_0)$, is then found by averaging over the random backgrounds,

$$p(\mathbf{g}|H_0) = \int p(\mathbf{g}|\mathbf{b})\ p(\mathbf{b}) d\mathbf{b}. \tag{5.9}$$

The corresponding probability density of the data vector under H_1 is

$$p(\mathbf{g}|H_1) = \int p\left(\mathbf{g}|\mathbf{b}, \mathbf{s}\right)\ p\left(\mathbf{b}\right) d\mathbf{b}. \tag{5.10}$$

As we have said in the previous chapter, the Poisson model is valid for medical imaging with x rays or gamma rays. We assume the probability density function of g given $\mathbf{b}$, denoted $p(\mathbf{g}|\mathbf{b})$, is the Poisson distribution:

$$p(\mathbf{g}|\mathbf{b}) = \prod_{m=1}^{M} \frac{b_m^{g_m} \exp\left(-b_m\right)}{g_m!}. \tag{5.11}$$

Implementation of the ideal observer is, however, difficult because the calculation of the likelihood ratio is not analytically tractable except for a few stylized problems, such as the detection of a known signal on a known background and a known signal on a random background with Gaussian distributions. The first task, referred to as SKE/BKE (signal known exactly, background known exactly) [Barrett 1997], is not very representative of clinical detection problems. In the second task the Hotelling

observer is an ideal observer, and we have studied ways to estimate the Hotelling template in the previous chapter. In more realistic problems, in which the random background has a non-Gaussian distribution, the probability density of the image data is often a high-dimensional integral without a closed form, so the likelihood ratio is difficult to calculate. More than that, we often do not know the statistical distribution of the random background, but instead have a set of training images.

5.3 The strategy of the ideal observer

The likelihood ratio used by the ideal observer is itself an ideal feature for classification; it maps $\mathbf{g}$ from an M-dimensional data space to a scalar test statistic λ. The design of a feature-extraction method can be based on an explanation of the ideal-observer's strategy. We have already shown that the ideal observer calculates the ratio of probability densities under each hypothesis. This ratio, however, has little practical use since the probability densities are difficult to estimate for high-dimensional data spaces. Thus, we need to express the likelihood ratio or the log-likelihood ratio in terms of some functions which are known or can be approximated. These new expressions for the ideal observer's discriminant function tell us the strategy of the ideal observer in signal-detection tasks. We will discuss these strategies when we have to decide whether the input data has a known signal added on the random background in the following three examples.

5.3.1 Example 1: Hotelling discriminant function

We first discuss the strategy of the ideal observer when $\mathbf{g}$ is a multivariate Gaussian random variable. In this case, the Hotelling observer is an ideal observer. The

Hotelling observer calculates the test statistic

$$\lambda_{\text{Hotelling}} = \mathbf{w}^T(\mathbf{g} - \langle \mathbf{g} \rangle) = \mathbf{s}^T\mathbf{K}^{-1}(\mathbf{g} - \langle \mathbf{g} \rangle). \tag{5.12}$$

We already know from chapter 4 that the covariance matrix of $\mathbf{g}$ is related to the covariance matrix of the noise vectors $\mathbf{n}$ and the covariance matrix of the object $\mathbf{f}$ by

$$\mathbf{K} = \mathbf{K}_n + \mathcal{H}\mathbf{K}_f\mathcal{H}^T. \tag{5.13}$$

By making use of a matrix-inversion formula [Harville 1997], we can express the inverse of $\mathbf{K}$ as

$$\begin{aligned} \mathbf{K}^{-1} &= \left(\mathbf{K}_n + \mathcal{H}\mathbf{K}_f\mathcal{H}^T\right)^{-1} \\ &= \mathbf{K}_n^{-1} - \mathbf{K}_n^{-1}\mathcal{H}(\mathbf{K}_f^{-1} + \mathcal{H}^T\mathbf{K}_n^{-1}\mathcal{H})^{-1}\mathcal{H}^T\mathbf{K}_n^{-1}. \end{aligned} \tag{5.14}$$

If we substitute (5.14) into the Hotelling discriminant function, then

$$\lambda_{\text{Hotelling}} = \mathbf{s}^T\mathbf{K}_n^{-1}(\mathbf{g} - \langle \mathbf{g} \rangle) - \mathbf{s}^T\mathbf{K}_n^{-1}\mathcal{H}(\mathbf{K}_f^{-1} + \mathcal{H}^T\mathbf{K}_n^{-1}\mathcal{H})^{-1}\mathcal{H}^T\mathbf{K}_n^{-1}(\mathbf{g} - \langle \mathbf{g} \rangle). \tag{5.15}$$

The Wiener estimator, which is also the maximum a posteriori (MAP) estimator for Gaussian noise models and Gaussian background variations, is [Barrett 1990]:

$$\widehat{\mathbf{f}}_{\text{map}} = (\mathbf{K}_f^{-1} + \mathcal{H}^T\mathbf{K}_n^{-1}\mathcal{H})^{-1}\mathcal{H}^T\mathbf{K}_n^{-1}(\mathbf{g} - \langle \mathbf{g} \rangle) + \langle \mathbf{f} \rangle. \tag{5.16}$$

Now we see that the Hotelling discriminant function can be expressed as

$$\begin{aligned} \lambda_{\text{Hotelling}} &= \mathbf{s}^T\mathbf{K}_n^{-1}(\mathbf{g} - \langle \mathbf{g} \rangle) - \mathbf{s}^T\mathbf{K}_n^{-1}\mathcal{H}(\widehat{\mathbf{f}}_{\text{map}} - \langle \mathbf{f} \rangle) \\ &= \mathbf{s}^T\mathbf{K}_n^{-1}(\mathbf{g} - \mathcal{H}\widehat{\mathbf{f}}_{\text{map}}). \end{aligned} \tag{5.17}$$

From this equation, we see that the ideal observer first estimates the random background by a Wiener estimator or a maximum a posteriori estimator, subtracts it from $\mathbf{g}$, and then implements a prewhitened matched filter based on the covariance of the measurement noise only.

5.3.2 Example 2: General ideal observer

We now discuss the more general situation in which the image $\mathbf{g}$ is a non-Gaussian random vector. The ideal observer calculates the ratio of two density functions under each hypothesis:

$$\begin{aligned}\Lambda_{\text{Ideal}} &= \frac{p(\mathbf{g}|H_1)}{p(\mathbf{g}|H_0)} \\ &= \frac{\int p(\mathbf{g}|\mathbf{b},\mathbf{s})\; p(\mathbf{b})d\mathbf{b}}{\int p(\mathbf{g}|\mathbf{b})\; p(\mathbf{b})d\mathbf{b}} \\ &= \int \frac{p(\mathbf{g}|\mathbf{b},\mathbf{s})}{p(\mathbf{g}|\mathbf{b})} \left[\frac{p(\mathbf{g}|\mathbf{b})\; p(\mathbf{b})}{\int p(\mathbf{g}|\mathbf{b})\; p(\mathbf{b})d\mathbf{b}}\right] d\mathbf{b}. \end{aligned} \tag{5.18}$$

By Bayes' theorem, the second factor in the final integral (in brackets) is equal to the posterior density $p(\mathbf{b}|\mathbf{g})$ of the random background given the image $\mathbf{g}$:

$$p(\mathbf{b}|\mathbf{g}) = \frac{p(\mathbf{g}|\mathbf{b})p(\mathbf{b})}{\int p(\mathbf{g}|\mathbf{b})p(\mathbf{b})d\mathbf{b}}. \tag{5.19}$$

We define the first factor in the integral as

$$\Lambda_{\text{bke}}(\mathbf{b}) = \frac{p(\mathbf{g}|\mathbf{b},\mathbf{s})}{p(\mathbf{g}|\mathbf{b})}, \tag{5.20}$$

where Λ_{bke} (bke stands for background known exactly) is the likelihood ratio of two probability densities assuming that the background $\mathbf{b}$ is given. Finally, the likelihood ratio is the posterior mean of Λ_{bke}:

$$\Lambda_{\text{Ideal}} = \int \Lambda_{\text{bke}}(\mathbf{b})p(\mathbf{b}|\mathbf{g})d\mathbf{b}. \tag{5.21}$$

Note that the calculation of Λ_{bke} requires only the probability densities for the data with a given background $\mathbf{b}$. These densities are often known from our prior knowledge of the imaging process.

Example 1 is a special case of this general situation, since, for Gaussian distributions, the posterior mean of Λ_{bke} is the same as the value of $\Lambda_{\text{bke}}(\widetilde{\mathbf{b}})$ when $\widetilde{\mathbf{b}}$ is the maximum a posteriori estimator for the random background. The equation

$$\Lambda_{\text{bke}}(\widetilde{\mathbf{b}}) = c\exp\left[\mathbf{s}^T\mathbf{K}_{\mathbf{n}}^{-1}(\mathbf{g}-\widetilde{\mathbf{b}})\right] \tag{5.22}$$

holds when the noise is a Gaussian random vector for a fixed background, where c is a data-independent constant. The ideal observer can also implement the log-likelihood ratio

$$\lambda_{\text{Ideal}} = \log \Lambda_{\text{bke}}(\tilde{\mathbf{b}}) = \mathbf{s}^t \mathbf{K}_{\mathbf{n}}^{-1}(\mathbf{g} - \tilde{\mathbf{b}}) + \log c. \tag{5.23}$$

Thus the ideal observer for Gaussian measurement noise and a Gaussian background density is equivalent to the Hotelling observer.

5.3.3 Example 3: Ideal observer for additive signal

We will now discuss the case when the signal $\mathbf{s}$ is additive on the background, i.e., $\mathbf{g} = \mathbf{b} + \mathbf{s} + \mathbf{n}$. When a signal is present, the probability density function of the image data $\mathbf{g}$ is

$$\begin{aligned} p(\mathbf{g}|H_1) &= \int p(\mathbf{g}|\mathbf{b}', \mathbf{s})\, p_{\mathbf{b}}(\mathbf{b}') d\mathbf{b}' \\ &= \int p(\mathbf{g}|\mathbf{b}'+\mathbf{s})\, p_{\mathbf{b}}(\mathbf{b}') d\mathbf{b}'. \end{aligned} \tag{5.24}$$

Changing the variables of the final integral, $\mathbf{b} = \mathbf{b}'+\mathbf{s}$, we have

$$p(\mathbf{g}|H_1) = \int p(\mathbf{g}|\mathbf{b})\, p_{\mathbf{b}}(\mathbf{b} - \mathbf{s}) d\mathbf{b}. \tag{5.25}$$

The likelihood ratio of the ideal observer can now be calculated as

$$\begin{aligned} \Lambda_{\text{Ideal}} &= \frac{p(\mathbf{g}|H_1)}{p(\mathbf{g}|H_0)} \\ &= \frac{\int p(\mathbf{g}|\mathbf{b})\, p_{\mathbf{b}}(\mathbf{b} - \mathbf{s}) d\mathbf{b}}{\int p(\mathbf{g}|\mathbf{b})\, p_{\mathbf{b}}(\mathbf{b}) d\mathbf{b}} \\ &= \int \frac{p_{\mathbf{b}}(\mathbf{b} - \mathbf{s})}{p_{\mathbf{b}}(\mathbf{b})} \left[\frac{p(\mathbf{g}|\mathbf{b})\, p_{\mathbf{b}}(\mathbf{b})}{\int p(\mathbf{g}|\mathbf{b})\, p_{\mathbf{b}}(\mathbf{b}) d\mathbf{b}} \right] d\mathbf{b}. \end{aligned} \tag{5.26}$$

The second factor in the final integral (in brackets) is once again equal to the posterior density $p(\mathbf{b}|\mathbf{g})$ of the random background given the image $\mathbf{g}$. The first factor in the

final integral is

$$\Lambda_{\text{noisefree}}(\mathbf{b}) = \frac{p_{\mathbf{b}}(\mathbf{b}-\mathbf{s})}{p_{\mathbf{b}}(\mathbf{b})}, \tag{5.27}$$

because this is the likelihood ratio when there is no measurement noise. Thus the ideal observer calculates the posterior mean of $\Lambda_{\text{noisefree}}(\mathbf{b})$ over the random background

$$\Lambda_{\text{Ideal}} = \int \Lambda_{\text{noisefree}}(\mathbf{b})\; p(\mathbf{b}|\mathbf{g}) d\mathbf{b}. \tag{5.28}$$

The calculation of $\Lambda_{\text{noisefree}}$ requires knowledge of the probability density of the background, which is unknown and difficult to estimate. Estimating this probability density from an ensemble of noise-free images is an ill-posed problem because the data space has a high dimensionality. There are two methods to calculate $\Lambda_{\text{noisefree}}$. The first method is to apply an invertible transform to "gaussianize" the image, then calculate the ratio two Gaussian densities. The second method is to create statistical model of the random background by using the maximum entropy principle, which we have discussed in chapter 3. We will discuss both of these methods soon.

5.4 Feature extraction based on ideal observer

From the analysis in the last section, we see that the ideal observer calculates either the posterior mean of $\Lambda_{\text{noisefree}}(\mathbf{b})$ or the posterior mean of $\Lambda_{\text{bke}}(\mathbf{b})$. The calculation of $\Lambda_{\text{bke}}(\mathbf{b})$ requires the conditional probability density of the image data $\mathbf{g}$ given the background $\mathbf{b}$, which is usually known. The calculation of $\Lambda_{\text{noisefree}}(\mathbf{b})$ requires the probability density of the random backgrounds, which is unknown and difficult to estimate. However, if we have a training set of noise-free images (random backgrounds), then we have a method to approximate $\Lambda_{\text{noisefree}}(\mathbf{b})$.

There is another difficulty not yet resolved. How do we calculate the posterior mean? In the next chapter we will use the Markov chain Monte carlo method to estimate the value of the posterior mean. But here we use a neural network classifier

with feature extraction to approximate the discriminant function of the ideal observer. From (5.21) and (5.28), we can see that the ideal observer is using the numbers $\Lambda_{\text{bke}}(\mathbf{b})$ or $\Lambda_{\text{noisefree}}(\mathbf{b})$ with $\mathbf{b}$ close to the posterior mode. This suggests that we compute the estimates of backgrounds which are close to the posterior mode from the data, and then evaluate $\Lambda_{\text{bke}}(\mathbf{b})$ or $\Lambda_{\text{noisefree}}(\mathbf{b})$ to obtain a feature vector. To compute these estimates, we design different low-pass filters, denoted as $\mathbf{h}_n$, $n = 1, 2, ...N$, and apply them to the data. The reason for this approach is that we assume backgrounds are smooth compared with measurement noise and lesions have small sizes; so both lesions and noises are not present after low-pass filtering. For example, the low-pass filters can have Gaussian shapes with different widths. Thus we have different estimators of the posterior mode

$$\mathbf{b}_n = \mathbf{h}_n * \mathbf{g}, \; n = 1, 2, ...N, \tag{5.29}$$

where $*$ represents a 2-D convolution operator. We substitute $\mathbf{b}_n, n = 1, 2, ...N$ into $\Lambda_{\text{bke}}(\mathbf{b})$ and $\Lambda_{\text{noisefree}}(\mathbf{b})$, and we have the components of the feature vector

$$\Lambda_{\text{bke}}(\mathbf{b}_n) \text{ and } \Lambda_{\text{noise free}}(\mathbf{b}_n), \; n = 1, 2, ...N, \tag{5.30}$$

or their log functions

$$\lambda_{\text{bke}}(\mathbf{b}_n) \text{ and } \lambda_{\text{noise free}}(\mathbf{b}_n), \; n = 1, 2, ...N. \tag{5.31}$$

We have tested the effectiveness of these feature-extraction methods by constructing a neural network classifier with two hidden layers. The neural network classifier is trained by the back-propagation algorithm [Haykin 1999]. The structure of the neural network is 10-4-4-1, i.e., 10 input neurons as a feature vector, 4 hidden neurons in the first hidden layer and 4 hidden neurons in the second hidden layer, and one output. The number of training samples is large (6000) in order to reduce the estimation error; that is, the inherent error in the neural network classifier due to the finite size of the training sample. The testing set is 1000 pairs of simulated samples

of images, one set of images has background and measurement noise and the other set of images has background, noise and a known signal.

In the first example, we generate lumpy backgrounds of type 2, which have a pixel mean value of 1000 counts, a correlation length of 10 pixels and a peak power spectrum value of $W(0) = 100,000$. A bounded Gaussian-shaped signal is inserted into the center of the image for one set of images:

$$s(\mathbf{r}) = a \exp\left(-\frac{|\mathbf{r}-\mathbf{r}_0|^2}{r_s^2}\right) \mathrm{cyl}\left(\frac{|\mathbf{r}-\mathbf{r}_0|}{4r_s}\right), \tag{5.32}$$

with $r_s = 3$ pixels and $a = 20$ counts; $\mathrm{cyl}(\frac{r}{d})$ is a cylinder function, $\mathrm{cyl}(\frac{r}{d}) = 1$ when $0 \leq r < \frac{d}{2}$ and 0 otherwise. Finally, independent Poisson noise is added to the background and signal. Fig. 5.2(a) shows the ROC curve of a Hotelling observer (which is approximately an ideal observer in this example) and the neural network classifier using $\lambda_{\mathrm{bke}}(\mathbf{b}_n), n = 1, .., 10$. Both classifiers have similar performance.

In the second example, we generate log-normal random backgrounds by exponentiating the sum of a lumpy background and an uncorrelated Gaussian noise process. The lumpy background has mean of 7 and a correlation length of 12 pixels with $W(0) = 600$. The variance of the uncorrelated Gaussian noise process is 0.01. The signal is also modeled as a Gaussian-shaped function with $r_s = 1$ pixels and $a = 45$ counts. Fig. 5.2(b) shows the ROC curve of two neural network classifiers with input feature vectors calculated by $\lambda_{\mathrm{bke}}(\mathbf{b}_n)$ and $\lambda_{\mathrm{noise\ free}}(\mathbf{b}_n)$, $n = 1, 2, ...10$ respectively, and the ideal observer for noise-free images, that is, one set of images has background and the other set of images has background and a known signal. For these noise-free images, the ideal observer calculates the likelihood ratio of log-normal densities. Since there is no measurement noise, the ROC curve of the ideal observer for noise-free images is an upper-bound for the ROC curve of the ideal observer for images degraded by the measurement noise.

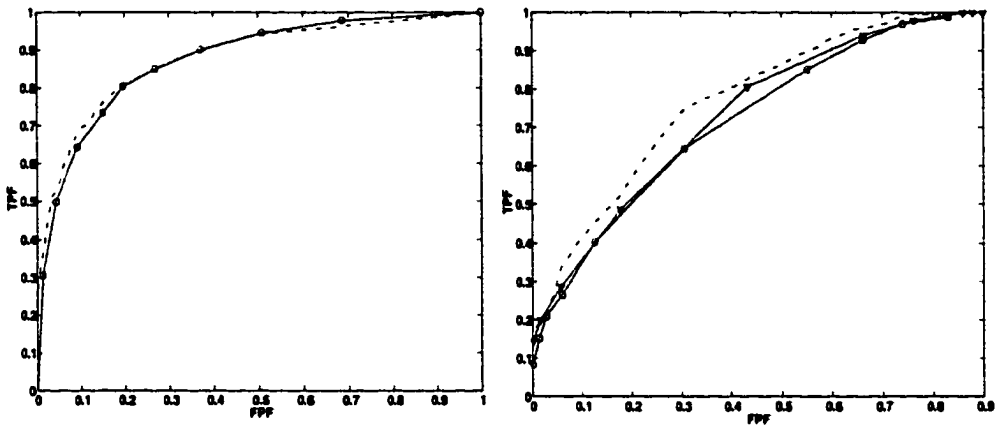


FIGURE 5.2. ROC performance of a neural network classifier with two hidden layers. (o) is the ROC curve by a neural network classifier with input variables $\lambda_{\text{bke}}(\mathbf{b}_n), n = 1, .., 10$. ($\nabla$) is the ROC curve by a neural network classifier with input variables $\lambda_{\text{noisefree}}(\mathbf{b}_n), n = 1, .., 10$. In (a) $\mathbf{b}$ has a Gaussian density and $\mathbf{n}$ is a Poisson noise, the ROC of Hotelling observer (−.) coincides with the ROC of the neural network classifier(o). In (b) $\mathbf{b}$ has a log-normal density and $\mathbf{n}$ is a Poisson noise, the ROC of both neural network classifiers are below the upper bound (−.).

5.5 Nonlinear discriminant analysis I

Now we will discuss the first method for approximating the ratio

$$\Lambda = \frac{p(\mathbf{b}-\mathbf{s})}{p(\mathbf{b})} \tag{5.33}$$

by using a set of training backgrounds. Since the probability density function of the random background is unknown and difficult to estimate, we wish to find an invertible transform such that the new random vector,

$$\mathbf{y} = \mathbf{T}(\mathbf{b}), \tag{5.34}$$

has a multivariate Gaussian density function. Therefore, the probability density function of $\mathbf{b}$ can be expressed by

$$\begin{aligned} p(\mathbf{b}) &= p_Y(\mathbf{y}) \left| \mathbf{J}\left(\frac{y_1,\ldots,y_M}{b_1,\ldots,b_M}\right)\right|, \\ &= p_Y(\mathbf{T}(\mathbf{b})) \left| \mathbf{J}\left(\frac{y_1,\ldots,y_M}{b_1,\ldots,b_M}\right)\right|. \end{aligned} \tag{5.35}$$

Since we restrict $\mathbf{T}$ as an invertible transform, the Jacobian $\mathbf{J}\left(\frac{y_1,\ldots,y_M}{b_1,\ldots b_M}\right) \neq 0$.

By substituting (5.35) into (5.33) and cancelling out the Jacobian terms in both the denominator and the numerator of (5.33), we have

$$\Lambda = \frac{p_Y(\mathbf{T}(\mathbf{b}-\mathbf{s}))}{p_Y(\mathbf{T}(\mathbf{b}))}. \tag{5.36}$$

If we assume $p_Y(.)$ is a Gaussian density function, then we have an analytic expression for the likelihood ratio and the log-likelihood. In terms of the original data vector $\mathbf{b}$, this discriminant is given by

$$\lambda = \log\Lambda = \mathbf{T}(\mathbf{b})^T \mathbf{K}^{-1}\mathbf{T}(\mathbf{b}) - \mathbf{T}(\mathbf{b}-\mathbf{s})^T \mathbf{K}^{-1}\mathbf{T}(\mathbf{b}-\mathbf{s}), \tag{5.37}$$

where $\mathbf{K}$ is the covariance matrix of $\mathbf{y}$.

Next, we will discuss how to design the invertible transform. It includes both linear parts and nonlinear parts; the linear parts use the discrete wavelet transform

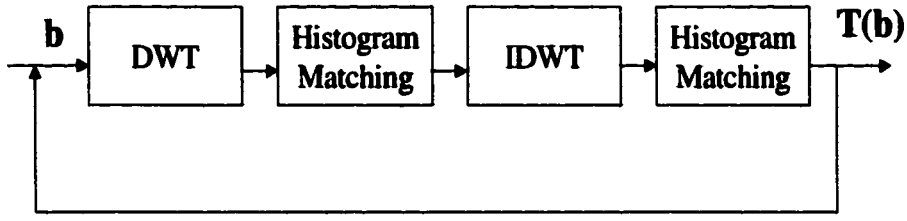


FIGURE 5.3. The invertible transform is an iterative procedure of DWT and IDWT followed by the histogram matching

(DWT) and the inverse discrete wavelet transform (IDWT), and the nonlinear parts use histogram matching. Both procedures are invertible transforms, so they are non-singular transforms.

5.5.1 Wavelet transform

The wavelet transform is an effective way to analyze data structures. For example, independent component analysis (ICA), which optimizes higher-order statistical measures, has been used to construct optimal bases for images. These optimal bases are spatially oriented and have spatial-frequency bandwidths of roughly one octave, which are similar to the wavelet basis functions. By using different wavelet basis functions, we can rotate the coordinates in the data space, and estimate the marginal density on each coordinate. Histogram matching transforms the marginal density on each coordinate to a Gaussian density function. This nonsingular transform method is similar to computer tomography, which reconstructs the object by measuring the projected x-ray counts on each view angle. The object can then be reconstructed by the central-slice theorem. In our case, the object is a multidimensional probability density function, and the view angles correspond to the wavelet basis functions.

This algorithm is also similar to those used for texture synthesis by pyramid-based techniques in chapter 3, which start with an input texture image and a white noise image, then match the histogram of both images after a steerable pyramid transform

[Heeger 1995]. But Heeger et al. use a steerable pyramid transform which is non-orthogonal and overcomplete, so the number of pixels in the pyramid is much greater than the number of pixels in the input image. Rather than synthesizing the texture from white noise, we are moving in the opposite direction, that is, we "gaussianize" the input image by a nonsingular transform.

5.5.2 Histogram matching

Histogram matching transforms the marginal density function to a particular density function. If the desired density function is a Gaussian density function as $N(0,1)$, the histogram matching maps a random variable x monotonically onto a new variable y by

$$y{=}\Phi^{-1}(2\text{cdf}(x) - 1), \tag{5.38}$$

where $\text{cdf}(x)$ is the cumulative distribution function of x and Φ^{-1} is the inverse cumulative distribution function of Gaussian random variables. Both of these functions are monotonic, so $\Phi^{-1}(2\text{cdf}(x) - 1)$ in (5.38) is also monotonic. In other words, the operation of histogram matching is invertible.

We still need to choose the variance of the new variable y. We do this by letting the differential entropy H of the new variable be the same as that of x. This constraint let the iterative nonsingular transform process stop automatically when all of the marginal densities are Gaussian functions, since if the marginal density is already a Gaussian density function, the variance needs to be invariant in order to let H be invariant, then $y = x$. The reason for letting the differential entropy rather than the variance be invariant is that the variance is a little "misleading" when the marginal distribution is not a Gaussian distribution. For example, if the marginal density has a very long tail due to a small number of outliers, then the variance might be large but the differential entropy is small.

There are two methods for estimating the differential entropy from samples of data. The simpler method is based on approximating the density function using the polynomial expansions of Gram-Charlier or Edgeworth, which approximate the differential entropy by skewness and kurtosis as shown in (2.99). Another method by Hyvarinen [Hyvarinen 1997] is based on an approximate maximum-entropy procedure, which is given by

$$H(x) = \frac{1}{2}\left[1 + \log\left(2\pi\sigma_x^2\right)\right] - \left[k_1\left(E\{\widehat{x}\exp(-\widehat{x}^2/2)\}\right)^2 + k_2\left(E\{|\widehat{x}|\} - \sqrt{2/\pi}\right)^2\right], \tag{5.39}$$

where $k_1 = 36/\left(8\sqrt{3}-9\right), k_2 = 1/\left(2-6/\pi\right)$, σ_x^2 is the estimated variance of x, and $\widehat{x}$ is the standardized x with zero mean and unit variance. The expectation values inside the bracket in (5.39) are estimated by averaging over samples of $\widehat{x}$. This approximation of entropy is more accurate than the approximation derived using the Gram-Charlier expansion. We will use Hyvarinen's method in our simulation example. Finally, we let the entropy of y equal to $H(x)$, and get the variance of y by

$$H(y) = \frac{1}{2}\left[1 + \log\left(2\pi\sigma_y^2\right)\right]. \tag{5.40}$$

5.5.3 Example

We have simulated 200 training images with a log-normal density. Each training image is generated by exponentiating the sum of a lumpy background of type 2 and an uncorrelated Gaussian noise process. The lumpy background has mean of 7 and a correlation length of 10 pixels with $W(0) = 600$. The variance of the uncorrelated Gaussian noise process is 0.01. For designing the invertible transform, the wavelet basis functions are selected as the Daubechies wavelets which have a minimal support for a given number of vanishing moments. We let the length of the Daubechies wavelet filters be 8. We know that the Bayesian ideal observer applies a pixel-wise logarithm function to gaussianize an input image. Fig. 5.4 shows the input image with a

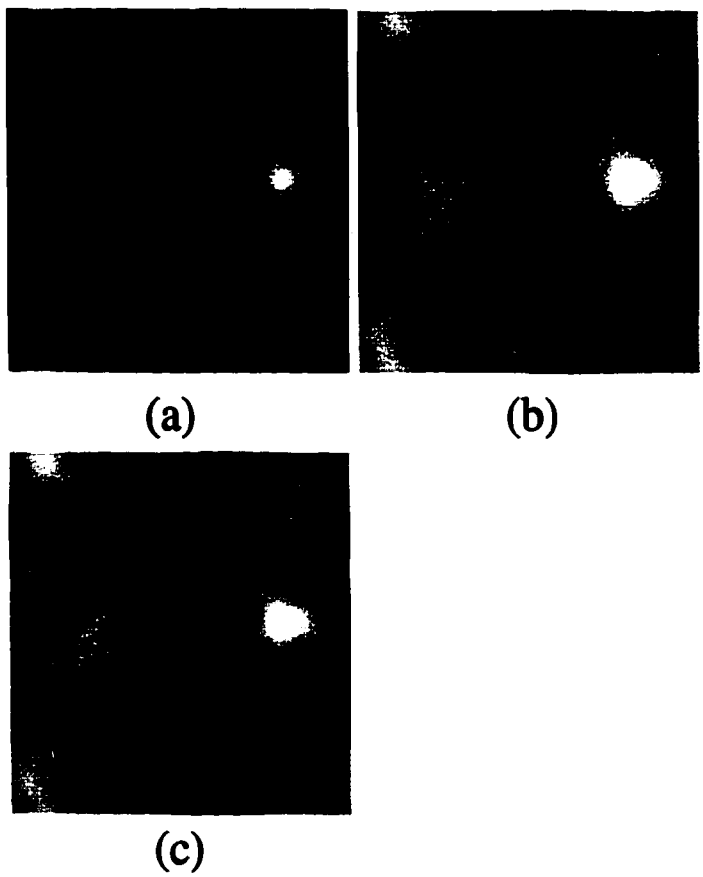
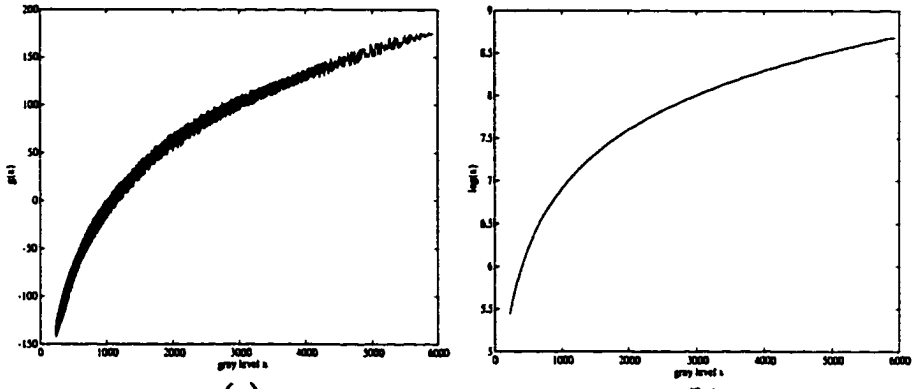


FIGURE 5.4. (a) The log-normal image; (b) The output image from a nonsingular transform; (c) The output image from a log function

lognormal density and the output images by our nonlinear transform method and the log-function. Fig. 5.5 shows the pixel-wise function by the above nonlinear transform method and the log-function. We see that the nonlinear transform is close to a log-function.

We also calculate the area under the ROC curve for an ideal Bayesian observer, our nonlinear discriminant analysis and linear discriminant analysis for the lognormal backgrounds with different correlation lengths. The signal is modeled as a Gaussian-shaped function with $r_s = 1$ pixels and $a = 45$ counts, and placed in the center of images. The ideal observer uses the nonsingular transform $\mathbf{T}(\mathbf{b}) = \log(\mathbf{b})$, then calcu-



(a) (b)

FIGURE 5.5. (a) is a pixel-wise function for the nonsingular transform $\mathbf{T}(\mathbf{b})$; (b) is a log function.

lates a test statistic by (5.37). On the other hand, $\mathbf{T}(\mathbf{b})$ in the nonlinear discriminant analysis makes use of wavelet transforms and histogram matching iteratively. Finally $\mathbf{T}(\mathbf{b}) = \mathbf{b}$ for a linear discriminant function.

Since the covariance matrix of the random background is a circulant matrix in this example (lumpy background of type 2), it can be diagonalized by the DFT. We calculate (5.37) in the Fourier domain using the DFT, that is

$$\lambda = \sum_{m=0}^{M-1} \sum_{n=0}^{M-1} \frac{|\mathrm{DFT}\,[\mathbf{T}(\mathbf{b})]|^2_{m,n} - |\mathrm{DFT}\,[\mathbf{T}(\mathbf{b}-\mathbf{s})]|^2_{m,n}}{\widehat{P}\,(m,n)}, \tag{5.41}$$

where $\widehat{P}\,(m,n)$ is the estimated power spectrum of $\mathbf{T}(\mathbf{b})$. The procedure used to estimate a power spectrum from an ensemble of image $\{\mathbf{b}_1, ..., \mathbf{b}_N\}$ is the two-dimensional extension of Welch's method [Welch 1967]. The procedure can be summarized in five steps [Rolland 1997]:

1. Given a set of images, compute the sample mean
2. Subtract from each image the sample mean to form a new set of images
3. Take the FFT of each nonsingular transformed image $\mathbf{T}(\mathbf{b}_n)$ to yield DFT[$\mathbf{T}(\mathbf{b}_n)$]
4. Compute the normalized periodogram as $|\mathrm{DFT}\,[\mathbf{T}(\mathbf{b}_n)]|^2 / (M \times M)$, where each image has a size of $M \times M$

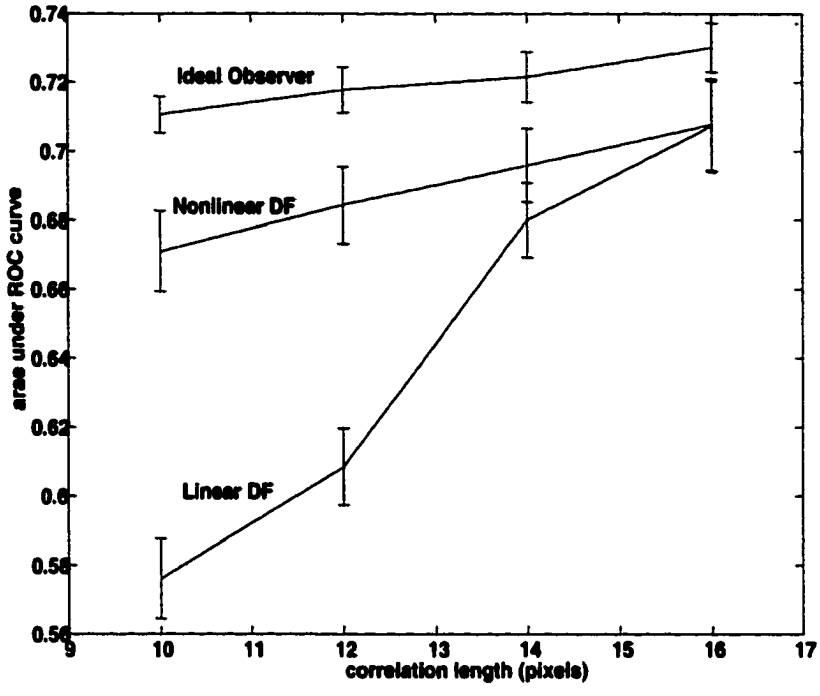


FIGURE 5.6. The area under ROC curve by three different methods.

5. Compute the average periodogram over the set of images

$$\widehat{P}(m,n) = \frac{1}{NM^2} \sum_{n=1}^{N} |\mathrm{DFT}\left[\mathbf{T}(\mathbf{b}_n)\right]|^2 . \tag{5.42}$$

We estimate the performance of the above observers using the area under ROC curve (AUC). We calculate the AUC by

$$\mathrm{AUC} = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \mathrm{step}\left(\lambda\left(\mathbf{b}_{1i}\right) - \lambda\left(\mathbf{b}_{0j}\right)\right), \tag{5.43}$$

where the data vectors $\mathbf{b}_{1i}$ are drawn from the signal-present set and the $\mathbf{b}_{0j}$ are drawn from the signal-absent set. The performance of our nonlinear discriminant analysis is close to the performance of the ideal observer, while the linear discriminant analysis is below the level of the other two methods.

5.6 Nonlinear discriminant analysis II

In this section we will discuss the second method for approximating the ratio

$$\Lambda = \frac{p(\mathbf{b}-\mathbf{s})}{p(\mathbf{b})} \tag{5.44}$$

by using a set of training backgrounds. We have discussed in chapter 3 that the statistical properties of training images can be explored by using the marginal distributions of a set of linear filters outputs. Given the marginal histograms of the filter outputs, we recall that the estimated statistical model is

$$\widehat{p}(b) = \frac{1}{Z}\prod_{\alpha=1}^{K} \exp\left\{-\sum_{\mathbf{r}} \phi^{(\alpha)}\left(F^{(\alpha)} * b(\mathbf{r})\right)\right\}, \tag{5.45}$$

where $\phi^{(\alpha)}$ is the potential function, which usually has a U shape. We have discussed how to estimate ϕ by using histogram matching methods. Eq. (5.45) is the maximum-entropy estimate of the probability density of the backgrounds from noise-free background samples. Hence, Λ is estimated by the ratio of two estimated probability density functions:

$$\begin{aligned}
\Lambda &= \frac{\widehat{p}(b-s)}{\widehat{p}(b)} \\
&= \frac{\prod_{\alpha=1}^{K} \exp\left\{-\sum_{\mathbf{r}} \phi^{(\alpha)}\left(F^{(\alpha)} * (b-s)(\mathbf{r})\right)\right\}}{\prod_{\alpha=1}^{K} \exp\left\{-\sum_{\mathbf{r}} \phi^{(\alpha)}\left(F^{(\alpha)} * b(\mathbf{r})\right)\right\}} \\
&= \prod_{\alpha=1}^{K}\prod_{\mathbf{r}} \frac{\exp\left\{\phi^{(\alpha)}\left(F^{(\alpha)} * b(\mathbf{r})\right)\right\}}{\exp\left\{\phi^{(\alpha)}\left(F^{(\alpha)} * (b-s)(\mathbf{r})\right)\right\}}
\end{aligned} \tag{5.46}$$

If the signal has a small size and known location at $\mathbf{r}_0$, then we can approximate the likelihood ratio by

$$\Lambda = \prod_{\alpha=1}^{K} \frac{\exp\left\{\phi^{(\alpha)}\left(F^{(\alpha)} * b(\mathbf{r}_0)\right)\right\}}{\exp\left\{\phi^{(\alpha)}\left(F^{(\alpha)} * (b-s)(\mathbf{r}_0)\right)\right\}}. \tag{5.47}$$

We call this method channelized-nonlinear discriminant analysis. The filter responses at location $\mathbf{r}_0$ are the channel outputs, and $\exp\left\{\phi^{(\alpha)}\left(F^{(\alpha)} * b(\mathbf{r}_0)\right)\right\}$ is proportional to the marginal density of the α^{th} channel output. If we define

$$v_\alpha^+ = F^{(\alpha)} * b(\mathbf{r}_0), \tag{5.48}$$

$$v_\alpha^- = F^{(\alpha)} * (b - s)(\mathbf{r}_0), \tag{5.49}$$

then $\mathbf{v}^+ = [v_1^+, v_2^+, ..., v_K^+]$ and $\mathbf{v}^- = [v_1^-, v_2^-, ..., v_K^-]$ are feature vectors, and Λ is the product of the ratio of marginal densities:

$$\Lambda = \prod_{\alpha=1}^{K} \frac{\exp\left\{\phi^{(\alpha)}(v_\alpha^+)\right\}}{\exp\left\{\phi^{(\alpha)}(v_\alpha^-)\right\}}. \tag{5.50}$$

Certainly the components of the feature vector are not independent of each other. In order to improve the performance of detection task, we can estimate an ICA matrix $\mathbf{W}$ using samples of feature vectors, and generate new feature vectors

$$\mathbf{u}^+ = \mathbf{W}\mathbf{v}^+, \tag{5.51}$$

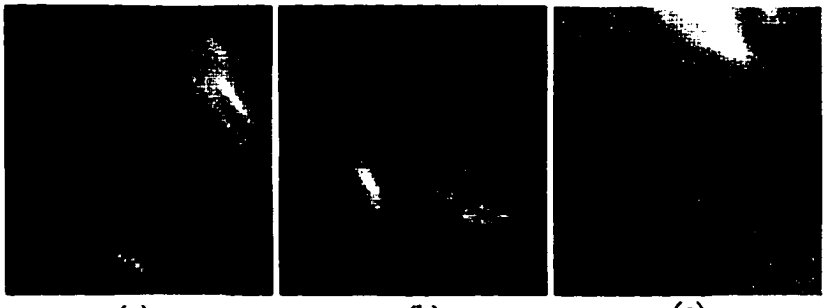
$$\mathbf{u}^- = \mathbf{W}\mathbf{v}^-, \tag{5.52}$$

where the components of the new feature vector are independent of each other. Then we can estimate a new potential function $\widehat{\phi}^{(\alpha)}$ so that $\exp\left\{\widehat{\phi}^{(\alpha)}(u_\alpha^+)\right\}$ is proportional to the marginal density of u_α^+. We rewrite the new Λ as

$$\Lambda = \prod_{\alpha=1}^{K} \frac{\exp\left\{\widehat{\phi}^{(\alpha)}(u_\alpha^+)\right\}}{\exp\left\{\widehat{\phi}^{(\alpha)}(u_\alpha^-)\right\}}. \tag{5.53}$$

In the following example, we choose the steerable pyramid as the set of filters. This is equivalent to applying the steerable pyramid transform on a background image and getting the channel outputs by the pyramid coefficients at location $\mathbf{r}_0$. We summarize the above steps as following:

1.Apply the steerable pyramid transform to a set of training images $\{b_n,\ n = 1, ..., N\}$



(a) (b) (c)

FIGURE 5.7. Clustered-blob lumpy backgrounds: Mean number of sub-blobs $N = 5, L_x = 5, L_y = 2$, $\alpha = 2.0, \beta = 0.5$. Image (a), Mean number of super-blobs $K = 10$. Image (b), $K = 50$. Image (c), $K = 100$.

2. Select the pyramid coefficients at signal location $\mathbf{r}_0$ as the components of feature vectors $\{\mathbf{v}_n^+,\ n = 1, ..., N\}$

3. Estimate an ICA matrix $\mathbf{W}$ using the set of feature vectors

4. Transform to a new set of feature vectors by $\mathbf{u}^+ = \mathbf{W}\mathbf{v}^+$

5. Estimate the marginal densities of $\{u_\alpha^+,\ \alpha = 1, ..., K\}$

6. Calculate the potential functions $\left\{\widehat{\phi}^{(\alpha)},\ \alpha = 1, ..., K\right\}$

We simulate a set of clustered-blob lumpy backgrounds by Bochud's algorithm [Bochud 1998]. The clustered-blob lumpy background contains a random number of super-blobs, distributed uniformly over the image. Each super-blob contains a random number of sub-blobs at a given orientation. The number of super-blobs and the number of sub-blobs is Poisson distributed. Fig. 5.7 illustrates three types of clustered-blob lumpy background with different mean number of super-blobs. For the set of image with signal-present, we insert a Gaussian-shaped signal in the center of the image.

We have compared this method with the channelized Hotelling observer and the full Hotelling observer. Since the clustered-blob lumpy backgrounds are ergodic and stationary, the covariance matrix is a Toeplitz matrix which is close to a circulant matrix when the size of image is large. So the covariance matrix can be approximately

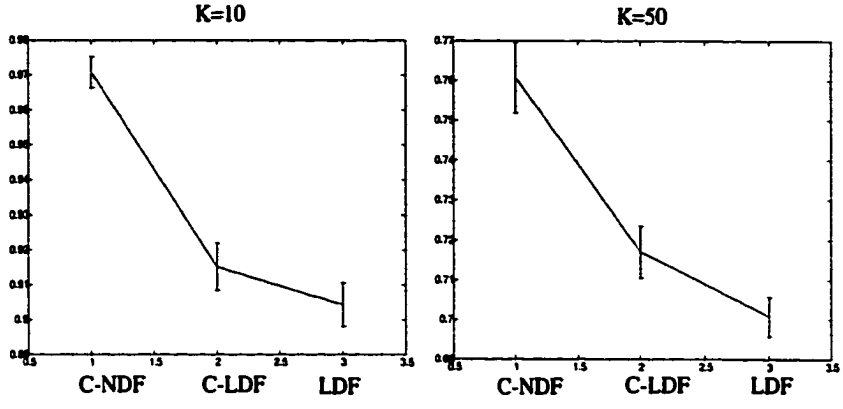


FIGURE 5.8. Observer performance (AUC) for two different sets of clustered-blob lumpy backgrounds. There are three observers: channelized nonlinear discriminant function (C-NDF), channelized-linear discriminant (C-LDF) and linear discriminant function by fourier transform (LDF)

diagonalized by the DFT, and we implement the full Hotelling discriminant function in the Fourier domain by the DFT. We estimate the performance of them using the area under ROC curve (AUC). The estimated AUC for the channelized nonlinear discriminant analysis is larger than for the other two methods.

Chapter 6

MARKOV CHAIN MONTE CARLO METHODS

6.1 Introduction

From the discussions in the last chapter, the ideal observer calculates a test statistic which is a posterior mean of Λ_{bke}

$$\Lambda = \int \Lambda_{\text{bke}}(b)\ p(b|g)db. \tag{6.1}$$

This test statistic can be approximated by Monte Carlo integration

$$\Lambda \approx \frac{1}{N}\sum_{n=1}^{N} \Lambda_{\text{bke}}(b_n). \tag{6.2}$$

If we can sample from the posterior distribution independently, then the law of large numbers ensures that the approximation can be made as accurate as desired by increasing the number of samples. Unfortunately, direct independent sampling from the posterior distribution is difficult in most problems.

In practice, either the distribution used to generate independent samples has to be different from (but similar to) the posterior distribution or the samples have correlations to each other [Tierney 1994]. The first method is called the *importance sampling* method, in which samples are weighted to make up for the difference between the posterior distribution and the sampling distribution. Many researchers have worked on the importance sampling for exploring posterior distributions [Stewart 1979], [Geweke 1989]. In high-dimensional space (the number of variables equals the number of pixels of a image), however, there are very few families of distributions which can be used as importance samplers [Evans 1995]. The second

method is called the *Markov chain Monte Carlo* (MCMC) method which samples from the posterior distribution by generating a Markov chain [Gelfand 1990]. We here consider using MCMC methods to approximate the ideal observer

6.2 Markov Chain

In MCMC the objective is to generate a sequence of samples that mimics a desired probability density function π. This objective can be achieved by constructing a homogeneous Markov chain with the invariant distribution as π.

Consider a stochastic process $\{X_n, n = 0, 1, 2, ...\}$ that takes on a finite or countable number of possible values. If $X_n = x$, then the process is said to be in state x at time n. We suppose that whenever the process is in state x, there is a fixed transition probability $P(x, y)$ that it will next be in state y. We assume that the chain is time-homogeneous; that is, $P(x, y)$ does not depend on n.

For the distribution of X_n to converge to a stationary distribution, the chain needs to satisfy three important properties [Roberts 1996]. First, it has to be *irreducible*. A Markov chain with invariant distribution π is irreducible if, for any initial state, it has positive probability of entering any set to which π assigns positive probability. Second, the chain needs to be *aperiodic*. A chain is periodic if there are potions of the state space it can only visit at certain regularly spaced times; otherwise, the chain is aperiodic. This stops the Markov chain from oscillating between different sets of states in a regular periodic movement. Finally, the chain must be *positive recurrent*. This can be expressed in terms of the existence of a stationary distribution π, such that if the initial value of X_0 is sampled from π, then all subsequent iterates will also be distributed according to π.

Theorem [Cinlar 1974] Suppose X is irreducible aperiodic. Then all states are

recurrent non-null if and only if the system of linear equations

$$\pi(y) = \sum_x \pi(x) P(x,y), \tag{6.3}$$

$$\sum_x \pi(x) = 1, \tag{6.4}$$

has a solution π. If there exists a solution π, then it is strictly positive, and we have

$$\lim_{n\to\infty} P^n(x,y) = \pi(y), \text{ for all } x, y. \tag{6.5}$$

In this case, π is a unique stationary distribution.

Most of the Markov chains produced in MCMC have stronger conditions than positive recurrent; that is, they are *reversible*. A Markov chain is said to be reversible if it is positive recurrent and satisfies the detailed balance equation:

$$\pi(x) P(x,y) = \pi(y) P(y,x). \tag{6.6}$$

The detailed balance essentially requires that in a very long time, the number of moves from x to y is identical to the number of moves from y to x. We can prove that if π satisfies (6.6), then (6.3) is automatically satisfied.

Proof: Substituting (6.6) into the right-hand side of (6.3), we have

$$\sum_x \pi(x) P(x,y) = \sum_x \pi(y) P(y,x) = \pi(y) \sum_x P(y,x). \tag{6.7}$$

Since the Markov chain must make a transition into some state,

$$\sum_x P(y,x) = 1. \tag{6.8}$$

Consequently, we see that π satisfies

$$\sum_x \pi(x) P(x,y) = \pi(y). \tag{6.9}$$

6.3 Metropolis-Hastings algorithm

There are many ways of constructing Markov chains in MCMC, but all of these methods are within the general framework of the Metropolis-Hastings algorithm. The Metropolis-Hastings algorithm was first developed by Metropolis et al. [Metropolis 1953] and generalized by Hastings [Hastings 1970]: if we are in state $X_n = x$ at time n, then the next state X_{n+1} is chosen by first sampling a candidate point y from a proposal function $q(y|x)$. The candidate point y is accepted with the acceptance probability

$$\alpha(x, y) = \min\left(1, \frac{\pi(y)q(x|y)}{\pi(x)q(y|x)}\right), \tag{6.10}$$

where π is the desired sampling density function. If the candidate point is accepted, then $X_{n+1} = y$. If the candidate point is rejected, then the chain does not move, i.e., $X_{n+1} = x$. Remarkably, the proposal function $q(.|.)$ can have any form and the Markov chain always satisfies the detailed-balance equation. This can be seen from the following argument.

Proof: If we assume

$$\pi(y)q(x|y) \geqslant \pi(x)q(y|x), \tag{6.11}$$

then from (6.10) we have

$$\begin{aligned} \alpha(x, y) &= 1, \\ \alpha(y, x) &= \frac{\pi(x)q(y|x)}{\pi(y)q(x|y)}. \end{aligned} \tag{6.12}$$

Hence,

$$\pi(x)q(y|x)\alpha(x, y) = \pi(y)q(x|y)\alpha(y, x). \tag{6.13}$$

Similarly, (6.13) holds when

$$\pi(y)q(x|y) < \pi(x)q(y|x). \tag{6.14}$$

Also, we know that the transition probability of the Markov chain is

$$P(x,y) = \begin{cases} q(y|x)\alpha(x,y) & \text{if } X_{n+1} = y \\ 1 - \sum_y q(y|x)\alpha(x,y) & \text{if } X_{n+1} = x \end{cases} . \tag{6.15}$$

If we substitute $P(x,y) = q(y|x)\alpha(x,y)$ into (6.13), then we obtain the detailed balance equation

$$\pi(x)P(x,y) = \pi(y)P(y,x) . \tag{6.16}$$

Next, we will list several special Metropolis-Hastings algorithms by using different proposal functions.

6.3.1 Metropolis algorithm

One of the simplest algorithms used in MCMC calculations is the Metropolis algorithm, which was originally introduced by Metropolis et al. for computing properties of substances composed of interacting individual molecules. In the Metropolis algorithm, trials are limited to steps taken away from the present position. Since the steps obey a symmetric distribution around zero, the Metropolis algorithm considers only symmetric proposal functions, having the form $q(y|x) = q(x|y)$. For the Metropolis algorithm, the acceptance probability is

$$\alpha(x,y) = \min\left(1, \frac{\pi(y)}{\pi(x)}\right) . \tag{6.17}$$

A special case of the Metropolis algorithm is the random-walk Metropolis, for which $q(y|x) = q(|y-x|)$, thus the chain is driven by a random walk process.

Most often, we use the Gaussian density function for the step function. When choosing the step function, its scale needs to be chosen carefully. A cautious transition probability function generating small steps will generally have a high acceptance rate, but has a long correlation time for samples and mixes slowly. A bold transition probability function generating large steps will often propose moves from the body to the tails of the distribution, giving a low probability of acceptance. Such a chain will frequently not move, again resulting in slow mixing.

6.3.2 Independent MCMC

If the candidate point for X_{n+1} is chosen from a fixed density function f, then the proposal function $q(y|x) = f(y)$, and the acceptance probability can be written as

$$\alpha(x, y) = \min\left(1, \frac{w(y)}{w(x)}\right), \tag{6.18}$$

where the weight function $w(x) = \pi(x)/f(x)$. It is useful to choose f to produce a weight function that is bounded, and as close to a constant as possible [Tierney 1994]. If the weight function is constant, then the Metropolis algorithm will never reject candidates, and the chain produces i.i.d. samples from π. Certainly, this goal may not be easily realized, otherwise, we can sample from the posterior distribution independently. On the other hand, it is safe to choose f thicker than the posterior density.

6.3.3 Single-component Metropolis-Hastings

In the above discussions we update the whole value of x whenever x is a scalar or a vector. If x is a vector then it is often more conveniently and computationally efficient to divide x into components $\{x_1, x_2, ..., x_M\}$, and then update these components one by one. We refer to it as single-component Metropolis-Hastings. Let $x_{-i} = \{x_1, ..., x_{i-1}, x_{i+1}, ...\}$, so x_{-i} comprises all of x except the ith component. The proposal function for updating x_i is $q(y_i|x_i, x_{-i})$ and the acceptance probability is

$$\alpha(x_{-i}, x_i, y_i) = \min\left(1, \frac{\pi(y_i, x_{-i})\, q(x_i|y_i, x_{-i})}{\pi(x_i, x_{-i})\, q(y_i|x_i, x_{-i})}\right). \tag{6.19}$$

Gibbs sampling is a special case of the single-component Metropolis-Hastings algorithm. For Gibbs sampling, the proposal function for updating x_i is the distribution of x_i conditioned on all the remaining components, called the full conditional distribution $\pi(x_i|x_{-i})$, which is defined by

$$\pi(x_i|x_{-i}) = \frac{\pi(x_i, x_{-i})}{\int \pi(x_i, x_{-i})\, dx_{-i}}. \tag{6.20}$$

By letting $q(y_i|x_i, x_{-i}) = \pi(y_i|x_{-i})$ and substituting this proposal function into (6.19), we see that

$$\alpha(x_{-i}, x_i, y_i) = 1. \tag{6.21}$$

In other words, we always accept the new candidate if we sample it from the full conditional distribution.

6.3.4 Auxiliary variable methods

A typical difficulty with single-component updating samplers is that they mix slowly when the components are highly correlated in the stationary distribution $\pi(x)$. Blocking highly correlated components may improve mixing. The idea of blocking highly correlated components is implied by auxiliary-variable algorithms, which partition the whole space into several clusters by introducing additional variables, often called bond variables, and update each cluster independently.

In auxiliary-variable algorithms, the variable x is augmented by an additional variable, u, with an arbitrary conditional distribution $p(u|x)$. The joint distribution $p(x, u) = \pi(x)p(u|x)$ is then the target of an MCMC simulation while the marginal distribution of the original variable remains $\pi(x)$. The joint distribution is constructed such that the MCMC scheme which samples over this joint distribution can lead to substantial gains in efficiency compared to standard approaches.

The first auxiliary variable method was proposed by Swendsen and Wang, designed to reduce critical slowing down in the Potts model. The Swendsen-Wang algorithm [Swendsen 1987] introduced a conditionally independent, binary auxiliary bond variable u_{ij} for each neighboring pair with

$$p(u_{ij} = 1|x) = 1 - \exp(-\beta I[x_i = x_j]), \tag{6.22}$$

where $u_{ij} = 1$ indicates the presence of a bond. Eq. (6.22) means that if $x_i = x_j$ then $p(u_{ij} = 1|x) = 1 - \exp(-\beta)$; if $x_i \neq x_j$ then $p(u_{ij} = 1|x) = 0$. Consideration of the

form of the conditional density $p(x|u)$ shows that this is a uniform distribution over pixel labels, subject to the constraints that clusters of pixels formed by pairs bonded together have the same label.

Auxiliary-variable methods have proved extremely effective in combatting the problems of critical slowing down in calculations with statistical physics models. In order to work well, however, the auxiliary variables need to be constructed so that either $p(x|u)$ can be simulated directly, or there is a rapidly mixing Metropolis-Hastings algorithm for this conditional distribution. This requirement may not be satisfied in all cases, and modifications of the Swendsen-Wang algorithm are necessary to specific applications. Nevertheless, the concept of auxiliary variables and data augmentation is very helpful in improving mixing. We will discuss a similar method called the reparameterization algorithm later as we apply it to image processing.

6.4 Statistical Efficiency of a MCMC sequence

The uncertainty in estimates of quantities derived from an MCMC sequence is a central issue. If samples v_n, $n = 1, ..., N$, are drawn from a scalar probability distribution by a stationary process, then the estimate of mean value, $\overline{v}$, is given by a sample mean:

$$\widehat{v} = \frac{1}{N}\sum_{n=1}^{N} v_n. \tag{6.23}$$

In this section we will discuss how to estimate the variance of $\widehat{v}$, and define the statistical efficiency of an MCMC sequence.

Since $\widehat{v}$ is an unbiased estimated of $\overline{v}$, $E\{\widehat{v}\} = \overline{v}$, the expected variance of $\widehat{v}$ is

$$\begin{aligned} var(\widehat{v}) &= E\{(\widehat{v}-\overline{v})^2\} \\ &= E\left\{\frac{1}{N^2}\sum_{j=1}^{N}(v_j-\overline{v})\sum_{k=1}^{N}(v_k-\overline{v})\right\} \\ &= \frac{1}{N^2}\sum_{j,k=1}^{N}E\{(v_j-\overline{v})(v_k-\overline{v})\}. \end{aligned} \tag{6.24}$$

We see from the above equation that the variance of $\widehat{v}$ is calculated by adding all the elements of the covariance matrix of v_n, $n = 1, ..., N$, then dividing by N^2. Since we assume that the Markov chain is a stationary process, the covariance matrix is a Toeplitz matrix and can be written as

$$\mathbf{K} = \sigma^2 \begin{bmatrix} \rho(0) & \rho(1) & \dots & \rho(N-1) \\ \rho(1) & \rho(0) & & \rho(N-2) \\ \dots & \dots & & \dots \\ \rho(N-1) & \rho(N-2) & \dots & \rho(0) \end{bmatrix}, \tag{6.25}$$

where σ^2 is the variance of v_n, $n = 1, ..., N$, $\rho(l)$ is the normalized autocovariance

$$\rho(l) = (\sigma^2)^{-1} E\{(v_k-\overline{v})(v_{k+l}-\overline{v})\}. \tag{6.26}$$

The summation of the elements of $\mathbf{K}$ is

$$\sum_{j,k=1}^{N}E\{(v_j-\overline{v})(v_k-\overline{v})\} = \sigma^2 N\left[\rho(0)+2\sum_{l=1}^{N-1}\rho(l)\right] - 2\sigma^2\sum_{l=1}^{N-1}l\rho(l). \tag{6.27}$$

Hence

$$var(\widehat{v}) = \frac{\sigma^2}{N}\left[\rho(0)+2\sum_{l=1}^{N-1}\rho(l)\right] - \frac{2\sigma^2}{N^2}\sum_{l=1}^{N-1}l\rho(l). \tag{6.28}$$

For stationary Markov chains, the value of v_n depends only on the value of the preceding element v_{n-1}, so $\rho(l)$ has an exponential function behavior, $\sum_{l=1}^{N-1} l\rho(l)$ converges to a constant when $N \to \infty$. Thus the second term in (6.28) goes to zero as $1/N^2$ and the first term goes to zero as $1/N$ when the number of samples increases.

We can approximate $var(\widehat{v})$ by neglecting the second term:

$$var(\widehat{v}) \approx \frac{\sigma^2}{N}\left[\rho(0) + 2\sum_{l=1}^{N-1}\rho(l)\right] = \frac{\sigma^2}{N}\sum_{l=-\infty}^{\infty}\rho(l). \tag{6.29}$$

If the samples are independent, then the normalized autocovariance is a delta function, $\rho(l) = \delta(l)$, and $\sum_{l=-\infty}^{\infty}\rho(l) = 1$, thus $var(\widehat{v}) = \frac{\sigma^2}{N}$. This result is in accordance with the central limit theorem.

We can estimate the variance σ^2 and the normalized autocorrelation from the Markov chain process. If the process has sufficiently converged to the pdf, the variance of the distribution is approximately the variance of the samples

$$\sigma^2 \approx S^2 = \frac{1}{N-1}\sum_{j=1}^{N}(v_k - \widehat{v})^2, \tag{6.30}$$

and the normalized autocovariance may be estimated from the sequence

$$\rho(l) \approx \frac{1}{S^2}\frac{1}{(N-l-1)}\sum_{j=1}^{N-1}(v_k - \widehat{v})(v_{k+l} - \widehat{v}). \tag{6.31}$$

The statistical efficiency of an MCMC sequence is defined as the reciprocal of the ratio of the number of MCMC trials need to achieve the same variance in an estimated quantity as are required for independent draws from the scalar density function [Hanson 1998]. Therefore,we see that the statistical efficiency is

$$\eta = \left[\sum_{l=-\infty}^{\infty}\rho(l)\right]^{-1}. \tag{6.32}$$

6.5 MCMC in image analysis

The most obvious feature of digital images is their large size. The size factor immediately favors the use of MCMC methods in a statistical approach to image analysis. Although other conventional numerical methods sometime provide practical routes to the calculation of point estimates of a true image, e.g. the maximum a posteriori

(MAP) estimate, MCMC is usually the only approach for assessing the variability of such estimates. A second feature is the spatial structure of images. The interplay between spatial variation of pixels is one of the most interesting aspects of image modelling from a mathematical perspective, and to construct prior models can be a considerable challenge. We have discussed several prior models in chapter 3. In contrast to the prior models of the spatial structure, the degradation model is simple and known in many cases. For example, the Poisson noise model is routinely assumed for medical imaging with x ray or gamma ray; therefore, if g denotes the recorded image and b is the background, then

$$p(g|b) = \prod_{n=1}^{M} p(g(\mathbf{r}_n)|b(\mathbf{r}_n)) = \prod_{n=1}^{M} \frac{b(\mathbf{r}_n)^{g(\mathbf{r}_n)} \exp(-b(\mathbf{r}_n))}{g(\mathbf{r}_n)!}. \tag{6.33}$$

Combining the prior density function and the conditional density function yields the posterior density function

$$\pi(b) = p(b|g) \propto p(g|b)p(b), \tag{6.34}$$

where $p(g|b)$ is assumed to be the Poisson conditional density in (6.33) and $p(b)$ is the prior density which is often assumed to be one of the Gibbs distributions discussed in chapter 3.

6.5.1 Single-component Metropolis-Hastings algorithm

The conditional density function $p(g|b)$ is also the likelihood function of b. If g has a Poisson density with mean b, then the likelihood function of b is the product of

gamma density functions:

$$\begin{aligned} L(b) &= \prod_{n=1}^{M} p(g(\mathbf{r}_n)\,|b(\mathbf{r}_n)) \\ &= \prod_{n=1}^{M} \frac{\lambda(\lambda b(\mathbf{r}_n))^{k_n-1}\exp(-\lambda b(\mathbf{r}_n))}{(k_n-1)!} \\ &= \prod_{n=1}^{M} \Gamma(b(\mathbf{r}_n)\,|\lambda, k_n), \end{aligned} \tag{6.35}$$

where $\lambda = 1$ and $k_n = g(\mathbf{r}_n) + 1$. Given $g(\mathbf{r}_n)$, $n = 1, ..., M$, sampling from the likelihood function requires only a one-dimensional gamma random variate generator. Thus it is computationally efficient for sampling from the likelihood function.

We suggest the proposal function as

$$q(\widehat{b}|b) = \prod_{n=1}^{h} p(g(\mathbf{r}_n)\,|\widehat{b}(\mathbf{r}_n)) \prod_{m=1}^{M-h} \delta_{\widehat{b}(\mathbf{r}_m)-b(\mathbf{r}_m),0}, \tag{6.36}$$

where h is the number of pixels being changed for the candidate image $\widehat{b}$, and δ is a Kronecker delta function. If $h = M$, then this method is the independent MCMC method. If $h < M$, then there are $M - h$ pixels unchanged between $\widehat{b}$ and b. Substituting (6.36) into (6.10), the acceptance probability is

$$\begin{aligned} & \alpha\left(b, \widehat{b}\right) \\ = & \min\left(1, \frac{\pi(\widehat{b})q(b|\widehat{b})}{\pi(b)q(\widehat{b}|b)}\right) \\ = & \min\left(1, \frac{p(g|\widehat{b})p(\widehat{b})\prod_{n=1}^{h} p(g(\mathbf{r}_n)\,|b(\mathbf{r}_n))\prod_{m=1}^{M-h}\delta(b(\mathbf{r}_m) - \widehat{b}(\mathbf{r}_m))}{p(g|b)p(b)\prod_{n=1}^{h} p(g(\mathbf{r}_n)\,|\widehat{b}(\mathbf{r}_n))\prod_{m=1}^{M-h}\delta(\widehat{b}(\mathbf{r}_m) - b(\mathbf{r}_m))}\right) \\ = & \min\left(1, \frac{p(\widehat{b})}{p(b)}\right). \end{aligned} \tag{6.37}$$

When h is large, the acceptance probability is very small and the chain moves slowly, so we let $h = 1$. Thus this method is a single-component Metropolis-Hastings method.

Algorithm 6.1: Single-component Metropolis-Hastings algorithm

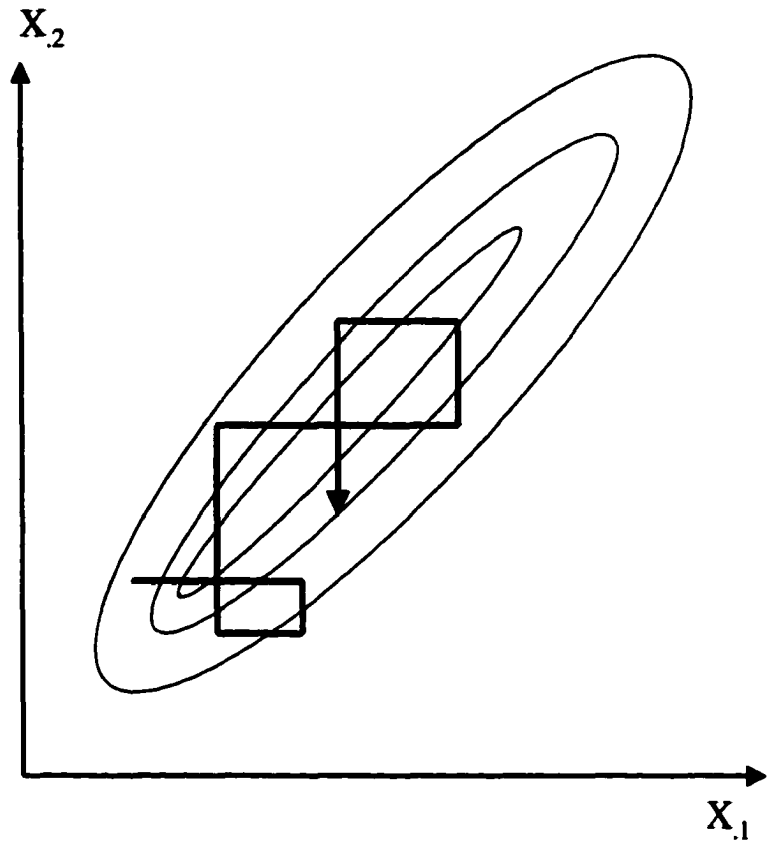


FIGURE 6.1. Illustrating a single-component Metropolis-Hastings algorithm for a bivariate target distribution $\pi(.)$. Components are updated alternately, producing alternate moves in horizontal and vertical directions.

1. Initialize b as the input image g
2. Repeat
   - 2.1 Set the candidate background $\widehat{b} = b$
   - 2.2 Uniformly select a location $\mathbf{r}_n$
   - 2.3 Update the pixel at $\mathbf{r}_n$ by $\widehat{b}(\mathbf{r}_n) \sim \Gamma\left(\widehat{b}(\mathbf{r}_n) | 1, g(\mathbf{r}_n) + 1\right)$
   - 2.4 Calculate the acceptance probability $\alpha\left(b, \widehat{b}\right) = \min\left(1, \frac{p(\widehat{b})}{p(b)}\right)$
   - 2.5 Accept $\widehat{b}$ with the probability $\alpha\left(b, \widehat{b}\right)$

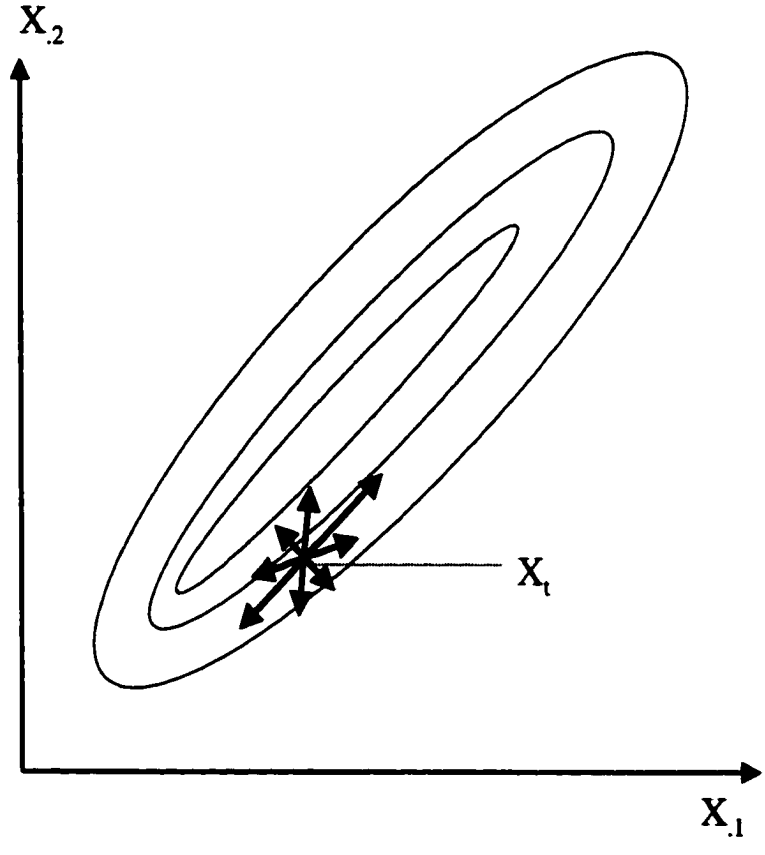


FIGURE 6.2. Illustrating the desired Metropolis-Hastings algorithm for a bivariate target distribution $\pi(.)$. Components are updated cooperatively, producing the efficient update directions..

6.5.2 Reparameterization algorithm

The single-component Metropolis-Hastings algorithm mixes slowly for two reasons: (1) only one pixel is updated in each iteration; (2) the update direction is not the same as the "diagonal line" of $\pi(b)$ as shown in Fig. 6.1. The second factor is the more important. In order to improve the sampling efficiency, we need to update pixels along the characteristic direction of $\pi(b)$ as shown in Fig. 6.2.

A solution to the slow mixing of the single component Metropolis algorithm is to transform an image b to a new representation such that the ordinary Metropolis-Hastings algorithm can mix easily in the transformed space. As we have discussed in chapter 2, the multiresolution representation of images is achieved by decomposing them into a set of spatial-frequency-bandpass images. Each bandpass image represents information about a particular fineness of detail or scale. We define a set of linear filters $\left\{F^{(\alpha)}, \alpha = 1, ..., K\right\}$ such that the filter responses are the complete representation of the input image. The filtered images are

$$I^{(\alpha)} = F^{(\alpha)} * b, \alpha = 1, ..., K, \tag{6.38}$$

and $I^{(\alpha)}(\mathbf{r})$ is the filter response at location $\mathbf{r}$. The notation $*$ denotes not only the linear convolution but also the multirate filtering. So we can use one of the three frame representations: (1) the Laplacian pyramid (2) the steerable pyramid and (3) the dyadic wavelet transform. The first two frames implement downsamplings after linear convolutions. The total size of $\left\{I^{(\alpha)}, \alpha = 1, ..., K\right\}$ is usually much larger than the size of b. For example, if we use a dyadic wavelet transform, then $I^{(\alpha)}$ is the filtered image without downsampling and has the same size as b, so the total size of data is increased by K times.

Recall that in chapter 3, we have discussed the maximum-entropy estimate of the

prior density function (6.39) from a set of training images.

$$\widehat{p}(b) = \frac{1}{Z}\prod_{\alpha=1}^{K} \exp\left\{-\sum_{\mathbf{r}} \phi^{(\alpha)}\left(F^{(\alpha)} * b(\mathbf{r})\right)\right\}. \tag{6.39}$$

The synthesized image using this prior model resembles the reference image. Note that $\widehat{p}(b)$ in (6.39) may not the same as the prior density function in the current integration problem, but we can use it to design a proposal function. As we know, Equ.(6.39) is a Markov random field model and the size of neighborhood is determined by the size of filters which may lie in a wide spatial range. Therefore, the single-component Metropolis sampling of the pixel at the location $\mathbf{r}_n$ in the filtered images actually updates the pixels in the neighborhood of $\mathbf{r}_n$ in the original image. Furthermore, the sampling complexity in the filtered images is the same as before because Eq. (6.39) implies that the filtered images are independent to each other.

We let the proposal function in the filtered image space be a symmetric Gaussian density function; that is, the pixel $I^{(\alpha)}(\mathbf{r}_n)$ is updated by adding a Gaussian random variable, $u^{(\alpha)}$, with zero mean and the variance being proportional to the variance of the filtered image $I^{(\alpha)}$,

$$I^{(\alpha)}(\mathbf{r}_n) + u^{(\alpha)} \rightarrow \widehat{I}^{(\alpha)}(\mathbf{r}_n), \ \alpha = 1, ..., K. \tag{6.40}$$

The rest of the pixels of the filtered images are unchanged. Then we recover the original image simply by transforming back the updated filtered images:

$$\left\{\widehat{I}^{(\alpha)}, \ \alpha = 1, ..., K\right\} \rightarrow \widehat{b}. \tag{6.41}$$

Since the Gaussian distribution is invariant under linear transforms, equivalently, we have a proposal function which is symmetric around zero (Gaussian distribution with zero mean) in the image space. So we actually use the random-walk Metropolis algorithm, and the acceptance probability reduces to

$$\alpha(b, \widehat{b}) = \min\left(1, \frac{\pi(\widehat{b})}{\pi(b)}\right), \tag{6.42}$$

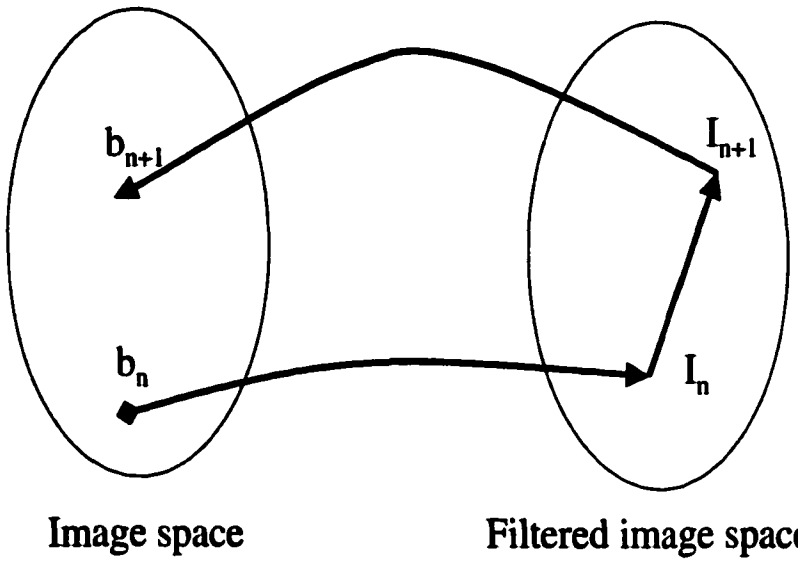


FIGURE 6.3. Illustrating the reparameterization algorithm

here $\pi(b)$ is the posterior density function. We illustrate this algorithm in Fig. 6.3.

Algorithm 6.2: Reparameterization algorithm

1. Given a reference image b^{obs}

2. Select a set of linear filters $\{F^{(\alpha)}, \alpha = 1, ..., K\}$

3. Compute the histograms of the filter responses

4. Initialize b as the input image g

5. Repeat

5.1 Generate the filtered images $\{I^{(\alpha)} = F^{(\alpha)} * b,\ \alpha = 1, ..., K\}$

5.2 Randomly pick a location $\mathbf{r}_n$

5.3 Sample a scalar random variable w_α from the Gaussian distribution with the same variance as that of $I^{(\alpha)}$

5.4 Update $\widehat{I}^{(\alpha)}(\mathbf{r}_n) = I^{(\alpha)}(\mathbf{r}_n) + \lambda w_\alpha$, where λ is a controlling parameter.

5.5 Reconstruct the synthesized image $\left\{\widehat{I}^{(\alpha)},\ \alpha = 1, ..., K\right\} \rightarrow \widehat{b}$

5.6 Accept the synthesize image $\widehat{b}$ with the probability as $\min\left(1, \frac{\pi(\widehat{b})}{\pi(b)}\right)$.

6.5.3 Simulation results

We let the random background b be the lumpy background of type 2, thus the prior distribution is a multivariate Gaussian distribution, and the autocorrelation function is

$$R(\mathbf{r}) = \frac{W(0)}{\pi r_b^2} \exp\left(-|\mathbf{r}|^2 / 2r_b^2\right), \tag{6.43}$$

where $W(0) = 10^7$counts and $r_b = 10$ pixels. The mean value of pixels is 1000 counts. The image g is generated by adding Poisson noise. We let the signal be a small Gaussian function with the radius of 2 pixels, and the peak value of the signal is 5 counts. We illustrate both the image data and the signal in Fig. 6.4.

Recall that the test statistic of the ideal observer can be approximated by

$$\Lambda \approx \frac{1}{N} \sum_{n=1}^{N} \Lambda_{\text{bke}}(b_n) = \frac{1}{N} \sum_{n=1}^{N} \exp\left(\lambda_{\text{bke}}(b_n)\right), \tag{6.44}$$

where $\lambda_{\text{bke}}(b)$ is the log-likelihood of two conditional Poisson densities:

$$\lambda_{\text{bke}}(b) = \log\left(\frac{p(g|b,s)}{p(g|b)}\right) = \sum_{m=1}^{M} g_m \log\left(1 + \frac{s_m}{b_m}\right) - s_m. \tag{6.45}$$

We illustrate samples of λ_{bke} from the Markov chains using both the single-component Metropolis-Hastings algorithm and the reparameterization algorithm. Since the consecutive iterations are highly correlated, we save the samples every 100 iterations. Fig. 6.5 illustrates the samples of λ_{bke} using the reparameterization algorithm, we see that the MCMC chain converges after about 4000 samples (or 4×10^5 iterations). Thus we discard an initial 4000 *burn-in* samples in estimating the value of Λ. The top graph of Fig. 6.6 illustrates the samples of λ_{bke} after the chain converges, which looks like a *white noise* process. We illustrate the autocovariance function of these samples in the bottom graph of Fig. 6.6. This autocovariance function resembles a delta function. The corresponding Λ_{bke} samples and its autocovariance function are in Fig. 6.7.

Λ	Ideal observer	Reparameterization	Single-component
1	0.709	0.709	0.843
2	0.699	0.773	0.535
3	1.398	1.594	2.031

TABLE 6.1. Estimated Λ by MCMC methods

We also illustrate the samples of λ_{bke} from the MCMC chain using the single-component Metropolis Hastings algorithm in Fig. 6.8. There is strong correlation between the samples, thus less statistical efficiency in the Monte Carlo integration.

Since the mean value of pixels is 1000 counts, i.e., the Poisson distribution is very close to the Gaussian distribution, we know that the image data has a Gaussian distribution and we can calculate the true value of the likelihood ratio. We list several Λ values estimated by both MCMC methods and the values calculated by the ideal observer in Table 6.1.

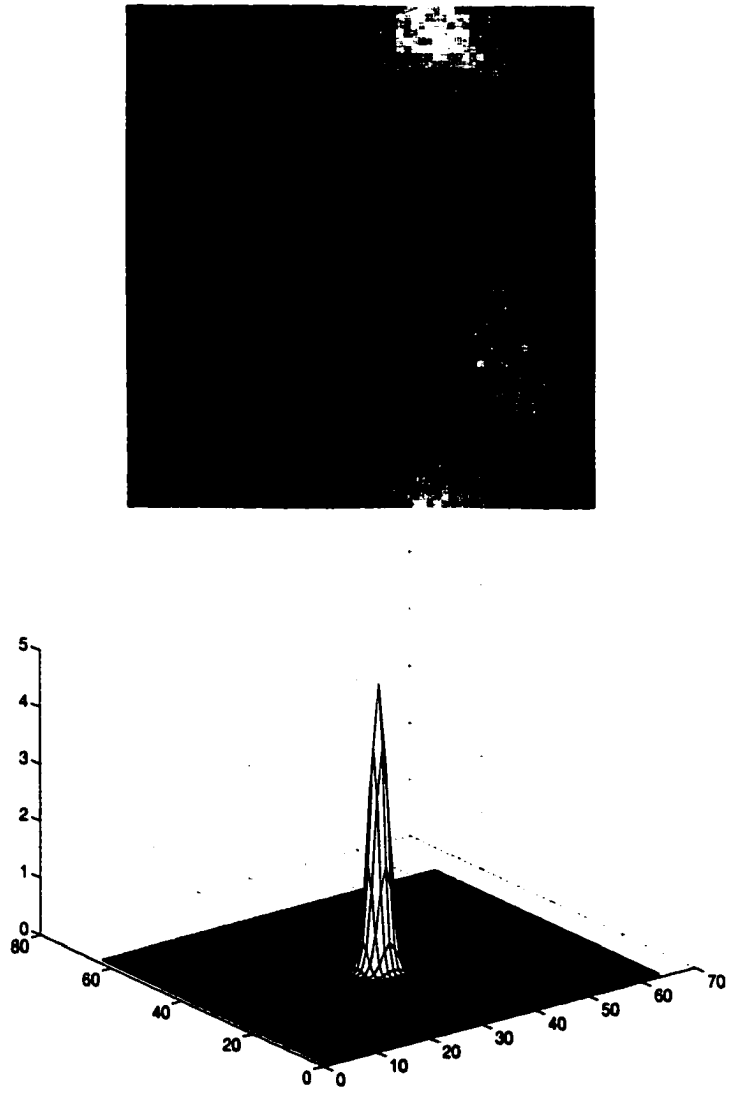


FIGURE 6.4. The input image g and the signal s.

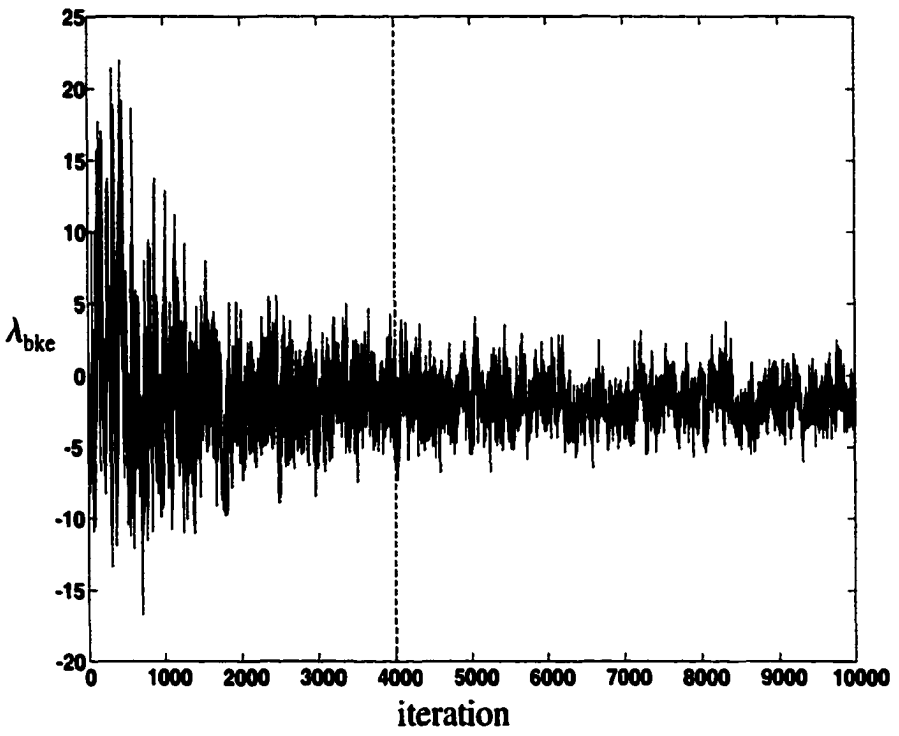


FIGURE 6.5. The samples of λ_{bke} from the MCMC chain using the reparameterization algorithm.

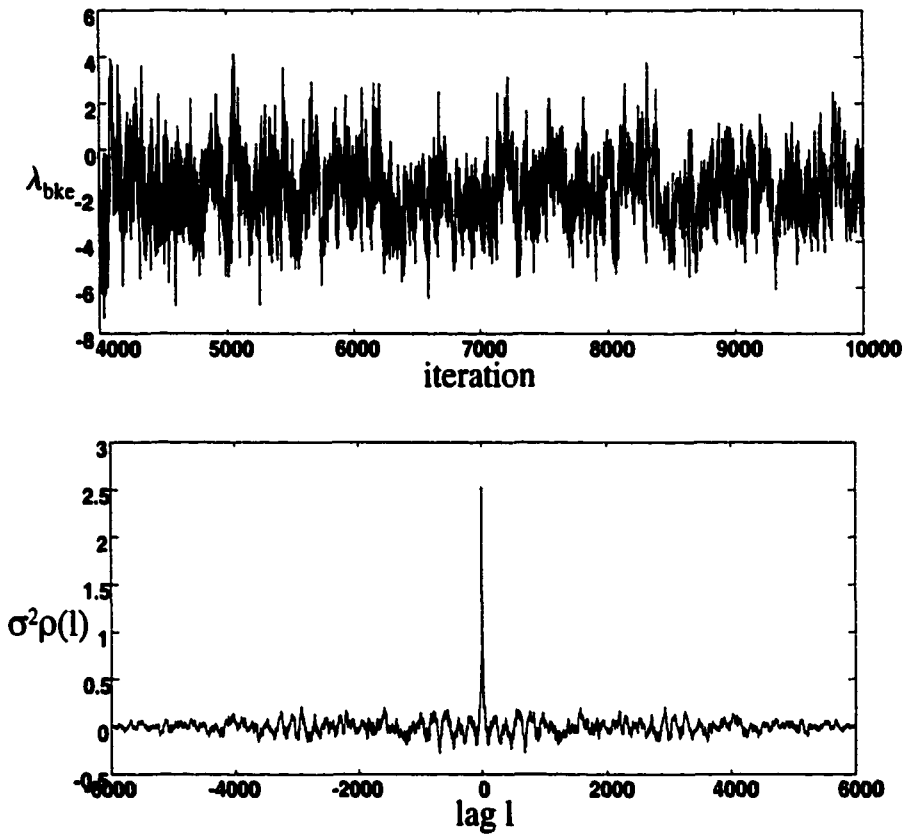


FIGURE 6.6. Top graph is the samples of λ_{bke} after an initial burn-in of 4000 iterations. Bottom graph is the autocovariance function of the samples.

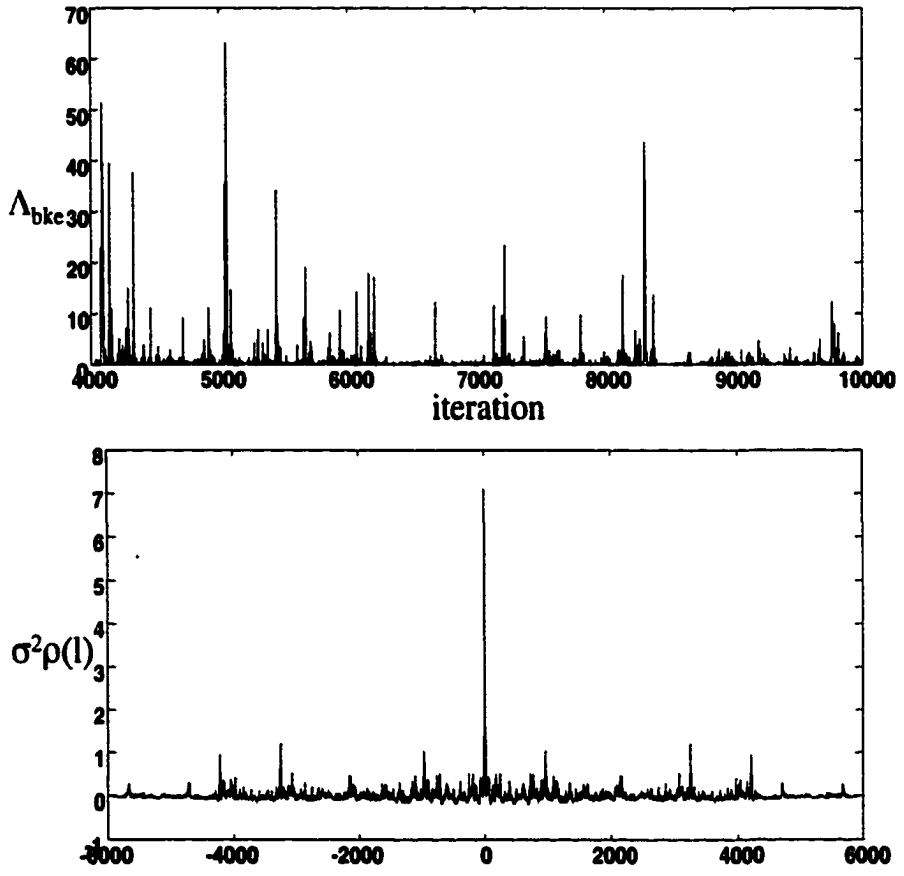


FIGURE 6.7. Top graph is the samples of Λ_{bke} after an initial burn-in of 4000 iterations. Bottom graph is the autocovariance function of the samples.

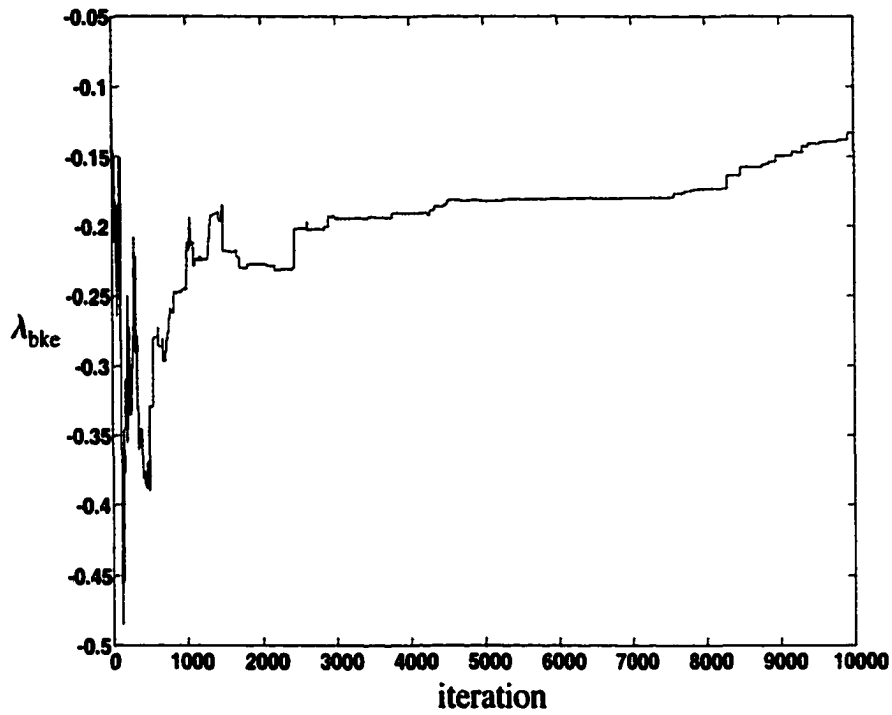


FIGURE 6.8. The samples of λ_{bke} from the MCMC chain using the single-component Metropolis Hastings algorithm.

Chapter 7

CONCLUSIONS

In this chapter we will summarize the main findings reported in this dissertation. We then highlight several questions that arose from this work and potential for further investigations.

7.1 Summary

The goal of this research is to develop computational methods for predicting how a given medical imaging system and reconstruction algorithm will perform when the resulting images are used by mathematical observers for tumor detection. These methods will then be used to compare imaging systems or optimize the parameters of a given system. They will also be used to compare reconstruction algorithms for a given imaging system and to find optimal values for the free parameters that such algorithms normally contain. The end result will be the ability to design medical imaging systems using software simulation of medical objects, imaging systems (including reconstruction algorithm) and observers. In order to realize this objective, we need to model and synthesize realistic medical images, model and optimize the imaging systems, construct the observer and estimate its performance.

Here the mathematical observer is the ideal observer which sets an upper limit to the performance as measured by the Bayesian risk or receiver operating characteristic analysis. Throughout the literature, however, the ideal observer is used only in very stylized tasks, for example, the detection of a disc signal superimposed on a uniform background or a random background with a Gaussian distribution and embedded in

white Gaussian noise. Thus it is necessary to construct or approximate the ideal observer with more complex images such as clinical images if we want to predict the medical image quality using the performance of the ideal observer.

This dissertation concentrates on constructing the ideal observer in complex detection problems and estimating its performance. We define our detection problem as a two-hypothesis detection task, where a known signal is superimposed on a random background with complicated distributions, and embedded in independent Poisson noise. The first challenge of this detection problem is that the distribution of the random background is usually unknown and difficult to estimate. Even though we have an ability to simulate realistic medical images, it is often the case that the full density function of these images is unknown. The second challenge is that the calculation of the ideal observer is computationally intensive for non-stylized problems and there are no analytical methods for these problems.

In order to solve these two problems, our work relies on multiresolution analysis of images. The multiresolution analysis is achieved by decomposing an image into a set of spatial-frequency-bandpass images, so each bandpass image represents information about a particular fineness of detail or scale. Connected with this method, we use three types of image representations by invertible linear transforms. They are the orthogonal wavelet transform, pyramid transform (or dyadic wavelet transform) and independent component analysis. All of these transforms share the same characteristics of the basis functions, namely, they are localized in spatial and frequency domains. These characteristics are in accordance with the findings from human and mammalian vision that the localized spatial and frequency representation of the natural images is capable of preserving both local and global information. This finding leads to a simple but very powerful texture modelling by using marginal densities of filter responses. In order to estimate the distribution of an ensemble of images given the empirical marginal distributions of filter outputs, we can use the maximum entropy principle and get a unique solution.

The ideal observer calculates a posterior mean of the ratio of conditional density functions, or the posterior mean of the ratio of two prior density functions, which are high-dimensional integrals. If the detection problem is not a stylized problem, we cannot calculate the integral analytically. But there are two ways to approximate the ideal observer. The first one is a classic decision process; that is, we construct a classifier following feature-extraction. The feature-extraction step is necessary for dimensionality reduction. We use the integrand of the posterior mean as features, which are calculated at an estimated background close to the posterior mode. The classifier combines these features to approximate the integral (or the ideal observer). Since we assume the conditional density of the image is an independent Poisson density function, we can calculate the first ratio easily. Regarding the second ratio, we can use the maximum-entropy estimate of the prior density functions to calculate its value.

Finally, if we know both the conditional density function and the prior density function, then we can also approximate the high-dimensional integral by Monte Carlo integration methods. If we can sample from the posterior distribution independently, then the law of large numbers ensures that the approximation can be made as accurate as desired by increasing the number of samples. Unfortunately, direct independent sampling from the posterior distribution is difficult in most problems. We consider using Markov chain Monte Carlo methods to approximate the ideal observer, which samples from the posterior distribution by generating a Markov chain. Since the calculation of the posterior mean is usually a very high-dimensional integration problem (the number of integrations equals the number of pixels), we must construct a Markov chain that can explore the posterior distribution efficiently. We give two proposal functions. The first proposal function is the likelihood function of random backgrounds. If the image data has an independent Poisson conditional distribution then the likelihood function is an independent gamma density function, so it is easy to generate the candidate sample by using a 1D gamma random variate generator.

The second method makes use of the multiresolution representation of the image by decomposing the image into a set of spatial frequency bands. Sampling one pixel in each band (in the same spatial location) equivalently updates a cluster of pixels in the neighborhood of the pixel location in the original image. Simulation results show that the second method mixes faster than the first method.

7.2 Future work

As we have discussed, the challenges of estimating the ideal observer are (1) learning statistics from images (2) computing a high-dimensional integral. We should investigate the potential for future work from these two aspects.

Although the findings from human and mammalian vision show that the localized spatial and frequency representation of natural images is capable of preserving both local and global information, the question phrased by Barlow & Tolhurst [Barlow 1992] "Why do we have edge detectors?" is still left open. That is: are there any coding principles that would predict the formation of localized, oriented filters (or receptive fields)? This question will not have a satisfying answer if we do not know the statistical structure of natural images. However, there is not even a clear definition of "natural image", and many authors just say that natural images are samples of pictures of trees, leaves, stones, and so on. For signal detection in medical imaging, we may ask whether medical images are natural images.

Recently people have investigated the statistical properties of high-pass filtered natural images: (boats, bark, toys, CTscan, Goldhill [Simoncelli 1999]), (trees, leaves [Bell 1997]), (mammogram [Heine 1999]). They have found that these high-passed natural images have highly non-Gaussian marginal histograms. In particular, the histograms are found to have much heavier tails and to be more sharply peaked at zero than the Gaussian density. Furthermore, these marginal densities are well-modeled

by a generalized Laplacian distribution. Another important finding is multiplicative scaling between wavelet coefficients; that is, the wavelet coefficients at a given scale are obtained from those at a coarser scale by multiplication with an independent random variable [Turiel 2000]. This implies a linear relation between the logarithms of the variables at two different scales [Simoncelli 1999]. If we can take these findings into account in an image model, then we may get a better estimate of the prior density function .

Regarding the computational issue, we have demonstrated an efficient MCMC scheme for sampling from the posterior distribution. However, the estimation of the performance of the ideal observer is still hampered by large computational costs. Currently, we implement this algorithm by MATLAB in one PC with 1Ghz Athlon CPU. We believe that this problem can be solved with an increase of our computational power and more computationally efficient coding.

Appendix A

CIRCULANTS AND DFT

Let $\mathbf{K}$ be a $N \times N$ circulant matrix, formed by cyclic shifts of the sequence $c(0)$, $c(1), \cdots, c(N-1)$:

$$\mathbf{K} = \begin{bmatrix} c(0) & c(N-1) & c(N-2) & \cdots & c(1) \\ c(1) & c(0) & c(N-1) & \cdots & c(2) \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ c(N-1) & \cdot & \cdot & \cdots & c(0) \end{bmatrix}. \tag{A.1}$$

The complete collection of eigenvectors of the circulant matrix is obtained from the unitary transform matrix [Andrews 1977]

$$\mathbf{F} = \left\{ \frac{1}{\sqrt{N}} \exp\left(\frac{-j2\pi mn}{N} \right) \right\}, \; 0 \le m, n \le N-1. \tag{A.2}$$

The eigenvalues of the circulant matrix are computed by the discrete Fourier transform (DFT) of the cyclic sequence $c(n)$ that makes up the circulant matrix:

$$\lambda(m) = \sum_{n=0}^{N-1} c(n) \exp\left(\frac{-j2\pi mn}{N} \right). \tag{A.3}$$

Thus any circulant matrix can be diagonalized by the DFT. That is,

$$\mathbf{FKF}^{\dagger} = \Lambda, \tag{A.4}$$

where $\Lambda = \text{Diag}\{\lambda(m), \; 0 \le m \le N-1\}$.

The DFT of the circular convolution of two sequences is equal to the product of their DFTs, that is, if

$$x_2(m) = \sum_{n=0}^{N-1} h(m-n)_c x_1(n), \; 0 \le m \le N-1, \tag{A.5}$$

then

$$\mathrm{DFT}\left\{x_2(m)\right\}_N = \mathrm{DFT}\left\{h(m)\right\}_N \ \mathrm{DFT}\left\{x_1(m)\right\}_N, \tag{A.6}$$

where $\mathrm{DFT}\{x(m)\}_N$ denotes the DFT of the sequence $x(m)$ of size N. Direct evaluation of (A.5) takes N^2 operations, while using the FFT for (A.6) will take $O(N \log_2 N)$ operations. This property is the most reason that people use FFT to implement the filtering of images. However, note that the $x_2(m)$ is the output of the circular convolution rather than the linear convolution. We can express Equ.(A.5) in terms of matrix-vector form:

$$\mathbf{x}_2 = \mathbf{H}_c \mathbf{x}_1, \tag{A.7}$$

where $\mathbf{H}_c$ is a circulant matrix.

Appendix B

Neural Network Classifiers and Bayesian Posterior Probability

In this appendix, we will prove that the outputs of neural-net classifiers are minimum-mean-squared error estimates of Bayesian posterior probability when the network has one output for each pattern class, desired outputs are 1 of M (one output unity corresponding to the correct class, all others zero), and a square-error cost function is used [Richard 1991].

Consider the problem of assigning an input feature vector $\mathbf{x}$ to one of M classes $\{H_i : i = 1, ..., M\}$. Let H_j denote the corresponding class of $\mathbf{x}$, $\{y_i(\mathbf{x}) : i = 1, ..., M\}$ the outputs of the network, and $\{d_i : i = 1, ..., M\}$ the desired outputs for all output nodes. For a 1 of M classification problem, $d_i = 1$ if $\mathbf{x}$ belongs to H_i and 0 otherwise, i.e., $d_i(H_j) = \delta_{ij}$. With a squared-error cost function, the network parameters are chosen to minimize the following:

$$\epsilon = \sum_{j=1}^{M} \left\{ E\left\{ \sum_{i=1}^{M} [y_i(\mathbf{x}) - d_i(H_j)]^2 \right\}_{\mathbf{x},H_j} \right\}, \tag{B.1}$$

where $E\{\cdot\}_{\mathbf{x},H_j}$ is the expectation operator of the joint probability density $p(\mathbf{x},H_j)$. Thus ϵ is a sum of mean-squared error for each input-class pair. By interchanging the summation and the integral, we have

$$\epsilon = \int \sum_{j=1}^{M} \sum_{i=1}^{M} [y_i(\mathbf{x}) - d_i(H_j)]^2 p(\mathbf{x},H_j)\ d\mathbf{x}. \tag{B.2}$$

Substituting $p(\mathbf{x},H_j) = p(H_j|\mathbf{x})\,p(\mathbf{x})$ in (B.2) yields

$$\epsilon = \int \sum_{j=1}^{M} \sum_{i=1}^{M} [y_i(\mathbf{x}) - d_i(H_j)]^2\, p(H_j|\mathbf{x})\, p(\mathbf{x})\; d\mathbf{x}. \tag{B.3}$$

Expanding the bracketed expression in (B.3), we have

$$\epsilon = \int \sum_{i=1}^{M} \sum_{j=1}^{M} \left[y_i^2(\mathbf{x})\, p(H_j|\mathbf{x}) - 2y_i(\mathbf{x})\, d_i(H_j)\, p(H_j|\mathbf{x}) + d_i^2(H_j)\, p(H_j|\mathbf{x})\right] p(\mathbf{x})\; d\mathbf{x}. \tag{B.4}$$

Since $y_i(\mathbf{x})$ is a function only of $\mathbf{x}$ and $\sum_{j=1}^{M} p(H_j|\mathbf{x}) = 1$, Eq. (B.4) can be expressed by

$$\epsilon = \int \sum_{i=1}^{M} \left[y_i^2(\mathbf{x}) - 2y_i(\mathbf{x}) \sum_{j=1}^{M} d_i(H_j)\, p(H_j|\mathbf{x}) + \sum_{j=1}^{M} d_i^2(H_j)\, p(H_j|\mathbf{x}) \right] p(\mathbf{x})\; d\mathbf{x}. \tag{B.5}$$

Note that $d_i(H_j) = \delta_{ij}$, so $\sum_{j=1}^{M} d_i(H_j)\, p(H_j|\mathbf{x}) = \sum_{j=1}^{M} d_i^2(H_j)\, p(H_j|\mathbf{x}) = p(H_i|\mathbf{x})$. We can simplify the above equation by

$$\epsilon = \int \sum_{i=1}^{M} \left[y_i^2(\mathbf{x}) - 2y_i(\mathbf{x})\, p(H_i|\mathbf{x}) + p(H_i|\mathbf{x})\right] p(\mathbf{x})\; d\mathbf{x}. \tag{B.6}$$

Adding and subtracting $\sum_{i=1}^{M} p^2(H_i|\mathbf{x})$ in (B.6) allows it to be cast in a form as

$$\epsilon = \int \sum_{i=1}^{M} [y_i(\mathbf{x}) - p(H_i|\mathbf{x})]^2\, p(\mathbf{x})\; d\mathbf{x} + \int \sum_{i=1}^{M} \left[p(H_i|\mathbf{x}) - p^2(H_i|\mathbf{x})\right]\; p(\mathbf{x})\; d\mathbf{x}. \tag{B.7}$$

The second integral in (B.7) is independent of the network outputs, the first integral is the sum of mean-squared error between the network outputs $y_i(\mathbf{x})$ and the Bayesian posterior probability $p(H_i|\mathbf{x})$. Thus when network parameters are chosen to minimize a squared-error cost function, outputs estimate the posterior probability.

REFERENCES

[Adelson 1984] Adelson, E.H. and Anderson, C.H. and Bergen, J.R. and Burt, P.J. and Ogden, J.M., "Pyramid Methods in Image Processing," RCA Engineer, 29-6, pp. 33-41.

[Amari 1996] Amari, S. and Cichocki, A. and Yang, H., "A New Learning Algorithm for Blind Signal Separation," Advances in Neural Information Processing Systems, Vol. 8, pp. 757-763.

[Andrews 1977] Andrews, H.C., Hunt, B.R., *Digital Image Restoration*, Prentive-Hall, Inc, Englewood Cliffs, NJ.

[Barlow 1992] Barlow, H.B. and Tolhurst, D.J., "Why do You Have Edge Detectors?" *Optical Society of America: Technical Digest*, Vol. 23, pp. 172.

[Barrett 1990] Barrett, H.H., "Objective Assessment of Image Quality: Effects of Quantum Noise and Object Variability," *Journal of the Optical Society of America A*, Vol. 7, pp. 1266-1278.

[Barrett 1992] Barrett, H.H. and Gooley, T. and Girodias, K. and Rolland, J. and White, T. and Yao, J., "Linear Discriminants and Image Quality", Image and Vision Computing, Vol. 10, No. 6, pp. 451-460.

[Barrett 1993] Barrett, H.H. and Yao, J. and Rolland, J. and Myers, K.J., "Model Observers for assessment of Image Quality", Proc. Natl. Acad. Sci. USa, Vol. 90, pp. 9758-9765.

[Barrett 1997] Barrett, H.H. and Abbey, C.K., "Bayesian Detection of Random Signals on Random Backgrounds," *Lecture Notes in Computer Science*, Vol. 1230, pp. 155-166, Springer-Verlag, Berlin.

[Barrett 1998a] Barrett, H.H. and Abbey, C.K. and Clarkson, E., "Objective Assessment of Image Quality. III. ROC Metrics, Ideal Observers, and Likelihood-Generating Functions", *Journal of the Optical Society of America A*, Vol. 15, pp. 1520-1535.

[Barrett 1998b] Barrett, H.H. and Abbey, C.K. and Gallas, B., "Stabilized Estimates of Hotelling-Observer Detection Performance in Patient-Structured Noise," *Proc. SPIE* 3340.

[Barrett 2001] Barrett, H.H. and Myers, K.J. and Gallas, B. and Clarkson, E. and Zhang, H., "Megalopinakophobia: Its Symptoms and Cures", *Proc. SPIE 4320.*

[Barrett Book] Barrett, H.H. and Myers, K.J., *Foundations of Image Science*, to be published.

[Battle 1987] Battle, G., "A Block Spin Construction of Ondelettes. Part I: Lemarie Functions," *Comm.. Math. Phys.*, Vol. 110, pp. 601-615.

[Bell 1995] Bell, A.J. and Sejnowski, T.J., "An Information-maximization approach to blind separation and blind deconvolution," Neural Computation, Vol. 6, pp.1129-1159.

[Bell 1997] Bell, A.J. and Sejnowski, T.J., "The Independent Components of Natural Scenes are Edge Filters," Vision Research, Vol. 37, No. 23, pp.3327-3338.

[Bergen 1991] Bergen, J.R., "Theories of Visual Texture Perception", *Spatial Vision*, pp. 114-133, CRC Press.

[Bochud 1998] Bochud, F.O. and Abbey, C.K. and Eckstein, M.P., "Statistical texture Synthesis of Mammographic Images with Clustered Lumpy Backgrounds," *Optics Express*, Vol. 4, No. 1, January.

[Burt 1983] Burt, P.J. and Adelson, E.H., "The Laplacian Pyramid as a Compact Image Code," *IEEE Transactions on Communications*, Vol. COM-31, No. 4.

[Burrus 1998] Burrus, C.S. and Gopinath, R.A. and Guo, H., *Introduction to Wavelets and Wavelet Transforms A Primer*, Prentice Hall.

[Cardoso 1997] Cardoso, J-F., "Infomax and Maximum Likelihood for Blind Source Separation," *IEEE Signal Processing Letters*, Vol.4, pp. 109-111.

[Chan 1988] Chan, H.P. and Doi, K. and Vyborny, C.J. and Lam, K.L. and Schmidt, R.A., "Computer-Aided Detection of Microcalcifications in Mammograms: Methodology and Preliminary Clinical Study," *Investigat. Radiol.*, Vol. 23, pp. 664-671.

[Chen 1983] Chen, P.C. and Pavlidis, T., "Segmentation by Texture Using Correlation," *IEEE Trans. Patt. Anal. Machine Intell.*, Vol. PAMI-5, pp. 64-69.

[Chubb 1991] Chubb, C. and Landy, M.S., "Orthogonal Distribution Analysis: A New Approach to the Study of Texture Perception," *Computational Models of Visual Processing*, pp. 291-301, MIT Press, Cambridge, MA.

[Cinlar 1974] Cinlar, E., *Introduction to Stochastic Processes*, Prentice-Hall, N.J..

[Clarkson 2000] Clarkson, E. and Barrett, H.H., "Approximations to Ideal-Observer Performance on Signal-Detection Tasks," *Applied Optics*, Vol. 39, No. 11, pp. 1783-1793.

[Cross 1983] Cross, G.R. and Jain, A.K., "Markov Random Field Texture Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-5, No. 1.

[Comon 1994] Comon, P., "Independent Component Analysis, A new Concept ?" Signal processing, Vol. 36, pp. 287-314.

[Daubechies 1992] Daubechies, I., "Ten Lectures on Wavelets", SIAM, Philadelphia, PA.

[Daubechies 1996] Daubechies, I., "Where do Wavelets Come From?- A Personal Point of View," *Proc. IEEE* Vol. 84, No. 4, pp. 510-513.

[Davies 1990] Davies, D.H. and Dance, D.R., "Automatic Computer Detection of Clustered Calcifications in Digital Mammograms," *Phys. Med. Biol.*, Vol. 35, pp. 1111-1118.

[Duffin 1952] Duffin, R.J. and Schaeffer, A.C., "A Class of Nonharmonic Fourier Series," Trans. Amer. Math. Soc., Vol. 72, pp. 341-366.

[Evans 1995] Evans, M. and Swartz, T., "Methods for Approximating Integrals in Statistics with Special Emphasis on Bayesian Integration Problems," *Statistical Science*, Vol. 10, No. 3, pp. 254-272.

[Fiete 1987] Fiete, R.D. and Barrett, H.H. and Smith. W.E. and Myers, K.J., "Hotelling Trace Criterion and Its Correlation with Human-Observer Performance," *Journal of the Optical Society of America A*, Vol. 4, pp. 945-953.

[Fisher 1936] Fisher, R.A., "The use of multiple measurements in taxonomic problems," *Ann. Eugenics* Vol. 7, pp. 179-188.

[Fukunaga 1990] Fukunaga, K., *Introduction to Statistical Pattern Recognition, Second ed.*, Academic Press, New York.

[Gelfand 1990] Gelfand, A.E. and Smith, A.F.M., "Sampling Based Approaches to Calculating Marginal Densities," *J. Amer. Statist. Assoc.* Vol. 85, pp.398-409.

[Geman 1984] Geman, S. and Geman, D., "Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 6, pp.721-741.

[Geweke 1989] Geweke, J., "Bayesian Inference in Econometric Models Using Monte Carlo Integration," *Econometrica*, Vo. 57, pp. 1317-1339.

[Gilks 1996] Gilks, W.R. and Richardson, S. and Spiegelhalter, D.J., Markov Chain Monte Carlo in Practice, Chapman & Hall/CRC.

[Girolami 1997a] Girolami, M., "An Alternative Perspective on Adaptive Independent Componenet Analysis Algorithms," Technical Report, Computing and Information Systems, Paisley Univsersity, Scotland, ISSN 1461-6122.

[Girolami 1997b] Girolami, M. and Fyfe, C., "Generalized Independent Component Analysis Through Unsupervised Learning With Emergent Bussgang Properties. In Proc. Internation Conference on Neural Networks, Houston, pp. 1788-1891.

[Green 1990] Green, P.J., "Bayesian Reconstructions From Emission Tomography Data Using a Modified EM Algorithm," *IEEE Transactions in Medical Imaging*, Vol.9, pp. 84-93.

[Hanson 1998] Hanson, K.M. and Cunningham, G.S., "Posterior Sampling with Improved Efficiency," *Proc. SPIE* 3338, 1998.

[Haralick 1973] Haralick, R.M. and Shanmugan, K. and Dinstein, I., "Texture Features for Image Classification," *IEEE Trans. Syst., Man, Cybern.*, Vol. SMC-8, No. 6, pp. 610-621.

[Harville 1997] Harville, D.A., Matrix Algebra From a Statistician's Perspective, Springer.

[Hastings 1970] Hastings, W.K., "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, Vol. 57, pp. 97-109.

[Haykin 1999] Haykin, S., *Nueral Network, A Comprehensive Foundation, Second ed.*, Prentice Hall, New Jersey.

[Heeger 1995] Heeger, D.J. and Bergen, J.R., "Pyramid-Based Texture Analysis/Synthesis", *Proc., International Conference on Image Processing* Vol 3, pp. 648-651.

[Heine 1999] Heine, J.J. and Deans, S.R. and Clarke, L.P., "Multiresolution Probability Analysis of Random Fields," *Journal of the Optical Society of America A*, Vol. 16, pp. 6-15.

[Hotelling 1931] Hotelling, H., "The generalization of Student's ratio," ANN. Math. Stat., Vol. 2, pp. 360.

[Hyvarinen 1997] Hyvarinen, A. "New Approximations of Differential Entropy for Independent Component Analysis and Projection Pursuit", *Technical Report A47*, Helsinki University of Technology, Laboratory of Computer and Information Science.

[Hyvarinen 1999] Hyvarinen, A. and Oja. E., "Independent Componenet Analysis: A Tutorial," http://www.cis.hut.fi/projects/ica/

[Karssemeijer 1991] Karssemeijer, N., "A Stochastic Model for Automated Detection of Calcifications in Digital Mammograms," in *Proc. 12th Int. Conf. Inform. Processing Med. Imag.*, Wye, UK, pp. 227-238.

[Kashyap 1982] Kashyap, R.L. and Chellappa, R. and Khotanzad, A., "Texture classification using features derived from random field models," *Patt. Recogn. Lett.*, Vol. 1, pp. 43-50.

[Kendall 1977] Kendall, M. and Stuart, A., *The Advanced Theory of Statistics*, Wiley, New York.

[Lee 1999] Lee, T. and Girolami, M. and Bell, A.J. and Sejnowski, T.J., "A Unifying Information-theoretic Framework for Independent Component Analysis," *International Journal of Computers and Mathematics with Applications.*

[Lemarie 1988] Lemarie, P.G., "Ondelettes a Localisation Exponentielle," *J. Math. Pures et Appl.*, Vol. 67, pp.227-236.

[Li 1995] Li, H.D. and Kallergi, M. and Clarke, L.P. and Jain, V.K. and Clark, R.A., "Markov Random Field for Tumor Detection in Digital Mammography," *IEEE Transactions on Medical Imaging*, Vol. 14, No. 3, September.

[Mallat 1989a] Mallat, S.G., "Multifrequency Channel Decompositions of Images and Wavelet Models," *IEEE Transactions on Acoustics. Sppech. and Signal Processing*, Vol. 37, No. 12, pp. 2091-2110.

[Mallat 1989b] Mallat, S.G., "A Theory of Multiresolution Signal Decomposition: The Wavelet Transform," IEEE Trans., PAMI-11, Vol.7, pp.674-693.

[Mallat 1989c] Mallat, S.G., "Multiresolution Approximation and Wavelet Orthonormal Bases of L2," Trans. Amer. math. Soc., pp. 315, pp.69-87.

[Mallat 1993] Mallat, S.G. and Zhang, Z., "Matching Pursuits with Time-Frequency Dictonaries," *IEEE Transactions on Signal Processing*, Vol. 41, No. 12, pp. 3397-3415.

[Mallat 1998] Mallat, S.G., *A Wavelet Tour of Signal Processing*, Academic Press, San Diego.

[Metropolis 1953] Metropolis, N. and Rosenbluth, A.W. and Teller, A.H. and Teller, E., "Equations of state calculations by fast computing machines," *J. Chem. Phys.* Vol. 21 pp. 1087-1091.

[Myers 1987] Myers, K.J. and Barrett, H.H., "Addition of a Channel mechanism to the Ideal-Observer Model," *Journal of the Optical Society of America A*, Vol. 4, pp. 2447-2457.

[Nandi 1999] Nandi, A.K., *Blind Estimation Using Higher-Order Statistics*, Kluwer Academic Publishers.

[Pearlmutter 1996] Pearlmutter, B. and Parra, L., "A Context-Sensitive Generalization of ICA," ICONIP'96, pp. 161-157.

[Richard 1991] Richard, M.D. and Lippmann, R.P., "Neural Network Classifiers Estimate Bayesian a Posteriori Probabilities," Neural Computation, Vol. 3, pp.461-483.

[Rioul 1993] Rioul, O., "A discrete-time multiresolution theory," *IEEE Trans. Signal Processing*, Vol. 41, No. 8, pp. 2591-2606.

[Roberts 1996] Robert, G.O., "Markov Chain Concepts Related to Sampling Algorithms," pp. 45-57, in *Markov Chain Monte Carlo in Practice*, eidted by Gilks,W.R. et al., Chapman & Hall/CRC.

[Rolland 1990] Rolland, J.P., University of Arizona, Ph.D. Dissertation, Factors Influencing Lesion Detection in Medical Imaging.

[Rolland 1992] Rolland, J.P. and Barrett, H.H., "Effects of Random Background Inhomogeneity on Observer Detection Performance," *Journal of the Optical Society of America A*, Vol. 9, pp. 649-658.

[Rolland 1997] Rolland, J.P. and Strickland, R.N., "An Approach to the Synthesis of Biological Tissue," *Optics Express*, 1, No.13, pp. 414-423.

[Ross 1996] Ross, S.M., *Stochastic Processes*, 2nd , John Wiley & Sons, Inc.

[Shensa 1992] Shensa, M.J., "The discrete wavelet transform: Wedding the a trous and Mallat algorithms," *IEEE Trans. Signal Processing*, Vol. 40, No. 10, pp. 2464-2482.

[Simoncelli 1992] Simoncelli, E.P. and Freeman, W.T. and Adelson, E.H. and Heeger, D.J., "Shiftable Multiscale Transforms," *IEEE Transactions on Information Theory*, Vol. 38, No. 2.

[Simoncelli 1995] Simoncelli, E.P. and Freeman, W.T., "The Steerable Pyramid: A Flexible Architecture for Multi-Scale Derivative Computation," *2nd Annual IEEE International Conference on Image Processing.*

[Simoncelli 1998] Simoncelli, E.P. and Portilla, J., "Texture Characterization via Joint Statistics of Wavelet Coefficient Magnitudes," *Proc. of 5th International Conference on Image Processing*, Vol. 1.

[Simoncelli 1999] Simoncelli, E.P., "Modeling the Joint Statistics of Images in the Wavelet Domain," *Proc. SPIE 44th Annual Meeting*, Vol. 3813.

[Strickland 1996] Strickland, R.N. and Hahn, H. II, "Wavelet Transforms for Detecting Microcalcifications in mammograms," *IEEE Transactions on Medical Imaging*, Vol. 15, No. 2, pp. 218-229.

[Stewart 1979] Stewart, L.T., "Multiparameter unvariate Bayesian inference," *J. Amer. Statist. Assoc.* Vol. 74, pp. 684-693.

[Smith 1986] Smith, W.E. and Barrett, H.H., "Hotelling Trace Criterion as a Figure of Merit for the Optimization of Imaging Systems," *Journal of the Optical Society of America A*, Vol. 3, pp. 717-725.

[Swendsen 1987] Swendsen, R.H. and Wang, J.S., "Nonuniversal Critical Dynamics in Monte Carlo Simulations," *Physical review Letters*, Vol. 58, pp. 86-88.

[Tierney 1994] Tierney, L., "Markov Chains for Exploring Posterior Distributions", Technical Report No. 560, School of Statistics, University of Minnesota.

[Turiel 2000] Turiel, A. and Parga, N., "Multifractal Wavelet Filter of Natural Images," *Physical Review Letters*, Vol. 85, No. 15., pp. 3325-3328.

[Unser 1986] Unser, M., "Local Linear Transforms for Texture Measurements," *Signal Processing*, Vol. 11, No. 1, pp. 61-79.

[Unser 1995] Unser, M., "Texture Classification and Segmentation Using Wavelet Frames," *IEEE Transactions on Image Processing*, Vol. 4, No. 11, pp. 1549-1560.

[Vaidyanathan 1990] Vaidyanathan, P.P., "Multirate Digital Filters, Filter Banks, Polyphase Networks, and Applications: A Tutorial," Proceedings of the IEEE, Vol. 78, No. 1.

[Van Trees 1968] Van Trees, H.L., *Detection, Estimation, and Modulation Theory*, Vol. 1, John Wiley, Noe York.

[Weir 1997] Weir, I.S., "Fully Bayesian Reconstructions From Single-Photon Emission Computed Tomography Data," *Journal of the American Statistical Association*, Vol. 92, No. 437.

[Welch 1967] Welch, P.D., "The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodograms," *IEEE Trans. Audio Electroacoust.* Vol. 15, pp. 70-73.

[Zhang 2001a] Zhang, H. and Clarkson, E. and Barrett, H.H., "Feature-extraction method based on the ideal observer," *Proc. SPIE 4320.*

[Zhang 2001b] Zhang, H. and Clarkson, E. and Barrett, H.H., "Nonlinear discriminant analysis," *Proc. SPIE 4320.*

[Zhu 1997] Zhu, S.C. and Mumford, D., "Prior Learning and Gibbs Reaction-Diffusion," IEEE Transaction On Pattern Analysis and Machine Intelligence, Vol. 19, No. 11.

[Zhu 1998] Zhu, S.C. and Wu, Y. and Mumford, D., "Filters, Random Fields and Maximum Entropy (FRAME)," *Int'l Journal of Computer Vision* 27(2) pp.1-20.