MULTIRESOLUTION IMAGE PROCESSING TECHNIQUES WITH APPLICATIONS IN TEXTURE SEGMENTATION AND NONLINEAR FILTERING

A Thesis
Submitted to the Faculty

of

Purdue University

by

Mary L. Comer

In Partial Fulfillment of the
Requirements for the Degree

of

Doctor of Philosophy

December 1995

To my brother Robert, who has taught me the most important lessons in life.

ACKNOWLEDGMENTS

I wish to express my sincere thanks to my Major Professor Edward J. Delp for his guidance and support. I greatly appreciate his care and dedication in constructively criticizing my work, including my thesis. I have truly enjoyed and benefited from working with him.

I would also like to thank the other members of my committee, Prof. C. Bouman, Prof. A. Maciejewski and Prof. B. Lucier, for their support.

I wish to thank the National Science Foundation for supporting this research.

Finally, I would like to thank my family for their constant support and encouragement, and my husband for being so understanding during my many hours away from home.

DISCARD THIS PAGE

TABLE OF CONTENTS

Page

DISCARD THIS PAGE

LIST OF TABLES

DISCARD THIS PAGE

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

ABSTRACT

Comer, Mary L. Ph.D., Purdue University, December 1995. Multiresolution Image Processing Techniques with Applications in Texture Segmentation and Nonlinear Filtering. Major Professor: Edward J. Delp.

We present a new algorithm for segmentation of textured images using a multiresolution Bayesian approach. The algorithm uses a multiresolution Gaussian autoregressive (MGAR) model for the pyramid representation of the observed image, and assumes a multiscale Markov random field model for the class label pyramid. Unlike other approaches, which have either used a single-resolution representation of the observed image or implicitly assumed independence between different levels of a multiresolution representation of the observed image, the models used in this thesis incorporate correlations between different levels of both the observed image pyramid and the class label pyramid. The criterion used for segmentation is the minimization of the expected value of the number of misclassified nodes in the multiresolution lattice. The estimate which satisfies this criterion is referred to as the "multiresolution maximization of the posterior marginals" (MMPM) estimate, and is a natural extension of the single-resolution maximization of the posterior marginals (MPM) estimate. The parameters of the MGAR model — the means, prediction coefficients, and prediction error variances of the different textures — are unknown. The expectation-maximization (EM) algorithm is used to estimate these parameters while simultaneously performing the segmentation. Analysis and experimental results demonstrating the performance of the algorithm are presented.

We also propose new approaches for the extension of binary and grayscale morphological operations to color imagery. We investigate two approaches for "color

morphology"—a vector approach and a component-wise approach. New vector morphological filtering operations are defined, and a set-theoretic analysis of these vector operations is presented. We also present experimental results comparing the performance of the vector approach and the component-wise approach for multiscale color image analysis and for noise suppression in color images.

Finally, we describe a video coding technique which generates an embedded bit stream that provides complete data rate scalability. This video coding scheme is based on the Embedded Zerotree Wavelet (EZW) algorithm for still image compression. We present experimental results demonstrating the performance of the algorithm at various data rates.

1. INTRODUCTION

This thesis addresses the problem of segmenting a textured image using a statistical approach. In the observed image, there are a number of regions, corresponding to different objects, or different textures. Each pixel in the image must be assigned to one of a finite number of classes depending on statistical properties of the pixel and its neighbors. The individual pixel classifications, or labels, form a two-dimensional field, with the same dimensions as the observed image, in which the value at a given spatial location reflects the class to which the corresponding pixel in the observed image belongs. This two-dimensional field containing the individual pixel classifications will be referred to as the label field. The label field is unknown and must be estimated from the observed image. Both the observed image and the label field will be modeled as discrete-parameter random fields.

If the image to be segmented has the property that pixels which are spatially close are likely to belong to the same class, then it is important for the segmentation algorithm to consider the classification of neighboring pixels when classifying a particular pixel. One way to do this is to divide the image into blocks and classify all pixels in a given block to the same class. However, this approach forces boundaries in the segmented image to occur along block boundaries, although the block boundaries do not necessarily correspond to region boundaries in the observed image. The use of statistical models for image segmentation has been proposed to address this problem [1, 2, 3, 4, 5]. In these approaches the label field and the observed image are modeled as discrete-parameter random fields and the segmentation problem becomes a statistical estimation problem. A Markov random field (MRF) model is often used for the label field. This model imposes the spatial constraint that neighboring pixels are

likely to belong to the same class, but it also allows neighboring pixels to be classified to different regions if the statistical properties of the observed image suggest that the pixels lie on an object boundary.

Statistical segmentation schemes generally segment an image by optimizing some criterion. Several algorithms which approximate the maximum *a posteriori* (MAP) estimate of the label field given the observed image have been proposed [1, 6, 7]. Another criterion which has been used is the minimization of the expected value of the number of misclassified pixels. The estimate which optimizes this criterion is known as the "maximizer of the posterior marginals" (MPM) estimate. It has been shown that the MPM estimation criterion is more appropriate for image segmentation than the MAP criterion [3]. This is because the MAP estimate assigns the same cost to every incorrect segmentation, regardless of the number of pixels at which the incorrect segmentation differs from the true segmentation, whereas the MPM estimate assigns a cost to an incorrect segmentation based on the number of incorrectly classified pixels in that segmentation.

As is the case with the MAP estimate, it is computationally infeasible to compute the MPM estimate exactly. A stochastic algorithm for approximating the MPM estimate of the label field was proposed in [3]. However, this algorithm assumes that the values of all parameters for the observed image and label field models are known *a priori*. If some of these model parameters are unknown, the algorithm in [3] cannot be used.

We have proposed the EM/MPM algorithm[1], a stochastic algorithm which combines the expectation-maximization (EM) algorithm for parameter estimation with the MPM algorithm for segmentation, to address this problem [8]. The same algorithm was also proposed in [9], along with a deterministic scheme that also approximates the MPM estimate of the label field and the EM estimates of model parameters. The EM/MPM algorithm, and its deterministic counterpart from [9], estimate

[1]In this thesis the single-resolution EM/MPM algorithm will be referred to simply as the "EM/MPM algorithm". The multiresolution extension of the EM/MPM algorithm will be referred to as the "multiresolution EM/MPM algorithm"

parameters at each stage of the algorithm using the current estimates of the marginal conditional probabilities of the class labels. This is referred to as a "soft-decision" scheme in [9], in contrast to "hard-decision" schemes which use the current segmentation to estimate parameters at each stage of the algorithm [6, 7, 10]. The soft-decision approach was shown in [9] to provide better results than the hard-decision approach.

As with other single-resolution approaches to image segmentation, it is difficult to process large-scale information effectively using the EM/MPM algorithm. Relatively small neighborhood sizes must be used to make the algorithm computationally feasible, but small neighborhoods are only able to describe small-scale behavior. The importance of utilizing information at various scales has led to the development of several multiresolution approaches to textured image segmentation [11, 12, 13, 14, 15, 16, 17]. Exploiting information at multiple resolutions offers several advantages over single-resolution approaches. First, algorithms which determine the classification of a given pixel based on local characteristics in the region containing the pixel can effectively make decisions based on larger neighborhoods by examining the image at coarser resolutions, without the increase in computational complexity which would result from simply using larger neighborhoods at the original image resolution. Second, processing an image at multiple resolutions precludes the need for *a priori* selection of one optimal resolution for processing. Also, computational complexity is reduced, since much of the work can be done at coarse resolutions, where there are significantly fewer pixels to process.

In this thesis we propose a multiresolution extension of the EM/MPM algorithm, using a multiresolution Gaussian auto-regressive (MGAR) model for the observed image pyramid (the pyramid representation of the observed image) and a multiresolution Markov random field (MMRF) model for the label pyramid (the pyramid containing the classifications of all nodes in the lattice upon which the observed image pyramid is defined). Our approach is different from previously proposed approaches in that we obtain a multiresolution representation of the observed image and model this representation as a stochastic process indexed by the nodes of a multiresolution lattice.

Previous approaches have used multiresolution models for the pixel labels, but have used either single-resolution representations of the observed image [11, 12, 14] or multiresolution representations of the observed image with the implicit assumption that the random variables at a given level of the observed image pyramid are independent from the random variables at other levels [13, 15, 16]. In [12] the MAP estimate of the label field at the coarsest resolution is approximated first, using iterated conditional modes (ICM), and the result is then propagated to the next-finer resolution, where the MAP estimate of the label field at that resolution is approximated using ICM. This process is continued until the finest resolution is reached. In [13] the image is first segmented at the coarsest resolution using a Bayes decision rule, and then a coarse-to-fine procedure is used to refine the segmented image. Varying-resolution simultaneous autoregressive (SAR) models are fit to the observed image in [15], and features from different resolutions are combined to segment the image using a K-means clustering algorithm. In [16] the observed image is modeled at each resolution as a Gauss Markov random field (GMRF). It is assumed that the GMRF parameters at the finest resolution are known, and the GMRF parameters at coarser resolutions are estimated using the values of these finest-resolution parameters. The MAP estimate is approximated at each resolution using ICM first at the coarsest levels, then at finer levels, as in [12]. The problem of texture discrimination, in which, given a set of known textures and a noisy observed image, the texture from the pre-defined set corresponding most closely to the observed data must be selected, is addressed using a multiscale stochastic model in [18]. The texture discrimination problem is different from the texture segmentation problem considered in this thesis, in which there is no pre-defined set of textures from which to select.

Our approach is also different from previously proposed multiresolution approaches in that we approximate the multiresolution maximizer of the posterior marginals (MMPM) estimate of the label pyramid, which is a natural extension of the single-resolution MPM estimate. Previous approaches have been based on MAP estimation. Using an MPM-based approach has two possible advantages. First, the cost function

which the MPM estimate minimizes is more appropriate for image segmentation than the cost function which the MAP estimate minimizes [3]. Second, using the MPM criterion facilitates the use of the EM algorithm to estimate unknown parameters of the MGAR model [8].

In Chapter 2 we describe the EM/MPM algorithm and present experimental results demonstrating its performance. In Chapter 3 the multiresolution EM/MPM algorithm is described, and experimental results demonstrating its performance are presented. Chapter 4 contains a theoretical analysis of the EM/MPM and multiresolution EM/MPM algorithms.

In Chapter 5 we address the problem of applying mathematical morphology to color images. Mathematical morphology has been shown to be useful for shape analysis in binary and grayscale images [19, 20, 21]. We investigate two approaches for "color morphology": a vector approach, in which color vectors are ranked using a multivariate ranking concept known as reduced ordering, and a component-wise approach, in which grayscale morphological operations are applied to each of the three color component images independently. New vector morphological filtering operations are defined, and a set-theoretic analysis of these vector operations is presented. We also present experimental results comparing the performance of the vector approach and the component-wise approach for two applications: multiscale color image analysis and noise suppression in color images.

In Chapter 6 we describe a video coding technique which generates an embedded bit stream that provides complete data rate scalability. Previously proposed techniques for rate-scalable video coding [22, 23] have provided partial rate-scalability, i.e., bit streams encoded using these techniques can be decoded at one of a finite set of data rates. The scheme described in Chapter 6 is completely rate-scalable in that the encoded bit stream can be decoded at any data rate. For each group of pictures (GOP, following MPEG terminology), the goal is to generate the bits representing that GOP in order of visual importance, so that any target data rate can be met, and the best decompressed sequence for that rate can be obtained by simply discarding

bits at the end of the bit stream for that GOP. Thus, one compressed version of the video sequence can provide a range of data rates from very low rates to very high rates, using the exact same encoder and decoder structures for any data rate. Also, the desired data rate can be dynamically changed in the decoder very easily from GOP to GOP. The algorithm described in Chapter 6 uses intraframe coding only, so that the GOP size is one frame. Experimental results are presented for various data rates.

2. THE EM/MPM ALGORITHM FOR SEGMENTATION OF TEXTURED IMAGES

This chapter describes the EM/MPM algorithm for simultaneous parameter estimation and segmentation of textured images. This algorithm uses a Markov random field (MRF) model for the pixel class labels and alternately approximates the MPM estimate of the pixel class labels and estimates parameters of the observed image model. In Section 2.1 the models used for the label field and the observed image are described. In Section 2.2 we describe the MPM segmentation algorithm for the case in which the values of all model parameters are known, and in Section 2.3 the EM algorithm is described for the case when the values of the marginal conditional probability mass functions of the label field are known. The EM/MPM algorithm is described in Section 2.4. Section 2.5 contains experimental results.

2.1 Image Models

In this chapter the label field will be denoted $\mathbf{X}$ and the observed image will be denoted $\mathbf{Y}$. The element in $\mathbf{X}$ at spatial location $(i, j) \in S$, where S is the rectangular pixel lattice on which $\mathbf{X}$ and $\mathbf{Y}$ are defined, is the random variable denoted by $X_{i,j}$. This notation is also used for $\mathbf{Y}$. To simplify the notation, the random variables in $\mathbf{X}$ and $\mathbf{Y}$ may also be indexed by a single letter, as in a lexicographical ordering, in which case the sth random variable in $\mathbf{X}$ will be denoted X_s and the sth random variable in $\mathbf{Y}$ will be Y_s. Throughout this chapter, $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_N)$, where N is the total number of pixels in S, will represent sample realizations of $\mathbf{X} = (X_1, X_2, \ldots, X_N)$ and $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_N)$. The space of all possible realizations

of $\mathbf{X}$ will be denoted $\Omega_{\mathbf{x}}$ and the space of possible realizations of $\mathbf{Y}$ will be denoted $\Omega_{\mathbf{y}}$.

2.1.1 Markov random field model

In this section we review the concept of a MRF and define the specific MRF model which we will use. Interested readers should see [2, 24] for a more detailed discussion of MRF models.

It is first necessary to define the concept of a neighborhood system for the pixel lattice S. The collection $\mathcal{G} = \{\mathcal{G}_s \subseteq S, \forall s \in S\}$ is a neighborhood system for S if, for every pixel $s \in S$, $s \notin \mathcal{G}_s$ and $s \in \mathcal{G}_r \Longleftrightarrow r \in \mathcal{G}_s$, for any $r \in S$. The elements of the set $\mathcal{G}_s$ are the neighbors of spatial location s. For example, in a four-point nearest-neighbor system, the neighbors of spatial location (i, j) are given by

$$\mathcal{G}_{(i,j)} = \{(i-1, j), (i, j-1), (i, j+1), (i+1, j)\} \tag{2.1}$$

for every interior pixel (i, j).

If, for every pixel $s \in S$, X_s is a discrete random variable, then $\mathbf{X}$ is a MRF with neighborhood system $\mathcal{G}$ if

$$P(X_s = x_s | X_r = x_r, r \neq s) = P(X_s = x_s | X_r = x_r, \forall r \in \mathcal{G}_s),\ \forall s \in S \tag{2.2}$$

where $P(A|B)$ is the conditional probability of the event A given the event B. Specification of a valid probability mass function in terms of the conditional probabilities of Equation 2.2 is difficult, but if $\mathbf{X}$ is defined to have a Gibbs distribution then it follows by the Hammersley-Clifford Theorem that $\mathbf{X}$ is a MRF [2, 25]. The form of a Gibbs distribution depends on the concept of cliques. A set of pixels $C \subseteq S$ is a clique if, for any pixels $s, r \in C$, $s \in \mathcal{G}_r$. Thus, the collection of all cliques, which we shall denote as $\mathcal{C}$, is induced by the neighborhood system. For example, in the four-point nearest-neighbor system, any pair of pixel locations in S which are horizontally or vertically adjacent form a clique, since two pixels which are horizontally or vertically adjacent are neighbors of each other. We shall assume that if $P(X_s = x_s) > 0\, \forall s$ then $P(\mathbf{X} = \mathbf{x}) > 0$, for any $\mathbf{x}$. This is referred to as the positivity condition. Assuming

this is true, it can be shown that $\mathbf{X}$ is a MRF if and only if $p_{\mathbf{X}}(\mathbf{x}) = P(\mathbf{X} = \mathbf{x})$ can be expressed in the form

$$p_{\mathbf{X}}(\mathbf{x}) = \frac{1}{z} \exp\left(-\frac{1}{T} \sum_{C \in \mathcal{C}} V_C(\mathbf{x})\right) \tag{2.3}$$

where $V_C(\mathbf{x})$ is a function which depends only on the values of $\mathbf{x}$ at pixel locations in the clique C, z is a normalizing constant, and T is a constant which represents "temperature" [25, 2]. A probability mass function which can be written in this form is referred to as a Gibbs distribution. A MRF can be specified in terms of the functions $V_C(\mathbf{x})$, $C \in \mathcal{C}$, instead of the conditional probabilities of Equation 2.2. This greatly simplifies the task of specifying a MRF.

Using the concepts discussed above, we now define the specific MRF model which will be used for the pixel label field in this chapter. The collection of cliques $\mathcal{G}$ will include all pairs of spatially horizontally or vertically adjacent pixels, plus all single pixels. The probability mass function of $\mathbf{X}$ is assumed to have the form

$$p_{\mathbf{X}}(\mathbf{x}) = \frac{1}{z} \exp\left(-\sum_{\{r,s\} \in \mathcal{C}} \beta t(x_r, x_s) - \sum_{\{r\} \in \mathcal{C}} \gamma_{x_r}\right) \tag{2.4}$$

where

$$t(x_r, x_s) = \begin{cases} 0 & \text{if } x_r = x_s \\ 1 & \text{if } x_r \neq x_s \end{cases} \tag{2.5}$$

The parameter β is known as the spatial interaction parameter, and $\{\gamma_k\}$ is a set of model parameters for single-pixel cliques. This model is similar to MRF models previously used for segmentation [6, 10, 7]. For every pixel $s \in S$, the set of values which the random variable X_s can take is $\{1, 2, \ldots, L\}$, where L is the number of different classes, or textures, in the image. This means that $\Omega_{\mathbf{X}} = \{\mathbf{x} : x_s \in \{1, 2, \ldots, L\} \forall s \in S\}$.

It will be assumed throughout this thesis that L is known *a priori*. The problem of automatically determining the number of classes is not addressed in this thesis, although it is an important problem which should be addressed in future work.

As is commonly done [25, 10, 9], we assume that the value of the spatial interaction parameter is known *a priori*. We have found experimentally that the optimal value of

β is not highly image-dependent, and that the performance of the algorithm remains fixed over a relatively large range of values of β.

The parameter γ_k can be viewed as a cost parameter for class k. If, for a given k, γ_k is high, then class k is less likely to occur than classes with lower costs. For applications in which there is *a priori* information about the relative sizes of the various classes, the parameters $\{\gamma_k\}$ can be selected to incorporate this information into the label field model. In the absence of such *a priori* information, γ_k will be assumed to be zero for every k.

2.1.2 Model for observed image

We also need a statistical model for the observed image. We will assume that the random variables $Y_1, Y_2, \ldots, Y_N$ are conditionally independent given the pixel label field $\mathbf{X}$. We will also assume that the conditional probability density function of Y_r given $\mathbf{X}$ depends only on the value of $\mathbf{X}$ at pixel location r. Using these two assumptions, the conditional probability density function of $\mathbf{Y}$ given $\mathbf{X}$ can be written as

$$\begin{aligned} f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x},\boldsymbol{\theta}) &= \prod_{r=1}^{N} f_{Y_r|\mathbf{X}}(y_r|\mathbf{x},\boldsymbol{\theta}) \\ &= \prod_{r=1}^{N} f_{Y_r|X_r}(y_r|x_r,\boldsymbol{\theta}) \end{aligned} \tag{2.6}$$

where $\boldsymbol{\theta}$ is a non-random vector whose elements are the unknown parameters of the conditional probability density function of $\mathbf{Y}$ given $\mathbf{X}$.

We will also model Y_r as conditionally Gaussian given X_r. The mean and variance of Y_r depend on the class to which pixel r belongs. Thus, all of the random variables in $\mathbf{Y}$ which represent class i, for any $i = 1, \ldots, L$, are independent and identically distributed (*iid*) Gaussian random variables with mean μ_i and variance σ_i^2. This model for the observed image has been used in previously proposed segmentation algorithms [10, 9].

The means and variances μ_i and σ_i^2, $i = 1, \ldots, L$, are the elements of the parameter vector $\boldsymbol{\theta}$, i.e., $\boldsymbol{\theta} = [\mu_1, \sigma_1^2, \ldots, \mu_L, \sigma_L^2]$. The space of all possible values of $\boldsymbol{\theta}$ will be denoted Ω_θ. We will assume that the elements of $\boldsymbol{\theta}$ are unknown.

Using the form of $f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$ given by Equation 2.6, the conditional probability density function of $\mathbf{Y}$ given $\mathbf{X}$ is

$$f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = \prod_{r=1}^{N} \frac{1}{\sqrt{2\pi\sigma_{x_r}^2}} \exp\left(-\frac{(y_r - \mu_{x_r})^2}{2\sigma_{x_r}^2}\right) \tag{2.7}$$

We will need to obtain the conditional probability mass function of $\mathbf{X}$ given $\mathbf{Y}$ to segment the image. Using Bayes' rule and Equations 2.4 and 2.7, we have

$$p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}) = \frac{f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) p_{\mathbf{X}}(\mathbf{x})}{f_{\mathbf{Y}}(\mathbf{y}|\boldsymbol{\theta})}$$

$$= \frac{1}{f_{\mathbf{Y}}(\mathbf{y}|\boldsymbol{\theta})} \left[\prod_{r=1}^{N} \frac{1}{\sqrt{2\pi\sigma_{x_r}^2}} \exp\left(-\frac{(y_r - \mu_{x_r})^2}{2\sigma_{x_r}^2}\right)\right] \left(\frac{1}{z}\right) \exp\left(-\sum_{\{r,s\}\in\mathcal{C}} \beta t(x_r, x_s) - \sum_{\{r\}\in\mathcal{C}} \gamma_{x_r}\right)$$

$$= \frac{1}{z f_{\mathbf{Y}}(\mathbf{y}|\boldsymbol{\theta})} \left[\prod_{r=1}^{N} \frac{1}{\sqrt{2\pi\sigma_{x_r}^2}}\right] \exp\left(-\sum_{r=1}^{N} \frac{(y_r - \mu_{x_r})^2}{2\sigma_{x_r}^2} - \sum_{\{r,s\}\in\mathcal{C}} \beta t(x_r, x_s) - \sum_{\{r\}\in\mathcal{C}} \gamma_{x_r}\right) \tag{2.8}$$

Since $f_{\mathbf{Y}}(\mathbf{y}|\boldsymbol{\theta})$ does not depend on $\mathbf{x}$, it is not considered in the optimization. It should be noted that $p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$ is also a Gibbs distribution.

2.2 MPM Segmentation Algorithm

In this section we assume that $\boldsymbol{\theta}$ is known and describe the MPM segmentation algorithm. For the MPM algorithm the segmentation problem is formulated as an optimization problem. The optimization criterion which is used is the minimization of the expected value of the number of misclassified pixels [3].

The image is segmented by minimizing the conditional expected value of a cost functional $R(\mathbf{X}, \mathbf{x})$, given the observed image $\mathbf{Y}$, over all possible realizations of the pixel label field $\mathbf{X}$, i.e., over all $\mathbf{x} \in \boldsymbol{\Omega}_{\mathbf{x}}$. The cost functional which will be used is

$$R(\mathbf{X}, \mathbf{x}) = \sum_{s=1}^{N} t(X_s, x_s) \tag{2.9}$$

where $t(\cdot,\cdot)$ is the function defined in Equation 2.5. Thus, $R(\mathbf{X},\mathbf{x})$ is the number of pixels at which $\mathbf{X}$ and $\mathbf{x}$ are not equal, i.e., the number of misclassified pixels. The value of $\mathbf{x}$ which minimizes the conditional expectation of this cost functional will be denoted as $\mathbf{x}^*$. Thus,

$$E[R(\mathbf{X},\mathbf{x}^*)|\mathbf{Y}=\mathbf{y}] \leq E[R(\mathbf{X},\mathbf{x})|\mathbf{Y}=\mathbf{y}] \quad \forall \mathbf{x} \in \Omega_{\mathbf{x}} \tag{2.10}$$

Using Equations 2.5 and 2.9, we have

$$\begin{aligned} E[R(\mathbf{X},\mathbf{x})|\mathbf{Y}=\mathbf{y}] &= E[\sum_{s=1}^{N} t(X_s,x_s)|\mathbf{Y}=\mathbf{y}] && (2.11)\\ &= \sum_{s=1}^{N} E[t(X_s,x_s)|\mathbf{Y}=\mathbf{y}] && (2.12)\\ &= \sum_{s=1}^{N} P(X_s \neq x_s|\mathbf{Y}=\mathbf{y}) && (2.13)\\ &= \sum_{s=1}^{N} (1-P(X_s=x_s|\mathbf{Y}=\mathbf{y})) && (2.14) \end{aligned}$$

We can minimize this sum by choosing for each pixel location $s \in S$ the value of x_s from the set $\{1,2,\ldots,L\}$ which minimizes $(1-P(X_s=x_s|\mathbf{Y}=\mathbf{y}))$ or, equivalently, the value which maximizes $P(X_s=x_s|\mathbf{Y}=\mathbf{y})$. The estimate of the label field $\mathbf{X}$ which is derived by independently maximizing this marginal conditional probability mass function for each $s \in S$ is known as the "maximizer of the posterior marginals" [3], and will be denoted $\mathbf{x}_{MPM}$. It can be seen from Equations 2.10 and 2.14 that $\mathbf{x}_{MPM}=\mathbf{x}^*$. Therefore, $\mathbf{x}_{MPM}$ minimizes over all $\mathbf{x}$ the expected number of misclassified pixels. To find $\mathbf{x}_{MPM}$ it is necessary to find for each $s \in S$ the value of k which maximizes

$$\begin{aligned} P(X_s=k|\mathbf{Y}=\mathbf{y}) &= p_{X_s|\mathbf{Y}}(k|\mathbf{y},\boldsymbol{\theta}) \\ &= \sum_{\mathbf{x}\in\Omega_{k,s}} p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y},\boldsymbol{\theta}) && (2.15) \end{aligned}$$

where $k \in \{1,2,\ldots,L\}$ and $\Omega_{k,s}=\{\mathbf{x}: x_s=k\}$. Exact computation of these marginal probability mass functions as in Equation 2.15 is computationally infeasible.

Marroquin et al. presented an algorithm for approximating these marginal probabilities to obtain an approximation to the MPM estimate of a MRF [3]. This algorithm

can be used to approximate $P(X_s = k|\mathbf{Y} = \mathbf{y})$ for each $s \in S$ and $k \in \{1, 2, \ldots, L\}$ as follows: Use the Gibbs sampler [25] to generate a discrete-time Markov chain $\mathbf{X}(t)$ which converges in distribution to a random field with probability mass function $p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$, given by Equation 2.8. The marginal conditional probability mass functions $p_{X_s|\mathbf{Y}}(k|\mathbf{y}, \boldsymbol{\theta})$, which are to be maximized, are then approximated as the fraction of time the Markov Chain spends in state k at pixel s, for each k and s.

For the Markov chain $\mathbf{X}(t)$ generated using the Gibbs sampler there are L^N possible states, corresponding to the L^N elements of $\Omega_{\mathbf{x}}$. At each step only one pixel is visited, so that $\mathbf{X}(t-1)$ and $\mathbf{X}(t)$ can differ at no more than one pixel location. At time t the state of $\mathbf{X}(t)$ at pixel s is a random variable $X_s(t)$. Let $q_t \in S$ be the pixel visited at time t. Then the state of $X_{q_t}(t)$ is determined by sampling from the conditional probability mass function $p_{X_{q_t}|\mathbf{Y},X_r,r\in\mathcal{G}_{q_t}}(k|\mathbf{y}, x_r(t-1), r \in \mathcal{G}_{q_t}, \boldsymbol{\theta})$.

If the sequence $\{q_1, q_2, q_3, \ldots\}$ contains every pixel $s \in S$ infinitely often, then for any initial configuration $\mathbf{x}(0) \in \Omega_{\mathbf{x}}$,

$$\lim_{t\to\infty} P(\mathbf{X}(t) = \mathbf{x}|\mathbf{Y} = \mathbf{y}, \mathbf{X}(0) = \mathbf{x}(0)) = p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}) \tag{2.16}$$

for every $\mathbf{x} \in \Omega_{\mathbf{x}}$ [25]. Thus, the Markov chain converges in distribution to a random field with probability mass function $p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$, i.e., $p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$ is the limiting distribution of the Markov chain. This fact will be used for the analysis of the EM/MPM algorithm presented in Chapter 4.

To describe the approximation of the marginal conditional probability mass function at each pixel we first define the function

$$u_{k,s}(t) = \begin{cases} 1 & \text{if } X_s(t) = k \\ 0 & \text{if } X_s(t) \neq k \end{cases} \tag{2.17}$$

Then, if T_s is the number of visits to pixel s made by the Gibbs sampler, then the approximations

$$p_{X_s|\mathbf{Y}}(k|\mathbf{y}, \boldsymbol{\theta}) \approx \frac{1}{T_s} \sum_{t=1}^{T_s} u_{k,s}(t) \qquad \forall k, s \tag{2.18}$$

provide the estimates of the values needed to obtain $\mathbf{x}_{MPM}$. At each pixel $s \in S$ the value of k which maximizes the right-hand side of of Equation 2.18 is used as the estimate of X_s.

2.3 EM Algorithm for Parameter Estimation

In order to implement the Gibbs sampler, we must estimate the value of $\boldsymbol{\theta}$. We will use the EM algorithm to estimate $\boldsymbol{\theta}$. The EM algorithm has been widely used for the estimation of parameters in incomplete-data problems [26, 27, 28, 29]. In an incomplete-data problem the observed data represent only a subset of the complete set of data. There also is a set of data which is unobserved, or hidden. For example, in our formulation the observed image $\mathbf{Y}$ represents the observed data, and the label field $\mathbf{X}$ represents the hidden data.

The EM algorithm approximates maximum-likelihood (ML) estimates. Finding the ML estimate of $\boldsymbol{\theta}$ based only on the observed data $\mathbf{Y}$ would involve finding the value $\boldsymbol{\theta}_{ML}$ which satisfies

$$f_{\mathbf{Y}}(\mathbf{y}|\boldsymbol{\theta}_{ML}) \geq f_{\mathbf{Y}}(\mathbf{y}|\boldsymbol{\theta}) \; \forall \boldsymbol{\theta} \in \Omega_{\theta} \tag{2.19}$$

where $f_{\mathbf{Y}}(\mathbf{y}|\boldsymbol{\theta})$ is the probability density function of $\mathbf{Y}$, parameterized by $\boldsymbol{\theta}$. The form of $f_{\mathbf{Y}}(\mathbf{y}|\boldsymbol{\theta})$ is not known, so the EM algorithm also uses the hidden data $\mathbf{X}$, since the form of $f_{\mathbf{XY}}(\mathbf{x},\mathbf{y}|\boldsymbol{\theta}) = f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x},\boldsymbol{\theta})p_{\mathbf{X}}(\mathbf{x})$ is known.

The EM algorithm is an iterative procedure. At each iteration two steps are performed: the expectation step and the maximization step. In general, if $\boldsymbol{\theta}(p)$ is the estimate of $\boldsymbol{\theta}$ at the pth iteration, then in the expectation step at iteration p the function

$$Q(\boldsymbol{\theta},\boldsymbol{\theta}(p-1)) = E[\log f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x},\boldsymbol{\theta})|\mathbf{Y}=\mathbf{y},\boldsymbol{\theta}(p-1)] + E[\log p_{\mathbf{X}}(\mathbf{x}|\boldsymbol{\theta})|\mathbf{Y}=\mathbf{y},\boldsymbol{\theta}(p-1)] \tag{2.20}$$

is computed. Since in our formulation the probability mass function of $\mathbf{X}$ does not depend on $\boldsymbol{\theta}$, we only use the first term of Equation 2.20. The estimate $\boldsymbol{\theta}(p)$ is obtained in the maximization step as the value of $\boldsymbol{\theta}$ which maximizes $Q(\boldsymbol{\theta},\boldsymbol{\theta}(p-1))$,

i.e., $\boldsymbol{\theta}(p)$ satisfies

$$Q(\boldsymbol{\theta}(p), \boldsymbol{\theta}(p-1)) \geq Q(\boldsymbol{\theta}, \boldsymbol{\theta}(p-1)) \;\; \forall \boldsymbol{\theta} \in \Omega_\theta \tag{2.21}$$

Substituting Equation 2.7 into Equation 2.20 gives

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}(p-1)) &= \sum_{s=1}^{N} E[\log(\frac{1}{\sqrt{2\pi\sigma_{x_s}^2}})|\mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1)] \;\; - \\ &\frac{1}{2}\sum_{s=1}^{N} E[\frac{(y_s - \mu_{x_s})^2}{\sigma_{x_s}^2}|\mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1)] \\ &= \sum_{s=1}^{N}\sum_{k=1}^{L} \log(\frac{1}{\sqrt{2\pi\sigma_k^2}}) P(X_s = k|\mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1)) \;\; - \\ &\frac{1}{2}\sum_{s=1}^{N}\sum_{k=1}^{L} \frac{(y_s - \mu_k)^2}{\sigma_k^2} P(X_s = k|\mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1)) \end{aligned} \tag{2.22}$$

Differentiating, setting to zero, and solving for $\boldsymbol{\theta}(p) = [\mu_1(p), \sigma_1^2(p), \ldots, \mu_L(p), \sigma_L^2(p)]$ gives

$$\mu_k(p) = \frac{1}{N_k(p)} \sum_{s=1}^{N} y_s p_{X_s|\mathbf{Y}}(k|\mathbf{y}, \boldsymbol{\theta}(p-1)) \tag{2.23}$$

and

$$\sigma_k^2(p) = \frac{1}{N_k(p)} \sum_{s=1}^{N} (y_s - \mu_k(p))^2 p_{X_s|\mathbf{Y}}(k|\mathbf{y}, \boldsymbol{\theta}(p-1)) \tag{2.24}$$

where

$$N_k(p) = \sum_{s=1}^{N} p_{X_s|\mathbf{Y}}(k|\mathbf{y}, \boldsymbol{\theta}(p-1)) \tag{2.25}$$

for $k = 1, \ldots, L$.

Equations 2.23 through 2.25 demonstrate the difficulty in using the EM algorithm with a MRF model for the pixel label field. The conditional probability mass functions $p_{X_s|\mathbf{Y}}(k|\mathbf{y}, \boldsymbol{\theta}(p))$ are very difficult to obtain when $\mathbf{X}$ is a MRF; in fact, these are the quantities we are trying to approximate to obtain $\mathbf{x}_{MPM}$. In general, this difficulty in obtaining $Q(\boldsymbol{\theta}, \boldsymbol{\theta}(p-1))$ is encountered when the hidden data are modeled as a MRF.

2.4 EM/MPM Algorithm

The EM/MPM algorithm [8, 9] combines the techniques described in Sections 2.2 and 2.3. First, the MPM algorithm is performed using an initial estimate of $\boldsymbol{\theta}$, say $\hat{\boldsymbol{\theta}}(0)$. After a certain number of iterations of the MPM algorithm, the resulting estimates of $p_{X_s|\mathbf{Y}}(k|\mathbf{y}, \hat{\boldsymbol{\theta}}(0))$ are used in Equations 2.23 through 2.25 to obtain an updated estimate of $\boldsymbol{\theta}$, say $\hat{\boldsymbol{\theta}}(1)$. This new estimate of $\boldsymbol{\theta}$ is then used in the MPM algorithm to find estimates of $p_{X_s|\mathbf{Y}}(k|\mathbf{y}, \hat{\boldsymbol{\theta}}(1))$, which are then used to update the estimate of $\boldsymbol{\theta}$. This process is continued until some suitable stopping point is reached.

The EM/MPM algorithm generates a (finite) collection of Markov chains $\mathbf{X}(1,t), \mathbf{X}(2,t), \ldots, \mathbf{X}(P+1,t)$, for some $P \geq 1$. Generation of $\mathbf{X}(p,t)$ is referred to as stage p of the algorithm. The estimate of $\boldsymbol{\theta}$ obtained during stage p is denoted by the random variable $\boldsymbol{\Theta}(p)$. The algorithm begins with the estimate $\boldsymbol{\Theta}(0) = \hat{\boldsymbol{\theta}}(0)$ for some $\hat{\boldsymbol{\theta}}(0) \in \Omega_\theta$. The Markov chain $\mathbf{X}(1,t)$ is generated using the procedure described in Section 2.2. The state of $X_{q_t}(1,t)$ is determined by sampling from the conditional probability mass function $p_{X_{q_t}|\mathbf{Y}, X_r, r\in\mathcal{G}_{q_t}, \boldsymbol{\Theta}(0)}(k|\mathbf{y}, x_r(1,t-1), r \in \mathcal{G}_{q_t}, \hat{\boldsymbol{\theta}}(0))$, where $p_{\mathbf{X}|\mathbf{Y},\boldsymbol{\Theta}(p)}(\mathbf{x}|\mathbf{y}, \hat{\boldsymbol{\theta}}(p))$, for any $p \geq 1$, has the same form as $p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$ given by Equation 2.8, with $\boldsymbol{\Theta}(p)$ random, unlike $\boldsymbol{\theta}$, which is deterministic. Thus, if $\hat{\boldsymbol{\theta}}(p) = [\hat{\mu}_1(p), \hat{\sigma}_1^2(p), \ldots, \hat{\mu}_L(p), \hat{\sigma}_L^2(p)]$, then

$$p_{\mathbf{X}|\mathbf{Y},\boldsymbol{\Theta}(p)}(\mathbf{x}|\mathbf{y}, \hat{\boldsymbol{\theta}}(p)) = \frac{1}{z f_{\mathbf{Y},\boldsymbol{\Theta}(p)}(\mathbf{y}|\hat{\boldsymbol{\theta}}(p))} \left[\prod_{r=1}^{N} \frac{1}{\sqrt{2\pi\hat{\sigma}_{x_r}^2(p)}}\right] \cdot$$

$$\exp\left(-\sum_{r=1}^{N} \frac{(y_r - \hat{\mu}_{x_r}(p))^2}{2\hat{\sigma}_{x_r}^2(p)} - \sum_{\{r,s\}\in\mathcal{C}} \beta t(x_r, x_s) - \sum_{\{r\}\in\mathcal{C}} \gamma_{x_r}\right) \tag{2.26}$$

After each pixel has been visited T_1 times, for some $T_1 \geq 1$, the estimate $\boldsymbol{\Theta}(1)$ is computed. Using the estimates $v_{k,s}(1,t)$ defined by (See the right-hand side of Equation 2.18)

$$v_{k,s}(1,t) = \frac{1}{t}\sum_{i=1}^{t} u_{k,s}(1,i) \tag{2.27}$$

where

$$u_{k,s}(1,t) = \begin{cases} 1 & \text{if } X_s(1,t) = k \\ 0 & \text{if } X_s(1,t) \neq k \end{cases} \tag{2.28}$$

the estimate $\boldsymbol{\Theta}(1) = [M_1(1), S_1(1), \ldots, M_L(1), S_L(1)]$ is computed using

$$M_k(1) = \frac{\sum_{s=1}^{N} y_s v_{k,s}(1, T_1)}{\sum_{s=1}^{N} v_{k,s}(1, T_1)} \tag{2.29}$$

and

$$S_k(1) = \frac{\sum_{s=1}^{N} (y_s - M_k(1))^2 v_{k,s}(1, T_1)}{\sum_{s=1}^{N} v_{k,s}(1, T_1)} \tag{2.30}$$

Note that these equations have the same form as the EM update equations (Equations 2.23 and 2.24), with $p_{X_s|\mathbf{Y}}(k|\mathbf{y}, \boldsymbol{\theta})$ replaced by the estimates $v_{k,s}(1, T_1)$.

The Markov chain $\mathbf{X}(2,t)$ is generated next using the procedure in Section 2.2, with limiting distribution $p_{\mathbf{X}|\mathbf{Y},\boldsymbol{\Theta}(1)}(\mathbf{x}|\mathbf{y}, \hat{\boldsymbol{\theta}}(1))$, where $\hat{\boldsymbol{\theta}}(1) \in \Omega_\theta$ is the value obtained for $\boldsymbol{\Theta}(1)$. After T_2 visits to each pixel, $\boldsymbol{\Theta}(2)$ is computed.

In general, the Markov chain $\mathbf{X}(p,t)$ is generated using the Gibbs sampler and sampling from the distribution $p_{\mathbf{X}|\mathbf{Y},\boldsymbol{\Theta}(p-1)}(\mathbf{x}|\mathbf{y}, \hat{\boldsymbol{\theta}}(p-1))$, and the estimate $\boldsymbol{\Theta}(p)$ is computed using the equations

$$M_k(p) = \frac{\sum_{s=1}^{N} y_s v_{k,s}(p, T_p)}{\sum_{s=1}^{N} v_{k,s}(p, T_p)} \tag{2.31}$$

and

$$S_k(p) = \frac{\sum_{s=1}^{N} (y_s - M_k(p))^2 v_{k,s}(p, T_p)}{\sum_{s=1}^{N} v_{k,s}(p, T_p)} \tag{2.32}$$

The final estimate of $\boldsymbol{\theta}$ is $\boldsymbol{\Theta}(P)$. The final segmentation is obtained by maximizing over all k the value $v_{k,s}(P+1, T_{P+1})$ for every $s \in S$, for some $T_{P+1} \geq 1$.

The algorithm can be summarized as follows: First, initial estimates $\hat{\boldsymbol{\theta}}(0)$ and $\mathbf{X}(1,0)$ of $\boldsymbol{\theta}$ and $\mathbf{X}$, respectively, are selected. Then, for $p = 1, \ldots, P$, stage p of the algorithm consists of two steps:

1. Perform T_p iterations of the MPM algorithm using $\hat{\boldsymbol{\theta}}(p-1)$ as the value of $\boldsymbol{\theta}$.

2. Use the EM update equations for $\boldsymbol{\theta}$ to obtain $\hat{\boldsymbol{\theta}}(p)$, using the values $v_{k,s}(p, T_p)$ as estimates of $p_{X_s|Y}(k|\mathbf{y}, \boldsymbol{\theta}(p-1))$.

After $\hat{\boldsymbol{\theta}}(P)$ has been obtained as the final estimate of $\boldsymbol{\theta}$, T_{P+1} iterations of the MPM algorithm are performed, using $\hat{\boldsymbol{\theta}}(P)$. The final segmentation is $\mathbf{X}(P+1, T_{P+1})$.

2.5 Experimental Results

The EM/MPM algorithm was applied to several different types of imagery. Results demonstrating the performance of the algorithm on synthetic imagery, infrared imagery, natural scenes, and mammograms are presented in this section. For all results presented in this section, the value of the spatial interaction parameter β was assumed to be 2.4, and T_p, the number of iterations of the Gibbs sampler at stage p, was set to 3 for every $p = 1, \ldots, P+1$. Unless otherwise noted, the relative cost of class k (i.e., γ_k in Equation 2.4) is assumed to be zero, for every k. Also, unless otherwise noted, initial estimates of the class means and variances were obtained using

$$\mu_k(0) = \frac{128}{L} + \frac{255k}{L} \tag{2.33}$$

for the means and setting the variance for each class to 20. For every pixel $s \in S$ the initial estimate of X_s was chosen to be uniformly distributed over all classes, and independent of the initial estimates at other pixels.

Figure 2.1(a) shows the first test image. This image is a composite of Brodatz textures wood and grass. The segmented image obtained from 70 stages of the EM/MPM algorithm is shown in Figure 2.1(b), and the segmentation after 500 iterations of the deterministic EM/MPM algorithm proposed in [9] is shown in Figure 2.1(c). It can be

seen that for this image the EM/MPM algorithm provides better performance than the deterministic EM/MPM algorithm.

The second test image, shown in Figure 2.2(a), is a composite of Brodatz textures cork and leather. The segmentation obtained after 300 stages of the EM/MPM algorithm is shown in Figure 2.2(b), and the segmentation after 500 iterations of the deterministic EM/MPM algorithm is shown in Figure 2.2(c). For this image the EM/MPM algorithm performs significantly better than the deterministic EM/MPM algorithm. The EM/MPM algorithm does have difficulty correctly classifying the pixels in the cork region. This is because the observed image model does not fit the cork texture well. It will be shown in the next chapter that the multiresolution EM/MPM algorithm provides better results for this image.

The third test image is shown in Figure 2.3(a). This image is a composite of Brodatz textures grass, pigskin, and straw. Results after 400 stages of the EM/MPM algorithm and 500 iterations of the deterministic EM/MPM algorithm are shown in Figures 2.3(b) and 2.3(c), respectively. Both algorithms have difficulty segmenting this image, although the EM/MPM algorithm again provides a better segmented image than the deterministic EM/MPM algorithm.

The final synthetic texture image is shown in Figure 2.4(a). This image is a composite of Brodatz textures grass and wood, and Corel texture leather. The resulting segmentation after 400 stages of the EM/MPM algorithm is shown in Figure 2.4(b), and the result after 500 iterations of the deterministic EM/MPM algorithm is shown in Figure 2.4(c). For this image the EM/MPM algorithm performs slightly better than the deterministic EM/MPM algorithm, although both algorithms have difficulty classifying the regions correctly.

Table 2.1 shows the percentage of pixels which were misclassified by the EM/MPM and deterministic EM/MPM algorithms for the four synthetic test images. For all images the EM/MPM algorithm performed better than the deterministic EM/MPM algorithm in terms of minimizing the number of misclassified pixels.

Table 2.1 Percentage of Misclassified Pixels

Image	EM/MPM	Deterministic EM/MPM
Figure 2.1(a)	2.4	5.0
Figure 2.2(a)	4.8	40.0
Figure 2.3(a)	39.2	51.7
Figure 2.4(a)	30.0	35.1

Figures 2.5 and 2.6 illustrate intermediate results as the EM/MPM and deterministic EM/MPM algorithms segment the image shown in Figure 2.1(a). Figures 2.5(a), (b), (c), and (d) show results after 30, 50, 70, and 300 stages, respectively, of the EM/MPM algorithm, and Figures 2.6(a), (b), (c), and (d) show results after 10, 20, 500, and 800 iterations, respectively, of the deterministic EM/MPM algorithm [9]. Table 2.2 shows the final parameter estimates obtained after 300 stages of the EM/MPM algorithm and 800 iterations of the deterministic EM/MPM algorithm, as well as the sample means and sample standard deviations computed using the true segmentation. The EM/MPM estimates of the means for the two classes are closer to the sample means than the deterministic EM/MPM estimates. The accuracy of the estimates of the means is critical to the performance of both algorithms. It is our feeling that the deterministic EM/MPM algorithm either overestimates or underestimates the model parameters.

The natural scene shown in Figure 2.7(a) was segmented using the EM/MPM algorithm. The result shown in Figure 2.7(b) was obtained using 70 stages of the algorithm, with $L = 2$ (where L is the number of classes in the image). Figure 2.7(c) shows the result from 70 stages of the algorithm with $L = 3$. In general, trees and buildings were classified to one class, and the ground and river were classified to another class.

Table 2.2 Parameter Estimates for Image in Figure 2.1(a)

	μ_1	σ_1	μ_2	σ_2
Sample Mean/Sample Standard Deviation	83.0	21.6	122.7	54.6
Initial Estimate	64.0	4.5	191.0	4.5
Final EM/MPM Estimate	81.7	20.7	122.9	53.6
Final Deterministic EM/MPM Estimate	80.9	21.5	126.6	52.8

The algorithm was also tested on infrared imagery. Figure 2.8 is composed of 16 infrared images which were segmented using the EM/MPM algorithm. The resulting segmentations are shown in Figures 2.9 through 2.24. In each of these figures the image shown in (b) is the segmentation resulting from 70 stages of EM/MPM with $L = 2$ and the image shown in (c) is the result after 70 stages with $L = 3$. The segmentations obtained using the deterministic EM/MPM algorithm on the infrared images were similar to the segmentations obtained using the EM/MPM algorithm except for the image shown in Figure 2.22. The result after 500 iterations of the deterministic EM/MPM algorithm for this image is shown in Figure 2.25(b). For this image the deterministic EM/MPM algorithm does not perform as well as the EM/MPM algorithm.

To segment mammography images, we use the *a priori* knowledge that a tumor is expected to cover a relatively small region compared to the normal tissue and the background. This information is incorporated into the label field model by using nonzero values for the parameters γ_k. The effect of this is to increase the cost at each pixel of belonging to the class coresponding to the tumor, as described in Section 2.1.1. Also, the fact that a tumor is expected to be relatively small and the fact that a tumor is usually associated with higher grayscale values than the other regions are used to compute initial estimates of the class means and variances. The grayscale values in the observed image are sorted, and the sample mean and variance of the

highest grayscale values are used for the initial parameter estimates for the tumor class. The remaining values are used to compute initial estimates of the parameters for the normal tissue and background region.

The mammography images were assumed to consist of three classes: background, normal tissue, and tumor. The values used for the class cost parameters were $\gamma_1 = \gamma_2 = 2.3$ and $\gamma_3 = 5$, where class 3 is the tumor class. These values were determined experimentally using a variety of sample mammography images.

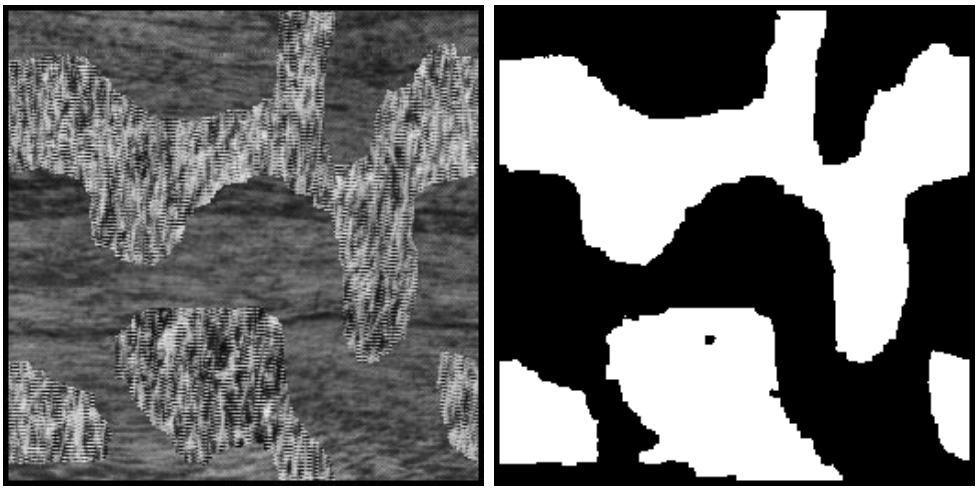
The first mammography image which was tested is shown in Figure 2.26(a). The corresponding truth image is shown in Figure 2.26(b). The segmented image obtained after 100 stages of the EM/MPM algorithm is shown in Figure 2.26(c). It can be seen that the algorithm segmented the three regions quite well.

The second mammography image is shown in Figure 2.27(a). The corresponding truth image is shown in Figure 2.27(b). The segmented image obtained after 70 stages of the EM/MPM algorithm is shown in Figure 2.26(c). Again, the algorithm segmented the three regions well.

The final mammography image is the synthetic image shown in Figure 2.28(a). The segmented image obtained after 70 stages of the EM/MPM algorithm is shown in Figure 2.28(b). The algorithm does not perform well on this image, finding many very small regions corresponding to local variations in the tissue.

2.6 Conclusion

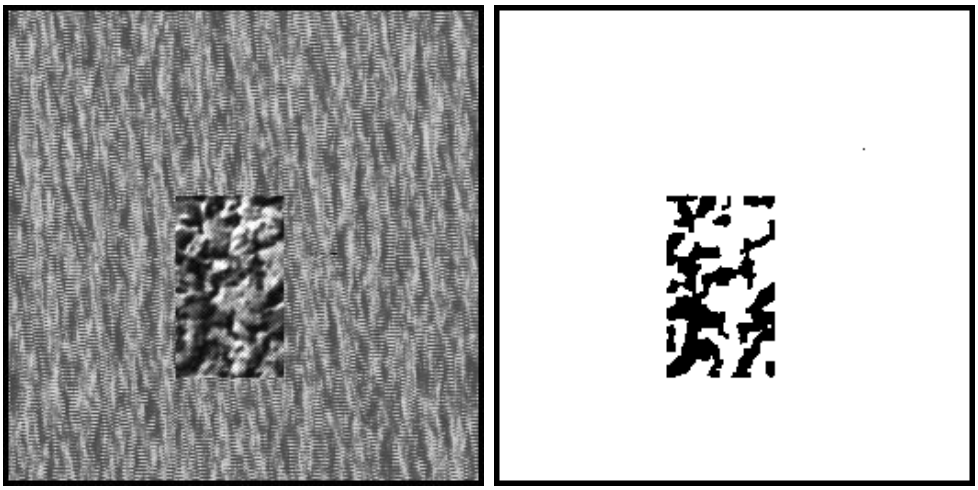
In this chapter we described the EM/MPM algorithm for simultaneous parameter estimation and segmentation of textured images. We presented experimental results demonstrating the performance of the algorithm, and compared these results with those obtained by the deterministic EM/MPM algorithm. Our results for synthetic images showed that the EM/MPM algorithm performs better than the deterministic EM/MPM algorithm in terms of minimizing the number of misclassified pixels.



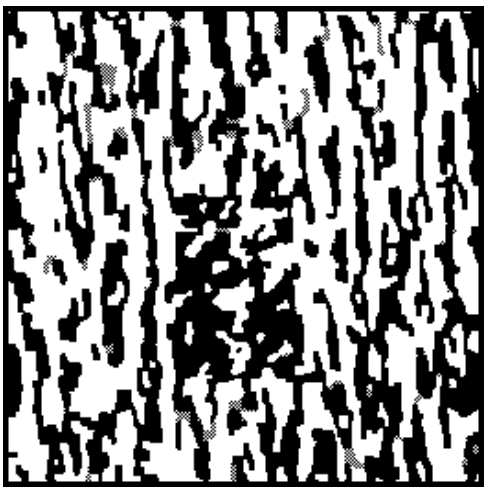
(a) (b)

(c)

Fig. 2.1. (a): Original image. (b): Segmented image obtained using EM/MPM. (c): Segmented image obtained using deterministic EM/MPM.

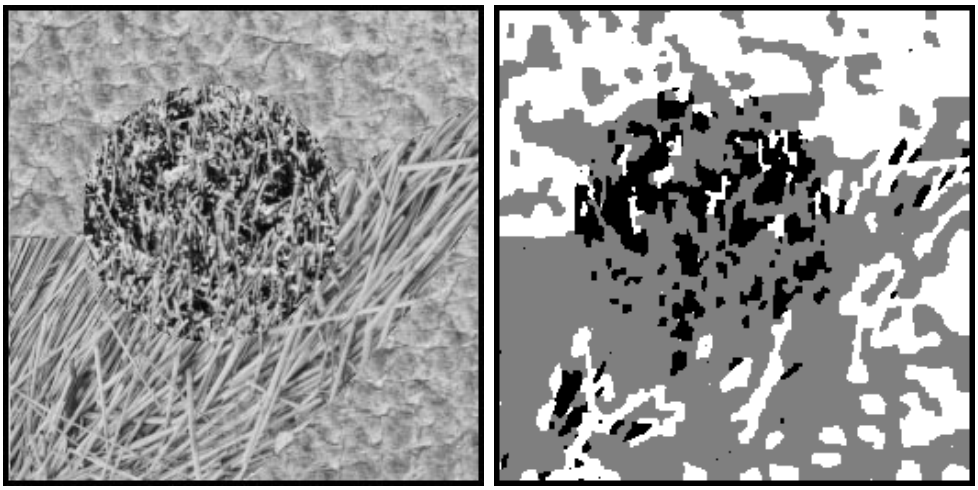


(a) (b)

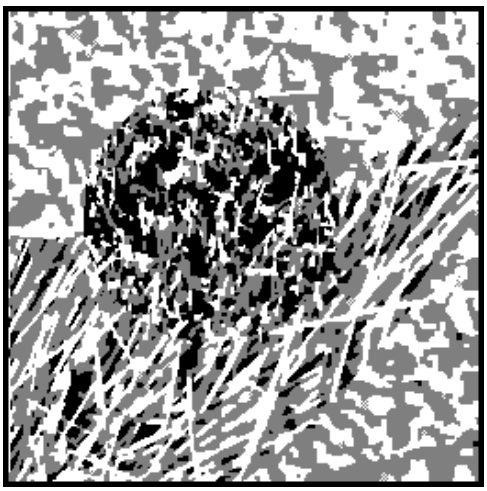


(c)

Fig. 2.2. (a): Original image. (b): Segmented image obtained using EM/MPM. (c): Segmented image obtained using deterministic EM/MPM.

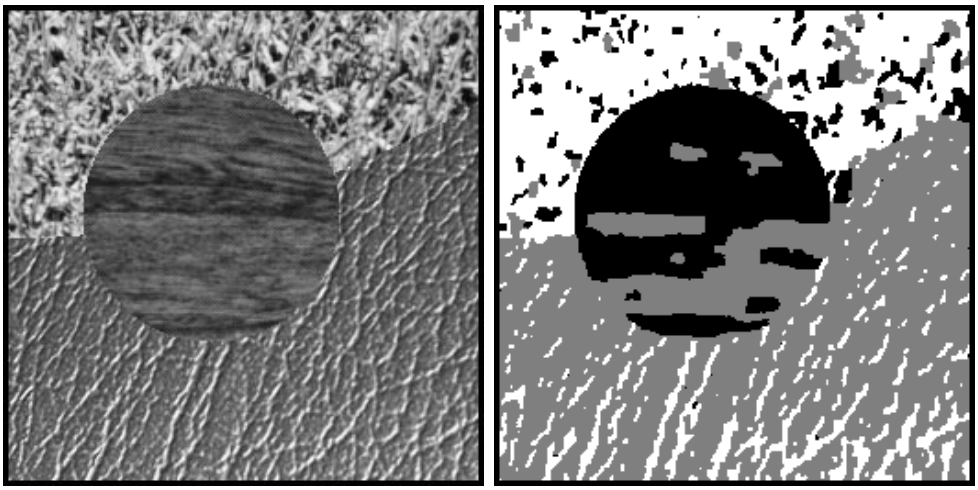


(a) (b)

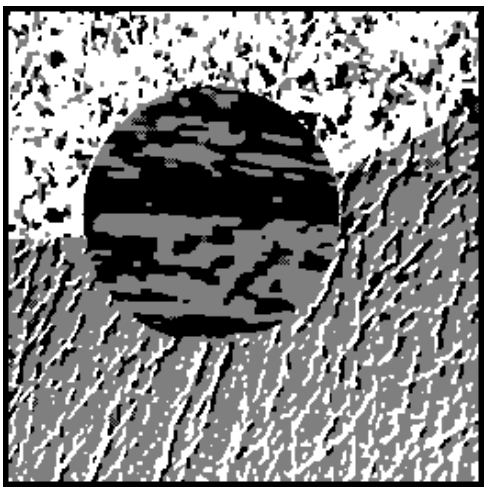


(c)

Fig. 2.3. (a): Original image. (b): Segmented image obtained using EM/MPM. (c): Segmented image obtained using deterministic EM/MPM.

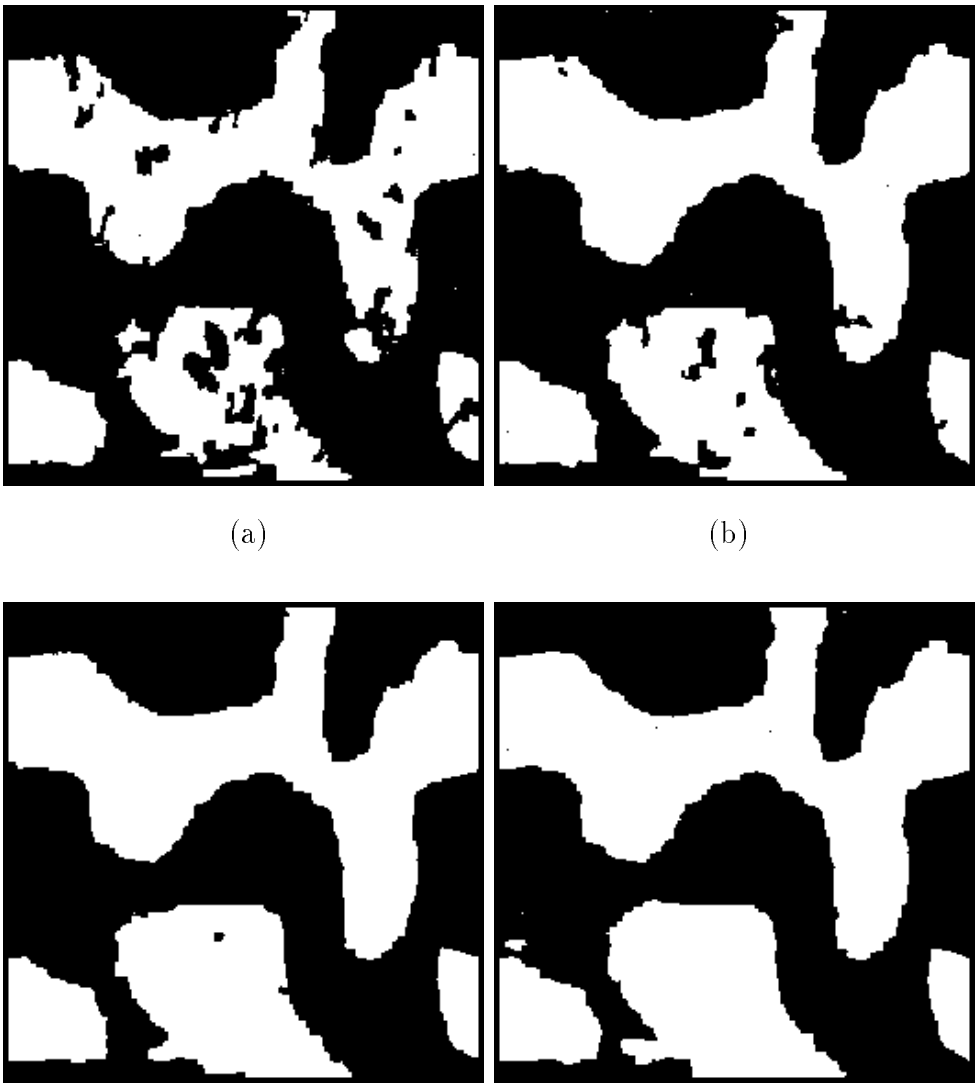


(a) (b)



(c)

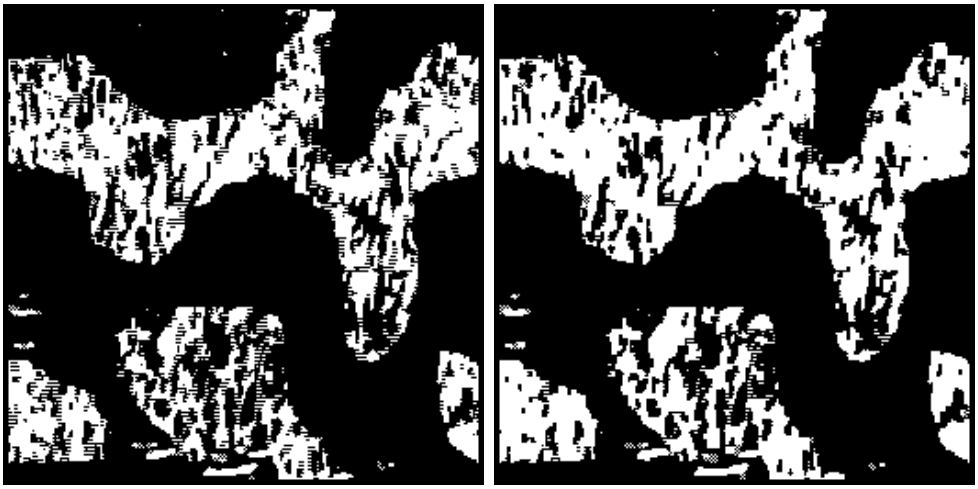
Fig. 2.4. (a): Original image. (b): Segmented image obtained using EM/MPM. (c): Segmented image obtained using deterministic EM/MPM.



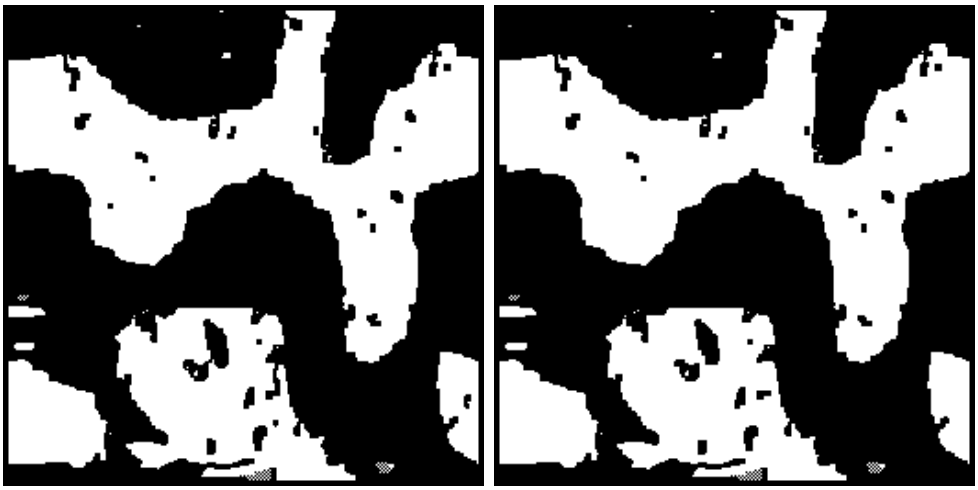
(a) (b)

(c) (d)

Fig. 2.5. (a): Segmented image after 30 stages of EM/MPM. (b): Segmented image after 50 stages of EM/MPM. (c): Segmented image after 70 stages of EM/MPM. (d): Segmented image after 300 stages of EM/MPM.

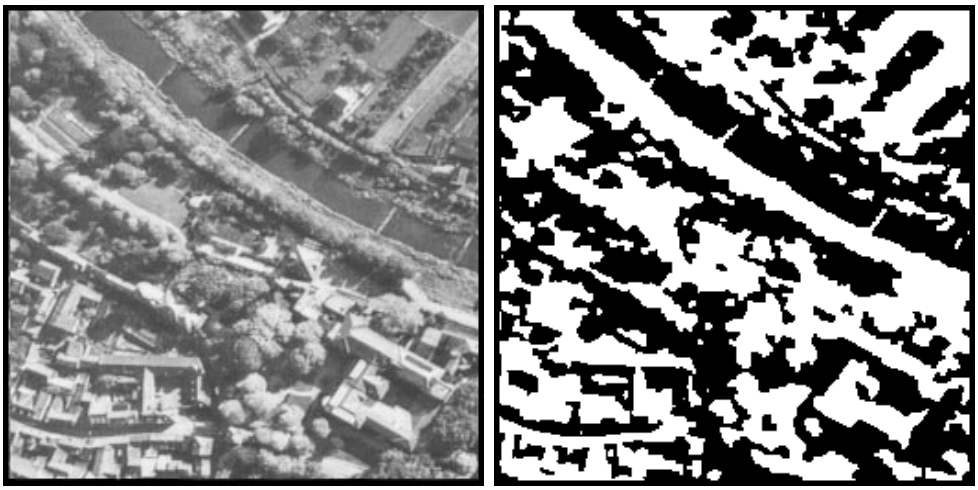


(a) (b)

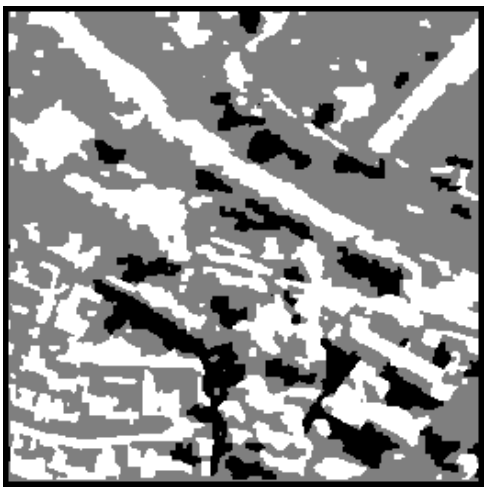


(c) (d)

Fig. 2.6. (a): Segmented image after 10 iterations of deterministic EM/MPM. (b): Segmented image after 20 iterations of deterministic EM/MPM. (c): Segmented image after 500 iterations of deterministic EM/MPM. (d): Segmented image after 800 iterations of deterministic EM/MPM.



(a) (b)



(c)

Fig. 2.7. (a): Original image. (b): Segmented image obtained using EM/MPM with 2 classes. (c): Segmented image obtained using EM/MPM with 3 classes.

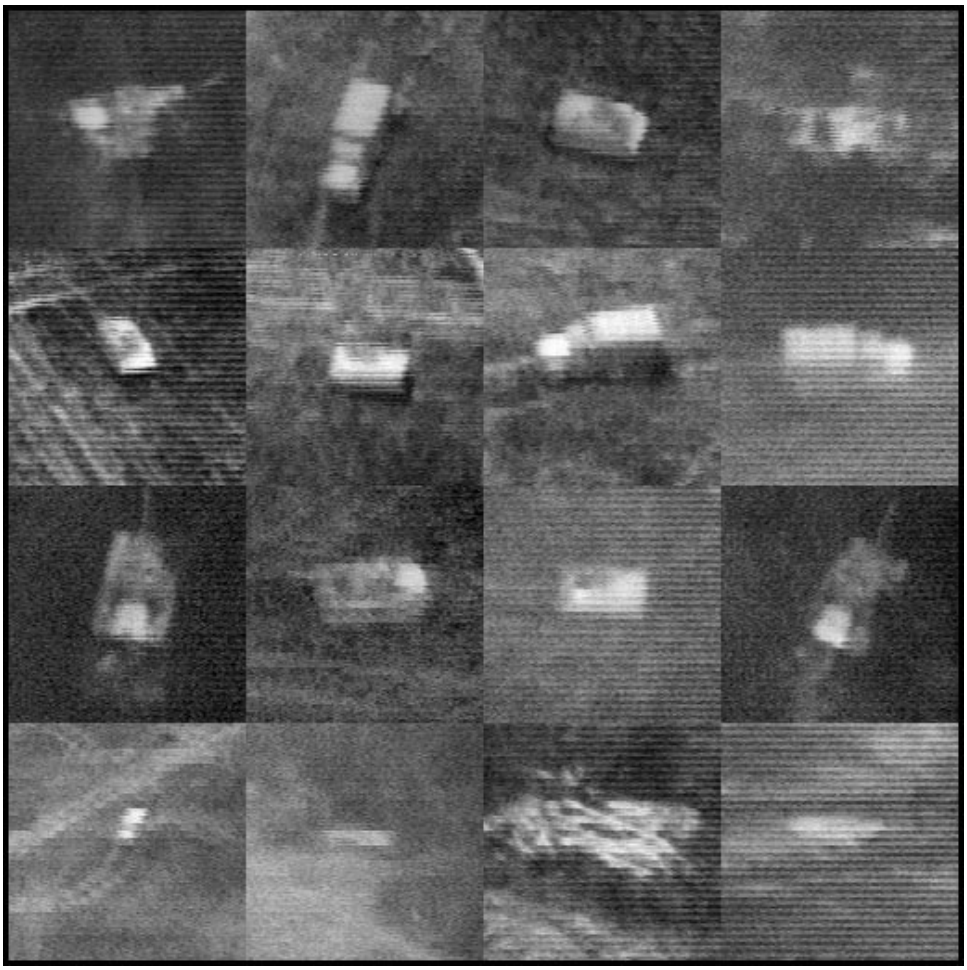
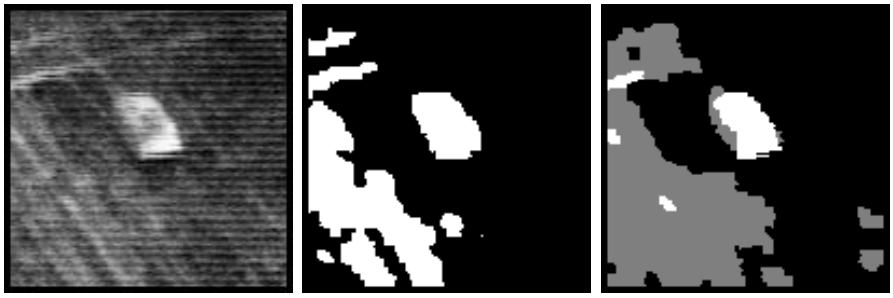


Fig. 2.8. Original infrared image.

(a) (b) (c)

Fig. 2.9. (a): Original image. (b): Segmented image obtained using EM/MPM with 2 classes. (c): Segmented image obtained using EM/MPM with 3 classes.



(a) (b) (c)

Fig. 2.10. (a): Original image. (b): Segmented image obtained using EM/MPM with 2 classes. (c): Segmented image obtained using EM/MPM with 3 classes.

(a) (b) (c)

Fig. 2.11. (a): Original image. (b): Segmented image obtained using EM/MPM with 2 classes. (c): Segmented image obtained using EM/MPM with 3 classes.

(a) (b) (c)

Fig. 2.12. (a): Original image. (b): Segmented image obtained using EM/MPM with 2 classes. (c): Segmented image obtained using EM/MPM with 3 classes.

(a) (b) (c)

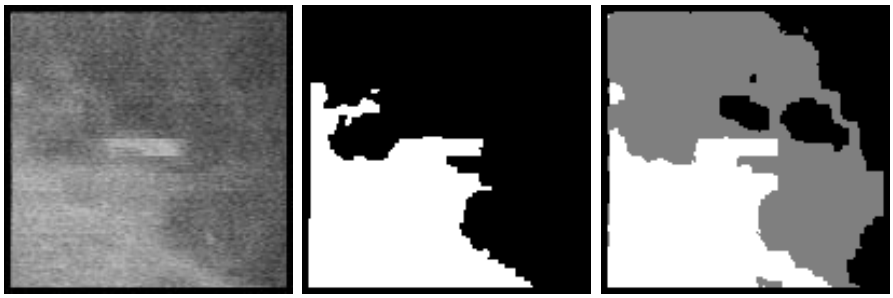
Fig. 2.13. (a): Original image. (b): Segmented image obtained using EM/MPM with 2 classes. (c): Segmented image obtained using EM/MPM with 3 classes.

(a) (b) (c)

Fig. 2.14. (a): Original image. (b): Segmented image obtained using EM/MPM with 2 classes. (c): Segmented image obtained using EM/MPM with 3 classes.

(a) (b) (c)

Fig. 2.15. (a): Original image. (b): Segmented image obtained using EM/MPM with 2 classes. (c): Segmented image obtained using EM/MPM with 3 classes.



(a) (b) (c)

Fig. 2.16. (a): Original image. (b): Segmented image obtained using EM/MPM with 2 classes. (c): Segmented image obtained using EM/MPM with 3 classes.

(a) (b) (c)

Fig. 2.17. (a): Original image. (b): Segmented image obtained using EM/MPM with 2 classes. (c): Segmented image obtained using EM/MPM with 3 classes.

(a) (b) (c)

Fig. 2.18. (a): Original image. (b): Segmented image obtained using EM/MPM with 2 classes. (c): Segmented image obtained using EM/MPM with 3 classes.

(a) (b) (c)

Fig. 2.19. (a): Original image. (b): Segmented image obtained using EM/MPM with 2 classes. (c): Segmented image obtained using EM/MPM with 3 classes.

(a) (b) (c)

Fig. 2.20. (a): Original image. (b): Segmented image obtained using EM/MPM with 2 classes. (c): Segmented image obtained using EM/MPM with 3 classes.

(a) (b) (c)

Fig. 2.21. (a): Original image. (b): Segmented image obtained using EM/MPM with 2 classes. (c): Segmented image obtained using EM/MPM with 3 classes.

(a) (b) (c)

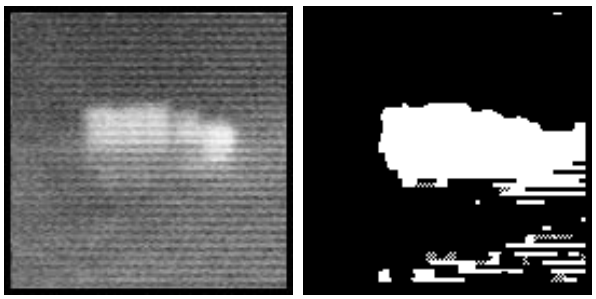
Fig. 2.22. (a): Original image. (b): Segmented image obtained using EM/MPM with 2 classes. (c): Segmented image obtained using EM/MPM with 3 classes.

(a) (b) (c)

Fig. 2.23. (a): Original image. (b): Segmented image obtained using EM/MPM with 2 classes. (c): Segmented image obtained using EM/MPM with 3 classes.

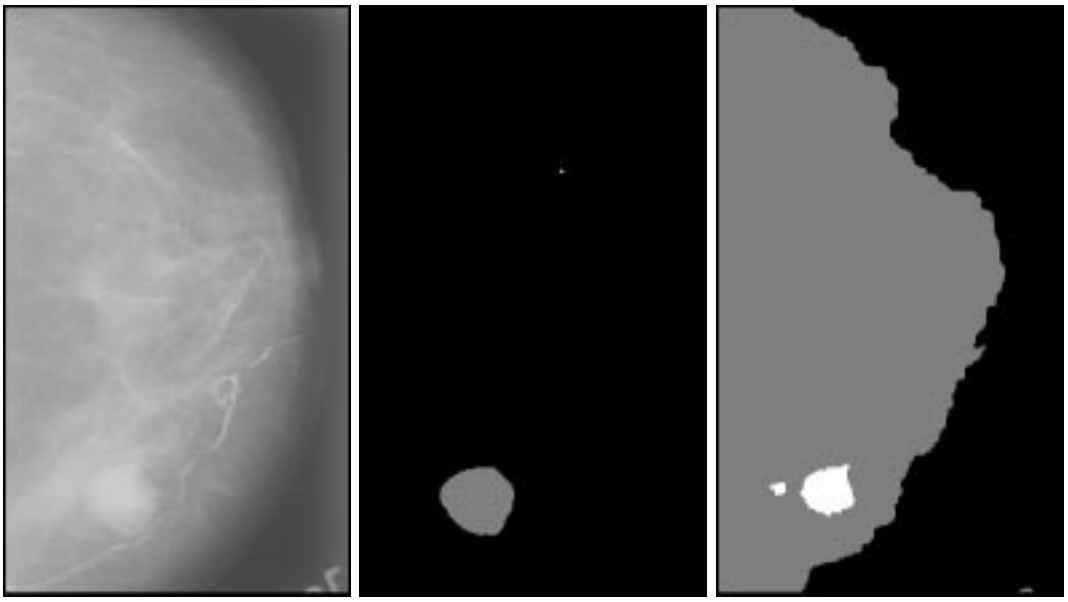
(a) (b) (c)

Fig. 2.24. (a): Original image. (b): Segmented image obtained using EM/MPM with 2 classes. (c): Segmented image obtained using EM/MPM with 3 classes.



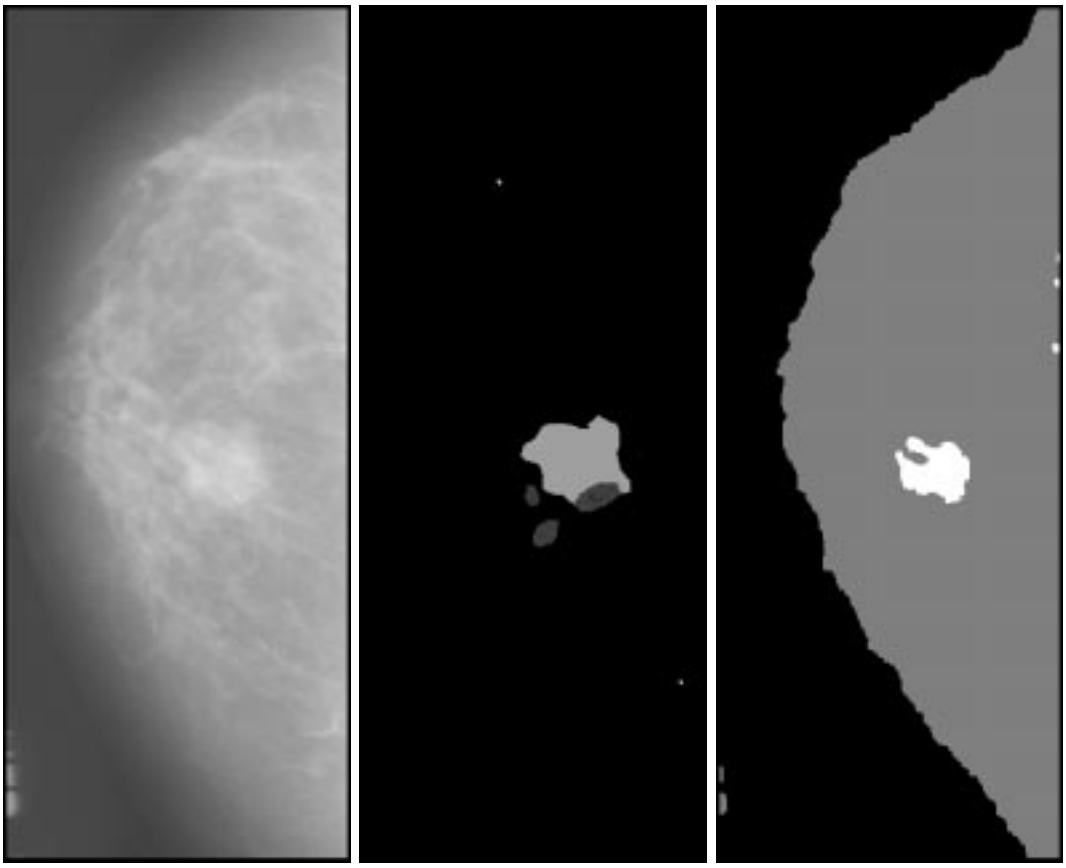
(a) (b)

Fig. 2.25. (a): Original image. (b): Segmented image obtained using deterministic EM/MPM.



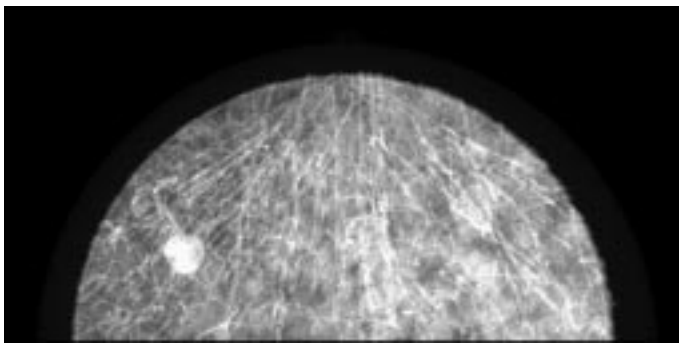
(a) (b) (c)

Fig. 2.26. (a): Original image. (b): Truth image. (c): Segmented image obtained using EM/MPM.

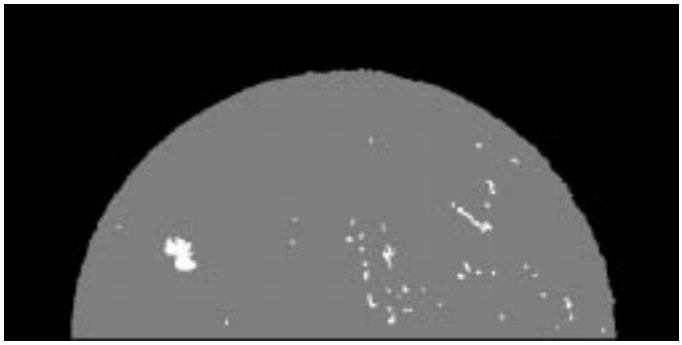


(a) (b) (c)

Fig. 2.27. (a): Original image. (b): Truth image. (c): Segmented image obtained using EM/MPM.



(a)



(b)

Fig. 2.28. (a): Original image. (b): Segmented image obtained using EM/MPM.

3. THE MULTIRESOLUTION EM/MPM ALGORITHM

This chapter describes a multiresolution extension of the EM/MPM algorithm described in the previous chapter. The multiresolution EM/MPM algorithm uses a multiresolution Gaussian autoregressive (MGAR) model for the pyramid representation of the observed image, and assumes a multiscale Markov random field model for the class label pyramid, and the criterion used for segmentation is the minimization of the expected value of the number of misclassified nodes in the multiresolution lattice.

In Section 3.1 we describe the statistical models used for the observed image pyramid and the label pyramid. In Section 3.2 we present the segmentation algorithm for the case when all the model parameters are known, and in Section 3.3 a method for estimating the parameters of the MGAR model is proposed. Section 3.4 describes the new multiresolution algorithm proposed for simultaneous segmentation and parameter estimation, combining the techniques discussed in Sections 3.2 and 3.3. Experimental results demonstrating the performance of the algorithm are presented in Section 3.5.

3.1 Statistical Models

In this chapter the observed data $\mathbf{Y}$ is a multiresolution representation of the observed image. Thus, $\mathbf{Y}$ is a stochastic process indexed by the nodes of a multiresolution lattice, such as the one shown in Figure 3.1. The class label pyramid $\mathbf{X}$ is defined on the same multiresolution lattice as $\mathbf{Y}$, and contains the classifications of the nodes in the lattice. The set of nodes in the lattice is denoted S. Each level in the lattice corresponds to a different spatial resolution, where level 0 represents the finest spatial resolution and level $M-1$ the coarsest spatial resolution. The set of lattice points at level n will be denoted $S^{(n)}$ and the random fields containing the observed image

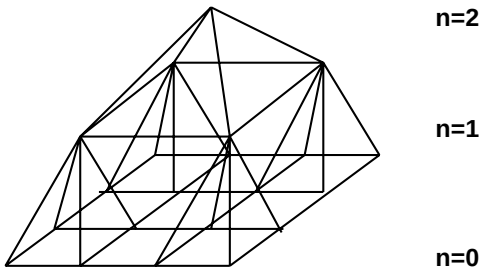


Fig. 3.1. Multiresolution lattice

at resolution n and the classifications of the nodes at resolution n will be denoted $\mathbf{Y}^{(n)}$ and $\mathbf{X}^{(n)}$, respectively. Each node in $S^{(n)}$ corresponds to 4^n pixels at the original image spatial resolution. Hence, each random variable in $\mathbf{X}^{(n)}$ represents the classification of a block of 4^n pixels in the original image. The random variable in $\mathbf{Y}^{(n)}$ at node $(i,j) \in S^{(n)}$ will be denoted $Y_{i,j}^{(n)}$, for $i = 0, \ldots, (I/2^n)-1$; $j = 0, \ldots, (J/2^n)-1$, where I and J are the number of rows and columns, respectively, in the original image. This notation will also be used for the random variables in $\mathbf{X}^{(n)}$. To simplify the notation, the random variables in $\mathbf{Y}$ and $\mathbf{X}$ may also be indexed by a single letter, as in a lexicographical ordering, in which case the sth random variable in $\mathbf{Y}$ will be denoted by Y_s, and the sth random variable in $\mathbf{X}$ will be X_s. Throughout this chapter, $\mathbf{y} = (y_1, y_2, \ldots, y_N)$ and $\mathbf{x} = (x_1, x_2, \ldots, x_N)$, where N is the total number of nodes in S, will represent sample realizations of $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_N)$ and $\mathbf{X} = (X_1, X_2, \ldots, X_N)$.

For every node $s \in S$, the set of values which the random variable X_s can take is $\{1, 2, \ldots, L\}$, where L is the number of different classes, or textures, in the image. We shall assume that L is known.

To segment the observed image, two models will be needed. A model will be needed for the conditional probability density function of $\mathbf{Y}$ given the classifications of all nodes in the multiresolution lattice. We will also need a model for the conditional probability mass function of the label pyramid $\mathbf{X}$.

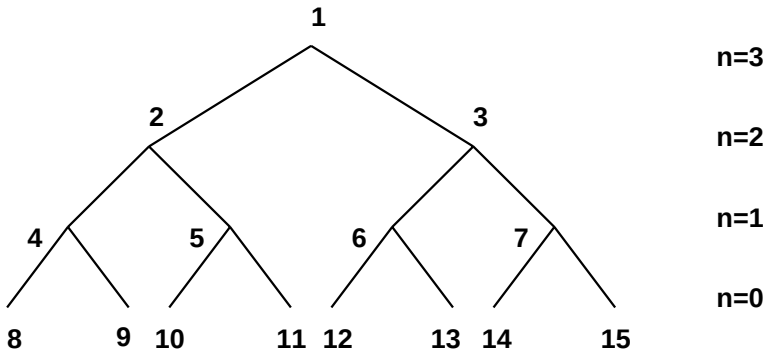


Fig. 3.2. Binary tree

3.1.1 MGAR model

To simplify the discussion we will assume that $\mathbf{Y}$ is indexed by the nodes in a binary tree, as shown in Figure 3.2. In order to define the model for $\mathbf{Y}$, we associate with the binary tree the ordering of the nodes shown in Figure 3.2. The nodes at level n are indexed from 2^{M-n} to $2^{M-n+1}-1$, where M is the number of levels in the tree.

The model which will be used for $\mathbf{Y}$ is a causal MGAR model, where the notion of causality for our model is defined by the ordering of the nodes of the tree defined above. In this model the random variables $Y_1, \ldots, Y_N$ are modeled as jointly Gaussian random variables conditioned on the classification of the tree nodes. The model can be described as follows: The value of the random variable Y_s can be predicted as a linear combination of the values of random variables at the current scale and at coarser scales. The prediction errors $\tilde{Y}_1, \ldots, \tilde{Y}_N$ form a sequence of independent random variables. The parameters used for the prediction of Y_s depend on the class to which node s belongs, i.e., x_s. The variance of the prediction error $\tilde{Y}_s$ also depends on x_s. The prediction errors can be written as

$$\tilde{Y}_s = Y_s - \mu_{x_s}^{(n_s)} + \sum_{r>0} a_{x_s,r}^{(n_s)} \left[Y_{s-r} - \mu_{x_s}^{(n_s)} \right] \tag{3.1}$$

where n_s is the level of node s in the tree, $\{a_{x_s,r}^{(n_s)}\}$ are the prediction coefficients at level n_s for class x_s, and $\tilde{Y}_1, \ldots, \tilde{Y}_N$ is a sequence of independent, Gaussian random

variables. The variance of $\tilde{Y}_s$ is given by

$$E[(\tilde{Y}_s - \mu_{x_s}^{(n_s)})^2] = (\sigma_{x_s}^{(n_s)})^2 \tag{3.2}$$

The MGAR model for class k at level n is said to be of order P if there are P values of r for which $a_{k,r}^{(n)}$ is non-zero.

Since the random variables $\tilde{Y}_1, \ldots, \tilde{Y}_N$ are independent Gaussian random variables, the form of their joint conditional probability density function given the classification of all tree nodes is known. However, we need the joint conditional probability density function of $Y_1, \ldots, Y_N$ given the classification of all tree nodes. By considering the sequences $Y_1, \ldots, Y_N$ and $\tilde{Y}_1, \ldots, \tilde{Y}_N$ as vectors, i.e., $\mathbf{Y} = [Y_1, \ldots, Y_N]^T$ and $\tilde{\mathbf{Y}} = [\tilde{Y}_1, \ldots, \tilde{Y}_N]^T$, $\tilde{\mathbf{Y}}$ can be written as

$$\tilde{\mathbf{Y}} = \mathbf{C} + \mathbf{A}\mathbf{Y} \tag{3.3}$$

where $\mathbf{C}$ is a constant matrix and $\mathbf{A}$ can be determined from Equation 3.1. Since $\tilde{\mathbf{Y}}_s$ is a linear combination of values of $\mathbf{Y}$ at node s and nodes which precede node s, but not at future nodes, the matrix $\mathbf{A}$ is lower triangular. Also, since the coefficient of the term Y_s in the expression for $\tilde{Y}_s$ is 1, all diagonal elements of $\mathbf{A}$ are 1. Thus the Jacobian of the transformation from $\mathbf{Y}$ to $\tilde{\mathbf{Y}}$ is 1, and the conditional probability density function of $\mathbf{Y}$ given $\mathbf{X}$ is [12]

$$f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = \prod_{s=1}^{N} \frac{1}{\sqrt{2\pi(\sigma_{x_s}^{(n_s)})^2}} \exp\left(-\frac{\tilde{y}_s^2}{2(\sigma_{x_s}^{(n_s)})^2}\right) \tag{3.4}$$

where

$$\tilde{y}_s = y_s - \mu_{x_s}^{(n_s)} + \sum_{r>0} a_{x_s,r}^{(n_s)} \left[y_{s-r} - \mu_{x_s}^{(n_s)}\right] \tag{3.5}$$

and the vector $\boldsymbol{\theta}$ contains the MGAR model parameters. The elements of $\boldsymbol{\theta}$ are $\mu_k^{(n)}$, $a_{k,r}^{(n)}$, and $(\sigma_k^{(n)})^2$, for $k = 1, \ldots, L$; $n = 0, \ldots, M-1$; and $r = 1, \ldots, P$. Since the filter coefficients used to obtain the Gaussian pyramid representation of the observed image sum to 1, the class means do not depend on the resolution level n, so that $\mu_k^{(n)} = \mu_k^{(m)} \equiv \mu_k$ for all n, m.

Fig. 3.3. Examples of cliques of type 1. Pairs of nodes marked with dots form cliques.

3.1.2 Model for class label pyramid

A MMRF model is used for the class label pyramid $\mathbf{X}$ [14]. This model is a natural extension of the single-resolution MRF model. In fact, the form of the probability mass function for a MMRF model is the same as that given for a MRF model in Chapter 2, i.e.,

$$p_{\mathbf{X}}(\mathbf{x}) = \frac{1}{z} \exp\left(-\frac{1}{T} \sum_{C \in \mathcal{C}} V_C(\mathbf{x})\right) \tag{3.6}$$

where $\mathbf{X}$ is now a multiresolution process, and the collection of cliques $\mathcal{C}$ is defined on the multiresolution lattice S.

For the multiscale MRF model used in this chapter, $\mathcal{C}$ consists of all pairs of nodes at the same resolution which are spatially horizontally or vertically adjacent to each other (type-1 cliques) and all pairs of nodes having a parent-child relationship to each other (type-2 cliques), plus all single nodes in S. Type-1 and Type-2 cliques are illustrated in Figures 3.3 and 3.4. The neighborhood system corresponding to this collection of cliques is shown in Figure 3.5.

The probability mass function of $\mathbf{X}$ can now be defined using the collection of cliques described above. Let $\mathcal{C}_1 \subseteq \mathcal{C}$ be the set of all cliques of type 1 and $\mathcal{C}_2 \subseteq \mathcal{C}$ the set of all cliques of type 2. Then the probability mass function of $\mathbf{X}$ is defined to be

$$p_{\mathbf{X}}(\mathbf{x}) = \frac{1}{z} \exp\left(- \sum_{\{r,s\} \in \mathcal{C}_1} \beta_1^{(n_r)} t(x_r, x_s) - \sum_{\{r,s\} \in \mathcal{C}_2} \beta_2^{(n_r)} t(x_r, x_s) - \sum_{\{r\} \in \mathcal{C}} \gamma_{x_r}\right) \tag{3.7}$$

where $\beta_1^{(n)}$ and $\beta_2^{(n)}$ are spatial interaction parameters at level n, node r is in level n_r, node r is assumed without loss of generality to be the parent node for each pair

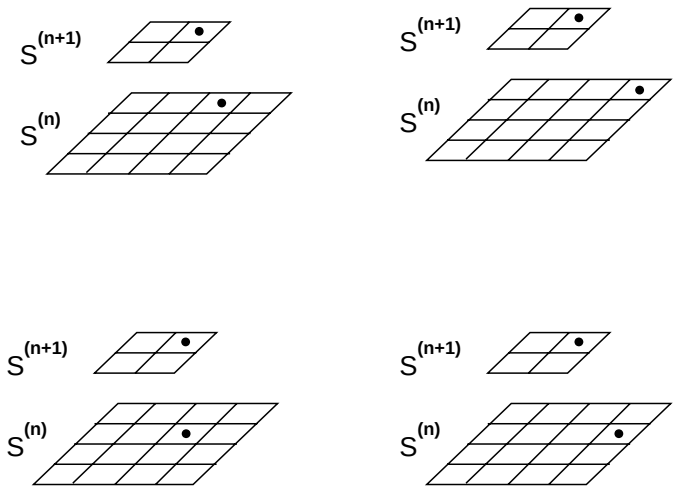


Fig. 3.4. Examples of cliques of type 2. Pairs of nodes marked with dots form cliques.

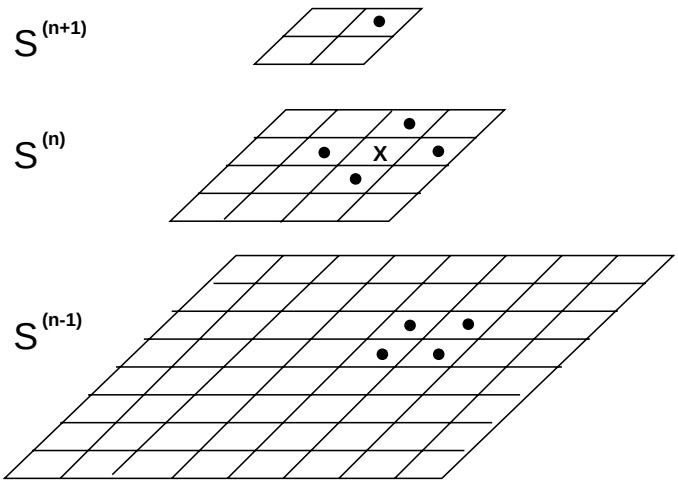


Fig. 3.5. Neighborhood system for multiscale MRF. Nodes marked with dots are neighbors of node marked with X.

of nodes in the summation representing type-2 cliques, γ_k is the cost for class k, and

$$t(x_r, x_s) = \begin{cases} 0 & \text{if } x_r = x_s \\ 1 & \text{if } x_r \neq x_s \end{cases} \tag{3.8}$$

To segment the observed image, the form for the conditional probability mass function of $\mathbf{X}$ given $\mathbf{Y}$ is needed. Using Bayes' rule and Equations 3.4 and 3.7, we have

$$p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}) = \frac{f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) p_{\mathbf{X}}(\mathbf{x})}{f_{\mathbf{Y}}(\mathbf{y}|\boldsymbol{\theta})} =$$

$$\frac{1}{f_{\mathbf{Y}}(\mathbf{y}|\boldsymbol{\theta})} \left[\prod_{s=1}^{N} \frac{1}{\sqrt{2\pi(\sigma_{x_s}^{(n_s)})^2}} \exp\left(-\frac{\tilde{y}_s^2}{2(\sigma_{x_s}^{(n_s)})^2} \right) \right] \cdot$$

$$\left(\frac{1}{z}\right) \exp\left(- \sum_{\{r,s\} \in \mathcal{C}_1} \beta_1^{(n_r)} t(x_r, x_s) - \sum_{\{r,s\} \in \mathcal{C}_2} \beta_2^{(n_r)} t(x_r, x_s) - \sum_{\{r\} \in \mathcal{C}} \gamma_{x_r} \right)$$

$$= \frac{1}{z f_{\mathbf{Y}}(\mathbf{y}|\boldsymbol{\theta})} \left[\prod_{s=1}^{N} \frac{1}{\sqrt{2\pi \sigma_{x_s}^{(n_s)})^2}} \right] \cdot$$

$$\exp\left(- \sum_{s=1}^{N} \frac{\tilde{y}_s^2}{2(\sigma_{x_s}^{(n_s)})^2} - \sum_{\{r,s\} \in \mathcal{C}_1} \beta_1^{(n_r)} t(x_r, x_s) - \sum_{\{r,s\} \in \mathcal{C}_2} \beta_2^{(n_r)} t(x_r, x_s) - \sum_{\{r\} \in \mathcal{C}} \gamma_{x_r} \right) \tag{3.9}$$

Since $f_{\mathbf{Y}}(\mathbf{y}|\boldsymbol{\theta})$ does not depend on $\mathbf{x}$, it will not need to be considered in the optimization. It should be noted that $p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$ is a Gibbs distribution.

3.2 Segmentation Algorithm

In this section we assume that $\boldsymbol{\theta}$ is known and describe the multiresolution segmentation algorithm. The MMPM algorithm follows the single-resolution MPM algorithm [3] very closely. The segmentation problem is formulated as an optimization problem. The optimization criterion which is used is the minimization of the expected value of the number of misclassified nodes in the multiresolution lattice.

Similarly to the single-resolution case, we can minimize this criterion by choosing for each node $s \in S$ the value of x_s from the set $\{1, 2, \ldots, L\}$ which minimizes $(1 - P(X_s = x_s|\mathbf{Y} = \mathbf{y}))$ or, equivalently, the value which maximizes $P(X_s = x_s|\mathbf{Y} = \mathbf{y})$.

The estimate of the label pyramid $\mathbf{X}$ which is derived by independently maximizing this marginal probability mass function for each $s \in S$ is the MMPM estimate of $\mathbf{X}$.

The same method described in Section 2.2 can be used here to approximate $P(X_s = k|\mathbf{Y} = \mathbf{y})$ for each $s \in S$ and $k \in \{1, 2, \ldots, L\}$. The Gibbs sampler is used to generate a Markov chain $\mathbf{X}(t)$ with limiting probability distribution $p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$ given by Equation 3.9. For the multiresolution case the state space of the Markov chain $\mathbf{X}(t)$ is the set of all of the L^N possible realizations of the label pyramid $\mathbf{X}$.

If T_s is the number of visits to node s made by the Gibbs sampler, then the estimates of $p_{X_s|\mathbf{Y}}(k|\mathbf{y}, \boldsymbol{\theta})$, which are maximized to obtain the MMPM estimate of $\mathbf{X}$, are provided by the approximations

$$p_{X_s|\mathbf{Y}}(k|\mathbf{y}, \boldsymbol{\theta}) \approx \frac{1}{T_s} \sum_{t=1}^{T_s} u_{k,s}(t) \qquad \forall k, s \tag{3.10}$$

where

$$u_{k,s}(t) = \begin{cases} 1 & \text{if } X_s(t) = k \\ 0 & \text{if } X_s(t) \neq k \end{cases} \tag{3.11}$$

3.3 Parameter Estimation

We will use the EM algorithm to estimate $\boldsymbol{\theta}$. In this section we describe the EM algorithm for the case when the values of the marginal conditional probability mass functions of the class label pyramid given the observed image pyramid are known.

The estimate of $\boldsymbol{\theta}$ at iteration p is the value $\boldsymbol{\theta}(p)$ which satisfies

$$Q(\boldsymbol{\theta}(p), \boldsymbol{\theta}(p-1)) \geq Q(\boldsymbol{\theta}, \boldsymbol{\theta}(p-1)) \, \forall \boldsymbol{\theta} \tag{3.12}$$

where

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}(p-1)) = E[\log f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})|\mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1)] \tag{3.13}$$

Substituting the expression for $f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$ into Equation 3.13 gives

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}(p-1)) = E[\log \prod_{s=1}^{N} \frac{1}{\sqrt{2\pi(\sigma_{x_s}^{(n_s)})^2}} \exp\left(-\frac{\tilde{y}_s^2}{2(\sigma_{x_s}^{(n_s)})^2}\right) |\mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1)]$$

$$= \sum_{s \in S} E[\log \left(\frac{1}{\sqrt{2\pi(\sigma_{x_s}^{(n_s)})^2}} \right) | \mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1)] - \frac{1}{2} \sum_{s \in S} E[\frac{\tilde{y}_s^2}{(\sigma_{x_s}^{(n_s)})^2} | \mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1)]$$

$$= \sum_{s \in S} \sum_{k=0}^{L-1} \log \left(\frac{1}{\sqrt{2\pi(\sigma_k^{(n_s)})^2}} \right) P(X_s = k | \mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1)) -$$

$$\frac{1}{2} \sum_{s \in S} \sum_{k=0}^{L-1} \frac{(\tilde{y}_s^k)^2}{(\sigma_k^{(n_s)})^2} P(X_s = k | \mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1)) \tag{3.14}$$

where

$$\tilde{y}_s^k = y_s - \mu_k + \sum_{r>0} a_{k,r}^{(n_s)} [y_{s-r} - \mu_k] \tag{3.15}$$

In the M-step at iteration p, $Q(\boldsymbol{\theta}, \boldsymbol{\theta}(p-1))$ must be maximized with respect to $\boldsymbol{\theta}$. Let $\hat{\boldsymbol{\theta}}$ be the value of $\boldsymbol{\theta}$ which maximizes $Q(\boldsymbol{\theta}, \boldsymbol{\theta}(p-1))$. By differentiating and setting to zero, it can be shown that $\hat{\boldsymbol{\theta}}$, with elements $\hat{\mu}_k$, $(\hat{\sigma}_k^{(n)})^2$, and $\hat{a}_{k,r}^{(n)}$, must satisfy the set of equations

$$\sum_{s \in S} \left[y_s - \hat{\mu}_k + \sum_{q>0} \hat{a}_{k,q}^{(n_s)} [y_{s-q} - \hat{\mu}_k] \right] P(X_s = k | \mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1)) = 0 \tag{3.16}$$

$$\sum_{s \in S^{(n)}} \left[y_s - \hat{\mu}_k + \sum_{q>0} \hat{a}_{k,q}^{(n_s)} [y_{s-q} - \hat{\mu}_k] \right] [y_{s-r} - \hat{\mu}_k] P(X_s = k | \mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1)) = 0 \tag{3.17}$$

$$\sum_{s \in S^{(n)}} \left[y_s - \hat{\mu}_k + \sum_{q>0} \hat{a}_{k,q}^{(n_s)} [y_{s-q} - \hat{\mu}_k] \right] P(X_s = k | \mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1)) -$$

$$(\hat{\sigma}_k^{(n)})^2 \sum_{s \in S^{(n)}} P(X_s = k | \mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1)) = 0 \tag{3.18}$$

for $k = 1, \ldots, L$; $n = 0, \ldots, M-1$; and $r = 1, \ldots, P$.

The value of $\boldsymbol{\theta}(p)$ can be obtained by solving the above system of equations for $\hat{\mu}_k$, $(\hat{\sigma}_k^{(n)})^2$, and $\hat{a}_{k,r}^{(n)}$ and setting $\boldsymbol{\theta}(p) = \hat{\boldsymbol{\theta}}$. However, Equation 3.17 is a nonlinear function of $\hat{\mu}_k$ and $\hat{a}_{k,r}^{(n)}$, which makes solving the system of equations difficult. To address this problem, we divide the M-step into two substeps: First the class means are estimated using the equations

$$\hat{\mu}_k = \frac{1}{N_k} \sum_{s=1}^{N} y_s P(X_s = k | \mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1)) \tag{3.19}$$

where

$$N_k = \sum_{s=1}^{N} P(X_s = k | \mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1)) \tag{3.20}$$

This equation for $\hat{\mu}_k$ is the equation which results if the random variables in the observed image pyramid are assumed to be conditionally independent given the class label pyramid. After the means have been estimated using Equation 3.19, the remaining MGAR model parameters can be estimated easily using Equations 3.17 and 3.18.

3.4 Multiresolution EM/MPM Algorithm

The multiresolution EM/MPM algorithm follows the same steps as the EM/MPM algorithm described in Section 2.4. The Markov chain generated during stage p of the algorithm is denoted $\mathbf{X}(p,t)$, and the estimate of $\boldsymbol{\theta}$ obtained during stage p of the algorithm is denoted by the random variable $\boldsymbol{\Theta}(p)$. For the multiresolution case the state space of the Markov chain $\mathbf{X}(p,t)$ is the set of all of the L^N possible realizations of the label pyramid $\mathbf{X}$ and the elements of $\boldsymbol{\Theta}(p)$ are the random variables $M_k(p), A_{k,r}^{(n)}(p), S_k^{(n)}(p)$ for $k = 1, \ldots L$; $n = 0, \ldots M-1$; and $r = 1, \ldots P$.

The Markov chain $\mathbf{X}(p,t)$ is generated using the Gibbs sampler and sampling from the distribution $p_{\mathbf{X}|\mathbf{Y},\boldsymbol{\Theta}(p-1)}(\mathbf{x}|\mathbf{y}, \hat{\boldsymbol{\theta}}(p-1))$ given by Equation 3.9 with $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}(p-1)$. The estimates

$$v_{k,s}(p,t) = \frac{1}{t} \sum_{i=1}^{t} u_{k,s}(p,i) \tag{3.21}$$

are used to obtain $\boldsymbol{\Theta}(p)$, using the equations

$$M_k(p) = \frac{\sum_{s=1}^{N} y_s v_{k,s}(p,T_p)}{\sum_{s=1}^{N} v_{k,s}(p,T_p)} \tag{3.22}$$

$$\sum_{s \in S^{(n)}} \left[y_s - M_k + \sum_{q>0} A_{k,q}^{(n_s)} [y_{s-q} - M_k] \right] [y_{s-r} - M_k] v_{k,s}(p,T_p) = 0 \tag{3.23}$$

$$\sum_{s \in S^{(n)}} \left[y_s - M_k + \sum_{q>0} A_{k,q}^{(n_s)} [y_{s-q} - M_k] \right] v_{k,s}(p,T_p) -$$

$$(S_k^{(n)})^2 \sum_{s \in S^{(n)}} v_{k,s}(p, T_p) = 0 \tag{3.24}$$

These equations are of the same form as Equations 3.19, 3.17, and 3.18, with $P(X_s = k|\mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1))$ replaced by $v_{k,s}(p, T_p)$.

The final estimate of $\boldsymbol{\theta}$ is $\boldsymbol{\Theta}(P)$. The final segmentation is obtained by maximizing over all k the value $v_{k,s}(P+1, T_{P+1})$ for every $s \in S$, for some $T_{P+1} \geq 1$.

3.5 Experimental Results

The multiresolution EM/MPM algorithm was applied to the same set of test images as the EM/MPM algorithm in Chapter 2. For the results presented in this section, the Gaussian pyramid decomposition described in [30], with the same filter weights as those used in [30], was used to obtain the multiresolution representation of the observed image, although the algorithm could also be used with other multiresolution decomposition schemes, e.g., averaging over 2x2 blocks and subsampling to obtain coarse-resolution data from fine-resolution data.

For all results presented in this section, three pyramid levels were used (i.e., $M = 3$). This means that the original image was examined at three different resolutions. It was found that for some images the use of more than three pyramid levels resulted in an insufficient number of pixels at the coarsest resolution for the algorithm to perform well.

Unless otherwise noted, the values $\beta_1^{(n)} = 2.4\ \forall n$, $\beta_2^{(0)} = \beta_2^{(1)} = 0.3$, and $T_p = 3$ for every $p = 1 \ldots, P+1$, were used. For all experiments described here, 70 stages of the single-resolution EM/MPM algorithm were first performed at the coarsest level of the pyramid, and the resulting segmentation propagated to other levels, to obtain an initial estimate of the label pyramid $\mathbf{X}$. Also, the estimates of the conditional probability mass functions of the class labels at the coarsest resolution, obtained using the EM/MPM algorithm at the coarsest resolution, were propagated to higher resolutions. These estimates were then used in Equations 3.19, 3.17 and 3.18 to obtain initial estimates of the MGAR model parameters. The MGAR prediction

Table 3.1 Percentage of Misclassified Pixels

Image	EM/MPM	Multiresolution EM/MPM
Figure 3.6(a)	2.4	3.1
Figure 3.7(a)	4.8	0.74
Figure 3.8(a)	39.2	12.3
Figure 3.9(a)	30.0	1.2

window used at node $(i,j) \in S^{(n)}$ contains nodes $(i-1,j-1) \in S^{(n)}, (i-1,j) \in S^{(n)}, (i,j-1) \in S^{(n)}, (i/2,j/2) \in S^{(n+1)}, (i/2,j/2+1) \in S^{(n+1)}, (i/2+1,j/2) \in S^{(n+1)}, (i/2+1,j/2+1) \in S^{(n+1)}$ if n is not the coarsest resolution, and nodes $(i-1,j-1) \in S^{(n)}, (i-1,j) \in S^{(n)}, (i-1,j+1) \in S^{(n)}, (i,j-1) \in S^{(n)}$ if n is the coarsest resolution. Unless otherwise noted, the class cost parameters were assumed to be zero.

Figures 3.6 through 3.9 show results from applying the multiresolution EM/MPM algorithm to the synthetic textured images, as well as the results from Chapter 2 for the EM/MPM algorithm for comparison. In each of these figures, the original image is shown in (a), the result obtained using the EM/MPM algorithm is shown in (b), and the result obtained from applying 20 stages of the multiresolution EM/MPM algorithm is shown in (c).

For the images shown in Figures 3.7, 3.8, and 3.9, the multiresolution EM/MPM algorithm performed significantly better than the EM/MPM algorithm. For the image shown in Figure 3.6, the two algorithms performed similarly.

Table 3.1 shows the percentage of pixels which were misclassified by the EM/MPM and multiresolution EM/MPM algorithms for the four synthetic test images. In general, the multiresolution algorithm provides much better performance in terms of the number of misclassified pixels, although for the image shown in Figure 3.6, the single-resolution algorithm provides slightly better results.

The segmentation obtained from applying 20 stages of the multiresolution EM/MPM algorithm with 2 classes to the natural scene in Figure 3.10(a) is shown in Figure 3.10(c), and the result from 20 stages of the multiresolution algorithm with 3 classes is shown in Figure 3.10(e). In general, the multiresolution algorithm provides smoother regions and models large-scale properties of the image better. The EM/MPM algorithm distinguishes individual buildings from their immediate surroundings, whereas the multiresolution EM/MPM algorithm distinguishes groups of buildings from larger ground regions.

Figures 3.11 through 3.26 show the results from applying 20 stages of the multiresolution EM/MPM algorithm to the infrared test images. In each of these figures the image shown in (b) is the segmentation resulting from the EM/MPM algorithm with $L = 2$, the image shown in (c) is the result obtained using the multiresolution EM/MPM algorithm with $L = 2$, (d) shows the results from EM/MPM with $L = 3$, and (e) shows the result from multiresolution EM/MPM with $L = 3$. In some cases, the multiresolution EM/MPM algorithm did not find 3 statistically distinct classes. The multiresolution algorithm performs better than the single-resolution algorithm for the images in Figures 3.12, 3.15, 3.16 (for $L = 3$), and 3.20, whereas the single-resolution algorithm does a better job of distinguishing the road and the vehicle from the background in Figure 3.14.

For the mammography images, the values $\beta_1^{(n)} = 2.4 \ \forall n$, $\beta_2^{(0)} = 2.4$, and $\beta_2^{(1)} = 1.0$ were used. As in the single-resolution case, the values used for the class cost parameters were $\gamma_1 = \gamma_2 = 2.3$ and $\gamma_3 = 5$.

The result from applying 10 stages of the multiresolution EM/MPM algorithm to the first mammography image is shown in Figure 3.27(d), and the result from applying 10 stages of the multiresolution algorithm to the second mammography image is shown in Figure 3.28(d). The cost parameters used were optimized experimentally for the single-resolution algorithm, and thus the segmented images obtained using the single-resolution algorithm are slightly better.

The result from applying 10 stages of the multiresolution EM/MPM algorithm to the synthetic mammography image is shown in Figure 3.29(c). For this image, the multiresolution approach performs better than the single-resolution approach, since it does not follow the small-scale variations in the texture of the normal tissue.

The final issue that we discuss in this section is a comparison of the amount of computation required to achieve convergence for two of the test images using the EM/MPM, deterministic EM/MPM, and multiresolution EM/MPM algorithms. We consider the convergence of the parameter estimates to do this, although the amount of computation could also be measured by studying convergence of the MPM algorithm at each stage of the EM/MPM and multiresolution EM/MPM algorithms. For the deterministic algorithm, this is not an issue, because for each iteration of the deterministic algorithm, only one iteration through the image is required to estimate the class label probabilities.

We use the number of visits per pixel as a measure of computational complexity because the number of operations required at each pixel visit are comparable for the three algortihms under consideration. Table 3.2 shows the number of visits to each pixel required for convergence of the parameter estimates obtained by the three algorithms for the two test images in Figures 3.6(a) and 3.7(a). It can be seen that the deterministic EM/MPM algorithm converges more quickly than the EM/MPM algorithm for second image, but for the first image the EM/MPM algorithm converges more quickly. This is a very interesting result, since the deterministic algorithm is supposed to have the advantage of faster convergence. The result is not completely surprising, however, because the EM algorithm is known to be slow to converge in some cases, and if the EM/MPM algorithm does a better job of estimating the class label probabilities which are needed to compute the EM updates, then it would be reasonable for it to converge more quickly.

The multiresolution EM/MPM algorithm converges more quickly than both single-resolution algorithms for both images. This is also not a surprising result, since much

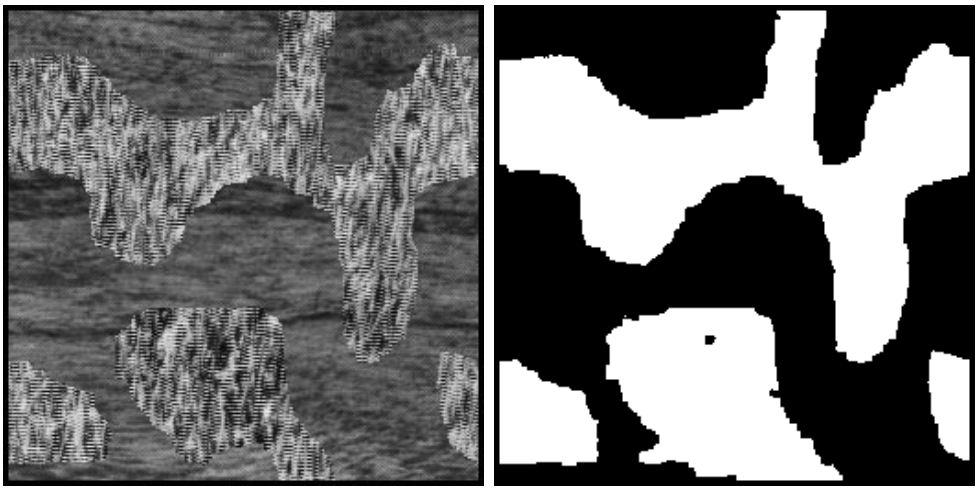
Table 3.2 Visits Per Pixel

Image	EM/MPM	Deterministic EM/MPM	Multiresolution EM/MPM
Figure 3.6(a)	1250	2160	550
Figure 3.7(a)	3000	1050	300

of the segmentation and parameter estimation can be done at the coarsest resolution, where there are fewer pixels to process.

3.6 Conclusion

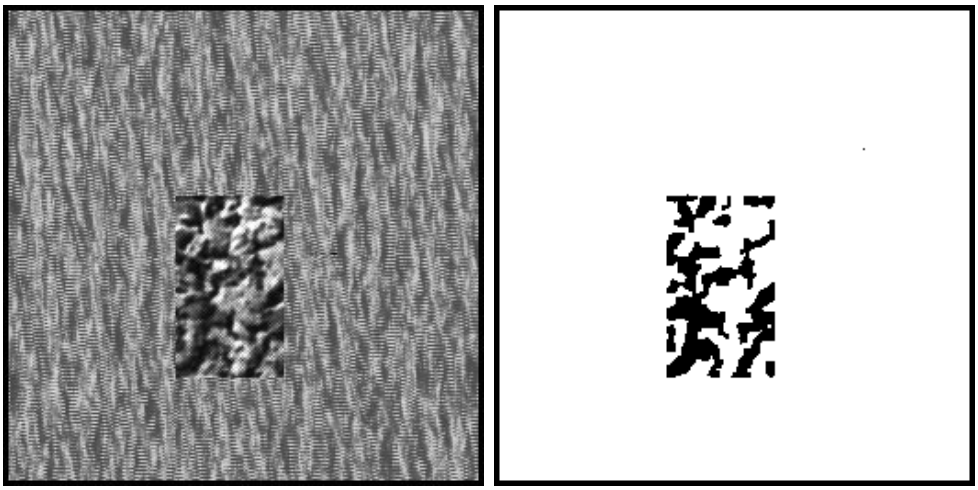
We have presented a multiresolution extension of the EM/MPM algorithm. The multiresolution EM/MPM algorithm effectively uses larger neighborhoods to perform the segmentation than the single-resolution algorithm, giving the multiresolution approach the ability to model large-scale properties of textures more effectively. The multiresolution approach performed significantly better than the single-resolution approach for the synthetic textured images tested.



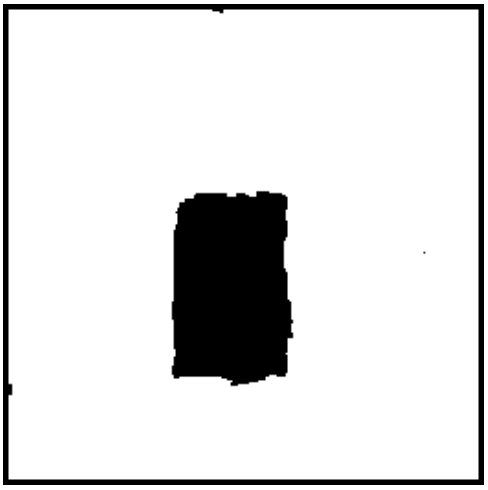
(a) (b)

(c)

Fig. 3.6. (a): Original image. (b): Segmented image obtained using EM/MPM algorithm. (c): Segmented image obtained using multiresolution EM/MPM algorithm.

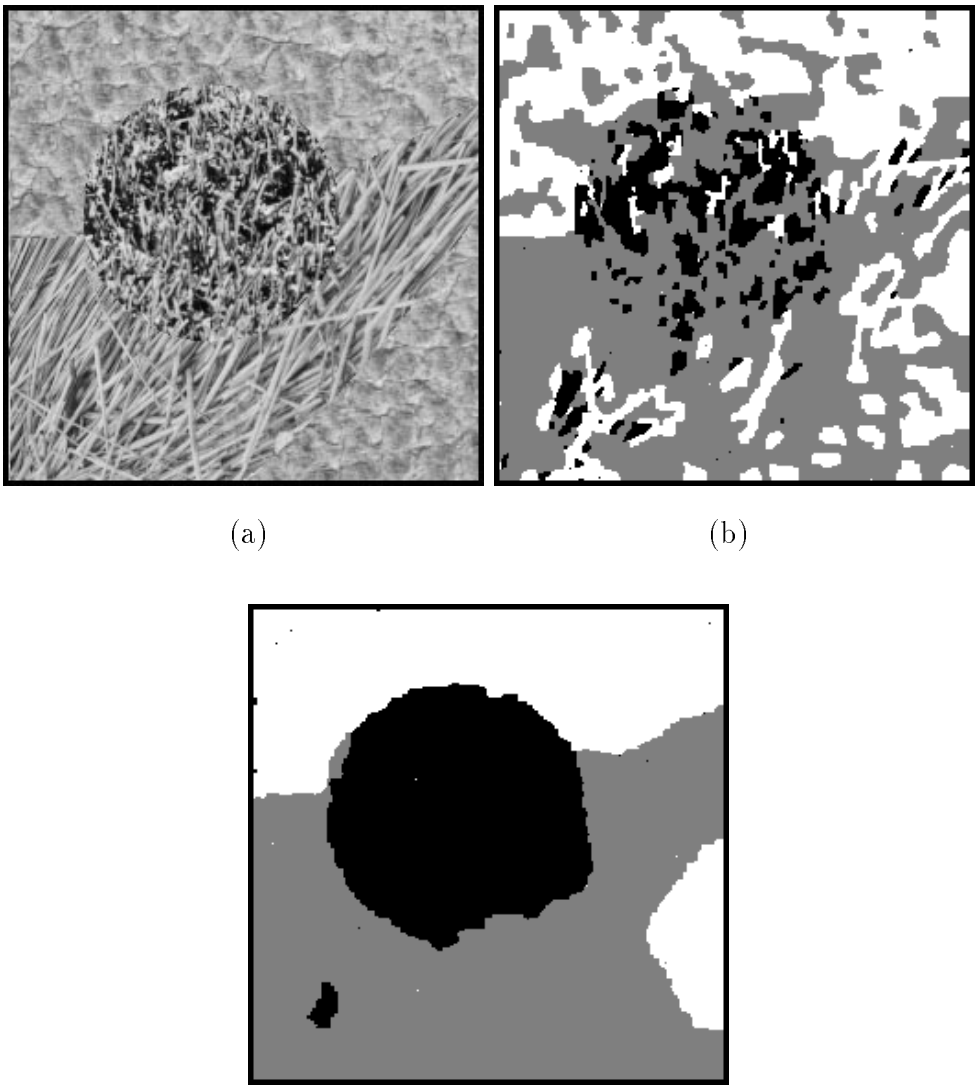


(a) (b)



(c)

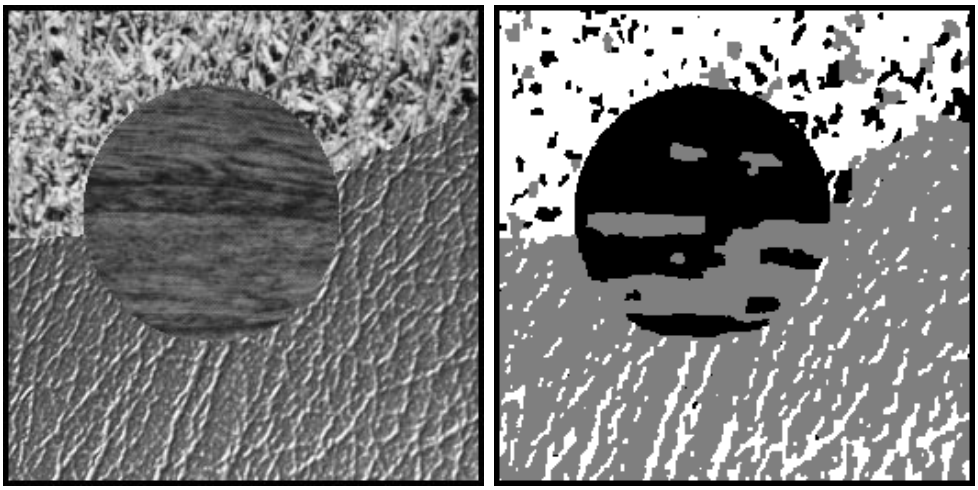
Fig. 3.7. (a): Original image. (b): Segmented image obtained using EM/MPM algorithm. (c): Segmented image obtained using multiresolution EM/MPM algorithm.



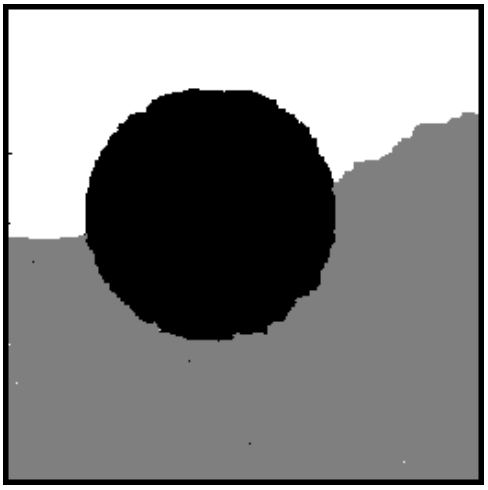
(a) (b)

(c)

Fig. 3.8. (a): Original image. (b): Segmented image obtained using EM/MPM algorithm. (c): Segmented image obtained using multiresolution EM/MPM algorithm.



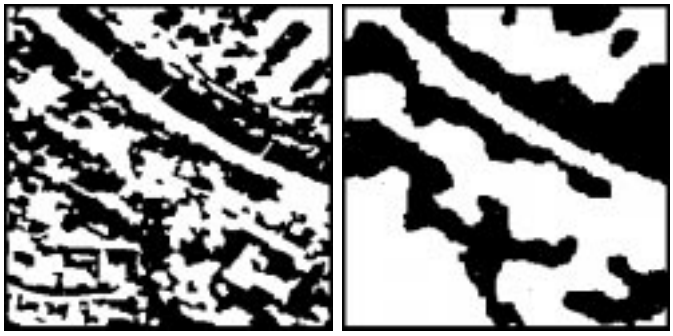
(a) (b)



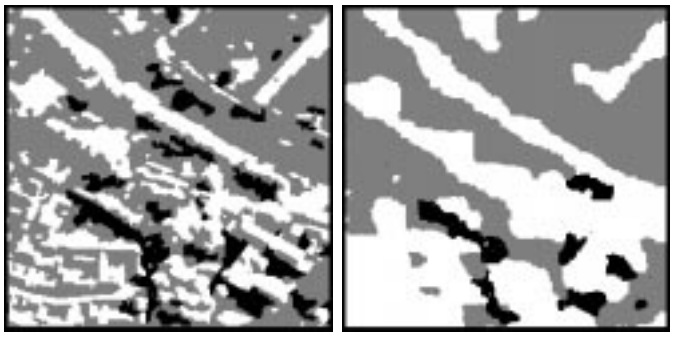
(c)

Fig. 3.9. (a): Original image. (b): Segmented image obtained using EM/MPM algorithm. (c): Segmented image obtained using multiresolution EM/MPM algorithm.

(a)

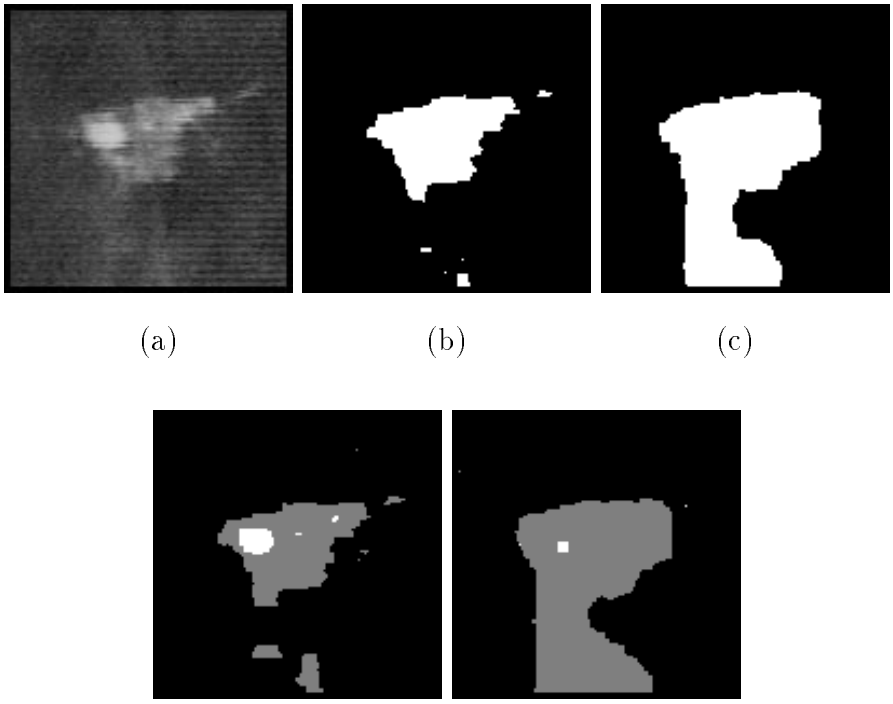


(b) (c)



(d) (e)

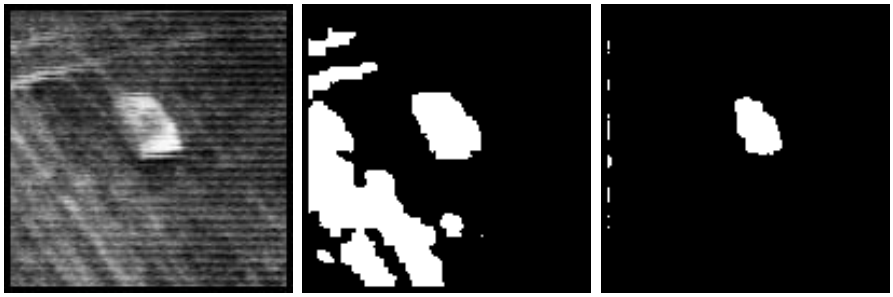
Fig. 3.10. (a): Original image. (b): Segmented image obtained using EM/MPM algorithm with 2 classes. (c): Segmented image obtained using multiresolution EM/MPM algorithm with 2 classes. (d): Segmented image obtained using EM/MPM algorithm with 3 classes. (e): Segmented image obtained using multiresolution EM/MPM algorithm with 3 classes.



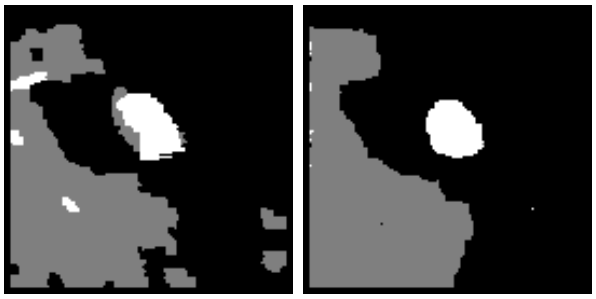
(a) (b) (c)

(d) (e)

Fig. 3.11. (a): Original image. (b): Segmented image obtained using EM/MPM algorithm with 2 classes. (c): Segmented image obtained using multiresolution EM/MPM algorithm with 2 classes. (d): Segmented image obtained using EM/MPM algorithm with 3 classes. (e): Segmented image obtained using multiresolution EM/MPM algorithm with 3 classes.



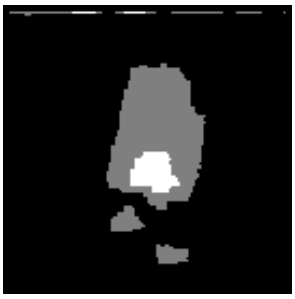
(a) (b) (c)



(d) (e)

Fig. 3.12. (a): Original image. (b): Segmented image obtained using EM/MPM algorithm with 2 classes. (c): Segmented image obtained using multiresolution EM/MPM algorithm with 2 classes. (d): Segmented image obtained using EM/MPM algorithm with 3 classes. (e): Segmented image obtained using multiresolution EM/MPM algorithm with 3 classes.

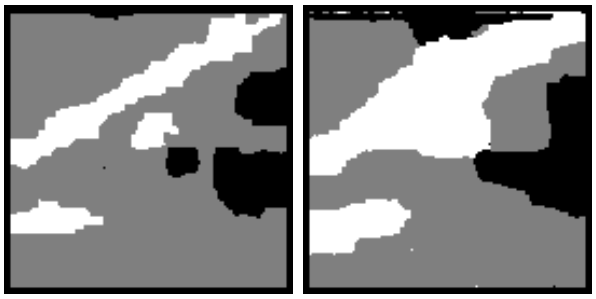
(a) (b) (c)



(d)

Fig. 3.13. (a): Original image. (b): Segmented image obtained using EM/MPM algorithm with 2 classes. (c): Segmented image obtained using multiresolution EM/MPM algorithm with 2 classes. (d): Segmented image obtained using EM/MPM algorithm with 3 classes.

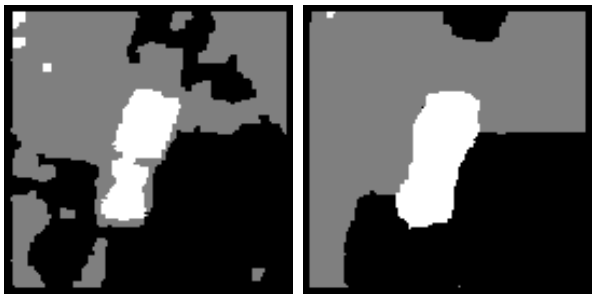
(a) (b) (c)



(d) (e)

Fig. 3.14. (a): Original image. (b): Segmented image obtained using EM/MPM algorithm with 2 classes. (c): Segmented image obtained using multiresolution EM/MPM algorithm with 2 classes. (d): Segmented image obtained using EM/MPM algorithm with 3 classes. (e): Segmented image obtained using multiresolution EM/MPM algorithm with 3 classes.

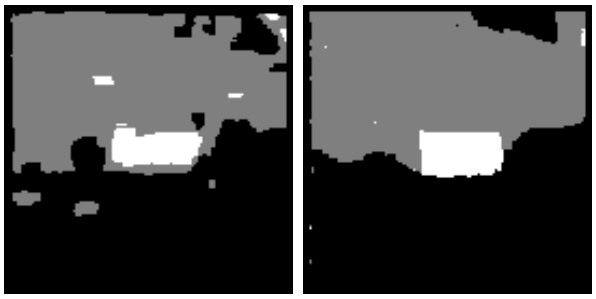
(a) (b) (c)



(d) (e)

Fig. 3.15. (a): Original image. (b): Segmented image obtained using EM/MPM algorithm with 2 classes. (c): Segmented image obtained using multiresolution EM/MPM algorithm with 2 classes. (d): Segmented image obtained using EM/MPM algorithm with 3 classes. (e): Segmented image obtained using multiresolution EM/MPM algorithm with 3 classes.

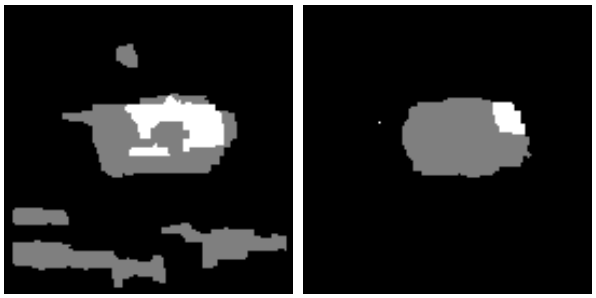
(a) (b) (c)



(d) (e)

Fig. 3.16. (a): Original image. (b): Segmented image obtained using EM/MPM algorithm with 2 classes. (c): Segmented image obtained using multiresolution EM/MPM algorithm with 2 classes. (d): Segmented image obtained using EM/MPM algorithm with 3 classes. (e): Segmented image obtained using multiresolution EM/MPM algorithm with 3 classes.

(a) (b) (c)



(d) (e)

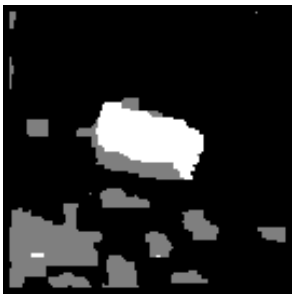
Fig. 3.17. (a): Original image. (b): Segmented image obtained using EM/MPM algorithm with 2 classes. (c): Segmented image obtained using multiresolution EM/MPM algorithm with 2 classes. (d): Segmented image obtained using EM/MPM algorithm with 3 classes. (e): Segmented image obtained using multiresolution EM/MPM algorithm with 3 classes.

(a) (b) (c)

(d)

Fig. 3.18. (a): Original image. (b): Segmented image obtained using EM/MPM algorithm with 2 classes. (c): Segmented image obtained using multiresolution EM/MPM algorithm with 2 classes. (d): Segmented image obtained using EM/MPM algorithm with 3 classes.

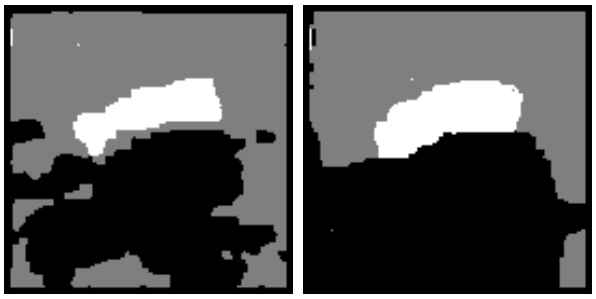
(a) (b) (c)



(d)

Fig. 3.19. (a): Original image. (b): Segmented image obtained using EM/MPM algorithm with 2 classes. (c): Segmented image obtained using multiresolution EM/MPM algorithm with 2 classes. (d): Segmented image obtained using EM/MPM algorithm with 3 classes.

(a) (b) (c)



(d) (e)

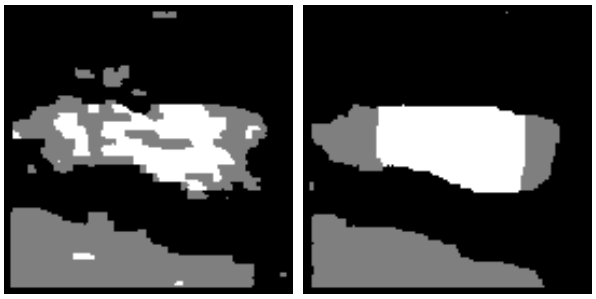
Fig. 3.20. (a): Original image. (b): Segmented image obtained using EM/MPM algorithm with 2 classes. (c): Segmented image obtained using multiresolution EM/MPM algorithm with 2 classes. (d): Segmented image obtained using EM/MPM algorithm with 3 classes. (e): Segmented image obtained using multiresolution EM/MPM algorithm with 3 classes.

(a) (b) (c)

(d) (e)

Fig. 3.21. (a): Original image. (b): Segmented image obtained using EM/MPM algorithm with 2 classes. (c): Segmented image obtained using multiresolution EM/MPM algorithm with 2 classes. (d): Segmented image obtained using EM/MPM algorithm with 3 classes. (e): Segmented image obtained using multiresolution EM/MPM algorithm with 3 classes.

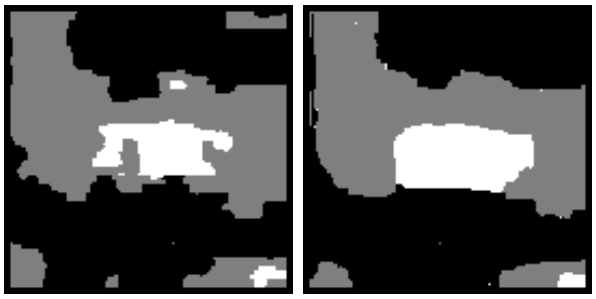
(a) (b) (c)



(d) (e)

Fig. 3.22. (a): Original image. (b): Segmented image obtained using EM/MPM algorithm with 2 classes. (c): Segmented image obtained using multiresolution EM/MPM algorithm with 2 classes. (d): Segmented image obtained using EM/MPM algorithm with 3 classes. (e): Segmented image obtained using multiresolution EM/MPM algorithm with 3 classes.

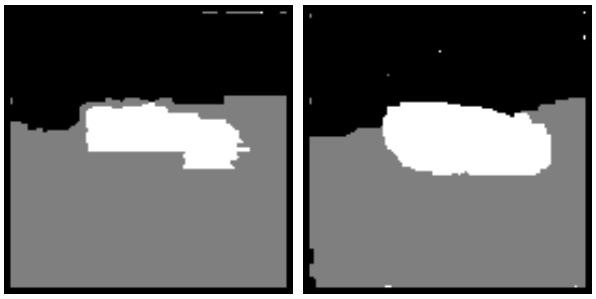
(a) (b) (c)



(d) (e)

Fig. 3.23. (a): Original image. (b): Segmented image obtained using EM/MPM algorithm with 2 classes. (c): Segmented image obtained using multiresolution EM/MPM algorithm with 2 classes. (d): Segmented image obtained using EM/MPM algorithm with 3 classes. (e): Segmented image obtained using multiresolution EM/MPM algorithm with 3 classes.

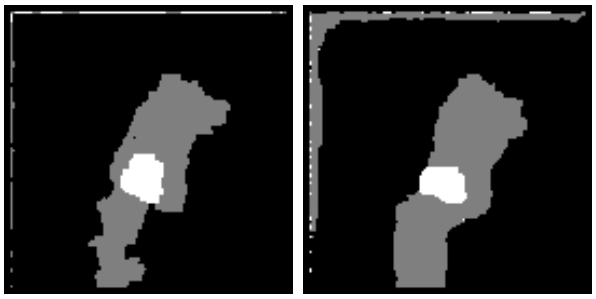
(a) (b) (c)



(d) (e)

Fig. 3.24. (a): Original image. (b): Segmented image obtained using EM/MPM algorithm with 2 classes. (c): Segmented image obtained using multiresolution EM/MPM algorithm with 2 classes. (d): Segmented image obtained using EM/MPM algorithm with 3 classes. (e): Segmented image obtained using multiresolution EM/MPM algorithm with 3 classes.

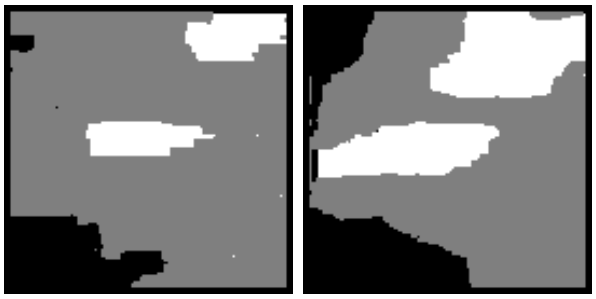
(a) (b) (c)



(d) (e)

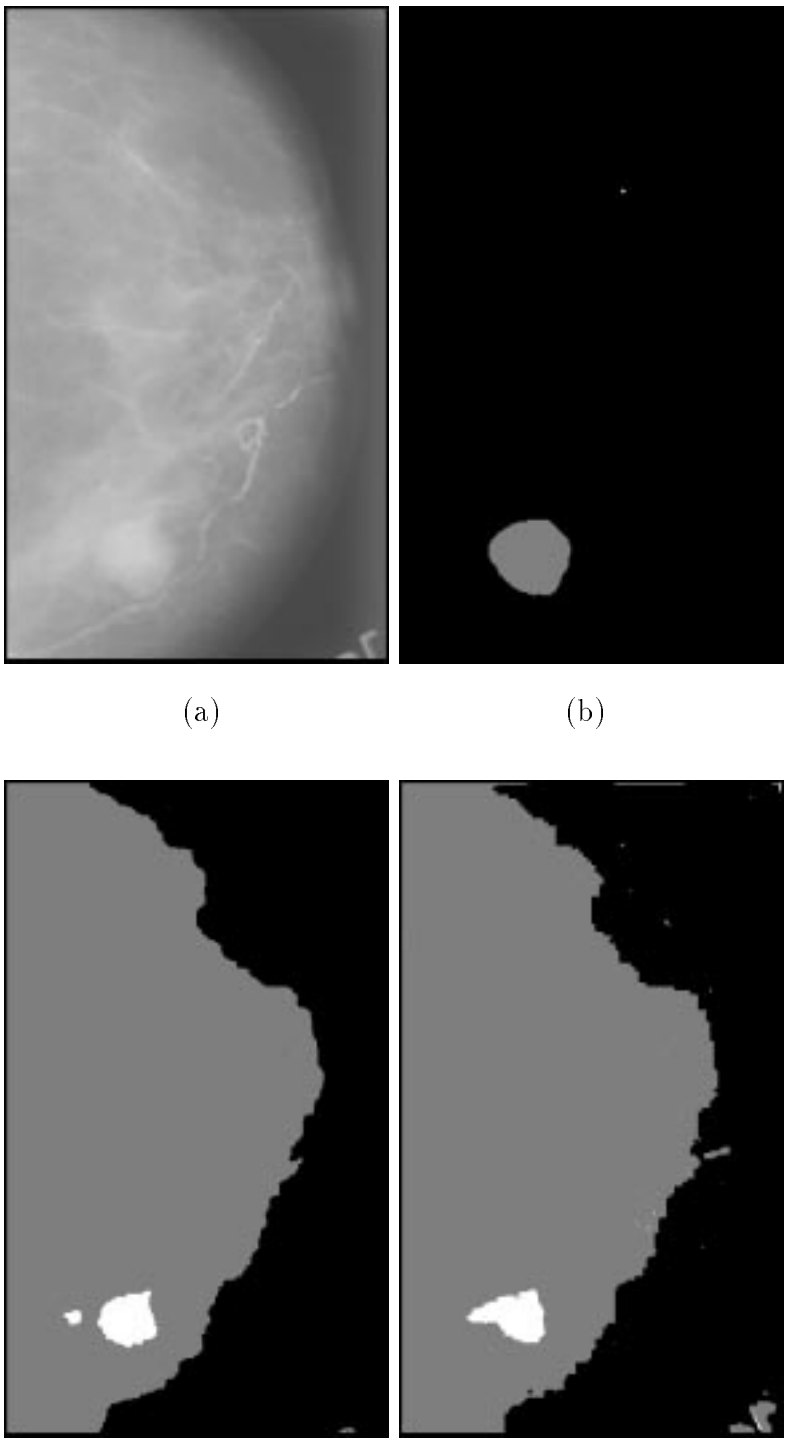
Fig. 3.25. (a): Original image. (b): Segmented image obtained using EM/MPM algorithm with 2 classes. (c): Segmented image obtained using multiresolution EM/MPM algorithm with 2 classes. (d): Segmented image obtained using EM/MPM algorithm with 3 classes. (e): Segmented image obtained using multiresolution EM/MPM algorithm with 3 classes.

(a) (b) (c)



(d) (e)

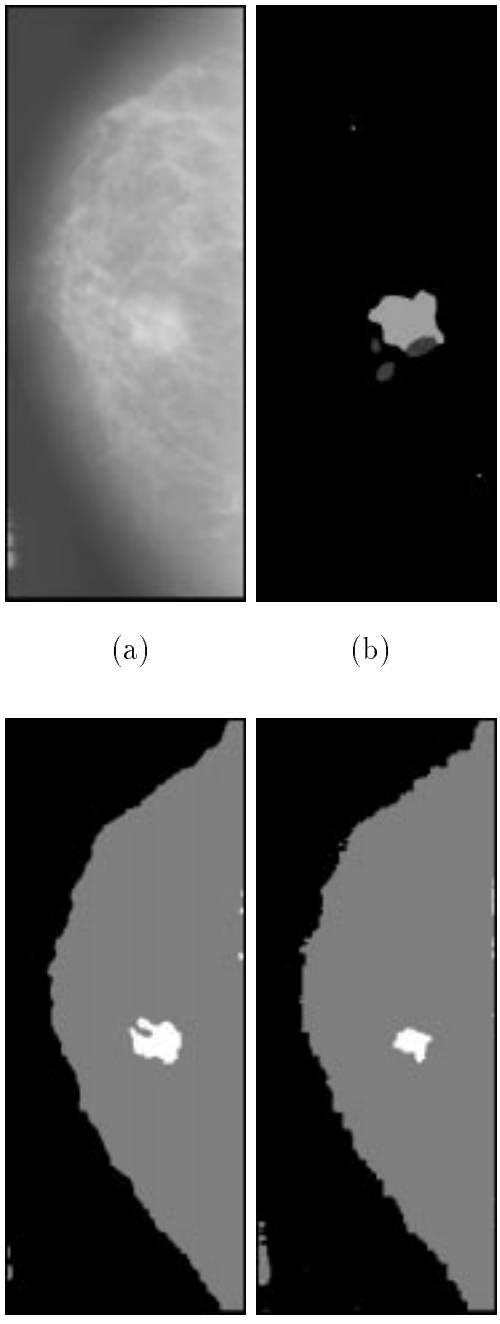
Fig. 3.26. (a): Original image. (b): Segmented image obtained using EM/MPM algorithm with 2 classes. (c): Segmented image obtained using multiresolution EM/MPM algorithm with 2 classes. (d): Segmented image obtained using EM/MPM algorithm with 3 classes. (e): Segmented image obtained using multiresolution EM/MPM algorithm with 3 classes.



(a) (b)

(c) (d)

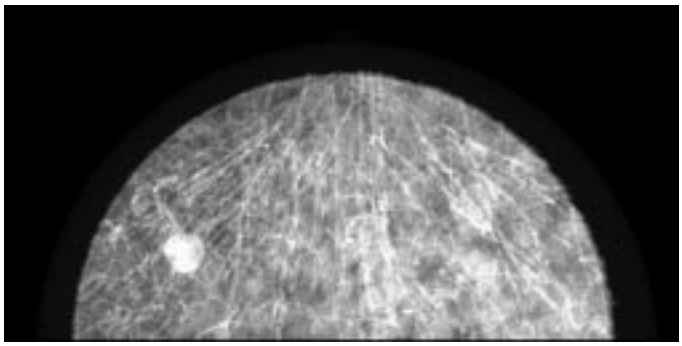
Fig. 3.27. (a): Original image. (b): Truth image. (c): Segmented image obtained using EM/MPM algorithm. (d): Segmented image obtained using multiresolution EM/MPM algorithm.



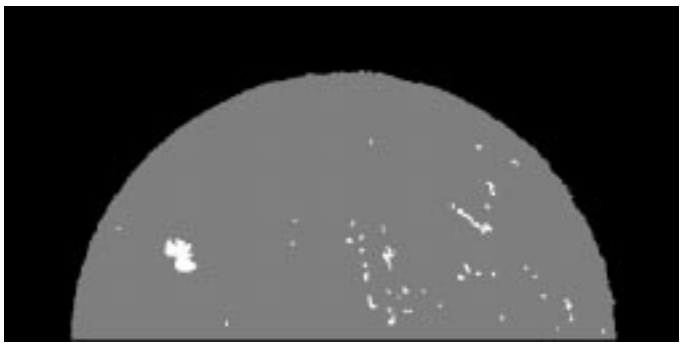
(a) (b)

(c) (d)

Fig. 3.28. (a): Original image. (b): Truth image. (c): Segmented image obtained using EM/MPM algorithm. (d): Segmented image obtained using multiresolution EM/MPM algorithm.



(a)



(b)

(c)

Fig. 3.29. (a): Original image. (b): Segmented image obtained using EM/MPM algorithm. (c): Segmented image obtained using multiresolution EM/MPM algorithm.

4. ANALYSIS OF THE EM/MPM PROCEDURE

In Chapter 2 we presented experimental results comparing the performance of the EM/MPM algorithm and the deterministic EM/MPM algorithm. The results suggested that the EM/MPM algorithm provides better performance in terms of minimizing the expected number of misclassified pixels. In this chapter we present analytical results relative to the EM/MPM algorithm which provide an explanation for its superior performance. The analysis is general enough to hold for both the single-resolution and multiresolution EM/MPM algorithms. We show two important results. First, we show that the estimates of the marginal probabilities of the class labels obtained during a given stage of the EM/MPM procedure converge with probability 1 to the true values of the class label probabilities, given the estimates of the model parameters obtained during the previous stage. Second, we show that the parameter estimates resulting from the EM/MPM procedure can be made arbitrarily close to the EM estimates of the parameters with probability 1, if a sufficient number of iterations is performed. These two results together imply that the final estimates of the marginal probabilities of the class labels obtained using the EM/MPM procedure will be close to the true values of the marginal probabilities of the class labels, to the extent that the EM estimates of the model parameters are close to the true values of the model parameters. Thus, the algorithm will eventually converge to the "correct" segmentation, as long as the EM estimates of the model parameters are close to the true values of the model parameters. The term "correct" is, of course, relative to the stochastic models used.

The analysis is presented for a more general case than the formulations of Chapters 2 and 3. For the purpose of analysis $\mathbf{X}$ is a collection of random variables which have

a Gibbs distribution, and the EM equations form a sequence $\{\boldsymbol{\theta}(p), p \geq 1\}$, where, for each p, $\boldsymbol{\theta}(p)$ is a function of $p_{X_s|\mathbf{Y}}(k|\mathbf{y}, \boldsymbol{\theta}(p-1))$. For example, $\mathbf{X}$ might be a two-dimensional MRF or a multiscale MRF, and $\boldsymbol{\theta}(p)$ might be given by Equations 2.23 through 2.25 or by Equations 3.16 through 3.18. The general EM/MPM procedure studied in this chapter consists of the following steps at stage p:

1. Generate a Markov chain $\mathbf{X}(p,t)$ with limiting distribution $p_{\mathbf{X}|\mathbf{Y},\boldsymbol{\Theta}(p-1)}(\mathbf{x}|\mathbf{y}, \hat{\boldsymbol{\theta}}(p-1))$, where $\boldsymbol{\Theta}(p-1)$ is the estimate of $\boldsymbol{\theta}$ obtained in the previous stage.

2. Approximate the class label probabilities using the equations

$$p_{X_s|\mathbf{Y}}(k|\mathbf{y}, \boldsymbol{\theta}) \approx \frac{1}{T_p} \sum_{t=1}^{T_p} u_{k,s}(p,t) \tag{4.1}$$

   where $u_{k,s}(p,t)$ is 1 if $X_s(p,t) = k$ and 0 otherwise.

3. Use the EM update equations to compute $\boldsymbol{\Theta}(p)$, the estimate of $\boldsymbol{\theta}$ obtained during stage p, using the class label probability estimates from Step 2.

After stage P has been completed, the final estimate of $\boldsymbol{\theta}$, $\boldsymbol{\Theta}(P)$, is computed using the EM update equations, and the MPM algorithm is performed once more, using the final estimate of $\boldsymbol{\theta}$, to obtain the final segmentation.

The EM/MPM procedure seeks to minimize the expected number of misclassified sites in S (the lattice on which $\mathbf{X}$ and $\mathbf{Y}$ are defined) and to approximate the EM estimates of $\boldsymbol{\theta}$. Minimizing the expected number of misclassified sites in S is equivalent to maximizing at each site the marginal conditional probability of the class label given the observed data over all possible classes. It is computationally infeasible to compute these probabilities exactly, so they must be estimated. The segmentation algorithm can be expected to perform well only if the estimates of the marginal probabilities of the class labels are close to the actual values. In the first part of the analysis presented in this chapter we will show that the estimates of the marginal probabilities of the class labels obtained during a given stage of the EM/MPM procedure converge with probability 1 to the true values of the class label probabilities evaluated with $\boldsymbol{\theta}$

equal to the current estimate of $\boldsymbol{\theta}$, conditioned on the estimate of $\boldsymbol{\theta}$ obtained during the previous stage.

The second part of the analysis concerns the parameter estimates $\boldsymbol{\Theta}(p), 1 \leq p \leq P$. In particular, we will describe conditions under which $\boldsymbol{\Theta}(P)$, the final estimate of $\boldsymbol{\theta}$, can be made arbitrarily close to the EM estimate of $\boldsymbol{\theta}$ with probability 1.

Convergence results are first presented for the MPM algorithm when the value of $\boldsymbol{\theta}$ is known. The analysis of the EM/MPM procedure is then presented.

4.1 Convergence of the MPM Algorithm

To determine the convergence properties of the right-hand side of Equation 4.1 when $\boldsymbol{\theta}$ is known, we use Theorem C from [25], which states that if there exists a τ such that $S \subseteq \{q_{t+1}, \ldots, q_{t+\tau}\}$ for all t, then for any function $\mathbf{g}$ on $\Omega_{\mathbf{X}}$ and for any starting configuration $\mathbf{x}(0) \in \Omega_{\mathbf{X}}$,

$$P(\lim_{t\to\infty} \frac{1}{t} \sum_{i=1}^{t} \mathbf{g}(\mathbf{X}(i)) = \int_{\Omega_{\mathbf{X}}} \mathbf{g}(\mathbf{x})\, d\pi(\mathbf{x}) | \mathbf{X}(0) = \mathbf{x}(0)) = 1 \tag{4.2}$$

where π is the limiting distribution of $\mathbf{X}(t)$ (in our case, $\pi(\mathbf{x}) = p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$). The condition $S \subseteq \{q_{t+1}, \ldots, q_{t+\tau}\}$ is easily satisfied. For example, if the pixels are visited in raster-scan order, then $\tau = N$. Let $\mathbf{g}(\mathbf{x})$ be a vector-valued function with elements

$$g_{k,s}(\mathbf{x}) = \begin{cases} 1 & \text{if } x_s = k \\ 0 & \text{if } x_s \neq k \end{cases} \tag{4.3}$$

ordered by a lexicographical ordering of k, s to form a vector. Then

$$\frac{1}{t} \sum_{i=1}^{t} g_{k,s}(\mathbf{X}(i)) = \frac{1}{t} \sum_{i=1}^{t} u_{k,s}(i) \tag{4.4}$$

and

$$\int_{\Omega_{\mathbf{X}}} g_{k,s}(\mathbf{x})\, d\pi(\mathbf{x}) = \sum_{\mathbf{x} \in \Omega_{k,s}} p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}) = p_{X_s|\mathbf{Y}}(k|\mathbf{y}, \boldsymbol{\theta}) \tag{4.5}$$

Hence, using Equation 4.2,

$$P(\lim_{t\to\infty} \frac{1}{t} \sum_{i=1}^{t} u_{k,s}(i) = p_{X_s|\mathbf{Y}}(k|\mathbf{y}, \boldsymbol{\theta})\ \forall k, s | \mathbf{X}(0) = \mathbf{x}(0)) = 1 \tag{4.6}$$

Thus, for any initial configuration, the estimates $\frac{1}{t}\sum_{i=1}^{t} u_{k,s}(i)$ converge to $p_{X_s|\mathbf{Y}}(k|\mathbf{y}, \boldsymbol{\theta})$ for all k, s, with probability 1.

4.2 Analysis of the EM/MPM Procedure

We will prove two results in this section. First, we will show that, for every $p = 1, \ldots, P+1$, $\forall k, s$,

$$\frac{1}{t}\sum_{i=1}^{t} u_{k,s}(p,i) \longrightarrow p_{X_s|\mathbf{Y},\boldsymbol{\Theta}(p-1)}(k|\mathbf{y}, \hat{\boldsymbol{\theta}}(p-1)) \tag{4.7}$$

as $t \longrightarrow \infty$ with probability 1, given that $\boldsymbol{\Theta}(p-1) = \hat{\boldsymbol{\theta}}(p-1)$. Then we will show that, if $\boldsymbol{\theta}(p)$ is a continuous function of $p_{X_s|\mathbf{Y}}(k|\mathbf{y}, \boldsymbol{\theta}(p-1))$ and $p_{X_s|\mathbf{Y}}(k|\mathbf{y}, \boldsymbol{\theta})$ is a continuous function of $\boldsymbol{\theta}$ for every k, s, and if $\boldsymbol{\theta}^*$ is the EM estimate of $\boldsymbol{\theta}$, then for any $\varepsilon > 0$, $|\boldsymbol{\Theta}(P) - \boldsymbol{\theta}^*| < \varepsilon$ if P and $T_p, 1 \leq p \leq P$ are chosen large enough. It can be shown that the continuity conditions required for the second result hold for the EM/MPM and multiresolution EM/MPM algorithms described in Chapters 2 and 3.

Stages $p = 1, \ldots, P$ of the EM/MPM procedure are used to obtain the final parameter estimate $\boldsymbol{\Theta}(P)$, and stage $P+1$ is used to obtain the segmented image. The segmented image is obtained using the estimate $\boldsymbol{\Theta}(P)$ for $\boldsymbol{\theta}$. Thus, the analytical results described above imply that the final estimates of the marginal probabilities of the class labels obtained using the EM/MPM procedure will be close to the true values of the marginal probabilities of the class labels, to the extent that the EM estimate of $\boldsymbol{\theta}$ is close to the true value of $\boldsymbol{\theta}$.

We now prove the analytical results described above. We are interested in the limiting behavior of the right-hand side of Equation 2.18, and also in properties of the estimate $\boldsymbol{\Theta}(P)$.

4.2.1 Convergence of the class label probabilities

Considering $\mathbf{X}(1,t)$ first, we know that, by construction,

$$P(\mathbf{X}(1,t) = \mathbf{x}(1,t) | \mathbf{X}(1,t-1) = \mathbf{x}(1,t-1), \mathbf{Y} = \mathbf{y}, \boldsymbol{\Theta}(0) = \hat{\boldsymbol{\theta}}(0)) =$$

$$p_{X_{q_t}|X_s, s \neq q_t, \mathbf{Y}, \boldsymbol{\Theta}(0)}(x_{q_t}(1,t) | x_s(1,t), s \neq q_t, \mathbf{y}, \hat{\boldsymbol{\theta}}(0)) \tag{4.8}$$

if $x_s(1,t) = x_s(1,t-1)\ \forall s \neq q_t$. If $x_s(1,t) \neq x_s(1,t-1)$ for any $s \neq q_t$ then the probability on the left-hand side of Equation 4.8 is equal to 0. Following the proof of

Theorem A in [25], it can be shown that, if $\{q_t, t \geq 1\}$ contains every $s \in S$ infinitely often then for any $\mathbf{x}(1,0) \in \Omega_{\mathbf{X}}$, $\mathbf{y} \in \Omega_{\mathbf{Y}}$, $\hat{\boldsymbol{\theta}}(0) \in \Omega_\theta$, and $\mathbf{x} \in \Omega_{\mathbf{X}}$,

$$\lim_{t\to\infty} P(\mathbf{X}(1,t) = \mathbf{x} | \mathbf{X}(1,0) = \mathbf{x}(1,0), \mathbf{Y} = \mathbf{y}, \boldsymbol{\Theta}(0) = \hat{\boldsymbol{\theta}}(0)) = p_{\mathbf{X}|\mathbf{Y},\boldsymbol{\Theta}(0)}(\mathbf{x}|\mathbf{y}, \hat{\boldsymbol{\theta}}(0)) \tag{4.9}$$

Also, using Theorem C from [25] as described in Section 4.1,

$$P(\lim_{t\to\infty} v_{k,s}(1,t) = p_{X_s|\mathbf{Y},\boldsymbol{\Theta}(0)}(k|\mathbf{y}, \hat{\boldsymbol{\theta}}(0)) | \mathbf{X}(1,0) = \mathbf{x}(1,0), \mathbf{Y} = \mathbf{y}, \boldsymbol{\Theta}(0) = \hat{\boldsymbol{\theta}}(0) \forall k, s)$$

$$= 1 \tag{4.10}$$

for any $\mathbf{x}(1,0) \in \Omega_{\mathbf{X}}, \mathbf{y} \in \Omega_{\mathbf{Y}}, \hat{\boldsymbol{\theta}}(0) \in \Omega_\theta$.

An important consequence of the method used to construct $\mathbf{X}(1,t), \mathbf{X}(2,t), \ldots, \mathbf{X}(P+1,t)$ is that for any $p \in \{2, \ldots, P+1\}$ and any $m_p \geq 1$, the random variables in $\mathbf{X}(p,1), \mathbf{X}(p,2), \ldots, \mathbf{X}(p,m_p)$ are conditionally independent of the random variables in stages $1, 2, \ldots, p-1$, given $\boldsymbol{\Theta}(p-1)$. This means that, if A_p denotes the event $\{\mathbf{X}(p-1,0) = \mathbf{x}(p-1,0), \ldots, \mathbf{X}(p-1,m_{p-1}) = \mathbf{x}(p-1,m_{p-1}), \boldsymbol{\Theta}(p-2) = \hat{\boldsymbol{\theta}}(p-2), \mathbf{X}(p-2,0) = \mathbf{x}(p-2,0), \ldots, \mathbf{X}(p-2,m_{p-2}) = \mathbf{x}(p-2,m_{p-2}), \ldots, \boldsymbol{\Theta}(1) = \hat{\boldsymbol{\theta}}(1), \mathbf{X}(1,0) = \mathbf{x}(1,0), \ldots, \mathbf{X}(1,m_1) = \mathbf{x}(1,m_1), \boldsymbol{\Theta}(0) = \boldsymbol{\theta}(0)\}$ and B_p denotes the event $\{\boldsymbol{\Theta}(p-1) = \hat{\boldsymbol{\theta}}(p-1)\}$, then

$$P(\mathbf{X}(p,1) = \mathbf{x}(p,1), \ldots, \mathbf{X}(p,m_p) = \mathbf{x}(p,m_p) | \mathbf{X}(p,0) = \mathbf{x}(p,0), \mathbf{Y} = \mathbf{y}, A_p, B_p) =$$

$$P(\mathbf{X}(p,1) = \mathbf{x}(p,1), \ldots, \mathbf{X}(p,m_p) = \mathbf{x}(p,m_p) | \mathbf{X}(p,0) = \mathbf{x}(p,0), \mathbf{Y} = \mathbf{y}, B_p) \tag{4.11}$$

Now, by construction, for any p,

$$P(\mathbf{X}(p,t) = \mathbf{x}(p,t) | \mathbf{X}(p,t-1) = \mathbf{x}(p,t-1), \mathbf{Y} = \mathbf{y}, B_p) =$$

$$p_{X_{q_t}|X_s, s\neq q_t, \mathbf{Y}, \boldsymbol{\Theta}(p-1)}(x_{q_t}(p,t) | x_s(p,t), \mathbf{y}, \hat{\boldsymbol{\theta}}(p-1)) \tag{4.12}$$

if $x_s(p,t) = x_s(p,t-1)\ \forall s \neq q_t$. If $x_s(p,t) \neq x_s(p,t-1)$ for any $s \neq q_t$ then the probability on the left-hand side of Equation 4.12 is equal to 0. Hence, using Theorems A and C from [25],

$$P(\lim_{t\to\infty} v_{k,s}(p,t) = p_{X_s|\mathbf{Y},\boldsymbol{\Theta}(p-1)}(k|\mathbf{y}, \hat{\boldsymbol{\theta}}(p-1)) \forall k, s | \mathbf{X}(p,0) = \mathbf{x}(p,0), \mathbf{Y} = \mathbf{y}, B_p)$$

$$= 1 \tag{4.13}$$

for any $\mathbf{x}(p,0) \in \Omega_{\mathbf{x}}, \mathbf{y} \in \Omega_{\mathbf{y}}$, and $\hat{\boldsymbol{\theta}}(p-1) \in \Omega_\theta$ and, by virtue of Equation 4.11, the estimates $v_{k,s}(p,t)$ converge with probability 1 to $p_{X_s|\mathbf{Y},\boldsymbol{\Theta}(p-1)}(k|\mathbf{y},\hat{\boldsymbol{\theta}}(p-1))$, given $\boldsymbol{\Theta}(p-1) = \hat{\boldsymbol{\theta}}(p-1)$, independently of all other random variables in stages $1,\ldots,p-1$.

4.2.2 Discussion

The above analysis has shown the desired result for the MPM portion of the EM/MPM procedure at each stage. In particular, we have shown that the estimates of the marginal probabilities of the class labels obtained during a given stage of the EM/MPM procedure converge with probability 1 to the true values of the class label probabilities evaluated with the model parameters equal to the current estimates of the model parameters, given the estimates of the model parameters obtained during the previous stage.

4.2.3 Analysis of the parameter estimates

We next examine the estimates $\{\boldsymbol{\Theta}(p), 1 \leq p \leq P\}$. If $\{\boldsymbol{\theta}(p), p \geq 1\}$ is the sequence of estimates obtained using the EM algorithm when the values of $p_{X_s|\mathbf{Y}}(k|\mathbf{y},\boldsymbol{\theta})$ are known, suppose that $\boldsymbol{\theta}(p)$ converges to $\boldsymbol{\theta}^*$ for some $\boldsymbol{\theta}^* \in \Omega_\theta$. We would like for $\boldsymbol{\Theta}(P)$ to be close to $\boldsymbol{\theta}^*$ in some sense. In particular, for an arbitrarily small but positive ε we address the question of whether $|\boldsymbol{\Theta}(P) - \boldsymbol{\theta}^*|$ can be made less than ε by making P and $T_1,\ldots,T_P$ large enough.

We define the notation $a_{k,s}(\boldsymbol{\theta}) = p_{X_s|\mathbf{Y}}(k|\mathbf{y},\boldsymbol{\theta})$. (Note that the dependence on $\mathbf{y}$ is suppressed in this notation). We assume that $\boldsymbol{\theta}(p)$ is a continuous function of $a_{k,s}(\boldsymbol{\theta}(p-1))$ for every k,s and that $a_{k,s}(\boldsymbol{\theta})$ is a continuous function of $\boldsymbol{\theta}$, for every k,s. We also assume that $\boldsymbol{\Theta}(0) = \boldsymbol{\theta}(0)$, i.e., the EM/MPM procedure is started with the same initial estimate of $\boldsymbol{\theta}$ as the EM sequence that converges to $\boldsymbol{\theta}^*$.

Fix $\varepsilon > 0$. Since $\boldsymbol{\theta}(p) \longrightarrow \boldsymbol{\theta}^*$, there exists a $P >= 1$ such that $|\boldsymbol{\theta}(p) - \boldsymbol{\theta}^*| < \varepsilon/2$ for all $p \geq P$. Fix P to be such a value. Then, to make $|\boldsymbol{\Theta}(P) - \boldsymbol{\theta}^*| < \varepsilon$, it is sufficient

to make $|\boldsymbol{\Theta}(P) - \boldsymbol{\theta}(P)| < \varepsilon/2$, since

$$|\boldsymbol{\Theta}(P) - \boldsymbol{\theta}^*| \leq |\boldsymbol{\Theta}(P) - \boldsymbol{\theta}(P)| + |\boldsymbol{\theta}(P) - \boldsymbol{\theta}^*| \tag{4.14}$$

and, hence,

$$|\boldsymbol{\Theta}(P) - \boldsymbol{\theta}(P)| < \frac{\varepsilon}{2} \Longrightarrow |\boldsymbol{\Theta}(P) - \boldsymbol{\theta}^*| < \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon \tag{4.15}$$

Since $\boldsymbol{\theta}(p)$ is a continuous function of $a_{k,s}(\boldsymbol{\theta}(p-1))$, there exists a $\delta_P > 0$ such that

$$|v_{k,s}(P, T_P) - a_{k,s}(\boldsymbol{\theta}(P-1))| < \delta_P \; \forall k, s \Longrightarrow |\boldsymbol{\Theta}(P) - \boldsymbol{\theta}(P)| < \frac{\varepsilon}{2} \tag{4.16}$$

Fix δ_P to satisfy this condition. For any $\hat{\boldsymbol{\theta}}(P-1) \in \Omega_\theta$, by the triangle inequality,

$$|v_{k,s}(P, T_P) - a_{k,s}(\boldsymbol{\theta}(P-1))| \leq$$

$$|v_{k,s}(P, T_P) - a_{k,s}(\hat{\boldsymbol{\theta}}(P-1))| + |a_{k,s}(\hat{\boldsymbol{\theta}}(P-1)) - a_{k,s}(\boldsymbol{\theta}(P-1))| \tag{4.17}$$

Looking at the first term on the right-hand side of Equation 4.17, we recall that $v_{k,s}(P,t)$ converges to $a_{k,s}(\hat{\boldsymbol{\theta}}(P-1))$ with probability 1, given that $\boldsymbol{\Theta}(P-1) = \hat{\boldsymbol{\theta}}(P-1)$. Considering the second term on the right-hand side of Equation 4.17, since, for every k, s, $a_{k,s}(\boldsymbol{\theta})$ is continuous at $\boldsymbol{\theta} = \boldsymbol{\theta}(P)$, there exists an ε_{P-1} such that

$$|\boldsymbol{\Theta}(P-1) - \boldsymbol{\theta}(P-1)| < \varepsilon_{P-1} \Longrightarrow |a_{k,s}(\boldsymbol{\Theta}(P-1)) - a_{k,s}(\boldsymbol{\theta}(P-1))| < \frac{\delta_P}{2} \; \forall k, s \tag{4.18}$$

Fix ε_{P-1} to be such a value. We choose a set of values $\{\varepsilon_p, \delta_p, 1 \leq p \leq P\}$, where $\varepsilon_P = \varepsilon/2$, and δ_P and ε_{P-1} are fixed as described above. For $p = P-1, P-2, \ldots, 2$, $\delta_p > 0$ is chosen to satisfy

$$|v_{k,s}(p, T_p) - a_{k,s}(\boldsymbol{\theta}(p-1))| < \delta_p \; \forall k, s \Longrightarrow |\boldsymbol{\Theta}(p) - \boldsymbol{\theta}(p)| < \varepsilon_p \tag{4.19}$$

and $\varepsilon_{p-1} > 0$ is chosen to satisfy

$$|\boldsymbol{\Theta}(p-1) - \boldsymbol{\theta}(p-1)| < \varepsilon_{p-1} \Longrightarrow |a_{k,s}(\boldsymbol{\Theta}(p-1)) - a_{k,s}(\boldsymbol{\theta}(p-1))| < \frac{\delta_p}{2} \tag{4.20}$$

Finally, the value δ_1 is chosen such that

$$|v_{k,s}(1, T_1) - a_{k,s}(\boldsymbol{\theta}(0))| < \delta_1 \; \forall k, s \Longrightarrow |\boldsymbol{\Theta}(1) - \boldsymbol{\theta}(1)| < \varepsilon_1 \tag{4.21}$$

From Equation 4.21, it can be seen that

$$P(|\boldsymbol{\Theta}(1)-\boldsymbol{\theta}(1)| < \varepsilon_1 | \boldsymbol{\Theta}(0) = \boldsymbol{\theta}(0)) \geq P(|v_{k,s}(1,T_1)-a_{k,s}(\boldsymbol{\theta}(0))| < \delta_1 \forall k,s | \boldsymbol{\Theta}(0) = \boldsymbol{\theta}(0)) \tag{4.22}$$

Recall that, for every k, s, $v_{k,s}(1,t)$ converges to $a_{k,s}(\boldsymbol{\theta}(0))$ with probability 1, given that $\boldsymbol{\Theta}(0) = \boldsymbol{\theta}(0)$. Let $(\Omega, \mathcal{F}, P)$ be the underlying probability space, i.e., Ω is the set of possible outcomes, $\mathcal{F}$ is a σ-algebra in Ω containing all events, and P is a probability measure. Then the event

$$A_{1,\theta(0)} = \{\omega \in \Omega : \lim_{t\to\infty} v_{k,s}(1,t,\omega) = a_{k,s}(\boldsymbol{\theta}(0))\ \forall k,s\}, \tag{4.23}$$

where $v_{k,s}(1,t)$ is written as $v_{k,s}(1,t,\omega)$ to explicitly denote its dependence on ω, occurs with probability 1, given that $\boldsymbol{\Theta}(0) = \boldsymbol{\theta}(0)$. Now, for every $\boldsymbol{\theta}(0) \in \Omega_\theta$ and $\omega \in A_{1,\theta(0)}$, there exists a $T(\boldsymbol{\theta}(0),\omega)$ such that, given that $\boldsymbol{\Theta}(0) = \boldsymbol{\theta}(0)$

$$|v_{k,s}(1,t,\omega) - a_{k,s}(\boldsymbol{\theta}(0))| < \delta_1\ \forall t \geq T(\boldsymbol{\theta}(0),\omega), \forall k,s \tag{4.24}$$

Letting $T_1 = \sup_{\theta(0)\in\Omega_\theta} \{ \sup_{\omega\in A_{1,\theta(0)}} \{T(\boldsymbol{\theta}(0),\omega)\}\}$ gives

$$A_{1,\theta(0)} \subseteq \{\omega \in \Omega : |v_{k,s}(1,t,\omega) - a_{k,s}(\boldsymbol{\theta}(0))| < \delta_1\ \forall t \geq T_1, \forall k,s\} \tag{4.25}$$

for any $\boldsymbol{\theta}(0) \in \Omega_\theta$. Then

$$P(|v_{k,s}(1,T_1) - a_{k,s}(\boldsymbol{\theta}(0))| < \delta_1 \forall k,s | \boldsymbol{\Theta}(0) = \boldsymbol{\theta}(0)) \geq P(A_{1,\theta(0)} | \boldsymbol{\Theta}(0) = \boldsymbol{\theta}(0)) = 1 \tag{4.26}$$

From Equations 4.22 and 4.26, we see that,

$$P(|\boldsymbol{\Theta}(1) - \boldsymbol{\theta}(1)| < \varepsilon_1 | \boldsymbol{\Theta}(0) = \boldsymbol{\theta}(0)) = 1\ \ \forall \boldsymbol{\theta}(0) \in \Omega_\theta \tag{4.27}$$

Then

$$P(|\boldsymbol{\Theta}(1) - \boldsymbol{\theta}(1)| < \varepsilon_1) = \int_{\Omega_\theta} P(|\boldsymbol{\Theta}(1) - \boldsymbol{\theta}(1)| < \varepsilon_1 | \boldsymbol{\Theta}(0) = \boldsymbol{\theta}(0)) f_{\boldsymbol{\Theta}(0)}(\boldsymbol{\theta}(0))\, d\boldsymbol{\theta}(0) \tag{4.28}$$

where $f_{\boldsymbol{\Theta}(0)}(\boldsymbol{\theta}(0))$ is the probability density function of $\boldsymbol{\Theta}(0)$, and from Equation 4.27,

$$P(|\boldsymbol{\Theta}(1) - \boldsymbol{\theta}(1)| < \varepsilon_1) = \int_{\Omega_\theta} f_{\boldsymbol{\Theta}(0)}(\boldsymbol{\theta}(0))\, d\boldsymbol{\theta}(0) = 1 \tag{4.29}$$

Next, consider $\boldsymbol{\Theta}(2)$. From Equation 4.19 with $p = 2$ it can be seen that

$$P(|\boldsymbol{\Theta}(2) - \boldsymbol{\theta}(2)| < \varepsilon_2) \geq P(|v_{k,s}(2, T_2) - a_{k,s}(\boldsymbol{\theta}(1))| < \delta_2 \forall k, s) \tag{4.30}$$

and, since

$$|v_{k,s}(2, T_2) - a_{k,s}(\boldsymbol{\theta}(1))| \leq |v_{k,s}(2, T_2) - a_{k,s}(\boldsymbol{\Theta}(1))| + |a_{k,s}(\boldsymbol{\Theta}(1)) - a_{k,s}(\boldsymbol{\theta}(1))| \tag{4.31}$$

we have that

$$P(|v_{k,s}(2, T_2) - a_{k,s}(\boldsymbol{\theta}(1))| < \delta_2 \ \forall k, s) \geq$$

$$P(|v_{k,s}(2, T_2) - a_{k,s}(\boldsymbol{\Theta}(1))| < \frac{\delta_2}{2} \ \forall k, s, \quad |a_{k,s}(\boldsymbol{\Theta}(1)) - a_{k,s}(\boldsymbol{\theta}(1))| < \frac{\delta_2}{2} \ \forall k, s)$$

$$= P(|v_{k,s}(2, T_2) - a_{k,s}(\boldsymbol{\Theta}(1))| < \frac{\delta_2}{2} \ \forall k, s \mid |a_{k,s}(\boldsymbol{\Theta}(1)) - a_{k,s}(\boldsymbol{\theta}(1))| < \frac{\delta_2}{2} \ \forall k, s) \ \cdot$$

$$P(|a_{k,s}(\boldsymbol{\Theta}(1)) - a_{k,s}(\boldsymbol{\theta}(1))| < \frac{\delta_2}{2} \ \forall k, s) \tag{4.32}$$

From the continuity of $a_{k,s}(\boldsymbol{\theta})$ at $\boldsymbol{\theta} = \boldsymbol{\theta}(1)$,

$$P(|a_{k,s}(\boldsymbol{\Theta}(1)) - a_{k,s}(\boldsymbol{\theta}(1))| < \frac{\delta_2}{2} \ \forall k, s) \geq P(|\boldsymbol{\Theta}(1) - \boldsymbol{\theta}(1))| < \varepsilon_1) = 1 \tag{4.33}$$

Also,

$$P(|v_{k,s}(2, T_2) - a_{k,s}(\boldsymbol{\Theta}(1))| < \frac{\delta_2}{2} \ \forall k, s \mid |a_{k,s}(\boldsymbol{\Theta}(1)) - a_{k,s}(\boldsymbol{\theta}(1))| < \frac{\delta_2}{2} \ \forall k, s)$$

$$= \int_{\Omega_\theta} P(|v_{k,s}(2, T_2) - a_{k,s}(\boldsymbol{\Theta}(1))| < \frac{\delta_2}{2} \ \forall k, s \mid$$

$$\boldsymbol{\Theta}(1) = \hat{\boldsymbol{\theta}}(1), |a_{k,s}(\boldsymbol{\Theta}(1)) - a_{k,s}(\boldsymbol{\theta}(1))| < \frac{\delta_2}{2} \ \forall k, s) \ \cdot$$

$$f_{\boldsymbol{\Theta}(1)}(\hat{\boldsymbol{\theta}}(1) | |a_{k,s}(\boldsymbol{\Theta}(1)) - a_{k,s}(\boldsymbol{\theta}(1))| < \frac{\delta_2}{2} \ \forall k, s)) \, d\hat{\boldsymbol{\theta}}(1) \tag{4.34}$$

The integrand in Equation 4.34 is zero if $|a_{k,s}(\hat{\boldsymbol{\theta}}(1)) - a_{k,s}(\boldsymbol{\theta}(1))| \geq \frac{\delta_2}{2}$ for any k, s, since the conditional probability density function of $\boldsymbol{\Theta}(1)$ is zero for this case. If $|a_{k,s}(\hat{\boldsymbol{\theta}}(1)) - a_{k,s}(\boldsymbol{\theta}(1))| < \frac{\delta_2}{2} \ \forall k, s$, then

$$P(|v_{k,s}(2, T_2) - a_{k,s}(\boldsymbol{\Theta}(1))| < \frac{\delta_2}{2} \ \forall k, s \mid$$

$$\boldsymbol{\Theta}(1) = \hat{\boldsymbol{\theta}}(1), |a_{k,s}(\boldsymbol{\Theta}(1)) - a_{k,s}(\boldsymbol{\theta}(1))| < \frac{\delta_2}{2} \ \forall k, s) =$$

$$P(|v_{k,s}(2,T_2) - a_{k,s}(\boldsymbol{\Theta}(1))| < \frac{\delta_2}{2}\ \forall k,s \mid \boldsymbol{\Theta}(1) = \hat{\boldsymbol{\theta}}(1)) \tag{4.35}$$

The value of T_2 can be selected the same way T_1 was selected. In particular, the event

$$A_{2,\hat{\theta}(1)} = \{\omega \in \Omega : \lim_{t\to\infty} v_{k,s}(2,t,\omega) = a_{k,s}(\hat{\boldsymbol{\theta}}(1))\ \forall k,s\}, \tag{4.36}$$

occurs with probability 1, given that $\boldsymbol{\Theta}(1) = \hat{\boldsymbol{\theta}}(1)$. For every $\hat{\boldsymbol{\theta}}(1) \in \Omega_\theta$ and $\omega \in A_{2,\hat{\theta}(1)}$, there exists a $T(\hat{\boldsymbol{\theta}}(1),\omega)$ such that, given that $\boldsymbol{\Theta}(1) = \hat{\boldsymbol{\theta}}(1)$,

$$|v_{k,s}(2,t,\omega) - a_{k,s}(\hat{\boldsymbol{\theta}}(1))| < \frac{\delta_2}{2}\ \forall t \geq T(\hat{\boldsymbol{\theta}}(1),\omega), \forall k,s \tag{4.37}$$

Letting $T_2 = \sup_{\hat{\theta}(1)\in\Omega_\theta} \{ \sup_{\omega\in A_{2,\hat{\theta}(1)}} \{T(\boldsymbol{\theta}(0),\omega)\}\}$ gives

$$A_{2,\hat{\theta}(1)} \subseteq \{\omega \in \Omega : |v_{k,s}(2,t,\omega) - a_{k,s}(\hat{\boldsymbol{\theta}}(1))| < \frac{\delta_2}{2}\ \forall t \geq T_2, \forall k,s\} \tag{4.38}$$

for any $\hat{\boldsymbol{\theta}}(1) \in \Omega_\theta$. Then for any $\hat{\boldsymbol{\theta}}(1)$

$$P(|v_{k,s}(2,T_2) - a_{k,s}(\hat{\boldsymbol{\theta}}(1))| < \frac{\delta_2}{2}\forall k,s|\boldsymbol{\Theta}(1) = \hat{\boldsymbol{\theta}}(1)) \geq P(A_{2,\hat{\theta}(1)}|\boldsymbol{\Theta}(1) = \hat{\boldsymbol{\theta}}(1)) = 1 \tag{4.39}$$

which, using Equation 4.34, leads to

$$P(|v_{k,s}(2,T_2) - a_{k,s}(\boldsymbol{\Theta}(1))| < \frac{\delta_2}{2}\ \forall k,s \mid |a_{k,s}(\boldsymbol{\Theta}(1)) - a_{k,s}(\boldsymbol{\theta}(1))| < \frac{\delta_2}{2}\ \forall k,s)$$

$$= \int_{\Omega_\theta} f_{\boldsymbol{\Theta}(1)}(\hat{\boldsymbol{\theta}}(1)||a_{k,s}(\boldsymbol{\Theta}(1)) - a_{k,s}(\boldsymbol{\theta}(1))| < \frac{\delta_2}{2}\ \forall k,s))\, d\hat{\boldsymbol{\theta}}(1) = 1 \tag{4.40}$$

From Equations 4.32, 4.33, and 4.40, we see that

$$P(|v_{k,s}(2,T_2) - a_{k,s}(\boldsymbol{\theta}(1))| < \frac{\delta_2}{2}\ \forall k,s) = 1 \tag{4.41}$$

and, thus, from Equation 4.30

$$P(|\boldsymbol{\Theta}(2) - \boldsymbol{\theta}(2)| < \varepsilon_2) = 1 \tag{4.42}$$

Continuing the above procedure, it can be shown that for $1 \leq p \leq P$,

$$P(|\boldsymbol{\Theta}(p) - \boldsymbol{\theta}(p)| < \varepsilon_p) = 1 \tag{4.43}$$

if the $T_p, 1 \leq p \leq P$, are made large enough. Thus

$$P(|\boldsymbol{\Theta}(P) - \boldsymbol{\theta}(P)| < \frac{\varepsilon}{2}) = 1 \tag{4.44}$$

which means that

$$\begin{aligned} P(|\boldsymbol{\Theta}(P) - \boldsymbol{\theta}^*| < \varepsilon) &\geq P(|\boldsymbol{\Theta}(P) - \boldsymbol{\theta}(P)| < \frac{\varepsilon}{2}, |\boldsymbol{\theta}(P) - \boldsymbol{\theta}^*| < \frac{\varepsilon}{2}) \\ &= P(|\boldsymbol{\Theta}(P) - \boldsymbol{\theta}(P)| < \frac{\varepsilon}{2}) = 1 \end{aligned} \tag{4.45}$$

Thus, $\boldsymbol{\Theta}(P)$ can be made arbitrarily close to $\boldsymbol{\theta}^*$ with probability 1.

4.2.4 Discussion

The above analysis provides the desired result relative to the parameter estimation portion of the EM/MPM procedure. In particular, we have shown that the parameter estimates resulting from the EM/MPM procedure can be made arbitrarily close to the EM estimates of the parameters with probability 1, if a sufficient number of iterations is performed. It should be noted that there is no guarantee that the number of iterations required at each stage for this result to hold are finite, but the analysis provides theoretical evidence that a sufficient number of iterations of the Gibbs sampler at each stage will provide an estimate of $\boldsymbol{\theta}$ close to the EM estimate.

4.3 Conclusion

We have presented a theoretical analysis of the EM/MPM procedure which explains its superior performance over the deterministic EM/MPM algorithm. In particular, we have shown that the estimates of the marginal probabilities of the class labels obtained during a given stage of the EM/MPM procedure converge with probability 1 to the true values of the class label probabilities evaluated with $\boldsymbol{\theta}$ equal to the current estimate of $\boldsymbol{\theta}$, given the estimate of $\boldsymbol{\theta}$ obtained during the previous stage. We have also shown that the parameter estimates resulting from the EM/MPM procedure can be made arbitrarily close to the EM estimates of the parameters with probability 1, if a sufficient number of iterations is performed. These analytical results imply that

the final estimates of the marginal probabilities of the class labels obtained using the EM/MPM procedure will be close to the true values of the marginal probabilities of the class labels, to the extent that the EM estimate of $\boldsymbol{\theta}$ is close to the true value of $\boldsymbol{\theta}$. There is no such guarantee that the deterministic EM/MPM algorithm will converge.

5. MORPHOLOGICAL OPERATIONS FOR COLOR IMAGE PROCESSING

Mathematical morphology has been shown to be useful for the processing and analysis of binary and grayscale images [19, 20, 21]. Morphological filtering of an image involves transforming the image into another image using a function or set, known as the structuring element, which acts as a probe sensitive to geometrical information. Geometrical features of the image that are similar in shape and size to the structuring element are preserved, while other features are extracted or suppressed. Morphology has been used to perform noise suppression, texture analysis, shape analysis, edge detection, skeletonization, and multiscale filtering for applications such as medical imaging, geological image processing, automated industrial inspection, image compression, and ECG signal analysis [19, 31, 32, 33, 34, 35, 36, 37, 38]. In this chapter we discuss the extension of mathematical morphology to color imagery.

Many techniques developed for use with monochrome images can be extended to color imagery by applying the algorithm to each of the color component images separately. An important question arises: Is component-wise spatial filtering sufficient? Algorithms that exploit spectral correlations could provide better performance and be more computationally efficient. For example, several authors have addressed this issue and proposed linear least-squares and minimum mean-square error restoration algorithms developed specifically for multispectral images [39, 40, 41, 42, 43, 44]. A general framework for linear filtering of multichannel signals in the frequency domain was presented in [45].

Several nonlinear filtering algorithms for multispectral image enhancement have been proposed [46, 47, 48]. These algorithms are based on the concept of ranking

multivariate data [49]. In [46] the vector median filter was developed as an extension of the median filter for scalar-valued signals. The vector median filter examines each pixel of a color image as a vector, rather than individual scalar components. The vector median of a collection of vectors is the vector from the collection which has the minimum aggregate distance from all other vectors in the collection [46, 49]. The output of the vector median filter at a given pixel of a color image is the vector median of the color vectors inside the filter window.

In [47] a class of ranked-order based filters for multichannel images was presented. At each pixel, a region of confidence is determined based on the order statistics of the vectors at neighboring pixels. If the vector at the given pixel does not lie inside this region of confidence, then it is assumed to be an outlier, and it is replaced by the vector in the region of confidence which is at a minimum distance. Thus, these ranked-order vector filters remove "atypical" samples, which makes them suitable for noise suppression in color images.

Vector directional filters for multichannel image processing were introduced in [48]. These filters process vector-valued signals in two steps. First, vectors are processed based on direction, or angle, resulting in the removal of vectors with atypical directions. Then, magnitude processing is performed using any classical grayscale image processing filter.

In this chapter we describe new approaches for applying mathematical morphology to color images. We introduce new vector morphological filters which are based on vector ranking concepts [49, 50], and present a set-theoretic analysis relative to these vector operations. We compare the performance of the vector morphological filters and component-wise morphological filters for the applications of multiscale image analysis and noise suppression.

5.1 Mathematical Morphology

Mathematical morphology is based on set theory. A morphological operation defined on a binary image is referred to as binary morphology. This involves representing

the image as a set $X \subseteq \mathbf{R}^2$ or $\mathbf{Z}^2$ (depending on whether the image is defined on a discrete or continuous lattice—an image defined on a discrete lattice will be referred to as a discrete-space image and an image defined on a continuous lattice will be referred to as a continuous-space image), where $\mathbf{R}$ is the set of real numbers and $\mathbf{Z}$ is the set of integers. Points in the image foreground are members of X and points in the background are members of the complement of X, designated X^c. The image is transformed by another set, known as the structuring element. The shape and size of the structuring element determine the resultant image [19, 20, 21, 51, 52, 53].

There are four basic binary morphological operations: dilation, erosion, opening, and closing, represented by the symbols $\oplus$, $\ominus$, $\circ$, and $\bullet$, respectively. The four operations are defined as follows:

$$X \oplus H = \{(x,y) : H_{(x,y)} \cap X \neq \emptyset\} \tag{5.1}$$

$$X \ominus H = \{(x,y) : H_{(x,y)} \subseteq X\} \tag{5.2}$$

$$X \circ H = (X \ominus H) \oplus H \tag{5.3}$$

$$X \bullet H = (X \oplus H) \ominus H \tag{5.4}$$

where X is the original image, $H \subseteq \mathbf{R}^2$ or $\mathbf{Z}^2$ is the structuring element, and $H_{(x,y)}$ is the translate of the set H by the vector $(x,y) \in \mathbf{R}^2$ or $\mathbf{Z}^2$.

Grayscale morphological operations are an extension of binary morphological operations to grayscale images. For grayscale operations, the image will be represented by the function $f(x,y)$, where $(x,y) \in \mathbf{R}^2$ or $\mathbf{Z}^2$, or simply f, and the structuring element will be the function $h(x,y)$, or h. Grayscale dilation, erosion, opening, and closing are defined as follows:

$$(f \oplus h)(x,y) = \sup_{(r,s)\in H} \{f(x-r, y-s) + h(r,s)\} \tag{5.5}$$

$$(f \ominus h)(x,y) = \inf_{(r,s)\in H} \{f(x+r, y+s) - h(r,s)\} \tag{5.6}$$

$$f \circ h = (f \ominus h) \oplus h \tag{5.7}$$

$$f \bullet h = (f \oplus h) \ominus h \tag{5.8}$$

where sup{} and inf{} denote the supremum and infimum operators, respectively, and $H \subseteq \mathbf{R}^2$ or $\mathbf{Z}^2$ is the support of $h(x,y)$. A special class of grayscale morphological filters, referred to as function-and-set-processing (FSP) filters [21], results when $h(x,y) = 0$ for every $(x,y) \in H$. The resulting operations in Equations 5.5 through 5.8 are then written as $f \oplus H$, $f \ominus H$, $f \circ H$, and $f \bullet H$.

Alternative definitions and detailed descriptions of the operations defined in this section can be found in [19, 20, 21, 52, 53].

5.2 Color Images and Color Spaces

It is well-known that a normal human observer can match any color stimulus by combining light from three primary sources in the correct proportions [54]. Therefore, any color can be represented by three values which correspond to the amounts of the three primaries required to match the color. These three values are known as the tristimulus values. In a digital color image, each pixel has associated with it a vector with three components which represent the tristimulus values of the color at that spatial location in the image. If the tristimulus values of a color image $\mathbf{f}(x,y)$ represent amounts of red, green, and blue primaries, then the image can be modeled as

$$\mathbf{f}(x,y) = [f_R(x,y), f_G(x,y), f_B(x,y)]^{\mathrm{T}} \tag{5.9}$$

where $f_R(x,y)$ is the red tristimulus value, $f_G(x,y)$ is the green value, and $f_B(x,y)$ is the blue value, at spatial location (x,y).

For many important applications it is more convenient to work in a color space other than RGB space. The YIQ color space is a linear transformation of the RGB space which consists of a luminance signal, Y, and two chrominance signals, I and Q. This is the standard system used for NTSC color television transmission in the United States [55]. The equations used to transform a set of tristimulus values R, G,

and B in RGB space to tristimulus values Y, I, and Q in YIQ space are

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{5.10}$$

The YIQ color space is used for some of the experiments presented in Section 5.5.

5.3 Color Morphology

The extension of mathematical morphology to color images is not straightforward. In [56] Serra discusses the generalization of morphology to its most basic elements, and concludes that the axioms can be reduced to three key ideas: an order relationship (for example, set inclusion for binary morphology), a supremum or an infimum pertaining to that order, and the possibility of admitting an infinity of operands. The first two of these, the order relationship and the supremum (or infimum), are missing in color images, because there is no unambiguous way to order two or more colors. The fact that these fundamental concepts of morphology do not apply to color images makes it difficult to define "color morphology". However, it is possible that some of the techniques can be extended to color images.

The problem of ordering multivariate data is not unique to mathematical morphology. Although there is no natural means for total ordering of multivariate samples, much work has been done to define concepts such as median, range, and extremes in multivariate analysis. Barnett proposed the classification of these sub-ordering principles into four groups: marginal ordering, reduced ordering, partial ordering, and conditional ordering [49]. The two types of sub-ordering used in this chapter are marginal ordering and reduced ordering.

In marginal ordering ranking takes place within one or more of the marginal sets of samples [49], i.e., scalar ranking is performed within each channel. Thus, to order a collection of color vectors using marginal ordering the components in each spectral band are ordered independently of the components in other spectral bands. Morphological operations which are defined using marginal ordering will

be referred to in this chapter as component-wise operations. The component-wise color dilation of $\mathbf{f}(x,y) = [f_R(x,y), f_G(x,y), f_B(x,y)]^T$ by the structuring element $\mathbf{h}(x,y) = [h_R(x,y), h_G(x,y), h_B(x,y)]^{\mathrm{T}}$ in RGB color space is defined as

$$(\mathbf{f} \oplus_c \mathbf{h})(x,y) = [(f_R \oplus h_R)(x,y), (f_G \oplus h_G)(x,y), (f_B \oplus h_B)(x,y)]^{\mathrm{T}} \qquad (5.11)$$

where the symbol $\oplus_c$ represents component-wise dilation and the dilations on the right-hand side of Equation 5.11 are grayscale dilations, and the functions $h_R(x,y)$, $h_G(x,y)$, and $h_B(x,y)$ represent structuring elements with regions of support H_R, H_G, and H_B, respectively. Component-wise color erosion, opening, and closing are defined similarly. A component-wise morphological filtering operation can also be implemented in any color space other than RGB.

Because the component images are filtered separately with the component-wise filter, there is a possibility of altering the spectral composition of the image, e. g., the color balance and object boundaries. For example, there is a possibility with this approach that an object could be removed or enhanced in one or two of the R, G, and B components, but not in all of them. This effect near spatial edges in an image is referred to as "edge jitter" [46].

To illustrate this problem, the test image shown on the left in Figure 5.1 was opened using a component-wise morphological filter with a 15×15 square structuring element.[1] The output of the opening filter is shown on the right in Figure 5.1. In the opened image the two smallest squares have changed hue. This effect would be unacceptable for many applications. For example, in object recognition color and shape both play important roles. If the component-wise filter is applied to a color image as a step in object recognition, then the effect shown in Figure 5.1, in which the spectral composition of image objects has been altered, may produce errors in further processing of the color image obtained from the filter.

[1]Due to the prohibitive cost of color printing and the inherent distortions associated with the halftone printing process, only luminance images will be shown in this thesis. The corresponding color images are available via anonymous ftp to skynet.ecn.purdue.edu (Internet address 128.46.154.48) in the directory /pub/dist/delp/color-morph.

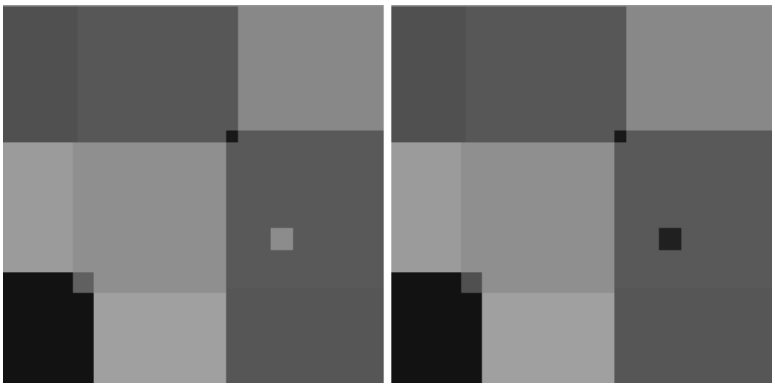


Fig. 5.1. Left: original image; right: result of component-wise color opening of original image with 15x15 square structuring element.

A different way to examine the problem of color morphology is to treat the color at each pixel as a vector. To motivate this approach, consider a color image with only two colors, representing an object and a background region. Let $\mathbf{f}$ be a color image which consists of the two colors $\mathbf{f}_1 = [R_1, G_1, B_1]^{\mathrm{T}}$ and $\mathbf{f}_2 = [R_2, G_2, B_2]^{\mathrm{T}}$. One way to analyze geometrical features of this image, which is perhaps more natural than the component-wise approach, is to view the pixels which represent the object as a set, and the pixels which represent the background as the complement of this set, as is done with a binary image. The justification for this approach is that both binary images and two-color color images represent scenes with two regions, and the geometrical information in both cases is determined by which pixels belong to the object and which belong to the background. Thus, we define the sets X and X^c as

$$X = \{(x, y) : \mathbf{f}(x, y) = \mathbf{f}_1\} \tag{5.12}$$

$$X^c = \{(x, y) : \mathbf{f}(x, y) = \mathbf{f}_2\} \tag{5.13}$$

In this case color dilation, erosion, opening, and closing would be defined the same way as binary dilation, erosion, opening, and closing, with X as the image foreground and X^c as the background. In this vector approach the data at each pixel in the filtered image represent either an object pixel or a background pixel, rather than, for example, the red component from the object and the blue and green components from the background, as is possible with the component-wise approach.

To extend the vector approach to color images with more than two colors, it is necessary to define an order relation which orders the colors as vectors, rather than ordering the individual components. This will be done using reduced ordering. In reduced ordering each multivariate observation is reduced to a single value, which is a function of the component values for that observation, with the multivariate samples ranked according to this single value [49]. To illustrate this type of ordering, let $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ be a collection of multivariate samples, where each $\mathbf{x}_i$ is a vector in $\mathbf{R}^p$. The n samples are to be ordered using a reduced ordering scheme. The first step is to map each $\mathbf{x}_i$ to a scalar value $d_i = d(\mathbf{x}_i)$ where $d : \mathbf{R}^p \rightarrow \mathbf{R}$. After d_i has been

obtained for each i, the vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ are ordered based on $d_1, d_2, \ldots, d_n$ as follows:

$$\mathbf{x}_{(1)} \leq \mathbf{x}_{(2)} \leq \cdots \leq \mathbf{x}_{(n)} \tag{5.14}$$

where $\mathbf{x}_{(r)}$ is the vector with corresponding scalar value $d_{(r)}$, and $d_{(r)}$ is the rth smallest element of the set $\{d_1, d_2, \ldots, d_n\}$.

We now use reduced ordering as described above to define new vector morphological filtering operations for color images. The structuring element for the vector morphological operations defined here is the set H, and the scalar-valued function used for the reduced ordering is $d : \mathbf{R}^3 \rightarrow \mathbf{R}$. The operation of vector dilation is represented by the symbol $\oplus_v$. The value of the vector dilation of $\mathbf{f}$ by H at the point (x, y) is defined as

$$(\mathbf{f} \oplus_v H)(x, y) = \mathbf{a} \tag{5.15}$$

where

$$\mathbf{a} \in \{\mathbf{f}(r, s) : (r, s) \in H_{(x,y)}\} \tag{5.16}$$

and

$$d(\mathbf{a}) \geq d(\mathbf{f}(r, s)) \; \forall \, (r, s) \in H_{(x,y)} \tag{5.17}$$

Similarly, vector erosion is represented by the symbol $\ominus_v$, and the value of the vector erosion of $\mathbf{f}$ by H at the point (x, y) is defined as

$$(\mathbf{f} \ominus_v H)(x, y) = \mathbf{b} \tag{5.18}$$

where

$$\mathbf{b} \in \{\mathbf{f}(r, s) : (r, s) \in H_{(x,y)}\} \tag{5.19}$$

and

$$d(\mathbf{b}) \leq d(\mathbf{f}(r, s)) \; \forall \, (r, s) \in H_{(x,y)} \tag{5.20}$$

Vector opening is defined as the cascade of vector erosion and vector dilation, and vector closing is defined as the cascade of vector dilation and vector erosion. With the above definitions for vector morphological operations we must impose the restriction

that the set H is a finite set, because if H is not finite then it is possible that no value of $\mathbf{a}$ satisfies Equations 5.16 and 5.17 or no value of $\mathbf{b}$ satisfies Equations 5.19 and 5.20.

With these definitions the output vector at each point in the image is, by definition, one of the vectors in the original image, so there is no possibility of introducing new color vectors into the image. It is possible that two different color values of $\mathbf{a}$ could satisfy Equations 5.16 and 5.17 or two different color values of $\mathbf{b}$ could satisfy Equations 5.19 and 5.20. In this case the output of the vector filter can be chosen based on positions in the structuring element window.

Often the metric used to perform reduced ordering is some type of distance metric [49]. The output of the vector filter will depend not only on the input image and the structuring element, but also on the scalar-valued function used to perform the reduced ordering. For many image processing applications it might make sense to use a characteristic of the human visual system, such as luminance, as a metric for reduced ordering. We investigate different choices for the scalar-valued function in later sections of the chapter.

Since mathematical morphology is based on set theory, it is important to investigate the new vector morphological operators defined above in terms of set operations. Serra discusses the need to analyze grayscale morphological operations not only in terms of transformations of grayscale functions, but also in terms of set transformations on the cross sections of those grayscale functions [19]. In fact, grayscale morphology was originally developed by representing functions as sets, using either cross sections or the umbra representation of a function, and applying binary morphological operations to those sets [19]. Similarly, if we define the concept of cross-sections of a color image, then vector morphological operations can be analyzed in terms of set transformations on these cross-sections.

We first review the set-theoretic analysis of FSP morphological operations for grayscale images. For a grayscale image f (discrete-space or continuous-space) the

set

$$X_t(f) = \{(x,y) \in D : f(x,y) \geq t\}, t \in V \tag{5.21}$$

where $V \subseteq \mathbf{R}$ or $\mathbf{Z}$ is the range of the function f, is known as the cross section of f at level t [19, 21]. If the function f is upper semi-continuous then the image can be reconstructed from its cross sections by

$$f(x,y) = \sup\{t \in V : (x,y) \in X_t(f)\} \tag{5.22}$$

The following equations show the relationship between grayscale dilation and erosion of the image f by the set H and binary dilation and erosion of the cross sections of f by H [21]:

$$X_t(f \oplus H) = X_t(f) \oplus H \iff (f \oplus H)(x,y) = \sup_{(r,s) \in H_{(x,y)}} \{f(r,s)\} \tag{5.23}$$

$$X_t(f \ominus H) = X_t(f) \ominus H \iff (f \ominus H)(x,y) = \inf_{(r,s) \in H_{(x,y)}} \{f(r,s)\} \tag{5.24}$$

Thus, with grayscale dilation and erosion defined as in Equations 5.5 and 5.6, the cross section at level t of $f \oplus H$ is equal to the binary dilation of the cross section at level t of f by the set H and the cross section at level t of $f \ominus H$ is equal to the binary erosion of the cross section at level t of f by the set H.

To derive analogous equations relating vector morphological operations to binary morphological operations, we propose the following definition for cross sections of a color image: The set

$$X_t(\mathbf{f}) = \{(x,y) : d(\mathbf{f}(x,y)) \geq t\} \tag{5.25}$$

is the cross section of the color image $\mathbf{f}$ at level t with respect to the function $d : \mathbf{R}^3 \rightarrow \mathbf{R}$. Reconstruction of a color image $\mathbf{f}$ from its cross sections is possible only if each value of d has a unique color vector associated with it. In this case, reconstruction of $\mathbf{f}$ is given by

$$d(\mathbf{f}(x,y)) = sup\{t \in V : (x,y) \in X_t(\mathbf{f})\} \tag{5.26}$$

Then, $\mathbf{f}(x,y)$ is the color vector corresponding to $d(\mathbf{f}(x,y))$.

The following proposition relates the vector operations of Equations 5.15 through 5.20 to the operations of binary morphology.

Proposition 1 Subject to the constraints

$$(\mathbf{f} \oplus_v H)(x,y) \in \{\mathbf{f}(r,s) : (r,s) \in H_{(x,y)}\} \tag{5.27}$$

and

$$(\mathbf{f} \ominus_v H)(x,y) \in \{\mathbf{f}(r,s) : (r,s) \in H_{(x,y)}\} \tag{5.28}$$

the following equations provide necessary and sufficient conditions for the vector dilation and erosion of f to be equivalent to binary dilations and erosions of the cross-sections of f:

$$X_t(\mathbf{f} \oplus_v H) = X_t(\mathbf{f}) \oplus H \iff d((\mathbf{f} \oplus_v H)(x,y)) \geq d(\mathbf{f}(r,s)) \; \forall (r,s) \in H_{(x,y)} \tag{5.29}$$

and

$$X_t(\mathbf{f} \ominus_v H) = X_t(\mathbf{f}) \ominus H \iff d((\mathbf{f} \ominus_v H)(x,y)) \leq d(\mathbf{f}(r,s)) \; \forall (r,s) \in H_{(x,y)} \tag{5.30}$$

Proof. First, we assume that $X_t(\mathbf{f} \oplus_v H) = X_t(\mathbf{f}) \oplus H$ and prove the right-hand side of Equation 5.29. Using this assumption, we have

$$\{(x,y) : d((\mathbf{f} \oplus_v H)(x,y)) \geq t\} = \{(x,y) : H_{(x,y)} \cap X_t(\mathbf{f}) \neq \emptyset\} \tag{5.31}$$

$$= \{(x,y) : \exists (r,s) \in H_{(x,y)} \ni d(\mathbf{f}(r,s)) \geq t\} \tag{5.32}$$

Hence,

$$d((\mathbf{f} \oplus_v H)(x,y)) \geq t \iff \exists (r,s) \in H_{(x,y)} \ni d(\mathbf{f}(r,s)) \geq t \tag{5.33}$$

Let

$$t_{max} = \max_{(r,s) \in H_{(x,y)}} \{d(\mathbf{f}(r,s))\} \tag{5.34}$$

Then

$$d((\mathbf{f} \oplus_v H)(x,y)) \geq t_{max} \geq d(\mathbf{f}(r,s)) \; \forall (r,s) \in H_{(x,y)} \tag{5.35}$$

Thus,

$$d((\mathbf{f} \oplus_v H)(x,y)) \geq d(\mathbf{f}(r,s)) \; \forall (r,s) \in H_{(x,y)} \tag{5.36}$$

Now we assume that $d((\mathbf{f} \oplus_v H)(x,y)) \geq d(\mathbf{f}(r,s))\ \forall (r,s) \in H_{(x,y)}$ and prove the left-hand side of Equation 5.29. Let $(x,y) \in X_t(\mathbf{f}) \oplus H$. Then

$$H_{(x,y)} \cap X_t(\mathbf{f}) = H_{(x,y)} \cap \{(r,s) : d(\mathbf{f}(r,s)) \geq t\} \neq \emptyset \tag{5.37}$$

which means that $\exists (r,s) \in H_{(x,y)}$ such that $d(\mathbf{f}(r,s)) \geq t$, and hence,

$$d((\mathbf{f} \oplus_v H)(x,y)) \geq t \tag{5.38}$$

by the assumption that the right-hand side of Equation 5.29 holds. Thus, $(x,y) \in X_t(\mathbf{f} \oplus_v H)$, and

$$X_t(\mathbf{f}) \oplus H \subseteq X_t(\mathbf{f} \oplus_v H) \tag{5.39}$$

Now, let $(x,y) \in X_t(\mathbf{f} \oplus_v H)$. Then

$$d((\mathbf{f} \oplus_v H)(x,y)) \geq t \tag{5.40}$$

which means that

$$\exists (r,s) \in H_{(x,y)} \ni d(\mathbf{f}(r,s)) \geq t \tag{5.41}$$

because of the constraint of Equation 5.27. Hence,

$$H_{(x,y)} \cap X_t(\mathbf{f}) \neq \emptyset \tag{5.42}$$

which implies that

$$(x,y) \in \{(r,s) : H_{(r,s)} \cap X_t(\mathbf{f}) \neq \emptyset\} \tag{5.43}$$

Thus $(x,y) \in X_t(\mathbf{f}) \oplus H$, and

$$X_t(\mathbf{f} \oplus_v H) \subseteq X_t(\mathbf{f}) \oplus H \tag{5.44}$$

Next, we assume that $X_t(\mathbf{f} \ominus_v H) = X_t(\mathbf{f}) \ominus H$ and prove the right-hand side of Equation 5.30. We have

$$\{(x,y) : d((\mathbf{f} \ominus_v H)(x,y)) \geq t\} = \{(x,y) : H_{(x,y)} \subseteq X_t(\mathbf{f})\} \tag{5.45}$$

$$= \{(x,y) : d(\mathbf{f}(r,s)) \geq t \forall (r,s) \in H_{(x,y)}\} \tag{5.46}$$

Hence,

$$d((\mathbf{f} \ominus_v H)(x,y)) \geq t \Longleftrightarrow d(\mathbf{f}(r,s)) \geq t \forall (r,s) \in H_{(x,y)} \tag{5.47}$$

Then, for any $t_0 \in \mathbf{R}$,

$$d((\mathbf{f} \ominus_v H)(x,y)) = t_0 \Longrightarrow d(\mathbf{f}(r,s)) \geq t_0 \forall (r,s) \in H_{(x,y)} \tag{5.48}$$

and, thus,

$$d((\mathbf{f} \ominus_v H)(x,y)) \leq d(\mathbf{f}(r,s)) \; \forall (r,s) \in H_{(x,y)} \tag{5.49}$$

Now we assume that $d((\mathbf{f} \ominus_v H)(x,y)) \leq d(\mathbf{f}(r,s)) \; \forall (r,s) \in H_{(x,y)}$ and prove the left-hand side of Equation 5.30. Let $(x,y) \in X_t(\mathbf{f}) \ominus H$. Then $H_{(x,y)} \subseteq X_t(\mathbf{f})$, which implies that for every $(r,s) \in H_{(x,y)}$, $d(\mathbf{f}(r,s)) \geq t$, and hence, using the constraint of Equation 5.28,

$$d((\mathbf{f} \ominus_v H)(x,y)) \geq t \tag{5.50}$$

Thus $(x,y) \in X_t(\mathbf{f} \ominus_v H)$. Now, let $(x,y) \in X_t(\mathbf{f} \ominus_v H)$. Then

$$d((\mathbf{f} \ominus_v H)(x,y)) \geq t \tag{5.51}$$

and hence,

$$\forall (r,s) \in H_{(x,y)} d(\mathbf{f}(r,s)) \geq t \tag{5.52}$$

which means that

$$H_{(x,y)} \subseteq X_t(\mathbf{f}) \tag{5.53}$$

and thus, $(x,y) \in X_t(\mathbf{f}) \ominus H$. □

Proposition 1 is important because it provides the following interpretation of the vector morphological operations defined in Equations 5.15 through 5.20: Vector dilation of $\mathbf{f}$ by H is equivalent to thresholding $\mathbf{f}$ at each level $t \in \mathbf{R}$ to obtain a set containing all pixels representing objects with d value greater than or equal to t and performing a binary dilation by H on this set, and then recombining the dilated sets to form the output color image.

In the following sections we investigate the use of the component-wise and vector morphological filters for multiscale image analysis and noise suppression. We present experimental results comparing the performance of the two types of filters.

5.4 Multiscale Image Analysis

The representation of image objects at multiple scales is important in many computer vision and image processing applications. A multiscale representation of an image consists of a set of images which are derived by filtering the original image with a family of filters of varying scale, or spatial extent. A multiresolution representation is obtained from a multiscale representation by subsampling the image at each level in the multiscale representation, where images at larger scales are subsampled by a greater factor than images at smaller scales.[2]

Multiscale and multiresolution image representations have become important for several reasons. First, in analyzing an image for computer vision applications, the appearance of an object in the image depends not only on the properties of the object itself, but also on the resolution of the image. Since it is often not possible to define *a priori* an optimal resolution for analyzing images, scale-space methods and multiscale representations have been proposed for computer vision problems, such as stereo matching, object recognition, and image segmentation [60, 61, 62, 63, 58, 12].

Multiresolution image representations are also important because of their usefulness in the area of image coding [30, 64, 65, 37]. In a multiresolution representation the lowest resolution image can be encoded at a low bit rate because it contains a much smaller number of pixels than the original image. Also, at each resolution level the difference between the image at that resolution and the image at the next resolution is obtained, and these difference images can be encoded at reduced bit rates due to their low entropies [30].

A third reason for processing images at multiple resolutions is that this can lead to reduced computational complexity in comparison with algorithms that process images at a single scale [66, 67]. By processing an image first at a low resolution, where there

[2]This terminology is taken from the computer vision literature [57, 58]. It should be noted that this is different from the terminology used in the signal processing literature [59], in which filtering a signal to change its frequency content affects the resolution of the signal and changing the sampling rate (for a discrete-time signal) affects the scale of the signal.

are fewer pixels to process, and then using higher resolution information to refine the solution, significant savings in computational complexity can be obtained.

The type of filtering used to obtain a multiscale representation of an image determines the properties of the resultant multiscale representation (and thus the multiresolution representation, also). Linear filters, such as wavelet filters, quadrature mirror filters, and Gaussian filters, are often used [68, 65, 30]. Multiscale image analysis using morphological filtering has been suggested for applications such as shape-size distributions, image compression, and edge enhancement [31, 57, 58, 37, 36, 69]. Although the linear filtering approach has the advantage that a multiscale representation obtained using linear filtering can be viewed as a space-frequency representation and its frequency content can be studied using Fourier analysis, the morphological filtering approach is more appropriate for quantifying shape information at different scales [57, 58].

In this section we compare the component-wise and vector approaches for obtaining multiscale representations of color images. We concentrate in particular on multiscale FSP operations. The multiscale FSP opening and closing of a discrete-space grayscale image f by the finite, connected set $H \subseteq \mathbf{Z}^2$ at scale n are defined as [57]

$$f \circ nH = (f \ominus nH) \oplus nH \tag{5.54}$$

and

$$f \bullet nH = (f \oplus nH) \ominus nH \tag{5.55}$$

for $n = 0, 1, 2, \ldots$, where

$$nH = H \oplus H \oplus \cdots \oplus H \quad (n-1 \ \ dilations) \tag{5.56}$$

The multiscale FSP opening and closing of a continuous-space grayscale image f by the compact, connected set $H \subseteq \mathbf{R}^2$ at scale r are defined as

$$f \circ rH = (f \ominus rH) \oplus rH \tag{5.57}$$

and

$$f \bullet rH = (f \oplus rH) \ominus rH \tag{5.58}$$

for $r \in \mathbf{R}$, $r \geq 0$, where

$$rH = \{rh : h \in H\} \tag{5.59}$$

We can perform a multiscale opening or closing on a color image by replacing the grayscale erosions and dilations of Equations 5.54, 5.55, 5.57, and 5.58 by either component-wise color dilations and erosions or vector dilations and erosions. From the example shown in Figure 5.1, it is clear that the component-wise filter can have some unexpected effects in terms of spectral filtering. This raises the question of how to include spectral information in a multiscale color image representation. The vector filter has an advantage over the component-wise filter in addressing this issue because it will not alter the spectral composition of image objects. To illustrate the importance of this advantage for multiscale image analysis, Figure 5.2 shows the component-wise multiscale opening in RGB space of a color image, and Figure 5.3 shows the vector multiscale opening of the same image, using luminance as the scalar-valued function for the reduced ordering.

The structuring element H used to create these multiscale representations is shown in Figure 5.4.

The vector filter provides a more appropriate multiscale representation of the image than the component-wise filter. The component-wise multiscale representation would be particularly unsatisfactory if the image was to be represented by its multiscale edges [70]. In this case a color edge detector applied at each scale would not produce an accurate representation of the image structure. Although the component-wise filter could be applied in a color space other than RGB, the undesirable effect illustrated in Figure 5.2 could still occur, since the individual component images would still be filtered independently.

Since the output of the vector filter depends on the scalar-valued function used for reduced ordering, the selection of this function provides flexibility in incorporating spectral information into the multiscale image representation. For example, certain

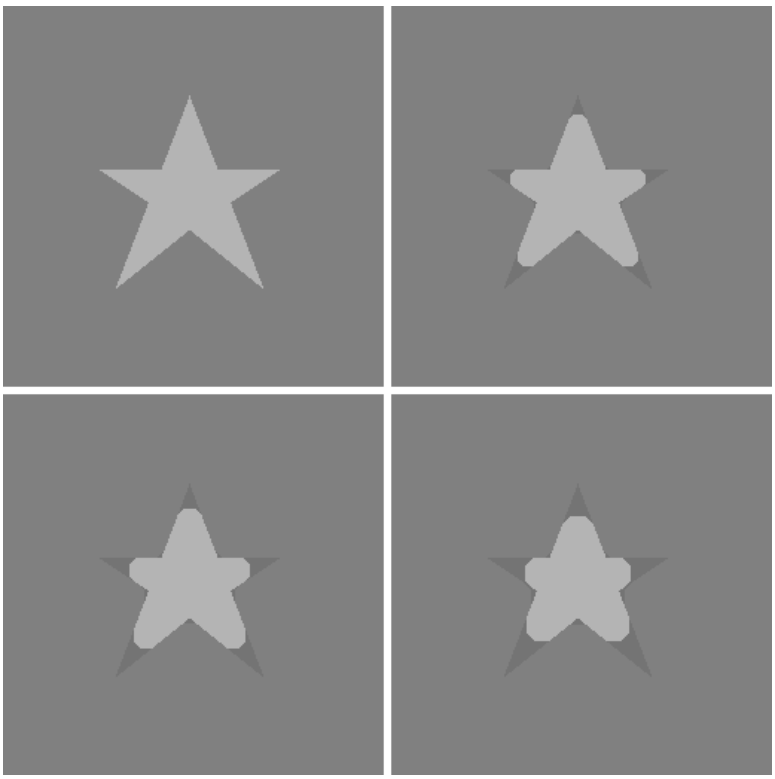


Fig. 5.2. Component-wise multiscale opening. Upper left: original image; upper right: result of component-wise opening, $n = 3$; lower left: result of component-wise opening, $n = 4$; lower right: result of component-wise opening, $n = 5$.

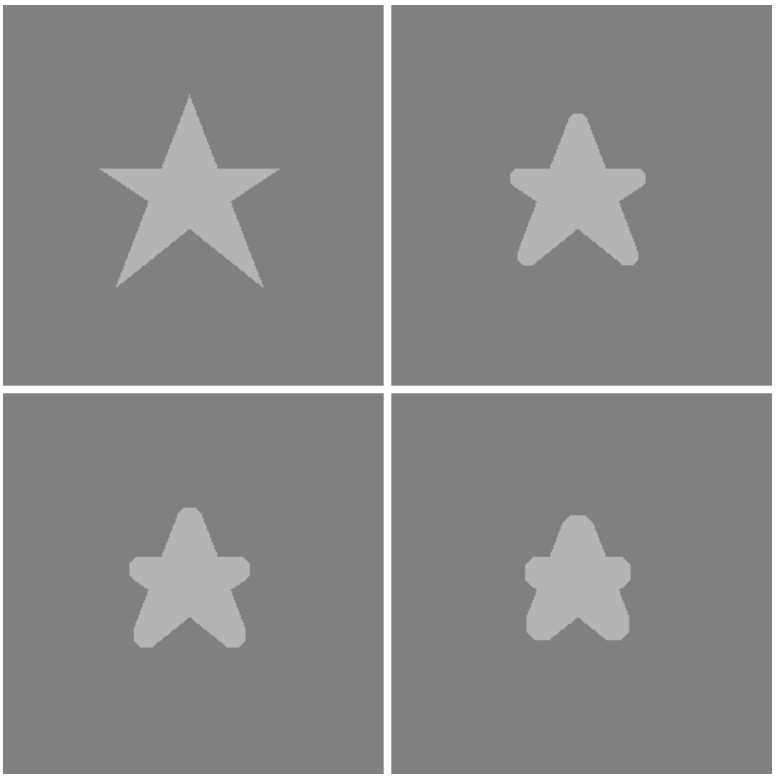


Fig. 5.3. Vector multiscale opening with $d(\mathbf{f}(x,y)) = f_Y(x,y)$ (luminance image). Upper left: original image; upper right: result of vector opening, $n = 3$; lower left: result of vector opening, $n = 4$; lower right: result of vector opening, $n = 5$.

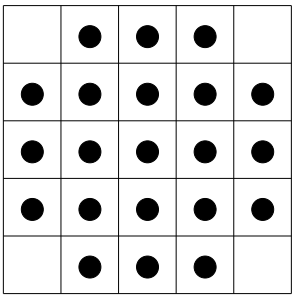


Fig. 5.4. Structuring element used for multiscale smoothing experiments.

linear combinations of the tristimulus values can be used. This would be written as

$$d(\mathbf{f}(x,y)) = a_R f_R(x,y) + a_G f_G(x,y) + a_B f_B(x,y) \tag{5.60}$$

if the image is filtered in the RGB color space. For the case $a_R = 0.299$, $a_G = 0.587$, and $a_B = 0.114$, $d(\mathbf{f}(x,y))$ becomes the luminance image. Multiscale opening in this case would suppress bright objects at each scale. A multiscale representation obtained using the luminance image as the scalar-valued function and the structuring element shown in Figure 5.4 is shown in Figure 5.5.

The values of a_R, a_G, and a_B can also be selected to enhance or suppress specific colors. For example, if $a_R = 1$, $a_G = 0$, and $a_B = 0$, then the effect of a multiscale opening would be to suppress objects with high red content. Similarly, if $a_R = 0$, $a_G = 1$, and $a_B = 0$ then green objects would be suppressed by a multiscale opening and if $a_R = 0$, $a_G = 0$, and $a_B = 1$ then blue objects would be suppressed. This would lead to a family of images parameterized by shape, size, and "color", which could be useful for an application such as object recognition. Figure 5.6 shows a multiscale representation (using the same original image and structuring element as those used for Figure 5.5) obtained using the red image as the scalar-valued function. It can be seen that the difference in the values of a_R, a_G, and a_B used for the two multiscale representations shown in Figures 5.5 and 5.6 strongly influences the resultant representation.

5.5 Image Enhancement

In this section we investigate the use of color morphology for noise suppression. First, the method used to simulate noisy color images is discussed, and then experimental results using the component-wise and vector filters are presented.

5.5.1 Simulation of noisy color images

The noise model used here is different from the one used in [47, 71], in which three independent random variables were added to each pixel, one to each of the three spectral components. Our model allows us to vary the levels of noise correlation

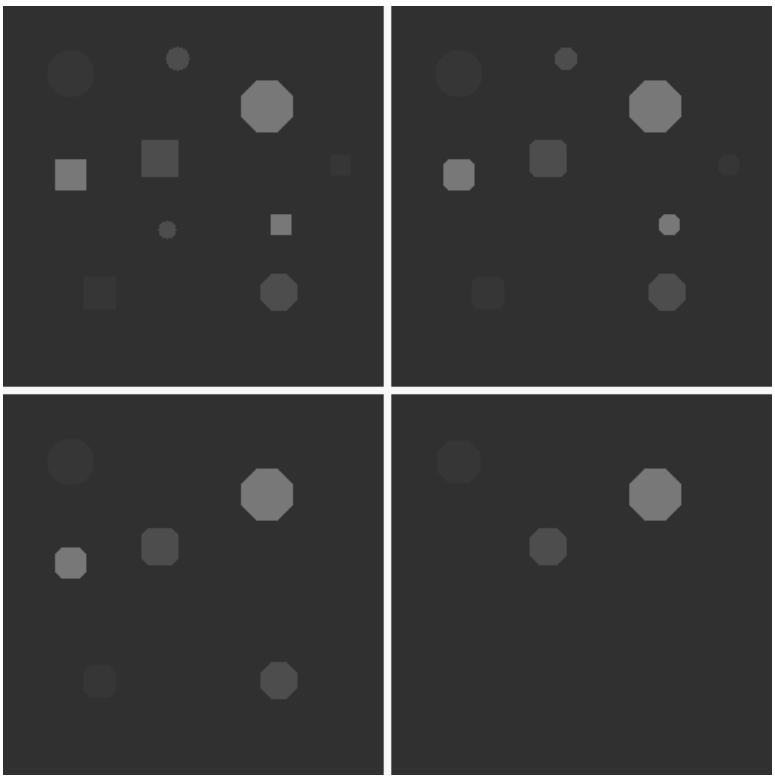


Fig. 5.5. Vector multiscale opening with $d(\mathbf{f}(x,y)) = f_Y(x,y)$. Upper left: original image; upper right: result of vector opening, $n = 3$; lower left: result of vector opening, $n = 4$; lower right: result of vector opening, $n = 6$.

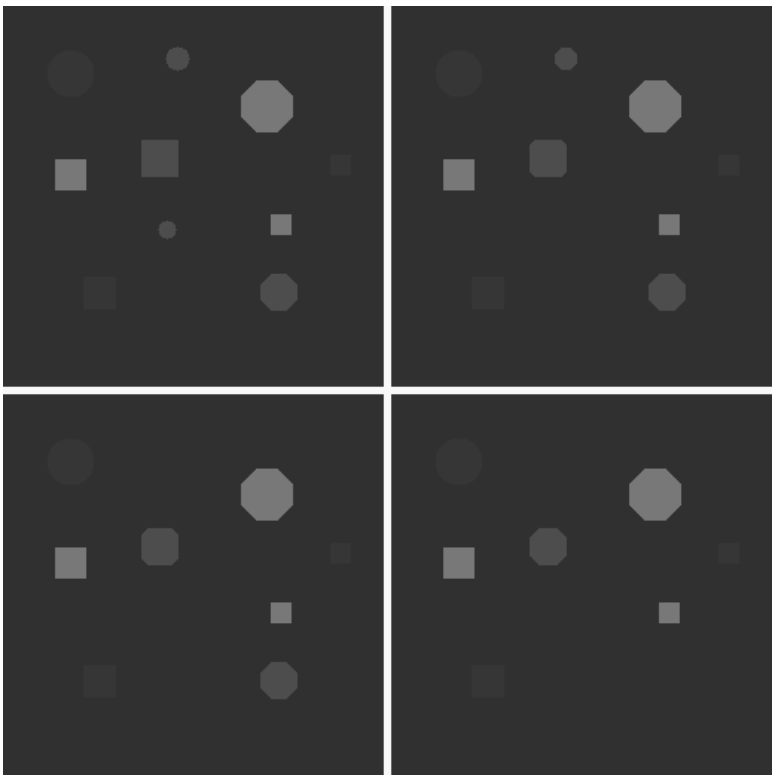


Fig. 5.6. Vector multiscale opening with $d(\mathbf{f}(x,y)) = f_R(x,y)$. Upper left: original image; upper right: result of vector opening, $n = 3$; lower left: result of vector opening, $n = 4$; lower right: result of vector opening, $n = 6$.

among the three spectral component images, using a method which is less ad hoc than that used in [48].

There are several reasons that it is important to consider spectrally correlated noise. First, the original image signal (the input to the imaging system) may contain noise which is correlated across the spectral bands. Second, if the three component images are obtained using the same sensor array with different color filters , then at each pixel the three vector components are obtained using the same sensor(s), which could lead to spectrally correlated noise. A third reason to consider spectrally correlated noise is that certain transformations of the image data will result in noise which is spectrally correlated. For example, consider an image in a color space $\mathrm{X_1X_2X_3}$ that has been transformed from another color space $\mathrm{Y_1Y_2Y_3}$. If the image was corrupted with additive noise while in the $\mathrm{Y_1Y_2Y_3}$ space, then at a given pixel the noisy $\mathrm{Y_1Y_2Y_3}$ color vector would be given by $[Y_1 + n_{Y_1}, Y_2 + n_{Y_2}, Y_3 + n_{Y_3}]^{\mathrm{T}}$, where Y_1, Y_2, and Y_3 are the tristimulus values at the given pixel, and n_{Y_1}, n_{Y_2}, and n_{Y_3} are random variables which represent the additive noise at the given pixel. If the transformation matrix from $\mathrm{Y_1Y_2Y_3}$ space to $\mathrm{X_1X_2X_3}$ space has elements a_{ij}, $i = 1, 2, 3$; $j = 1, 2, 3$, then the transformed color vector is

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} Y_1 + n_{Y_1} \\ Y_2 + n_{Y_2} \\ Y_3 + n_{Y_3} \end{bmatrix} = \begin{bmatrix} X_1 + n_{X_1} \\ X_2 + n_{X_2} \\ X_3 + n_{X_3} \end{bmatrix} \tag{5.61}$$

where X_1, X_2, and X_3 are the $\mathrm{X_1X_2X_3}$ tristimulus values at the given pixel for the non-noisy image, and $n_{X_1} = a_{11}n_{Y_1} + a_{12}n_{Y_2} + a_{13}n_{Y_3}$, $n_{X_2} = a_{21}n_{Y_1} + a_{22}n_{Y_2} + a_{23}n_{Y_3}$, and $n_{X_3} = a_{31}n_{Y_1} + a_{32}n_{Y_2} + a_{33}n_{Y_3}$ are random variables which represent the additive noise in the $\mathrm{X_1}$, $\mathrm{X_2}$, and $\mathrm{X_3}$ image planes, respectively. Hence, in this situation, the noise correlation between spectral planes would in general be nonzero.

To simulate noisy color images with spectrally correlated noise, the classical process used to whiten correlated random variables can be reversed [72]. At each pixel, a vector $\mathbf{Z}$, of three uncorrelated, unit-variance, random samples is generated. These samples are then mapped, through a linear transformation, to a vector $\mathbf{N}$ with three

samples with variances and correlation coefficients which can be selected as desired. The first step is to specify the desired covariance matrix for the three random variables representing the noise. If the noise is to be added to an image which is in the RGB color space, then this matrix can be written as:

$$\Sigma = \begin{bmatrix} \sigma_R^2 & \rho_{RG}\sigma_R\sigma_G & \rho_{RB}\sigma_R\sigma_B \\ \rho_{RG}\sigma_R\sigma_G & \sigma_G^2 & \rho_{GB}\sigma_G\sigma_B \\ \rho_{RB}\sigma_R\sigma_B & \rho_{GB}\sigma_G\sigma_B & \sigma_B^2 \end{bmatrix} \tag{5.62}$$

where σ_R^2, σ_G^2, and σ_B^2 are the variances of the red, green, and blue noise components, respectively, and ρ_{RG}, ρ_{RB}, and ρ_{GB} are the spectral correlation coefficients for the red and green, red and blue, and green and blue noise components, respectively. It should be noted that the noise could be added in a color space other than RGB color space.

The transformation from $\mathbf{Z}$ to $\mathbf{N}$, where $\mathbf{N}$ is a three-dimensional random vector with covariance matrix Σ, is needed. To find this transformation, the transformation from $\mathbf{N}$ to $\mathbf{Z}$ is first found. This is the whitening transformation, which converts correlated samples to uncorrelated, unit-variance samples. It has the form

$$\mathbf{Z} = \Lambda^{-\frac{1}{2}}\Phi^{\mathrm{T}}\mathbf{N} \tag{5.63}$$

where Λ and Φ are the eigenvalue and eigenvector matrices of Σ. This means that $\mathbf{N}$ can be obtained as

$$\mathbf{N} = \Phi^{\mathrm{T}}\Lambda^{\frac{1}{2}}\mathbf{Z} \tag{5.64}$$

Using the method described above, the spectral correlation of the noise can be specified. To simulate the noise spatially, an ϵ-mixture of Gaussian noise is added to the R, G, and B components of a color image [38]. The probability density function of this noise is given by

$$f_{\mathbf{N}}(\mathbf{n}) = \epsilon\frac{1}{(2\pi)^{3/2}|\Sigma_1|^{1/2}}\exp[-\frac{1}{2}\mathbf{n}^{\mathrm{T}}\Sigma_1^{-1}\mathbf{n}]+$$

$$(1-\epsilon)\frac{1}{(2\pi)^{3/2}|\Sigma_2|^{1/2}}\exp[-\frac{1}{2}\mathbf{n}^{\mathrm{T}}\Sigma_2^{-1}\mathbf{n}], \tag{5.65}$$

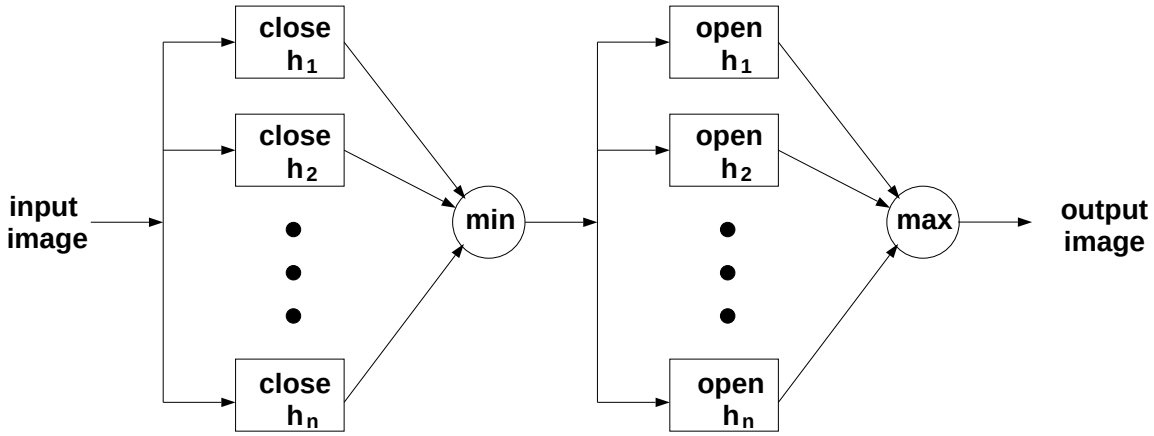


Fig. 5.7. Block diagram of 2DCO filter for grayscale images.

where $\mathbf{N} = [N_1, N_2, N_3]^{\mathrm{T}}$ is the random noise vector, and Σ_1 and Σ_2 are covariance matrices corresponding to impulsive and non-impulsive noise, respectively. Thus, with probability ϵ, a noise vector with covariance matrix Σ_1 is added at a given pixel. The variances $\sigma^2_{R_1}, \sigma^2_{G_1}$, and $\sigma^2_{B_1}$ for this noise are large. This allows spatially impulsive noise to be simulated. With probability $1 - \epsilon$, a noise vector with covariance matrix Σ_2 is added. The variances $\sigma^2_{R_2}, \sigma^2_{G_2}$, and $\sigma^2_{B_2}$ for this noise are small. This allows for simulation of non-impulsive noise.

5.5.2 Experimental results

This section presents results of experiments involving noise suppression in color images. The grayscale morphological filter used for all of these experiments was the 2DCO filter introduced by Stevenson and Arce [51] and extended by Song and Delp (See Figure 5.7) [52, 53]. This filter consists of a cascade of two stages each of which consists of multiple morphological operators of one type (opening or closing). In each stage multiple operators are applied to the input image using different structuring elements. The output of each stage is selected according to the morphological operator used: maximum for the output of the opening stage, minimum for the output of the closing stage. If the first stage consists of closing operators and the second stage consists of opening operators, then for a grayscale input image f and a set of

structuring elments $\{h_1, h_2, \ldots, h_n\}$, the output y of the first stage is given by

$$y = \min\{f \bullet h_1, f \bullet h_2, \ldots, f \bullet h_n\} \tag{5.66}$$

The output z of the second stage is given by

$$z = \max\{y \circ h_1, y \circ h_2, \ldots, y \circ h_n\} \tag{5.67}$$

The purpose of each stage is to preserve the geometrical features in the image that match any one of the given structuring elements. For the experiments presented in this section, four structuring elements consisting of lines with different directions were used [51].

The 2DCO component-wise filter consists of applying the grayscale 2DCO filter to each of the three component images independently. To reduce computational complexity it is also possible to filter only the luminance image in the YIQ space. Hunt and Kubler suggest filtering only the luminance image as a simple approach for deblurring color imagery [39]. For morphological filtering this approach is equivalent to component-wise filtering as in Equation 5.11 with $H_I = H_Q = \{(0,0)\}$ and $h_I(0,0) = h_Q(0,0) = 0$.

The 2DCO vector filter is defined as a two-stage filter where the output $\mathbf{g}(x,y)$ of the first stage with input $\mathbf{f}(x,y)$ and structuring elements $\{H_1, H_2, \ldots, H_n\}$ is given by

$$\mathbf{g}(x,y) \in \{(\mathbf{f} \bullet_v H_1)(x,y), (\mathbf{f} \bullet_v H_2)(x,y), \ldots, (\mathbf{f} \bullet_v H_n)(x,y)\} \tag{5.68}$$

and

$$d(\mathbf{g}(x,y)) \leq d((\mathbf{f} \bullet_v H_i)(x,y)) \; \forall i = 1, 2, \ldots, n \tag{5.69}$$

The output $\mathbf{p}(x,y)$ of the second stage is given by

$$\mathbf{p}(x,y) \in \{(\mathbf{g} \circ_v H_1)(x,y), (\mathbf{g} \circ_v H_2)(x,y), \ldots, (\mathbf{g} \circ_v H_n)(x,y)\} \tag{5.70}$$

and

$$d(\mathbf{p}(x,y)) \geq d((\mathbf{g} \circ_v H_i)(x,y)) \; \forall i = 1, 2, \ldots, n \tag{5.71}$$

Fig. 5.8. Original image used for noise suppression experiments.

Noise was added to a color image in RGB space as discussed in the previous section, for two different values of spectral correlation between component images. Figure 5.8 shows the original image used for the noise suppression experiments. The noisy image shown in the upper left of Figure 5.9 has parameters $\epsilon = 0.05$; $\sigma_{R_1} = \sigma_{G_1} = \sigma_{B_1} = 100$, for the impulsive noise and $\sigma_{R_2} = \sigma_{G_2} = \sigma_{B_2} = 10$ for the non-impulsive noise; and $\rho_{RG} = \rho_{RB} = \rho_{GB} = 0.95$ for both impulsive and non-impulsive noise. The noise between component planes in this image is highly correlated.

The result of a component-wise 2DCO filter applied in RGB space is shown in Figure 5.9.[3] Figure 5.9 also shows the result of filtering the luminance image, Y, and converting the filtered Y and unfiltered I and Q to RGB space for display. This method performs as well as the component-wise filtering in RGB space. Because the noise is highly correlated across spectral bands, the chrominance images, which depend on the spectral composition, were not affected significantly by the noise.

Also shown in Figure 5.9 is the result of the application of the vector 2DCO filter in RGB space. The Euclidean norm

$$d(\mathbf{f}(x,y)) = [(f_R(x,y))^2 + (f_G(x,y))^2 + (f_B(x,y))^2]^{1/2} \tag{5.72}$$

[3]We are not aware of any generally accepted metric for measuring distortion in noisy color images. Therefore, evaluation of our results is subjective.

Fig. 5.9. Upper left: noisy version of original image, high spectral correlation; upper right: result of component-wise filter applied in RGB space; lower left: result of filtering only Y in YIQ space; lower right: result of vector filter.

was used as the metric for reduced ordering. This filtering method has results similar to the component-wise filtering in RGB space.

The second noisy image used in the experiments is shown in Figure 5.10. In this image, the values of ϵ and the variances are the same as those used for Figure 5.8. However, this image has $\rho_{RG} = \rho_{RB} = \rho_{GB} = 0.45$ for both types of noise. Thus, the spectral correlation of the noise is lower.

The result of component-wise filtering in RGB space is shown in the upper right of Figure 5.10. It can be seen that this filter has the same level of performance for this image as it did for the image of Figure 5.8.

The lower left image in Figure 5.10 shows the result of filtering only the luminance and converting to RGB for display. In this case, this method does not perform as well as the component-wise RGB filter. This method is less effective for noise which has lower spectral correlation.

Also shown in Figure 5.10 is the result of the application of the vector filter to the image with lower noise spectral correlation. This method also does not perform as well as the component-wise RGB filter for noise with lower spectral correlation. It is noted in [46] that the vector median filter also does not perform as well as the component-wise median filter when the noise in the different vector components is independent. In general, with spectrally uncorrelated noise the vector approach will not perform as well as the component-wise approach due to the restriction that the output of the vector filter must be one of the input vectors inside the filter window.

5.6 Conclusion

We have developed and compared two approaches to mathematical morphology for color images — the component-wise approach and the vector approach. It was shown that the component-wise filters can alter the spectral composition of image objects, producing undesirable artifacts. Multiscale color image representations illustrating these artifacts were shown. The new vector filters do not have this disadvantage, because all color vectors in the filtered image are selected from the color vectors in the

Fig. 5.10. Upper left: noisy version of original image, low spectral correlation; upper right: result of component-wise filter applied in RGB space; lower left: result of filtering only Y in YIQ space; lower right: result of vector filter.

original image. For noise suppression in color images the component-wise approach performed better than the vector approach for noise with low spectral correlation. For noise with high spectral correlation the component-wise and vector approaches had similar performance.

6. RATE-SCALABLE VIDEO CODING USING A ZEROTREE WAVELET APPROACH

In this chapter we describe a video coding technique which offers complete data rate scalability. This means that, from a single compressed bit stream, any target data rate can be met, and the best decompressed sequence for that rate can be obtained by simply discarding bits at the end of each group of pictures (GOP) in the bit stream. The video coding scheme is based on the embedded zerotree wavelet (EZW) algorithm for still image compression [73].

In the next section we describe the EZW algorithm. We then discuss the extension of this algorithm for video coding, and present experimental results for various data rates.

6.1 Embedded Zerotree Wavelet Coding

The development of wavelet representations of images has led to efficient image processing and computer vision algorithms. The discrete wavelet transform of an image provides a set of wavelet coefficients which represent the image at multiple scales [74, 68]. Fewer coefficients are required to represent the image at coarse scales than at finer scales. Thus, if the wavelet representation is viewed as a multi-level data structure, with the coarsest scale coefficients at the top level and the finest scale coefficients at the bottom level, then the structure can be viewed as a pyramid. Each node in the pyramid represents one wavelet coefficient.

Wavelet representations have formed a basis for several image compression algorithms [64, 75, 76, 77, 78, 79]. One of the key issues in the development of any wavelet based image compression algorithm is the quantization strategy. One approach is

to assume that coefficients at different levels of the pyramid are independent and quantize the coefficients separately [64, 65]. However, this approach does not exploit correlations between coefficients at different levels of the pyramid. Recently, a number of promising quantizers, known as zerotree quantizers, exploiting correlations between different levels of the pyramid, have been proposed [75, 76, 73]. The key idea in [76, 73] is to eliminate systematically the coefficients which do not convey much information about the image by assigning zero to all the wavelet coefficients at the finer scales corresponding to the same spatial location of a coefficient whose magnitude is below a given threshold. As a result, only a single symbol is transmitted for all the finer scale coefficients. The motivation for using this quantization technique is the assumption that if the energy of a node corresponding to a spatial region of the image is below a given threshold, then the energy of the coefficients at finer scales corresponding to the same region are also expected to be below the threshold.

The EZW algorithm, based on the zerotree quantizer presented in [73] for still image compression, forms the basis for the video coding scheme described in this chapter. We briefly describe the EZW algorithm here. Interested readers should see [73] for a more detailed discussion.

The first step in the EZW algorithm is to apply a discrete wavelet transform to the original image. This provides a pyramid representation of the image. Multiple passes are then made through the pyramid to quantize the coefficients in an embedded manner. During each pass, the coefficients are compared to a threshold value. Coefficients which are greater than the threshold are considered to be significant, and coefficients which are less than the threshold are insignificant. The significance map, which specifies at each node in the pyramid whether the coefficient at that node is significant for that pass or not, is then encoded. If a coefficient and all of its descendents are insignificant, then the subtree containing the coefficient and its descendents is said to be a zerotree. The entire zerotree can then be represented by only one encoded symbol.

After the significance map has been encoded during a given pass through the pyramid, the values of the coefficients must be quantized. All insignificant coefficients are quantized to zero. Significant coefficients are quantized using a successive approximation scheme which is similar to bit-plane encoding.

At the completion of a given pass through the pyramid, the threshold is decreased. The next pass through the pyramid is performed using this smaller threshold, so that there will be more significant coefficients as the algorithm progresses from pass to pass. This accounts for the embedded nature of the bit stream generated by the algorithm. As soon as the target data rate has been achieved, the algorithm stops.

6.2 Video Coding Using the EZW Algorithm

To extend the EZW algorithm for video coding, a three-dimensional wavelet decomposition can be used. Several techniques using three-dimensional subband coding of video have been proposed [23, 80, 81]. The idea is to filter groups of pictures (GOPs) both spatially and temporally to form a three-dimensional multiresolution representation of the GOP that represents multiple resolutions in both space and time. The coefficients representing the multiresolution representation are then quantized to reduce the data rate of the GOP.

For the results presented in this chapter, we used only intraframe coding. The filter coefficients given in [82] were used for the wavelet transform. The salesman sequence, of which frames 90, 91, and 92 are shown in Figure 6.1, was compressed using the EZW algorithm with intraframe coding at a rate of 3 Mbps. Figure 6.2 shows the results from decompressing frames 90, 91, and 92 at a rate of 256 kbps, Figure 6.3 shows the three frames decompressed at a rate of 800 kbps, and Figure 6.4 shows results from decompression at a rate of 2 Mbps. These results were all obtained from the same compressed sequence. Clearly, the results improve significantly as the data rate increases. It should be noted that using interframe as well as intraframe coding would improve the results substantially.

(a) (b) (c)

Fig. 6.1. (a): Frame 90 of original salesman sequence. (b): Frame 91 of original salesman sequence. (c): Frame 92 of original salesman sequence.

6.3 Conclusion

We have described a rate-scalable video coding scheme based on the EZW algorithm for still image compression. Results obtained by decompressing a single bitstream at several different data rates were presented. The addition of interframe coding to the algorithm would provide significantly improved results.

(a) (b) (c)

Fig. 6.2. (a): Frame 90 of salesman sequence at 256 kbps. (b): Frame 91 of salesman sequence at 256 kbps. (c): Frame 92 of salesman sequence at 256 kbps.

(a) (b) (c)

Fig. 6.3. (a): Frame 90 of salesman sequence at 800 kbps. (b): Frame 91 of salesman sequence at 800 kbps. (c): Frame 92 of salesman sequence at 800 kbps.

(a) (b) (c)

Fig. 6.4. (a): Frame 90 of salesman sequence at 2 Mbps. (b): Frame 91 of salesman sequence at 2 Mbps. (c): Frame 92 of salesman sequence at 2 Mbps.

7. CONCLUSION

We have presented and studied the EM/MPM algorithm and its multiresolution extension. In Chapter 2 we described the EM/MPM algorithm and presented experimental results demonstrating its performance. We also compared the performance of the EM/MPM algorithm to the deterministic EM/MPM algorithm proposed in [9]. The EM/MPM algorithm was shown experimentally to perform significantly better than the deterministic EM/MPM algorithm in terms of minimizing the number of misclassified pixels.

The multiresolution EM/MPM algorithm was presented in Chapter 3. Experimental results comparing the performance of the EM/MPM and multiresolution EM/MPM algorithms were presented. In general, it may not be possible to say which algorithm performs better, single-resolution or multiresolution, because the answer depends on the desired scale which the user is interested in emphasizing. Actually, the single-resolution algorithm described in Chapter 2 can be viewed as a special case of the multiresolution algorithm, with the number of levels equal to one. Thus, the multiresolution algorithm provides the ability to adjust the scale which is emphasized in the segmentation. The behavior of the multiresolution algorithm can be adjusted to emphasize a certain scale by adjusting the number of levels used in the multiresolution decomposition. Another method which could be used to weight certain scales more heavily than other scales is to change the criterion used for the optimization in the multiresolution MPM algorithm. This is one topic for future research.

In comparing the amount of computation required for convergence of the EM/MPM, deterministic EM/MPM, and multiresolution EM/MPM algorithms, we found that

the multiresolution algorithm required the fewest number of visits per pixel to converge. The deterministic EM/MPM algorithm does not necessarily converge more quickly than the EM/MPM algorithm, although faster convergence is supposed to be the advantage of the deterministic approach. It should be noted that the amount of computation required for convergence was measured by examining convergence of the parameter estimates. A more detailed study of the relative computational requirements of the three approaches would involve measuring the convergence of the MPM algorithm at each stage of the EM/MPM procedure. This is a topic for future work, although we believe the results will show the same trend as we found in our study.

In Chapter 4 a theoretical analysis of the general EM/MPM procedure was presented. The analysis is general enough to hold for both the single-resolution and multiresolution EM/MPM algorithms. We showed that the estimates of the marginal probabilities of the class labels obtained during a given stage of the EM/MPM or multiresolution EM/MPM algorithm converge with probability 1 to the true values of the class label probabilities evaluated with the model parameters equal to the current estimates of the model parameters, given the estimates of the model parameters obtained during the previous stage. This is important because these estimates are maximized to obtain the segmented image, and if the estimates are not close to the true values, then the segmentation algorithm cannot be expected to perform well. We also showed that the model parameter estimates resulting from the EM/MPM algorithm can be made arbitrarily close to the EM estimates of the model parameters with probability 1, if a sufficient number of iterations of the algorithm is performed. This means that the final estimates of the marginal probabilities of the class labels obtained using the EM/MPM procedure will be close to the true values of the marginal probabilities of the class labels, to the extent that the EM estimates of the model parameters are close to the true values of the model parameters. The analysis presented in Chapter 4 is important because it explains why the EM/MPM algorithm provides better experimental results than the deterministic EM/MPM algorithm, since there is no guarantee of convergence for the deterministic EM/MPM algorithm. The analysis

does not suggest that the multiresolution algorithm should perform better than the single-resolution algorithm. The multiresolution approach simply provides a better model for segmenting images.

The segmentation algorithms presented in this thesis assume that the number of classes is known *a priori*. The problem of automatically determining the number of classes is an important problem which should be addressed in future work.

In Chapter 5 we addressed the problem of applying mathematical morphology to color images. We investigated two approaches for color morphology, the component-wise approach and the vector approach. It was shown that the component-wise filters can alter the spectral composition of image objects, producing undesirable artifacts. Multiscale color image representations illustrating these artifacts were shown. The new vector filters do not have this disadvantage, because all color vectors in the filtered image are selected from the color vectors in the original image. For noise suppression in color images the component-wise approach performed better than the vector approach for noise with low spectral correlation. For noise with high spectral correlation the component-wise and vector approaches had similar performance.

Finally, in Chapter 6 we described a rate-scalable approach to video coding. We presented results obtained by decompressing a single bitstream at a number of different data rates. The results presented in Chapter 6 were obtained using only intraframe coding. The next step in the development of the algorithm is to replace the two-dimensional spatial wavelet transform with a three-dimensional spatial/temporal wavelet transform. The three-dimensional wavelet transform would then be applied to each GOP, and the bits in the encoded bitstream for each GOP would be generated in order of importance to provide an embedded nature to the encoded sequence. The use of interframe coding in such a manner would improve the quality of the decoded video with respect to the results presented in Chapter 6.

REFERENCES

REFERENCES

[1] H. Derin and H. Elliot, "Modeling and segmentation of noisy and textured images using Gibbs random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 1, pp. 39–55, January 1987.

[2] J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *Journal of the Royal Statistical Society B*, vol. 36, pp. 192–236, 1974.

[3] J. Marroquin, S. Mitter, and T. Poggio, "Probabilistic solution of ill-posed problems in computational vision," *Journal of the American Statistical Association*, vol. 82, no. 397, pp. 76–89, March 1987.

[4] F. S. Cohen and D. B. Cooper, "Simple parallel hierarchical and relaxation algorithms for segmen ting noncausal Markovian random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, pp. 195–219, March 1987.

[5] H. Derin and W. S. Cole, "Segmentation of textured images using Gibbs random fields," *Computer Vision, Graphics, and Image Processing*, vol. 35, pp. 72–98, 1986.

[6] P. A. Kelly, H. Derin, and K. D. Hart, "Adaptive segmentation of speckled images using a hierarchical random field model," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, pp. 1626–1641, October 1988.

[7] S. Lakshmanan and H. Derin, "Simultaneous parameter estimation and segmentation of Gibbs random fields using simulated annealing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 8, pp. 799–813, August 1989.

[8] M. L. Comer and E. J. Delp, "Parameter estimation and segmentation of noisy or textured images using the EM algorithm and MPM estimation," *Proceedings of the 1994 IEEE International Conference on Imag e Processing*, November 1994, Austin, Texas, pp. 650–654.

[9] J. Zhang, J. W. Modestino, and D. A. Langan, "Maximum-likelihood parameter estimation for unsupervised model-ba sed image segmentation," *IEEE Transactions on Image Processing*, vol. 3, no. 4, pp. 404–420, July 1994.

[10] T. N. Pappas, "An adaptive clustering algorithm for image segmentation," *IEEE Transactions on Signal Processing*, vol. 40, pp. 901–914, April 1992.

[11] C. Bouman and M. Shapiro, "A multiscale random field model for Bayesian image segmentation," *IEEE Transactions on Image Processing*, vol. 3, no. 2, pp. 162–177, March 1994.

[12] C. Bouman and B. Liu, "Multiple resolution segmentation of textured images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 2, pp. 99–113, February 1991.

[13] I. Ng, J. Kittler, and J. Illingworth, "Supervised segmentation using a multiresolution data representati on," *Signal Processing*, vol. 31, no. 2, pp. 133–163, March 1993.

[14] Z. Kato, M. Berthod, and J. Zerubia, "Multiscale Markov random field models for parallel image classification," *Proceedings of the Fourth International Conference on Computer Vision*, May 1993, Berlin, Germany.

[15] J. Mao and A. K. Jain, "Texture classification and segmentation using multiresolution sim ultaneous autoregressive models," *Pattern Recognition*, vol. 25, no. 2, pp. 173–188, February 1992.

[16] S. Krishnamachari and R. Chellappa, "Multiresolution Gauss Markov random field models," tech. rep., University of Maryland, College Park, Maryland, January 1995.

[17] M. L. Comer and E. J. Delp, "Multiresolution image segmentation," *Proceedings of the 1995 IEEE International Conference on Acou stics, Speech, and Signal Processing*, May 1995, Detroit, Michigan, pp. 2415–2418.

[18] M. Luettgen and A. Willsky, "Likelihood calculation for a class of multiscale stochastic model s, with application to texture discrimination," *IEEE Transactions on Image Processing*, vol. 4, no. 2, pp. 194–207, February 1995.

[19] J. Serra, *Image Analysis and Mathematical Morphology.* New York: Academic Press, 1982.

[20] R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 4, pp. 532–550, July 1987.

[21] P. Maragos and R. W. Schafer, "Morphological filters–Part I: Their set-theoretic analysis and relations to linear shift-invariant filter," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-35, no. 8, pp. 1153–1169, August 1987.

[22] T. Hanamura, W. Kameyama, and H. Tominaga, "Hierarchical coding scheme of video signal with scalability and compatibility," *Signal Processing: Image Communications*, vol. 5, pp. 159–184, 1993.

[23] D. Taubman and A. Zakhor, "Multirate 3-D subband coding of video," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 572–588, September 1994.

[24] R. Kinderman and J. L. Snell, *Markov Random Fields and Their Applications.* Providence, Rhode Island: American Mathematical Society, 1980.

[25] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 6, pp. 721–741, November 1984.

[26] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society B*, vol. 39, pp. 1–38, 1977.

[27] C. F. J. Wu, "On the convergence properties of the EM algorithm," *The Annals of Statistics*, vol. 11, no. 1, pp. 95–103, 1983.

[28] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *Society for Industrial and Applied Mathematics Review*, vol. 26, no. 2, pp. 195–239, April 1984.

[29] M. Aitkin and D. B. Rubin, "Estimation and hypothesis testing in finite mixture models," *Journal of the Royal Statistical Society Series B*, vol. 47, no. 1, pp. 67–75, 1985.

[30] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532–540, April 1983.

[31] P. Maragos, "Morphological systems for multidimensional signal processing," *Proceedings of the IEEE*, vol. 78, no. 4, pp. 690–709, April 1990.

[32] S. R. Sternberg, "Biomedical image processing," *IEEE Computer Magazine*, pp. 22–34, January 1983.

[33] C. H. G. Wright, E. J. Delp, and J. N. C. Gallagher, "Nonlinear target enhancement for the hostile nuclear environment," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 1, pp. 122–145, January 1990.

[34] A. M. Darwish and A. K. Jain, "A rule based approach for visual pattern inspection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, pp. 56–68, January 1988.

[35] D. Schonfeld and J. Goutsias, "Optimal morphological pattern restoration from noisy binary image s," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 1, pp. 14–29, January 1991.

[36] F. Sun and P. Maragos, "Experiments on image compression using morphological pyramids," *Proceedings of the SPIE Visual Communications and Image Processing Conference IV* vol. 1199, November 1989, pp. 1303–1312.

[37] L. A. Overturf, M. L. Comer, and E. J. Delp, "Color image coding using morphological pyramid decomposition," *IEEE Transactions on Image Processing*, vol. 4, no. 2, pp. 177–185, February 1995.

[38] C. H. Chu and E. J. Delp, "Impulsive noise suppression and background normalization of electrocardiogram signals using morphological operators," *IEEE Transactions on Biomedical Engineering*, vol. 36, no. 2, pp. 262–273, February 1989.

[39] B. R. Hunt and O. Kubler, "Karhunen-loeve multispectral image restoration, part 1: Theory," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-32, no. 3, pp. 592–600, June 1984.

[40] N. Ohyama, M. Yachida, E. Badique, J. Tsujiuchi, and T. Honda, "Least squares filter for color image restoration," *Journal of the Optical Society of America A*, vol. 5, no. 1, pp. 19–24, January 1988.

[41] N. P. Galatsanos and R. T. Chin, "Digital restoration of multichannel images," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-37, no. 3, pp. 415–421, March 1989.

[42] A. M. Tekalp and G. Pavlovic, "Multichannel image modeling and Kalman filtering for multispect ral image restoration," *Signal Processing*, vol. 19, no. 3, pp. 221–232, March 1990.

[43] N. P. Galatsanos, A. Katsaggelos, R. T. Chin, and A. D. H. lery, "Least squares restoration of multichannel images," *IEEE Transactions on Signal Processing*, vol. 39, no. 10, pp. 2222–2236, October 1991.

[44] N. P. Galatsanos and R. T. Chin, "Restoration of color images by multichannel Kalman filtering," *IEEE Transactions on Signal Processing*, vol. 39, no. 10, pp. 2237–2252, October 1991.

[45] A. K. Katsaggelos, K. T. Lay, and N. P. Galatsanos, "A general framework for frequency domain multichannel signal proc essing," *IEEE Transactions on Image Processing*, vol. 2, no. 3, pp. 417–420, July 1993.

[46] J. Astola, P. Haavisto, and Y. Neuvo, "Vector median filters," *Proceedings of the IEEE*, vol. 78, no. 4, pp. 678–689, April 1990.

[47] R. Hardie and G. Arce, "Ranking in $\mathbf{R}^p$ and its use in multivariate image estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 1, no. 2, pp. 197–209, June 1991.

[48] P. E. Trahanias and A. N. Venetsanopolous, "Vector directional filters–a new class of multichannel image processing filters," *IEEE Transactions on Image Processing*, vol. 2, no. 4, pp. 528–534, October 1993.

[49] V. Barnett, "The ordering of multivariate data," *Journal of the Royal Statistical Society A*, vol. 139, pp. 318–355, 1976.

[50] M. L. Comer and E. J. Delp, "An empirical study of morphological operators in color image enhancement," *Proceedings of the SPIE Conference on Image Processing Algorithms and Techniques III*, vol. 1657, February 1992, San Jose, California, pp. 314–325.

[51] R. L. Stevenson and G. R. Arce, "Morphological filters: Statistics and further syntactic properties," *IEEE Transactions on Circuits and Systems*, vol. 34, no. 11, pp. 1292–1305, 1987.

[52] J. Song and E. J. Delp, "The analysis of morphological filters with multiple structuring elements," *Computer Vision, Graphics, and Image Processing*, vol. 50, pp. 308–328, June 1990.

[53] J. Song and E. J. Delp, "A study of the generalized morphological filter," *Circuits, Systems, and Signal Processing*, vol. 11, no. 1, pp. 229–252, 1991.

[54] D. L. MacAdam, *Color Measurement: Theme and Variations*. Berlin: Springer-Verlag, 1985.

[55] R. W. G. Hunt, *The Reproduction of Colour in Photography, Printing, & Television*. England: Fountain Press, 1987.

[56] J. Serra, *Image Analysis and Mathematical Morphology Volume 2: Theoretical Advances*. London: Academic Press, 1988.

[57] P. Maragos, "Pattern spectrum and multiscale shape representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 701–716, July 1989.

[58] M. Chen and P. Yan, "A multiscaling approach based on morphological filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 694–700, July 1989.

[59] M. Vetterli and C. Herley, "Wavelets and filter banks: Theory and design," *IEEE Transactions on Signal Processing*, vol. 40, no. 9, pp. 2207–2232, September 1992.

[60] A. Rosenfeld and G. J. Vanderbrug, "Coarse-fine template matching," *IEEE Transactions on Systems, Man, and Cybernetics*, pp. 104–107, February 1977.

[61] A. P. Witkin, "Scale-space filtering," *Proceedings of the International Joint Conference on Artificial Intelligence*, 1983, pp. 1019–1022.

[62] A. L. Yuille and T. A. Poggio, "Scaling theorems for zero-crossings," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, January 1986.

[63] F. Mokhtarian and A. Mackworth, "Scale-based description and recognition of planar curves and 2D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, January 1986.

[64] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Transactions on Image Processing*, vol. 1, no. 2, pp. 205–220, April 1992.

[65] J. W. Woods and S. D. O'Neil, "Subband coding of images," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, pp. 1278–1288, October 1986.

[66] P. J. Burt, *The Pyramid As a Structure for Efficient Computation*, pp. 6–35. Berlin: Springer-Verlag, 1984.

[67] D. Terzopoulos, "Image anaysis using multigrid relaxation methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 129–139, March 1986.

[68] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, July 1989.

[69] A. Toet, "A morphological pyramidal image decomposition," *Pattern Recognition Letters*, vol. 9, no. 1, pp. 255–261, May 1989.

[70] S. Mallat and S. Zhong, "Characterization of signals from multiscale edges," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 7, pp. 710–732, July 1992.

[71] I. Pitas and P. Tsakalidis, "Multivariate ordering in color image filtering," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 1, no. 3, pp. 247–259, September 1991.

[72] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. Boston: Academic Press, 1990.

[73] J. M. Shapiro, "Application of the embedded wavelet hierarchical image coder to very low bit rate image coding," *Proceedings of the 1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, April 1993, Minneapolis, Minnesota, pp. 558–561.

[74] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Communications on Pure and Applied Mathematics*, vol. 41, pp. 909–996, November 1988.

[75] A. S. Lewis and G. Knowles, "Image compression using the 2-d wavelet transform," *IEEE Transactions on Image Processing*, vol. 1, pp. 244–250, April 1992.

[76] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3445–3462, December 1993.

[77] R. A. DeVore, B. Jawerth, and B. J. Lucier, "Image compression through wavelet transform coding," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 719–746, March 1992.

[78] B. Yazici, M. L. Comer, R. L. Kashyap, and E. J. Delp, "A tree structured Bayesian scalar quantizer for wavelet based image c ompression," *Proceedings of the 1994 IEEE International Conference on Image Proc essing*, vol. III, November 13–16 1994, Austin, Texas, pp. 339–342.

[79] Z. Xiong, K. Ramchandran, M. T. Orchard, and K. Asai, "Wavelet packets-based image coding using joint space-frequency quantization," *Proceedings of the 1994 IEEE International Conference on Image Processing*, November 1994, Austin, Texas, pp. 324–328.

[80] J.-R. Ohm, "Three-dimensional subband coding with motion compensation," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 559–571, September 1994.

[81] C. I. Podilchuk, N. S. Jayant, and N. Farvardin, "Three-dimensional subband coding of video," *IEEE Transactions on Image Processing*, vol. 4, no. 2, pp. 125–139, February 1995.

[82] E. H. Adelson, E. Simoncelli, and R. Hingorani, "Orthogonal pyramid transforms for image coding," *Proceedings of the SPIE*, vol. 845, October 1987, Cambridge, Massachusetts, pp. 50–58.

VITA

VITA

Name: Mary L. Comer

Address:

School of Electrical Engineering
1285 Electrical Engineering Building
Purdue University
West Lafayette, IN 47907-1285

Office Phone: (317) 494-3645

Fax: (317) 494-6440

e-mail: `lafuze@ecn.purdue.edu`

World-Wide Web: `http://yara.ecn.purdue.edu/~lafuze`

Education:

Degree	*Date*	*School*
PhD	1995	Purdue University

Dissertation: Multiresolution Segmentation of Textured Images Using the EM/MPM Algorithm

Advisor: Edward J. Delp

MSEE	1993	Purdue University
BSEE	1990	Purdue University

Professional Experience:

June 1993 - October 1993	Consultant, Miles Inc., Elkhart, Indiana
January 1992 - August 1992	Research Assistant, Purdue University, West Lafayette, Indiana
June 1990 - December 1991	Teaching Assistant, Purdue University, West Lafayette, Indiana
May 1989 - August 1989	Summer Undergraduate Research Intern, Purdue University, West Lafayette, Indiana
May 1986 - May 1989	Co-Op Student, Delco Remy Division of General Motors, Anderson, Indiana

Awards and Honors:

- National Science Foundation Graduate Fellowship, 1992 – 1995
- First Place Student Presentation, Purdue Electrical Engineering Industrial Institute Spring 1994 Workshop
- National Merit Scholarship, 1985
- Hoosier Scholarship, 1985

Professional Organizations:

Eta Kappa Nu

IEEE

Papers:

1. M. L. Comer and E. J. Delp, "The EM/MPM algorithm for segmentation of textured images: Analysis and further experimental results," *to be submitted to IEEE Transactions on Image Processing.*

2. M. L. Comer and E. J. Delp, "Segmentation of textured images using a multiresolution Gaussian auto-regressive model," *to be submitted to IEEE Transactions on Image Processing.*

3. M. L. Comer and E. J. Delp, "Morphological operations for color image processing," *submitted to IEEE Transactions on Image Processing.*

4. L. A. Overturf, M. L. Comer, and E. J. Delp, "Color image coding using morphological pyramid decomposition," *IEEE Transactions on Image Processing*, vol. 4, no. 2, pp. 177–185, February 1995.

5. M. L. Comer and E. J. Delp, "Multiresolution image segmentation," *Proceedings of the 1995 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, May 1995, Detroit, Michigan, pp. 2415–2418.

6. M. L. Comer and E. J. Delp, "Parameter estimation and segmentation of noisy or textured images using the EM algorithm and MPM estimation," *Proceedings of the 1994 IEEE International Conference on Image Processing*, vol. II, November 13–16 1994, Austin, Texas, pp. 650–654.

7. B. Yazici, M. L. Comer, R. L. Kashyap, and E. J. Delp, "A tree structured Bayesian scalar quantizer for wavelet based image compression," *Proceedings of the 1994 IEEE International Conference on Image Processing*, vol. III, November 13–16 1994, Austin, Texas, pp. 339–342.

8. G. W. Cook, M. L. Comer, and E. J. Delp, "An investigation of the use of high performance computing for multiscale color image smoothing using mathematical morphology," *Proceedings of the SPIE Conference on Image Modeling*, vol. 1904, January 31–February 4, 1993, San Jose, California, pp. 104–114.

9. M. L. Comer and E. J. Delp, "An empirical study of morphological operators in color image enhancement," *Proceedings of the SPIE Conference on Image Processing Algorithms and Techniques III*, vol. 1657, February 1992, San Jose, California, pp. 314–325.

10. L. A. Overturf, M. L. Comer, and E. J. Delp, "Color image coding using morphological pyramid decomposition," *Proceedings of the SPIE Conference on Human Vision, Visual Processing, and Digital Display III*, vol. 1666, February 1992, San Jose, California, pp. 265–275.