CLASSIFICATION OF IMAGES USING SPATIAL RANDOM TREES.

W. Wang, I. Pollak, C.A. Bouman, and M.P. Harper

Purdue University
School of Electrical and Computer Engineering
West Lafayette, IN 47907

1. INTRODUCTION.

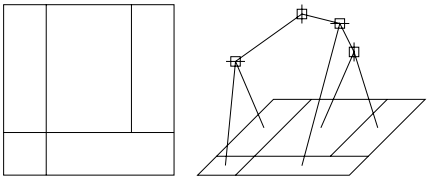
In this work we develop a new methodology for constructing hierarchical stochastic image models called spatial random trees (SRTs). Similarly to [2, 4], our models are stochastic processes on trees with each leaf corresponding to a single pixel. Our key innovation, however, is that the tree structure itself is random and is generated by a probabilistic grammar [15]. We are motivated by the wide use of probabilistic grammars in natural language processing, for example, to model the structure of sentences [10], as well as by more recent efforts to apply probabilistic grammars to 2-D problems such as optical character recognition [14].

We build upon our earlier research on stochastic grammars [11–13], and demonstrate a close relationship between SRTs and multitree dictionaries introduced in [6–8]. We show how to find the MAP estimate of both the tree structure and the tree states for an SRT model. An EM algorithm [3] which we use to train the model and which we call the center-surround algorithm is described in [16]. It is an extension of the forward-backward [10] and inside-outside [1, 9, 10] algorithms which are used for training hidden Markov models and probabilistic context-free grammars, respectively.

2. MULTITREE DICTIONARIES.

We first review the framework of multitree dictionaries [6–8] which we will use to develop our SRT image models. Multitree dictionaries are defined using grammar formalism [5, 10]. We define a *grammar* $G = (A, S)$ to be a pair which consists of a set A of *symbols*, and a function S which maps symbols to finite sets of symbols. If $a \in A$ and $\alpha \in S(a)$, the expression $a \rightarrow \alpha$ is called a *split* or a *production*, and has the interpretation that the symbol a generates the set α.

For example, the symbols may represent different rectangular tiles of an image, as in Fig. 1, or coefficients of an image with respect to different orthogonal bases. By starting with a single element of A, we can generate various sets of elements of A via recursive splitting—i.e., recursive application of productions, see Fig. 1. This process can be visualized as a tree where each production $a \rightarrow \alpha$ is depicted as a node labeled a whose children are labeled with the elements of α. Following [6–8], we let a *multitree dictionary* $\mathcal{D}_a(G)$ be the set of all such trees that can be produced by the grammar G, starting with the root symbol a. When there is no possibility of confusion, we will simply denote such a dictionary by $\mathcal{D}$. We say that a grammar $G = (A, S)$ is *finite-depth* if, for every $a \in A$, $\mathcal{D}_a(G)$ is a finite set containing only finite-depth trees. This can be insured by only allowing a finite set of symbols to be descendants of a, and not allowing a to be its own descendant. In this paper, we only work with finite-depth grammars.



(a) A tiling. (b) A tree of splits.

Fig. 1: An illustration of rectangular image tilings: (a) a tiling; (b) a corresponding tree of splits.

Suppose that each symbol $u \in A$ is assigned a cost $c(u)$, and that each production $u \rightarrow \alpha$ is assigned a cost $\bar{c}(u \rightarrow \alpha)$. Further assume that $\text{COST}(t)$ for any tree $t \in \mathcal{D}_a(G)$ is the sum of the individual costs of all the productions comprising t, plus the sum of the costs of all its leaves:

$$\text{COST}(t) = \sum_{(u \rightarrow \alpha) \in t} \bar{c}(u \rightarrow \alpha) + \sum_{u \in \text{leaves}(t)} c(u). \quad (1)$$

We would like to find the best tree in the dictionary $\mathcal{D}_a(G)$, i.e., the tree t_a^* whose cost is the smallest:

$$t_a^* = \arg \min_{t \in \mathcal{D}_a(G)} \text{COST}(t).$$

We denote the corresponding cost by C_a^*, i.e., $C_a^* = \text{COST}(t_a^*)$. This problem can be solved using an efficient recursive algorithm for best tree search described in [6–8]. To illustrate this algorithm, we suppose that the only allowed split of the

This work was supported in part by a National Science Foundation (NSF) grant IIS-0329156, an NSF CAREER award CCR-0093105, a Xerox Foundation grant, and ARDA under contract number MDA904-03-C-1788. Part of this work was carried out while the fourth author was on leave at NSF. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the view of NSF.

symbol a is: $a \rightarrow b_1\ b_2$. There is a tree $\{a\}$ in the multitree dictionary $\mathcal{D}_a(G)$ which consists of one node labeled a, with $\text{COST}(\{a\}) = c(a)$. For any other tree $t \in \mathcal{D}_a(G)$, its left subtree t_{left} is in $\mathcal{D}_{b_1}(G)$, and its right subtree t_{right} is in $\mathcal{D}_{b_2}(G)$. Therefore, since the cost is additive, $\text{COST}(t) = \bar{c}(a \rightarrow b_1\ b_2) + \text{COST}(t_{left}) + \text{COST}(t_{right})$. Consequently, the optimal tree is:

$$t_a^* = \begin{cases} \begin{array}{ccc} & a & \\ t_{b_1}^* & & t_{b_2}^* \end{array} & \text{if } \bar{c}(a \rightarrow b_1\ b_2) + C_{b_1}^* + C_{b_2}^* < c(a) \\ \{a\} & \text{otherwise.} \end{cases}$$

In other words, we find the best trees $t_{b_1}^*$ and $t_{b_2}^*$ in the dictionaries $\mathcal{D}_{b_1}(G)$ and $\mathcal{D}_{b_2}(G)$, respectively, and compare their total cost plus the cost of the root production $a \rightarrow b_1\ b_2$, with the cost of the tree $\{a\}$. We have a similar recursion in the general case, as described in [6, 8].

We emphasize that, despite its appearance, our fast recursive search algorithm for the globally optimal tree is *neither* a greedy search *nor* an exhaustive search algorithm. While the number of trees can be exponential in the number of symbols, the complexity of this algorithm is only O(number of all productions) which, in many applications, can be made linear or polynomial in the number of symbols.

3. SPATIAL RANDOM TREE = MULTITREE DICTIONARY + PROBABILITY.

There are a variety of ways to adapt the framework of multitree dictionaries to probabilistic modeling. We now describe one such construction. We suppose that the discrete image domain Q is a $2^L \times 2^L$ square where L is a fixed integer. We define the following grammar $G = (A, S)$ to describe hierarchical segmentations of the domain Q. Following the conventions used in context-free grammars [10], three types of symbols are defined: the root symbol which can only appear at the root, the nonterminal symbols which can only appear at the nonroot internal nodes, and the terminal symbols which can only appear at the leaves.

- We use the root symbol $u = (0, Q)$.
- The nonterminal symbols have the form $u = (j, R)$ where R is a dyadic rectangular subset of Q (i.e., obtainable through a recursive dyadic splitting of Q) such that $|R| > 1$, and the number j is an integer from a fixed set $\{1, \ldots, M\}$ which can represent, for example, the classification of the image pixels belonging to R into one of M classes.
- The terminal symbols have the form $u = (j, \{n\})$ where $\{n\}$ is a 1×1 rectangle (i.e., $n \in Q$) and j is as above.

We let $\mathcal{R}$ be the set of all valid regions, i.e., the set of all dyadic subrectangles of Q, including 1×1 rectangles. We now define the corresponding set of productions.

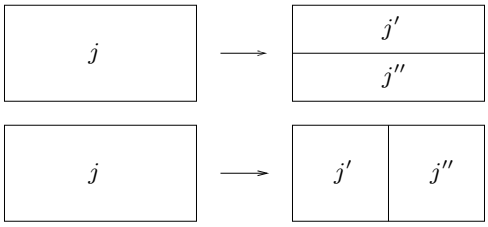


Fig. 2: Nonroot productions for Construction 1: a horizontal split into two congruent rectangles (top) and a vertical split into two congruent rectangles (bottom).

- The root productions are $(0, Q) \rightarrow (j, Q)$ for $j = 1, \ldots, M$.
- The remaining productions are defined for all nonterminal symbols $u = (j, R)$. These productions are $(j, R) \rightarrow (j', R')\ (j'', R'')$, for all $j, j', j'' = 1, \ldots, M$ and for all partitions of R into two congruent rectangles R' and R'',[1] as illustrated in Fig. 2.

For each symbol $u = (j, R)$, we call j the *state* and we call R the *region*. If this symbol appears in a tree, we say that the region R is *labeled* j.

For every root or nonterminal symbol u, we specify a probability distribution $\bar{\mathbf{p}}_u$ on $S(u)$ (recall that $S(u)$ is the set of all righthand sides of the productions starting with "$u \rightarrow$"): $\sum_{\alpha \in S(u)} \bar{\mathbf{p}}_u(\alpha) = 1$, and let the cost of each production be the corresponding negative log-probability, $\bar{c}(u \rightarrow \alpha) = -\log \bar{\mathbf{p}}_u(\alpha)$. Our observation model is as follows. For $j = 1, \ldots, M$, we let $\mathbf{p}_j$ be a probability density function over $\mathbb{R}$. A typical choice would be a Gaussian density, $\mathbf{p}_j(x) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(x-\mu_j)^2}{2\sigma_j^2}}$, where μ_j and σ_j are parameters which depend on j. Our model can thus be viewed as a generative process which first uses the productions of Fig. 2 to recursively subdivide the domain Q into progressively smaller rectangles until every rectangle is a single pixel, and then samples the value of each pixel from the conditional distribution $\mathbf{p}_j$.

Given an image f, we specify the cost of every leaf of the tree t as follows:

$$c(u) = \begin{cases} \infty & \text{if } u \text{ is the root symbol} \\ & \text{or a nonterminal symbol,} \\ -\log \mathbf{p}_j(f_n) & \text{if } u = (j, \{n\}). \end{cases}$$

In other words, we impose that each leaf must be an individual pixel, and we make the cost of a pixel be the negative log of its probability density.

We define the joint probability/probability density of a tree t and the data f as the product of the probabilities of the productions in the tree and the conditional probability

[1] Note that, for a 1×2^k or $2^k \times 1$ rectangle R there is only one possible partition into two congruent rectangles, and for any other dyadic rectangle R there are two possible partitions, namely, along the vertical and horizontal lines through the center of R.

densities of the observations:

$$\text{JOINT-PROB}(t, f) \stackrel{\triangle}{=} \prod_{(u\to\alpha)\in t} \bar{\mathbf{p}}_u(\alpha) \prod_{(j,\{n\})\in \text{leaves}(t)} \mathbf{p}_j(f_n).$$

Using an argument described in [11], it can be shown that we have defined a legitimate probability distribution on the set of all pairs (t, f), i.e., that

$$\int_{\mathbb{R}^{|Q|}} \sum_{t\in\mathcal{D}} \text{JOINT-PROB}(t, f) df = 1, \tag{2}$$

where $\mathcal{D}$ is the set of all trees generated by our grammar. This motivates defining the following probability density for images in $\mathbb{R}^{|Q|}$:

$$\text{IMAGE-PROB}(f) \stackrel{\triangle}{=} \sum_{t\in\mathcal{D}} \text{JOINT-PROB}(t, f). \tag{3}$$

Note that $\text{COST}(t)$ defined in Eq. (1) is then, for a fixed image f, given by $\text{COST}(t) = -\log \text{JOINT-PROB}(t, f)$, and therefore the problem of finding the maximum a posteriori probability (MAP) tree given an image is equivalent to minimizing $\text{COST}(t)$ and can be solved using the recursion described in the previous section. Moreover, the probability density of an image can be calculated using a simple modification of this recursion, essentially by replacing all minimizations of probabilities with sums. As we show in [16], this can be used with the EM algorithm [3] to estimate the parameters of the model, i.e., the production probabilities $\bar{\mathbf{p}}_u(\alpha)$ and the parameters of the leaf distributions $\mathbf{p}_j$. Note that, since our grammar describes a region-splitting process, any symbol (j, R) can occur at most once in a tree, and, moreover, any production $(j, R) \to (j', R')\ (j'', R'')$ can occur at most once in a tree. Therefore, we either need to use large amounts of training data to estimate the production probabilities, or reduce the number of independent parameters. We do the latter by requiring that, if $u = (j, R)$ and $u' = (j, R')$ where R and R' are congruent, then $\bar{\mathbf{p}}_u = \bar{\mathbf{p}}_{u'}$. This is further discussed in [16], in a more general setting.

We call such probabilistic models *spatial random trees* to emphasize the fact that the underlying grammar models spatial organization. We note that, in addition to defining a probability distribution over all images supported on Q, Eq. (3) induces a probability distribution over all images supported on any fixed subset of the domain Q: such a distribution is obtained via marginalization.

4. IMAGE CLASSIFICATION.

We now apply our model to image classification. Our first data set consists of 450 64×64 eight-bit grayscale images of nine Brodatz textures, D21, D53, D77, D17, D78, D79, D55, D76, and D84—50 images for each of the nine textures. All images are preprocessed by equalizing their histograms, to ensure that correct classification cannot be done based solely on some simple global histogram characteristics. In a single experiment, each set of 50 images of a texture is partitioned into a training set of 10 images and a test set of 40 images. Two training images and two test images from Experiment 1 are shown in Fig. 3 for the textures D78 and D79. The training set is used to train an SRT model described in the previous section, using the center-surround algorithm [16]. We use $M = 5$. This is done for each class, resulting in nine SRT models. In the testing stage of the experiment, the probability of every test image given each of the nine SRT models is calculated, and the image is classified according to the highest probability. The five different experiments in Table 1 correspond to selecting five different sets of ten images from each class as the training set. Thus, a total of $10 \times 5 \times 9 = 450$ training images and $40 \times 5 \times 9 = 1800$ test images are used in the five experiments. As Table 1 shows, the correct classification rate is 100%.

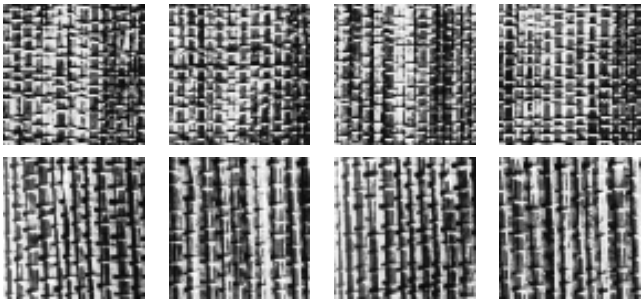


Fig. 3: An illustration of Brodatz texture images: from left to right, two training images and two correctly classified test images for the texture D78 (top row); two training images and two correctly classified test images for the texture D79 (bottom row).

	Correct classification rate	
	Training set	Test set
Experiment 1	100%	100%
Experiment 2	100%	100%
Experiment 3	100%	100%
Experiment 4	100%	100%
Experiment 5	100%	100%
Overall	100%	100%

Table 1: Correct classification percentages for the experiments with Brodatz texture data set. Each experiment uses 10 64×64 images from each of nine texture classes as training data and 40 64×64 images from each texture class as test data.

Experiments 6-10 classify natural images of houses, buildings, and store fronts. We have 50 64×64 grayscale, histogram-equalized images in each of the three categories, and use 40 of them per category as the training set and the remaining 10 as the test set, for a total of $40 \times 5 \times 3 = 600$ training images and $10 \times 5 \times 3 = 150$ test images in the five experiments. Some of these images are shown in Fig. 4. Note from Table 2 that the correct classification rate for the test images is consistently above 95%, with a total of only 4 misclassified images out of 150. This is despite the fact that each class consists of many heterogeneous images which are quite

Fig. 4: Images of houses (top row), buildings (second row), and stores (bottom row), used in Experiments 6–10.

	Correct classification rate	
	Training set	Test set
Experiment 6	100%	96.7%
Experiment 7	100%	96.7%
Experiment 8	100%	100.0%
Experiment 9	100%	96.7%
Experiment 10	100%	96.7%
Overall	100%	97.3%

Table 2: Correct classification percentages for the experiments with the house-building-store data set. Each experiment uses 40 64×64 images from each of three classes as training data and 10 64×64 images from each class as test data.

complex.

5. CONCLUSIONS.

We have constructed a novel spatial random tree image model and showed its close relationship with best basis search problems. We have illustrated our model by using it to classify natural images and images of Brodatz textures.

6. REFERENCES

[1] J. Baker, "Trainable grammars for speech recognition," *Speech Communications Papers for the 97th Meeting of the Acoust. Soc. of America* (D. Klatt and J. Wolf, eds.), 1979.

[2] M. Basseville, A. Benveniste, K. C. Chou, S. A. Golden, R. Nikoukhah, and A. S. Willsky, "Modeling and estimation of multiresolution stochastic processes," *IEEE Trans. on Information Theory*, vol. 38, no. 2, pp. 766–784, March 1992.

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

[4] 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.

[5] J.E. Hopcroft and J.D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 1979.

[6] Y. Huang, I. Pollak, and C.A. Bouman. Image compression with multitree tilings. In *Proceedings* of ICASSP, March 18-23, 2005, Philadelphia, PA.

[7] Y. Huang, I. Pollak, M.N. Do, and C.A. Bouman. Optimal tilings and best basis search in large dictionaries. In *Proc. 37th Asilomar Conference on Signals, Systems, and Computers*, Nov. 9-12, 2003, Pacific Grove, CA.

[8] Y. Huang, I. Pollak, M.N. Do, and C.A. Bouman. Fast search for best representations in multitree dictionaries. Technical Report TR-ECE-04-09, Purdue University, School of Electrical and Computer Engineering, December 2004.

[9] K. Lari and S. Young, "The estimation of stochastic context-free grammars using the inside-outside algorithm," *Computer Speech and Language*, vol. 4, pp. 35–56, 1990.

[10] C. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[11] I. Pollak, J.M. Siskind, M.P. Harper, and C.A. Bouman. Spatial random trees and the center-surround algorithm. Technical Report TR-ECE-03-03, Purdue University, School of Electrical and Computer Engineering, January 2003.

[12] I. Pollak, J. M. Siskind, M. P. Harper, and C. A. Bouman. Modeling and estimation of spatial random trees with application to image classification. In *Proc. ICASSP*, Hong Kong, 2003.

[13] I. Pollak, J. M. Siskind, M. P. Harper, and C. A. Bouman. Parameter estimation for spatial random trees using the EM algorithm. In *Proc. ICIP*, Barcelona, 2003.

[14] D. Potter, *Compositional Pattern Recognition*. PhD thesis, Brown University, 1999. `http://www.dam.brown.edu/people/dfp.`

[15] D. Sankoff, "Branching processes with terminal types: application to context-free grammars," *Journal of Applied Probability*, vol. 8, pp. 233–240, 1971.

[16] W. Wang, I. Pollak, T.-S. Wong, C.A. Bouman, and M.P. Harper. Hierarchical stochastic image grammars for classification and segmentation. Submitted to *IEEE Trans. Im. Proc.*