A Multiscale Random Field Model for Bayesian Image Segmentation

Charles A. Bouman, *Member, IEEE*, and Michael Shapiro

Abstract—**Many approaches to Bayesian image segmentation have used maximum *a posteriori* (MAP) estimation in conjunction with Markov random fields (MRF). Although this approach performs well, it has a number of disadvantages. In particular, exact MAP estimates cannot be computed, approximate MAP estimates are computationally expensive to compute, and unsupervised parameter estimation of the MRF is difficult.**

In this paper, we propose a new approach to Bayesian image segmentation that directly addresses these problems. The new method replaces the MRF model with a novel multiscale random field (MSRF) and replaces the MAP estimator with a sequential MAP (SMAP) estimator derived from a novel estimation criteria. Together, the proposed estimator and model result in a segmentation algorithm that is not iterative and can be computed in time proportional to MN where M is the number of classes and N is the number of pixels. We also develop a computationally efficient method for unsupervised estimation of model parameters.

Simulations on synthetic images indicate that the new algorithm performs better and requires much less computation than MAP estimation using simulated annealing. The algorithm is also found to improve classification accuracy when applied to the segmentation of multispectral remotely sensed images with ground truth data.

I. INTRODUCTION

HARALICK and Shapiro have suggested that a good segmentation of a image should separate the image into simple regions with homogeneous behavior [1]. In recent years, many authors have used Bayesian estimation techniques as a framework for computing segmentations that best compromise between these two opposing objectives [2]–[4]. These methods model the shape of segmented regions in addition to the behavior of pixels in each homogeneous region. The segmentation is then computed by estimating the best label for each pixel.

A number of estimation techniques and region models have been used for the Bayesian segmentation problem. Typically, the labels of image pixels are modeled as a Markov random field (MRF) or, equivalently, as a Gibbs distributions [5]. These models are used because they only require the specification of spatially local interactions using a set of local parameters. This is important since spatially local interactions result in segmentation algorithms that only require local computations. Most often, the image is then segmented by approximately computing the maximum *a posteriori* (MAP) estimate of the pixel labels.

These statistical approaches to segmentation provide an important framework and have improved results in the application of segmentation to natural scenes [6], tomographic cross sections [7], texture images [2], and multispectral remotely sensed images [8]–[11]. However, the approach has a number of important disadvantages.

Computing the MAP estimate requires the minimization of a discrete functional with many local minima. Exact minimization is intractable; therefore, methods for approximately minimizing the true MAP estimate must be used. These methods include simulated annealing [12], greedy minimization [3], dynamic programming [4], and multiresolution minimization [13]–[16]. However, all of these approaches require approximations in the 2-D case and are either iterative or very computationally expensive.

The MRF model has a limited ability to describe large scale behaviors. For example, we may know that segmented regions are likely to be at least 50 pixels wide. However, it is difficult to accurately incorporate this information by specifying the interactions of adjacent pixels. The model can be improved by using a larger neighborhood for each pixel, but this rapidly increases the number of parameters of interaction and the complexity of the segmentation algorithms. The fundamental limitation of local models is that they do not allow behavior to be directly controlled at different spatial scales. This is of critical importance since scale variation occurs naturally in images and is important in quantifying image behavior [17], [18].

The MAP estimate does not have desirable properties for the segmentation problem [19], [20]. The MAP estimate minimizes the probability that any pixel in the image will be misclassified. This is an excessively conservative criteria since any segmentation algorithm is likely to result in some misclassified pixels. In practice, it has been noted that MAP estimation has some undesirable global properties that may actually make an approximate minimization more desirable [3], [20]. For example, in multiresolution segmentation, MRF correlations parameters were found to *increase* at coarser scales [14], [15]. This is counter to the physical intuition that coarser sampling should produce less correlation.

The maximizer of the posterior marginals (MPM) estimator has been suggested [19] as an alternative to MAP estimation since it minimizes the probability of classification error. However, it may only be approximately computed in

---

Manuscript received June 14, 1992; revised March 18, 1993. The associate editor coordinating the review of this paper and approving it for publication was Dr. Michael Unser. This work was supported by the US Army Construction Engineering Research Laboratory, grant number DACA8890D0029 and an NEC faculty Fellowship.

C. A. Bouman is with the School of Electrical Engineering, Purdue University, West Lafayette, IN 47907-0501.

M. Shapiro is with the US Army Construction Engineering Research Laboratory, Champaign, IL 61826-9005.

IEEE Log Number 9215226.

a computationally expensive procedure similar to simulated annealing. In addition, the MPM criteria does not consider the spatial placement of errors when distinguishing among the quality of segmentations.

Finally, parameter estimation of MRF's is difficult. When parameters are above the "critical temperature," there may be no consistent estimator as the image size grows to infinity [21]. Methods have been developed to estimate MRF parameters from images being segmented [22], but they are computationally expensive.

In this paper, we attempt to address these difficulties by introducing a new approach to Bayesian image segmentation. This method replaces the MRF model with a novel multiscale random field (MSRF) and replaces the MAP estimator with a sequential MAP (SMAP) estimator derived from a new estimation criteria. Together, the proposed estimator and model result in a segmentation algorithm that is not iterative and can be computed in time proportional to MN, where M is the number of classes and N is the number of pixels. We also develop a method for estimating the parameters of the MSRF model directly from the image during the segmentation process. This allows images with differing region sizes to be segmented accurately without specific prior knowledge of their behavior.

The MSRF model we propose is composed of a series of random fields progressing from coarse to fine scale. Each field is assumed to only depend on the previous coarser field. Therefore, the series of fields form a Markov chain in scale or resolution. Further, we assume that points in each field are conditionally independent given their coarser scale neighbors. This leads to a rich model with computationally tractable properties. In fact, Luettgen *et al.* have shown in independent work that models similar to the MSRF actually include MRF's as a subclass [23].

In earlier work, Chou, Willsky, Benveniste, Basseville, Golden, and Nikoukhah [24]–[28] have shown that Markov chains in scale can be used to model continuously valued Gaussian processes in one and two dimensions. This work has resulted in fast algorithms for problems such as optical flow estimation [29]. Estimation for these models is performed using a generalizations of Kalman filtering [26]–[28]. This approach is ideal for Gaussian models since the MAP, conditional mean, and minimum mean squared estimates coincide and may be computed using only recursively computed first and second-order statistics. However, since our model requires discrete values to represent pixel classes, these methods are not applicable.

The MSRF model has a number of advantages over fixed scale MRF's. The Markov chain structure facilitates straightforward methods for parameter estimation since it eliminates difficulties with intractable normalizing constants (partition functions) found in MRF's. Yet, the model does not impose an unnatural spatial ordering to the pixels since the Markov chain is in scale. In addition, since explicit parameters are available to control both coarse and fine scale behavior, the MSRF model can more accurately describe image behavior.

The SMAP estimation method results from minimizing the expected size of the largest misclassified region. This is accomplished by assigning progressively larger cost to errors at coarser scale. Intuitively, the criteria accounts for the fact that an error at coarse scale is more grievous since it causes the misclassification of many pixels. The SMAP criteria results in a series of optimization steps going from coarse to fine scale. At each scale, the best segmentation is computed given the previous coarser segmentation and the observed data. Each maximization step is computationally simple and noniterative if the region parameters are known. The complete procedure is reminiscent of pyramidal pixel linking procedures [30], [31] but requires local computations much like those used in Bayesian networks [32].

If the region parameters are unknown, they may be estimated using an iterative procedure at each scale. This iterative procedure, based on the expectation maximization (EM) algorithm [33], is implemented by subsampling the image. Therefore, parameter estimation only increases the required computation by approximately a factor of two.

Finally, we note the multispectral SMAP segmentation algorithm is available in the geographical resources analysis support system (GRASS) Version 4.1 [34]. GRASS is a public domain geographic information system (GIS).

Section II describes the general structure of our segmentation approach, whereas Section III develops the detailed segmentation formulas. Finally, Section IV applies the algorithm to both synthetic images and remotely sensed multispectral images with corresponding ground truth data.

II. MULTISCALE SEGMENTATION APPROACH

The random field Y is the image that must be segmented into regions of distinct statistical behavior. (We use upper case letters to denote random quantities, while lower case letters denote the corresponding deterministic realizations.) Individual pixels in Y are denoted by Y_s, where s is a member of a 2-D lattice of points S.

The basis of our segmentation approach is a hierarchical or doubly stochastic model as shown in Fig. 1. This model assumes that the behavior of each observed pixel is dependent on a corresponding unobserved label pixel in X. Each label specifies one of M possible classes, each with it own statistical behavior. The dependence of observed pixels on their labels is specified through $p_{y|x}(y|x)$ the conditional distribution of Y given X. Prior knowledge about the size and shapes of regions will be modeled by the prior distribution $p(x)$.

Since a variety of features can be used with this approach, it is a general framework for the segmentation problem. For the texture segmentation problem, a stochastic texture model can be used for $p_{y|x}(y|x)$ [4], [35], [15], or texture feature vectors can be extracted at each pixel [36]–[38] and modeled with a multivariate distribution. However, we will use segmentation of multispectral remotely sensed images as the target application for our examples. In this case, each pixel, Y_s, will be a vector of D spectral components.

In the following sections, we first describe the general structure of a MSRF model for $p(x)$, and we develop a sequential MAP estimation approach for computing the best

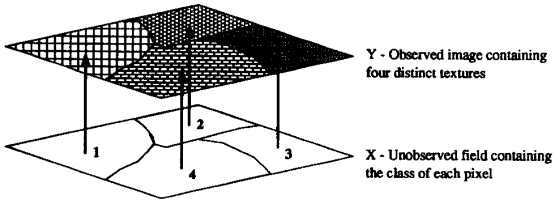


Fig. 1. Structure of a doubly stochastic random field used in segmentation. The behavior of the image (e.g., texture, gray scale, color, or multispectral values) given the class labels is defined by the conditional distribution $p_{y|x}(y|x)$. Prior information is contained in the distribution of the class labels $p(x)$.

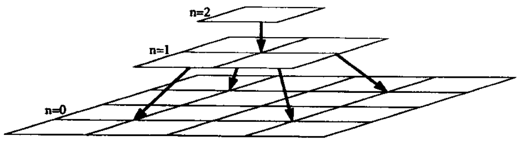


Fig. 2. Pyramid structure of the MSRF. The random field at each scale is causally dependent on the coarser scale field above it.

segmentation. The detailed models and recursion formulas resulting from this framework are then derived in Section III.

A. Multiscale Random Field Model

In this section, we develop a multiscale random field (MSRF) model composed of a series of random fields at varying scales or resolutions. Fig. 2 depicts the pyramid structure of the MSRF. At each scale, n, the segmentation or labeling is denoted by the random field $X^{(n)}$, and the set of lattice points is denoted by $S^{(n)}$. In particular, $X^{(0)}$ is assumed to be the finest scale random field with each point corresponding to a single image pixel. Each label at the next coarser scale $X^{(1)}$ then corresponds to a group of four points in the original image. Therefore, the number of points in $S^{(1)}$ is $1/4$ the number of points in $S^{(0)}$.

The fundamental assumption of the MSRF model is that the sequence of random fields from coarse to fine scale form a Markov chain. Therefore, the distribution of $X^{(n)}$ given all coarser scale fields is only dependent on $X^{(n+1)}$. This is a reasonable assumption since $X^{(n+1)}$ should contain all the relevant information from previous coarser scales. Formally, this Markov chain relation may be state as

$$\begin{aligned} P(X^{(n)} = x^{(n)} | X^{(l)} = x^{(l)}\ l > n) &= P(X^{(n)} = x^{(n)} | X^{(n+1)} = x^{(n+1)}) \\ &= p_{x^{(n)}|x^{(n+1)}}(x^{(n)}|x^{(n+1)}). \end{aligned} \tag{1}$$

Correspondingly, the exclusive dependence of Y on $X^{(0)}$ implies that

$$\begin{aligned} P(Y \in dy | X^{(n)}\ n > 0) &= P(Y \in dy | X^{(0)}) \\ &= p_{y|x^{(0)}}(y|x^{(0)}). \end{aligned} \tag{2}$$

The joint distribution of X and Y may then be expressed as the product of these distributions

$$\begin{aligned} &P(Y \in dy, X = x) \\ &\quad = p_{y|x^{(0)}}(y|x^{(0)}) \left\{ \prod_{n=0}^{L-1} p_{x^{(n)}|x^{(n+1)}}(x^{(n)}|x^{(n+1)}) \right\} \\ &\qquad p_{x^{(L)}}(x^{(L)}) \end{aligned}$$

where L is the coarsest scale in X. This Markov structure in scale has the isotropic behavior associated with MRF's, but in addition, the causal dependence in scale results in a noniterative segmentation algorithm and direct methods of parameter estimation.

B. Sequential MAP Estimation

In order to segment the image Y, we must accurately estimate the pixel labels in X. Bayesian estimation techniques are the natural approach since we have assumed the existence of a prior distribution $p(x)$. Generally, Bayesian estimators attempt to minimize the average cost of an erroneous segmentation. This is done by solving the optimization problem

$$\hat{x} = \arg\min_{x} E[C(X,x)|Y=y] \tag{3}$$

where $C(X,x)$ is the cost of estimating the true segmentation X by the approximate segmentation x. Notice that X is a random quantity, whereas x is a deterministic argument. Of course, the choice of the functional $C(\cdot,\cdot)$ is of critical importance since it determines the relative importance of errors.

In order to understand the deficiencies of the MAP estimate, we will first look at the assumptions of its derivation. The MAP estimate is the solution to (3) when the cost functional is given by

$$C_{\mathrm{MAP}}(X,x) = 1 - \delta(X - x)$$

where $\delta(X-x)$ is 1 when $X = x$ and 0 otherwise. Since $C_{\mathrm{MAP}}(X,x) = 1$ whenever any pixel is incorrectly labeled, the MAP estimate maximizes the probability that *all* pixels will be correctly labeled. Of course, a segmentation need not be completely accurate at all pixels to be useful. Even good segmentations will normally have erroneously classified pixels along region boundaries. This is particularly true in high-resolution images where the misclassification of a single pixel is not significant. Therefore, the MAP estimate can be excessively conservative [19]–[20].

The implications of the MAP criteria appear even more inappropriate for the estimation of the MSRF introduced in the previous sections. The cost function used for MAP estimation of a MSRF is

$$\begin{aligned} C_{\mathrm{MAP}}(X,x) &= 1 - \delta(X - x) \\ &= 1 - \prod_{n=0}^{L} \delta(X^{(n)} - x^{(n)}). \end{aligned}$$

This cost function is 1 if a labeling error occurs at any scale n of the segmentation. Consequently, this function assigns equal cost to a single mislabeled pixel at $n = 0$ or the mislabeling of approximately 256 pixels at $n = 4$. This cost assignment is clearly undesirable.

TABLE I

UPPER BOUND OF ERROR TERM $\mathcal{E}$ AS A FUNCTION OF SCALE FOR THREE 512×512 IMAGES. AT MODERATE AND FINE RESOLUTIONS, THE CORRECT SEGMENTATION IS UNCERTAIN AND $\mathcal{E}$ VERY SMALL.

Scale	Pixel Size	Bound on $\mathcal{E}$		
		image 1	image 2	image 3
0	1×1	0	0	0
1	2×2	$10^{-1114.22}$	$10^{-1642.87}$	$10^{-1073.88}$
2	4×4	$10^{-272.98}$	$10^{-501.67}$	$10^{-283.79}$
3	8×8	$10^{-61.82}$	$10^{-136.86}$	$10^{-75.61}$
4	16×16	$10^{-7.96}$	$10^{-19.13}$	$10^{-12.04}$
5	32×32	$10^{-1.16}$	$10^{-3.13}$	$10^{-2.53}$
6	64×64	$10^{-1.72}$	$10^{-0.55}$	$10^{-1.11}$
7	128×128	$10^{-0.01}$	$10^{-0.01}$	$10^{-0.01}$
8	256×256	$10^{-0.00}$	$10^{-0.00}$	$10^{-0.00}$

Ideally, a desirable cost function should assign progressively greater cost to segmentations with larger regions of misclassified pixels. To achieve this goal, we propose the following alternative cost function:

$$C_{\mathrm{SMAP}}(X, x) = \frac{1}{2} + \sum_{n=0}^{L} 2^{n-1} C_n(X, x)$$

where

$$C_n(X, x) = 1 - \prod_{i=n}^{L} \delta(X^{(i)} - x^{(i)}).$$

The behavior of C_{SMAP} is solely a function of the coarsest scale K that contains a misclassified pixel. More precisely, let K be the unique scale such that $X^{(K)} \neq x^{(K)}$, but $X^{(i)} = x^{(i)}$ for all $i > K$. Then, the functions C_n are given by

$$C_n(X, x) = \begin{cases} 1 & \text{if } n \leq K \\ 0 & \text{if } n > K \end{cases}$$

and the total cost is given by $C_{\mathrm{SMAP}}(X, x) = 2^K$. This error at scale K will generally lead to the misclassification of a group of pixels at the finest scale. The width of this misclassified group of pixels will be approximately $2^K = C_{\mathrm{SMAP}}(X, x)$. Therefore, the SMAP cost function has the following intuitive interpretation:

$$C_{\mathrm{SMAP}}(X, x) \approx \text{width of the largest grouping of misclassified pixels.}$$

We can determine the estimator which minimizes this proposed cost by evaluating (3).

$$\begin{aligned} \hat{x} &= \arg\min_x E[C_{\mathrm{SMAP}}(X, x) | Y = y] \\ &= \arg\min_x \sum_{n=0}^{L} 2^{n-1} \Big\{ 1 - P(X^{(i)} = x^{(i)} \; i \geq n | Y = y) \Big\} \\ &= \arg\max_x \sum_{n=0}^{L} 2^n P(X^{(i)} = x^{(i)} \; i \geq n | Y = y). \end{aligned}$$

Since the random fields $X^{(n)}$ form a Markov Chain, we will compute this estimate recursively in the scale parameter n. This is done by assuming that $\hat{x}^{(i)}$ has been computed for $i > n$ and using this result to compute $\hat{x}^{(n)}$. In Appendix A, we show that this recursive approach yields the following expression for the solution:

$$\hat{x}^{(n)} = \arg\max_{x^{(n)}} \Big\{ \log p_{x^{(n)}|x^{(n+1)},y}(x^{(n)} | \hat{x}^{(n+1)}, y) + \mathcal{E}(x^{(n)}) \Big\}$$

where $\mathcal{E}$ is a second order term which may be bounded by

$$0 \leq \mathcal{E}(x^{(n)}) \leq \max_{x^{(n-1)}} p_{x^{(n-1)}|x^{(n)},y}(x^{(n-1)} | \hat{x}^{(n)}, y) \ll 1.$$

Table I gives computed upper bounds for $\mathcal{E}$ as a function of scale. (Details of the computation are given in the Section IV). For our problem, the approximation that $\mathcal{E} \ll 1$ is very good. To see this, notice that $x^{(n-1)}$ is an interpolation of the coarser segmentation $\hat{x}^{(n)}$, given the image y. Normally, there will be many pixels in the interpolation $x^{(n-1)}$ for which the correct labeling is uncertain. This is particularly true around the boundaries of objects. Since the number of unique labeling combinations for these pixels is enormous, the probability of any particular combination will be small. In fact, for the models we will use, this probability goes to 0 as the number of pixels N increases. Therefore

$$\lim_{N \to \infty} \mathcal{E}\Big(x^{(n)}\Big) = 0.$$

At very coarse scales, the number of labels becomes small, and often, only one reasonable interpolation will exist (i.e., $\mathcal{E} \approx 1$). However, in this case, the correct labeling of pixels at the coarser scale n must also be unambiguous, and any reasonable estimator should have good performance.

Ignoring the contribution of $\mathcal{E}$ results in the following recursive equations:

$$\begin{aligned} \hat{x}^{(L)} &= \arg\max_{x^{(L)}} \log p_{x^{(L)}|y}(x^{(L)} | y) \\ \hat{x}^{(n)} &= \arg\max_{x^{(n)}} \log p_{x^{(n)}|x^{(n+1)},y}\Big(x^{(n)} | \hat{x}^{(n+1)}, y\Big). \end{aligned}$$

The recursion is initialized by determining the MAP estimate of the coarsest scale field given the observed data. The segmentation at each finer scale is then found by computing the MAP estimate of $X^{(n)}$, given $\hat{x}^{(n+1)}$ and the image y. Due to this structure, we refer to this estimator as a sequential MAP (SMAP) estimator.

By using Bayes rule, the Markov properties of X, and assuming that $X^{(L)}$ is uniformly distributed, the SMAP recursion may be rewritten in a form which is more easily computed.

$$\hat{x}^{(L)} = \arg\max_{x^{(n)}} \log p_{y|x^{(L)}}(y | x^{(L)}) \tag{4}$$

$$\begin{aligned} \hat{x}^{(n)} = \arg\max_{x^{(n)}} \Big\{ &\log p_{y|x^{(n)}}(y | x^{(n)}) \\ &+ \log p_{x^{(n)}|x^{(n+1)}}(x^{(n)} | \hat{x}^{(n+1)}) \Big\}. \end{aligned} \tag{5}$$

The two terms in (5) play roles analogous to those of the likelihood function and prior distribution in conventional Bayesian estimation. The first term of the maximization gives the likelihood of the observed data y, given the labeling at scale n. The second term of the maximization embodies the *a priori* information about the behavior of X. Therefore, this term biases the solution to favor segmentations with large regions and smooth boundaries.

The SMAP estimator has a number of additional advantages. In the next section, we will introduce specific models so that each optimization step of (5) may be computed with a single noniterative pass. This is in contrast to the MAP and MPM estimators, which require computationally expensive iterative optimization methods [12], [3], [19]. Further, the SMAP estimator has a subtle advantage over MPM. The MPM method chooses each pixel individually to minimize the probability of error. However, it does not consider the spatial placement of errors. Since the SMAP method attempts to minimize the spatial size of errors, it will tend to generate a subjectively more desirable segmentation.

III. SEGMENTATION ALGORITHM

In this section, we define the specific models we will use for Y and X. This will be done by specifying the conditional density $p_{y|x^{(0)}}(y|x^{(0)})$, together with the coarse to fine scale transition densities $p_{x^{(n)}|x^{(n+1)}}(x^{(n)}|x^{(n+1)})$. We then develop an adaptive segmentation algorithm that estimates the parameters of the MSRF during the segmentation process.

For the multispectral segmentation problem, we restrict ourselves to models with observed pixels that are conditionally independent given their class labels. This implies that the spatial texture of regions will not be used as a discriminating feature. Instead, we shall rely on the multispectral characteristics of classes to discriminate distinct regions. This approach is supported by the fact that spatial correlation has been found to be weak within regions of multispectral images corresponding to a single ground cover [39], [40]. Using this assumption, the conditional density function for the image has the form

$$p_{y|x^{(0)}}(y|x^{(0)}) = \prod_{s\in S^{(0)}} p_{y_s|x_s^{(0)}}(y_s|x_s^{(0)})$$

where $p_{y_s|x_s^{(0)}}(\cdot|k)$ is the conditional density function for an individual pixel given the class label k. Since each pixel is composed of D multispectral components, $p_{y_s|x_s^{(0)}}(\cdot|k)$ is a multivariate density function. We will use a multivariate Gaussian mixture density in numerical experiments, but other distributions can just as well be applied.

More generally, the methods used throughout this paper are applicable to any model that can be expressed in the form

$$\log p_{y|x^{(0)}}(y|x^{(0)}) = \sum_{s\in S^{(0)}} l_s(y|x_s^{(0)}) + c(y).$$

where the functions l_s depend on all of y, and c is an arbitrary function of y. This type of model has been used extensively in texture segmentation applications [4], [41], [35], [15].

We will restrict our choice of models for X to have two important properties. First, the pixels in $X^{(n)}$ will be conditionally independent, given the pixels in $X^{(n+1)}$. Second, each pixel $X_s^{(n)}$ will only be dependent on a local neighborhood of pixels at the next coarser scale. This set of neighboring locations to s will be denoted by ∂s. Based on these properties, the transition distribution from coarse to fine scale must have the form

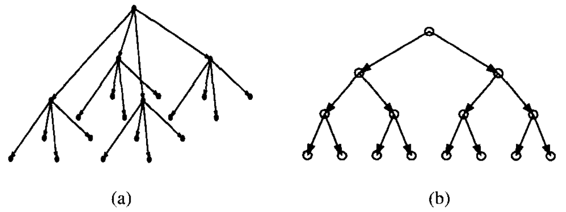

(a) (b)

Fig. 3. (a) Quadtree structure used for MSRF model; (b) 1-D analog to quadtree structure.

$$p_{x^{(n)}|x^{(n+1)}}(x^{(n)}|x^{(n+1)}) = \prod_{s\in S^{(n)}} p_{x_s^{(n)}|x_{\partial s}^{(n+1)}}(x_s^{(n)}|x_{\partial s}^{(n+1)})$$

where $p_{x_s^{(n)}|x_{\partial s}^{(n+1)}}$ is the probability density for $x_s^{(n)}$ given its neighbors at the coarser scale $x_{\partial s}^{(n+1)}$.

These two assumptions assure that the pixels in $x^{(n)}$ will be conditionally independent, given the pixels in $x^{(n+1)}$. However, we note that nearby pixels in $x^{(n)}$ may still be highly dependent since they will share coarser scale neighbors.

The choice of neighborhood ∂s for the pixel s is important since it will define the structure of our multiscale pyramid. We will employ two types of neighborhoods. The first corresponds to a quadtree structure and allows simple and exact calculation of the SMAP segmentation. The second neighborhood corresponds to a more general graph structure and can account for more complex interactions that occur across blocks of the quadtree. Therefore, this model produces smoother, less blocky segmentations. Unfortunately, the graph structured pyramid does not allow exact calculation of the SMAP estimator. Therefore, a hybrid model, which incorporates both the quadtree and graph structure is used to approximate the exact SMAP estimate.

A. Quadtree Model

The first pyramid structure that we consider is based on a conventional quadtree. The structure of the quadtree is shown in Fig. 3(a), and the 1-D analog is shown in Fig. 3(b). Since this is a tree structure, each point in the pyramid is only dependent on a single point at the coarser scale. This coarser scale neighbor of a point s is called the parent of s and will be denoted by the function $d(s)$. In a similar vein, the nth successive parent of a point will be denoted $d^n(s)$, and $d^{-n}(s)$ will be the set of all points which occur n levels down from s in the tree structure.

We choose the following transition function to model the probability that $X_s^{(n)}$ has class m, given that its parent is of class k.

$$p_{x_s^{(n)}|x_{\partial s}^{(n+1)}}(m|k) = \theta_{n,0}\delta_{m,k} + \frac{1-\theta_{n,0}}{M} \tag{6}$$

where $\delta_{m,k}$ is the unit sample function, and M is the number of possible classes. The parameter $\theta_{n,0} \in [0,1]$ is the probability that the labeling will remain the same from scale $n+1$ to n. If a class change does occur, it is equally likely to be any one of the remaining class types. At fine resolutions, the neighboring pixels are more likely to have the same class. Therefore, $\theta_{n,0}$ will generally be an increasing function of resolution (decreasing function of n). Notice that this distribution only depends on the the scale n through $\theta_{n,0}$ but does not depend on the particular pixel s.

An important property of the quadtree structure is that the conditional distribution of Y, given $X^{(n)}$, has a product form that may be computed using a simple fine-to-coarse recursion. Let $y_s^{(n)}$ be the set of leaves of the quadtree that are on the branch starting at $s \in S^{(n)}$. Specifically, $y_s^{(n)} = \{y_r : d^n(r) = s\}$. In Appendix B, we show that the conditional density for Y given $X^{(n)}$ has the product form

$$p_{y|x^{(n)}}(y|x^{(n)}) = \prod_{s \in S^{(n)}} p_{y_s^{(n)}|x_s^{(n)}}(y_s^{(n)}|x_s^{(n)}). \tag{7}$$

Furthermore, the density functions $p_{y_s^{(n)}|x_s^{(n)}}$ may be computed using the following recursion, where M is the number of class labels.

$$\begin{aligned} &p_{y_s^{(n+1)}|x_s^{(n+1)}}(y_s^{(n+1)}|k) \\ &= \prod_{r \in d^{-1}(s)} \sum_{m=1}^{M} p_{y_r^{(n)}|x_r^{(n)}}(y_r^{(n)}|m)\; p_{x_r^{(n)}|x_{\partial r}^{(n+1)}}(m|k). \end{aligned} \tag{8}$$

In practice, dynamic range considerations mandate that the logarithm of these functions be computed and stored. Therefore, we define the log likelihood function

$$l_s^{(n)}(k) \triangleq \log p_{y_s^{(n)}|x_s^{(n)}}(y_s^{(n)}|k)$$

where the dependence on y is suppressed for clarity. Substituting the transition distribution of (6) into (8) and converting to log likelihood functions yields the new recursion

$$\begin{aligned} l_s^{(0)}(k) &= \log p_{y_s|x_s^{(0)}}(y_s|k) \\ l_s^{(n+1)}(k) &= \sum_{r \in d^{-1}(s)} \log \left\{ \theta_{n,0} \exp\{l_r^{(n)}(k)\} \right. \\ &\qquad \left. + \frac{1-\theta_{n,0}}{M} \sum_{m=1}^{M} \exp\{l_r^{(n)}(m)\} \right\}. \end{aligned} \tag{9}$$

To perform the SMAP segmentation, we must compute these log likelihood functions at every point in the quadtree. The number of points in the pyramid is approximately given by $\frac{4N}{3}$, where N is the number of pixels at the finest scale. Since each of the log likelihood functions may be stored in the form of M numbers, the total required storage is approximately $(4MN)/3$. Because the second sum of (9) is not a function of k, each log likelihood function can be computed in time proportional to M. Therefore, the total computation time for evaluating the log likelihood functions is $\mathcal{O}(MN)$.

We note that in the limiting case of $\theta_{n,0}(k) = 1$, the recursion reduces to simple averaging.

$$l_s^{(n+1)}(k) = \sum_{r \in d^{-1}(s)} l_r^{(n)}(k)$$

This is equivalent to assuming that all the pixels in the block of pixels $y_s^{(n+1)}$ must have the same label. Of course, this is often not the case since some of pixel blocks are likely to fall on region boundaries. These blocks then have a mixture of pixels from the two regions. Therefore, simple averaging can cause pixels on region boundaries to be misclassified. This is particularly true if the statistical average of the two regions appears to have characteristics of a third class. Another advantage of the more accurate recursion (9) is that a small group of anomalous pixels will not adversely effect the classification of a large region. Simple linear averaging of the log likelihood function tends to be easily biased by a few pixels that are statistical outliers, whereas the more accurate recursion is robust to such errors.

Once the likelihood functions are computed, the SMAP segmentation may be efficiently computed using (5).

$$\begin{aligned} \hat{x}^{(n)} = \arg\max_{x^{(n)}} \sum_{s \in S^{(n)}} &\left\{ l_s^{(n)}(x_s^{(n)}) \right. \\ &\left. + \log p_{x_s^{(n)}|\hat{x}_{\partial s}^{(n+1)}}(x_s^{(n)}|\hat{x}_{\partial s}^{(n+1)}) \right\}. \end{aligned}$$

This expression is easily evaluated by minimizing with respect to each pixel individually

$$\hat{x}_s^{(n)} = \arg\max_{1 \le k \le M} \left\{ l_s^{(n)}(k) + \log p_{x_s^{(n)}|\hat{x}_{\partial s}^{(n+1)}}(k|\hat{x}_{\partial s}^{(n+1)}) \right\}. \tag{10}$$

This coarse-to-fine segmentation operation requires order $\mathcal{O}(MN)$ computation time. Therefore, the complete segmentation process consists of a fine-to-coarse and a coarse-to-fine operation, each of which require $\mathcal{O}(MN)$ computation.

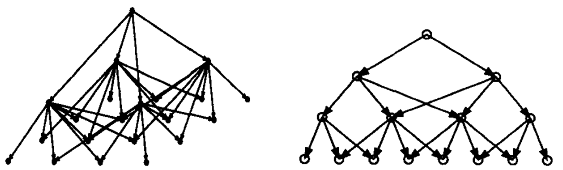
Although the quadtree model results in an exact expression for computing the SMAP segmentation, it does not completely capture some aspects of image behavior. In the following section, we introduce an augmented model which improves on the quadtree.

B. Pyramidal Graph Model

An important disadvantage of the quadtree model is that pixels that are spatially adjacent may not have common neighbors at the next coarser scale. Therefore, the model does not enforce continuity of region boundaries when they pass across branches of the quadtree. For example, if the image is broken into four quadrants at the second level of the quadtree, then a region boundary will not be constrained to be smooth as it passes from one of these quadrants to the next.

This weakness may be corrected by increasing the number of coarse scale neighbors for each pixel. Such a pyramidal graph structure is shown in Fig. 4(a), and an analogous one dimensional graph is shown in Fig. 4(b). Notice that each point has three neighbors at the next coarser scale.

In order to express the positions of these neighbors, we explicitly denote a pixel at scale n as $s = (i, j)$, where i and j are the row and column indices starting at 0 and ranging through the width minus one and the height minus one, respectively. The three neighbors at scale $n+1$ may then be computed using the function $\text{odd}(i) = 1$ if i is odd and -1



(a) (b)

Fig. 4. (a) Augmented pyramidal graph structure used for MSRF model; (b) 1-D analog of pyramidal graph structure.

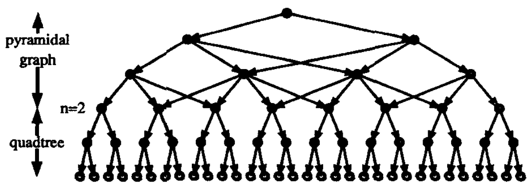


Fig. 5. 1-D analog to hybrid graph structure. Scales $n > 2$ use a pyramidal graph structure, and scales $n \leq 2$ use a quadtree structure.

if i is even and the greatest smaller integer function $\lfloor \cdot \rfloor$.

$$
\begin{aligned}
s_1 &= (\lfloor i/2 \rfloor, \lfloor j/2 \rfloor)\\
s_2 &= (\lfloor i/2 \rfloor, \lfloor j/2 \rfloor) + (\mathrm{odd}(i), 0)\\
s_3 &= (\lfloor i/2 \rfloor, \lfloor j/2 \rfloor) + (0, \mathrm{odd}(j)).
\end{aligned}
$$

The transition function that we choose for this pyramid graph is a natural extension of the transition function used for the quadtree-based model.

$$
\begin{aligned}
&\tilde{p}_{x_s^{(n)}|x_{\partial s}^{(n+1)}}(m|i,j,k)\\
&= P(X_s^{(n)} = m | X_{s_1}^{(n+1)} = i, X_{s_2}^{(n+1)} = j, X_{s_3}^{(n+1)} = k)\\
&= \frac{\theta_{n,1}}{7}(3\delta_{m,i} + 2\delta_{m,j} + 2\delta_{m,k}) + \frac{1-\theta_{n,1}}{M}. \qquad (11)
\end{aligned}
$$

Notice that we use the notation $\tilde{p}$ to distinguish from the transition distribution used for the quadtree model. As with the quadtree model, the parameter $\theta_{n,1} \in [0,1]$ determines the probability that the label of the fine scale point will be the same as one of the coarser scale points. Conversely, $1 - \theta_{n,1}$ is the probability that a new label will be randomly chosen from the available labels.

The disadvantage of the pyramid graph structure is that the likelihood function for the labels does not have a product form as was the case for the quadtree in (7). Therefore, there is no simple fine-to-coarse recursion with the form of (8) for computing the likelihood of the image y, given the labels $x^{(n)}$.

For computations at a single scale n, this problem may be circumvented by assuming that the pyramid has a quadtree structure for scales finer than n and a graph structure for coarser scales. A 1-D analog to this hybrid pyramid structure is shown in Fig. 5 for $n = 2$. Notice that for levels above n, the pyramid is a graph, but below n, the pyramid has a simple tree structure. Using this hybrid pyramid, the conditional likelihood of (5) then has the computable form

$$
\begin{aligned}
\log p_{y,x^{(n)}|x^{(n+1)}}(y, x^{(n)}|\hat{x}^{(n+1)}) = &\sum_{s\in S^{(n)}} l_s^{(n)}(x_s^{(n)})\\
&+ \log \tilde{p}_{x_s^{(n)}|x_{\partial s}^{(n+1)}}(x_s^{(n)}|\hat{x}_{\partial s}^{(n+1)})
\end{aligned}
$$

which results in the following formula for the SMAP estimate of $X^{(n)}$:

$$
\hat{x}_s^{(L)} = \arg\max_{1\leq k\leq M} l_s^{(L)}(k) \qquad (12)
$$

$$
\hat{x}_s^{(n)} = \arg\max_{1\leq k\leq M} \left\{ l_s^{(n)}(k) + \log \tilde{p}_{x_s^{(n)}|x_{\partial s}^{(n+1)}}(k|\hat{x}_{\partial s}^{(n+1)}) \right\}. \qquad (13)
$$

where $\tilde{p}$ is the transition function of (11), and $l_s^{(n)}(k)$ are computed using the recursion of (9).

Of course, the application of the above formula at all scales is an approximation to the true SMAP segmentation since we may not legitimately change our model during the segmentation process. However, we believe that this approximation is reasonable since the likelihood functions are primarily dependent on the image data and have only a secondary dependence on the pyramid structure. Intuitively, if the pixels in $y_s^{(n)}$ appear to be principally from class k, then the likelihood function $l_s^{(n)}(k)$ should be relatively large, regardless of the pyramid structure used.

All the following analysis and experimentation will assume the use of this hybrid quadtree-graph structure. As in the case of the quadtree, segmentation using the hybrid structure is performed in two steps, each of which only requires order MN computation. The first step is a fine-to-coarse computation given by (9). The second step is a coarse-to-fine computation given by (13). Together, the total computation is of order $\mathcal{O}(MN)$.

C. Parameter Estimation

In typical applications, one does not have prior information about the exact behavior of the segmentation X. However, it is possible to determine this information directly from the image as it is being segmented. To do this, we must estimate the parameters $\theta_n = [\theta_{n,0}, \theta_{n,1}]$ during the segmentation process.

The parameters $\theta_{n,0}$ are required for fine-to-coarse operations used in computing the log likelihood functions, and the parameters $\theta_{n,1}$ are required for coarse-to-fine operations used in the SMAP segmentation. However, we will estimate both of these parameter vectors during the coarse-to-fine operations. This means that the segmentation process will require two full passes composed of the following steps:

1. Perform fine-to-coarse operations using initial parameter values $\theta_{n,0} = 1$.
2. Estimate parameters $\theta_{n,0}$ and $\theta_{n,1}$ during coarse-to-fine operations.
3. Perform fine-to-coarse operations using estimated parameters.
4. Reestimate parameters $\theta_{n,1}$ during final coarse-to-fine segmentation.

The estimation procedures of 2 and 4 is performed sequentially at each scale. Each transition parameter vector θ_n is estimated concurrently with the computation of the segmentation $\hat{x}^{(n)}$. The computational overhead of this estimation procedure is greatly reduced by subsampling the image at high resolutions. Since the number of pixels at high resolutions is great, this subsampling does not substantially impact on the accuracy of the estimated parameters.

The two-pass process implies that parameter estimation will increase computation by at least a factor of 2. However, the additional computation required within each iteration is minimal due to subsampling. Therefore, the total increase in computation for parameter estimation is generally close to 2.

We begin by deriving the sequential parameter estimation procedure. The transition parameter $\theta_{n,1}$ is estimated by finding the maximum likelihood value given the image y and the previous coarse scale segmentation $\hat{x}^{(n+1)}$. Formally, we compute the solution to the optimization criteria

$$\hat{\theta}_{n,1} \in \arg \max_{\theta_{n,1} \in \Omega} p_{y|x^{(n+1)}}(y|\hat{x}^{(n+1)}, \theta_{n,1})$$

where Ω is the set of valid parameter values. Using the hybrid pyramid structure of Section III-B, the log likelihood function L has the specific form

$$\begin{aligned} L(\theta_{n,1}) &= \log p_{y|x^{(n+1)}}(y|\hat{x}^{(n+1)}, \theta_{n,1}) \\ &= \sum_{s \in S^{(n)}} \log \Bigg(\sum_{k=1}^{M} \exp(l_s^{(n)}(k)) \\ &\qquad \cdot \tilde{p}_{x_0^{(n)}|x_{\partial 0}^{(n+1)}}(k|\hat{x}_{\partial s}^{(n+1)}, \theta_{n,1}) \Bigg) \end{aligned}$$

where the dependence of $\tilde{p}$ on n is through the parameter $\theta_{n,1}$. Notice that $\tilde{p}$ uses the subscript $x_0^{(n)}|x_{\partial 0}^{(n+1)}$ to emphasize that the conditional distribution assumed in (11) does not depend on s.

This likelihood $L(\theta_{n,1})$ may be maximized as a function of $\theta_{n,1}$ by using the expectation-maximization (EM) algorithm [33], [42]. From the form of the function $\tilde{p}$, we know that $L(\theta_{n,1})$ is a convex function of $\theta_{n,1}$. We will use this fact to show that the EM algorithm is guaranteed to converge to a value of $\theta_{n,1}$, which maximizes L.

To apply the EM algorithm, we must first compute the function

$$\begin{aligned} Q(\theta_{n,1}, \theta'_{n,1}) = E\Big[\log \Big(& p_{y|x^{(n)}}(y|X^{(n)}) \\ & \cdot p_{x^{(n)}|x^{(n+1)}}(X^{(n)}|\hat{x}^{(n+1)}, \theta_{n,1})\Big) \\ = & \; |Y = y, X^{(n+1)} \hat{x}^{(n+1)}, \theta'_{n,1}\Big] \end{aligned}$$

where we remind the reader that $X^{(n)}$ is a random object. The EM algorithm is then the following iterative procedure for finding a sequence of parameters that converge to $\hat{\theta}_{n,1}$:

$$\theta_{n,1}^{p+1} \in \arg \max_{\theta_{n,1} \in \Omega} Q(\theta_{n,1}, \theta_{n,1}^{p}).$$

To further simplify this expression, we must formulate a sufficient statistic for the transition distribution $\tilde{p}_{x_0^{(n)}|x_{\partial 0}^{(n+1)}}(m|i, j, k, \theta_{n,1})$ of (11). This statistic counts the number of distinct transitions that occur from a particular set of coarse scale neighbors $x_{\partial s}^{(n+1)}$ to a point $x_s^{(n)}$. Specifically, we may define the sufficient statistic T so that

$$\begin{aligned} &\log \tilde{p}_{x_0^{(n)}|x_{\partial 0}^{(n+1)}}(m|i, j, k, \theta_{n,1}) \\ &= \sum_{l=0}^{1} \sum_{h=0}^{2} T_{l,h}(m|i, j, k) \; V_{l,h}(\theta_{n,1}) \end{aligned} \tag{14}$$

where T and V have the functional forms

$$T_{l,h}(m|i, j, k) = \begin{cases} 1 & \text{if } \delta_{m,i} = l \text{ and } \delta_{m,j} + \delta_{m,k} = h \\ 0 & \text{otherwise} \end{cases}$$

and

$$V_{l,h}(\theta_{n,1}) = \log \left(\frac{\theta_{n,1}}{7}(3l + 2h) + \frac{1 - \theta_{n,1}}{M} \right).$$

By substituting (14) into the expression for for Q, we obtain the simpler expression

$$\theta_{n,1}^{p+1} \in \arg \max_{\theta_{n,1} \in \Omega} \left\{ \sum_{l=0}^{1} \sum_{h=0}^{2} \bar{T}_{l,h}(\theta_{n,1}^{p}) \; V_{l,h}(\theta_{n,1}) \right\} \tag{15}$$

where $\overline{T}_{l,h}$ is defined in (16) at the bottom of this page.

Evaluation of $\bar{T}_{l,h}$ is computationally expensive since it requires a summation over all the points in $S^{(n)}$. However, accurate estimation of θ_n only requires a representative sampling of the image data. This is particularly true at high resolutions where the number of pixels far exceeds the number required to accurately estimate these parameters. Therefore, we subsample the points in $S^{(n)}$ at period $P^{(n)}$ in both the horizontal and vertical directions, and we choose $P^{(n)} \propto \sqrt{2}^{-n}$ so that we will still have an increasing number of samples at finer scales. Experimental results given in the following section will show that subsampling substantially reduces computation without adversely effecting the performance of parameter estimation.

The two steps of each iteration in the EM algorithm are then as follows:

E—Compute $\bar{T}$ using the parameter $\theta_{n,1}^{p}$, sampling period $P^{(n)}$, and (16).

M—Compute $\theta_{n,1}^{p+1} \in \arg\max_{\theta_{n,1} \in \Omega} Q(\theta_{n,1}, \theta_{n,1}^{p})$ using (15).

$$\begin{aligned} &\bar{T}_{l,h}(\theta_{n,1}) \\ &= \sum_{s \in S^{(n)}} E\Big[T_{l,h}(X_s^{(n)}|\hat{x}_{\partial s}^{(n+1)})|Y = y, X^{(n+1)} = \hat{x}^{(n+1)}, \theta_{n,1}\Big] \\ &= \sum_{s \in S^{(n)}} \frac{\sum_{k=1}^{M} T_{l,h}(k|\hat{x}_{\partial s}^{(n+1)}) \; \exp(l_s^{(n)}(k)) \; \tilde{p}_{x_0^{(n)}|x_{\partial 0}^{(n+1)}}(k|\hat{x}_{\partial s}^{(n+1)}, \theta_{n,1})}{\sum_{k=1}^{M} \exp(l_s^{(n)}(k)) \; \tilde{p}_{x_0^{(n)}|x_{\partial 0}^{(n+1)}}(k|\hat{x}_{\partial s}^{(n+1)}, \theta_{n,1})}. \end{aligned} \tag{16}$$

The M step can be efficiently computed since it requires the maximization of a convex function Q over an interval. This may be done, for example, with the golden section search method [43]. The question remains as to whether the algorithm converges to the global maximum of L. To answer this question, we adapt the convergence results of Wu [44] and Redner and Walker [45] to prove the following theorem in Appendix C.

Theorem 1: If i) Ω is a compact, convex set, ii) $L(\theta)$ and $Q(\theta, \theta')$ are continuous and differentiable on an open set containing Ω, and iii) $L(\theta)$ is convex, then any limit point $\hat{\theta}$ of the sequence $\{\theta^p\}_1^\infty$ has the property that

$$\hat{\theta} \in \arg\max_{\theta \in \Omega} L(\theta).$$

Notice that assumption ii) does not strictly hold since Q becomes infinite at the points 0 and 1. In practice, this problem can be resolved by estimating $\theta_{n,1}$ over some smaller interval $\Omega = [0 + \epsilon_1, 1 - \epsilon_1]$. In any case, the introduction of ϵ_1 is required for numerical stability.

Finally, the parameters $\theta_{n,0}$ are estimated using the statistics $\bar{T}$ resulting from convergence of the EM algorithm:

$$\hat{\theta}_{n,0} = \frac{\sum_{h=0}^{2} \bar{T}_{1,h}}{\sum_{l=0}^{1} \sum_{h=0}^{2} \bar{T}_{l,h}}. \tag{17}$$

These values are then used in the second pass of fine-to-coarse computations.

The complete SMAP segmentation algorithm with parameter estimation is summarized in the following:

1. Set the initial parameter values for all n, $\hat{\theta}_{n,0} = 1$, and $\hat{\theta}_{L-1,1} = 0.5$.
2. Compute the likelihood functions using (9) and the parameters $\hat{\theta}_{n,0}$.
3. Compute $\hat{x}^{(L)}$ using (12).
4. For scales $n = L - 1$ to $n = 0$
   (a) Use the EM algorithm to iteratively compute $\hat{\theta}_{n,1}$ and $\bar{T}$. Subsample by $P^{(n)}$ when computing $\bar{T}$, and stop when $|\theta_{n,1}^{p+1} - \theta_{n,1}^{p}| < \epsilon_2$.
   (b) Compute $\hat{\theta}_{n,0}$ using (17).
   (c) Compute $\hat{x}^{(n)}$ using (13).
   (d) Set $\hat{\theta}_{n-1,1} = \hat{\theta}_{n,1}(1 - 10\epsilon_2)$.
5. Repeat steps 2 through 4.

Since global convergence of the EM algorithm is guaranteed, the choice of ϵ_1 and ϵ_2 only impacts on the accuracy of convergence. We have used the values $\epsilon_1 = 10^{-6}$ and $\epsilon_2 = 10^{-4}$ in all our experimentation and have never found them to be problematic. Generally, smaller values of ϵ_2 will lead to better estimates but slower convergence, and we have found that $\epsilon_1 \ll \epsilon_2$. We also note that Step 4d is used to accelerate convergence of the EM algorithm by starting the new parameter values near the previous parameter values.

IV. EXPERIMENTAL RESULTS

In this section, we compare the performance of the SMAP algorithm with MAP segmentation using a MRF prior model for the pixel classes. This is done using a variety of synthetic images.

TABLE II
MEAN AND STANDARD DEVIATION OF EACH TEXTURE FOR THREE DIFFERENT IMAGES

		Class designation					
		0	1	2	3	4	5
image 1	μ	127.0	145.0	101.6	163.0	76.1	199.0
	σ	32	32	32	32	32	32
image 2	μ	127.0	137.1	112.7	147.2	98.4	167.5
	σ	32	32	32	32	32	32
image 3	μ	127	127	127	127	127	127
	σ	8.00	10.55	13.93	18.37	24.25	32

All results of the SMAP algorithm used unsupervised estimation of the MSRF parameters. Unless otherwise stated, subsampling was used in all experiments, and the subsampling period was always chosen using the formula

$$P(n) = \max\{\lfloor 2^{(L-n-3)/2} \rfloor, 1\}.$$

This generally resulted in sufficient sampling to accurately estimate parameters while keeping the computational overhead of parameter estimation to a minimum.

For our comparison with MAP estimation, we choose a conventional 8-pt neighborhood MRF model [15] for the class labels X. Specifically, the probability distribution has the form

$$p_x(x) = \frac{1}{z} \exp\left\{-\lambda\left(\sqrt{2} - 1\right)\left(t_1(x) + t_2(x)/\sqrt{2}\right)\right\}$$

where $\lambda = 1.5$, t_1 is the number of horizontal and vertical neighbors of different class, and t_2 is the number of diagonal neighbors of different class. This appeared to yield the best overall results.

Since the MAP estimate can not be exactly computed, an optimization method must also be chosen. We used simulated annealing (SA) [12], ICM [3], and multiple resolution segmentation (MRS) [15]. The ICM and SA methods were started with the maximum likelihood estimate of the segmentation. SA used a temperature schedule of the form

$$\frac{1}{T_{n+1}} = \frac{1}{T_n} + \Delta$$

where $T_0 = 1$, $T_{\text{final}} = .2666$, and Δ was chosen to achieve the desired number of iterations. We used both 100 and 500 iterations to compare the performance. After the desired number of iterations, ICM was used (equivalently $T = 0$) to assure convergence to a local minima. These annealing parameters were chosen since they seemed to give the best performance over the range of test images used. In practice, the choice of annealing parameters must be a compromise since the optimal parameters depend on the specific image being process.

The MRS algorithm differs from ICM and SA since it effectly uses different values of λ at each scale. The scale dependent λ is used because the MAP estimate would otherwise have unreasonable behavior at coarse scales [15]. Intuitively, the MRS algorithm attempts to approximately correct the undesirable properties of the MAP estimator by varying the prior model.

TABLE III
PERCENTAGE OF EACH CLASS LABEL THAT WAS CORRECTLY CLASSIFIED AND CLASS AVERAGES OF CLASSIFICATION ACCURACY

		Class label						Class Average
		0	1	2	3	4	5	
image 1	SMAP (subsampling)	100%	95%	96%	94%	93%	78%	92.6%
	SMAP (no subsampling)	100%	95%	96%	94%	93%	78%	92.6%
	SA 500	100%	97%	97%	96%	95%	96%	96.8%
	SA 100	99%	93%	96%	95%	95%	96%	95.7%
	ICM	60%	31%	62%	74%	94%	96%	69.5%
	MRS	100%	96%	96%	95%	92%	95%	95.7%
image 2	SMAP (subsampling)	99%	94%	93%	86%	53%	70%	82.5%
	SMAP (no subsampling)	99%	94%	93%	87%	60%	67%	83.3%
	SA 500	100%	81%	44%	59%	69%	74%	71.2%
	SA 100	74%	57%	10%	32%	74%	77%	54.0%
	ICM	23%	12%	6%	14%	97%	96%	41.3%
	MRS	99%	93%	92%	74%	63%	72%	82.2%
image 3	SMAP (subsampling)	100%	95%	95%	92%	61%	58%	83.5%
	SMAP (no subsampling)	100%	94%	95%	91%	60%	60%	83.3%
	SA 500	100%	0%	94%	92%	18%	75%	63.1%
	SA 100	100%	0%	40%	49%	52%	62%	50.5%
	ICM	100%	0%	1%	14%	14%	62%	31.8%
	MRS	100%	95%	96%	80%	52%	57%	80.0%

Three 512 × 512 synthetic images shown in Fig. 6 were used to test the SMAP and MAP algorithms. Each contained six white Gaussian textures with distinct means and variances as shown in Table II. In each case, the background is class 0, and the circles are numbered in order starting from the top. The radius of each subsequent row of circles was chosen to decrease by a factor of 2.

Figs. 7–9 show the result of segmenting the three test images using the SMAP and MAP algorithms. The percentage of misclassified pixels is tabulated for each class in Table III, and the computational requirements are shown in Table IV. For comparison, each table also lists the results of SMAP segmentation without pixel subsampling and the results of the MRS algorithm. For these test images, the performance of SMAP with and without subsampling was not significantly different. However, subsampling reduced computation by approximately a factor of 3. Table V gives the specific values of the estimated parameters at each scale. It also supports the conclusion that subsampling does not significantly degrade the accuracy of the estimates.

Inspection of the segmented test images indicates that the SMAP segmentation performed comparably to or substantially better than 500 iterations of SA. The SMAP algorithm appears to make fewer large misclassification errors. This may be due to the improved objective of minimizing the largest misclassification. We also note that the less smooth boundaries of the SMAP segmentation sometimes resulted in slightly higher misclassification rates for the smooth shapes in these synthetic images. However, the less smooth boundaries may actually be more desirable for segmenting the natural shapes of real images. Generally, the SMAP segmentation required less computation than even the worst performing algorithm ICM and much less computation than SA.

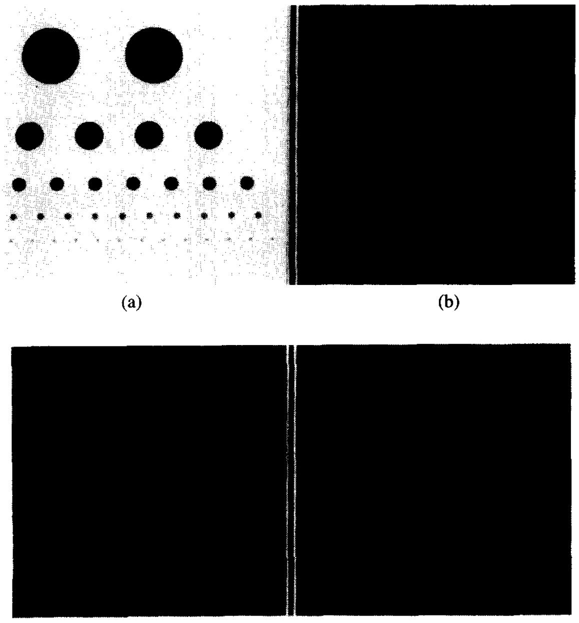

(a) (b)
(c) (d)

Fig. 6. (a) Image depicting the classes in synthetic test images. Each of the six classes is indicated by a distinct gray level. Three synthetic test images are (b) image 1, (c) image 2, and (d) image 3. Each region is distinguished by its mean and variance.

Inspection of Fig. 7 indicates that for the parameter value of $\lambda = 1.5$, the MAP segmentations tend to produce smoother boundaries than the SMAP algorithm. Since the circular regions of this synthetic image have very smooth boundaries, this results in slightly better classification accuracy for the 500 iteration MAP segmentation. However, the 100 iteration MAP segmentation has still not completely converged. Notice that along boundaries, groupings of pixels form that are misclassified into a third class representing the average behavior of the two classes. In addition, scattered misclassification of individual pixels in the MAP segmentations result from the inability to control small scale behavior independently of large scale behavior. The ICM algorithm performed very poorly.

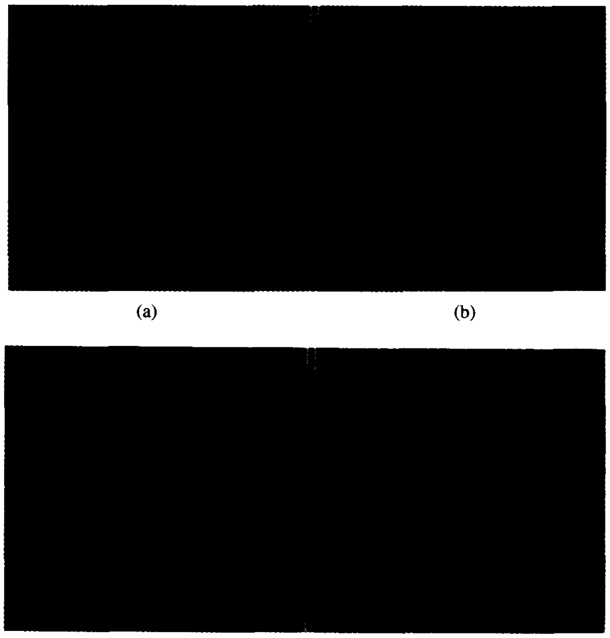


Fig. 7. Segmentations of image 1. Each color denotes a different class: (a) SMAP; (b) MAP with 500 iterations of SA; (c) MAP with 100 iterations of SA; (d) MAP with ICM.

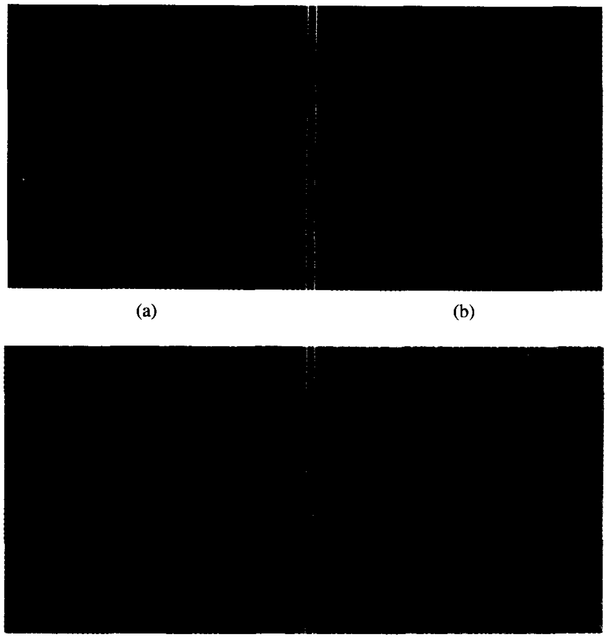


Fig. 8. Segmentations of image 2. Each color denotes a different class: (a) SMAP; (b) MAP with 500 iterations of SA; (c) MAP with 100 iterations of SA; (d) MAP with ICM.

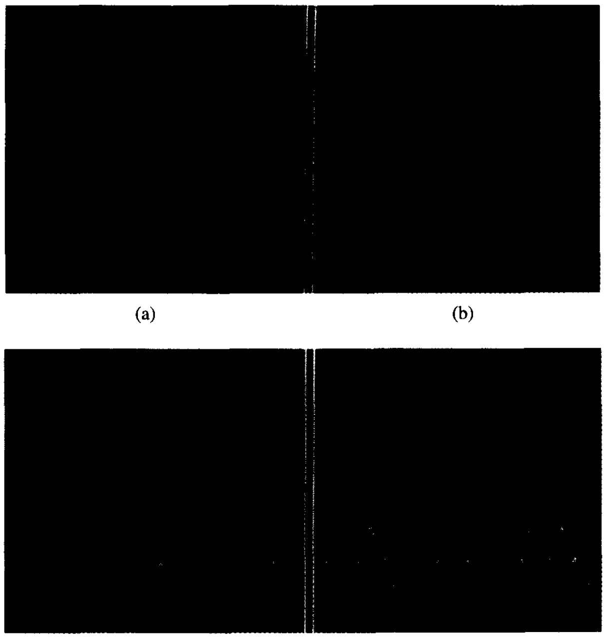


Fig. 9. Segmentations of image 3. Each color denotes a different class: (a) SMAP; (b) MAP with 500 iterations of SA; (c) MAP with 100 iterations of SA; (d) MAP with ICM.

TABLE IV
NUMBER OF REPLACEMENT OPERATIONS REQUIRED PER IMAGE PIXEL FOR THE THREE SYNTHETIC TEST IMAGES. THE SMAP ALGORITHM IS LISTED WITH AND WITHOUT PARAMETER ESTIMATION.

	Replacements per pixel						
	SMAP (no estimation)	SMAP (subsampled estimation)	SMAP (no subsampling)	SA 500	SA 100	ICM	MRS
image1	1.33	3.13	9.12	504	105	28	10.81
image2	1.33	3.55	10.47	506	108	28	10.81
image3	1.33	3.14	8.15	505	104	10	8.89

Fig. 8 shows that for image 2, the overall performance of the SMAP algorithm is substantially better than 500 iterations of SA for all classes but 4. As before, false average classes formed at region boundaries for the MAP segmentation. For this image, both the 100 iterations of SA and ICM failed to properly segment the image.

Fig. 9 is distinct because the mean of each region is equal, and classes are only distinguished by their variance or texture. In this example, the SMAP algorithm produces better results than the alternative methods. The largest regions are completely absent in the 500 iterations of SA. This may be due to the suboptimal solution of SA or the nature of MAP criteria. In addition, the 100 iterations of SA have not converged.

Table IV shows the number of replacement operations required per image pixel for each method. For the SMAP algorithm, replacement operations include either of the following two basic operations: evaluation of a pixel's class using (13) or evaluation of the expectation term in (16) at a single pixel. All of these replacement operations are comparable since they each require order M operations.

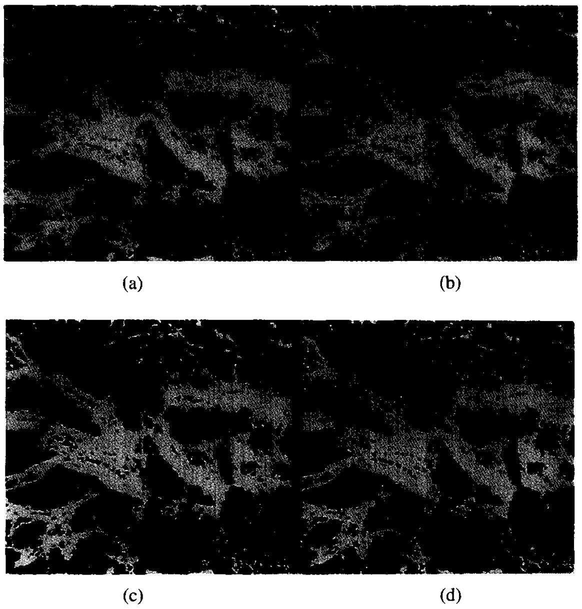
TABLE V
EFFECT OF SUBSAMPLING ON PARAMETER ESTIMATION

		Estimated Parameters 0-3			
		$(\theta_{0,0},\theta_{0,1})$	$(\theta_{1,0},\theta_{1,1})$	$(\theta_{2,0},\theta_{2,1})$	$(\theta_{3,0},\theta_{3,1})$
image	Subsampling	(0.9872, 0.9981)	(0.9815, 0.9990)	(0.9718, 0.9953)	(0.9607, 1.0000)
1	No Subsampling	(0.9876, 0.9980)	(0.9833, 0.9989)	(0.9723, 0.9974)	(0.9533, 0.9952)
image	Subsampling	(0.9835, 0.9945)	(0.9749, 0.9954)	(0.9677, 0.9932)	(0.9523, 1.0000)
2	No Subsampling	(0.9851, 0.9961)	(0.9784, 0.9971)	(0.9688, 0.9963)	(0.9531, 0.9936)
image	Subsampling	(0.9866, 0.9989)	(0.9769, 0.9998)	(0.9629, 0.9963)	(0.9175, 1.0000)
3	No Subsampling	(0.9860, 0.9989)	(0.9773, 0.9998)	(0.9594, 0.9979)	(0.9197, 0.9947)
		Estimated Parameters 4-7			
		$(\theta_{4,0},\theta_{4,1})$	$(\theta_{5,0},\theta_{5,1})$	$(\theta_{6,0},\theta_{6,1})$	$(\theta_{7,0},\theta_{7,1})$
image	Subsampling	(0.9124, 0.9899)	(0.9132, 0.9714)	(0.9444, 0.9326)	(1.0000, 1.0000)
1	No Subsampling	(0.9124, 0.9900)	(0.9132, 0.9712)	(0.9444, 0.9325)	(1.0000, 1.0000)
image	Subsampling	(0.9126, 0.9852)	(0.9004, 0.9745)	(0.9444, 0.9328)	(1.0000, 1.0000)
2	No Subsampling	(0.9126, 0.9852)	(0.9004, 0.9745)	(0.9444, 0.9329)	(1.0000, 1.0000)
image	Subsampling	(0.7728, 0.9823)	(0.7214, 0.9804)	(0.6401, 0.9329)	(0.2500, 1.0000)
3	No Subsampling	(0.7728, 0.9823)	(0.7214, 0.9804)	(0.6401, 0.9329)	(0.2500, 1.0000)

Fig. 10. 430 × 400 subregion of a multispectral remotely sensed SPOT image. Ground truth measurements were taken at 90 positions (transects) located throughout the full image. Each transect is approximately 5 pixels in size.

For each test image, the SMAP algorithm with parameter estimation (and subsampling) required considerably less computation than ICM and much less than SA. The addition of parameter estimation to the SMAP algorithm increased the computation by approximately a factor of 2. Image 3 is the worst case with an increase of 2.66. We note that this measure of computation does not include the calculation of likelihood functions.

In order to test accuracy with real data, a multispectral remotely sensed image was segmented, and the results were compared with measured ground truth to determine classification accuracy. Fig. 10 shows a typical 430 × 400 subregion of a 927 × 1097 three-band SPOT image with a spatial resolution of 20 m. The SPOT image is displayed in color with the infrared band as red, the visible red band as green, and the green band as blue. Ground truth information was collected along 90 positions (transects) placed randomly throughout the full image [46] (only a subset of these transects are contained in the displayed subregion). Each transect is 100 m long (approximately 5 pixels) with a random orientation. Along each transect, detailed measurements of ground cover were made, but for this experiment, we only consider five classes based on the aggregate measure of percent bare ground for each transect. Sixty of the transects were randomly chosen to use for training the class models, and the remaining 30 were used to test the segmentation algorithms performance.



(a) (b)

(c) (d)

Fig. 11. Segmentations of multispectral remotely sensed SPOT image for subregions of size 430 × 400: (a) SMAP segmentation; (b) MAP with 100 iterations of SA; (c) MAP with ICM; (d) maximum likelihood segmentation. For each image class 1 → black; class 2 → red; class 3 → green; class 4 → blue; class 5 → white.

Each of the five classes was modeled as a multivariate Gaussian mixture. The parameters of the mixture model were estimated using the EM algorithm [45], and the model order was chosen for each class using the Rissanen criteria [47]. The mixture model is important because it captures the multimodal spectral behavior that is typical of most classes.

Subregions of the SMAP, SA, ICM, and maximum likelihood segmentations are shown in Fig. 11. Only 100 iterations of SA were used due to the excessive computation time required to process the large image. The percent misclassification was also computed using the 30 testing transects for each class. The results are tabulated in Table VI. For each class and algorithm, the average region size was also computed.

TABLE VI
TABULATED RESULTS FOR THE SEGMENTATION OF MULTISPECTRAL SPOT DATA WITH GROUND TRUTH. CLASSES WERE FORMED FROM RANGES OF PERCENTAGE BARE GROUND, AND PERCENT CLASSIFICATION ACCURACY WAS TABULATED FOR EACH CLASS. THE AVERAGE REGION SIZE FOR EACH CLASS IS ALSO LISTED.

		Class 1	Class 2	Class 3	Class 4	Class 5	Class Average
Percent Bare Ground		0%	1-10%	11-21%	21-30%	31-100%	
Accuracy	SMAP	88.1%	27.8%	17.5%	27.5%	93.5%	50.88%
	SA 100	88.1%	22.5%	17.5%	27.5%	96.0%	50.32%
	ICM	87.5%	23.2%	17.5%	27.5%	93.5%	49.84%
	ML	78.5%	26.5%	17.5%	20.0%	85.1%	45.52%
Average Region Area	SMAP	131.7	18.7	10.7	21.2	57.7	48.0
	SA 100	126.9	16.4	7.5	14.7	49.9	43.1
	ICM	112.2	13.2	7.2	12.9	44.2	37.9
	ML	32.1	3.9	3.9	4.7	16.1	12.1

The classification accuracy of the SMAP segmentation was substantially better than the maximum likelihood algorithm and slightly better than ICM. The SMAP and SA algorithms had approximately comparable accuracy with SMAP performing 5.3% better for class 2 and SA performing 2.5% better for class 5. Inspection of the segmented images in Fig 11 indicates that the SA algorithm produced smoother boundaries than the SMAP result. This extra smoothing of the SA algorithm tends to remove the smaller regions that are more common in class 2.

It is also interesting that the SMAP segmentation produces the largest average region sizes. This is because the SMAP segmentation produces fewer regions containing very few (e.g., 1 or 2) pixels.

V. CONCLUSION

We have presented a new criteria and model for statistical image segmentation. The SMAP estimator is proposed because it minimizes the expected size of the largest misclassified region, and it results in a segmentation algorithm that is computationally simple. The MSRF model uses a pyramid structure to capture the characteristics of image behavior at various scales. Because the MSRF has a Markov chain structure, its parameters can be estimated efficiently from the image during segmentation.

Experiments with synthetic data indicate that the SMAP algorithm performs comparably to or substantially better than MAP estimation using a Markov random field model and simulated annealing. In addition, the SMAP algorithm requires less computation than ICM and much less then simulated annealing. The SMAP algorithm was also tested on multispectral SPOT data and found to improve segmentation accuracy over ML segmentation.

APPENDIX A

Assume that $\hat{x}^{(i)}$ has been computed for $i > n$. We will then compute $\hat{x}^{(n)}$ using the Markov chain structure of the random fields $X^{(n)}$: (see the bottom of the page.) We next define a

$$\begin{aligned}
\hat{x}^{(n)} &= \arg\max_{x^{(n)}} \max_{x^{(i)}\ i\neq n} \sum_{k=0}^{L} 2^k P(X^{(i)} = x^{(i)}\ i \geq k | Y = y) \\
&= \arg\max_{x^{(n)}} \max_{x^{(i)}\ i<n} \max_{x^{(i)}\ i>n} \sum_{k=0}^{L} \\
&\quad 2^k P(Y \in dy, X^{(i)} = x^{(i)} i \geq k) \\
&= \arg\max_{x^{(n)}} \max_{x^{(i)}\ i<n} \left\{ \sum_{k=0}^{n} 2^k P\Big(Y \in dy, X^{(i)} = x^{(i)}\ k \leq i \leq n | X^{(i)} = \hat{x}^{(i)}\ i > n) P(X^{(i)} = \hat{x}^{(i)}\ i > n) \right. \\
&\qquad\qquad \left. + \sum_{k=n+1}^{L} 2^k P(Y \in dy, X^{(i)} = \hat{x}^{(i)}\ i > k\Big) \right\} \\
&= \arg\max_{x^{(n)}} \max_{x^{(i)}\ i<n} \sum_{k=0}^{n} \\
&\quad 2^k P(Y \in dy, X^{(i)} = x^{(i)}\ k \leq i \leq n\ | X^{(n+1)} = \hat{x}^{(n+1)}) \\
&= \arg\max_{x^{(n)}} \left\{ P\Big(Y \in dy, X^{(n)} = x^{(n)} | X^{(n+1)} = \hat{x}^{(n+1)}) + \max_{x^{(i)}\ i<n} \sum_{k=0}^{n-1} 2^{k-n} P(Y \in dy, X^{(i)} \right. \\
&\qquad \left. = x^{(i)}\ k \leq i \leq n | X^{(n+1)} = \hat{x}^{(n+1)}\Big) \right\}.
\end{aligned}$$

residue term $R(x^{(n)})$ so that the following equality holds:

$$
\begin{aligned}
\hat{x}^{(n)} &= \arg\max_{x^{(n)}} P(Y \in dy, X^{(n)} = x^{(n)} | \\
&\qquad\qquad X^{(n+1)} = \hat{x}^{(n+1)})(1 + R(x^{(n)})) \\
&= \arg\max_{x^{(n)}} p_{x^{(n)}|x^{(n+1)},y}(x^{(n)}|\hat{x}^{(n+1)}, y)(1 + R(x^{(n)}))
\end{aligned}
$$

Specifically, $R(x^{(n)})$ is given by

$$
R(x^{(n)}) = \max_{x^{(i)}\ i<n} \times \frac{\sum_{k=0}^{n-1} 2^{k-n} P(Y \in dy, X^{(i)} = x^{(i)} k \leq i \leq n | X^{(n+1)} = \hat{x}^{(n+1)})}{P(Y \in dy, X^{(n)} = x^{(n)} | X^{(n+1)} = \hat{x}^{(n+1)})}.
$$

Since this expression is the ratio of positive quantities, we know that $R \geq 0$. Further, we may bound R from above as shown in the expression at the bottom of this page. Finally, we have that

$$
\begin{aligned}
&\hat{x}^{(n)} \\
&= \arg\max_{x^{(n)}} \Big\{ \log p_{x^{(n)}|x^{(n+1)},y}(x^{(n)}|\hat{x}^{(n+1)}, y) \\
&\quad + \log(1 + R(x^{(n)})) \Big\} \\
&= \arg\max_{x^{(n)}} \Big\{ \log p_{x^{(n)}|x^{(n+1)},y}(x^{(n)}|\hat{x}^{(n+1)}, y) + \mathcal{E}(x^{(n)}) \Big\}
\end{aligned}
$$

where

$$
\begin{aligned}
0 \leq \mathcal{E}(x^{(n)}) &= \log(1 + R(x^{(n)})) \\
&\leq \max_{x^{(n-1)}} p_{x^{(n-1)}|x^{(n)},y}(x^{(n-1)}|x^{(n)}, y).
\end{aligned}
$$

APPENDIX B

In this Appendix, we show by induction that for a quadtree-based pyramid structure the distribution of Y given $X^{(0)}$ has the form of (7) and that the terms in the product may be computed using the recursion of (8). For $n = 0$, these relations are true by assumption. Therefore, assuming the result for scale n yields

$$
\begin{aligned}
&P(Y \in dy | X^{(n+1)} = x^{(n+1)}) \\
&= \sum_{x^{(n)}} P(Y \in dy | X^{(n)} = x^{(n)}) p_{x^{(n)}|x^{(n+1)}}(x^{(n)}|x^{(n+1)}) \\
&= \sum_{x^{(n)}} \left\{ \prod_{r \in S^{(n)}} p_{y_r^{(n)}|x_r^{(n)}}(y_r^{(n)}|x_r^{(n)}) \right\} \\
&\quad \left\{ \prod_{r \in S^{(n)}} p_{x_r^{(n)}|x_{\partial r}^{(n+1)}}(x_r^{(n)}|x_{\partial r}^{(n+1)}) \right\} \\
&= \prod_{r \in S^{(n)}} \sum_{x_r^{(n)}=1}^{M} p_{y_r^{(n)}|x_r^{(n)}}(y_r^{(n)}|x_r^{(n)}) p_{x_r^{(n)}|x_{\partial r}^{(n+1)}}(x_r^{(n)}|x_{\partial r}^{(n+1)}) \\
&= \prod_{s \in S^{(n+1)}} \prod_{r \in d^{-1}(s)} \\
&\quad \sum_{x_r^{(n)}=1}^{M} p_{y_r^{(n)}|x_r^{(n)}}(y_r^{(n)}|x_r^{(n)}) p_{x_r^{(n)}|x_s^{(n+1)}}(x_r^{(n)}|x_s^{(n+1)}) \\
&= \prod_{s \in S^{(n+1)}} p_{y_s^{(n+1)}|x_s^{(n+1)}}(y_s^{(n+1)}|x_s^{(n+1)})
\end{aligned}
$$

where

$$
\begin{aligned}
p_{y_s^{(n+1)}|x_s^{(n+1)}}(y_s^{(n+1)}|k) = \prod_{r \in d^{-1}(s)} \sum_{m=1}^{M} \\
p_{y_r^{(n)}|x_r^{(n)}}(y_r^{(n)}|m) \\
p_{x_r^{(n)}|x_s^{(n+1)}}(m|k).
\end{aligned}
$$

APPENDIX C

In this Appendix, we prove Theorem 1 by extending basic results on the convergence of the EM algorithm [44], [45].

By the stated assumptions and Theorem 4.1(v) of [45], any limit point $\hat{\theta}$ of the sequence $\{\theta^p\}_1^\infty$ has the property that $\hat{\theta} \in \arg\max_{\theta \in \Omega} Q(\theta, \hat{\theta})$. Since Ω is convex and Q is differentiable, this implies that for all $\theta \in \Omega$

$$
D^{1,0} Q(\hat{\theta}, \hat{\theta})(\theta - \hat{\theta}) \leq 0
$$

where $D^{1,0}$ computes the gradient of Q with respect to the first argument.

Let Ω_o be an open set containing Ω such that Q and L are continuous, differentiable on Ω_o. Then, define the continuous, differentiable function $H(\theta, \hat{\theta}) = Q(\theta, \hat{\theta}) - L(\theta)$

$$
\begin{aligned}
R(x^{(n)}) &= \max_{x^{(i)}\ i<n} \sum_{k=0}^{n-1} 2^{k-n} P(X^{(i)} = x^{(i)}\ k \leq i \leq n-1 | X^{(n)} = x^{(n)}, Y = y) \\
&\leq \max_{x^{(n-1)}} \sum_{k=0}^{n-1} 2^{k-n} P(X^{(n-1)} = x^{(n-1)} | X^{(n)} = x^{(n)}, Y = y) \\
&\leq \max_{x^{(n-1)}} P(X^{(n-1)} = x^{(n-1)} | X^{(n)} = x^{(n)}, Y = y) \\
&= \max_{x^{(n-1)}} p_{x^{(n-1)}|x^{(n)},y}(x^{(n-1)}|x^{(n)}, y).
\end{aligned}
$$

on Ω_o. It has been shown [33] that H has the property $\hat{\theta} \in \arg\max_{\theta \in \Omega_o} H(\theta, \hat{\theta})$, which implies that

$$\begin{aligned} DL(\hat{\theta}) &= D^{1,0}Q(\hat{\theta}, \hat{\theta}) + D^{1,0}H(\hat{\theta}, \hat{\theta}) \\ &= D^{1,0}Q(\hat{\theta}, \hat{\theta}). \end{aligned}$$

Therefore, for any $\theta \in \Omega$, we have

$$DL(\hat{\theta})(\theta - \hat{\theta}) = D^{1,0}Q(\hat{\theta}, \hat{\theta})(\theta - \hat{\theta}) \leq 0.$$

Since L is a convex function and Ω is a convex set, this implies that $\hat{\theta}$ is a global maximum of L.

ACKNOWLEDGMENT

The authors would like to thank C. Bagley of the Natural Resources Management Team, US Army CERL, for providing SPOT data and ecological analysis.

REFERENCES

[1] R. Haralick and L. Shapiro, "Image segmentation techniques," *Comput. Vision Graphics Image Processing*, vol. 29, pp. 100–132, 1985.

[2] C. Therrien, "An estimation-theoretic approach to terrain image segmentation," *Comput. Vision Graphics Image Processing*, vol. 22, pp. 313–326, 1983.

[3] J. Besag, "On the statistical analysis of dirty pictures," *J. Roy. Statist. Soc. B*, vol. 48, no. 3, pp. 259–302, 1986.

[4] H. Derin and H. Elliott, "Modeling and segmentation of noisy and textured images using Gibbs random fields," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-9, no. 1, pp. 39–55, Jan. 1987.

[5] J. Besag, "Spatial interaction and the statistical analysis of lattice systems", *J. Roy. Statist. Soc. B*, vol. 36, no. 2, pp. 192–236, 1974.

[6] T. Pappas, "An adaptive clustering algorithm for image segmentation," *IEEE Trans. Signal Processing*, vol. 40, no. 4, pp. 901–914, Apr. 1992.

[7] K. Sauer and C. Bouman, "A local update strategy for iterative reconstruction from projections," *IEEE Trans. Signal Processing*, Feb. 1993,

[8] M. Zhang, R. Haralick, and J. Campbell, "Multispectral image context classification using stochastic relaxation," *IEEE Trans. Syst. Man Cyber.*, vol. 20, no. 1, pp. 128–140, Feb. 1990.

[9] B. Jeon and D. A. Landgrebe, "Spatio-temporal contextual classification based on Markov random field model," in *Proc. 1991 Int. Geosci. Remote Sensing Symp.* (Espoo, Finland), 1991, pp. 1819–1822.

[10] Z. Kato, J. Zerubia, and M. Berthod, "Satellite image classification using a modified Metropolis dynamics," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*(San Francisco, CA), Mar. 23–26, 1992, pp. 573–576, vol. 3.

[11] B. Jeon and D. A. Landgrebe, "Classification with spatio-temporal interpixel class dependency contexts," submitted to *IEEE Trans. Geosci. Remote Sensing*.

[12] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 6, no. 6, pp. 721–741, Nov. 1984.

[13] B. Gidas, "A renormalization group approach to image processing problems," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 11, no. 2, pp. 164–180, Feb. 1989.

[14] C. Bouman and B. Liu, "Segmentation of textured images using a multiple resolution approach," *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing* (New York), Apr. 11–14, 1988, pp. 1124–1127.

[15] C. Bouman and B. Liu, "Multiple resolution segmentation of textured images," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 13, no. 2, pp. 99–113, Feb. 1991.

[16] P. Perez and F. Heitz, "Multiscale markov random fields and constrained relaxation in low level image analysis," in*Proc. IEEE Int. Conf. Acoust. Speech Signal Processing* (San Francisco, CA), Mar. 23–26, 1992, pp. 61–64, vol. 3.

[17] A. Pentland, "Fractal-based description of natural scenes," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-6, pp. 661–674, Nov. 1984.

[18] S. Peleg, J. Naor, R. Hartley, and D. Avnir, "Multiple resolution texture analysis and classification," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-6, no. 4, pp. 518–523, July 1984.

[19] J. Marroquin, S. Mitter, and T. Poggio, "Probabilistic solution of ill-posed problems in computational vision," *J. Amer. Stat. Assoc.* vol. 82, pp 76–89, Mar. 1987.

[20] R. Dubes, A. Jain, S. Nadabar, and C. Chen, "MRF model-based algorithms for image segmentation," in*Proc. 10th Int. Conf. Patt. Recogn.* (Atlantic City, NJ), June 1990, pp. 808–814.

[21] D. Pickard, "Inference for discrete Markov fields: The simplest nontrivial case," *J. Amer. Stat. Assoc.*, vol. 82, pp. 90–96, Mar. 1987.

[22] S. Lakshmanan and H. Derin, "Simultaneous parameter estimation and segmentation of Gibbs random fields using simulated annealing," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 11, no. 8, pp. 799–813, Aug. 1989.

[23] M. Luettgen, W. Karl, A. Willsky, and R. Tenney, "Multiscale representations of Markov random fields," submitted to *IEEE Trans. Signal Processing* Special Issue on Wavelets and Signal Processing.

[24] K. Chou, A. Willsky, A. Benveniste, and M. Basseville, "Recursive and iterative estimation algorithms for multi-resolution stochastic processes," in*Proc. 28th Conf. Decision Contr.* (Tampa, FL), Dec. 13–15, 1989, pp. 1184–1189, vol. 2.

[25] K. Chou, S. Golden, and A. Willsky, "Modeling and estimation of multiscale stochastic processes," in*Proc. IEEE Int. Conf. Acoust. Speech Signal Processing* (Toronto), May 14–17, 1991, pp. 1709–1712.

[26] M. Basseville *et al.*, "Modeling and estimation of multiresolution stochastic processes," *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 766–784, Mar. 1992.

[27] M. Basseville, A. Benveniste, and A. Willsky, "Multiscale autoregressive processes, Part I: Schur-Levinson parametrizations," *IEEE Trans. Signal Processing*, vol. 40, no. 8, pp. 1915–1934, Aug. 1992.

[28] ——, "Multiscale autoregressive processes, Part II: Lattice structures for whitening and modeling," *IEEE Trans. Signal Processing*, vol. 40, no. 8, pp. 1935–1954, Aug. 1992.

[29] K. Chou, S. Golden, M. Luettgen, and A. Willsky, "Modeling and estimation of multiresolution stochastic processes and random fields," in *Proc. Seventh Workshop Multidimensional Signal Processing* (Lake Placid, NY), Sept. 23–25, 1991, p. 3.8.

[30] P. Burt, T. Hong, and A. Rosenfeld, "Segmentation and estimation of image region properties through cooperative hierarchical computation," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. SMC-11, no. 12, pp. 802–809, Dec. 1981.

[31] H. Antonisse, "Image segmentation in pyramids," *Comput. Vision Graphics Image Processing*, vol. 19, pp. 367–383, 1982.

[32] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* San Mateo, CA: Morgan Kaufmann, 1988.

[33] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc. B*, vol. 39, no. 1, pp. 1–38, 1977.

[34] J. Westervelt, M. Shapiro, and D. P. Gerdes, *GRASS Version 4.1 User's Reference Manual*, US Army Construction Eng. Res. Labs,. Office of GRASS Integration, Champain IL, ADP Rep. under preparation 1993.

[35] B. Manjunath, T. Simchony, and R. Chellappa, "Stochastic and deterministic networks for texture segmentation," *IEEE Trans. Acoust. Speech Signal Processing*, vol. 38, no. 6, pp. 1039–1049, June 1990.

[36] K. I. Laws, "Textured image segmentation," Ph.D. dissertation, Dept. Eng., Univ. Southern Calif., Los Angeles, 1980.

[37] M. Unser and M. Eden, "Multiresolution feature extraction and selection for texture segmentation," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 11, no. 7, pp 717–728, July 1989.

[38] ——, "Nonlinear operators for improving texture segmentation based on features extracted by spatial filtering," *IEEE Trans. Syst. Man Cyber.*, vol. 20, no. 4, pp. 804–815, July/Aug. 1990.

[39] D. Landgrebe, "The development of a spectral-spatial classifier for earth observational data," *Patt. Recogn.*, vol. 12, pp. 165–175, 1980.

[40] R. Kettig and D. Landgrebe, "Classification of multispectral image data by extraction and classification of homogeneous objects," *IEEE Trans. Geosci. Electron.*, vol. GE-14, no. 1, pp. 19–26, Jan. 1976.

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

[42] L. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains," *Ann. Math. Statist.*, vol. 41, no. 1, pp. 164–171, 1970.

[43] W. Press, B. Flannery, S. Teukolsky, and Vetterling, *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge, UK: Cambridge University Press, 1988.

[44] C. Wu, "On the convergence properties of the EM algorithm," *Annals Statist.*, vol. 11, no. 1, pp. 95–103, 1983.

[45] E. Redner and H. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Rev.*, vol. 26, no. 2, Apr. 1984.

[46] D. Tazik *et al.*, "U.S. Army land condition-trend analysis (LCTA) plot inventory field methods," USACERL Tech. Rep. N-92/03, Feb. 1992.

[47] J. Rissanen, "A universal prior for integers and estimation by minimum description length," *Annals Statist.*, vol. 11, no. 2, pp. 417–431, 1983.

Charles Bouman (M'89) was born in Philadelphia, PA. He received the B.S.E.E. degree form the University of Pennsylvania in 1981 and the M.S. degree in electrical engineering from the University of California at Berkeley in 1982. In 1987 and 1989, respectively, he received the M.A. and Ph.D. degrees in electrical engineering from Princeton University.

From 1982 to 1985, he was a staff member in the Analog Device Technology Group at the Massachusetts Institute of Technology Lincoln Laboratory. His work there included the development of algorithms and architectures that used wide bandwidth analog signal processing devices. In 1989, he joined the faculty of the School of Electrical Engineering at Purdue University, West Lafayette, IN, as an Assistant Professor. His research interests include image segmentation, multiscale processing, statistical tomography, and the application of human visual system models to the display of printing.

Dr. Bouman is an assosciate editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING

Michael Shapiro was born in Chicago, IL, in 1951. He received the B.S. degree in mathematics and computer science in 1973 and the M.S. degree in mathematics in 1974 from the University of Illinois.

He is currently employed at the U.S. Army Construction Engineering Laboratory, Champaign, IL, as a software enginneer. He has been working on the design and development of the Geographic Resources Analysis Support System (GRASS), which is a geographic information system with image processing capabilities for remotely sensed imagery.