Multiscale Image Segmentation Using Joint Texture and Shape Analysis

Ramesh Neelamani, Justin Romberg, Hyeokho Choi, Rudolf Riedi, and Richard Baraniuk

Department of Electrical and Computer Engineering, Rice University, Houston, TX 77005, USA

ABSTRACT

We develop a general framework to simultaneously exploit texture and shape characterization in multiscale image segmentation. By posing multiscale segmentation as a model selection problem, we invoke the powerful framework offered by minimum description length (MDL). This framework dictates that multiscale segmentation comprises multiscale texture characterization and multiscale shape coding. Analysis of current multiscale maximum a posteriori (MAP) segmentation algorithms reveals that these algorithms implicitly use a shape coder with the aim to estimate the optimal MDL solution, but find only an approximate solution.

Towards achieving better segmentation estimates, we first propose a shape coding algorithm based on zero-trees which is well-suited to represent images with large homogeneous regions. For this coder, we design an efficient tree-based algorithm using dynamic programming that attains the optimal MDL segmentation estimate. To incorporate arbitrary shape coding techniques into segmentation, we design an iterative algorithm that uses dynamic programming for each iteration. Though the iterative algorithm is not guaranteed to attain exactly optimal estimates, it more effectively captures the prior set by the shape coder. Experiments demonstrate that the proposed algorithms yield excellent segmentation results on both synthetic and real world data examples.

Keywords: Segmentation, texture, shape, minimum description length (MDL), wavelets, hidden Markov trees (HMT)

1. INTRODUCTION

An image segmentation algorithm aims to assign a *class label* to each pixel of an image based on the properties of the pixel and its relationship with its neighbors. A "good" segmentation separates an image into simple regions with homogeneous properties, each with a different "texture".[1]

Recently, many authors have applied statistical techniques to jointly estimate the region shapes and determine their classes.[2–4] Statistical techniques regard a sampled image x with support S as a realization of a random field X with distinct and consistent stochastic behavior in different regions. Let $R_1, R_2, \ldots, R_k$ be a partition of the support S of x; i.e., $R_i \bigcap R_j = \emptyset, i \neq j$, and $\bigcup_i R_i = S$. Let x_{R_i} denote the subset of image pixels supported by the region R_i. We assume that each region R_i is fully covered by the texture $\lambda_i \in \{\Lambda_1, \Lambda_2, \ldots, \Lambda_{N_c}\}$ and that no two neighboring R_i contain the same texture. The pixels in the image subregion x_{R_i} are assumed distributed with joint probability density function (pdf) $f(x_{R_i}|\lambda_i)$. Segmentation involves separating the image into regions R_i and assigning the corresponding texture λ_i to each. The problem can also be rephrased as: given an image x, estimate for each pixel a class label from $\{\Lambda_1, \Lambda_2, \ldots, \Lambda_{N_c}\}$. A realization $c \in \{\Lambda_1, \Lambda_2, \ldots, \Lambda_{N_c}\}^S$ with support S from the *labeling field* C records the class label λ of each pixel and is called the *segmentation map.*

Email: neelsh,jrom,choi,riedi,richb@ece.rice.edu. Web: www.dsp.rice.edu. This work was supported by the National Science Foundation grant CCR–99–73188, DARPA/AFOSR grant F49620–97–1–0513, ONR grant N00014–99–1–0219, Texas Instruments, and the Rice Consortium for Computational Seismic/Signal Interpretation.

1.1. Maximum likelihood (ML) segmentation

The maximum likelihood (ML) segmentation estimate is obtained by maximizing the likelihood $f(x|c)$, i.e.,

$$c^{\mathrm{ML}} := \arg\max_{c} f(x|c). \tag{1}$$

However, the joint likelihood $f(x|c)$ of the entire image is never available in practice and needs to be estimated. A common simplifying assumption in likelihood estimation is that the pixels $x_{i,j}$ of the image are independent given the segmentation map c

$$f(x|c) \approx \prod_{x_{i,j} \in x} f(x_{i,j}|c). \tag{2}$$

ML segmentation now reduces to pixel-wise maximization of $f(x|c)$. The independence assumption, though convenient, is unfortunately not valid, since texture characteristics are not localized to individual pixels. Hence, estimates of $f(x_{i,j}|c)$ are not robust, making ML segmentation estimates unsatisfactory.

1.2. Maximum a posteriori (MAP) segmentation

ML segmentation implicitly assumes that any arrangement of class labels is equally likely. This assumption is undesirable, since, for example, large homogeneous regions are more likely in typical applications. MAP segmentation attempts to capture the class label spatial dependence by maximizing the likelihoods weighted by a prior $p(c)$ for each segmentation map c, i.e.,

$$c^{\mathrm{MAP}} := \arg\max_{c} f(x|c)p(c). \tag{3}$$

Such a modification can potentially provide better estimates than the ML method. However, MAP proves intractable, since there exist far too many possible arrangements of class labels. Thus, while ML estimates are useless, exact MAP estimates are intractable.

1.3. MAP $\equiv$ MDL

MAP segmentation aims to maximize the likelihood weighted by the priors on the class labels (see (3)). Alternatively, taking negative logs in (3), MAP estimation can also be expressed as

$$c^{\mathrm{MAP}} = \arg\min_{c}[-\log_2 f(x|c) - \log_2 p(c)]. \tag{4}$$

From an information theoretic perspective, the two terms $-\log_2 f(x|c)$ and $-\log_2 p(c)$ can be interpreted as the code length in bits required to describe the observed image assuming a given segmentation map c, and the code length required to describe the c respectively.[5] Thus, MAP estimation simply aims to find c that minimizes the total number of bits required to describe both the data and the segmentation map. This framework is commonly known as the *minimum description length* (MDL) principle.[6–8]

Rissanen proposed the MDL principle as an information theoretic approach to model selection.[6–8] MDL advocates that from a given choice of competing models, the "best" model is the one that provides the shortest *description*. The description comprises of the number of bits required to code the data assuming the given model, and the number of bits required to describe the model itself.

Leclerc[9] and Keeler[10] first posed segmentation as a model selection problem and invoked MDL to find the solution. Later, Kerfoot analyzed the MDL criterion for segmentation in more detail.[11] The different competing models are the arrangements of the class labels, with each class label corresponding to a pixel in the image to be segmented. For each model c, if the likelihood $f(x|c)$ and the prior $p(c)$ are available, then the MDL estimate can be obtained by minimizing the description length (see (4)).

Since MAP and MDL are equivalent, the MDL estimate at the pixel level also has to deal with the problems of unreliable likelihood estimates and too many possible models to search from.

1.4. Multiscale MAP segmentation

The distinguishing characteristics of a texture are not localized to individual pixels but are observed over a neighborhood of pixels. Hence, *classification windows* of different sizes are used by current multiscale MAP segmentation algorithms.[12–15] Large windows usually provide sufficiently accurate likelihood estimates by capturing the neighborhood dependencies using the rich statistical information available. Consequently, classification using ML produces accurate segmentations in large homogeneous regions. However, since a large window risks containing pixels of different classes, the resolution of the resulting segmentation is unacceptable particularly along the boundaries between regions. In contrast, a small window is more appropriate near the boundaries between regions, because it reduces the possibility of having multiple classes in the window. However, ML classification of small windows is unreliable due to the paucity of statistical information. To overcome this "blockiness vs. robustness" tradeoff, multiscale MAP segmentation builds on classification decisions made at coarser scales in the following way (see Figures 2 and 3 for more details on the notations used):

1. Given the classification decisions at the coarser scale, a *context*, typically chosen as a group of pixels neighboring the parent in the coarser scale, is defined. A predefined function uses this context to set a prior on the class label of the current pixel. The function is estimated by either ad hoc rules or by using training samples.[12–15]
2. With the prior and the likelihood for the pixels at the current resolution now available, MAP classification finds the class label that maximizes the likelihood weighted by the prior obtained in step 1. This yields the segmentation map at the current resolution.

This process is iterated to finally obtain a segmentation map at the finest resolution. By making concrete class labeling decisions at each resolution, and assuming that the class labels of the child are independent given the class labels of the parents, the problem of optimizing over all possible arrangements of class labels is avoided.[12–15]

1.5. Multiscale MAP $\equiv$ multiscale MDL

Given the classification window class labels, the likelihoods used by current multiscale MAP algorithms define the code length required to describe the observed image at different resolutions. The priors calculated using the context equivalently define the code length to describe each class label. If the collection of class labels at all scales are considered to be one of the competing models, then the function defining the priors provides the description length for each competing model. Thus, posed within an MDL framework, current multiscale MAP algorithms can be interpreted as trying to estimate the optimal multiscale MDL model that minimizes the description length of the data.

1.6. Elements of multiscale segmentation

The formulation of multiscale MAP segmentation using MDL is significant because MDL provides a clean framework to decompose multiscale segmentation into two distinct components, namely, texture characterization and shape coding. Henceforth, for the sake of clarity, we will assume that the image segmentation involves two class labels.

1.6.1. Texture characterization

Estimating the code length in bits to describe the observed image assuming a class label requires a pdf model for each class. Different techniques can be used to estimate the pdf models.[16–18] One such technique proposed by Crouse et al[16] uses the *hidden Markov tree* (HMT) model, a parametric statistical model for wavelet transforms that characterizes textures. Since textures are well characterized by their *singularity* (edge and ridge) structure, the wavelet domain, which facilitates efficient multiscale edge detection, is well-suited for modeling and characterizing different textures. The robust likelihood estimates provided by the HMT models are naturally arranged in the form of a quad-tree (see Figure 3).

1.6.2. Shape coding

For a two-class problem, describing the class labels is equivalent to binary image coding. Since coding a binary image is equivalent to coding the shape of its boundaries, binary shape coders can provide the code length in bits required to describe the class labels. Two simple binary shape coders, zero-tree significance map (ZSM) coder and the edge-persistent (EP) coder described in the later sections complement the quad-tree structure of the likelihoods provided by texture characterization techniques such as the HMT.[16] The ZSM coder is well-suited to represent binary images with large homogeneous regions as is typical in segmentation maps. Similar to the ZSM coder, the EP coder is also suited to represent binary images, but in addition, it also accounts for the evolution of the boundaries of the homogeneous regions more efficiently.

1.7. Realizations using the MDL formulation

Using the clean framework provided by MDL, we can now identify the shortcomings of current multiscale segmentation techniques. Thanks to MDL, we can link mature fields of binary image coding and shape characterization to segmentation.

Current multiscale MAP algorithms make hard irreversible segmentation decisions at coarse scales that impose a prior for segmentation at finer scales. However, it is easy to envision situations where such decisions are locally optimal, but globally sub-optimal. Hence, the current MAP algorithms do not attain the optimal MDL estimate.

Current multiscale segmentation algorithms resort to training or ad hoc methods to choose a function that defines the prior on the different pixels, thereby tuning the segmentation algorithm to specific application scenarios. From an MDL perspective, this is equivalent to determining the "right" coder to describe the binary class labels at different resolutions. Immediately, we realize that the "right" coder is the one that efficiently encodes the collection of the different resolution class label binary images. Since the binary coding is a mature field, we can now tap into its rich literature for potential solutions. Further, since it is easy to characterize shape in binary images, we can now potentially incorporate explicit shape characterization concepts as well so as to tune the segmentation to favor specific shapes.

The ability to explicitly incorporate different coding methods and notions of shape available in the literature, however, hinges on designing algorithms that can efficiently search over the many possible model classes.

1.8. Tree-based joint texture and shape analysis

Since the likelihood estimates provided by the HMT and the class label description provided by shape coding techniques such as ZSM coder and EP coder are conveniently defined on quad-trees, we are naturally led to searching for tree-based algorithms to estimate the optimal MDL solution.

In a ZSM coder, the code length of a child is determined only by the corresponding parent on the quad-tree. Hence, it becomes possible to use a dynamic programming algorithm[19] consisting of a simple up-down sweep of the quad-tree to estimate the optimal segmentation map. In contrast to current multiscale MAP algorithms, the class labeling decisions at any level in this algorithm are influenced not only by the parent class labels, but also by the class labeling decisions and likelihoods at all other levels. In an EP coder, the code length of the child is determined by not only the parent but also by its neighbors. The efficient ZSM dynamic programming algorithm is not immediately applicable in the EP case. Hence, we design an iterative algorithm based on dynamic programming for each iteration to search for the multiscale segmentation map. The framework of the algorithm is general enough to accommodate and exploit the global perspective of any shape-based coding algorithm.

Both the proposed segmentation algorithms provide robust estimates for the synthetic example shown in Figure 1. However, the iterative EP coder algorithm has better resolution along the boundaries in the segmentation map. As shown in Figure 7, the proposed algorithms provide excellent estimates for real world examples as well.

2. MDL AND SEGMENTATION

The MDL principle adopts an information theoretic approach to address the following philosophical model selection question: among a given set of models, which model provides the "best" description for the observed data? Rissanen quantified "best" using the notion of code length in bits[5] as: the smaller the code length, the better the model.[6–8,21] Thus, MDL's answer to the model selection problem is to choose the model that gives the minimum description length. The MDL principle also agrees with statistical inference's fundamental philosophy proposed by the founding

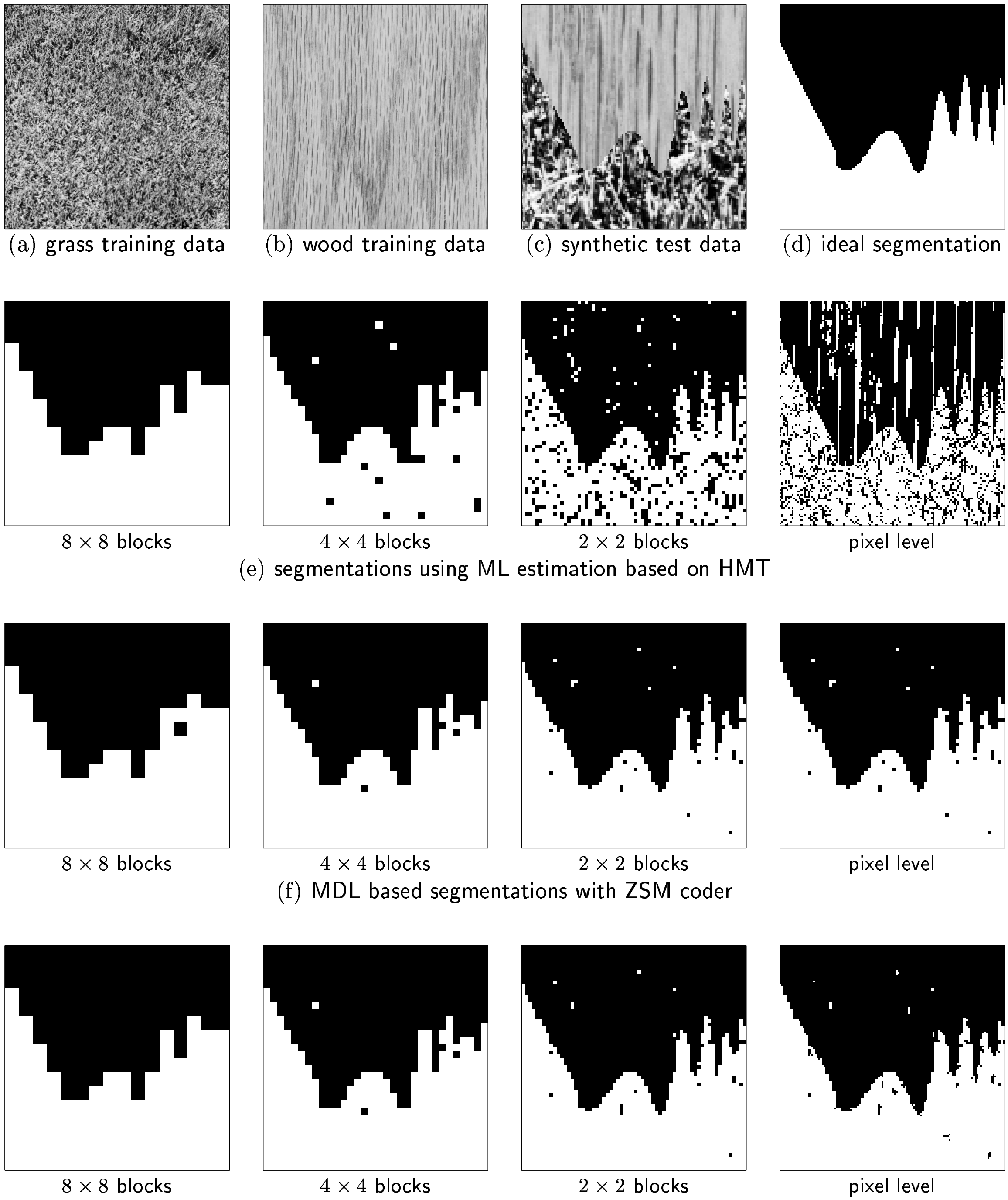


Figure 1. *Synthetic image segmentation. (a)* 512×512 *grass texture image.*[20] *(b)* 512×512 *wood texture image.*[20] *(c)* 128×128 *grass/wood mosaic test image* $\widetilde{x}$ *to be segmented. (d) The segmentation map used to synthesize the test data. (e) Maximum likelihood (ML) segmentation map at a resolution of* 8×8, 4×4, 2×2, *and pixel-sized dyadic squares. Classification accuracy increases with block size (towards coarser scales) because more statistical information is available for the class label decision. This, however, comes at a cost of reduced boundary resolution. (f) Optimal MDL estimate when the ZSM coder is used. An efficient dynamic programming algorithm can be used to search the exact minimizer of the MDL criterion (see Section 5). (g) Sub-optimal MDL estimate when the EP coder is used. An iterative algorithm based on dynamic programming for each iteration yields an excellent albeit sub-optimal estimate. The boundaries in the estimate have better resolution than the estimate in (f) due to the properties of the EP coder.*

fathers such as Fisher that "the objective of statistics is to reduce data", and "we must not over-fit data by too complex models".[22]

2.1. MDL principle

Let x be the observed data, and let $\{p_i(x|\theta_i)\}_{i=1}^K$ be K competing probabilistic descriptions of x. Each of the K models is characterized by their respective parameters θ_i. The Shannon code length for describing the data using model i is given by $-\log_2 p_i(x|\theta_i)$.[5] Let $L(\theta_i)$ denote the number of bits required to describe the parameters of the i^{th} model. Then, the optimal model M^{opt} proposed by MDL is

$$M^{opt} := \arg \min_{\theta_i \in \{\theta_i, \ldots, \theta_K\}} [-\log_2 p_i(x|\theta_i) + L(\theta_i)]. \tag{5}$$

If the parameters θ_i are distributed as $p_\theta(\theta_i)$ then

$$L(\theta_i) := -\log_2 p_\theta(\theta_i). \tag{6}$$

For deterministic parameters, Rissanen's framework for specifying the parameters with asymptotic approximations can be used.[23]

MDL has been applied to a wide variety of problems such as regression,[24] density estimation,[21] denoising,[25] and segmentation.[9–11] We now describe the formulation of segmentation as a model selection problem, and the role played by MDL in the solution.

2.2. MDL formulation of segmentation

Leclerc[9] and Keeler[10] proposed the use of MDL to perform segmentation. Kerfoot further analyzed the use of MDL criterion for segmentation.[11] The image x of support S to be segmented is the data to be described under the hypothetical MDL experiment. If $\{\Lambda_1, \Lambda_2, \ldots, \Lambda_{N_c}\}$ are the different possible class labels that any pixel can assume, then the number of different segmentation maps c possible is N_c^S. Each of the N_c^S different maps is a competing model. Let $f(x|c)$ be the likelihood of the data obtained using the model c. Then, according to MDL, the optimal segmentation map is given by

$$M^{opt} := \arg \min_c [-\log_2 f(x|c) + L(c)], \tag{7}$$

where $L(c)$ is the code length required to describe the segmentation map c. For the sake of clarity, we have henceforth assumed that the images contain just two types of class labels, i.e., $N_c = 2$.

At the pixel-level, the estimates of $f(x|c)$ are unreliable. Further, the number of possible models to search from is enormous. Since textures are better characterized at multiple resolutions, likelihoods estimates at different resolutions can provide a good approximation to $f(x|c)$.[16–18] The reliability of the likelihoods can be exploited to design robust algorithms that eliminate a majority of the unreasonable models.

2.3. Multiscale segmentation using MDL

We first formulate the problem of segmentation in the multiscale framework using MDL. Again, the observed image is the data to be described by the hypothetical MDL experiment. However, in contrast to using pixel-level descriptions, the multiresolution coefficients of the image are described. Each competing model in multiscale MDL segmentation is one particular arrangement of class labels at all resolutions. In contrast to the formulation described in Section 2.2, each model is a hierarchical sequence of binary images, not just a single binary image. Multiscale segmentation using MDL aims to find the arrangement of the class labels at all resolutions such the total of the bits required code the model and the multiresolution coefficients under this model is minimized. Thus,

$$\overline{M}^{opt} := \arg \min_{\overline{c}} [-\log_2 f(x|\overline{c}) + L(\overline{c})], \tag{8}$$

where $\overline{M}^{opt}$ is the optimal model, $\overline{c}$ is any arrangement of class labels at all resolutions, $f(x|\overline{c})$ denote the multiscale likelihoods, and $L(\overline{c})$ denotes the number of bits required to describe the model $\overline{c}$.

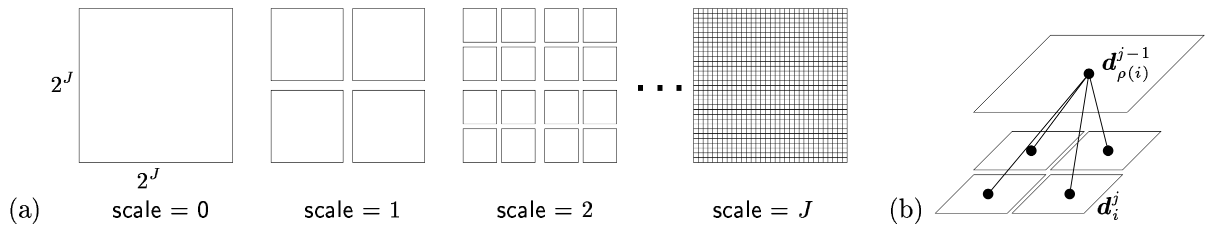


Figure 2. *(a) Image x divided into dyadic squares d_i^j, where j denotes the scale, and i denotes one of the 2^{2j} dyadic squares within the scale respectively. Each dyadic square can be associated with a subtree of Haar wavelet coefficients. (b) Quad-tree structure of dyadic squares. The dyadic square $d_{\rho(i)}^{j-1}$ splits into four child squares at scale j.*

2.3.1. Coding the data ≡ texture characterization

Optimizing over the MDL criterion to obtain $\overline{M}^{opt}$ (see (8)) requires knowledge of the likelihoods $f(x|\overline{c})$. However, these are never known in practice and must be estimated. *Texture characterization*[16–18] provide an estimate of the likelihood $f(x|\overline{c})$ for different $\overline{c}$. We outline the texture characterization using the HMT model of Crouse et al[16] in Section 3.

2.3.2. Coding the model ≡ shape coding

For a two-class segmentation problem, describing the models $\overline{c}$ involves describing the binary symbols in the model. Coding a binary image is equivalent to coding the shape of its boundaries. Hence, model specification can be viewed either as a shape coding or a binary image coding problem.[26,27] Thus, $L(\overline{c})$ in (7) is the number of bits taken by some fixed image coder to compress the segmentation map. In Section 4, we describe two shape coding algorithms that can describe the model.

3. TEXTURE CHARACTERIZATION

Textures are characterized by consistent stochastic behavior of the pixels within their region of support. Using the notations set out in the introduction, for a region R_i occupied by any texture $\lambda \in \{\Lambda_1, \Lambda_2, \ldots, \Lambda_{N_c}\}$, the respective likelihood $f(x_{R_i}|\lambda)$ dictates the probabilistic distribution of the pixels in R_i. The behavior of pixels from different regions is assumed to be independent of each other. Thus, we can write

$$f(x|c) = \prod_{i=1}^{k} f(x_{R_i}|c). \tag{9}$$

The above information is crucial to invoke statistical techniques in applications such as segmentation.

In multiscale segmentation using MDL, we desire that our texture models be defined for all possible shapes (all subsets of S). Of course, the simplest way to do this to make each pixel statistically independent given the class map. However, this approach does not adequately capture the statistical properties of texture. Textures contains many *singularities* such as edges and ridges. Since the wavelet transform can be interpreted as a multiscale edge detector that represents the singularity content of an image at multiple scales, the wavelet domain provides the ideal platform to model textures. The hidden Markov tree model proposed by Crouse et al[16] is a wavelet-domain parametric model that can be defined on an arbitrary shape and captures the rich, multiscale properties of texture.

3.1. Wavelets

The wavelet transform captures the singularity content of an image at multiple scales and three different orientations (see Figures 2 and 3). Wavelets overlying a singularity such as an edge yield large wavelet coefficients; wavelets overlying a smooth region yield small coefficients. In combination, the multiscale singularity detection property and tree structure imply that image singularities manifest themselves as cascades of large wavelet coefficients through scale along the branches of the quad-tree.[28] Conversely, smooth regions lead to cascades of small coefficients.

The wavelet transform represents the texture at a nested set of scales[28] $j = 0, \ldots, J$. Each wavelet coefficient captures information about a $M2^{-j} \times M2^{-j}$ dyadic block in the image, where M is the side-length of the image support. Four "child" wavelet coefficients at a given scale nest inside one "parent" at the next coarser scale, giving

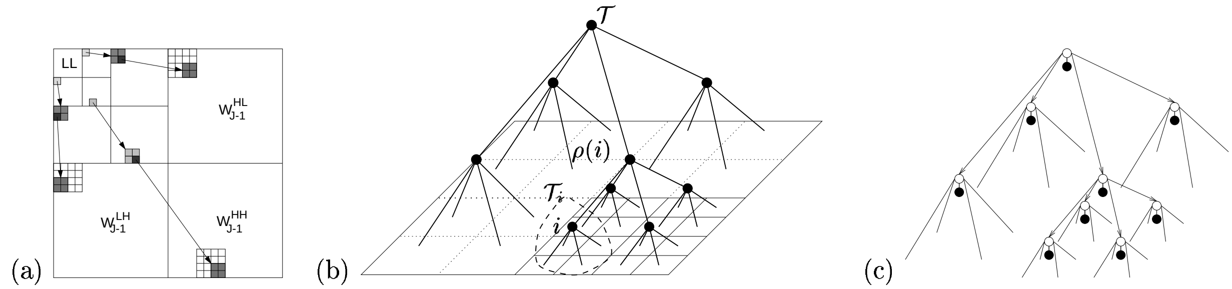


Figure 3. *(a) Parent-child dependencies of the three 2-D wavelet transform subbands: Each arrow points from a parent wavelet coefficient to its four children at the next finer scale. (b) More detailed view of the quad-tree structure in one subband. Each black node corresponds to a wavelet coefficient. The figure also illustrates our tree indexing notation: $\mathcal{T}_i$ is the subtree of coefficients rooted at node i, and $\rho(i)$ is the parent of node i. (c) 2-D wavelet hidden Markov tree (HMT) model. We model each wavelet coefficient (black node) as a Gaussian mixture controlled by a hidden state variable (white node). To capture the persistence across scale property of wavelet transforms, the states are connected vertically across scale in Markov-1 chains.*[16]

rise to a quad-tree structure of wavelet coefficients similar to the structure of the dyadic squares (see Figures 2 and 3). In particular, with the *Haar wavelet transform*, each wavelet coefficient node in the wavelet quad-tree corresponds to a wavelet supported exactly on the corresponding dyadic image square.

3.2. Hidden Markov tree texture model

Crouse et al[16] have developed the *hidden Markov tree* (HMT) model, a parametric statistical model for wavelet transforms that can be used to characterize textures. The HMT captures the fact that large and small wavelet coefficients cascade through scale by using states are connected in a Markovian probabilistic quad-tree similar to that of the wavelet coefficients. The output provided by the HMT is a model $\mathcal{M}$ that provides the approximate joint pdf of the wavelet coefficients of the texture. Further, the HMT allows the likelihood calculation of texture regions inside dyadic blocks of arbitrary size, which can be used to represent regions of arbitrary shape.

4. SHAPE CHARACTERIZATION AND CODING

Shape is a concept intuitively well understood yet difficult to precisely define.[29] The shape of an object in an image can be described by its boundary, which is a curve in two dimensions, or the arrangement of pixels within the boundary. The concept of shape is used in applications such as object recognition, compression, etc. For typical gray scale images, shape description is typically preceded by some kind of edge detection. However, for binary images, the description becomes simpler. Further, coding the boundaries of the object is equivalent to coding the binary image.[30]

In multiscale segmentation using MDL, model description requires the specification of the arrangement of class labels. For the two-class segmentation case, this arrangement of class labels is equivalent to specifying a sequence of binary images of different resolutions. The choice of binary coder can significantly influence the segmentation estimate. Hence, the binary image coder should be chosen according to the application. If the chosen coder efficiently codes typical shapes encountered in the application, then these favored shapes are well-segmented in final estimate. We present two algorithms that can be used to describe hierarchical binary images containing large homogeneous regions, a feature that characterizes typical segmentation maps.

4.1. Zero-tree significance map (ZSM) coder

The past decade has witnessed a significant improvement over traditional image compression techniques such as JPEG thanks to wavelets and Shapiro's zero-tree coding algorithm.[31] Most state-of-the-art image compression algorithms[32–34] first transform an image to obtain the wavelet coefficients, and spend bits only on the large coefficients to achieve compression. However, the decoding of the image needs not only the value of the large wavelet coefficients, but also the locations of these coefficients.

The zero-tree algorithm uses significance maps to denote the locations of the large and small wavelet coefficients. The significance map comprises of a quad-tree of binary symbols, each corresponding to a wavelet coefficient. Zeros in the binary maps denote that corresponding wavelet coefficient is "insignificant", i.e., small.

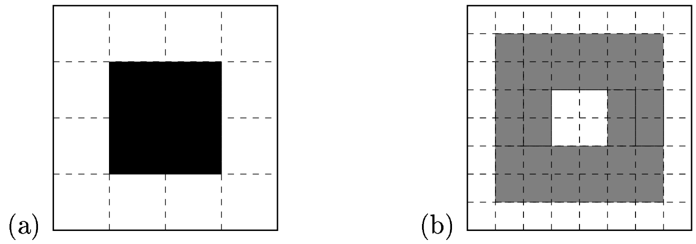


Figure 4. *(a) An 4×4 binary image of parent pixels at some arbitrary scale. (b) An 8×8 binary image of the child pixels that directly lies below the parents pixels in a quad-tree. The shaded pixels are called edge children, while all the other pixels are called homogeneous children.*

The significance map is encoded using the following two rules:

1. If a parent coefficient is insignificant, i.e., zero, then all its four corresponding children are also assumed to be insignificant, i.e., zero.
2. If any of the child coefficient differs from a parent coefficient, then its location is specified.

Given the parent binary map, the cost of encoding the binary map for the children is summarized by the following two cases.

1. **Child = parent**: No cost in incurred, since this is the default assumption made by the decoder.
2. **Child ≠ parent**: The decoder requires the location of this child. This costs $\log_2 N_L$ bits, where N_L denotes the number of coefficients at that resolution in the quad-tree.

We term such a coder the zero-tree significance map (ZSM) coder. The ZSM coder is well-suited for coding binary images with large homogeneous regions, and is particularly tailored for representing binary images containing shapes comprising dyadic blocks. In preceding papers, Cohen et al[35] exploited similar ideas to perform binary image coding in a hierarchical fashion.

4.2. Edge-persistent (EP) coder

In a hierarchical binary image description, a shape refines at a fine scale by invoking transitions around the edges of the coarser scale. The ZSM coder described above uniformly penalizes all children that do not agree with their parents. A uniform penalty is justified when the probability of transition from a zero/one parent to a one/zero child is the same everywhere. However, this assumption does not hold around the edges. Hence, we propose a simple modification to exploit this phenomenon using the EP coder. This coder is similar to multiresolution contour coding proposed by Lerman et al.[36]

Before describing this algorithm, we refer the reader to Figure 4 where we introduce a few simple definitions. We can now describe the binary coding algorithm using the following rules.

1. *Homogeneous* children are assumed to have the same binary value as the parent on the quad-tree.
2. All *edge* children are transmitted to the decoder with code-length determined by the probability of transition around the edges for the image. The probability of transition is assumed to be uniform along the edges and needs to be estimated or predetermined from training images. Since binary image coding is not the primary focus of this paper, we assume an ad hoc value for the transition probability in the paper.

Given the parent binary map, the cost of encoding the child binary map is summarized by the following cases:

1. **Homogeneous child = parent**: No cost in incurred, since this is the default assumption made by the decoder.
2. **Homogeneous child ≠ parent**: The decoder requires the location of this child. This costs $\log_2 N_L$ bits, where N_L denotes the number of pixels in the binary image containing the child.

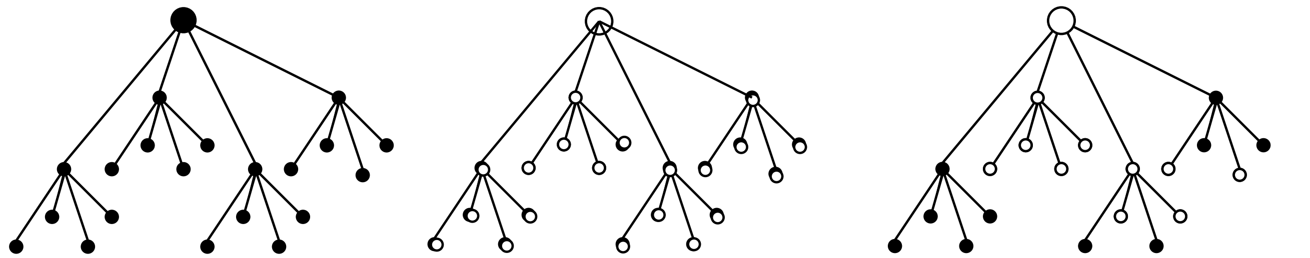


Figure 5. *Graphical formulation of multiscale segmentation. (a) The likelihoods for different resolution dyadic blocks d_i^j of the image assuming Λ_0 texture can be arranged as a quad-tree whose corresponding nodes are weighted by $-\log_2 f(d_i^j|\Lambda_0)$. These likelihoods are provided by texture characterization techniques such as HMT (see Section 3). (b) Quad-tree with nodes weighted by $-\log_2 f(d_i^j|\Lambda_1)$. (c) A collection of segmentation maps at different resolutions can be arranged in the form of a quad-tree. Black or white nodes denotes that the corresponding dyadic square in the collection of different resolution segmentation maps is labeled as Λ_0 or Λ_1 respectively.*

3. **Edge child**: The cost in bits is determined by the transition probability from parent to child around the edges.

Like the ZSM coder, the EP coder is also well-suited to represent binary images with large homogeneous regions and, in particular, to represent binary images containing shapes comprising dyadic blocks. Further, it also allows for refinement along the boundaries.

5. TREE-BASED JOINT TEXTURE AND SHAPE ANALYSIS

We are now in a position to design algorithms to estimate the segmentation map at multiple scales. We first formulate the optimization required in multiscale segmentation graphically. A collection of segmentation maps at different resolutions can be arranged in the form of a quad-tree (see Figure 5(c)). For eg., if dyadic block d_i^j at scale j is labeled as Λ_0, then so is the corresponding node on the quad-tree representing the multiscale segmentation map. The aim of multiscale MDL segmentation is to construct a quad-tree that optimizes the MDL criterion. The likelihoods obtained from characterization techniques such as the HMT[16] can be used to construct two identical quad-trees whose nodes are weighted by the code lengths $-\log_2 f(d_i^j|\Lambda_0)$ and $-\log_2 f(d_i^j|\Lambda_1)$ respectively (see Figures 5(a) and (b)). The shape coder used by MDL dictates the cost to make a choice between a Λ_0 or Λ_1 node. Multiscale MDL segmentation aims to construct a quad-tree such that the sum of the total node weights in the quad-tree, and the cost of choosing the nodes is minimum.

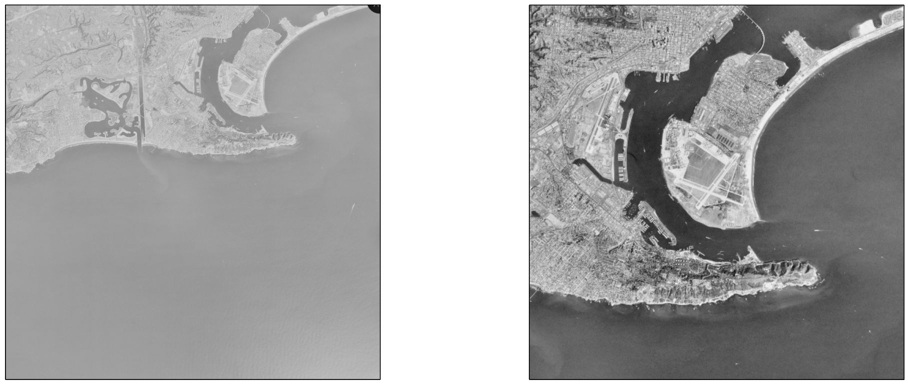
5.1. Optimal segmentation with ZSM coder

In the ZSM coder, the code length of any child node depends only on the immediate parent in the quad-tree. Graphically, this implies that the cost of choosing the node class depends only on the choice made at the parent in the coarser scale. In such a case, a dynamic programming[19] algorithm comprising of a simple and efficient up-sweep followed by a down-sweep on a quad-tree yields the segmentation map that exactly minimizes the MDL criterion. The complexity of such an algorithm is just $O(M)$, where M is the number of pixels in the image.

5.2. Sub-optimal iterative segmentation with arbitrary shape coders

For coders such as EP, the dynamic programming approach proves impractical. In an EP coder, the code length of any child node also depends on the choices made at the neighbors of the immediate parent node in the quad-tree. Though dynamic programming can estimate the exact minimizer with an up-down sweep of the quad-tree, the number of states to be stored during the up-sweep explodes exponentially with the depth of the quad-tree making the algorithm infeasible. For other hierarchical coders such as JBIG,[26] the cost of coding a child depends not only on the parents neighborhood but also on its own neighborhood. To be able to incorporate any arbitrary shape coder in the MDL framework, we devise a efficient albeit sub-optimal iterative algorithm that can accommodate any arbitrary coder using dynamic programming for each iteration:

1. **Step 1**: Make a robust initial estimate using the ZSM segmentation algorithm.



(a) training data (b) test data

Figure 6. *Original aerial photos to be segmented. (a) 1024 × 1024 aerial photo.*[20] *(b) 512 × 512 test sub-image x. The homogeneous ground/sea regions outside the test sub-image were used to train two HMTs.*

2. **Step 2**: Assume that the multiscale segmentation map obtained in the previous iteration is exact. Calculate the costs for a node to have the same or different class label as the parent node. Do this for each node in the quad-tree using the chosen coder.

3. **Step 3**: With the costs of choosing between the nodes now available, use dynamic programming to find class labels for the minimizer tree. Repeat step 2.

Though we have not yet rigorously analyzed the convergence properties of this algorithm, in practice, the algorithm seems to capture the perspective of the image coder with just a few iterations.

6. RESULTS

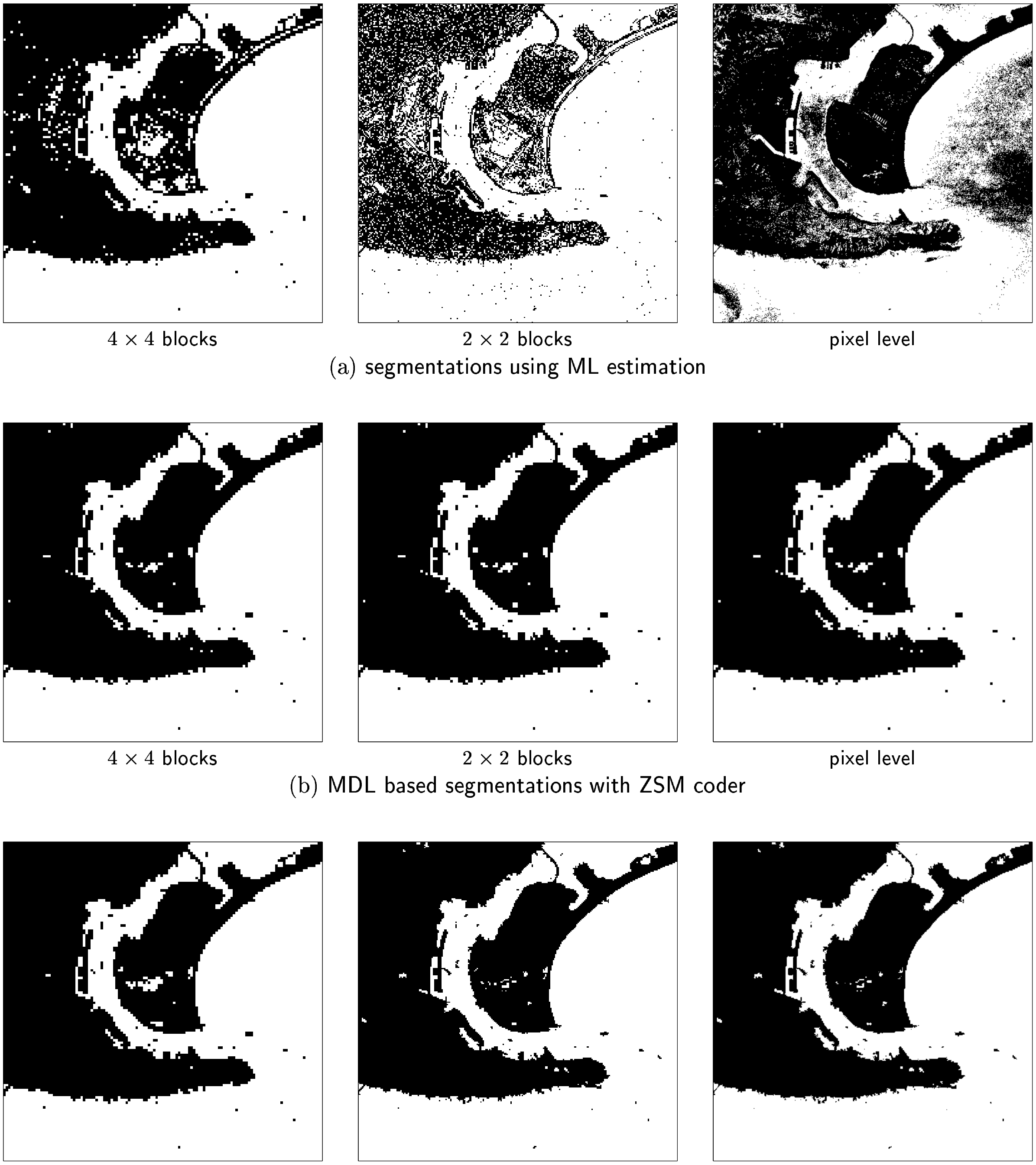
We tested the proposed segmentation algorithms using a real world aerial photograph example.[20] The original is an 1024 × 1024 image with "ground" and "sea" regions as shown in Figure 6(a). Figure 6(b) shows an 512 × 512 sub-block of the full image that is to be segmented. Wavelet HMTs, trained on hand-segmented blocks from the 1024×1024 aerial photo in Figure 6(a) provide the texture likelihoods for the image. The pixel-level likelihoods are obtained by using the pixel brightness of ground and sea textures.

Figure 7(a) show the segmentations at the finest three scales obtained by using ML at each scale. Figure 7(b) shows the segmentations that exactly minimizes of the MDL criterion when a ZSM coder is assumed. The segmentations are obtained an efficient dynamic programming algorithm. Figure 7(c) using the segmentation estimate obtained afters three iterations of the iterative segmentation algorithm when the EP coder is assumed. Though the estimate is not guaranteed to exactly minimize the corresponding MDL criterion, the segmentation results are more desirable with better resolution along the boundaries, since the EP coder captures the characteristics of the ideal segmentation maps better.

7. CONCLUSIONS

In this paper, we have proposed a general framework to simultaneously incorporate texture and shape information into segmentation.

We have formulated the problem of multiscale segmentation as a model selection problem and used MDL to estimate the segmentation map at multiple scales. Thanks to the formulation, we realize that segmentation can be decomposed into texture characterization and shape coding. With the realization that using different shape coding schemes tunes the segmentation to different applications, we can now turn to the mature fields of binary image and shape coding for ideas to design application-specific segmentation algorithms. We also realize that the current multiscale MAP segmentation algorithms attain sub-optimal estimates under the MDL criterion.



4 × 4 blocks | 2 × 2 blocks | pixel level

(a) segmentations using ML estimation

4 × 4 blocks | 2 × 2 blocks | pixel level

(b) MDL based segmentations with ZSM coder

4 × 4 blocks | 2 × 2 blocks | pixel level

(c) MDL based segmentations with EP coder

Figure 7. *Aerial photo segmentation. (a) Maximum likelihood (ML) segmentation of Figure 6(b) at resolutions of* 4×4, 2×2, *and pixel-sized dyadic squares. All the likelihoods were obtained using trained HMTs. (b) MDL segmentation estimate with ZSM coder. An efficient dynamic programming algorithm can be used to search the exact minimizer of the MDL criterion (see Section 5). The segmentation estimates fail to refine with finer scales, since the cost incurred by the ZSM coder to encode the required transitions is very high. (c) MDL segmentation estimate with EP coder. The iterative algorithm based on dynamic programming for each iteration yields an excellent albeit sub-optimal estimate. The pixel-level segmentation estimate from Figure 7(b) is used as the initial estimate over which the iterations act. The boundaries in the estimate have better resolution than the estimate in Figure 7(b) due to the properties of the EP coder.*

We designed a tree-based algorithm that searches for the optimal MDL segmentation map when the ZSM shape coding technique is assumed. The algorithm attains the solution with an efficient up-down sweep of a quad-tree. Realizing that the exact MDL solution is not practically attainable for an arbitrary coder, we designed an iterative segmentation procedure that can potentially accommodate any coder, so that the resulting segmentation can reflect the essence of the coder. We tested this algorithm with the proposed edge-persistent coding scheme to obtain desirable results on both synthetic and real world data.

We are currently studying the convergence and exact computational complexity of the proposed iterative segmentation algorithm. We are also testing the impact of existing shape coding techniques on the final segmentation results.

ACKNOWLEDGMENTS

We would like to thank Mike Orchard for sharing his philosophies with us. We would also like to thank Sanjeeb Dash for helping us design the dynamic programming algorithm.

REFERENCES

1. R. Haralick and L. Shapiro, "Image segmentation techniques," *Comput. Vision, Graphics, Image Proc.* **29**, pp. 100–132, 1985.
2. C. Therrien, "An estimation–theoretic approach to terrain image segmentation," *Comput. Vision, Graphics, Image Proc.* **22**, pp. 313–326, 1983.
3. H. Derin and H. Elliot, "Modeling and segmentation of noisy and textured images using Gibbs random fields," *IEEE Trans. on Pattern Anal. and Mach. Intell.* **9**, pp. 39–55, Jan. 1987.
4. C. Bouman and M. Shapiro, "A multiscale random field model for Bayesian image segmentation," *IEEE Trans. Image Processing* **3**, pp. 162–177, March 1994.
5. T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons, Inc., New York, 1991.
6. J. Rissanen, "Modeling by shortest data description," *Automatica* **14**, pp. 465–471, 1978.
7. J. Rissanen, *Stochastic Complexity in Stastistical Inquiry*, World Scientific, Singapore, 1989.
8. J. Rissanen, "Stochastic complexity in learning," in *Computational Learning Theory: Second European Conference(EuroCOLT'95)*, P. Vitanyi, ed., pp. 196–210, Springer, Berlin,, 1995.
9. Y. Leclerc, "Construction simple stable descriptions for image partitioning," *International Journal on Computer Vision* **3**, pp. 72–102, Mar. 1989.
10. K. Keeler, "Minimum–length encoding of planar subdivision topologies with application to image segmentation," in *The theory and application of minimal–length encoding workshop notes*, vol. Stanford university, pp. 95–99, American Association for Artificial Intelligence, Mar. 1990.
11. I. B. Kerfoot, *Information–theoretic segmentation criteria for vector fields: Analysis and design.* Masters dissertation, University of Illinois–Champaign, 1992.
12. H. Choi and R. Baraniuk, "Multiscale document segmentation using wavelet-domain hidden Markov models," in *Proc. of IST/SPIE's 12th Annual Symp. - Electronic Imaging 2000*, (San Jose, CA), Jan. 2000.
13. H. Choi and R. Baraniuk, "Interpolation and denoising of nonuniformly sampled data using wavelet domain processing," in *Proc. IEEE Int. Conf. on Acoust., Speech, Signal Proc. — ICASSP '99*, (Phoenix, AZ), March 1999.
14. H. Cheng and C. A. Bouman, "Trainable context model for multiscale segmentation," in *Proc. IEEE Int. Conf. on Image Proc. — ICIP '98*, (Chicago, IL), Oct. 4–7 1998.
15. H. Cheng, C. A. Bouman, and J. P. Allebach, "Multiscale document segmentation," in *Proc. IS&T 50th Annual Conference*, pp. 417–425, (Cambridge, MA), May 18–23 1997.
16. M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, "Wavelet-based statistical signal processing using hidden Markov models," *IEEE Trans. Signal Proc.* **46**, pp. 886–902, April 1998.
17. C. Bouman and B. Liu, "Multiple resolution segmentation of textured images," *IEEE Trans. on Pattern Anal. and Mach. Intell.* **13**, pp. 99–113, Feb. 1991.
18. B. Manjunath, T. Simochony, and R. Chellappa, "Stochastic and deterministic networks for texture segmentation," *IEEE Trans. Acoust., Speech, Signal Processing* **38**(6), pp. 1039–1049, 1990.
19. R. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, New Jersey, 1957.

20. "The USC-SIPI image database." `sipi.usc.edu/services.html`.
21. A. Barron, J. Rissanen, and B. Yu, "The minimum description length principle in coding and modeling," *IEEE Trans. Inform. Theory* **44**, pp. 2743–2760, Oct. 1998.
22. R. A. Fisher, "On the mathematical foundations of theoretical statistics," *Phil. Trans. Roy. Soc. London, Ser. A* **222**, pp. 309–368, 1922.
23. J. Rissanen, "A universal prior for integers and estimation by minimum description length," *Annals of Statistics* **11**, pp. 416–431, 1983.
24. J. Rissanen, "Universal coding, information, prediction, and estimation," *IEEE Trans. Inform. Theory* **30**, pp. 629–636, Jul 1984.
25. J. Liu and P. Moulin, "Complexity-regularized image denoising," *Preprint* , 2000.
26. P. Howard et. al., "The emerging JBIG2 standard," *IEEE Trans. Circuits Syst.* **8**, Sep. 1998.
27. B. Martins and S. Forchhammer, "Tree coding of bilevel images," *IEEE Trans. Image Processing* **7**, pp. 517–612, May 1998.
28. S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, San Diego, 1998.
29. S. Marshall, "Review of shape coding techniques," *Image and Vision Computation* **7**, pp. 281–294, Nov. 1989.
30. J. R. R. Estes and V. R. Algazi, "Efficient error free chain coding of binary documents," in *Data Compression Conference '95: Data Compression*, J. A. Storer and M. Coh, eds., pp. 122–131, (Snowbird, Utah), Mar. 1995.
31. J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing* **41**(12), pp. 3445–3462, 1993.
32. A. Said and W. A. Pearlman, "An image multiresolution representation for lossless and lossy image compression," *IEEE Trans. Image Processing* **5**(9), pp. 1303–1310, 1996.
33. Z. Xiong, K. Ramchandran, and M. Orchard, "Space–frequency quantization for wavelet image coding," *IEEE Trans. Image Processing* **6**, pp. 677–693, May 1997.
34. S. Servetto, K. Ramchandran, and M. Orchard, "Image coding based on a morphological representation of wavelet data," *IEEE Trans. Image Processing* **8**(9), pp. 1161–1174, 1999.
35. Y. Cohen, M. Landy, and M. Pavel, "Hierarchical coding of binary images," *IEEE Trans. on Pattern Anal. and Mach. Intell.* **7**, pp. 284–298, May 1985.
36. J. Lerman, S. Kulkarni, and J. Koplowitz, "Multiresolution chain coding of contours," *Proc. IEEE Int. Conf. Image Processing — ICIP '94* **2**, pp. 615–619, 1994.