Bayesian Pixel Classification Using Spatially Variant Finite Mixtures and the Generalized EM Algorithm

S. Sanjay-Gopal and Thomas J. Hebert, *Member, IEEE*

***Abstract*—A spatially variant finite mixture model is proposed for pixel labeling and image segmentation. For the case of spatially varying mixtures of Gaussian density functions with unknown means and variances, an expectation-maximization (EM) algorithm is derived for maximum likelihood estimation of the pixel labels and the parameters of the mixture densities. An *a priori* density function is formulated for the spatially variant mixture weights. A generalized EM algorithm for maximum *a posteriori* estimation of the pixel labels based upon these prior densities is derived. This algorithm incorporates a variation of gradient projection in the maximization step and the resulting algorithm takes the form of grouped coordinate ascent. Gaussian densities have been used for simplicity, but the algorithm can easily be modified to incorporate other appropriate models for the mixture model component densities. The accuracy of the algorithm is quantitatively evaluated through Monte Carlo simulation, and its performance is qualitatively assessed via experimental images from computerized tomography (CT) and magnetic resonance imaging (MRI).**

I. Introduction

IMAGE segmentation is essentially a process of pixel classification wherein the image pixels are segmented into subsets by assigning the individual pixels to classes. Given a known or estimated number of classes, each class can be associated with a numeric label, so that pixel labeling consists of assigning a numeric label to each pixel. Pixel classification or labeling can be based on simple grey-level thresholding, color, or local property values [1]. Simple grey-level thresholding typically provides acceptable results only in the absence of statistical noise and partial volume effects.

Relaxation labeling [2], [3] is an iterative scheme that has been proposed for pixel labeling of images corrupted by noise. Here, label assignments for all pixels are initially formed based upon the observed image. Then this initial labeling is iteratively changed to maximize a particular function that models user-defined relationships between neighboring pixel labels [2]. Linear relaxation converges to a solution that depends only upon the user-defined relationships, but upon neither the observed image nor the initial labeling [3]. Nonlinear relaxation labeling provides a solution that depends upon the initial labeling and is typically a reasonable improvement. However, its convergence properties have not been established [3]. The common drawback of both linear and nonlinear relaxation labeling is that these methods make little or no use of the observed image, except perhaps to initialize the label configuration for the iterative algorithm. Relaxation labeling remains a somewhat *ad hoc* technique that often provides an improvement of labels that have been assigned using some other labeling method.

Several researchers have proposed solutions for the segmentation problem using statistical image models [4]–[7]. In [4], Lakshmanan and Derin have proposed and analyzed an adaptive segmentation algorithm based on modeling the image as a Gibbs random field corrupted by additive, zero-mean, independent, and identically distributed (i.i.d.) Gaussian random noise of known variance. Based on the assumption that the number of regions and the corresponding grey levels are known, their iterative algorithm first estimates the parameters characterizing the Gibbs distribution. These estimates are subsequently used to segment the image. Even with these significant assumptions, the resulting algorithm is computationally involved and the authors suggest several approximations for a faster implementation. The algorithm, however, is appealing from a convergence point of view. In [5], Manjunath and Chellappa model the observed image pixel intensities as a Gauss–Markov random field and estimate the parameters characterizing this random field by clustering the image into nonoverlapping regions of uniform texture. These estimated parameters are subsequently used to segment the image. The authors have also presented an interesting example where a simple nearest neighbor classification algorithm yields results comparable to those obtained using a computationally involved algorithm such as the one proposed in [4]. Derin and Elliot [6] have developed a dynamic programming approach to textured image segmentation. Bouman and Liu [7] have examined multiresolution approaches to texture segmentation. They have used an information theoretic approach to estimate the number of underlying texture classes in the image and have used simulated annealing to maximize a posterior function based on modeling the individual textures as a causal nonhomogeneous Gaussian autoregressive random field. A common drawback of the statistical model-based methods discussed above is that they are all computationally intensive.

Recently, finite mixture models [8] have attracted considerable interest for pixel labeling [9]–[11]. The application of these models to labeling is based on the assumption that the intensity (grey-scale or color) value of each pixel in the observed image is a sample from a finite mixture distribution.

Manuscript received May 27, 1996; revised October 3, 1997. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Dmitry B. Goldgof.

S. Sanjay-Gopal is with the Department of Radiology, University of Michigan, Ann Arbor, MI 48109-0904 USA.

T. J. Hebert is with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77204-4793 USA (e-mail: thebert@uh.edu).

Publisher Item Identifier S 1057-7149(98)04368-1.

Upon estimating the parameters of this distribution a suitable labeling rule is applied to assign labels to the pixels of an image. As shown in [10] and [11], finite mixture models can also be used with other appropriate models for partial volume effect for optimal pixel labeling. They also provide an attractive alternative to some of the more complicated Markov random field (MRF) models and their related computationally intensive algorithms. There are several instances, similar to the example presented in [5], where a simple algorithm would suffice and a computationally involved algorithm is not needed. The work presented in this paper aims to bridge the gap between some of the computationally involved algorithms based on random fields and less intensive segmentation algorithms based on mixture models.

The organization of this paper is as follows. In Section II, we closely examine finite mixture models. We discuss the application of this model to pixel classification and point out the many drawbacks of applying the classical mixture model (described in [8]) to labeling. A spatially variant finite mixture model for pixel classification is proposed in Section III to overcome these drawbacks and an algorithm for the maximum likelihood (ML) estimation of the pixel labels is derived in Section IV. In Section V, we formulate a generalized expectation maximization (GEM) [12] algorithm for maximum *a posteriori* (MAP) estimation of pixel labels based upon the spatially variant model. This algorithm is fast, robust, easy to implement, and it embeds the attractive properties of finite mixture models in a Bayesian framework. Finally, it is flexible in that it can be applied to any arbitrary image data where the number of classes is known. The number of classes can be estimated using a suitable method [13], [14]. Note that although we have derived the algorithm on the assumption of Gaussian densities, it can easily be modified to include other models for the component densities of the mixture model. Finally, in Section VI we document an extensive analysis of the algorithm by means of computer simulation and experimental phantom studies. We also present examples of the results of applying the proposed algorithms to segment magnetic resonance images (MRI's).

II. The Classical Mixture Model and Pixel Labeling

To define the classical finite mixture model (CMM), let $\vec{\boldsymbol{x}}_i$ denote the observation at the ith pixel of an image. Let observations $\vec{\boldsymbol{x}}_i, 1 \leq i \leq N$ be modeled as i.i.d. Let $\{f_j(\vec{\boldsymbol{x}}_i|\vec{\theta}^j)\}$ be a set of L density functions, each having its own vector of parameters $\vec{\theta}^j$. Here, subscripts are used to index the elements of a vector or matrix, while superscripts are used to index a particular vector from within a set of vectors. Let a vector of parameters $\vec{\pi}$ satisfy $\vec{\pi}_j \geq 0\ \forall j$ and $\Sigma_{j=1}^{L} \vec{\pi}_j = 1$. Now, the CMM [8] addresses the problem in which the joint conditional density $f_x(\vec{\boldsymbol{x}}|\vec{\pi}, \vec{\theta}^1, \cdots, \vec{\theta}^L)$ of the observations is formed as

$$f_x(\vec{\boldsymbol{x}}|\vec{\pi}, \vec{\theta}^1 \cdots \vec{\theta}^L) = \prod_{i=1}^{N} \sum_{j=1}^{L} \vec{\pi}_j f_j(\vec{\boldsymbol{x}}_i|\vec{\theta}^j). \tag{1}$$

The $\vec{\pi}_j$'s are termed the mixing weights and the $f_j(\cdot)$'s are termed the component densities.

Titterington [8] derived an EM algorithm for ML estimation of the parameters of the component densities for the case where $f_j(\vec{\boldsymbol{x}}_i|\vec{\theta}^j)$ are all assumed univariate Gaussian and measurements $\vec{\boldsymbol{x}}$ are i.i.d. Let $L \times 1$ vectors $\vec{\mu}$ and $\vec{\sigma}$ contain, respectively, the unknown means and standard deviations so that

$$f_j(\vec{\boldsymbol{x}}_i|\vec{\theta}^j) = f_j(\vec{\boldsymbol{x}}_i|\vec{\mu}_j, \vec{\sigma}_j) = \frac{1}{\sqrt{2\pi}\vec{\sigma}_j} \exp\left[-\frac{(\vec{\boldsymbol{x}}_i - \vec{\mu}_j)^2}{2(\vec{\sigma}_j)^2}\right]. \tag{2}$$

Then, the iterative EM algorithm for estimating the parameters of the component densities is given by

$$\vec{\boldsymbol{w}}_j^{i(k)} = \frac{\vec{\pi}_j^{(k)} f_j(\vec{\boldsymbol{x}}_i|\vec{\mu}_j^{(k)}, \vec{\sigma}_j^{(k)})}{\sum_{l=1}^{L} \vec{\pi}_l^{(k)} f_l(\vec{\boldsymbol{x}}_i|\vec{\mu}_l^{(k)}, \vec{\sigma}_l^{(k)})} \tag{3}$$

$$\vec{\pi}_j^{(k+1)} = \frac{1}{N} \sum_{i=1}^{N} \vec{\boldsymbol{w}}_j^{i(k)} \tag{4}$$

$$\vec{\mu}_j^{(k+1)} = \frac{1}{N\vec{\pi}_j^{(k+1)}} \sum_{i=1}^{N} \vec{\boldsymbol{w}}_j^{i(k)} \vec{\boldsymbol{x}}_i \tag{5}$$

and

$$[(\vec{\sigma}_j)^2]^{(k+1)} = \frac{1}{N\vec{\pi}_j^{(k+1)}} \sum_{i=1}^{N} \vec{\boldsymbol{w}}_j^{i(k)} [\vec{\boldsymbol{x}}_i - \vec{\mu}_j^{(k+1)}]^2. \tag{6}$$

As has been shown in [8], the likelihood function of (1) exhibits local maxima and Titterington's EM algorithm will converge to one of these.

Parameters $\vec{\pi}, \vec{\mu}$, and $\vec{\sigma}$ all describe the global properties of the measurements $\vec{\boldsymbol{x}}$, but they do not tell how to assign labels to the pixels. To examine this, begin by observing that the CMM in (1) incorporates classes in the following manner. Let there be N pixels, each of which belongs to one of L classes. Let vectors $\{\vec{\boldsymbol{p}}^i\}$ contain the pixel labels such that $\vec{\boldsymbol{p}}_j^i = 1$, if the ith pixel belongs to the jth class, and $\vec{\boldsymbol{p}}_j^i = 0$, otherwise. Let the probability of any pixel belonging to the jth class be denoted by $\vec{\pi}_j$, so that $\text{Prob}\,(\vec{\boldsymbol{p}}_j^i = 1) = \vec{\pi}_j, \forall i$. Associated with each class is a component density $f_j(\vec{\boldsymbol{x}}_i|\vec{\theta}^j)$. If $\vec{\boldsymbol{p}}_j^i = 1$, then the corresponding pixel value $\vec{\boldsymbol{x}}_i$ is generated from the component density $f_j(\vec{\boldsymbol{x}}_i|\vec{\theta}^j)$; i.e.,

$$f(\vec{\boldsymbol{x}}_i|\vec{\boldsymbol{p}}_j^i = 1, \vec{\theta}^j) = f_j(\vec{\boldsymbol{x}}_i|\vec{\theta}^j). \tag{7}$$

Bayes rule gives

$$\text{Prob}\,(\vec{\boldsymbol{p}}_j^i = 1|\vec{\boldsymbol{x}}_i, \vec{\theta}^j) = \frac{\vec{\pi}_j f_j(\vec{\boldsymbol{x}}_i|\vec{\theta}^j)}{f(\vec{\boldsymbol{x}}_i|\vec{\theta}^1 \cdots \vec{\theta}^L)} \tag{8}$$

where

$$f(\vec{\boldsymbol{x}}_i|\vec{\theta}^1 \cdots \vec{\theta}^L) = \sum_{j=1}^{L} \vec{\pi}_j f_j(\vec{\boldsymbol{x}}_i|\vec{\theta}^j) \tag{9}$$

is simply some constant value dependent upon measurement $\vec{\boldsymbol{x}}_i$. In general, $\vec{\boldsymbol{x}}_i$ can be used to estimate $\vec{\boldsymbol{p}}^i$ through (8) if the parameters $\{\vec{\theta}^j\}$ of the component densities are known. Now, to assign labels to the pixels, if the parameters $\{\vec{\theta}^j\}$

were known, the Bayes classifier could be used to assign a label to pixel i by solution of

$$\max_j \text{Prob}(\vec{\boldsymbol{p}}_j^i = 1 | \vec{\boldsymbol{x}}_i, \vec{\theta}^j). \tag{10}$$

Since the component density parameters $\{\vec{\theta}^j\}$ are not known, the common approach is to use ML estimates of parameters $\{\vec{\theta}^j\}$ in place of the parameters themselves in the Bayes classifier. In the case of univariate Gaussian component densities, this entails use of ML estimates $\vec{\pi}^{\text{ML}}, \vec{\mu}^{\text{ML}}$, and $\vec{\sigma}^{2^{\text{ML}}}$ from the EM algorithm in the Bayes classifier of (8)–(10) [15]–[18]. This approach often provides acceptable estimates of the pixel labels $\{\vec{\boldsymbol{p}}^i\}$. However, this is not an optimal approach because the labels computed on the basis of ML estimates for $\vec{\pi}$ and $\{\vec{\theta}^j\}$ are not themselves ML estimates. Instead, ML estimation of the labels requires the likelihood function $f(\vec{\boldsymbol{x}}|\vec{\pi}, \vec{\boldsymbol{p}}^1 \cdots \vec{\boldsymbol{p}}^N, \vec{\theta}^1 \cdots \vec{\theta}^L)$. In addition, the CMM has another drawback upon application to pixel classification. Note the independence assumption on the elements of $\vec{\boldsymbol{x}}$ in (1). This implies that elements $\vec{\boldsymbol{x}}_i$ are uncorrelated. As shown in Appendix A, the covariance matrix for $\vec{\boldsymbol{x}}$, under the assumption of Gaussian component densities $\{f_j(\vec{\boldsymbol{x}}_i|\vec{\theta}^j)\}$ is the diagonal matrix

$$\boldsymbol{C} = \left[\sum_{j=1}^{L} \vec{\pi}_j (\vec{\sigma}_j^2 + \vec{\mu}_j^2) - \left(\sum_{j=1}^{L} \vec{\pi}_j \vec{\mu}_j \right)^2 \right] \boldsymbol{I} \tag{11}$$

where $\boldsymbol{I}$ is the identity matrix. The CMM of (1) prohibits the existence of spatial correlation between the observations at neighboring pixels. In most applications, measurements at neighboring pixels are spatially correlated. One can reintroduce spatial correlation into the labeling process using the CMM by imposing dependence structures in the form of Markov chains and two-dimensional (2-D) MRF models on the complete data of the EM algorithm; but, again, this is a nonoptimal approach, and the resulting algorithms have been found to be considerably complicated and computationally intractable [19].

III. A Spatially Variant Mixture Model for Pixel Labeling and Image Segmentation

The preceding discussion about the CMM indicates the need for a better, more flexible model for pixel labeling. Several researchers have suggested modifications to the CMM [15]–[17]. In this section we propose the *spatially variant finite mixture model* (SVMM) for pixel labeling. Let $\vec{\boldsymbol{p}}_j^i$ denote the probability of the ith pixel belonging to the jth class; $0 \le \vec{\boldsymbol{p}}_j^i \le 1$ and $\Sigma_j \vec{\boldsymbol{p}}_j^i = 1 \ \forall i$. As in the CMM, $\vec{\boldsymbol{x}}_i$ denotes the observation at the ith pixel of an image, and $\{f_j(\vec{\boldsymbol{x}}_i|\vec{\theta}^j)\}$ is a set of L density functions, each having its own vector of parameters $\vec{\theta}^j$. The SVMM defines the density function of the observation $\vec{\boldsymbol{x}}_i$ as

$$f(\vec{\boldsymbol{x}}_i|\vec{\boldsymbol{p}}^1 \cdots \vec{\boldsymbol{p}}^N, \vec{\theta}^1 \cdots \vec{\theta}^L) = \sum_{j=1}^{L} \vec{\boldsymbol{p}}_j^i f_j(\vec{\boldsymbol{x}}_i|\vec{\theta}^j). \tag{12}$$

If the observations $\vec{\boldsymbol{x}}_i$ are modeled as statistically independent, the joint conditional density of the observations can be formed as

$$f_x(\vec{\boldsymbol{x}}|\vec{\boldsymbol{p}}^1 \cdots \vec{\boldsymbol{p}}^N, \vec{\theta}^1 \cdots \vec{\theta}^L) = \prod_{i=1}^{N} \sum_{j=1}^{L} \vec{\boldsymbol{p}}_j^i f_j(\vec{\boldsymbol{x}}_i|\vec{\theta}^j). \tag{13}$$

It can easily be proved that (12) integrates to one and represents a valid density. Note that the SVMM poses no restriction on the form of the underlying component densities except that these densities be identifiable. Although in this paper we have modeled these densities as Gaussian for deriving algorithms for pixel labeling, the SVMM is general enough to provide the user the flexibility to choose any appropriate model for these densities.

The SVMM presents the following several distinct advantages for pixel labeling as compared to the CMM.

1) It lends itself easily to direct ML estimation of the pixel labels $\vec{\boldsymbol{p}}_j^i$ through specification of the likelihood function in (13).
2) It facilitates the modeling of spatial correlation directly on the label information $\{\vec{\boldsymbol{p}}^i\}$ through specification of prior densities $f(\vec{\boldsymbol{p}}^1 \cdots \vec{\boldsymbol{p}}^N)$.
3) It incorporates the CMM as a special case wherein $\vec{\boldsymbol{p}}_j^i = \vec{\pi}_j \ \forall i$. Hence, the SVMM can also be applied to data that conform to the CMM.
4) As proven in Appendix B, under some mild conditions, ML estimation of the label probabilities $\vec{\boldsymbol{p}}_j^i$ converges to 0's and 1's, so that labeling is unambiguous.

IV. ML Pixel Labeling

In this section, we derive an EM algorithm for ML estimation of both the pixel label probabilities $\{\vec{\boldsymbol{p}}^i\}$ and the parameter vectors $\{\vec{\theta}^j\}$ using the joint conditional density of (13). Here we use component densities $\{f_j(\vec{\boldsymbol{x}}_i|\vec{\theta}^j)\}$ that are univariate Gaussian.

The EM method [12] for ML estimation provides a straightforward, convenient algorithm for maximizing (13). It first requires specification of a "complete" data set $\vec{\boldsymbol{y}}$ as opposed to the "incomplete," noisy image data $\vec{\boldsymbol{x}}$. This specification is dependent upon the creativity of the researcher and sets the form of the algorithm [20], [21]. EM algorithms constitute a class of algorithms. In deriving an EM algorithm, the conditional density function $f_y(\vec{\boldsymbol{y}}|\vec{\boldsymbol{p}}^1 \cdots \vec{\boldsymbol{p}}^N, \vec{\theta}^1 \cdots \vec{\theta}^L)$ for the complete data $\vec{\boldsymbol{y}}$ must first be derived. Then, given any initial estimate of the parameters $\{\vec{\boldsymbol{p}}^{i^{(0)}}\}$ and $\{\vec{\theta}^{j^{(0)}}\}$, an EM algorithm for estimating $\{\vec{\boldsymbol{p}}^i\}$ and $\{\vec{\theta}^j\}$ consists of two steps at each iteration: an E-step where a conditional expected value is computed, and an M-step where the current parameter estimates are updated by maximizing this conditional expectation. For conciseness, we define parameter sets Ψ and their estimates $\Psi^{(k)}$ from k iterations of the algorithm as

$$\Psi \equiv (\vec{\boldsymbol{p}}^1 \cdots \vec{\boldsymbol{p}}^N, \vec{\theta}^1 \cdots \vec{\theta}^L) \tag{14}$$

and

$$\Psi^{(k)} \equiv (\vec{\boldsymbol{p}}^{1^{(k)}} \cdots \vec{\boldsymbol{p}}^{N^{(k)}}, \vec{\theta}^{1^{(k)}} \cdots \vec{\theta}^{L^{(k)}}). \tag{15}$$

Then, the E-step is simply to form

$$Q(\Psi|\Psi^{(k)}) \equiv E_{\vec{\boldsymbol{y}}}\{\ln[f_y(\vec{\boldsymbol{y}}|\Psi)]|\vec{\boldsymbol{x}}, \Psi^{(k)}\} \tag{16}$$

and the M-step is given by

$$\max_{\Psi} Q(\Psi|\Psi^{(k)}). \tag{17}$$

The complete data used by Titterington [8] to derive an EM algorithm for the CMM can also be applied to derive an EM algorithm for the SVMM. To define this complete data, let $\vec{\boldsymbol{z}}^i, i = 1,2,\cdots,N$ denote $L \times 1$ random indicator vectors, each of which takes a value from the set of vectors $\mathcal{L}$ defined by

$$\mathcal{L} = \{\vec{\boldsymbol{e}}^j; \vec{\boldsymbol{e}}^j_{l=j} = 1, \vec{\boldsymbol{e}}^j_{l\neq j} = 0, 1 \leq j,l \leq L\}. \tag{18}$$

Note that $\vec{\boldsymbol{z}}^i_j$ is a discrete random variable with probability function defined by $\text{Prob}\,(\vec{\boldsymbol{z}}^i = \vec{\boldsymbol{e}}^j) = \vec{\boldsymbol{p}}^i_j \ \forall i,j$. Now the complete data $\vec{\boldsymbol{y}}$ is defined by $\vec{\boldsymbol{y}}^T \equiv (\vec{\boldsymbol{x}}^T, \vec{\boldsymbol{z}}^{1^T}, \cdots, \vec{\boldsymbol{z}}^{N^T})$ with superscript T denoting vector transpose. Here, $\vec{\boldsymbol{x}}$ and $\{\vec{\boldsymbol{z}}^i\}$ are defined to be statistically independent so that

$$f_y(\vec{\boldsymbol{y}}|\Psi) = \prod_{i=1}^{N}\prod_{j=1}^{L}[\vec{\boldsymbol{p}}^i_j f_j(\vec{\boldsymbol{x}}_i|\vec{\theta}^j)]^{\vec{\boldsymbol{z}}^i_j}. \tag{19}$$

For the case of Gaussian component densities $f_j(\vec{\boldsymbol{x}}_i|\vec{\theta}^j)$, the resulting E-step is given by

$$Q(\Psi|\Psi^{(k)}) = \sum_{i=1}^{N}\sum_{j=1}^{L} E\{\vec{\boldsymbol{z}}^i_j|\vec{\boldsymbol{x}}_i, \Psi^{(k)}\}\left[\ln(\vec{\boldsymbol{p}}^i_j) - \frac{1}{2} \cdot \ln(2\pi\vec{\sigma}^2_j) - \frac{(\vec{\boldsymbol{x}}_i - \vec{\mu}_j)^2}{2\vec{\sigma}^2_j}\right]. \tag{20}$$

From Appendix C

$$E\{\vec{\boldsymbol{z}}^i_j|\vec{\boldsymbol{x}}_i, \Psi^{(k)}\} \equiv \vec{\boldsymbol{w}}^{i^{(k)}}_j = \frac{\vec{\boldsymbol{p}}^{i^{(k)}}_j \dfrac{1}{\sqrt{2\pi\vec{\sigma}^{2^{(k)}}_j}} \exp\left[-\dfrac{(\vec{\boldsymbol{x}}_i - \vec{\mu}^{(k)}_j)^2}{2\vec{\sigma}^{2^{(k)}}_j}\right]}{\displaystyle\sum_{l=1}^{L} \vec{\boldsymbol{p}}^{i^{(k)}}_l \frac{1}{\sqrt{2\pi\vec{\sigma}^{2^{(k)}}_l}} \exp\left[-\frac{(\vec{\boldsymbol{x}}_i - \vec{\mu}^{(k)}_l)^2}{2\vec{\sigma}^{2^{(k)}}_l}\right]}. \tag{21}$$

The form of (20) suggests that, at the kth iteration, $Q(\Psi|\Psi^{(k)})$ can be maximized independently with respect to each $\{\vec{\boldsymbol{p}}^i\}$, then with respect to $\vec{\mu}_j$, and finally with respect to $\vec{\sigma}^2_j$, resulting in the following closed form expressions for the M-step of the algorithm:

$$\vec{\boldsymbol{p}}^{i^{(k+1)}}_j = \frac{\vec{\boldsymbol{w}}^{i^{(k)}}_j}{\displaystyle\sum_{l=1}^{L}\vec{\boldsymbol{w}}^{i^{(k)}}_l} = \vec{\boldsymbol{w}}^{i^{(k)}}_j \tag{22}$$

$$\vec{\mu}^{(k+1)}_j = \frac{1}{\displaystyle\sum_{i=1}^{N}\vec{\boldsymbol{w}}^{i^{(k)}}_j}\sum_{i=1}^{N}\vec{\boldsymbol{w}}^{i^{(k)}}_j\vec{\boldsymbol{x}}_i \tag{23}$$

$$[(\vec{\sigma}_j)^2]^{(k+1)} = \frac{1}{\displaystyle\sum_{i=1}^{N}\vec{\boldsymbol{w}}^{i^{(k)}}_j}\sum_{i=1}^{N}\vec{\boldsymbol{w}}^{i^{(k)}}_j[\vec{\boldsymbol{x}}_i - \vec{\mu}^{(k+1)}_j]^2. \tag{24}$$

Here, use has been made of the fact $\Sigma_{j=1}^{L}\vec{\boldsymbol{w}}^{i^{(k)}}_j = 1\ \forall i,k$.

This EM algorithm, defined by (21)–(24), yields ML estimates for the pixel labels $\vec{\boldsymbol{p}}^i_j$, as well as the parameter vectors $\vec{\mu}$ and $\vec{\sigma}$. Additionally, as shown in Appendix B, at convergence $\vec{\boldsymbol{p}}^i_j \in (0,1)\ \forall i,j$ subject to a mild constraint. It should be noted that EM algorithms have convergence to a local maximum, and that the likelihood function of (13) has multiple local maxima, so that the algorithm will converge to one of these local maxima dependent on the algorithm initialization as discussed in Section VI-E.

V. Bayesian Pixel Classification Using a Generalized EM Algorithm

The SVMM, when applied to ML estimation of pixel labels, results in simple closed-form expressions for the M-step of the EM algorithm. However, it should be noted that given a single 2-D image with N image pixels, this algorithm estimates a large number of parameters:

1) N label parameter vectors $\{\vec{\boldsymbol{p}}^i\}$ each of length L;
2) a L-element vector $\vec{\mu}$ of means;
3) a L-element vector $\vec{\sigma}$ of standard deviations.

For multispectral images, in applications such as radar imaging and MRI, the dimensionality of the data approaches that of the parameters to be estimated. For a single 2-D image, however, the number of parameters being estimated is greater than the number of data values, so that the ML estimates themselves have large variances. As a result of these variances, ML estimates of regions of an image with all pixels of the same class will contain random arrangements of different labels. Thus, when the estimated pixel labels are viewed as an image, the variances of these estimates obscure some of the underlying spatial correlation within the labeled image. This is not surprising, because the likelihood function does not incorporate local correlations between the neighboring label parameters $\vec{\boldsymbol{p}}^i$. Local correlation can be incorporated in the estimation process through application of a suitable prior density function $f(\vec{\boldsymbol{p}}^1\cdots\vec{\boldsymbol{p}}^N, \vec{\theta}^1\cdots\vec{\theta}^L)$ that models this local correlation. Given this prior density, the *a posteriori* density function can be formed and maximized for MAP estimation of the label parameters $\{\vec{\boldsymbol{p}}^i\}$ and the component density parameters $\{\vec{\theta}^j\}$. The general form of this *a posteriori* density function is

$$f(\vec{\boldsymbol{p}}^1\cdots\vec{\boldsymbol{p}}^N, \vec{\mu}, \vec{\sigma}|\vec{\boldsymbol{x}}) = \frac{f_x(\vec{\boldsymbol{x}}|\vec{\boldsymbol{p}}^1\cdots\vec{\boldsymbol{p}}^N, \vec{\mu}, \vec{\sigma}) f(\vec{\boldsymbol{p}}^1\cdots\vec{\boldsymbol{p}}^N, \vec{\mu}, \vec{\sigma})}{r(\vec{\boldsymbol{x}})} \tag{25}$$

where $f_x(\cdot)$ is defined in (13) and $r(\vec{\boldsymbol{x}})$ is a constant dependent on image $\vec{\boldsymbol{x}}$ that does not affect the location of the maximum with respect to the parameters.

To derive an EM algorithm for MAP estimation of the parameters $\{\vec{\boldsymbol{p}}^i\}, \vec{\mu}$, and $\vec{\sigma}$, we again use the complete data $\vec{\boldsymbol{y}}^T \equiv (\vec{\boldsymbol{x}}^T, \vec{\boldsymbol{z}}^{1^T}, \cdots, \vec{\boldsymbol{z}}^{N^T})$. The E-step for the EM algorithm for MAP estimation consists of forming

$$Q_{\text{MAP}}(\Psi|\Psi^{(k)}) \equiv E_{\vec{\boldsymbol{y}}}\{\ln[f_y(\vec{\boldsymbol{y}}|\Psi)]|\vec{\boldsymbol{x}}, \Psi^{(k)}\} + \ln\{f(\vec{\boldsymbol{p}}^1\cdots\vec{\boldsymbol{p}}^N, \vec{\mu}, \vec{\sigma})\} \tag{26}$$

where the parameter sets Ψ and $\Psi^{(k)}$ are defined by (14) and (15) with $\{\vec{\theta}^j\} = (\vec{\mu}, \vec{\sigma})$ for univariate Gaussian component densities. The complete data likelihood function $f_y(\vec{\boldsymbol{y}}|\Psi)$ is defined by (19).

A powerful model for the parameters $\{\vec{\boldsymbol{p}}^i\}$ that incorporates local correlation is given by the MRF model defined through a Gibbs density function [22], [23]

$$f(\vec{\boldsymbol{p}}^1 \cdots \vec{\boldsymbol{p}}^N) = \frac{1}{K_\beta} \exp\left[-U(\vec{\boldsymbol{p}}^1 \cdots \vec{\boldsymbol{p}}^N)\right] \tag{27}$$

where

$$U(\vec{\boldsymbol{p}}^1 \cdots \vec{\boldsymbol{p}}^N) = \beta \sum_{c \in \mathcal{C}} V_c(\vec{\boldsymbol{p}}^1 \cdots \vec{\boldsymbol{p}}^N) \tag{28}$$

and $\mathcal{C}$ denotes the set of all cliques within the image. Here β and K_β are scalar constants. The set of clique types is specified by the order of the neighborhood assigned to the model. Function $V_c(\cdot)$ is termed the clique potential function for a clique of type c. The joint density should assign higher probability to sets of label vectors $\{\vec{\boldsymbol{p}}^i\}$ wherein neighboring $\vec{\boldsymbol{p}}^i$'s are similar and lower probability otherwise. Let $\{\vec{\boldsymbol{p}}^i\}$ be modeled as being described by two-pixel cliques consisting of horizontally or vertically adjacent pixel pairs, both of which are denoted here as c_2. Let potential functions $V_c(\cdot)$ on all other clique types be defined equal to zero, so that

$$\sum_{c \in \mathcal{C}} V_c(\vec{\boldsymbol{p}}^1 \cdots \vec{\boldsymbol{p}}^N) = \sum_{(i,m) \in c_2} \sum_{j=1}^{L} (\vec{\boldsymbol{p}}_j^i - \vec{\boldsymbol{p}}_j^m)^2 \tag{29}$$

where pixels i and m are nearest neighbors. Then

$$f(\vec{\boldsymbol{p}}^1 \cdots \vec{\boldsymbol{p}}^N, \vec{\mu}, \vec{\sigma}) = \frac{1}{Z_\beta} \exp\left[-\beta \sum_{(i,m) \in \mathcal{C}} \sum_{j=1}^{L} (\vec{\boldsymbol{p}}_j^i - \vec{\boldsymbol{p}}_j^m)^2\right] \cdot f(\vec{\mu}) f(\vec{\sigma}) \tag{30}$$

where parameter vectors $\vec{\mu}$ and $\vec{\sigma}$ are modeled as independent from each other and from $\{\vec{\boldsymbol{p}}^i\}$ by specifying prior density functions $f(\vec{\mu})$ and $f(\vec{\sigma})$ as uniformly distributed over the corresponding feasible parameter spaces.

Returning to the E-step, using (16), (27), and (28), the E-step of (26) for the case of Gaussian component densities is given by

$$\begin{aligned} Q_{\mathrm{MAP}}(\Psi|\Psi^{(k)}) = & \sum_{i=1}^{N} \sum_{j=1}^{L} E\{\vec{\boldsymbol{z}}_j^i | \vec{\boldsymbol{x}}_i, \Psi^{(k)}\} \\ & \cdot \left[\ln(\vec{\boldsymbol{p}}_j^i) - \frac{1}{2}\ln(2\pi\vec{\sigma}_j^2) - \frac{(\vec{\boldsymbol{x}}_i - \vec{\mu}_j)^2}{2\vec{\sigma}_j^2}\right] \\ & - \beta \sum_{c \in \mathcal{C}} V_c(\vec{\boldsymbol{p}}^1 \cdots \vec{\boldsymbol{p}}^N) - \ln(K_\beta) \\ & + \ln f(\vec{\mu}) + \ln f(\vec{\sigma}) \end{aligned} \tag{31}$$

with $E\{\vec{\boldsymbol{z}}_j^i | \vec{\boldsymbol{x}}_i, \Psi^{(k)}\} = \vec{\boldsymbol{w}}_j^{i^{(k)}}$ given by (21). Scalars $\ln f(\vec{\mu})$ and $\ln f(\vec{\sigma})$ will not affect the result from the subsequent M-step, so that these terms can be dropped from $Q_{\mathrm{MAP}}(\Psi|\Psi^{(k)})$. Given a fixed β value, the same holds true for the scalar term $\ln(K_\beta)$.

The M-step of the EM algorithm requires the maximization of $Q_{\mathrm{MAP}}(\Psi|\Psi^{(k)})$ at each iteration subject to the constraints

$$0 \le \vec{\boldsymbol{p}}_j^i \le 1; \quad \sum_{j=1}^{L} \vec{\boldsymbol{p}}_j^i = 1 \tag{32}$$

for $1 \le i \le N$ and $1 \le j \le L$. Note that $Q_{\mathrm{MAP}}(\Psi|\Psi^{(k)})$ is the same as $Q(\Psi|\Psi^{(k)})$ except for the addition of the prior terms. These terms preclude the existence of a closed-form expression for the maximum in the M-step, so that an iterative optimization technique for maximizing $Q_{\mathrm{MAP}}(\Psi|\Psi^{(k)})$ would be required. Rather than maximizing $Q_{\mathrm{MAP}}(\Psi|\Psi^{(k)})$ at each iteration M-step, simply finding $\Psi^{(k+1)}$ at each iteration such that

$$Q_{\mathrm{MAP}}(\Psi^{(k+1)}|\Psi^{(k)}) > Q_{\mathrm{MAP}}(\Psi^{(k)}|\Psi^{(k)}) \tag{33}$$

results in a GEM algorithm [12], [23] for MAP estimation of the label probabilities.

A. An M-Step Using Gradient Projection

Gradient projection [24] is an established technique for constrained optimization. Here, at each iteration the updated parameters always lie within the hyperplane wherein the constraints are satisfied. Updated parameters are found by adding a component along the direction defined by projecting the gradient of the function being optimized onto that hyperplane. Note that MAP estimation could be implemented through gradient projection applied to the conditional likelihood defined in (13). However, a better updating structure is obtained by applying it within the framework of the EM algorithm, because the M-step of the algorithm allows independent maximization of $Q_{\mathrm{MAP}}(\Psi|\Psi^{(k)})$ with respect to the parameters sets $\{\vec{\boldsymbol{p}}^i\}, \vec{\mu}$, and $\vec{\sigma}$.

Let the vector $\vec{\boldsymbol{q}}^i$ denote the gradient of $Q_{\mathrm{MAP}}(\Psi|\Psi^{(k)})$ with respect to the label probability vector $\vec{\boldsymbol{p}}^i$ so that

$$\vec{\boldsymbol{q}}_j^i \equiv \frac{\partial Q_{\mathrm{MAP}}}{\partial \vec{\boldsymbol{p}}_j^i} = \frac{\vec{\boldsymbol{w}}_j^{i^{(k)}}}{\vec{\boldsymbol{p}}_j^i} - \beta \sum_{c \in \mathcal{C}} \frac{\partial V_c(\vec{\boldsymbol{p}}^1 \cdots \vec{\boldsymbol{p}}^N)}{\partial \vec{\boldsymbol{p}}_j^i} \tag{34}$$

and let $\vec{\boldsymbol{q}}^{i^{(k)}}$ denote the evaluation of $\vec{\boldsymbol{q}}^i$ at $\vec{\boldsymbol{p}}^i = \vec{\boldsymbol{p}}^{i^{(k)}}$. Projecting $\vec{\boldsymbol{q}}^{i^{(k)}}$ onto the plane of the constraints consists of multiplying by a matrix $\boldsymbol{R}^{i,k}$ to obtain

$$\vec{\boldsymbol{d}}^{i^{(k)}} = \boldsymbol{R}^{i,k} \vec{\boldsymbol{q}}^{i^{(k)}} \tag{35}$$

where $R^{i,k}$ denotes a $L * L$ projection matrix whose elements are determined by the constraints that are active at the ith pixel and the kth iteration. The plane of the constraints on the label probability vector $\vec{\boldsymbol{p}}^{i^{(k)}}$, the gradient vector $\vec{\boldsymbol{q}}^{i^{(k)}}$, and its projection $\vec{\boldsymbol{d}}^{i^{(k)}}$ onto the plane of the constraints are illustrated in Fig. 1 for the case of three labels.

B. Form of the Projection Matrix

As shown in [24], the projection matrix $\boldsymbol{R}^{i,k}$ can be expressed as

$$\boldsymbol{R}^{i,k} = \boldsymbol{I} - \boldsymbol{A}^{i^T} \{\boldsymbol{A}^i \boldsymbol{A}^{i^T}\}^{-1} \boldsymbol{A}^i. \tag{36}$$

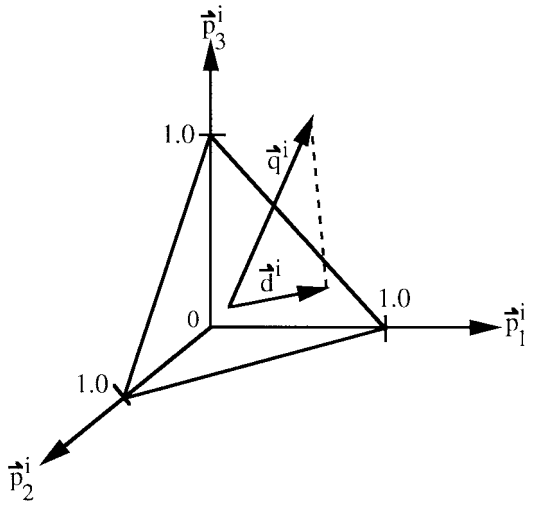


Fig. 1. For a three-class problem, shown is the plane where the constraints on the pixel label probability vector $\vec{\boldsymbol{p}}^{\,i}$ are satisfied.

Here, $\boldsymbol{I}$ is an $L * L$ identity matrix and $\boldsymbol{A}^i$ is an $M * L$ matrix that depends on the M active constraints for the ith vector $\vec{\boldsymbol{p}}^{\,i^{(k)}}$. The computation of $\boldsymbol{R}^{i,k}$ requires obtaining the inverse of $\boldsymbol{A}^i \boldsymbol{A}^{i^T}$. However, for the case of pixel labeling with constraints defined by (32), $\boldsymbol{A}^i$ exhibits a specific structure (see Appendix D) such that the general form of the inverse can be formulated. The constraint $\Sigma_j \vec{\boldsymbol{p}}_j^{\,i} = 1$ is always active. The constraint $\vec{\boldsymbol{p}}_j^{\,i} \geq 0$ is a working constraint when the condition holds that:

$$\text{condition}\,(\vec{\boldsymbol{p}}_j^{\,i}, \vec{\boldsymbol{q}}_j^{\,i}) \equiv [\vec{\boldsymbol{p}}_j^{\,i} = 0 \quad \text{and} \quad \vec{\boldsymbol{q}}_j^{\,i} < 0]. \tag{37}$$

This leads (see Appendix D) to the following form for the projection matrix:

$$\boldsymbol{R}_{j,l}^{i,k} = \begin{cases} 0, & \text{if condition } (\vec{\boldsymbol{p}}_j^{\,i^{(k)}}, \vec{\boldsymbol{q}}_j^{\,i^{(k)}}) \\ & \text{or condition } (\vec{\boldsymbol{p}}_l^{\,i^{(k)}}, \vec{\boldsymbol{q}}_l^{\,i^{(k)}}) \text{ holds.} \\ \dfrac{K-1}{K}, & \text{if } j = l \text{ and condition } (\vec{\boldsymbol{p}}_j^{\,i^{(k)}}, \vec{\boldsymbol{q}}_j^{\,i^{(k)}}) \\ & \text{does not hold.} \\ -\dfrac{1}{K}, & \text{otherwise} \end{cases} \tag{38}$$

where $K = L-$ number of elements in $\vec{\boldsymbol{p}}^{\,i^{(k)}}$ satisfying condition $(\vec{\boldsymbol{p}}_j^{\,i^{(k)}}, \vec{\boldsymbol{q}}_j^{\,i^{(k)}})$. The projection matrix as defined by (38) allows movement of $\vec{\boldsymbol{p}}_j^{\,i}$ away from zero when the partial derivative with respect to $\vec{\boldsymbol{p}}_j^{\,i}$ is positive and disallows movement when the partial derivative is negative.

C. GEM MAP Classification Algorithm

The GEM algorithm for simultaneous MAP estimation of label parameter vectors $\{\vec{\boldsymbol{p}}^{\,i}\}$ and parameter vectors $(\vec{\mu}, \vec{\sigma}^2)$ can now be described as consisting of following three steps at each iteration k.

Step 1: For $i = 1, 2, \cdots, N$ and $j = 1, 2, \cdots, L$ compute

$$\vec{\boldsymbol{w}}_j^{\,i^{(k)}} = \frac{\vec{\boldsymbol{p}}_j^{\,i^{(k)}} f_j(\vec{\boldsymbol{x}}_i | \vec{\mu}_j^{(k)}, \vec{\sigma}_j^{(k)})}{\sum_{l=1}^{L} \vec{\boldsymbol{p}}_l^{\,i^{(k)}} f_l(\vec{\boldsymbol{x}}_i | \vec{\mu}_l^{(k)}, \vec{\sigma}_l^{(k)})} \tag{39}$$

where

$$f_j(\vec{\boldsymbol{x}}_i | \vec{\mu}_j^{(k)}, \vec{\sigma}_j^{(k)}) = \frac{1}{\sqrt{2\pi}\vec{\sigma}_j^{(k)}} \exp\left[-\frac{(\vec{\boldsymbol{x}}_i - \vec{\mu}_j^{(k)})^2}{2\vec{\sigma}_j^{2^{(k)}}} \right]. \tag{40}$$

Step 2: Sequentially visit all N pixels.

Step 2a: At each pixel site i compute

$$c_1 = \sum_{j=1}^{L} \vec{\boldsymbol{w}}_j^{\,i^{(k)}} \ln \vec{\boldsymbol{p}}_j^{\,i^{(k)}} - \beta \sum_{m \in \eta_i} V_m(\vec{\boldsymbol{p}}^{\,i^{(k)}}, \vec{\boldsymbol{p}}^{\,m})$$

$$\vec{\boldsymbol{q}}_j^{\,i^{(k)}} = \frac{f_j(\vec{\boldsymbol{x}}_i | \vec{\mu}_j^{(k)}, \vec{\sigma}_j^{(k)})}{\sum_{l=1}^{L} \vec{\boldsymbol{p}}_l^{\,i^{(k)}} f_l(\vec{\boldsymbol{x}}_i | \vec{\mu}_j^{(k)}, \vec{\sigma}_j^{(k)})} - \beta \cdot \sum_{m \in \eta_i} \left[\frac{\partial V_m(\vec{\boldsymbol{p}}^{\,i}, \vec{\boldsymbol{p}}^{\,m})}{\partial \vec{\boldsymbol{p}}_j^{\,i}} \right]_{\vec{\boldsymbol{p}}^{\,i} = \vec{\boldsymbol{p}}^{\,i^{(k)}}}.$$

Step 2b: Determine the active constraints on $\vec{\boldsymbol{p}}^{\,i^{(k)}}$ and form projection matrix $\boldsymbol{R}^{i,k}$ using (38). Project the gradient vector $\vec{\boldsymbol{q}}^{\,i^{(k)}}$ onto the plane of active constraints to obtain

$$\vec{\boldsymbol{d}}^{\,i^{(k)}} = \boldsymbol{R}^{i,k} \vec{\boldsymbol{q}}^{\,i^{(k)}}.$$

Set $\alpha = 1.0$.

Step 2c: Compute

$$\vec{\boldsymbol{p}}^{\,i^{(k+1)}} = \vec{\boldsymbol{p}}^{\,i^{(k)}} + \alpha * \vec{\boldsymbol{d}}^{\,i^{(k)}}.$$

Step 2d: Compute

$$c_2 = \sum_{j=1}^{L} \vec{\boldsymbol{w}}_j^{\,i^{(k)}} \ln \vec{\boldsymbol{p}}_j^{\,i^{(k+1)}} - \beta \sum_{m \in \eta_i} V_m(\vec{\boldsymbol{p}}^{\,i^{(k+1)}}, \vec{\boldsymbol{p}}^{\,m}).$$

Step 2e: If $c_2 < c_1$ multiply α by 0.5 and go to step 2c. Otherwise go to the next pixel and perform steps 2a–e.

Step 3: Compute

$$\vec{\mu}_j^{(k+1)} = \frac{1}{\sum_{i=1}^{N} \vec{\boldsymbol{w}}_j^{\,i^{(k)}}} \sum_{i=1}^{N} \vec{\boldsymbol{w}}_j^{\,i^{(k)}} \vec{\boldsymbol{x}}_i$$

$$[(\vec{\sigma}_j)^2]^{(k+1)} = \frac{1}{\sum_{i=1}^{N} \vec{\boldsymbol{w}}_j^{\,i^{(k)}}} \sum_{i=1}^{N} \vec{\boldsymbol{w}}_j^{\,i^{(k)}} [\vec{\boldsymbol{x}}_i - \vec{\mu}_j^{(k+1)}]^2.$$

Go to Step 1.

In Step 2, $V_m(\vec{\boldsymbol{p}}^{\,i}, \vec{\boldsymbol{p}}^{\,m})$ is the potential function of the Gibbs distribution associated with both horizontal and vertical two-pixel cliques. The iteration number has been omitted from $\vec{\boldsymbol{p}}^{\,m}$ as the neighborhood set η_i could include pixels with updated label parameter vectors as well as pixels with label parameter vectors yet to be updated. Step 2c essentially implements coordinate ascent with the direction of movement determined by the projection of the gradient onto the plane of working constraints. Step 2e ensures that the function $Q_{\text{MAP}}(\Psi | \Psi^{(k)})$ does not decrease along the direction of

movement. Note that since $Q_{\mathrm{MAP}}(\Psi|\Psi^{(k)})$, as given by (31), contains separate terms involving $\vec{p}^{\,i}$ and $\{\vec{\mu}, \vec{\sigma}^2\}$, we can independently maximize with respect to $\vec{p}^{\,i}$ and $\{\vec{\mu}, \vec{\sigma}^2\}$. Step 3 is the same as in ML estimation discussed earlier, due to the assumption of a uniform prior on $\vec{\mu}$ and $\vec{\sigma}$.

VI. Experimental Results and Discussion

This section begins with a sensitivity analysis of the application of the CMM and the SVMM to pixel classification. We examine the performance of their corresponding EM algorithms with respect to the mixing weights, the mean value of each class, the variances in each class, and the number of classes in the image. We present the results from an extensive Monte Carlo simulation to compare these EM algorithms with the GEM MAP classification algorithm presented in Section V-C. Finally, we present a few examples of the application of these three algorithms to the segmentation of MRI's. All results were run in double precision code.

A. A Sensitivity Analysis of EM-CMM and ML-SVMM Algorithms in Images with No Local Correlation

This section documents the sensitivity of the EM algorithm incorporating the CMM (EM-CMM) and that incorporating the SVMM (ML-SVMM) with respect to the unknown parameters. The results of this sensitivity analysis indicate that the ML-SVMM algorithm does at least as well (typically somewhat better) than the EM-CMM algorithm and serve to justify the use of the SVMM for pixel classification.

Pseudorandom images conforming to the density function in (1) for the CMM were generated using specified $(\vec{\pi}, \vec{\mu}, \vec{\sigma})$ values by means of the following two-step procedure.

1) The *true* class of each image pixel was randomly selected from a multinomial distribution over L classes for which the probability of occurrence of class j is $\vec{\pi}_j$.
2) A pseudorandom Gaussian data value for each image pixel was generated according to its *true* class and its corresponding mean and variance. For each set of specified $(\vec{\pi}, \vec{\mu}, \vec{\sigma})$ values, ten different noisy images were generated and classified first by the EM-CMM in (3)–(6) followed by the Bayes classifier in (8)–(10), then by the ML-SVMM algorithm in (21)–(24).

1) Sensitivity to the Mixing Weights: The sensitivity of the EM-CMM and ML-SVMM algorithms to the mixing weights was examined using ten image realizations at each parameter setting over a specified range of mixing weights. The number of classes L was set at three, with class means $\vec{\mu} = (100, 140, 180)$ and standard deviations $\vec{\sigma} = (20, 20, 20)$. The mixing parameter $\vec{\pi}_1$ was then varied from 0.34–0.95 ensuring that $\Sigma_j \vec{\pi}_j = 1$ with $\vec{\pi}_j = (1 - \vec{\pi}_1)/(L-1)\ \forall j \neq 1$. Adjoining classes are separated by two standard deviations. This results in 15.87% of the pixels in a class having data values closer to the mean of each adjoining class than to the mean of its own class. Over the whole image, the mean number of pixels having data values closer to the mean of another class than its own is given by $(\vec{\pi}_1 + 2\vec{\pi}_2 + \vec{\pi}_3) \times 15.87\%$. The classification error is expressed as a percentage of the total number of pixels in the image. The results of applying the two EM algorithms (EM-CMM and ML-SVMM) are shown in Fig. 2. The following two properties are apparent in these plots.

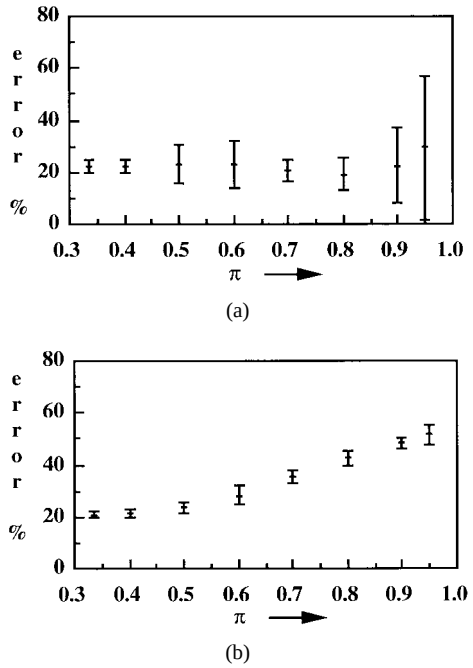


Fig. 2. Sensitivity analysis—effect of mixing weights: Percentage labeling error as $\vec{\pi}_1 = \pi$ is varied from 0.334–0.95 with $\vec{\mu} = (100, 140, 180)$, and $\vec{\sigma} = (20, 20, 20)$. (a) EM algorithm based upon the classical mixture model (EM-CMM). (b) ML EM algorithm based upon the spatially variant mixture model (ML-SVMM).

1) When an image does not have roughly the same number of pixels in each class, the EM-CMM classifier performs better than the ML-SVMM classifier.
2) The variance in the EM-CMM classifier is much larger than that in the ML-SVMM classifier.

2) Sensitivity to the Separation of Class Means: The sensitivity of the EM-CMM and ML-SVMM algorithms to the separation of the class means was examined using ten image realizations at each parameter setting over a specified range of separations. The number of classes was set at $L = 3$, with standard deviations $\vec{\sigma} = (20, 20, 20)$, and mixing weights $\vec{\pi} = (0.5, 0.25, 0.25)$. This choice of $\vec{\pi}$ results in comparable mean classification error between the two algorithms. With $\vec{\mu}_2$ and $\vec{\mu}_3$ set at 140 and 180, respectively, $\vec{\mu}_1$ was varied from 75–130. The classification error is shown in Fig. 3. As one would anticipate, the classification error increases for both algorithms as the separation between adjacent class means $\vec{\mu}_1$ and $\vec{\mu}_2$ is decreased. Again, we observe that the variance in the EM-CMM classifier is much larger than that in the ML-SVMM classifier.

3) Sensitivity to the Variance Within Each Class: The sensitivity of the EM-CMM and ML-SVMM algorithms to the variances within each class was examined using ten image realizations at each parameter setting over a specified range of variances. The number of classes was set at $L = 3$, with mixing weights $\vec{\pi} = (0.25, 0.25, 0.5)$ and class means $\vec{\mu} = (100, 140, 180)$. Note that the class separation is $\Delta\mu = 40$. With $\vec{\sigma}_1 = 20$ and $\vec{\sigma}_2 = 20$, the variance $\vec{\sigma}_3^2$ was varied from

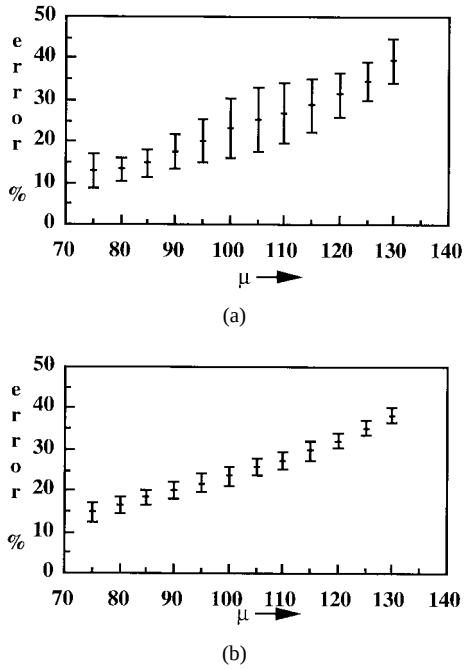


Fig. 3. Sensitivity analysis—effect of the separation of class means: Percentage classification error with $\vec{\mu}_1 = \mu$ being varied from 75 to 130, with $\vec{\mu}_2 = 140, \vec{\mu}_3 = 180, \vec{\pi} = (0.5, 0.25, 0.25)$, and $\vec{\sigma} = (20, 20, 20)$. (a) EM-CMM algorithm, and (b) ML-SVMM algorithm.

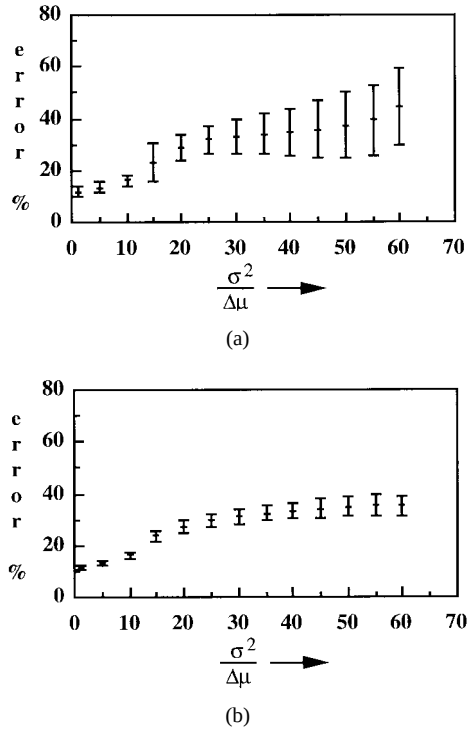


Fig. 4. Sensitivity analysis—effect of class variances: Percentage labeling error variation as $\vec{\sigma}_3^2 = \sigma^2$ is varied from 50–2400, with $\vec{\sigma}_1 = \vec{\sigma}_2 = 20$ and $\vec{\pi} = (0.25, 0.25, 0.5)$. (a) EM-CMM algorithm, and (b) ML-SVMM algorithm.

50–2400 so that the ratio $\vec{\sigma}_3/\Delta\mu$ varied from 0.177–1.223. The corresponding classification errors for both algorithms (EM-CMM and ML-SVMM) are shown in Fig. 4. As expected the classification error from both algorithms increases as the ratio $\vec{\sigma}_3/\Delta\mu$ increases. Comparable mean classification error rates are observed, but once again the variance in classification error is considerably larger for the EM-CMM algorithm than for the ML-SVMM algorithm.

4) Sensitivity to the Number of Classes: Finally, the sensitivity of the EM-CMM and ML-SVMM algorithms to the number of classes in the image was examined using ten image realizations at each parameter setting over a specified range of the number of classes from $L = 3$ to $L = 8$. The mixing weights were set at $\vec{\pi}_j = 1/L \ \forall j$, and $\vec{\mu}_j = j \times 100$ for $j = 1, \cdots, L$. If the separation of the class means is large in comparison to the class variances, both algorithms will perform extremely well and both will be insensitive to the number of classes. Therefore, the sensitivity to number of classes was examined at two different variances, $\vec{\sigma}_j^2 = 625 \ \forall j$ and $\vec{\sigma}_j^2 = 2000 \ \forall j$. The resulting classification errors are depicted in Fig. 5. As can be seen, the classification error for ML-SVMM remains fairly constant as the number of labels is increased. However, the classification error increases for EM-CMM. This indicates that the ML-SVMM algorithm is, to some degree, insensitive to the number of inherent pixel classes. Based on our experiments, our observation is that the ML-SVMM algorithm yields less error than the EM-CMM algorithm as long as the different pixel classes are somewhat equally represented. If the pixels in the observed image arise predominantly from one class, then both algorithms result in approximately the same classification error. The results of the sensitivity tests illustrate that the mean results from the ML-SVMM algorithm are comparable to those of the EM-CMM algorithm. The ML-SVMM algorithm yields a marked lower error variance as exhibited in Figs. 2–5.

B. A Comparison of EM-CMM, ML-SVMM, and GEM MAP Algorithms in Images with Local Spatial Correlation

The EM-CMM and ML-SVMM algorithms are based upon a model wherein image pixels are statistically independent. Typical real-world images involve a high degree of local spatial correlation. It is this correlation that is modeled within the *a priori* Gibbs distribution. With this in mind, we have performed a second set of Monte Carlo simulations to compare the performance of the GEM MAP classification algorithm presented in Section V-C with the EM-CMM and ML-SVMM classification algorithms. The 114×102 segmented image of a residential scene in Fig. 6 shows six classes. The number of pixels in each class and the respective class means are tabulated in Table I.

The class variances were all simultaneously varied from 200–3000. At each set of parameter values, ten noisy image realizations were generated using the true classified image of Fig. 6. The pixels in the resulting noisy images were classified using the EM-CMM, ML-SVMM, and GEM MAP algorithms. For the GEM MAP algorithm, the sum-squared-error prior described in Section V was used with an empirically determined β value of 100. The classification error from these three algorithms is plotted in Fig. 7 as a function of $\vec{\sigma}_j/\Delta\mu$. For comparison we incorporated the true values for $\vec{\pi}, \vec{\mu}$, and $\vec{\sigma}$ in a Bayes classifier of (8)–(10). The resulting classification error represents the minimum Bayes error (disregarding context or

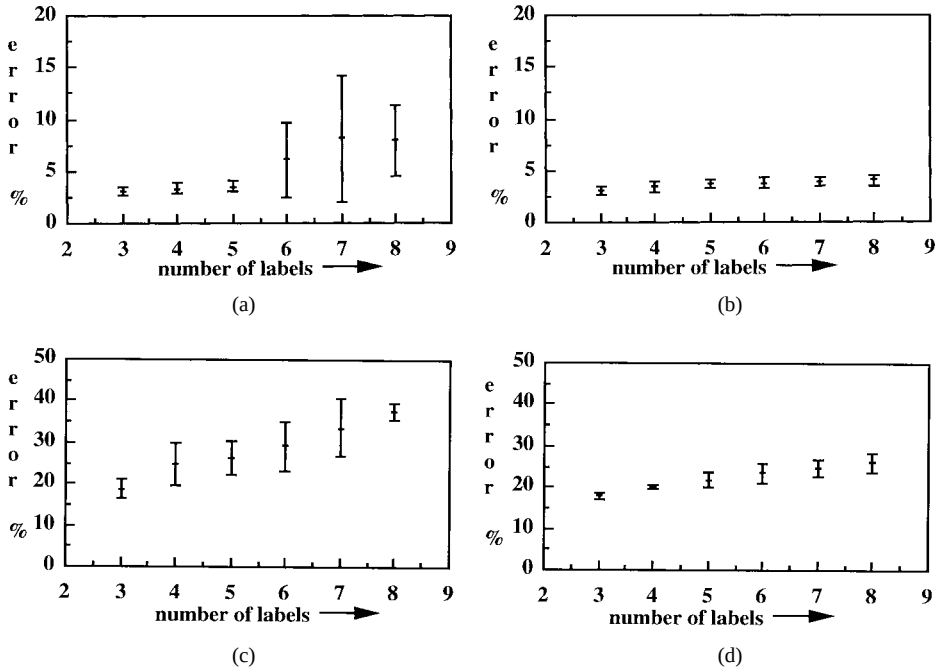


Fig. 5. Sensitivity analysis—effect of the number of classes L: Percentage classification error as L is varied from $L = 3$ to $L = 8$, with $\vec{\pi}_j = 1/L \ \forall j$, and $\vec{\mu}_j = j * 100$ for $j = 1, \cdots, L$. (a) EM-CMM algorithm with $\vec{\sigma}_j^2 = 625 \ \forall j$. (b) ML-SVMM algorithm with $\vec{\sigma}_j^2 = 625 \ \forall j$. (c) EM-CMM algorithm with $\vec{\sigma}_j^2 = 2000 \ \forall j$. (d) ML-SVMM algorithm with $\vec{\sigma}_j^2 = 2000 \ \forall j$.

Fig. 6. Spatially correlated image containing six classes described in Table I.

TABLE I
SIX PIXEL CLASSES, THE MEAN VALUE FOR EACH CLASS, AND THE PORTION OF THE IMAGE PIXELS THAT ARE IN EACH CLASS CONTAINED IN THE IMAGE IN FIG. 6

label	mean	portion of image
Windows	100	0.154558
Sky	200	0.174104
Car	300	0.126335
Lawn	400	0.187738
House	500	0.234649
Trees	600	0.122616

neighbor relationships) and is also shown in Fig. 7 along with the errors that result from the use of a simple K-means algorithm. As can be seen from Fig. 7, EM-CMM and ML-SVMM yield more or less the same percentage classification error. The GEM MAP algorithm consistently yields lower classification error as the class variances are increased. Fig. 8 shows a noisy data image from Fig. 6 and the corresponding classified images pixels using the EM-CMM, ML-SVMM, and GEM MAP algorithms. The EM-CMM and ML-SVMM algorithms yield comparable results in this example. The labeling errors are spread evenly throughout the image. The GEM MAP algorithm yields considerably lower classification error, with each class comprising more homogeneous regions. The benefits from the *a priori* model are more pronounced at high noise levels.

C. An Application to Computed Tomography: An Experimental Phantom Study

For a visual evaluation of the segmentations obtained using the EM-CMM, ML-SVMM, and GEM MAP algorithms we have performed an experimental computed tomography (CT) phantom study. This experimental study was performed using a thorax phantom which included objects representing various anatomical structures such as the heart, lungs, spinal cord, liver, and chest wall. These structures were all represented in one or more of the 103 slices obtained in the phantom study. Each reconstructed slice consisted of 512×512 pixels. Fig. 9 shows one such reconstructed CT slice along with the segmentation results obtained with EM-CMM, ML-SVMM, and GEM MAP algorithms. Fig. 9(a) illustrates the different

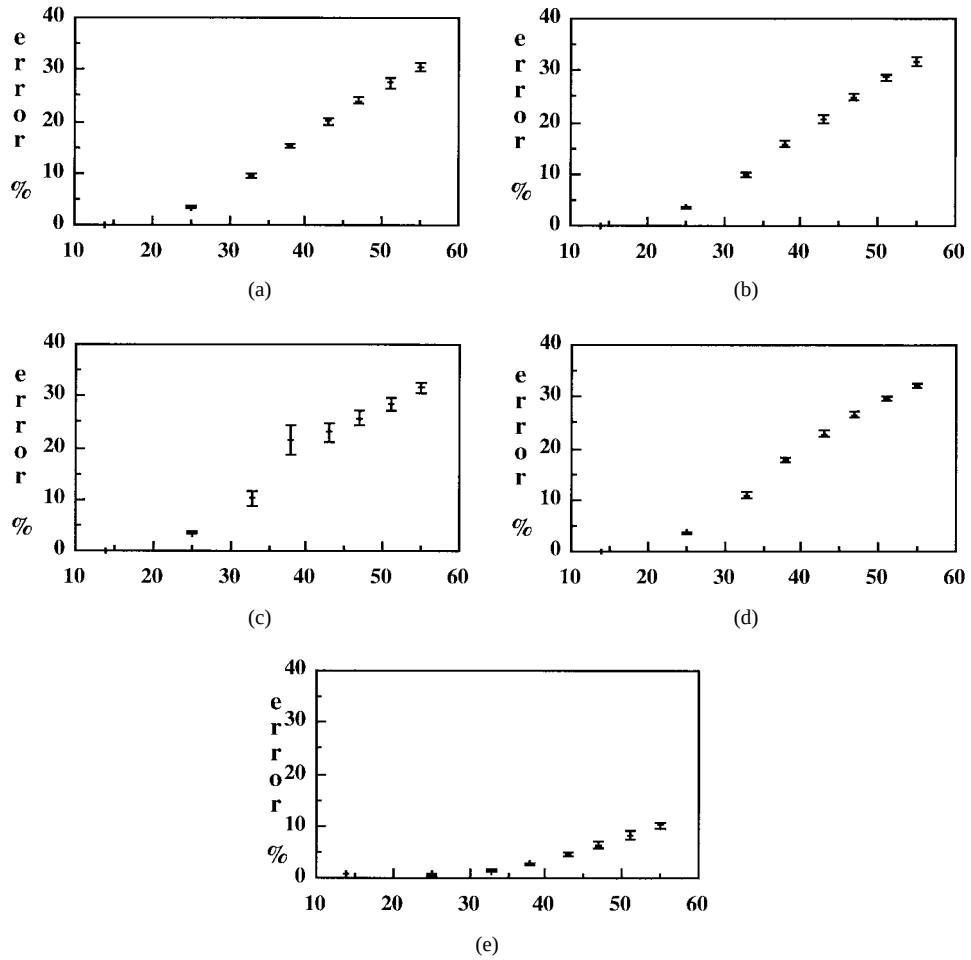


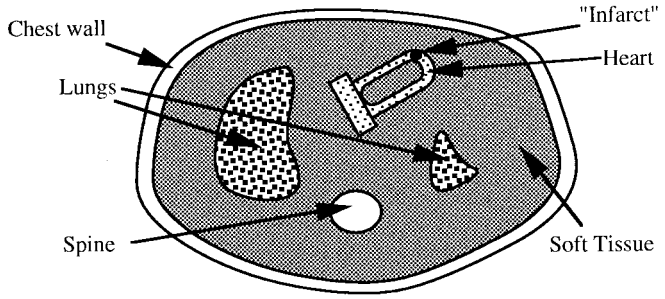
Fig. 7. Sensitivity to class variances within an image with local spatial correlation (using the segmented image in Fig. 6). Percentage classification error plotted with respect to $\sigma/\Delta\mu \times 100$ as all class variances $\vec{\sigma}_j^2$ are varied from 200–3000 with $\vec{\mu}_j = j \times \Delta\mu \ \forall\, j$ where $\Delta\mu = 100$ for (a) Bayes classifier using true values for $\vec{\mu}_j$ and $\vec{\sigma}_j^2$; (b) K-Means algorithm; (c) EM-CMM algorithm; (d) ML-SVMM algorithm; (e) GEM MAP algorithm with $\beta = 100$.

Fig. 8. Spatially correlated images containing six classes: noisy image generated using the *true* label image shown in Fig. 6 with $\vec{\sigma}_j^2 = 2000 \ \forall j$ (top left), the labeling from the EM-CMM algorithm (top right), the labeling from the ML-SVMM algorithm (bottom left), and the labeling from the GEM MAP algorithm using $\beta = 100.0$ (bottom right).

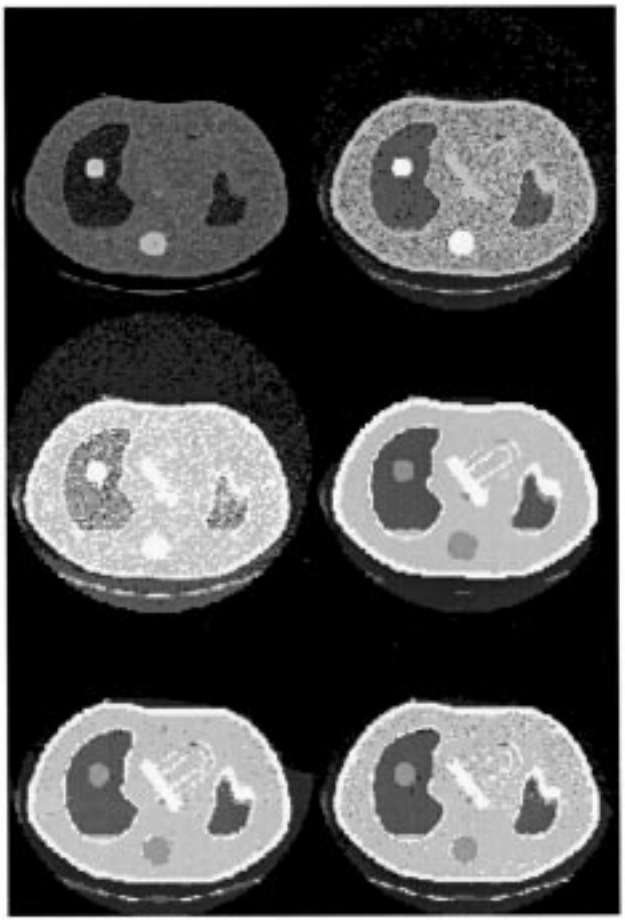
structures that can be visualized in this slice. For the GEM MAP algorithm we show results based on 200 iterations using the sum-squared-error prior (29) with β values of 0.5, 1, and 10. The per iteration computation time on a DEC Alphastation (series 500) was 8.6 s. As can be seen from Fig. 9, the EM-CMM and ML-SVMM algorithms yield somewhat comparable results. As expected, the GEM MAP classification algorithm yields superior segmentation compared to EM-CMM and ML-SVMM algorithms. In particular, note that the "infarct" or occlusion in the heart is more clearly delineated in the segmentation resulting from the use of the GEM MAP algorithm.

D. An Application to MRI's

In this section, we present results from applying the EM-CMM, ML-SVMM, and the GEM MAP algorithms for segmenting brain MRI's. Fig. 10 shows a T_2-weighted MRI slice of the brain of a 28-year old normal male subject. T_2-weighted MRI images differentiate between three classes of tissue (bone marrow, white matter, grey matter/cerebro spinal fluid) plus a class for air. Also shown are the labels estimated using the EM-CMM and ML-SVMM algorithms. The GEM MAP algorithm of Section V-C was applied with a sum-squared-error prior (29) and empirically determined β values of 0.001, 0.1, and 1.0. These values were chosen based on



(a)



(b)

Fig. 9. Experimental CT phantom study. (a) Illustration of the different structures of the thorax phantom. (b) Top left: CT slice. Top right: segmentation from the EM-CMM algorithm. Middle left: segmentation from the ML-SVMM algorithm. Middle right: segmentation from the GEM MAP algorithm using $\beta = 10.0$. Bottom left: segmentation from the GEM MAP algorithm using $\beta = 1.0$. Bottom right: segmentation from the GEM MAP algorithm using $\beta = 0.5$.

visual evaluation of the resulting segmentations, which are also shown in Fig. 10. Close examination of the resulting images leads to the following observations.

1) EM-CMM and ML-SVMM algorithms tend to yield erroneous labels along the boundaries of the grey matter/csf and the white matter.
2) The GEM MAP algorithm gives acceptable labeling results over a wide range of β values (0.001–1.0).
3) The GEM MAP algorithm yields more homogeneous class regions separated by a crisper boundary. Note the clear delineation of the space between skull and surface of the brain in the labeled images from the GEM MAP algorithm.

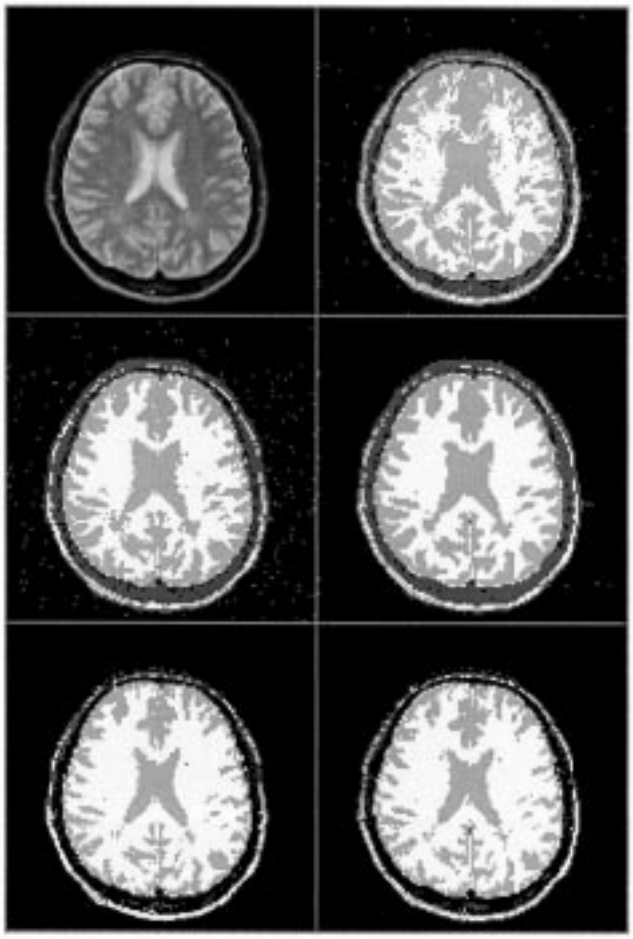


Fig. 10. Segmentation of an MRI image. Top left: MRI T_2-weighted image. Top right: segmentation from the EM-CMM algorithm. Middle left: segmentation from the ML-SVMM algorithm. Middle right: segmentation from the GEM MAP algorithm using $\beta = 1$. Bottom left: segmentation from the GEM MAP algorithm using $\beta = 0.1$. Bottom right: segmentation from the GEM MAP algorithm using $\beta = 0.001$.

E. Practical Aspects

There are several practical issues regarding implementation of the classification algorithms presented here. The first issue is initialization. The *a posteriori* functions based upon the prior models presented have multiple local maxima. Since EM algorithms have convergence to local (not global) maxima, initialization is very important. Our experience suggests several simple rules for initializing the parameters to be estimated such that convergence to an undesirable local maximum can be avoided.

1) **Initializing the mixture weights**. *In the absence of any a priori information*, for the EM-CMM algorithm set all mixture weights to $\vec{\pi}_j = 1/L$, for the ML-SVMM algorithm set all elements of $\vec{\boldsymbol{p}}^i$ to $\vec{\boldsymbol{p}}^i_j = 1/L$ where L is the number of classes. If prior information is available regarding the mixture weights, it can be used to suggest other appropriate methods for initializing the

mixture weights. More importantly, such prior information should be incorporated into the solution through specification of prior density functions for the weights.

2) **Initializing the class means**. It is anticipated that the classes in an image have means that are separated from each other and that those means somewhat span the dynamic range in the image. One method for initializing the means that works well is to form a coarse histogram of $4L$ quantization levels over the range of data values, $xmin$ to $xmax$. Then initialize the L means to the L peaks in the coarse histogram. This ensures that the initial means are separated by at least $(xmax - xmin)/(4L)$. It is not advised to initialize all class means to values at or near the peak value in a histogram of the data image, as this seems to often lead to convergence to an undesirable local maximum wherein one of the classes is estimated as having a large mixture weight and a large class variance. Again, if prior information regarding the class means is available, it should be incorporated in a prior density function.
3) **Initializing the class variances**. To avoid convergence to an undesirable local maximum, initialize the variances within each class to a large value. The range of the data is helpful in determining a suitable large initial value. Where there are three or more classes in the data, set the initial variances to $\sigma = 0.1\ (xmin - xmax)^2$.

One should avoid using initializations $\vec{\pi}_j^{(0)} = (1/L), \vec{\mu}_1^{(0)} = \vec{\mu}_2^{(0)} = \cdots = \vec{\mu}_L^{(0)}$ and $\vec{\sigma}_1^{(0)} = \vec{\sigma}_2^{(0)} = \cdots = \vec{\sigma}_L^{(0)}$ all at the same time, as this is a fixed point of the EM-CMM algorithm [18] as well as the ML-SVMM algorithm. This is easily seen from (22)–(24). Also, in the general case, we recommend not biasing the initial mixture weights and variances toward particular values *unless prior information is available*.

The second issue in applying finite mixture models is whether to implement the resulting algorithms in single or double precision. Our experience is that the parameter estimates and the pixel labels generated by these algorithms are quite insensitive to whether or not double precision is used throughout. However, if one desires to monitor the likelihood function at each iteration, double precision arithmetic is definitely required in multiplications and additions used to compute the likelihood of a set of parameters.

The third issue of practical importance is that of how many iterations are required to achieve acceptable convergence of the parameter estimates. On all examples shown here, we have run 500 iterations of the EM-CMM and ML-SVMM algorithms. This yielded parameter estimates that had converged to at least four significant digits. The pixel labels typically did not change after 300–400 iterations. Approximately 150 iterations were required for similar convergence of the GEM MAP algorithm, but each iteration required approximately three to four times more computation. We have found similar results with a variety of other real images.

The fourth issue of practical importance is the specification of the parameter β in the GEM MAP algorithm. Fig. 11 shows the typical effect of β on the classification error. For $\beta = 0$, MAP is equivalent to ML. Small positive values of β significantly reduce the classification error over ML, while very large values of β can significantly increase the classification error. This behavior is typical of any image with positive correlation. Examination of the potential function in (29) in light of the result from Appendix B shows that the contribution to Q_{MAP} by the prior is relatively independent of L. For a given application, it is recommended to initially use a small value $\beta = 0.01$; one may then compare the classification with ML-SVMM to determine if a slightly larger or smaller value for β is warranted. We are currently investigating this topic with the focus on deriving statistical [25] or constrained approaches [26].

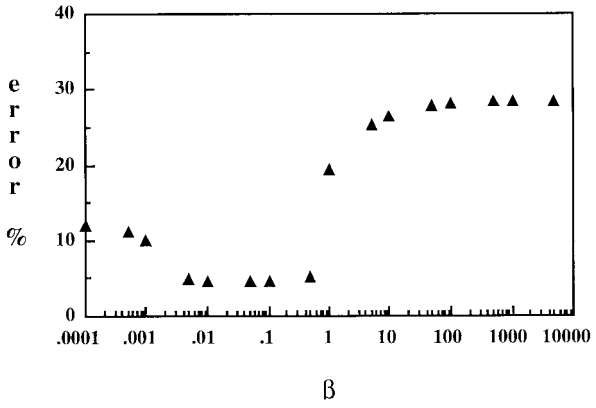


Fig. 11. Percentage classification error as β is varied from 0.0001 to 5000. To accommodate a wide range of values for β a logarithmic scale has been used for the horizontal axis.

As discussed in Section I, the algorithms described in this paper are all based on knowing L in advance. If L is not known then it should be suitably estimated using one of several well-known information theoretic approaches [13], [14].

VII. Summary and Conclusions

We have investigated the use of mixture models for pixel classification and have proposed an SVMM for the same purpose. While modifications to the CMM for image segmentation have been attempted previously, the model proposed in this paper has the important feature of incorporating a spatial variation in the mixture weights assigned to different pixel locations. Based on the assumption that the observed image pixel values are statistically independent and that the number of pixel classes is known, an EM algorithm for ML estimation of the class of each pixel has been derived for the SVMM. Comprehensive Monte Carlo simulations compare and contrast the applicability of the CMM and the SVMM for pixel labeling. From the detailed simulations of Section VI-A, we see that the use of the SVMM in an EM algorithm for pixel classification yields a classification accuracy highly comparable to one using the CMM.

We have also presented a fast generalized EM algorithm for the Bayesian estimation of pixel label parameters. This algorithm combines experimental knowledge from the observations and the researcher's insight in the form of prior densities on the label parameters. As the results in Sections VI-B–D indicate this algorithm yields a vast improvement in classification

accuracy compared to EM-CMM and ML-SVMM algorithms. We wish to point out the fact that the application of the SVMM leads to improved accuracy in pixel classification in images that exhibit spatial correlation. In images with no spatial correlation, use of the SVMM instead of the CMM is not expected to yield a vast improvement in classification accuracy. In this case, as the results of Section VI-A indicate, the SVMM yields classification accuracies highly comparable to the CMM. This is understandable because, as stated earlier, with trivial assumptions the CMM is identical to the SVMM.

It should be noted that the assumption of Gaussian component densities is not a restrictive one; the GEM MAP algorithm can easily be modified to incorporate component densities other than Gaussian. Also note that the underlying model, i.e., SVMM, for the algorithm is based on characterizing the different pixel classes by different grey levels. Although the SVMM does not model texture, it does provide a powerful means for incorporating prior spatial information about the pixel labels. It also provides an attractive alternative to some of the more complicated MRF models and their related computationally intensive algorithms. There are several instances, similar to the example presented in [5], where a simple model and algorithm would suffice and a computationally involved algorithm is not needed. In such instances, the GEM MAP algorithm can be applied to yield superior segmentation. In other cases, it can be used as a powerful preprocessor before applying other computationally involved algorithms incorporating more sophisticated models as dictated by the problem. In any case, the GEM MAP algorithm presented in this paper bridges the gap between some of the computationally involved algorithms based on random fields and simpler segmentation algorithms based on mixture models.

APPENDIX A

In this appendix, the covariance matrix for data image pixels is derived.

Consider the finite mixture density function under the assumption of Gaussian component densities

$$f(\vec{\boldsymbol{x}}_i) = \sum_{j=1}^{L} \vec{\pi}_j f_j(\vec{\boldsymbol{x}}_i|\vec{\mu}_j, \vec{\sigma}_j). \tag{41}$$

The mean $\vec{\boldsymbol{m}} = E\{\vec{\boldsymbol{x}}\}$ of this density function is given by

$$\vec{\boldsymbol{m}}_i = E\{\vec{\boldsymbol{x}}_i\} = \sum_{j=1}^{L} \vec{\pi}_j \vec{\mu}_j. \tag{42}$$

Now consider the covariance matrix $\boldsymbol{C}$ for $\vec{\boldsymbol{x}}$. For any two data values $\vec{\boldsymbol{x}}_i$ and $\vec{\boldsymbol{x}}_j$

$$\boldsymbol{C}_{i,j} = E[(\vec{\boldsymbol{x}}_i - \vec{\boldsymbol{m}}_i)(\vec{\boldsymbol{x}}_j - \vec{\boldsymbol{m}}_j)] = 0, \quad \forall i \neq j \tag{43}$$

and

$$\begin{aligned} \boldsymbol{C}_{i,i} &= E[(\vec{\boldsymbol{x}}_i - \vec{\boldsymbol{m}}_i)^2] = E[\vec{\boldsymbol{x}}_i^2] - \vec{\boldsymbol{m}}_i^2 \\ &= \int_{-\infty}^{\infty} \vec{\boldsymbol{x}}_i^2 \sum_j \vec{\pi}_j f_j(\vec{\boldsymbol{x}}_i|\vec{\mu}_j, \vec{\sigma}_j)\, d\vec{\boldsymbol{x}} - \vec{\boldsymbol{m}}_i^2 \end{aligned} \tag{44}$$

from which

$$\boldsymbol{C}_{i,i} = \sum_{j=1}^{L} \vec{\pi}_j(\vec{\sigma}_j^2 + \vec{\mu}_j^2) - \left(\sum_{j=1}^{L} \vec{\pi}_j \vec{\mu}_j\right)^2. \tag{45}$$

APPENDIX B

Lemma: Given the spatially variant finite mixture model

$$\prod_{i=1}^{N} \sum_{j=1}^{L} \vec{\boldsymbol{p}}_j^i f_j(\vec{\boldsymbol{x}}_i|\vec{\theta}^j)$$

there exists a ML estimate $\hat{\boldsymbol{p}}_j^{i^{\mathrm{ML}}}$ (probability of the ith pixel belonging to the jth class) that both maximizes this model and additionally satisfies

$$\hat{\boldsymbol{p}}_j^{i^{\mathrm{ML}}} \in \{0, 1\}. \tag{46}$$

Proof: Consider the ML estimation problem of maximizing the likelihood function

$$\max_{\vec{\theta}^1 \ldots \vec{\theta}^L, \vec{\boldsymbol{p}}^1 \ldots \vec{\boldsymbol{p}}^N} \prod_{i=1}^{N} \sum_{j=1}^{L} \vec{\boldsymbol{p}}_j^i f_j(\vec{\boldsymbol{x}}_i|\vec{\theta}^j). \tag{47}$$

Let the individual component densities $f_j(\vec{\boldsymbol{x}}_i|\vec{\theta}^j)$ be bounded for all feasible values of $\vec{\theta}^j$ and $\vec{\boldsymbol{x}}_i$. The product in (47) is then maximized if each of the terms $\Sigma_{j=1}^{L} \vec{\boldsymbol{p}}_j^i f_j(\vec{\boldsymbol{x}}_i|\vec{\theta}^j)$ is maximized. Any feasible estimate $\hat{\theta}^1 \cdots \hat{\theta}^L$ and $\hat{\boldsymbol{p}}^1 \cdots \hat{\boldsymbol{p}}^N$, by definition satisfies $0 \leq \hat{\boldsymbol{p}}_j^i \leq 1$ and $\Sigma_{j=1}^{L} \hat{\boldsymbol{p}}_j^i = 1$ and $0 \leq f_j(\vec{\boldsymbol{x}}_i|\hat{\theta}^j)$, so that

$$\sum_{j=1}^{L} \hat{\boldsymbol{p}}_j^i f_j(\vec{\boldsymbol{x}}_i|\hat{\theta}^j) \leq \max_j f_j(\vec{\boldsymbol{x}}_i|\hat{\theta}^j). \tag{48}$$

A global maximum of the single term $\Sigma_{j=1}^{L} \hat{\boldsymbol{p}}_j^i f_j(\vec{\boldsymbol{x}}_i|\hat{\theta}^j)$ with respect to $\{\hat{\boldsymbol{p}}_j^i\}$ occurs where (48) holds with equality. This includes the case where $\hat{\theta}^1 \cdots \hat{\theta}^L$ are precisely ML estimates $\hat{\theta}^{1^{\mathrm{ML}}} \cdots \hat{\theta}^{L^{\mathrm{ML}}}$. One manner in which (48) will hold with equality is where

$$\hat{\boldsymbol{p}}_j^i = \begin{cases} 1, & \text{where } j = \arg\max_l f_l(\vec{\boldsymbol{x}}_i|\hat{\theta}^{i^{\mathrm{ML}}}) \\ 0, & \text{otherwise.} \end{cases} \tag{49}$$

These $\{\hat{\boldsymbol{p}}_j^i\}$ constitute ML estimates $\hat{\boldsymbol{p}}^{i^{\mathrm{ML}}}$ that together with $\hat{\theta}^{1^{\mathrm{ML}}} \cdots \hat{\theta}^{L^{\mathrm{ML}}}$ form the global maximum of (47). These estimates $\hat{\boldsymbol{p}}_j^{i^{\mathrm{ML}}} \in \{0, 1\}$ are unique as long as the

$$\arg\max_l f_l(\vec{\boldsymbol{x}}_i|\hat{\theta}^{i^{\mathrm{ML}}})$$

is unique.

$$\boldsymbol{R}^{i,k} = \frac{1}{L-2}\begin{bmatrix} (L-3) & -1 & -1 & \cdot & \cdot & -1 & 0 & -1 & 0 \\ -1 & (L-3) & -1 & \cdot & \cdot & -1 & 0 & -1 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ -1 & -1 & -1 & \cdot & \cdot & (L-3) & 0 & -1 & 0 \\ 0 & 0 & 0 & \cdot & \cdot & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & \cdot & \cdot & -1 & 0 & (L-3) & 0 \\ 0 & 0 & 0 & \cdot & \cdot & 0 & 0 & 0 & 0 \end{bmatrix}$$

APPENDIX C

In this appendix, we derive the expression for $E\{\vec{z}_j^i|\vec{x}_i,\Psi\}$ in (21).

Where vector Ψ is defined as $\Psi \equiv [\vec{p}^1 \cdots \vec{p}^N, \vec{\theta}^1 \cdots \vec{\theta}^L]$

$$f(\vec{z}^1 \cdots \vec{z}^N|\vec{x},\Psi) = \frac{f(\vec{z}^1 \cdots \vec{z}^N, \vec{x}|\vec{p}^1 \cdots \vec{p}^N, \vec{\theta}^1 \cdots \vec{\theta}^L)}{f(\vec{x}|\vec{p}^1 \cdots \vec{p}^N, \vec{\theta}^1 \cdots \vec{\theta}^L)}. \tag{50}$$

Using (13) and (19)

$$f(\vec{z}^1 \cdots \vec{z}^N|\vec{x},\Psi) = \prod_i \frac{\prod_j [\vec{p}_j^i f_j(\vec{x}_i|\vec{\theta}^j)]^{\vec{z}_j^i}}{\sum_k \vec{p}_k^i f_k(\vec{x}_i|\vec{\theta}^k)} \tag{51}$$

whereby

$$f(\vec{z}^i|\vec{x}_i,\Psi) = \frac{\prod_j [\vec{p}_j^i f_j(\vec{x}_i|\vec{\theta}^j)]^{\vec{z}_j^i}}{\sum_k \vec{p}_k^i f_k(\vec{x}_i|\vec{\theta}^k)}. \tag{52}$$

Therefore, the conditional expectation of the jth element $\vec{z}_j^i$ is given by

$$\begin{aligned} E\{\vec{z}_j^i|\vec{x}_i,\Psi\} &= \sum_{\{\vec{z}_j^i\}} \vec{z}_j^i f(\vec{z}_j^i|\vec{x}_i,\vec{p}^i\vec{\theta}^1 \cdots \vec{\theta}^L) \\ &= 1 * f(\vec{z}_j^i = 1|\vec{x}_i,\vec{p}^i,\vec{\theta}^1 \cdots \vec{\theta}^L) \\ &\quad + 0 * f(\vec{z}_j^i = 0|\vec{x}_i,\vec{p}^i,\vec{\theta}^1 \cdots \vec{\theta}^L) \\ &= \frac{\vec{p}_j^i f_j(\vec{x}_i|\vec{\theta}^j)}{\sum_{l=1}^{L} \vec{p}_l^i f_l(\vec{x}_i|\vec{\theta}^l)} \equiv \vec{w}_j^i \quad \text{using (52).} \end{aligned} \tag{53}$$

For the case of Gaussian component densities

$$E\{\vec{z}_j^i|\vec{x}_i,\Psi\} = \frac{\vec{p}_j^i \frac{1}{\sqrt{2\pi\vec{\sigma}_j^2}} \exp\left[\frac{(\vec{x}_i - \vec{\mu}_j)^2}{2\vec{\sigma}_j^2}\right]}{\sum_{l=1}^{L} \vec{p}_l^i \frac{1}{\sqrt{2\pi\vec{\sigma}_l^2}} \exp\left[\frac{(\vec{x}_i - \vec{\mu}_l)^2}{2\vec{\sigma}_l^2}\right]}. \tag{54}$$

APPENDIX D

In this appendix, the form of the gradient projection matrix is derived.

At the kth iteration of the GEM MAP algorithm, the general form of the matrix $\boldsymbol{R}^{i,k}$ for projecting the gradient vector $\vec{\boldsymbol{q}}^{i^{(k)}}$ (corresponding to the ith parameter vector $\vec{\boldsymbol{p}}^{i^{(k)}}$) onto the plane of constraints [24] is

$$\boldsymbol{R}^{i,k} = [\boldsymbol{I} - \boldsymbol{A}^{i^T}(\boldsymbol{A}^i\boldsymbol{A}^{i^T})^{-1}\boldsymbol{A}^i]$$

where $\boldsymbol{A}^i$ is a $M * L$ matrix composed of M rows of the M constraints that are active for the current $\vec{\boldsymbol{p}}^{i^{(k)}}$. Recall that the constraints are: (a) $\Sigma_j \vec{\boldsymbol{p}}_j^i = 1$; and (b) $\vec{\boldsymbol{p}}_j^i \geq 0$ for all i,j. Constraint (a) is always active. First, let the only active constraint be (a). In this case, $\boldsymbol{A}^i$ is a $1 * L$ matrix given by

$$\boldsymbol{A}^i = [1 \cdots 1]$$

so that

$$\boldsymbol{R}^{i,k} = \frac{1}{L}\begin{bmatrix} (L-1) & -1 & \cdot & -1 \\ -1 & \cdot & & \cdot \\ \cdot & & \cdot & -1 \\ -1 & \cdot & -1 & (L-1) \end{bmatrix}.$$

Now, e.g., let two additional constraints be active:

$$\vec{\boldsymbol{p}}_L^i = 0 \quad \text{and} \quad \vec{\boldsymbol{p}}_{L-2}^i = 0.$$

Then $\boldsymbol{A}^i$ is a $3 * L$ matrix given by

$$\boldsymbol{A}^i = \begin{bmatrix} 1 & \cdot & \cdot & \cdot & \cdot & 1 \\ 0 & \cdot & 0 & 1 & 0 & 0 \\ 0 & \cdot & \cdot & \cdot & 0 & 1 \end{bmatrix}$$

so that we obtain the expression shown at the top of the page. Generalizing, let K denote the number of nonzero entries in $\vec{\boldsymbol{p}}^i$ with $K \leq L$. The general form of the projection matrix for the specific application of maximization in the M-step is

$$\boldsymbol{R}_{j,l}^{i,k} = \begin{cases} 0, & \text{if } \vec{\boldsymbol{p}}_j^i = 0 \text{ or } \vec{\boldsymbol{p}}_l^i = 0 \\ \dfrac{K-1}{K}, & \text{if } j = l \text{ and } \vec{\boldsymbol{p}}_j^i \neq 0 \\ -\dfrac{1}{K}, & \text{otherwise.} \end{cases}$$

ACKNOWLEDGMENT

The authors are thankful to Dr. P. Bland, Department of Radiology, University of Michigan, for providing the experimental CT phantom study, and to Dr. C. Lin and the Medical Physics Department of Methodist Hospital for providing the MRI images used in the evaluation of the pixel classification algorithms.

REFERENCES

[1] A. Rosenfeld amd A. C. Kak, *Digital Picture Processing*. New York: Academic, 1982.
[2] R. A. Hummel and S. W. Zucker, "On the foundations of relaxation labeling processes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, pp. 267–287, 1983.
[3] A. Rosenfeld, R. A. Hummel, and S. W. Zucker, "Scene labeling by relaxation operations," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, pp. 420–433, 1976.
[4] S. Lakshmanan and H. Derin, "Simultaneous parameter estimation and segmentation of Gibbs random fields using simulated annealing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 799–813, 1989.
[5] B. S. Manjunath and R. Chellappa, "Unsupervised texture segmentation using Markov random field models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 478–482, 1991.
[6] H. Derin and H. Elliot, "Modeling and segmentation of noisy and textured images using Gibbs random fields," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, pp. 39–55, 1987.
[7] C. Bouman and B. Liu, "Multiple resolution segmentation of textured images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 99–113, 1991.
[8] D. M. Titterington, A. F. M. Smith, and U. E. Makov, *Statistical Analysis of Finite Mixture Distributions*. New York: Wiley, 1985.
[9] H. S. Choi, D. R. Haynor, and Y. Kim, "Partial volume tissue classification of multichannel magnetic resonance images—A mixel model," *IEEE Trans. Med. Imag.*, vol. 10, pp. 395–407, 1991.
[10] P. Santago and H. D. Gage, "Statistical models of partial volume effect," *IEEE Trans. Image Processing*, vol. 4, pp. 1531–1540, 1995.
[11] ———, "Quantification of MR brain images by mixture density and partial volume modeling," *IEEE Trans. Med. Imag.*, vol. 12, pp. 566–574, 1993.
[12] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data," *J. R. Stat. Soc.*, vol. 39, pp. 1–38, 1977.
[13] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Automat. Contr.*, vol. AC-19, pp. 716–723, 1974.
[14] J. Zhang and J. W. Modestino, "A model-fitting approach to cluster validation with application to stochastic model-based image segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 1009–1017, 1990.
[15] S. L. Sclove, "Application of the conditional population-mixture model to image segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, pp. 428–433, 1983.
[16] Z. Liang, J. R. MacFall, and D. P. Harrington, "Parameter estimation and tissue segmentation from multispectral MR images," *IEEE Trans. Med. Imag.*, vol. 13, pp. 441–449, 1994.
[17] Z. Liang, R. J. Jaszczak, and R. E. Coleman, "On reconstruction and segmentation of piecewise continuous images," in *Proc. IPMI*, vol. 12, pp. 94–104, 1991.
[18] ———, "Parameter estimation of finite mixtures using the EM algorithm and information criteria with application to medical image processing," *IEEE Trans. Nucl. Sci.*, vol. 39, pp. 1126–1133, 1992.
[19] D. M. Titterington, "Some recent research in the analysis of mixture distributions," *Statistics*, vol. 21, pp. 619–641, 1990.
[20] J. A. Fessler and A. O. Hero, III, "Penalized maximum-likelihood image reconstruction using space-alternating generalized EM algorithms," *IEEE Trans. Image Processing*, vol. 4, pp. 1417–1429, 1995.
[21] T. J. Hebert and K. Lu, "Expectation maximization algorithms, null spaces, and MAP image restoration," *IEEE Trans. Image Processing*, vol. 4, pp. 1084–1095, 1995.
[22] S. Geman and S. D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol PAMI-6, pp. 721–741, 1984.
[23] T. Hebert and R. Leahy, "A generalized EM algorithm for 3-d bayesian reconstruction from poisson data using Gibbs priors," *IEEE Trans. Med. Imag.*, vol. 8, pp. 194–202, 1989.
[24] D. G. Luenberger; *Linear and Nonlinear Programming*. Reading, MA: Addison-Wesley, 1984.
[25] T. Hebert and R. Leahy, "Statistic-based MAP image restoration from Poisson data using Gibbs priors," *IEEE Trans Signal Processing*, vol 40, pp. 2290–2303, Sept. 1992.
[26] V. Mesarovic, N. Galatsanos, and A. Katsaggelos, "Regularized constrained total least squares image restoration," *IEEE Trans. Image Processing*, vol 4, pp. 1096–1108, Aug 1995.

S. Sanjay-Gopal received the B.E. degree in electronics and communication engineering from the College of Engineering, Guindy, Madras, India, in 1989, and the M.S. degree in biomedical engineering and the Ph.D. degree in electrical engineering in 1991 and 1996, respectively, both from the University of Houston, Houston, TX.

He is currently with the Department of Radiology, University of Michigan Medical Center, Ann Arbor. His research interests include statistical image processing, automated image analysis, pattern recognition, computer vision, computer-aided diagnosis, and neural networks.

Thomas J. Hebert (S'84–M'89) received the B.S. degree in electrical engineering from the University of Florida, Gainesville, in 1984, and the M.S. and Ph.D. degrees in electrical engineering from the University of Southern California (USC), Los Angeles, in 1986 and 1989, respectively.

From 1984 through 1988, he was an American Electronics Association Fellow with the Signal and Image Processing Institute, USC. He joined the Department of Electrical and Computer Engineering, University of Houston, Houston, TX, in 1989, where he is currently an Associate Professor. He is also Adjunct Professor with the Department of Radiology, Baylor College of Medicine, Houston. His research interests are in the application of statistics and estimation theory in image processing and medical imaging.