Multiscale Document Segmentation[1]

Hui Cheng, Charles A. Bouman, and Jan P. Allebach
School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47907-1285

Abstract

In this paper, we propose a new approach to document segmentation which exploits both local texture characteristics and image structure to segment scanned documents into regions such as text, background, headings and images. Our method is based on the use of a multiscale Bayesian framework. This framework is chosen because it allows accurate modeling of both the image characteristics and contextual structure of each region. The parameters which describe the characteristics of typical images are extracted from a database of training images which are produced by scanning typical documents and hand segmenting them into the desired components. This training procedure is based on the expectation maximization (EM) algorithm and results in approximate maximum likelihood (ML) estimates of the model parameters for region textures and contextual structure at various resolutions. Once the training procedure is performed, scanned documents may be segmented using a fine-to-coarse-to-fine procedure that is computationally efficient.

1. Introduction

With the advent of modern publishing technologies, the layout of today's documents has never been more complex. Most of them contain not only text and background regions, but also graphics, tables and images. Therefore scanned documents must often be segmented before other document processing techniques, such as compression or rendering, can be applied.

Traditional approaches to document segmentation, usually referred as top-down methods [7, 10], involve partitioning the document images into blocks and then classifying each block [5, 13, 16, 17]. This kind of approach was first proposed by Wong, Casey and Wahl in 1982 [17]. They applied a technique called the run length smoothing algorithm (RLSA) to partitioning a binary document image into blocks. Each block was then classified as text or image according to some statistical features, such as the total number of black pixels, and the horizontal white-black transitions of data. A similar algorithm was also investigated by Wang et al. for newspaper layout analysis [16]. In 1993, Chauvet and coworkers [5] presented a recursive block algorithm based on RLSA. They introduced the linear closing with variable length structuring elements to extract features for block classification. Another approach reported by Krishnamoorthy et al. [13] used a 2-D X-Y tree structure to represent the page layout. A more detailed survey of these approaches can be found in [9].

Traditional approaches work well in a pre-specified environment with simple layout, such as postal addresses, business correspondences and technical papers. Components of these documents are assumed to be rectangular in shape with relatively uniform font and size. For example, the RLSA assumes that the average number of white pixels between characters and between lines are known beforehand. However, the performance of such approaches degrades significantly when different components are touching or overlapping. So the robustness of these algorithms is of concern. Also, these approaches are sensitive to skew, so a good skew correction algorithm is needed in the preprocessing stage.

Alternatively, texture based approaches [7, 10, 11] treat different components of a document image, such as text, background, images or headings, as different textures. The scanned document images are first convolved with a set of masks to generate feature vectors. Each feature vector is then classified into different classes using a pre-trained classifier, such as a neural network [7, 11]. The final step is post-processing. For example, the post-processing in [11] includes smoothing the labeled image, merging nearby regions, removing small components and separating text from line drawings. Generally, texture based approaches are more robust than traditional approaches since they are less likely to make large scale errors when the input document image does not completely satisfy the underlying assumptions.

One problem associated with texture based approaches is the mask size used for extracting local features. When the mask size is small, it is difficult to detect large scale textures such as large fonts. On the contrary, if a large mask is chosen, the computation will increase dramatically, and the number of parameters required to accurately

[1] This project was supported by the Xerox Corporation.

[2] Appeared in IS&T 50th Annual Conference, pp. 417-425, Cambridge, MA, May 18-23, 1997.

modeled the texture becomes excessive.

In this paper, we propose a new approach to the document segmentation which is based on a multiscale Bayesian framework [4]. This approach exploits both local texture characteristics and image structure to segment the scanned documents into different regions such as text, background, headings and images. With a multiscale image model, the local texture characteristics are extracted at each resolution via a wavelet decomposition. The desired image structure is represented by a multiscale context model. Using this context model, correct image structures can be enforced by penalizing segmentations containing incorrect image structures. Once a complete model is formulated, a document image can be segmented using the sequential maximum a posteriori (SMAP) estimator [4].

The parameters needed for both the image model and the context model are estimated from a database of training images which are produced by scanning typical documents and hand segmenting them into desired components. The training procedure is based on the expectation maximization (EM) algorithm and results in approximate maximum likelihood (ML) estimation of the model parameters for region textures and contextual structure at each resolution.

Since both local texture characteristics and contextual information are extracted at various resolutions, our approach can model and classify a large number of classes which are difficult to classify with a single scale segmentation approach. Also, the SMAP estimator only needs one fine-to-coarse-to-fine propagation through resolutions; so, the algorithm is computationally efficient. Finally, all parameters are estimated off-line, further reducing the computation.

This paper is organized as follows. In section 2, we will review the general Bayesian segmentation scheme and introduce our multiscale document segmentation model. Section 3 will present the segmentation algorithm. The problem of feature extraction and parameter estimation will be discussed in section 4. Section 5 presents the experimental results, and the conclusion of this study is in section 6.

2. Multiscale Document Segmentation

The Bayesian approach to segmentation is based on a doubly stochastic model as shown in Fig. 1. The scanned document image is modeled as a random field Y, and X represents the unknown segmentation. The dependence between the observed image and its segmentation is modeled by the conditional distribution of Y given X, $p_{y|x}(y|x)$. The prior knowledge about the size and shapes of regions is incorporated in the prior distribution $p(x)$. Using Bayes rule, the posterior distribution $p_{x|y}(x|y)$ can be calculated. Then the image can be segmented by calculating an estimate of X given Y.

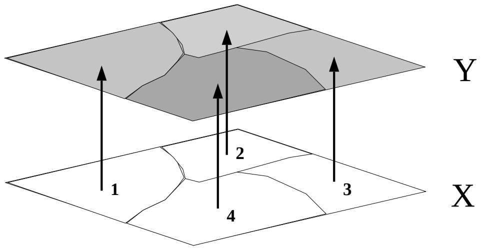


Figure 1: *The Bayesian segmentation model. Y is an image containing different regions. X is the unobserved field containing the class of each pixel. The behavior of Y given X is defined as a conditional probability distribution.*

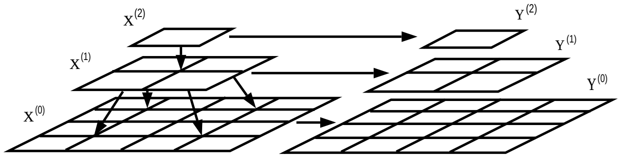


Figure 2: *The multiscale document segmentation model. The right pyramid is the image model, and the left pyramid is the context model. The arrows indicates the direction of dependency.*

2.1. Multiscale Document Segmentation Model

The multiscale document segmentation model (MSDM) is a double pyramidal structure (see Fig. 2). Both pyramids have the same lattice structure $S^{(n)}$ $(n = 0, 1, \ldots, L)$. $S^{(0)}$ is the lattice at the finest scale with each point $s^{(0)} \in S^{(0)}$ corresponding to a single image pixel. Each pixel $s^{(n)}$ at scale n corresponds to a group of four pixels $s_i^{(n-1)}$ $(i = 1, 2, 3, 4)$ at the finer scale $n-1$, where $s_i^{(n-1)}$ are the children of $s^{(n)}$. Therefore the number of pixels in $S^{(n)}$ is 1/4 the number of pixels in $S^{(n-1)}$. The right pyramid of the MSDM is the multiscale image model. It consists of several random fields. Each random field, denoted as $Y^{(n)}$ $(n = 0, 1, \ldots, L)$, models the scanned document at a certain scale which is specified by the 2-D lattice $S^{(n)}$. For each pixel $s \in S^{(n)}$, Y_s is defined as a random variable. The left pyramid is called the multiscale context model which extracts and enforces the correct context for the segmentation. This context model consists of random fields $X^{(n)}$ which corresponds to the 2-D lattice $S^{(n)}$. For $s \in S^{(n)}$, the corresponding label X_s specifies one of M possible classes of a document.

The MSDM can model and classify a large variety of classes which are interesting in document segmentation. For example, half-tone and continuous-tone images may be distinguished by using fine scale features; while low resolution features can better discriminate classes such as headings and graphics. The multiscale context model works similarly. In order to remove random noise and make a smooth segmentation, we only need fine scale contextual information, such as the majority of labels in a 3×3 neighborhood at the finest resolution. But to classify graphics from text, very coarse scale contextual information may be useful.

For the convenience of later discussion, we define

$$X^{(\leq n)} = \bigcup_{i \leq n} X^{(i)}$$

In the same way, we can define $X^{(>n)}$. These definitions for X can be also applied to Y and S. We also use X, Y to denote $X^{(\leq L)}$ and $Y^{(\leq L)}$, respectively.

The fundamental assumption of the context model is that the sequence of random fields $X^{(n)}$ from coarse to fine forms a Markov chain. This can be formally expressed as

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

where $p_{x^{(n)}|x^{(n+1)}}(x^{(n)}|x^{(n+1)})$ is called the transition probability.

For the image model, we assume that

1. $Y^{(n)}$ are conditionally independent given X.

$$P(Y \in dy|X) = \prod_{n=0}^{L} P(Y^{(n)} \in dy^{(n)}|X) \tag{2}$$

2. $Y^{(n)}$ is exclusively dependent on $X^{(n)}$.

$$\begin{aligned} & P(Y^{(n)} \in dy^{(n)}|X = x) \\ = \; & P(Y^{(n)} \in dy^{(n)}|X^{(n)} = x^{(n)}) \\ \equiv \; & p_{y^{(n)}|x^{(n)}}(y^{(n)}|x^{(n)}) \end{aligned} \tag{3}$$

From (1), (2) and (3), we get

$$\begin{aligned} P(Y \in dy, X = x) = \prod_{n=0}^{L} \big\{ & p_{y^{(n)}|x^{(n)}}(y^{(n)}|x^{(n)}) \\ & p_{x^{(n)}|x^{(n+1)}}(x^{(n)}|x^{(n+1)}) \big\} \end{aligned} \tag{4}$$

2.2. Sequential MAP Estimation

Using the MSDM, the segmentation problem is transferred into an optimization problem which is to minimize the average cost of an erroneous segmentation.

$$\hat{x} = \arg \min_{x} E\big[C(X, x)|Y = y\big] \tag{5}$$

where $C_n(X, x)$ is the cost of estimating the true segmentation X by the approximate segmentation x.

Because a misclassified pixel at coarse resolution will affect more pixels than a misclassified pixel at fine resolution. The average cost of an erroneous segmentation is defined as

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

where $C_n(\cdot, \cdot)$ is defined as

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

where $\delta(X^{(i)} - x^{(i)}) = 1$ if $X^{(i)} = x^{(i)}$ and $\delta(X^{(i)} - x^{(i)}) = 0$ if $X^{(i)} \neq x^{(i)}$. Using an argument similar to that in [4], it can be shown that the solution to (5) is approximately given by the recursive equations

$$\hat{x}^{(L)} = \arg \max_{x^{(L)}} \log p_{y^{(\leq L)}|x^{(L)}}(y^{(\leq L)}|x^{(L)}) \tag{6}$$

$$\begin{aligned} \hat{x}^{(n)} = \; & \arg \max_{x^{(n)}} \big\{ \log p_{y^{(\leq n)}|x^{(n)}}(y^{(\leq n)}|x^{(n)}) \\ & + \log p_{x^{(n)}|x^{(n+1)}}(x^{(n)}|\hat{x}^{(n+1)}) \big\} \end{aligned} \tag{7}$$

We refer to the solution of (6) and (7) as the SMAP estimator. Notice that (7) consists of two terms, one is related to Y, the data term, and the other is the context term which is solely determined by the transition probability. If we associate correct context with a higher transition probability, and assign incorrect context with a lower transition probability, then the segmentation which does not have a correct context will be penalized by the context term and is unlikely to be selected as the final segmentation.

3. Document Segmentation Algorithm

In section 2, we developed a general model for document segmentation. All specific models will be defined in this section. They include the assumptions for $X^{(n)}$ and $Y^{(n)}$, the probability density $p_{y^{(n)}|x^{(n)}}(y^{(n)}|x^{(n)})$, and the transition densities $p_{x^{(n)}|x^{(n+1)}}(x^{(n)}|x^{(n+1)})$.

3.1. Image Model

For the multiscale image model, we assume that the observed pixels are conditionally independent given their class labels. This assumption leads to a simple product form of the conditional probability $Y^{(n)}$ given $X^{(n)}$.

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

where $p_{y_s^{(n)}|x_s^{(n)}}(\cdot|k)$ is the conditional density function for an individual pixel given its class label k. This conditional

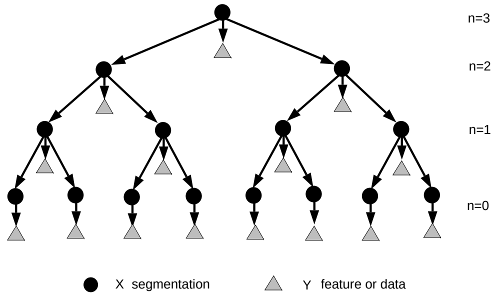


Figure 3: *The 1-D analog of the quadtree model. The triangles represent the image model, and the circles denote the context model. Arrows indicate the direction of dependency among the nodes.*

independence assumption does not ignore the dependency among image pixels. Instead, this dependency is well analyzed using the context model discussed in the following section.

3.2. Context Model

For the context model, we will restrict our choice to have two important properties. First, the labels in $X^{(n)}$ will be conditionally independent, given the labels in $X^{(n+1)}$. This is the Markov chain property discussed in section 2.1. Second, for a pixel s in $S^{(n)}$, where $n < L$, the label $X_s^{(n)}$ will only be dependent on the labels of a neighborhood at the next coarser scale $S^{(n+1)}$. This set of neighboring locations to s is defined as ∂s. Based on these properties, the transition probability mass function from coarse to fine scale must have the form

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

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

The choice of the neighborhood system ∂s will finally determine the context model. We will start from a simple neighborhood system, the quadtree structure. Then discuss a more general structure, the pyramidal graph structure. Because of the advantages and the disadvantages of each model, we will employ a hybrid model which incorporates both the quadtree and the pyramidal graph structure.

3.2.1. Quadtree Model

In the quadtree model, each point s in the pyramid is dependent only on one point at the next coarser scale, which is called the parent of s. The 1-D analog of the quadtree structure is shown in Fig. 3. We define $d_s^{(n)}$ as the set of children of s at scale n. We also define $z^{(\leq n)}(s)$ as the image features which are descendents of s at scales less than or equal to n.

$$z^{(\leq n)}(s) = \bigcup_{i \leq n} \bigcup_{r \in d_s^{(i)}} y_r^{(i)}$$

Because of the quadtree structure and the conditional independence of Y given X, the distribution of Y given X has the following product form

$$\begin{aligned} & p_{y^{(\leq n)}|x^{(n)}}(y^{(\leq n)}|x^{(n)}) \\ = & \prod_{s \in S^{(n)}} p_{z_s^{(\leq n)}|x_s^{(n)}}(z_s^{(\leq n)}|x_s^{(n)}) \qquad (8) \end{aligned}$$

Furthermore, the density function $p_{z_s^{(\leq n)}|x_s^{(n)}}(z_s^{(\leq n)}|x_s^{(n)})$ can be calculated using the following recursion.

$$\begin{aligned} & p_{z_s^{(\leq n+1)}|x_s^{(n+1)}}(z_s^{(\leq n+1)}|k) \\ = & p_{z_s^{(\leq n)}|x_s^{(n+1)}}(z_s^{(\leq n)}|k)\, p_{y_s^{(n+1)}|x_s^{(n+1)}}(y_s^{(n+1)}|k) \\ = & \left\{ \prod_{r \in d_s^{(n)}} \sum_{m=1}^{M} p_{z_r^{(\leq n)}|x_r^{(n)}}(z_r^{(\leq n)}|m) \right. \\ & \left. p_{x_r^{(n)}|x_s^{(n+1)}}(m|k) \right\} p_{y_s^{(n+1)}|x_s^{(n+1)}}(y_s^{(n+1)}|k) \qquad (9) \end{aligned}$$

where M is the number of classes. Dynamic range considerations mandate the logarithm of these functions be computed and stored. Therefore, we define the log likelihood function

$$l_s^{(\leq n)}(k) = \log p_{z_s^{(\leq n)}|x_s^{(n)}}(z_s^{(\leq n)}|k)$$

Then (9) can be rewritten as a recursion

$$\begin{aligned} & l_s^{(\leq n+1)}(k) = \log p_{y_s^{(n)}|x_s^{(n)}}(y_s^{(n)}|k) \\ & + \sum_{r \in d_s^{(n)}} \log \left\{ \sum_{m=1}^{M} \exp \left[l_r^{(\leq n)}(m)\, p_{x_r^{(n)}|x_s^{(n+1)}}(m|k) \right] \right\} \end{aligned}$$

Once the likelihood functions are computed, the SMAP segmentation can be efficiently computed using (6) and (7).

$$\begin{aligned} \hat{x}_s^{(L)} &= \arg \max_{1 \leq k \leq M} l_s^{(\leq L)}(k) \\ \hat{x}_s^{(n)} &= \arg \max_{1 \leq k \leq M} \left\{ l_s^{(\leq n)}(k) \right. \\ & \qquad \left. + \log\, p_{x_s^{(n)}|\hat{x}_{\partial s}^{(n+1}}(k|\hat{x}_{\partial s}^{(n+1)}) \right\} \end{aligned}$$

Although the quadtree model results in an exact expression for computing the SMAP segmentation, it has the obvious disadvantage that the quadtree model does not enforce smooth boundaries in the segmentation. This is due to the fact that spatially adjacent pixels may not have common neighbors at coarser scales.

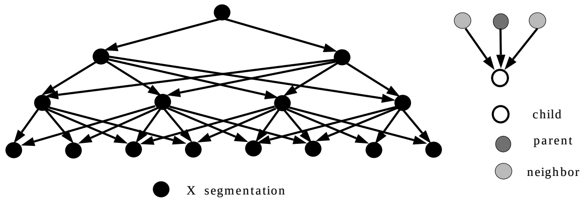


Figure 4: *The 1-D analog of the pyramidal graph model and the 1-D analog of its neighborhood system.*

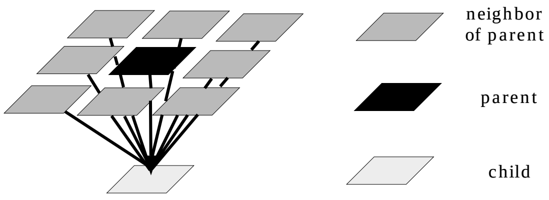


Figure 5: *The neighborhood system used in the pyramidal graph model.*

3.2.2. Pyramidal Graph Model

The weekness of the quadtree model can be overcome by increasing the number of coarse neighbors. The resulting model is the pyramidal graph model, shown in Fig. 4. In the pyramidal graph model, each internal pixel s has 9 coarse neighbors, ∂s (Fig. 5).

Although the pyramidal graph model is more precise than the quadtree model, its likelihood function does not have a product form as in (8) and does not result in a simple fine-to-coarse recursion with the form of (9) for computing $Y^{(\leq n)}$ given the labels $X^{(n)}$.

3.2.3. Hybrid Model

As a compromise between the performance and the computational complexity, we use the hybrid pyramid structure (Fig. 6) in our segmentation algorithm. For the computation at a single scale n, the hybrid model assumes that at all levels above n, points are connected as in the pyramidal graph model, and at levels below or equal to n, points are connected as in the quadtree model. So, the conditional likelihood of (7) has the form

$$\log p_{y^{(\leq n)},x^{(n)}|\hat{x}^{(n+1)}}(y^{(\leq n)}, x^{(n)}|\hat{x}^{(n+1)}) = \sum_{s\in S^{(n)}} \left\{ l_s^{(\leq n)}(x_s^{(n)}) + \log \tilde{p}_{x_s^{(n)}|\hat{x}_{\partial s}^{(n+1)}}(x_s^{(n)}|\hat{x}_{\partial s}^{(n+1)}) \right\}$$

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

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

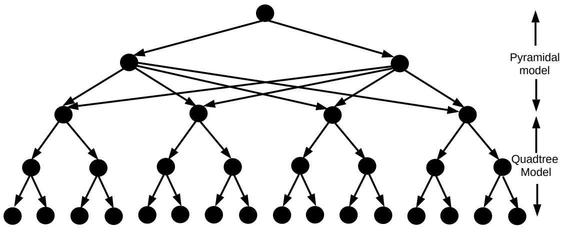


Figure 6: *The 1-D analog of the hybrid model. At levels above n, points are connected as in the pyramidal graph model. At levels below or equal to n, points are connected as in the quadtree model.*

4. Feature Extraction and Parameter Estimation

In this section, we will propose feature extraction and parameter estimation schemes for both the image model and the context model. For document segmentation applications, one usually has sufficient document images for feature extraction and parameter estimation. Therefore, in stead of estimating parameters on-line, an off-line training approach (or supervised training) can be used to alleviate the on-line computation, and improve the accuracy.

For the supervised training, a database of training images is produced by scanning typical documents and hand segmenting them into the desired components.

4.1. Feature Extraction and Parameter Estimation for Image Model

The fundamental assumption of the image model is that $Y^{(n)}$ is conditionally independent given X (2). In order to satisfy this assumption, an orthogonal decomposition of the original document image is desirable. Therefore, we choose the wavelet decomposition [15] to generate the feature vector at each pixel. Also, the wavelet decomposition gives us the desired lattice structure that has been defined in section 2.1. In order to reduce computation, we use the Haar wavelet basis for this purpose. The resulting feature vector at each pixel consists of three elements which correspond to the wavelet coefficients at HH, HL, LH bands, respectively.

The conditional probability distribution of an individual pixel given its class label, $p_{y_s^{(n)}|x_s^{(n)}}(y_s|x_s)$ is modeled as a multivariate Gaussian mixture density,

$$p_{y_s^{(n)}|x_s^{(n)}}(y_s|x_s) = \sum_{k=1}^{K} \gamma_k \, p_{y_s^{(n)}|x_s^{(n)},k}(y_s|x_s, k)$$

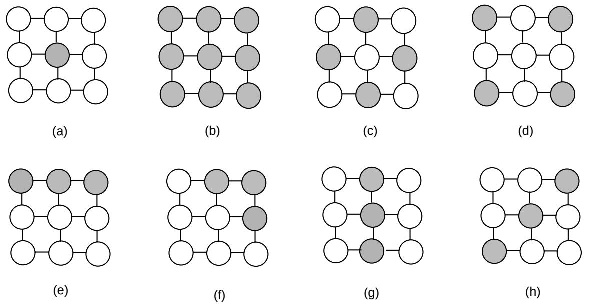


Figure 7: *Context features used in experiments: (a) the label of the parent pixel, (b) the majority, (c) the majority of four nearest neighbors, (d) the majority of four corner neighbors, (e) the majority of a line boundary (one of four line boundaries is shown in this figure), (f) the majority of a corner boundary (one of four corner boundaries is shown in this figure), (g) the majority of a center line (one of two center lines is shown in this figure), (h) the majority of a diagonal center line (one of two diagonal center lines is shown in this figure). When calculating a feature, only shaded circles are considered.*

where $p_{y_s^{(n)}|x_s^{(n)},k}(y_s|x_s,k)$ is a Gaussian density with mean $\mu_{x_s,k}$, and covariance matrix $C_{x_s,k}$. $\gamma_k \in [0,1]$ and $\sum_k \gamma_k = 1$. For large K, the Gaussian mixture density can approximate any probability density and its parameters can be estimated using the expectation maximization (EM) algorithm [1, 6]. The order K is chosen for each class using the Rissanen criteria [14].

4.2. Tree Based Classifier for Context Model

The problem of extracting context can also be viewed as a prediction problem. We want to predict the class of $X_s^{(n)}$ using the classes of $X_{\partial s}^{(n+1)}$. Since this prediction problem is discrete in nature, we use a decision tree structure to implement it. In the decision tree, each interior node corresponds to a test, and each terminal node (a leaf) is assigned the conditional probability of $X_s^{(n)}$ given all the test results on the path from the root to this leaf. The tests which are used in our approach are Boolean operations. In other words, one is allowed to ask "true-false" questions about $X_{\partial s}^{(n+1)}$ in order to predict the class of $X_s^{(n)}$. For example, one can ask, "Is the parent of s text?", or "Are the majority of the 9 coarse neighbors of s heading or background?". This querying procedure will stop when the class of $X_s^{(n)}$ can be determined with enough confidence or the number of questions asked reaches the maximal number of questions allowed. In our context model, the questions about $X_{\partial s}^{(n+1)}$ can be only asked about the features shown in Fig. 7. In order to allow documents to be scanned both horizontally and vertically, all questions allowed in our model are also 90° rotation symmetric.

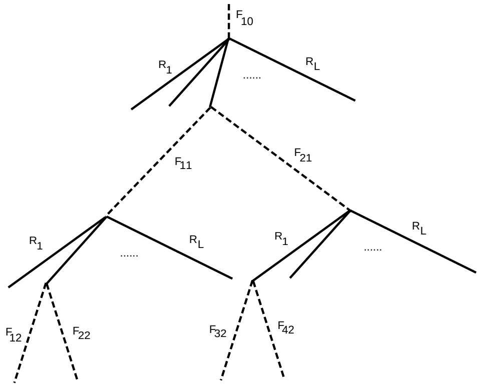


Figure 8: *How a decision tree is built using the minimum entropy test. R_i is a Boolean operation and $F_{j,k}$ is the set of values that can be taken by $X_{\partial s}^{(n+1)}$ at the node $\pi_{j,k}$.*

With a database of training images and "ground truth" segmentations, the decision tree can be built off-line. Ideally, we want to build an optimal decision tree in the sense that it minimizes the classification error. But for practical applications, an optimal decision tree is too difficult, or too computationally expensive to build. So, in this section, we present a sub-optimal approach to build a decision tree using a greedy algorithm, called the minimum entropy test [8]. In this approach, the test which is chosen for each new node in the tree is the one which minimizes the residual uncertainly at that node.

In order to formulate the entropy testing rule more precisely, we define $H(U|B)$ as the entropy of a random variable U given an event B.

$$H(U|B) = -\sum_u p(U=u|B)\log_2 P(U=u|B)$$

where in our case, B are the outputs of Boolean operations. Each Boolean operation can take one of two values, 1 or 0. Let R be a set of Boolean functions. $R_i(X_{\partial s}^{(n+1)})$ is the output of the i-th Boolean function given $X_{\partial s}^{(n+1)}$. The j-th node on the k-th level of a decision tree is denoted as $\pi_{j,k}$, where $k \geq 0$, $1 \leq j \leq 2^k$, and $\pi_{0,1}$ is the root. The test chosen for $\pi_{j,k}$ is denoted as $R_{\pi_{j,k}}$. Also, let Ω be the values which can be taken by $X_{\partial s}^{(n+1)}$, and $F_{j,k}$ be the set of values which can be taken by $X_{\partial s}^{(n+1)}$ at the node $\pi_{j,k}$. Then the decision tree can be built as follows (Fig. 8) using the minimum entropy test.

1. Start from the root $\pi_{1,0}$ with $k=0$, $j=1$, and $F_{1,0}=\Omega$.

2. Find i^*, such that

$$i^* = \arg\min_i H(X_s^{(n)}|R_i(X_{\partial s}^{(n+1)}),\ X_{\partial s}^{(n+1)} \in F_{j,k})$$

3. Choose R_{i^*} to be $R_{\pi_{j,k}}$.
4. Partition $F_{j,k}$ into $F_{2j-1,k+1}$ and $F_{2j,k+1}$, such that

$$F_{2j-1,k+1} = \{x_{\partial s}^{(n+1)} \in F_{j,k},\ \ R_{i^*}(x_{\partial s}^{(n+1)}) = 1\}$$

$$F_{2j-1,k+1} = \{x_{\partial s}^{(n+1)} \in F_{j,k},\ \ R_{i^*}(x_{\partial s}^{(n+1)}) = 0\}$$

5. If the entropy of the tree is small enough or the maximal level of the decision tree is reached, stop.
6. Use a breadth-first rule to pick a new node.
7. Go to (2).

For our context model, a decision tree is built for each resolution, except the coarsest one. After decision trees are constructed, we need to estimate their parameters θ, where θ is a vector whose elements are the conditional probabilities of $X_s^{(n)}$ at each leaf of the decision trees. Because $X^{(>0)}$ are unknown, we use the EM algorithm [6] to estimate θ. The EM algorithm can be written as the following iterative procedure for finding a sequence $\theta^{(k)}$ that converges to the ML estimate of θ.

$$\theta^{(k+1)} = \arg\max_{\theta} E\big[\log p_\theta(X^{(>0)})|Y, X^{(0)}, \theta^{(k)}\big] \quad (10)$$

where $X^{(0)}$ is the "ground truth" segmentation. The expectation in (10) is difficult to compute. However, assuming the pyramidal graph model, we can generate samples from the distribution $p(X^{(>0)}|Y, X^{(0)}, \theta^{(k)})$ using the Metropolis algorithm [12], and approximate the expectation in (10) using the histogram of samples.

5. Simulation Results

We test our algorithm with an image data base consisting of 40 document images which are scanned at 100dpi. 20 images are used as training images (see Fig. 9), and the remaining 20 images are used as test images. Training images are manually segmented into text, image, heading and background (see Fig. 9). These segmentations are used as "ground truth" for parameter estimation. The algorithm is coded in C and runs on an HP model 755 workstation.

For the image model, we only extract local texture information from resolution 0 (the finest resolution) to resolution 4. For each resolution, the image features are modeled using a Gaussian mixture model as discussed in section 4.1. Each Gaussian mixture density contains 10 or fewer mixture components. For the context model in our experiment, each decision tree has 5 levels, 2^5 leaves.

We did two simulations. Simulation I shows the importance of both the multiscale image model and the context model (see Fig. 10). Fig. 10(a) is the original image, and Fig. 10(b) is the segmentation result when we used the proposed document segmentation algorithm which consists of the multiscale image model and the hybrid context model. The segmentation is smooth with most of the areas labeled correctly. In Fig. 10(c) and (d), we show two segmentation results with a degraded image model or a degraded context model. Fig. 10(c) is the segmentation result when we use a degraded image model and the same hybrid context model. The degraded image model extracts texture information only from the finest scale. Therefore, the resulting segmentation has many misclassified areas. Fig. 10(d) is the segmentation result when we use a degraded context model the quadtree model, and the multiscale image model. Comparing Fig. 10(c) and Fig. 10(d), we see the significant improvement of the segmentation result due to the hybrid context model. In simulation II, we tested our algorithm on the 20 test images. Two of the results are shown in Fig. 11.

From both simulation results, we see that most of the regions are classified correctly. Even single text lines, page marks (see Fig. 11(b)), periods, and the dots in the character "i" (see Fig. 11(b) and Fig. 11(d)) are labeled correctly. The gradually changing background, such as the background in Fig. 10, is also segmented out as a solid and smooth background region. But the textured background, such as the heading in Fig. 10, is segmented as image instead of heading and background.

6. Conclusion

A new approach for document segmentation has been proposed in this paper. The model captures the characteristics of document images by extracting both local and contextual information from different scales and modeling them as multiscale random fields. The SMAP estimator is used to segment the document image by minimizing the expected size of the largest misclassified region.

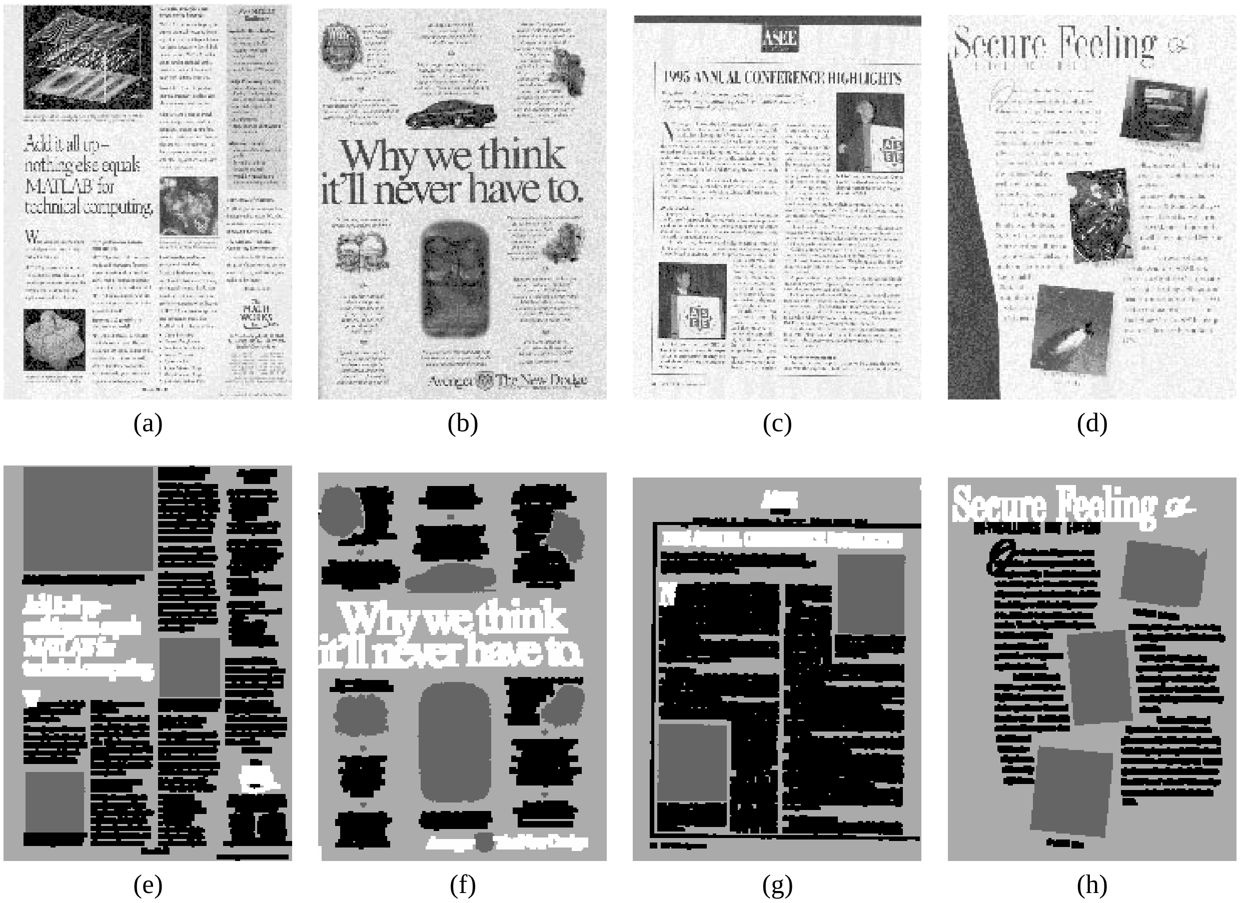
Experiments with real document images indicate that the new approach is computationally efficient and improves the segmentation accuracy.

References

[1] M. Aitkin and D.B. Rubin, "Estimation and hypothesis testing in finite mixture models", *J. of Royal Statistical Soc.*, vol. B-47, no. 1, pp. 67-75, 1985.

[2] D.S. Bloomberg, "Multiresolution Morphological analysis of document images", *Visual Communications and Image Processing '92*, SPIE vol. 1818, pp. 648-662, 1992.

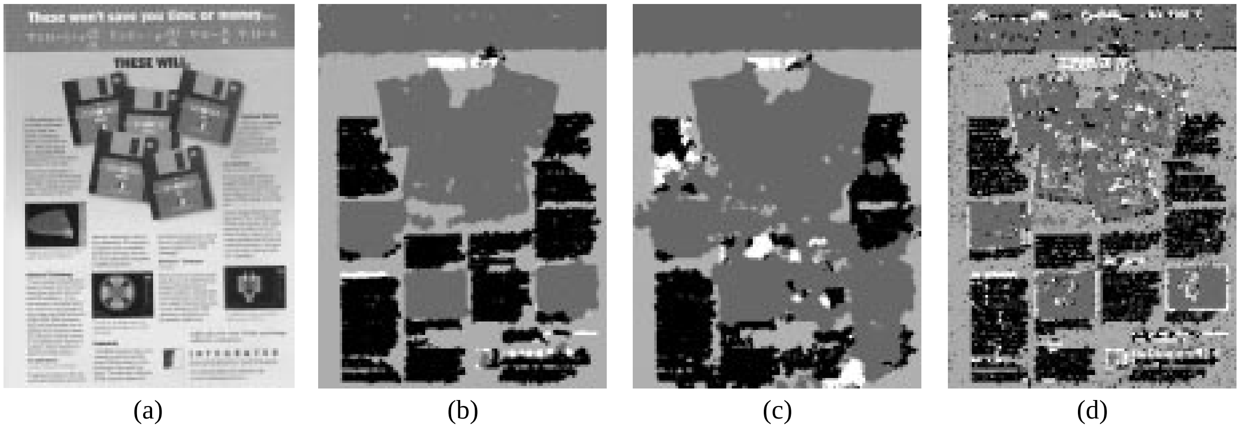
[3] P.J. Bones, T.C. Griffin, C.M. Carey-Smith, "Segmentation of document images", *Image Communica-*


Add it all up – nothing else equals MATLAB for technical computing.
Why we think it'll never have to.
ASEE
1995 ANNUAL CONFERENCE HIGHLIGHTS
Secure Feeling

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

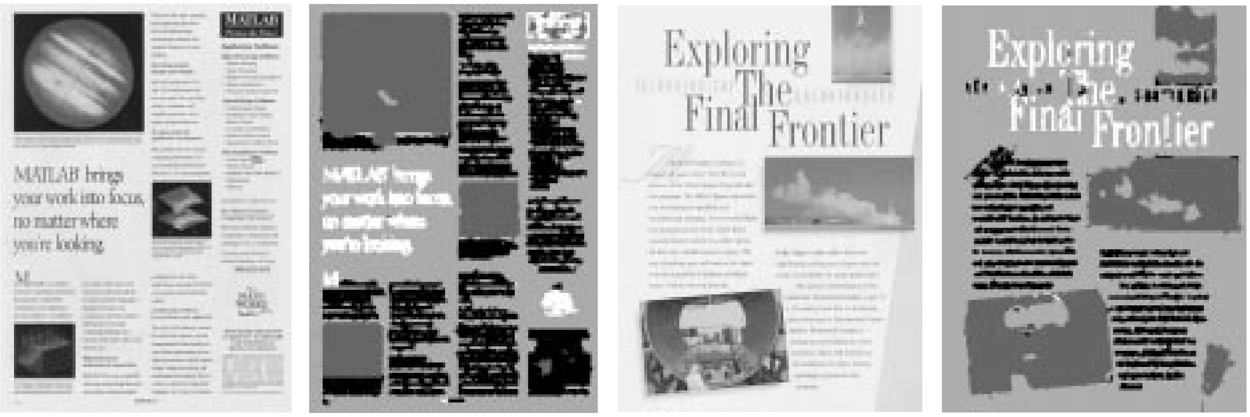
(e) (f) (g) (h)

Figure 9: *Training images and the corresponding "ground truth" segmentations: (a)-(d) are training images, and (e)-(h) are "ground truth" segmentations. The dark areas are text, the dark grey areas are images, the light grey corresponds to background, and the white areas are the headings.*



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

Figure 10: *Simulation I. (a) The original image. (b) The segmentation result with the multi-resolution image model and the hybrid context model. (c) The segmentation result with the single resolution image model and the hybrid context model. (d) The segmentation result with the multi-resolution image model and the quadtree context model. The dark areas are text, the dark grey areas are images, the light grey corresponds to background, and the white areas are the headings.*


MATLAB brings your work into focus, no matter where you're looking
Exploring The Final Frontier

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

Figure 11: *The original image, the segmentation results with the multi-resolution image model, and the hybrid context model of simulation II. The dark areas are text, the dark grey areas are images, the light grey corresponds to background, and the white areas are the headings.*

tions and Workstations , SPIE vol. 1258, pp. 78-88, 1990.

[4] C.A. Bouman and M. Shapiro, “A Multiscale Random Field Model for Bayesian Image Segmentation”, *IEEE Tran. on Image Processing*, vol. 3, no. 2, pp. 162-177, March 1994.

[5] P. Chauvet, J. Lopez-Krahe, E. Tafin and H. Maitre, “System for an intelligent office document analysis, recognition and description”, *Signal Processing*, vol. 32, pp. 161-190, 1993.

[6] 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.

[7] K. Etemad, D. Doermann, and R. Chellappa, “Page segmentation using decision integration and wavelet packets”, *Proc. Int. Conf. on Pattern Recognition*, vol. 2, pp. 345-349, 1994.

[8] D. Geman and B. Jedynak, “An active testing model for tracking roads in satellite images”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 5, Jan. 1996.

[9] R.M. Haralick, “Document image understanding: geometric and logical layout”, *Proc. IEEE Computer Soc. Conf. on Computer Vision and Pattern Recognition*, vol. 8, pp. 385-390, 1994.

[10] A.K. Jain and S. Bhattacharjee, “Text segmentation using Gabor filters for automatic document processing”, *Machine Vision and Applications*, vol. 5, pp. 196-184, 1992.

[11] A.K. Jain and Y. Zhong, “Page Segmentation Using Texture Analysis”, *Pattern Recognition*, vol. 29, no. 5, pp. 743-770, 1996.

[12] N. Metropolis, A.W. Rosenbluth, A.H. Teller, and E. Teller, “Equations of state calculations by fast computing machines”, *J. Chem. Phys.*, vol. 21, pp. 1087-1092, 1953.

[13] M.Krishnamoorthy, G. Nagy, S. Seth, and M. Viswanathan, “Syntactic segmentation and labeling of digitized pages from technical journals”, *IEEE Tran. on Pattern Analysis and Machine Intelligence*, vol. 15, no.7, pp. 737-747, July 1993.

[14] J. Rissanen, “A universal prior for integers and estimation by minimum description length”, *Annals of Statistics*, vol. 11, no. 2, pp. 417-431, 1983.

[15] P.P. Vaidyanathan, *Multirate systems and filter banks*, Prentice Hall, Englewood Cliffs, NJ, 1993.

[16] D. Wang and S.N. Srihari, “Classification of newspaper image blocks using texture analysis”, *Computer Vision, Graphics, and Image Processing*, vol. 47, pp. 327-352, 1989.

[17] K.Y. Wong, R.G. Casey, and F.M. Wahl, “Document analysis system”, *IBM J. Res. Develop.*, Vol 26, pp. 647-656, 1982.