Trainable Context Model for Multiscale Segmentation *†

Hui Cheng and Charles A. Bouman
School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47907-1285
{hui, bouman}@ ecn.purdue.edu

Abstract

Most previous approaches to Bayesian segmentation have used simple prior models, such as Markov random fields (MRF), to enforce regularity in the segmentation. While these methods improve classification accuracy, they are not well suited to modeling complex contextual structure.

In this paper, we propose a context model for multiscale segmentation which can capture very complex behaviors on both local and global scales. Our method works by using binary classification trees to model the transition probabilities between segmentations at adjacent scales. The classification trees can be efficiently trained to model essential aspects of contextual behavior. In addition, the data model in our approach is novel in the sense that it can incorporate the correlation among the wavelet feature vectors across scales. We apply our method to the problem of document segmentation to illustrate its usefulness.

1 Introduction

It is well known that accurate prior modeling can substantially improve segmentation accuracy by incorporating prior information regarding context. To this end, previous approaches to statistical segmentation have generally relied on simple prior models, such as Markov random fields (MRF), that encourage the formation of large uniformly classified regions. However, in many applications prior information regarding context can be very complex, with both local and global aspects. For example, when segmenting a scene, we may know that "sky" may not be surrounded by "ground" or that "cars" must have smooth boundaries while "trees" have irregular boundaries.

In this paper, we introduce a multiscale Bayesian context model which can capture complex aspects of both local and global contextual behavior. The method is based on the use of tree classifiers [1, 2] to model the transition probabilities between adjacent scales in the multiscale structure. This multiscale structure is similar to previously proposed segmentation models [3, 4, 5, 6], with the segmentations at each scale forming a Markov chain in scale. However, the tree based classifier allows for much more complex transition rules, while using only a moderate number of parameters. Moreover, the tree based classifier is computationally efficient to both train and apply.

In addition, the multiscale data model in this approach can capture both the textural information at all scales and the correlation among feature vectors across scales. This is different from most of other approaches which assume independence among image features. In our data model, the textural information at various scales is captured through a hidden Markov model [7], and the dependence of features between adjacent scales is extracted using linear prediction.

We apply this algorithm to the problem of document segmentation because it represents a excellent example of an application in which complex contextual dependencies can be very important [8]. The model is trained using training images and their "ground truth" segmentations in a one-pass coarse-to-fine process.

2 Multiscale Segmentation Model

Fig. 1 shows the structure of our multiscale segmentation model [6]. The pyramid to the left of Fig. 1 is the multiscale context model. Each point in the pyramid contains a value $x_s^{(n)}$ which is a class label at scale n and position s in the 2-D lattice $S^{(n)}$. Each label $x_s^{(n)}$ is assumed to depend on some neighborhood of points at the coarser scale, $x_{\partial s}^{(n+1)}$. Here, ∂s denotes a neighborhood of points in the lattice $S^{(n+1)}$ at scale $n+1$. Therefore, the segmentations $x^{(n)}$ forms a Markov chain in scale.

The pyramid to the right of Fig. 1 is the multiscale date model which contains the feature vectors

*This work was supported by the Xerox Corporation.
†Appeared in ICIP '98, vol. 1, pp. 610-614, Chicago, IL, October 4-7, 1998.

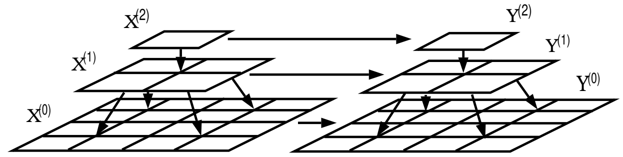


Figure 1: The structure used for our multiscale segmentation model. The left pyramid models the contextual behavior, while the right pyramid models the data features extracted using a Haar wavelet transform.

extracted from the gray scale image using a simple Haar basis wavelet transform. Each feature, $y_s^{(n)}$, is a vector of three numbers corresponding to the three wavelet coefficients at that location. The feature vector $y_s^{(n)}$ is assumed conditionally independent given the class labels $x_s^{(n)}$ and feature vector $y_{ds}^{(n+1)}$ at the coarser scale, where $ds \in \partial s$ is referred to as the parent of s. We also assume that $y_s^{(n)}$ can be predicted by $\alpha_{x_s}^{(n)} y_{ds}^{(n+1)} + \beta_{x_s}^{(n)}$, and the prediction error $\tilde{y}_s^{(n)}$ are conditionally independent given $x_s^{(n)}$,

$$\tilde{y}_s^{(n)} = y_s^{(n)} - \left[\alpha_{x_s}^{(n)} y_{ds}^{(n+1)} + \beta_{x_s}^{(n)}\right]$$

where $\alpha_{x_s}^{(n)}$, $\beta_{x_s}^{(n)}$ are the prediction coefficients for class $x_s^{(n)}$ at scale n. Our data model is different from other approaches in that it models not only the textural information at each scale, but also captures the correlation among the wavelet feature vectors across scales.

To compute the segmentation, we use the sequential MAP (SMAP) segmentation approach of [3]. In the SMAP approach, classification proceeds sequentially from coarse to fine scales, by computing the MAP segmentation given Y and the previous coarser scale segmentation $\hat{x}^{(n+1)}$. Formally, the SMAP segmentation is given by

$$\begin{aligned} \hat{x}_s^{(n)} \quad &= \quad \arg \max_{0 \le k \le M-1} \left\{ \log p_{y|x_s^{(n)}}(y|k) \right. \qquad (1) \\ &\qquad \left. + \ \log p_{x_s^{(n)}|x_{\partial s}^{(n+1)}}(k|\hat{x}_{\partial s}^{(n+1)}) \right\} \end{aligned}$$

where M is the number of classes. Importantly, the term $\log p_{y|x_s^{(n)}}(y|k)$ may be efficiently computed in closed form using a simple hidden Markov model define for a quadtree [6]. Notice that (1) consists of two terms, the first is a data term while the second enforces *a priori* knowledge regarding context, and is solely determined by the transition probability $p_{x_s^{(n)}|x_{\partial s}^{(n+1)}}(k|\hat{x}_{\partial s}^{(n+1)})$.

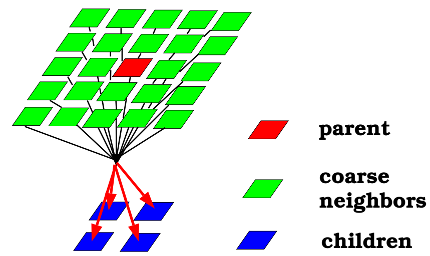


Figure 2: This diagram illustrates the transition probabilities that interpolate a pixel at the previous coarser resolution into four pixels at the current resolution using a 5×5 coarse scale neighborhood.

3 Trainable Context Model

For our context model, we assume that each label $x_s^{(n)}$ only depends on the labels of a neighborhood of pixels at the previous coarser resolution $x_{\partial s}^{(n+1)}$. Intuitively, this is a model for interpolating a pixel $s^{(n+1)}$ into four children pixels $s_i^{(n)}$ $(i = 1, 2, 3, 4)$ at the next finer resolution.

Fig. 2 shows the case of interpolating four children pixels using a 5×5 neighborhood at the coarser scale. Since there are four distinct children for each coarse neighborhood, we will use four distinct transition probability functions, $p_i^{(n)}(\cdot|\cdot)$ $(i = 1, 2, 3, 4)$ at each resolution n. Define an indicator function $I(s)$, where $I(s) = i$, if s is the i-th child of its parent. Also, we use the variables c and f to denote $x_s^{(n)}$ and $x_{\partial s}^{(n+1)}$ respectively. Then, the transition probabilities are given by

$$p_{x_s^{(n)}|x_{\partial s}^{(n+1)}}(c|f) = \quad p_{I(s)}^{(n)}(c|f) \ . \qquad (2)$$

Unfortunately, $p_i^{(n)}(c|f)$ may be very difficult to estimate, if the coarse scale neighborhood is large. For example, a 5×5 neighborhood using $M = 2$ results in 2^{25} possible values of f, which makes direct parameter estimation impossible.

In order to model each of the transition probabilities $p_i^{(n)}(c|f)$, we use a class probability tree (CPT) [1] as shown in Fig. 3. The CPT represents a sequence of decisions or tests that must be made in order to compute the distribution of c given f. The input to the tree is $\tilde{f}$, an encoded version of the feature vector f. At each interior node, a splitting rule is used to determine which of the two child nodes should be taken. In our case, the splitting rule is computed by comparing $A_t \tilde{f} - \mu_t$ to 0, where A_t is a pre-computed matrix and μ_t is a pre-computed scalar. In this way, $\tilde{f}$ goes down

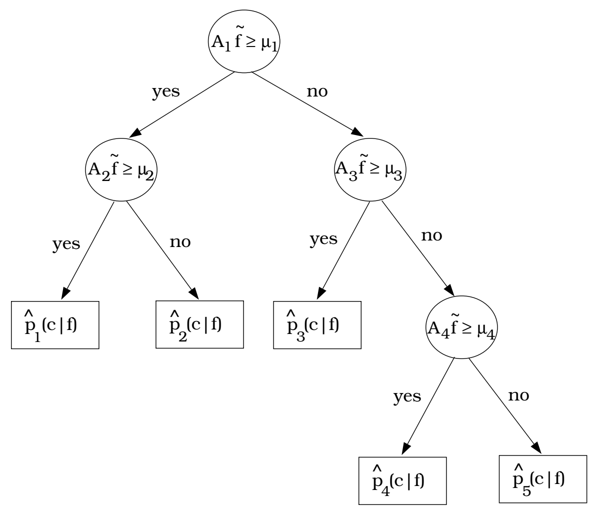


Figure 3: The class probability tree used to represent transition probabilities between adjacent scales. At each interior node, a linear test is used to split the node. The probability mass function $p_i^{(n)}(c|f)$ is approximated separately at each leaf node.

the tree until it reaches a leaf node. Each leaf node $\tilde{t}$ is assigned a probability mass function $\hat{p}_{\tilde{t}}(c)$. When $\tilde{f}$ reaches a leaf node $\tilde{t}$, $\hat{p}_{\tilde{t}}(c)$ is used as an approximation to the true transition probability $p_i^{(n)}(c|f)$.

To construct a CPT, we use the recursive tree construction algorithm (RTCA) of Gelfand, Ravishankar, and Delp [2] together with a least squares based splitting rule. The RTCA partitions the training sample set into two halves and alternates the role of each half. Initially, a tree is grown using the first half and pruned on the second half. Then the pruned tree is re-grown using the second half and pruned using the first half. This process is repeated until the tree converges.

The sample sets for constructing a CPT are generated from a database of training images which is produced by scanning typical documents. Scanned images are then manually segmented into desired components which form the "ground-truth" segmentation at each resolution.

The training of the CPT's is done in a coarse-to-fine procedure which is illustrated in Fig. 4. The method works by estimating the transition probabilities $p_i^{(n)}(c|f)$ from the "ground truth" segmentation $\tilde{x}^{(n)}$ and the coarser scale SMAP segmentation $\hat{x}^{(n+1)}$. Importantly, $\hat{x}^{(n+1)}$ does not depend on the transition probabilities $p_i^{(n)}(c|f)$. This can be seen from (1), the equation for computing the SMAP segmentation. This is a crucial fact since it allows $\hat{x}^{(n+1)}$ to be computed before $p_i^{(n)}(c|f)$ is estimated. Once $p_i^{(n)}(c|f)$ is esti-

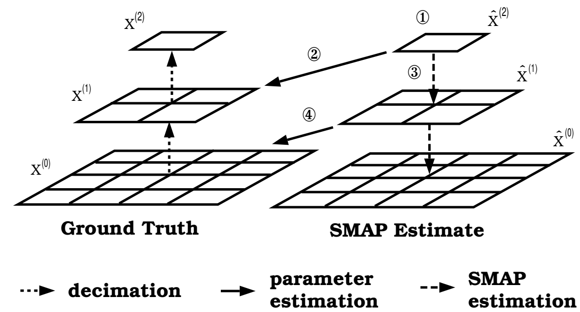
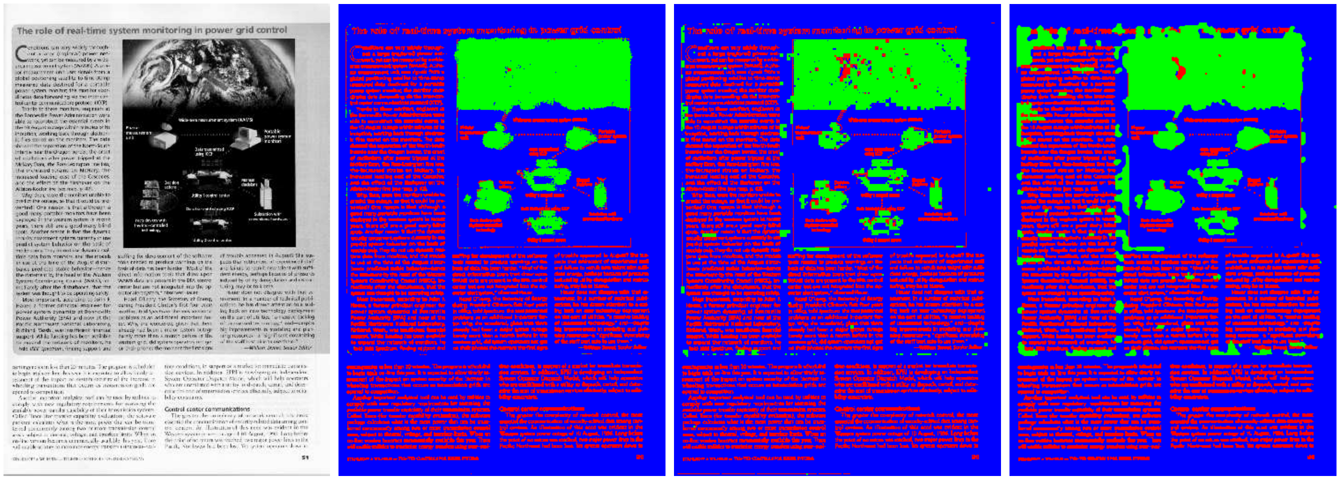


Figure 4: Parameter estimation of the context model: (1) Compute the segmentation at the coarsest resolution, $\hat{x}^{(2)}$. (2) Estimate the transition probabilities $p_i^{(1)}(c|f)$ using the SMAP segmentation $\hat{x}^{(2)}$ and the decimated "ground truth" segmentation $\tilde{x}^{(1)}$. (3) Compute $\hat{x}^{(1)}$ using $p_i^{(1)}(c|f)$. (4) Estimate $p_i^{(0)}(c|f)$ using $\hat{x}^{(1)}$ and $\tilde{x}^{(0)}$. This procedure is then repeated for all scales.

mated, it is then used to compute $\hat{x}^{(n)}$, allowing the estimation of $p_i^{(n-1)}(c|f)$. This process is then recursively repeated until the transition parameters at all scales are estimated. This training procedure is very computationally efficient and yields accurate segmentation results.

4 Simulations

Fig. 5 shows an example of segmenting a document image that is not contained in the training data set. The document is segmented into background, text and image classes. The model was first trained using 20 scanned document images together with hand segmented "ground truth". Fig. 5(a) is the original image, Fig. 5(b) shows the result of segmentation using the proposed segmentation algorithm, referred as the trainable SMAP (TSMAP) algorithm, with a 5×5 coarse scale neighborhood, Fig. 5(c) shows the segmentation using TSMAP with a 1×1 coarse scale neighborhood, and Fig. 5(d) shows the segmentation using only the finest resolution features combined with the MRF as the context model. Notice that the larger neighborhood dramatically improves the quality and accuracy of segmentation. It is interesting to note that "image" regions are enforced to be uniform, while "text" regions are allowed to be small with fine detail. Even single text lines, reverse text and page marks are labeled correctly. Our algorithm also correctly classifies image regions from different backgrounds, such as paper background, halftone background and black background. Among the segmentation results shown in Fig. 5, the segmentation computed using the MRF



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

Figure 5: (a) Original image. (b) Segmentation result using TSMAP with a 5×5 neighborhood. (c) Segmentation result using TSMAP with a 1×1 neighborhood. (d) Segmentation using MRF. Red, green and blue represent text, image and background, respectively.

is the poorest. There are many small background regions between text lines that are misclassified as image regions, and the boundaries of image regions are quite irregular. The misclassifications indicate that the MRF fails to capture enough contextual information. Fig. 6 shows the segmentation results of two other images outside the training set using TSMAP with a 5×5 coarse neighborhood.

In Fig. 7, we show the effect of the number of training images on the quality of the resulting segmentation. The TSMAP algorithm with a 5×5 coarse scale neighborhood is trained on three training sets which consist of 20, 10 and 5 training images, respectively. The resulting segmentations are shown in Fig. 7(b)-(d). Notice that the segmentations degrade gracefully as the number of training images decreases. There is little difference between the segmentation computed using the TSMAP trained on 20 images and the segmentation based on 10 training images. But when the number of training images is too small, such as 5, the segmentation result (see Fig. 7 (d)) begins to degrade. The experiment shows that the TSMAP algorithm is not excessively sensitive to the size of the training set.

References

[1] L. Breiman, J. H. Friendman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Belmont, CA: Wadsworth International Group, 1984.

[2] S. B. Gelfand, C. S. Ravishankar, and E. J. Delp, "An iterative growing and pruning algorithm for classification tree design," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, no. 2, pp. 163–177, February 1984.

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

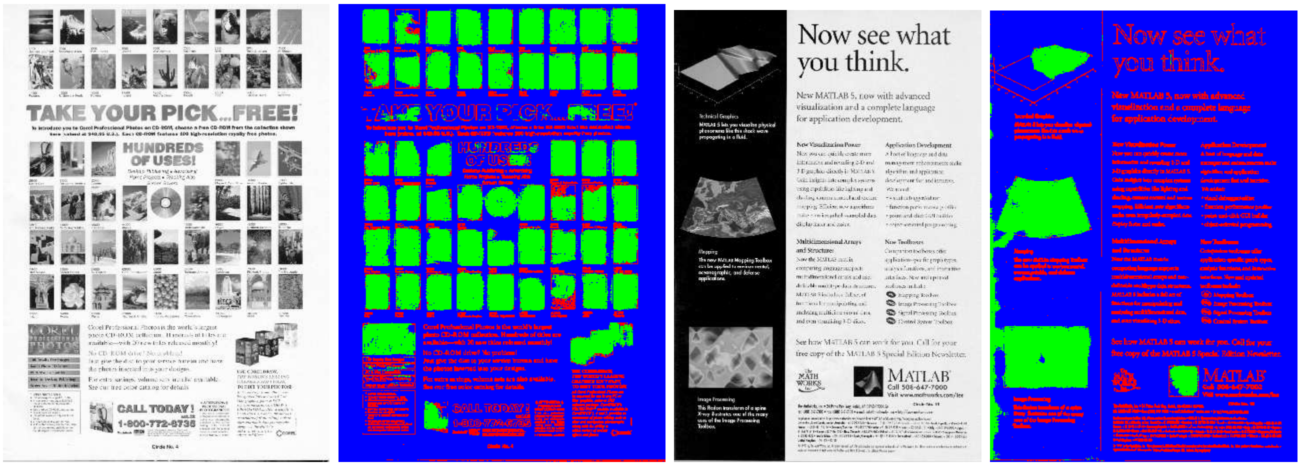
[4] J. M. Laferte, F. Heitz, P. Perez, and E. Fabre, "Hierarchical statistical models for the fusion of multiresolution image date," *Proc. Int'l Conf. on Computer Vision*, June 1995, Cambridge, MA, pp. 908–913.

[5] M. L. Comer and E. J. Delp, "Multiresolution image segmentation," *Proc. of IEEE Int'l Conf. on Acoust., Speech and Sig. Proc.*, May 1995, Detroit, Michigan, pp. 2415–2418.

[6] H. Cheng, C. A. Bouman, and J. P. Allebach, "Multiscale document segmentation," *Proc. of IS&T's 50th Annual Conf.*, May 1997, Cambridge, MA, pp. 417–425.

[7] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, "Wavelet-based statistical signal processing using hidden markov models," *IEEE Trans. on Signal Processing*, vol. 46, no. 4, pp. 886–902, April 1998.

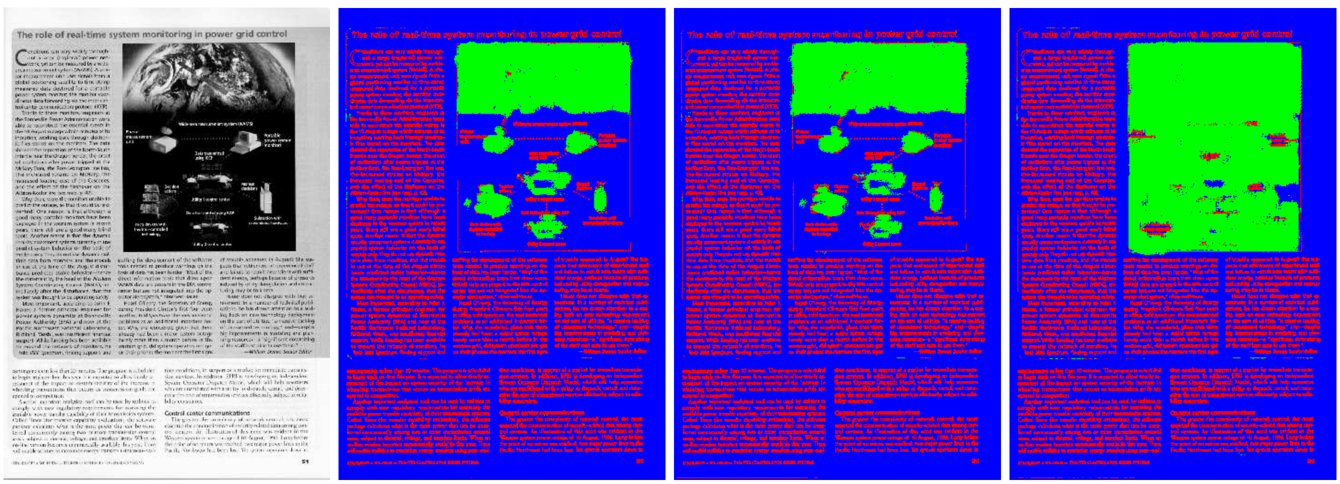
[8] R. M. Haralick, "Document image understanding: Geometric and logical layout," *Proc. of IEEE Computer Soc. Conf. on Computer Vision and*



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

Figure 6: (a) Original image. (b) Segmentation result using TSMAP with a 5×5 neighborhood. (c) Original image. (d) Segmentation result using TSMAP with a 5×5 neighborhood. Red, green and blue represent text, image and background, respectively.

Pattern Recognition, vol. 8, April 1994, Seattle, WA, pp. 385–390.



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

Figure 7: (a) Original image. (b) TSMAP segmentation when trained on 20 images. (c) TSMAP segmentation when trained on 10 images. (d) TSMAP segmentation when trained on 5 images. Red, green and blue represent text, image and background, respectively.