# MULTIRESOLUTION IMAGE SEGMENTATION

*Mary L. Comer and Edward J. Delp*

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

## ABSTRACT

In this paper we present a new algorithm for segmentation of noisy or textured images using a multiresolution Bayesian approach. Our algorithm is different from previously proposed multiresolution segmentation techniques in that we use a multiresolution Gaussian autoregressive (AR) model for the pyramid representation of the observed image. Our algorithm also approximates the "maximization of the posterior marginals" (MPM) estimate of the pixel class labels at each resolution, from coarsest to finest, unlike previously proposed techniques, which have been based on MAP estimation. Experimental results are presented to demonstrate the performance of the new algorithm.

## 1. INTRODUCTION

This paper addresses the problem of segmenting an image using a statistical model. Each pixel in the observed image must be assigned membership to one of a finite number of classes depending on the statistical properties of the pixel and its neighbors. The individual pixel classifications, or labels, form a matrix or two-dimensional field, with the same dimensions as the observed image, in which the value at a given spatial location reflects the class to which the corresponding pixel in the observed image belongs. This matrix containing the individual pixel classifications will be referred to as the *label field.* The label field is unknown and must be estimated from the observed image.

Recently, several multiresolution approaches to image segmentation have been proposed [1, 2, 3]. These approaches estimate the label field first at coarse resolutions and then proceed to finer resolutions to refine the segmentation. In this paper we propose a new segmentation algorithm which uses a multiresolution autoregressive (AR) model to model the observed data pyramid, i.e., the multiresolution decomposition obtained from the observed image. Our approach is different from previously proposed approaches in that we obtain a multiresolution representation of the observed image and model this representation as a stochastic process indexed by the nodes of a multiresolution lattice. Previous approaches have used multiresolution models for the pixel labels, but have used either single-resolution representations of the observed image or multiresolution representations of the observed image with the implicit assumption that the random variables at a given level of the observed data pyramid are independent from the random variables at other levels [1, 2, 3]. We feel that our approach is more appropriate since the model we use incorporates the correlations between different levels of the observed data pyramid.

Our approach is also different from previously proposed multiresolution approaches in that we approximate the "maximization of the posterior marginals" (MPM) estimate of the label field, instead of the MAP estimate [1, 2, 4]. This has two possible advantages. First, the average cost function which the MPM estimate minimizes is more appropriate for image segmentation than the the average cost function which the MAP estimate minimizes [4]. Second, the use of the MPM criterion will allow us to use the EM algorithm to estimate parameters of the probability mass function of the pixel class labels, using the algorithm described in [5].

## 2. NEW APPROACH

In our problem formulation the observed data $\mathbf{Y}$ is a multiresolution representation of the observed image, obtained using a Gaussian pyramid decomposition [6]. Thus, $\mathbf{Y}$ is a stochastic process indexed by the nodes of a multiresolution lattice, such as the one shown in Figure 1. Each level in the lattice corresponds to a different spatial resolution, where level 0 represents the finest spatial resolution and level $M-1$ the coarsest spatial resolution. The set of lattice points at level $n$ will be denoted $S^{(n)}$ and the random field containing the classification of the nodes at level $n$ will be denoted $\mathbf{X}^{(n)}$. Each node in $S^{(n)}$ corresponds to $4^n$ pixels at the original image spatial resolution. Hence, each random variable in $\mathbf{X}^{(n)}$ represents the classification of a block

This work was partially supported by a National Science Foundation Graduate Fellowship. Address all correspondence to E. J. Delp (ace@ecn.purdue.edu or (317)494-1740)



0-7803-2431-5/95 $4.00 

of $4^n$ pixels in the original image.

To segment the observed image, two models will be needed. A model will be needed for the conditional probability density function of $\mathbf{Y}$ given the classifications of all nodes in the multiresolution lattice. We will also need a model for the conditional probability mass function of the label field $\mathbf{X}^{(n)}$ given the MPM approximations of the label fields at coarser resolutions.

### 2.1. Multiresolution AR Model

To simplify the discussion we will assume that the random variables in $\mathbf{Y}$ are zero-mean, and that $\mathbf{Y}$ is indexed by the nodes in a binary tree, as shown in Figure 2. Extension to the lattice shown in Figure 1 is a matter of notation.

In order to define the model for $\mathbf{Y}$, we associate with the binary tree the ordering of the nodes shown in Figure 2. The nodes at level n are indexed from $2^{M-n}$ to $2^{M-n+1}-1$, where $M$ is the number of levels in the tree. Given a particular level $n$ and a particular node $s \in S^{(i)}$ for some $i > n$, the index of the ancestor of node $s$ which lies in level $n$ will be denoted $d^n(s)$. For example, if $s$ is node 11 in Figure 2, then $d^2(s) = 2$ and $d^1(s) = 5$.

The model which will be used is a causal Gaussian multiresolution AR model, where the notion of causality for our model is defined by the ordering of the nodes of the tree defined above. In this model the random variables $Y_1, \ldots, Y_K$, where $Y_s$ is the random variable at node $s$ and $K$ is the total number of nodes in the tree, are modeled as jointly Gaussian random variables conditioned on the classification of the tree nodes. The form of the AR model is

$$Y_s = -a_{x_s}(1)Y_{s-1} - a_{x_s}(2)Y_{\lfloor s/2 \rfloor} + Y'_s \qquad (1)$$

where node $\lfloor s/2 \rfloor$ is the parent of node $s$ in the tree, $\{a_{x_s}(j)\}, j = 1, 2$ are AR model parameters for class $x_s$, and $Y'_1, \ldots, Y'_K$ is a sequence of independent, Gaussian random variables. The variance of $Y'_s$ is given by

$$E[(Y'_s)^2] = \sigma^2_{x_s} \qquad (2)$$

Intuitively, this model can be described as follows: The value of the random variable $Y_s$ can be predicted as a linear combination of the values of the random variables at the previous node at the same resolution level as node $s$ and the parent node of node $s$. The prediction errors $Y'_1, \ldots, Y'_K$ form a sequence of independent random variables. The parameters used for the prediction of $Y_s$ depend on the class to which node $s$ belongs, i.e., $x_s$. The variance of the prediction error $Y'_s$ also depends on $x_s$.

Since the random variables $Y'_1, \ldots, Y'_K$ are independent Gaussian random variables, the form of the joint conditional probability density function of $Y'_1, \ldots, Y'_K$ given the classification of all tree nodes is known. However, we need the joint conditional probability density function of $Y_1, \ldots, Y_K$ given the classification of all tree nodes. By writing the sequences $Y_1, \ldots, Y_K$ and $Y'_1, \ldots, Y'_K$ as vectors, i.e., $\mathbf{Y} = [Y_1, \ldots, Y_K]^T$ and $\mathbf{Y}' = [Y'_1, \ldots, Y'_K]^T$, $\mathbf{Y}'$ can be written as a linear transformation of $\mathbf{Y}$:

$$\mathbf{Y}' = \mathbf{A}\mathbf{Y} \qquad (3)$$

where $\mathbf{A}$ is a matrix which can be determined from Equation 1. Since $\mathbf{Y}'_s$ is a linear combination of values of $\mathbf{Y}$ at node $s$ and nodes which precede node $s$, but not at future nodes, the matrix $\mathbf{A}$ is lower triangular. Also, since the coefficient of the term $Y_s$ in the expression for $Y'_s$ is 1, all diagonal elements of $\mathbf{A}$ are 1. Thus the Jacobian of the transformation from $\mathbf{Y}$ to $\mathbf{Y}'$ is 1, and the conditional probability density function of $\mathbf{Y}$ given $\mathbf{X}$ (where $\mathbf{X}$ is the sequence $X_1, \ldots, X_K$) is [7]

$$f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}, \theta) = \prod_{s=1}^{K} \frac{1}{\sqrt{2\pi\sigma^2_{x_s}}} \exp\left(-\frac{(y'_s)^2}{2\sigma^2_{x_s}}\right) \qquad (4)$$

where

$$y'_s = y_s + a_{x_s}(1)y_{s-1} + a_{x_s}(2)y_{\lfloor s/2 \rfloor} \qquad (5)$$

### 2.2. Model for Class Label Pyramid

At each spatial resolution the pixel class labels are modeled as a Markov random field (MRF). Specifically, the probability mass function of the label pyramid $\mathbf{X}$, $p_{\mathbf{X}}(\mathbf{x}) = P(\mathbf{X} = \mathbf{x}) = P(X_1 = x_1, \ldots, X_N = x_N)$ is assumed to have the form

$$p_{\mathbf{X}}(\mathbf{x}) = \frac{1}{z} \exp\left(-\sum_{\{r,s\} \in \mathcal{C}} \beta t(x_r, x_s)\right) \qquad (6)$$

where $z$ is a normalizing constant, $\beta$ is the spatial interaction parameter, $\mathcal{C}$ is the set of all cliques defined on $S$ (the set of all nodes in the multiresolution lattice), and

$$t(x_r, x_s) = \begin{cases} 0 & \text{if } x_r = x_s \\ 1 & \text{if } x_r \neq x_s \end{cases} \qquad (7)$$

For the work presented in this paper $\mathcal{C}$ contains all pairs of nodes which are vertically or horizontally adjacent and are at the same resolution level in the pyramid. Correlations between different resolution levels in the label pyramid are incorporated into the algorithm by using as an initial value for the segmentation at level $n$ the classification obtained by upsampling the final segmentation at level $n + 1$.

### 2.3. New Algorithm

The segmentation begins at the coarsest resolution level, where $\hat{\mathbf{x}}^{(M-1)}$, an approximation to the MPM estimate of $\mathbf{X}^{(M-1)}$, is obtained using the algorithm described in [5]. The algorithm then proceeds to finer resolutions until $\hat{\mathbf{x}}^{(0)}$, which is the final segmentation, is obtained.

We now describe the new segmentation algorithm at resolution level $n$ for a given value of $n$ between 0 and $M-2$. First, we assume that $\hat{\mathbf{x}}^{(M-1)}, \ldots, \hat{\mathbf{x}}^{(n+1)}$ have already been obtained using the algorithm described here. We will then obtain $\hat{\mathbf{x}}^{(n)}$ using the MPM algorithm described in [5]. To simplify notation, let

$$\bar{\mathbf{X}}^{(n-1)} = (\mathbf{X}^{(0)}, \ldots, \mathbf{X}^{(n-1)}) \tag{8}$$

and

$$\hat{\bar{\mathbf{X}}}^{(n+1)} = (\mathbf{X}^{(n+1)}, \ldots, \mathbf{X}^{(M-1)}) \tag{9}$$

We will use a Gibbs sampler to generate a Markov chain $\mathbf{x}^i$ (where each element in the chain is a random field indexed by the set $S^{(n)}$) which converges in distribution to a random field with conditional probability mass function

$$p_{\mathbf{X}^{(n)}|\mathbf{Y},\bar{\mathbf{X}}^{(n-1)},\bar{\mathbf{X}}^{(n+1)}}(\mathbf{x}^{(n)}|\mathbf{y},\bar{\mathbf{x}}_{d^n}^{(n-1)},\hat{\bar{\mathbf{x}}}^{(n+1)},\boldsymbol{\theta}) \tag{10}$$

which is the conditional probability mass function of the label field at resolution $n$ given the entire multiscale representation $\mathbf{Y}$, the MPM approximations of the label fields at coarser resolutions, and the values of $\mathbf{x}^{(i)}$ for $i < n$ obtained by propagating the classifications $\mathbf{x}^{(n)}$ down the tree to finer resolutions. These values of $\mathbf{x}^{(i)}$ for $i < n$ are denoted $\mathbf{x}_{d^n}^{(i)}$, where the $s$th element of $\mathbf{x}_{d^n}^{(i)}$ is

$$\left(\mathbf{x}_{d^n}^{(i)}\right)_s = x_{d^n(s)}^{(n)} \tag{11}$$

for $i < n$. $\boldsymbol{\theta}$ is a vector containing the unknown parameters of this probability mass function.

The objective of the MPM algorithm described in [5] is to generate a Markov chain which converges in distribution to a random field with conditional probability mass function $p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y},\boldsymbol{\theta})$, where $\mathbf{Y}$ and $\mathbf{X}$ are single-resolution random fields and $\boldsymbol{\theta}$ contains the mean and variance of each class. The problem we are solving here is the same, except that now the Markov chain generated by the Gibbs sampler will converge in distribution to a random field with conditional probability mass function

$$p_{\mathbf{X}^{(n)}|\mathbf{Y},\bar{\mathbf{X}}^{(n-1)},\bar{\mathbf{X}}^{(n+1)}}(\mathbf{x}^{(n)}|\mathbf{y},\bar{\mathbf{x}}_{d^n}^{(n-1)},\hat{\bar{\mathbf{x}}}^{(n+1)},\boldsymbol{\theta}) \tag{12}$$

Using Bayes rule to determine the form for the conditional probability mass function which will be used to implement the Gibbs sampler, we obtain

$$p_{\mathbf{X}^{(n)}|\mathbf{Y},\bar{\mathbf{X}}^{(n-1)},\bar{\mathbf{X}}^{(n+1)}}(\mathbf{x}^{(n)}|\mathbf{y},\bar{\mathbf{x}}_{d^n}^{(n-1)},\hat{\bar{\mathbf{x}}}^{(n+1)},\boldsymbol{\theta}) =$$

$$\frac{f_{\mathbf{Y}|\bar{\mathbf{X}}^{(n-1)},\mathbf{X}^{(n)},\bar{\mathbf{X}}^{(n+1)}}(\mathbf{y}|\bar{\mathbf{x}}_{d^n}^{(n-1)},\mathbf{x}^{(n)},\hat{\bar{\mathbf{x}}}^{(n+1)},\boldsymbol{\theta})}{f_{\mathbf{Y}|\bar{\mathbf{X}}^{(n-1)},\bar{\mathbf{X}}^{(n+1)}}(\mathbf{y}|\bar{\mathbf{x}}_{d^n}^{(n-1)},\hat{\bar{\mathbf{x}}}^{(n+1)},\boldsymbol{\theta})}$$

$$p_{\mathbf{X}^{(n)}|\bar{\mathbf{X}}^{(n-1)},\bar{\mathbf{X}}^{(n+1)}}(\mathbf{x}^{(n)}|\bar{\mathbf{x}}_{d^n}^{(n-1)},\hat{\bar{\mathbf{x}}}^{(n+1)},\boldsymbol{\theta}) \tag{13}$$

The term in the denominator does not depend on $\mathbf{x}^{(n)}$. The form for the conditional probability density function of $\mathbf{Y}$ will be determined using Equation 4 and writing the conditional probability density function of $\mathbf{Y}$ as

$$f_{\mathbf{Y}|\bar{\mathbf{X}}^{(n-1)},\mathbf{X}^{(n)},\bar{\mathbf{X}}^{(n+1)}}(\mathbf{y}|\bar{\mathbf{x}}_{d^n}^{(n-1)},\mathbf{x}^{(n)},\hat{\bar{\mathbf{x}}}^{(n+1)},\boldsymbol{\theta}) =$$

$$\prod_{s\in S^{(i)}, i\geq n}^{K} \frac{1}{\sqrt{2\pi\sigma_{x_s}^2}} \exp\left(-\frac{(y_s')^2}{2\sigma_{x_s}^2}\right) \cdot$$

$$\prod_{s\in S^{(i)}, i<n}^{K} \frac{1}{\sqrt{2\pi\sigma_{x_{d^n(s)}}^2}} \exp\left(-\frac{(y_s')^2}{2\sigma_{x_{d^n(s)}}^2}\right) \tag{14}$$

where in the second term $y_s'$ is computed using the coefficients $\{a_{x_{d^n(s)}}(j)\}$. Equation 14 is the equation which will be used to incorporate the observed data $\mathbf{Y}$ into the Gibbs sampler to implement the MPM algorithm.

After the estimate $\hat{\mathbf{x}}^{(n)}$ has been obtained, the segmentation algorithm proceeds to resolution $n-1$ to obtain $\hat{\mathbf{x}}^{(n-1)}$, using as an initial value for $\hat{\mathbf{x}}^{(n-1)}$ the label field obtained by assigning at each node in $S^{(n-1)}$ the classification of the node's parent at resolution $n$. This process is continued until $\hat{\mathbf{x}}^{(0)}$ has been obtained.

## 3. EXPERIMENTAL RESULTS

For the experimental results described in this section, the parameters which were assumed to be unknown were the means of the $L$ classes in the observed data pyramid. These means were estimated simultaneously with the segmentation as described in [5]. The values of the AR parameters (prediction coefficients and prediction error variances) and the spatial interaction parameter $\beta$ were pre-selected.

Figure 3 shows results after applying the algorithm described in the previous section to a textured image composed of grass and wood. A three-level pyramid was used. Fifty iterations of the algorithm in [5] were performed at the coarsest resolution, with 10 iterations of the algorithm described above performed at each of the finer resolutions.

Figure 4 shows results after applying the segmentation algorithm to an x-ray mammography image. The image shown on the left in Figure 4 was segmented using 3 pyramid levels, 10 iterations of the algorithm in [5] at the coarse resolution, and 10 iterations of the algorithm described above at each of the finer resolutions. The truth image, which identifies the location of a stellate lesion, is shown in the middle in the figure, and the segmented image is shown on the right. The lesion was well segmented from its surrounding area.

## 4. CONCLUSION

We have presented a new multiresolution segmentation algorithm. In future work we plan to add parameter estimation for the AR parameters, investigate different models for the label pyramid, and study the computational complexity of the algorithm.

A postscript version of this paper and the image data sets are available via anonymous ftp to `skynet.ecn.purdue.edu` in the directory `/pub/dist/delp/icassp95`.

## 5. REFERENCES


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

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

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

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

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

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

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


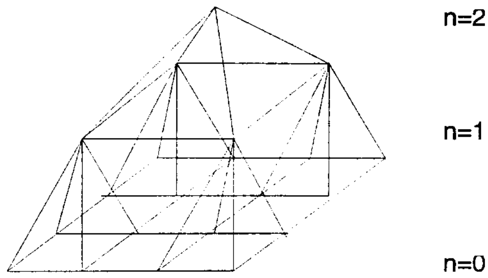


Figure 1: Multiresolution lattice

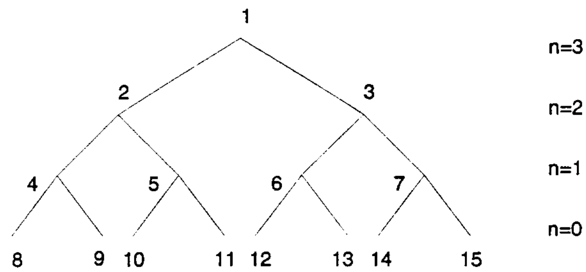


Figure 2: Binary tree

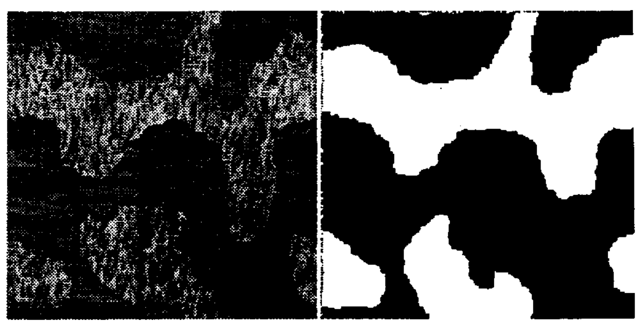

Figure 3: Left: Original image; Right: Segmented image.

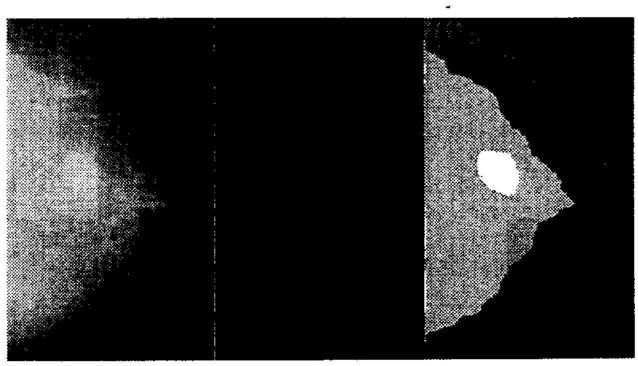

Figure 4: Left: Original image; Middle: Truth image; Right: Segmented image.