Hierarchical Stochastic Image Grammars for Classification and Segmentation

Wiley Wang, Ilya Pollak, *Senior Member, IEEE*, Tak-Shing Wong, Charles A. Bouman, *Fellow, IEEE*, Mary P. Harper, *Senior Member, IEEE*, and Jeffrey M. Siskind

***Abstract*—We develop a new class of hierarchical stochastic image models called spatial random trees (SRTs) which admit polynomial-complexity exact inference algorithms. Our framework of multitree dictionaries is the starting point for this construction. SRTs are stochastic hidden tree models whose leaves are associated with image data. The states at the tree nodes are random variables, and, in addition, the structure of the tree is random and is generated by a probabilistic grammar. We describe an efficient recursive algorithm for obtaining the maximum *a posteriori* estimate of both the tree structure and the tree states given an image. We also develop an efficient procedure for performing one iteration of the expectation-maximization algorithm and use it to estimate the model parameters from a set of training images. We address other inference problems arising in applications such as maximization of posterior marginals and hypothesis testing. Our models and algorithms are illustrated through several image classification and segmentation experiments, ranging from the segmentation of synthetic images to the classification of natural photographs and the segmentation of scanned documents. In each case, we show that our method substantially improves accuracy over a variety of existing methods.**

***Index Terms*—Dictionary, estimation, grammar, hierarchical model, image classification, probabilistic context-free grammar, segmentation, statistical image model, stochastic context-free grammar, tree model.**

I. Introduction

IN THIS work, we develop a new methodology for constructing hierarchical stochastic image models called spatial random trees (SRTs). Similar to [2], [10], [13]–[15], [18], [33], [36], [37], [45], [56], and [57], our models are stochastic hidden tree models whose leaf nodes are associated with image data. Our key innovation, however, is that not only the states at the nodes of the tree are random variables, but also the tree structure itself is random and is generated by a probabilistic grammar [24], [35], [47]. While grammars have been used to develop tree-structured models for one-dimensional (1-D) signals [38], the generalization to two dimensions is not direct because the leaves of a tree generated by a grammar cannot be naturally mapped to a two-dimensional (2-D) grid. We solve this problem by uniquely associating every symbol in the grammar with a 2-D region. The most important contribution of our paper is the introduction of a rich class of probability distributions over trees which allows for exact calculation of likelihoods in polynomial time. Armed with such a probability distribution, we can use Bayesian techniques to perform typical tasks of parameter estimation and classification.

We are motivated by the wide use of probabilistic context-free grammars (PCFGs) in natural language processing, for example, to model the structure of sentences [22], [38]. PCFGs are based on the concept of branching stochastic processes which have been used in studying population dynamics since 1845 [6], [31], [32], [54]. These problems have been posed in 1-D where the objects under consideration (e.g., words in a sentence) have a natural linear arrangement; or even in "0-D" where the arrangement of objects, such as molecules of different types in a population of particles, does not matter. In addition to early efforts to apply both deterministic grammars (see, e.g., [21], [46], [49], and references therein) and probabilistic grammars [21] in 2-D, probabilistic grammars have more recently been applied to such 2-D problems as optical character recognition [23], [43] and analyzing the layout of document images [30]. However, currently existing image models with stochastic tree structure—both grammar-based [23], [43] and those that do not use grammars [25], [52], [55]—do not admit computationally feasible exact inference algorithms.

The main contribution of this paper is the introduction of SRTs which are stochastic grammar models for images and other multidimensional data, and which admit polynomial complexity exact inference algorithms. The second key contribution of this paper is the development of these exact inference algorithms, and the demonstration of their use in several segmentation and classification examples. We build upon our earlier research on stochastic grammars [40]–[42], [50] and multitree dictionaries [27]–[29]. Specifically, we use the framework of multitree dictionaries as the starting point for our novel construction of SRTs. Section II reviews multitree dictionaries and the associated efficient algorithm for finding the globally optimal tree in a multitree dictionary. Section III introduces new SRT models through two examples and shows how to use the optimal tree extraction algorithm of Section II to find the maximum *a posteriori* (MAP) estimate of both the tree structure and the tree states for an SRT model. These examples

Manuscript received April 27, 2005; revised September 26, 2005. This work was supported in part by the National Science Foundation (NSF) under Grant IIS-0329156 and CAREER Award CCR-0093105; in part by the Xerox Foundation; and in part by ARDA under Contract MDA904-03-C-1788. Part of this work was carried out while M. P. Harper was on leave from the NSF. Part of this work was presented at the 2005 IEEE Statistical Signal Processing Workshop, the SPIE Optics and Photonics 2005 Symposium, and the 2006 IS&T/SPIE Symposium on Electronic Imaging. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the view of NSF. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mario A. T. (G. E.) Figueiredo.

The authors are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: ipollak@ecn.purdue.edu).

Digital Object Identifier 10.1109/TIP.2006.877496

are extended in Section IV, where we present our SRT framework in its full generality. Section V is primarily devoted to the maximum likelihood (ML) parameter estimation for SRTs. We introduce the center-surround algorithm which is an efficient algorithm for computing the expectation-maximization (EM) [3] updates. The center-surround algorithm is a generalization of the forward-backward algorithm [44] and the inside-outside algorithm [1], [34], [38]. Two other inference problems are also discussed in Section V: finding the maximum posterior marginal (MPM) estimates of the leaf states, and choosing the most probable model within a Bayesian hypothesis testing framework.

Section VI illustrates our models and algorithms through several image classification and segmentation experiments. We demonstrate the utility of our tree models by comparing the performance of several classifiers on a database of 150 natural images. The classifier based on an SRT model achieves over 97% correct classification rate; whereas, classifiers which do not use the hidden tree structure perform very poorly. We also use our model to classify Brodatz textures. In this task, we reduce the misclassification rate by a factor of 4.5 as compared to a recent algorithm [12]. We use synthetic images from [10] to show the improved segmentation performance of our algorithm, as compared to a quadtree-based algorithm called SMAP [10]. Finally, we demonstrate the potential of our models for document image processing, by segmenting a document image into regions and classifying each region. The ability to do this robustly is important for various tasks such as document image compression and analysis, and database organization and search.

II. Multitree Dictionaries

We first review the framework of multitree dictionaries [27]–[29] which we will use to develop our SRT image models. Multitree dictionaries are defined using grammar formalism [26], [38]. 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 $a \to \alpha$ 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 $a \to \alpha$ is depicted as a node labeled a whose children are labeled with the elements of α. Following [27]–[29], 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.

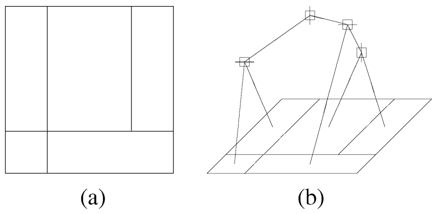


Fig. 1. Illustration of rectangular image tilings: (a) a tiling; (b) a corresponding tree of splits.

Suppose that each symbol $u \in A$ is assigned a cost $c(u)$, and that each production $u \to \alpha$ is assigned a cost $\bar{c}(u \to \alpha)$. Further assume that $\textsc{cost}(t)$ 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

$$\textsc{cost}(t) = \sum_{(u\to\alpha)\in t} \bar{c}(u \to \alpha) + \sum_{u \in \text{leaves}(t)} c(u). \qquad (1)$$

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

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

We denote the corresponding cost by C_a^*, i.e., $C_a^* = \textsc{cost}(t_a^*)$. This problem can be solved using an efficient recursive algorithm for best tree search described in [28], [29]. The pseudocode for this algorithm is reproduced in Fig. 2. To illustrate this algorithm, we suppose that the only allowed split of the symbol a is $a \to \{b_1, b_2\}$. There is a tree $\{a\}$ in the multitree dictionary $\mathcal{D}_a(G)$ which consists of one node labeled a, with $\textsc{cost}(\{a\}) = c(a)$. 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 $\textsc{cost}(t) = \bar{c}(a \to \{b_1, b_2\}) + \textsc{cost}(t_{left}) + \textsc{cost}(t_{right})$. Consequently, the optimal tree is

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

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 $a \to \{b_1, b_2\}$, with the cost of the tree $\{a\}$. We have a similar recursion in the general case, as shown in Fig. 2 and described in [29].

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 $O(\text{number of all productions})$ which, in many applications,

```
$(C_a^*, s_a^*) \leftarrow$ best_production($a$) {
    if $C_a^*$ has been computed
        get $C_a^*$ and $s_a^*$ from the global data structure TABLE;
    else {
        $s_a^* \leftarrow \varnothing$;            //Initialize $s_a^*$
        $C_a^* \leftarrow c(a)$;           //Initialize $C_a^*$
        for $\alpha \in S(a)$ {
            for $b \in \alpha$
                $(C_b^*, s_b^*) \leftarrow$ best_production($b$);
            if $\bar{c}(a \to \alpha) + \sum_{b \in \alpha} C_b^* < C_a^*$ {
                $s_a^* \leftarrow \alpha$;
                $C_a^* \leftarrow \bar{c}(a \to \alpha) + \sum_{b \in \alpha} C_b^*$;
            }
        }
        record $C_a^*$ and $s_a^*$ in the global data structure TABLE;
    }
    return $C_a^*$ and $s_a^*$;
}
```

(a) Recursive calculation of best productions and costs.

```
$t_a^* \leftarrow$ best_tree($a$) {
    get $s_a^*$ from the global data structure TABLE;
    if $s_a^*$ is the empty set
        $t_a^* \leftarrow \{a\}$;
    else {
        $k \leftarrow 0$;
        for $b \in s_a^*$ {
            $k \leftarrow k + 1$;
            $b_k \leftarrow b$;
            $t_{b_k}^* \leftarrow$ best_tree($b_k$);
        }
                               $a$
                         /      |      \
        $t_a^* \leftarrow$     $t_{b_1}^*$    ...    $t_{b_k}^*$   ;
    }
    return $t_a^*$;
}
```

(b) Recursive generation of best tree.

Fig. 2. Pseudocode for the calculation of the best productions and costs, and for the generation of the globally optimal tree.

can be made linear or polynomial in the number of symbols. As we will see in Sections III and IV, judicious choice of A and S can make the complexity of the search algorithm polynomial or even linear in the number of data points.

III. Spatial Random Trees: Two Example Models

There are a variety of ways to adapt the framework of multitree dictionaries to probabilistic modeling. We now give two examples where we use this framework to develop probabilistic models for images defined on a fixed domain.

A. SRT Construction 1

In our first example, we suppose that the discrete image domain Q is a $2^L \times 2^L$ square where L is a fixed integer.[1] We define the following grammar $G = (A, S)$ to describe hierarchical segmentations of the domain Q. Following the conventions used in context-free grammars [38], 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 $a = (0, Q)$.
- The nonterminal symbols have the form $a = (j, R)$ 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 $\{1, \ldots, J\}$ which can represent, for example, the classification of the image pixels belonging to R into one of J classes.

[1]The use of a fixed finite domain reflects that, in any practical application, there exists a largest image domain of interest.

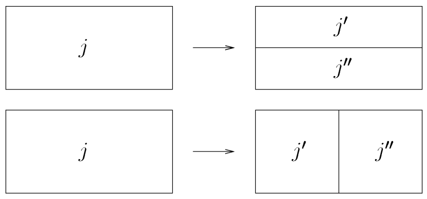


Fig. 3. Nonroot productions for Construction 1. (Top) Horizontal split into two congruent rectangles. (Bottom) Vertical split into two congruent rectangles.

- The terminal symbols have the form $a = (j, \{n\})$ 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.

- The root productions are $(0, Q) \to \{(j, Q)\}$ for $j = 1, \ldots, J$.
- The remaining productions are defined for all nonterminal symbols $a = (j, R)$. These productions are $(j, R) \to \{(j', R'), (j'', R'')\}$, for all $j, j', j'' = 1, \ldots, J$ and for all partitions of R into two congruent rectangles R' and R'',[2] as illustrated in Fig. 3.

For each symbol $a = (j, R)$, 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 a, we specify a probability distribution $\bar{\mathbf{p}}_a$ on $S(a)$ (recall that $S(a)$ is the set of all righthand sides of the productions starting with

[2]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.

"$a \quad \to$"): $\sum_{\alpha \in S(a)} \bar{\mathbf{p}}_a(\alpha) = 1$, and let the cost of each production be the corresponding negative log-probability, $\bar{c}(a \to \alpha) = -\log \bar{\mathbf{p}}_a(\alpha)$. Our observation model is as follows. For $j = 1, \ldots, J$, we let $\mathbf{p}_j$ be a probability distribution over the set of all valid pixel values. If pixels are modeled as continuous valued, a typical choice would be a Gaussian density, $\mathbf{p}_j(x) = (1/\sqrt{2\pi}\sigma_j)e^{-(x-\mu_j)^2/2\sigma_j^2}$, where μ_j and σ_j are parameters which depend on j. If pixels are modeled as discrete valued, a typical choice would be a probability mass function $\mathbf{p}_j(r)$ over all grayscale values $r = 0, \ldots, 255$. Our model can, thus, be viewed as a generative process which first uses the productions of Fig. 3 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 shown in the equation at the bottom of the page. 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 (or probability density).

We define the joint probability distribution of a tree t and the data f as the product of the probabilities of the productions in the tree and the conditional probability distributions of the observations

$$\text{JOINT-PROB}(t, f) \triangleq \prod_{(a \to \alpha) \in t} \bar{\mathbf{p}}_a(\alpha) \prod_{(j,\{n\}) \in \text{leaves}(t)} \mathbf{p}_j(f_n).$$

Using an argument described in [40], it can be shown that we have defined a legitimate probability distribution on the set of all pairs (t, f), i.e., that[3]

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

where $\mathcal{D}$ is the set of all trees generated by our grammar. This motivates defining the following probability distribution over images supported on Q:

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

Note that $\text{COST}(t)$ defined in (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 MAP tree given an image is equivalent to minimizing $\text{COST}(t)$ and can be solved using the recursion of Fig. 2 described in the previous section. Moreover, the probability (or 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 Section V, this can be used with the EM algorithm [3] to estimate the parameters of the model, i.e., the production probabilities $\bar{\mathbf{p}}_a(\alpha)$ 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 $a = (j, R)$ and $a' = (j, R')$, where R and R' are congruent, then $\bar{\mathbf{p}}_a = \bar{\mathbf{p}}_{a'}$. This is further discussed in Section V, in a more general setting.

We call such probabilistic models *spatial random trees* to emphasize the fact that the underlying grammar[4] models spatial organization. We note that, in addition to defining a probability distribution over all images supported on Q, (3) induces a probability distribution over all images supported on any fixed subset of the domain Q: Such a distribution is obtained via marginalization.

We now derive the computational complexity of the MAP tree extraction algorithm for this construction, shown in Table I. Since every rectangle can only be split into two congruent rectangles, the total number of valid rectangles is $O(|Q|)$. Therefore, the total number of nonterminal symbols (j, R) is $O(J|Q|)$. For each nonterminal symbol (j, R), the optimal cost is computed by searching over $2J^2$ possibilities: splitting R either horizontally or vertically and assigning one of J states to each of the two children. Therefore, the total time complexity of the search is $O(J^3|Q|)$. For each symbol, $O(1)$ numbers need to be stored: the location of the optimal split, the optimal cost, and the optimal states of the two children. Therefore, the memory complexity of the search is $O(J|Q|)$.

TABLE I
COMPUTATIONAL COMPLEXITY FOR THE MAP TREE EXTRACTION ALGORITHMS FOR OUR MODELS, WHERE J IS THE NUMBER OF STATES, $|Q|$ IS THE TOTAL NUMBER OF PIXELS IN THE IMAGE DOMAIN, AND $|A|$ IS THE TOTAL NUMBER OF SYMBOLS (j, R)

	Time complexity	Space complexity
Construction 1	$O(J^3\|Q\|)$	$O(J\|Q\|)$
Construction 2	$O(J^3\|Q\|^{2.5})$	$O(J\|Q\|^2)$
General construction	O(number of productions)	$O(\|A\|)$

[3]Equation (2) assumes $\mathbb{R}$-valued images; for eight-bit grayscale images, the integral is replaced with a summation over $\{0, \ldots, 255\}^{|Q|}$.

[4]Since grammars generate 1-D *strings* of symbols [26] while SRTs do not, this is a slight abuse of the standard language-theoretic terminology. The use of this terminology is appropriate, however, since the structure of our construction is similar to that of a PCFG.

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

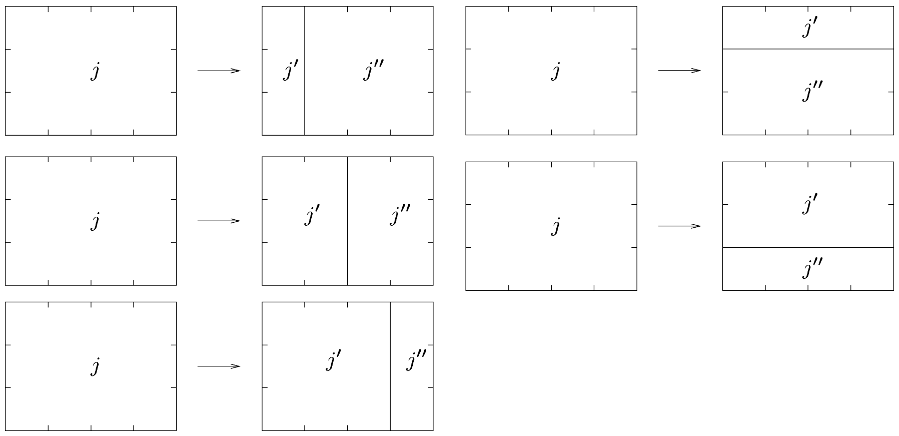


Fig. 4. Nonterminal productions for Construction 2, for a 3 × 4 rectangle.

B. SRT Construction 2

In our second example of SRT construction, we suppose that the image domain Q is an arbitrary discrete rectangle and specify the following grammar $G = (A, S)$ for describing rectangular tilings of Q. First, we partition the set $\{1, \ldots, J\}$ into two subsets: the terminal states $\mathcal{J}_{\text{term}}$ and the nonterminal states $\mathcal{J}_{\text{nonterm}}$. The terminal states can only label the leaf regions, and the nonterminal states can only label the nonleaf regions. We then proceed to define the symbols and productions as summarized in table shown at the bottom of the page.

There are four differences from Construction 1 here: We let the set of all valid regions $\mathcal{R}$ be the set of all subrectangles of Q rather than just dyadic subrectangles; we allow arbitrary splits of every rectangle into two subrectangles rather than only splitting in the middle (see Fig. 4 and compare with Fig. 3); we allow arbitrary subrectangles to occur at the leaves rather than having leaves be individual pixels—and, because of this, we have a separate set of terminal states which indicate when the recursive splitting terminates (in Construction 1, a branch of the splitting process terminated whenever a 1×1 rectangle was reached).

For every root or nonterminal symbol a, we specify a probability distribution $\bar{\mathbf{p}}_a$ on $S(a)$; for every terminal symbol a, we specify a probability distribution $\tilde{\mathbf{p}}_a$ over the valid pixel values (again, a typical choice is a Gaussian with mean μ_a and variance σ_a^2 for $\mathbb{R}$-valued images or a discrete probability mass function over $\{0, \ldots, 255\}$ for eight-bit grayscale images). We define the joint probability distribution of a tree $t \in \mathcal{D}$ and an image f as follows:

$$\text{JOINT-PROB}(t, f) \triangleq \prod_{(a \to \alpha) \in t} \bar{\mathbf{p}}_a(\alpha) \prod_{(j,R) \in \text{leaves}(t)} \prod_{n \in R} \tilde{\mathbf{p}}_{(j,R)}(f_n).$$

The normalization (2) still holds, and we use (3) to define the probability distribution over images. We define the cost of each production as $\bar{c}(a \to \alpha) = -\log \bar{\mathbf{p}}_a(\alpha)$, and the cost of each leaf as shown in the equation at the bottom of the page. Using these definitions in (1), we have $\text{COST}(t) = -\log \text{JOINT-PROB}(t, f)$, and, therefore, finding the MAP tree given an image again amounts to minimizing $\text{COST}(t)$ and can be accomplished through the algorithm of Fig. 2 and [28], [29]. Note that, since in our present construction

root symbol	$(0, Q)$
nonterminals	(j, R) where $j \in \mathcal{J}_{\text{nonterm}}$ and R is an arbitrary subrectangle of Q with $\lvert R \rvert > 1$.
terminals	(j, R) where $j \in \mathcal{J}_{\text{term}}$ and R is an arbitrary subrectangle of Q.
root productions	$(0, Q) \to \{(j, Q)\}$, for all $j \in \{1, \ldots, J\}$.
other productions	$(j, R) \to \{(j', R'), (j'', R'')\}$ where $j \in \mathcal{J}_{\text{nonterm}}$; $j', j'' \in \{1, \ldots, J\}$; and R' and R'' are any two disjoint rectangles that partition the rectangle R (see Fig. 4).

$$c(a) \triangleq \begin{cases} \infty, & \text{if } a \text{ is the root symbol or a nonterminal symbol} \\ -\sum_{n \in R} \log \tilde{\mathbf{p}}_{(j,R)}(f_n), & \text{if } a = (j, R) \text{ is a terminal symbol} \end{cases}$$

the leaves can be rectangles of any size, in this case, our search algorithm in fact finds the best rectangular tiling of an image.

The total number of valid rectangles R is now $O(|Q|^2)$. Therefore, via reasoning similar to Section III-A, we infer that the time complexity of the MAP tree search for Construction 2 is $O(J^3|Q|^{2.5})$ and the memory complexity is $O(J|Q|^2)$, as shown in Table I.

As in Construction 1, we reduce the number of parameters by imposing additional constraints. Specifically, for any three states j, j', j'', we require that the probabilities of all productions of the form $(j,R) \to \{(j',R'),(j'',R'')\}$ depend on only one parameter. We describe in Section V how these parameters, as well as the parameters of the observation distributions $\tilde{\mathbf{p}}_a$, can be estimated from the image data.

IV. Spatial Random Trees: General Formulation

We now generalize our Constructions 1 and 2. This will allow us to develop a single set of inference algorithms which specialize to both constructions, rather than use two separate sets of formulas. In addition, the model we presently introduce is more general than either Construction 1 or 2 and provides a framework for designing other SRT models.

We generalize our Constructions 1 and 2 to images $f: Q \to X$ defined on an arbitrary finite set of points Q and taking values in some set X. The only requirement we impose on the set X is that a probability distribution can be defined on it. For example, X can be $\mathbb{R}$ or $\{0,\ldots,255\}$. We also introduce two other extensions. First, we allow the set $\mathcal{R}$ of valid regions to be an arbitrary collection of subsets of Q, not necessarily rectangles. Second, we allow partitions of a region into an arbitrary number of regions, not necessarily two. In practice, allowing $\mathcal{R}$ to include every subset of Q and allowing every region to be partitioned arbitrarily would make the algorithm of Fig. 2 intractable. Therefore, both $\mathcal{R}$ and the set of valid partitions need to be carefully chosen. This choice is application-dependent.

Given a domain Q, the set $\mathcal{R}$ of valid regions, and a positive integer J, we now specify a grammar $G=(A,S)$ and the corresponding multitree dictionary $\mathcal{D}$. As previously, we use the root symbol $a=(0,Q)$. The remaining symbols are of the form (j,R) where $j \in \{1,\ldots,J\}$ and $R \in \mathcal{R}$. We incorporate the concept of tree termination in our dictionary by separating these symbols into two sets: the set A_{term} of terminals which can only appear at the leaves of trees, and the set A_{nonterm} of nonterminals which can only appear at the internal nodes of trees. For example, in Construction 1, A_{term} consists of all symbols (j,R) where R is a 1×1 rectangle; in Construction 2, A_{term} consists of all symbols (j,R) where R is an arbitrary rectangle and j is an element of a designated set $\mathcal{J}_{\text{term}}$ of terminal states. By definition, there are no productions which start with a terminal symbol, i.e., $S(a)=\varnothing$ for any $a \in A_{\text{term}}$. For the root symbol and nonterminal symbols, the productions are specified in the table at the bottom of the page.

While there are various ways of defining the probability distribution $\text{PRIOR-PROB}(t)$ over the dictionary $\mathcal{D}$ which lead to tractable computations, we focus here on the method used in Constructions 1 and 2 and borrowed from the literature on branching processes [31], [32] and stochastic context-free languages [22], [24], [35], [38], [47]: Namely, for every root or nonterminal symbol a, we define a probability distribution $\bar{\mathbf{p}}_a$ on $S(a)$, and let the probability of a tree t be the product of all its production probabilities

$$\text{PRIOR-PROB}(t) \triangleq \prod_{(a\to\alpha)\in t} \bar{\mathbf{p}}_a(\alpha).$$

Our strategy for specifying a probability distribution on images then consists of the following steps.

1. For every terminal $a=(j,R)$, specify a probability distribution $\mathbf{p}_{(j,R)}(x)$ over the set $X^{|R|}$ of all images supported on R.

2. For any tree $t \in \mathcal{D}$, specify the conditional distribution of an image f given the tree t as follows:

$$\text{OBSERVATION-PROB}(f|t) \triangleq \prod_{(j,R)\in \text{leaves}(t)} \mathbf{p}_{(j,R)}(f_R) \quad (4)$$

where f_R is the restriction of the image f to the region R.

3. Specify a probability distribution $\text{PRIOR-PROB}(t)$ over $\mathcal{D}$, by specifying $\bar{\mathbf{p}}_a(\alpha)$.

4. Set, for any $t \in \mathcal{D}$ and image f

$$\text{JOINT-PROB}(t,f) \triangleq \text{PRIOR-PROB}(t)\cdot\text{OBSERVATION-PROB}(f|t) \quad (5)$$
$$= \prod_{(a\to\alpha)\in t} \bar{\mathbf{p}}_a(\alpha) \prod_{(j,R)\in \text{leaves}(t)} \mathbf{p}_{(j,R)}(f_R). \quad (6)$$

root productions	$(0,Q) \to \{(j,Q)\}$ for all $j \in \{1,\ldots,J\}$.
other productions	$a \to \alpha$ for all nonterminals $a=(j,R)$ and all finite sets $\alpha = \{(j_1,R_1),\ldots,(j_{\vert\alpha\vert},R_{\vert\alpha\vert})\}$ such that : • $(j_k,R_k) \in A_{\text{term}} \cup A_{\text{nonterm}}$; • $R_1,\ldots,R_{\vert\alpha\vert}$ are disjoint sets that partition R.

.

5. Define

$$\text{IMAGE-PROB}(f) \triangleq \sum_{t\in\mathcal{D}} \text{JOINT-PROB}(t,f). \tag{7}$$

The posterior probability is then

$$\text{POSTERIOR-PROB}(t|f) = \frac{\text{JOINT-PROB}(t,f)}{\text{IMAGE-PROB}(f)}.$$

Maximizing this for a fixed image f is equivalent to maximizing $\text{JOINT-PROB}(t,f)$. In general, this involves an exhaustive search over all trees $t\in\mathcal{D}$, which is typically not computationally feasible. However, when $\text{PRIOR-PROB}(t)$ can be factored out to reduce $-\log \text{JOINT-PROB}(t,f)$ to the form (1), then maximizing the posterior probability can be efficiently done using the recursive algorithm of Fig. 2. This is the case for Constructions 1 and 2, and also for our general construction, as evident from (6). Specifically, substituting the definitions given at the bottom of the page into (1) results in $\text{COST}(t) = -\log \text{JOINT-PROB}(t,f)$. In this case, the computational complexity is not $O(|\mathcal{D}|)$ but is instead only $O(\text{number of all productions})$ which is drastically smaller in typical applications, and can be controlled by judiciously designing the set $\mathcal{R}$ of valid regions and the set $S(a)$ of valid splits for every symbol $a=(j,R)$, as demonstrated in our Constructions 1 and 2. In addition, the number J of possible states j influences the computational complexity, as shown in Table I. The choice of J is often dictated by the problem specification, as in our segmentation examples in Section VI below; or, alternatively, J can be estimated using model selection techniques similar to [5], [9], [20], [51].

V. Parameter Estimation and Other Inference Problems

A. Supervised Learning

Given I independent observations of images $f^{(1)},\ldots,f^{(I)}$ and corresponding trees $t^{(1)},\ldots,t^{(I)}\in\mathcal{D}$, we can estimate the parameters of the observation distributions $\mathbf{p}_a(x)$ and the probabilities $\bar{\mathbf{p}}_a(\alpha)$ of all productions $a\to\alpha$ by maximizing the following log-likelihood function, obtained by multiplying together the joint distributions of (6) for all i and taking the log

$$L_{\text{supervised}} = \sum_{i=1}^{I}\left[\sum_{(a\to\alpha)\in t^{(i)}} \log\bar{\mathbf{p}}_a(\alpha) + \sum_{(j,R)\in \text{leaves}(t^{(i)})} \log \mathbf{p}_{(j,R)}(f_R^{(i)})\right].$$

Suppose first that $I=1$. In this case, if no constraints are imposed on the production probabilities, the maximum likelihood estimate $\hat{\bar{\mathbf{p}}}_a(\alpha)$ will be 1 if the production $a\to\alpha$ appears in the tree $t^{(1)}$, 0 if the nonterminal a appears in the tree but the production $a\to\alpha$ does not, and arbitrary[5] if the nonterminal a does not appear in the tree. This happens because, in our model, each production can occur at most once in a single tree. This means that either a large amount of data (i.e., a large I) is needed to obtain meaningful estimates of the production probabilities, or some additional constraints need to be imposed to reduce the number of independent parameters. We typically do the latter by grouping productions into several equivalence classes, and imposing that the probabilities of all productions within the same equivalence class depend on a single parameter. We induce these equivalence classes by first defining equivalences between symbols (denoted $a\sim a'$), and then declaring two productions $a\to\{a_1,\ldots,a_{|\alpha|}\}$ and $a'\to\{a'_1,\ldots,a'_{|\alpha|}\}$ equivalent if $a\sim a', a_1\sim a'_1,\ldots,a_{|\alpha|}\sim a'_{|\alpha|}$. We let $[a]$, $[\alpha]$, and $[a\to\alpha]$ denote the equivalence classes of a symbol a, a set of symbols α, and a production $a\to\alpha$, respectively. We let $\rho_a([\alpha])$ be the conditional probability of a transition from the symbol a into the equivalence class $[\alpha]$—in other words

$$\rho_a([\alpha]) = \sum_{\alpha'\colon\, \alpha'\in S(a) \text{ and } \alpha'\in[\alpha]} \bar{\mathbf{p}}_a(\alpha'). \tag{8}$$

We, moreover, impose that, if $a\sim a'$, then $\rho_a([\alpha])=\rho_{a'}([\alpha])$, and we denote this common probability by $\rho_{[a]}([\alpha])$. We require each individual production probability $\bar{\mathbf{p}}_a(\alpha)$ to be linearly related to the probability $\rho_{[a]}([\alpha])$

$$\bar{\mathbf{p}}_a(\alpha) = g(a,\alpha)\cdot\rho_{[a]}([\alpha]). \tag{9}$$

The coefficients $g(a,\alpha)$ are application-dependent but fixed. They must be nonnegative and satisfy the following constraint:

$$\sum_{\alpha'\colon\, \alpha'\in S(a)\text{ and } \alpha'\in[\alpha]} g(a,\alpha') = 1$$

[5]As long as $\hat{\bar{\mathbf{p}}}_a$ is a legitimate probability distribution.

$$\bar{c}(a\to\alpha) = -\log\bar{\mathbf{p}}_a(\alpha), \text{for every production } a\to\alpha$$
$$c(a) = \begin{cases} \infty, & \text{if } a \text{ is the root symbol or a nonterminal symbol} \\ -\log\mathbf{p}_{(j,R)}(f_R), & \text{if } a=(j,R) \text{ is a terminal symbol} \end{cases}$$

in order to enforce (8). By differentiating $L_{\text{supervised}}$ and using Lagrange multipliers to insure that probabilities sum to one, we find that the maximum likelihood estimate of $\rho_{[a]}([\alpha])$ is

$$\hat{\rho}_{[a]}([\alpha]) = \frac{\sum_{i=1}^{I} h_{[a\rightarrow\alpha]}(t^{(i)})}{\sum_{i=1}^{I} h_{[a]}(t^{(i)})} \tag{10}$$

where $h_{[a\rightarrow\alpha]}(t^{(i)})$ is the number of occurrences of $[a \rightarrow \alpha]$ in the tree $t^{(i)}$, and $h_{[a]}(t^{(i)})$ is the number of occurrences of $[a]$ in the tree $t^{(i)}$. Once $\rho_{[a]}([\alpha])$ has been estimated, the estimates of the individual production probabilities are determined through (9).

For example, in Construction 1 we let $(j,R) \sim (j',R')$ if $j = j'$ and R and R' are congruent, and let $g(a,\alpha) = 1$. In Construction 2, we let $(j,R) \sim (j',R')$ if $j = j'$. Then, one possible assignment of the coefficients is $g(a',\alpha') = 1/(m+n-2)$, for any $a' = (j,R) \in [a]$ where R is an $m \times n$ rectangle, and any $\alpha' \in S(a')$. In this case, $\rho_{[(j,R)]}([\{(j',R'),(j'',R'')\}])$ is the conditional probability of transitioning from any rectangle labeled j to any two rectangles labeled j' and j''. The factor of $1/(m+n-2)$ puts a uniform distribution on the position of the split, since there are $m+n-2$ ways of splitting an $m \times n$ rectangle into two subrectangles.

We now address parameter estimation for the observation model. We first assume the following conditionally independent Gaussian model for the observations:

$$\mathbf{p}_{(j,R)}(f_R) = \prod_{n\in R} \tilde{\mathbf{p}}_{(j,R)}(f_n) \tag{11}$$

where $\tilde{\mathbf{p}}_{(j,R)}$ is a multivariate Gaussian density with mean vector μ_a and covariance matrix Λ_a. Thus, we can model vector-valued observations which could correspond, for example, to color intensities or feature vectors. We also assume that the terminals are partitioned into equivalence classes, with $\tilde{\mathbf{p}}_a = \tilde{\mathbf{p}}_{a'}$ for any $a \sim a'$. Maximizing the log-likelihood function with respect to μ_a and Λ_a results in the following formulas, where $[a] \cap \text{leaves}(t^{(i)})$ is the set of all leaves of the tree $t^{(i)}$ which belong to the equivalence class $[a]$

$$\hat{\mu}_{[a]} = \frac{\sum_{i=1}^{I} \sum_{(j',R')\in[a]\cap\text{leaves}(t^{(i)})} \sum_{n\in R'} f_n^{(i)}}{\sum_{i=1}^{I} \sum_{(j',R')\in[a]\cap\text{leaves}(t^{(i)})} |R'|} \tag{12}$$

$$\hat{\Lambda}_{[a]} = \frac{\sum_{i=1}^{I} \sum_{(j',R')\in[a]\cap\text{leaves}(t^{(i)})} \sum_{n\in R'} (f_n^{(i)} - \hat{\mu}_{[a]})(f_n^{(i)} - \hat{\mu}_{[a]})^\top}{\sum_{i=1}^{I} \sum_{(j',R')\in[a]\cap\text{leaves}(t^{(i)})} |R'|}. \tag{13}$$

Similar formulas can be derived for other commonly used parametric families of observation models. For example, if $\tilde{\mathbf{p}}_{(j,R)}$ in (11) is a probability mass function over $\{0,1,\ldots,255\}$, as in our experiment in Section VI-C below, then its maximum likelihood estimate is the normalized histogram

$$\hat{\tilde{\mathbf{p}}}_{[a]}(r) = \frac{h_{[a],r}}{\sum_{r'=0}^{255} h_{[a],r'}} \tag{14}$$

where $h_{[a],r}$ is the number of pixels with value $r \in \{0,\ldots,255\}$ in all the leaf regions in the training data which are in the equivalence class $[a]$.

This parameter estimation scenario, where both images and corresponding trees are given, is called supervised learning [22]. Unsupervised learning—addressed in the next subsection—is the situation where the parameters are estimated from an image or a set of images alone, without the corresponding trees.

B. Unsupervised Learning

We are now given I images $f^{(1)},\ldots,f^{(I)}$ which are the observations of I independent random fields which we denote $F^{(1)},\ldots,F^{(I)}$. We denote the corresponding random trees by $T^{(1)},\ldots,T^{(I)}$. These trees are not observed. Our goal is again to estimate the parameters of the observation distributions and the probabilities of all productions by maximizing the new log-likelihood function $L_{\text{unsupervised}}$, obtained by multiplying together the image probability densities of (7) for all i and taking the log

$$\begin{aligned} &L_{\text{unsupervised}} \\ &= \sum_{i=1}^{I} \log \text{IMAGE-PROB}(f^{(i)}) \\ &= \sum_{i=1}^{I} \log \sum_{t\in\mathcal{D}} \prod_{(a\rightarrow\alpha)\in t} \bar{\mathbf{p}}_a(\alpha) \prod_{(j,R)\in\text{leaves}(t)} \mathbf{p}_{(j,R)}(f_R^{(i)}). \end{aligned}$$

The maximization must be performed subject to the normalization constraints on the production probability distributions and the observation probability distributions. Differentiating $L_{\text{unsupervised}}$ and using Lagrange multipliers yields equations similar to (10), (12), and (13) [see (15)–(17), shown at the bottom of the next page], where we are again assuming a Gaussian observation model. The expectations in these formulas, of course, themselves depend on the parameters which we are trying to estimate. The EM algorithm [3] is an iterative strategy which starts with some initial assignment of the parameters, and during each step calculates the updated values of the parameters by evaluating the righthand sides of (15)–(17) using the current values of the parameters. As shown in [3], this procedure converges to a local maximum of $L_{\text{unsupervised}}$. We prove in the Appendix that the parameter updates can be computed using (18)–(21), shown at the bottom of the next page, and corresponding to (15), (9), (16), and (17), respectively. As shown in the Appendix, the center variables for each image are given by the recursive formula (22), shown at the bottom of the next page. A recursive formula for calculating the surround variables is also derived in the Appendix. To introduce this formula, we need the following additional piece of notation. We let $P(a)$ be the set of all productions whose righthand side contains a

$$P(a) \triangleq \{d \to \alpha : a \in \alpha\}.$$

The surround variables for each image are then given by recursive formula (23), shown at the bottom of the next page. The center and surround formulas can be efficiently implemented as recursive subroutines. Note that the center variables must be computed and stored first, since they are used in the computation of the surround variables. Note also that, as we show in the Appendix

$$\text{IMAGE-PROB}(f^{(i)}) = \text{CENTER}^{(i)}_{(0,Q)}. \tag{24}$$

We call the parameter update algorithm of (18)–(23) the *center-surround algorithm.* It is a generalization of the forward-backward [44] and inside-outside algorithms [1], [34], [38] which implement the EM updates for hidden Markov models and standard 1-D probabilistic context-free grammars, respectively. The complexity of computing all the center and surround variables can be shown to be the same as that of computing the MAP tree. Formulas similar to (20) and (21) can be derived for other commonly used parametric families of observations models.

C. Other Inference Problems

Once the model parameters are estimated from the training data or otherwise specified, we use SRTs for several classifica-

$$\hat{\rho}_{[a]}([\alpha]) = \frac{\sum_{i=1}^{I} E[h_{[a\to\alpha]}(T^{(i)})|F^{(i)} = f^{(i)}]}{\sum_{i=1}^{I} E[h_{[a]}(T^{(i)})|F^{(i)} = f^{(i)}]} \tag{15}$$

$$\hat{\mu}_{[a]} = \frac{\sum_{i=1}^{I} E\left[\sum_{(j',R')\in[a]\cap\text{leaves}(T^{(i)})} \sum_{n\in R'} F_n^{(i)} \middle| F^{(i)} = f^{(i)}\right]}{\sum_{i=1}^{I} E\left[\sum_{(j',R')\in[a]\cap\text{leaves}(T^{(i)})} |R'| \middle| F^{(i)} = f^{(i)}\right]} \tag{16}$$

$$\hat{\Lambda}_{[a]} = \frac{\sum_{i=1}^{I} E\left[\sum_{(j',R')\in[a]\cap\text{leaves}(T^{(i)})} \sum_{n\in R'} (F_n^{(i)} - \hat{\mu}_{[a]})(F_n^{(i)} - \hat{\mu}_{[a]})^\top \middle| F^{(i)} = f^{(i)}\right]}{\sum_{i=1}^{I} E\left[\sum_{(j',R')\in[a]\cap\text{leaves}(T^{(i)})} |R'| \middle| F^{(i)} = f^{(i)}\right]} \tag{17}$$

$$\rho^{\text{updated}}_{[a]}([\alpha]) = \frac{\sum_{i=1}^{I} \frac{\sum_{(a'\to\alpha')\in[a\to\alpha]} \bar{\mathbf{p}}_{a'}(\alpha')\text{SURROUND}^{(i)}_{a'} \prod_{b\in\alpha'} \text{CENTER}^{(i)}_{b}}{\text{IMAGE-PROB}(f^{(i)})}}{\sum_{i=1}^{I} \frac{\sum_{a'\in[a]} \text{SURROUND}^{(i)}_{a'}\text{CENTER}^{(i)}_{a'}}{\text{IMAGE-PROB}(f^{(i)})}} \tag{18}$$

$$\bar{\mathbf{p}}^{\text{updated}}_{a}(\alpha) = g(a,\alpha)\cdot \rho^{\text{updated}}_{[a]}([\alpha]), \tag{19}$$

$$\mu^{\text{updated}}_{[a]} = \frac{\sum_{i=1}^{I} \sum_{a'=(j',R')\in[a]} \sum_{n\in R'} f_n^{(i)} \frac{\text{SURROUND}^{(i)}_{a'}\text{CENTER}^{(i)}_{a'}}{\text{IMAGE-PROB}(f^{(i)})}}{\sum_{i=1}^{I} \sum_{a'=(j',R')\in[a]} |R'| \frac{\text{SURROUND}^{(i)}_{a'}\text{CENTER}^{(i)}_{a'}}{\text{IMAGE-PROB}(f^{(i)})}} \tag{20}$$

$$\Lambda^{\text{updated}}_{[a]} = \frac{\sum_{i=1}^{I} \sum_{a'=(j',R')\in[a]} \sum_{n\in R'} (f_n^{(i)} - \mu^{\text{updated}}_{[a]})(f_n^{(i)} - \mu^{\text{updated}}_{[a]})^\top \ \frac{\text{SURROUND}^{(i)}_{a'}\text{CENTER}^{(i)}_{a'}}{\text{IMAGE-PROB}(f^{(i)})}}{\sum_{i=1}^{I} \sum_{a'=(j',R')\in[a]} |R'| \frac{\text{SURROUND}^{(i)}_{a'}\text{CENTER}^{(i)}_{a'}}{\text{IMAGE-PROB}(f^{(i)})}} \tag{21}$$

tion and segmentation tasks in Section VI. In these tasks, it is necessary to solve three types of inference problems: MAP tree estimation, ML hypothesis testing, and MPM estimation of the leaf states. We have already described the algorithm for MAP tree estimation; we now briefly describe the other two inference problems and their solutions.

Given K SRT models, one for each of K classes of images, we can classify an image f by choosing the most probable class given f, i.e., by solving the following hypothesis testing problem:

$$k^* = \arg\max_{1\le k\le K} \text{PROB}(\text{image} f \text{ belongs to class } k|f).$$

This is achieved through evaluating the likelihood of the kth SRT model using (24), multiplying it by the prior probability of the kth hypothesis, and maximizing over k. In all our examples in Section VI, the prior is uniform (i.e., the probability of every hypothesis is $1/K$). This corresponds to choosing the maximum likelihood hypothesis

$$k^* = \arg\max_{1\le k\le K} \text{IMAGE-PROB}(f|\text{image} f \text{ belongs to class } k).$$

In another example in Section VI, we use our SRT model described in Construction 1 to label every pixel of an image with one of several class labels. We do this by extracting the maximum posterior marginal state [39] for every pixel n

$$\hat{j}_{\text{MPM}}(n) = \arg\max_j \text{PROB}(j|f,n).$$

It can be shown that

$$\hat{j}_{\text{MPM}}(n) = \arg\max_j \text{SURROUND}_{(j,\{n\})}\text{CENTER}_{(j,\{n\})}$$

where the center and surround variables are computed using the recursions in (22) and (23).

VI. Experimental Results

In this section, we illustrate our models and algorithms through several classification and segmentation examples.

A. Classification of Images With Construction 1

1) Experiment 1, Natural Images: Our first data set consists of 150 64×64 eight-bit grayscale images of houses, buildings, and store fronts, 50 images in each category, shown in Fig. 5. All images are preprocessed by equalizing their histograms, to insure 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 single class is partitioned into a training set of 40 images and a test set of ten images. The training set is used to train an SRT model described in Construction 1, using the center-surround algorithm of (18)–(23). We use $J=5$. The initial values of the production probabilities are chosen using an idea from [10]: A transition from any state to itself is more likely than a transition to another state. Thus, for example, the initial value of the production probability for $(j,R)\to\{(j',R'),(j'',R'')\}$ is smaller than that for $(j,R)\to\{(j,R'),(j,R'')\}$. To initialize the five means for the Gaussian observation distributions, the grayscale range [0, 255] is partitioned into five intervals of size 51; the initial values for the means are chosen to be the midpoints for these intervals (i.e., 25.5, 76.5, 127.5, 178.5, and 229.5), and the standard deviations are all initialized to $\sigma_j=51$. Parameter estimation is then done for each class, resulting in three SRT models. In the testing stage of the experiment, the likelihood of each of the three SRT models is calculated for every test image, and the maximum likelihood classification is obtained for the image. The five different experiments in Table II correspond to selecting five different sets of ten images from each class as the test set. Rows 1, 6, and 11 of Fig. 5 contain the test images for Experiment 1-1, rows 2, 7, and 12 contain the test images for Experiment 1-2, etc. Thus, a total of $10\times 5\times 3=150$ test images and $40\times 5\times 3=600$ training images are used in the five experiments. Note from the first two columns of numbers in Table II that the correct classification rate for the test images is consistently above 95%, with a total of only four misclassified test images out of 150 and no misclassified training images. This is despite the fact that each class consists of many heterogeneous images which are quite complex.

We use three other classification strategies for comparison. We first use the models learned in Experiments 1-1 through 1-5 to classify negative versions of all images in the database, i.e.,

$$\text{CENTER}_a^{(i)} = \begin{cases} \sum\limits_{\alpha\in S(a)} \bar{\mathbf{p}}_a(\alpha)\prod\limits_{b\in\alpha}\text{CENTER}_b^{(i)}, & \text{if } a \text{ is a nonterminal or } a=(0,Q)\\ \mathbf{p}_a(f_R^{(i)}), & \text{if } a=(j,R) \text{ is a terminal}\end{cases} \tag{22}$$

$$\text{SURROUND}_a^{(i)} = \begin{cases} 1, & \text{if } a=(0,Q)\\ \sum\limits_{(d\to\alpha)\in P(a)} \bar{\mathbf{p}}_d(\alpha)\text{SURROUND}_d^{(i)}\prod\limits_{b\in\alpha,\, b\ne a}\text{CENTER}_b^{(i)}, & \text{if } a\ne(0,Q)\end{cases} \tag{23}$$

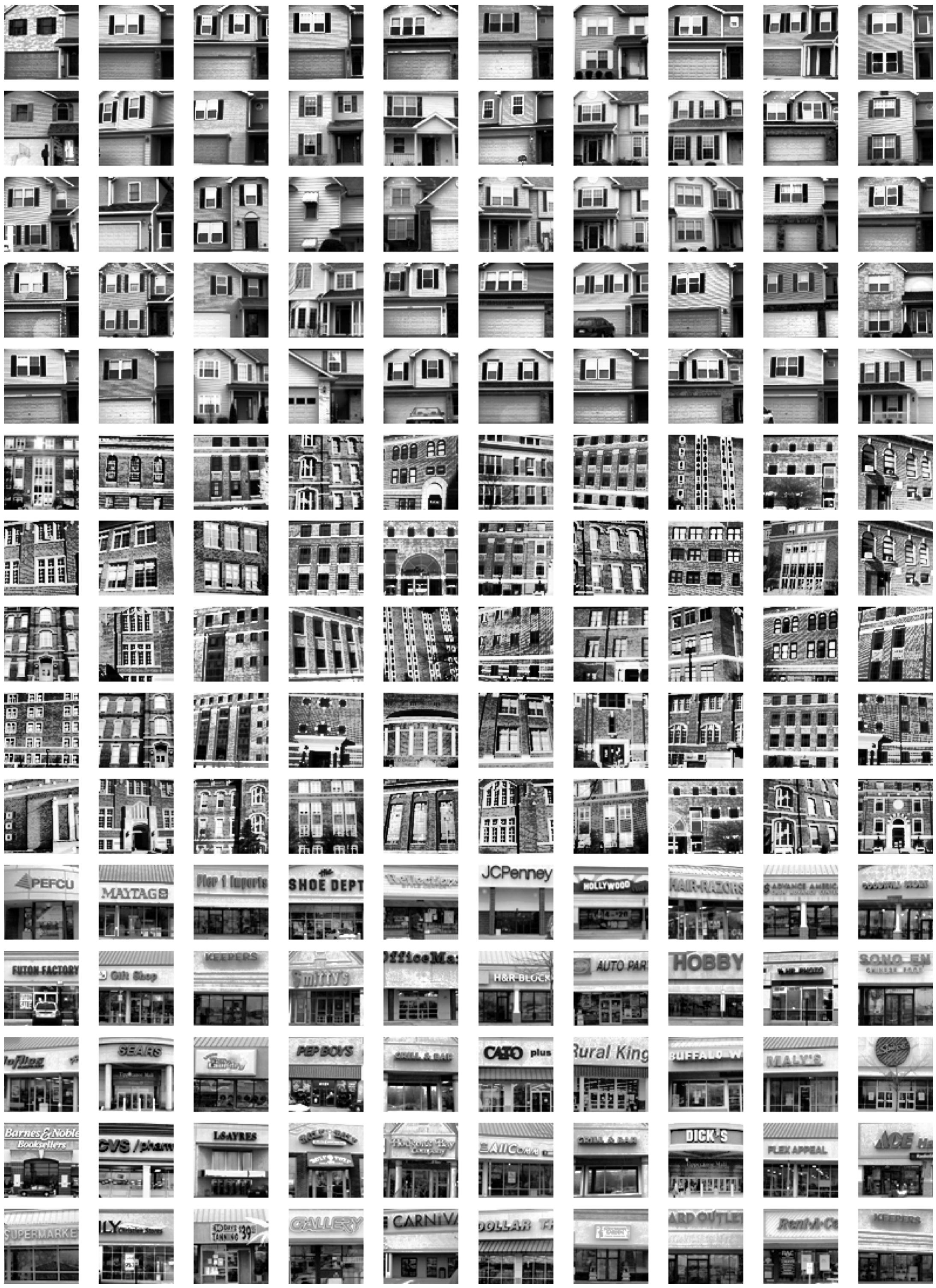


Fig. 5. Experiment 1. Database of images of (top five rows) houses, (middle five rows) buildings, and (bottom five rows) store fronts.

we replace every pixel value r with $255 - r$ in every image. Our algorithm still produces quite accurate classification results—about 80% correct classifications, as shown in columns 3 and 4 in Table II. This ability to classify most of the images after switching around their intensity values demonstrates the significance of the hierarchical structure learned by our parameter estimation algorithm and the overall robustness of the method.

In the second baseline experiment, we use an i.i.d. Gaussian mixture model where each pixel is modeled as a mixture of five Gaussians. The means and the variances for the Gaussian components as well as the mixture probabilities are estimated from the training data using an EM algorithm [9], resulting in a Gaussian mixture model for each image class. We then extract the maximum likelihood classifications of images based on these models. Note that an i.i.d. Gaussian mixture model is a

TABLE II
CORRECT CLASSIFICATION PERCENTAGES FOR THE EXPERIMENTS WITH THE HOUSE-BUILDING-STORE DATA SET

	Correct classification rate							
	SRT				Gaussian mixture			
	original images		negative images		pixels		wavelet coefficients	
	Training set	Test set	Training set	Test set	Training set	Test set	Training set	Test set
Experiment 1-1	100.0%	96.7%	78.3%	83.3%	33.3%	33.3%	56.7%	53.3%
Experiment 1-2	100.0%	96.7%	76.7%	80.0%	33.3%	33.3%	53.3%	53.3%
Experiment 1-3	100.0%	100.0%	82.5%	70.0%	33.3%	33.3%	57.3%	43.3%
Experiment 1-4	100.0%	96.7%	80.8%	83.3%	33.3%	33.3%	60.7%	23.3%
Experiment 1-5	100.0%	96.7%	81.7%	80.0%	33.3%	33.3%	55.3%	56.7%
Overall	100.0%	97.3%	80.0%	79.3%	33.3%	33.3%	56.7%	46.0%

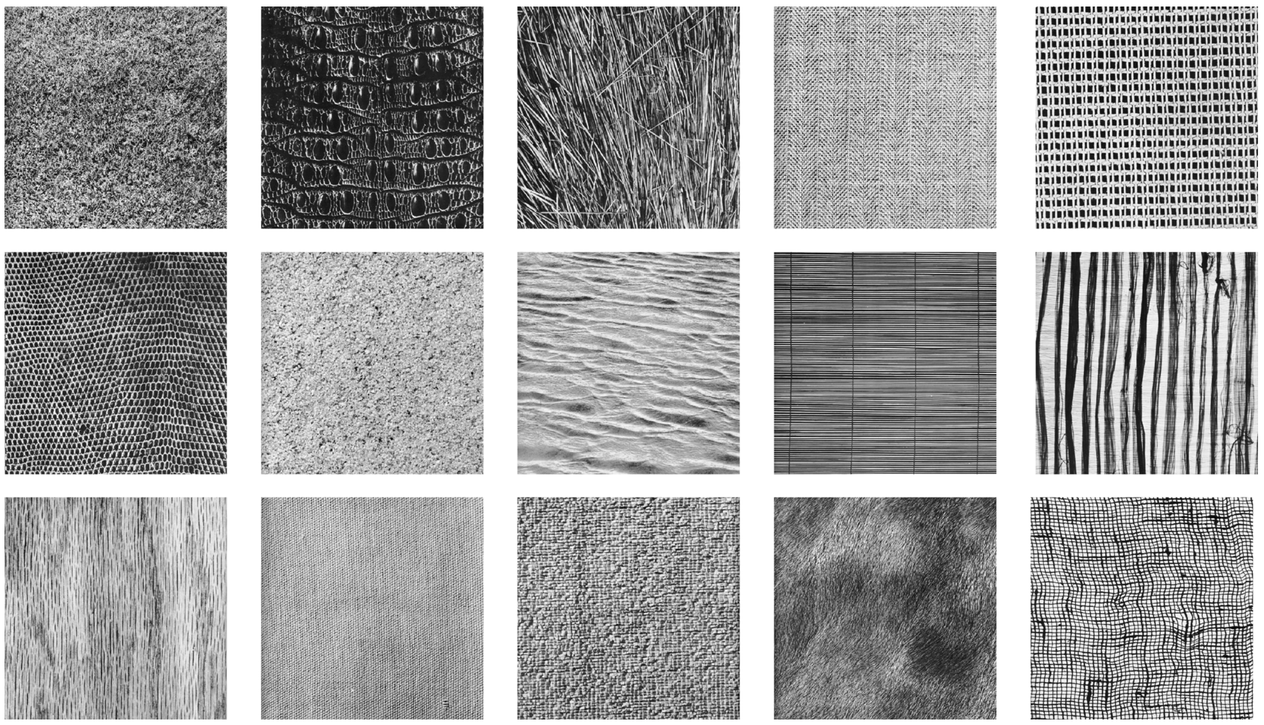


Fig. 6. Fifteen Brodatz texture images used in Experiment 2.

particular case of our SRT model which is obtained by entirely removing the hidden random tree part of our model—i.e., the part responsible for capturing the spatial organization. The i.i.d. model, thus, attempts to classify images based essentially on their histograms, which in this example is futile since all images have approximately the same histogram. We, therefore, would expect this classifier to perform similarly to a random assignment of the class labels, which is precisely what happens: In this example, it classifies every image as a store front, resulting in 1/3 correct classification rate for all experiments, as shown in the columns 6 and 7 of Table II.

Our final baseline experiment uses an i.i.d. Gaussian mixture model with five components to model the wavelet coefficients of the images, obtained with the Daubechies-8 wavelet. In essence, this strategy classifies images based on the histograms of their wavelet coefficients. The results are somewhat better than a random assignment of class labels since multiscale nature of the data is captured in the wavelet coefficients; however, the overall correct classification rate of the test data is below 50% and is less than half of the correct classification rate of the SRTs. This further demonstrates the utility of our framework in hierarchical modeling of images.

2) Experiment 2, Brodatz Textures: In this experiment, we use 15 Brodatz textures which were used in [12]: D9, D10, D15, D17, D20, D22, D29, D37, D49, D51, D68, D77, D84, D93, and D103, shown in Fig. 6. As in [12], we select 30 64×64 patches from each texture image as the training data and use the remaining 70 64×64 patches[6] as the test set. The training set is used to train an SRT model described in Construction 1, using the center-surround algorithm. The initialization of the parameter values is done as in Experiment 1. Parameter estimation is done for each class, resulting in fifteen SRT models. In the testing stage of the experiment, the maximum likelihood classifications are produced. The three different experiments in Table III correspond to selecting three nonoverlapping sets of thirty patches from each class as the training set. The masks for these training patches are selected using a random number

[6]Our full images are 640×640 and taken from www.ux.his.no/~tranden/brodatz.html.

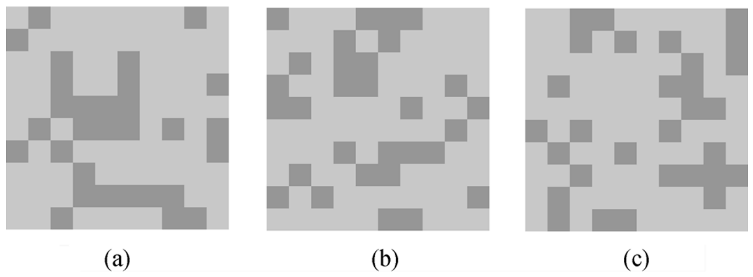


Fig. 7. Locations of the training patches for the texture experiments are shown in dark gray. (a) Experiment 2-1. (b) Experiment 2-2. (c) Experiment 2-3.

TABLE III
CORRECT CLASSIFICATION PERCENTAGES FOR THE EXPERIMENTS WITH BRODATZ TEXTURE DATA SET

	Correct classification rate	
	Training set	Test set
Experiment 2-1	99.1%	98.3%
Experiment 2-2	99.1%	97.4%
Experiment 2-3	98.2%	96.7%
Overall	98.8%	97.5%

generator, and are shown in Fig. 7(a)–(c), for Experiments 2-1, 2-2, and 2-3, respectively. Thus, a total of $30 \times 3 \times 15 = 1350$ training images and $70 \times 3 \times 15 = 3150$ test images are used in the three experiments. As Table III shows, the correct classification rate for the test images is 97.5%. The best result reported on these textures in [12] is 88.3% (second column of [12, Table III]). In addition, a multiscale texture classification method based on wavelets which is also used in [12], produces 83.2% correct classifications (third column of [12, Table III]).

B. Classification of Image Pixels With Construction 1

We now use our model to classify each image pixel into one of several classes and compare our results with the SMAP algorithm [10] which is based on a hidden tree model with a fixed quadtree structure. We use three 512×512 synthetic images from [10][7] shown in Fig. 8(b)–(d). Each image has six different Gaussian textures: one for the background and one for each row of circles. The class pattern image is shown in Fig. 8(a). In each of the three images, every pixel is uncorrelated with any other pixel; the differences between the textures are due to the different means and variances, shown in Table IV (class 1 is the background, and classes 2–6 are the five rows of circles, top to bottom). Just as in [10], we assume that the correct number of classes $J = 6$ is known, and that the correct means μ_j and variances σ_j for the Gaussian observation distributions are also known and are the same as those in Table IV. For each 512×512 image, we first estimate the maximum likelihood production probabilities using our center-surround algorithm described in the previous section. We then use the estimated grammar to obtain the maximum posterior marginal estimate [39] of the class for each pixel, by maximizing the posterior marginal of j for each leaf $(j, \{n\})$, as described in the previous section. Since our model is based on rectangular regions whereas we are attempting to extract circular boundaries, our model suffers from block artifacts similar to the ones observed in [10]. To alleviate this problem, we use the following simple strategy: We apply our classification algorithm to 16 different shifts of the input image, and select the classification for each pixel using a majority vote. In [10], blocking artifacts were suppressed by using a graph model which is more complicated than a tree. Our segmented images are shown in Fig. 8(e)–(g). As Table V shows, our algorithm outperforms the SMAP algorithm of [10], misclassifying about 10% fewer pixels than SMAP (the overall number of misclassified pixels in the three images for our algorithm and for SMAP is about 100 000 and 110 000, respectively).[8]

Many segmentation problems, however, are better addressed by models which do not restrict the leaves to be single pixels and do not restrict the regions to be dyadic rectangles. In the next subsection, we consider the scenario where both the leaves and the internal tree nodes can be arbitrary rectangles. The scenario where the leaves are arbitrarily shaped regions is considered elsewhere [50].

C. Classification of Image Regions With Construction 2

Accurately segmenting a document image and classifying the segments is an important first step in many applications. For example, in document image coding, different compression strategies are used for the text regions and image regions. We apply our Construction 2 to a document segmentation task.

Given a typical document image with several million pixels (several thousand by several thousand), we first preprocess to reduce dimensionality: We compute the column and row averages and segment these two 1-D signals using a simple edge detector, to produce a rectangular (not necessarily uniform) grid Q which is 81×61, as shown in Fig. 9.

We develop a simple grammar for Q, based on Construction 2. The grammar has four states: j =1 is the unique nonterminal state, and j =2, 3, 4 are terminals for images, text, and background, respectively. We let $r_1 = \varepsilon$, $r_2 = r_3 = r_4 = (1 - \varepsilon)/3$, and construct the production probabilities by letting $\bar{\mathbf{p}}_a(\alpha) = r_{j'} r_{j''}/(m + n - 2)$ for any nonterminal symbol $a = (1, R)$ where R is an $m \times n$ rectangle, and for any $\alpha = \{(j', R'), (j'', R'')\}$. The small number ε which, in this experiment, is set to 2^{-300}, essentially controls the number of leaves in the MAP tree. We model the observation probability distributions $\tilde{\mathbf{p}}_a$ as arbitrary discrete probability mass functions over the set $\{0, \ldots, 255\}$ of all grayscale values. We estimate

[7]www.ece.purdue.edu/~bouman/software/segmentation/data/results_94_paper.tar.gz

[8]As in [10], the last column of Table V gives the averages of the preceding six columns. The numbers may not exactly match since all of them are rounded.

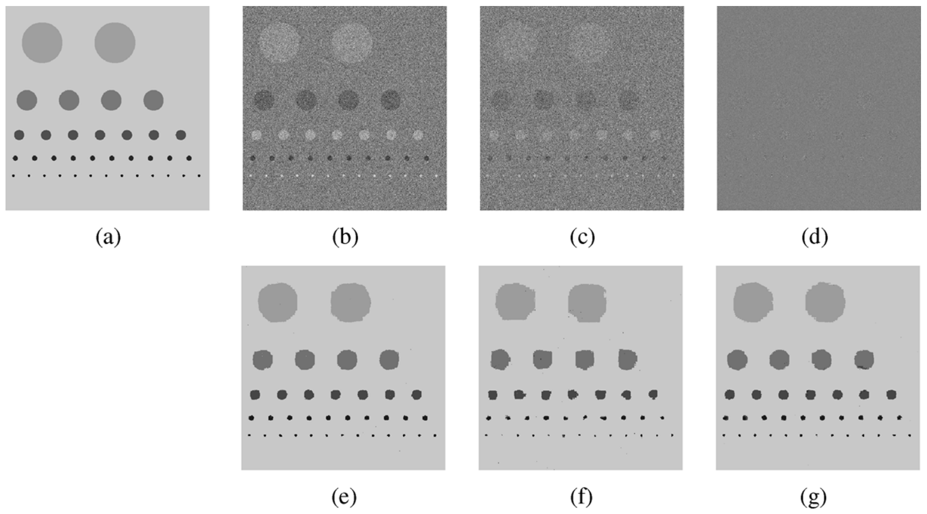


Fig. 8. Pixelwise classification experiments: (a) Ground truth, (b–d) input images, and (e–g) the respective results.

TABLE IV
MEAN AND STANDARD DEVIATION OF EACH TEXTURE FOR THREE DIFFERENT IMAGES

		Class designation					
		1	2	3	4	5	6
image 1	μ	127.0	145.0	101.6	163.0	76.1	199.0
	σ	32	32	32	32	32	32
image 2	μ	127.0	137.1	112.7	147.2	98.4	167.5
	σ	32	32	32	32	32	32
image 3	μ	127	127	127	127	127	127
	σ	8.00	10.55	13.93	18.37	24.25	32

these distributions in a supervised mode, from ten hand-segmented training images. Each distribution is estimated as the histogram of the pixel values within the corresponding regions in the training images, see (14).

Given a test image, we then extract the MAP tree which finds the optimal rectangular tiling of the grid Q and labels each tile with one of the three terminal states. Fig. 10 shows the results for one test image which is not a member of the training set. Fig. 10(a) and (b) illustrates that, even with this simple model, we are able to quite accurately segment a document image and classify its regions. For comparison, we provide in Fig. 10(c) and (d) the optimal tiling and labels for a model where the only allowed split locations for each rectangle are in the middle vertically or horizontally, as in Construction 1. The resulting tiling of Fig. 10(c) is clearly inferior to the one shown in Fig. 10(a), and the resulting classification of the tiles of Fig. 10(d) is clearly inferior to the one shown in Fig. 10(b). The computational price paid for better performance in this case is quite modest: The results of Fig. 10(a) and (b) took about 10 min to compute whereas the results of Fig. 10(c) and (d) took about 5 min on the same machine.

VII. CONCLUSION

We have constructed a novel class of grammar-based spatial random tree image models. Motivated by the standard 1-D PCFGs, we have developed 2-D grammars that explicitly incorporate spatial information as part of the grammar. We have developed exact inference algorithms for ML parameter estimation, model selection via hypothesis testing, MAP estimation of the tree structure and states, and MPM estimation. These inference algorithms have the same form as their 1-D counterparts [38]; however, they can be applied not only to 1-D strings, but also to 2-D images and, in fact, to multidimensional data with any number of dimensions. We have illustrated our models by using them to classify natural images and images of Brodatz textures, and to segment a document image and synthetic images.

In the illustrations, we used two specific instances of our model, both of which are based on rectangular regions. Given a specific application, an important design issue arises of either choosing one of these two constructions or designing a different one. There are several factors that go into the design process: More general models result in higher computational complexity and require more training data since they have more parameters. However, they may provide significant improvements as compared to the simpler models, as in the example of Section VI-C.

Our illustrative classification and segmentation algorithms are handicapped by the fact that they use very rudimentary features (namely, the grayscale values of individual pixels). An open research avenue is combining our statistical models with more sophisticated features [50] such as, for example, wavelet coefficients [37], [45].

We note that there is a close relationship between the MAP tree extraction algorithm for SRTs (and the corresponding algorithm for PCFGs) and best-basis search algorithms. As pointed out in [29], standard wavelet packet and dyadic local cosine dictionaries [16], [17], as well as anisotropic 2-D wavelet packet dictionaries [4], [19], are all particular cases of multitree dictionaries, and the corresponding best-basis search algorithms are particular cases of the algorithm in Fig. 2. The specialization of our algorithm to dyadic rectangular tilings yields the “dyadic CART” algorithm of [19] which has been used to construct optimal classification trees in [8], [48]. We are currently investigating the application of more general versions of our algorithm to other related problems which involve the construction of optimal trees, such as tree-structured vector quantization [11].

Hierarchical models are important both because they naturally describe many aspects of the world and also because of

TABLE V
CORRECT CLASSIFICATION PERCENTAGES FOR OUR SRT MODEL AND FOR SMAP OF [10]

		Class label						Class Average
		1	2	3	4	5	6	
image 1	SRT	100%	96%	97%	95%	91%	93%	95.2%
	SMAP	100%	95%	96%	94%	93%	78%	92.6%
image 2	SRT	100%	92%	92%	86%	63%	44%	79.5%
	SMAP	99%	94%	93%	86%	53%	70%	82.5%
image 3	SRT	100%	96%	94%	93%	72%	74%	88.1%
	SMAP	100%	95%	95%	92%	61%	58%	83.5%
overall	SRT	100%	94.8%	94.3%	91.4%	75.4%	70.0%	87.6%
	SMAP	100%	94.7%	94.7%	90.7%	69.0%	68.7%	86.2%

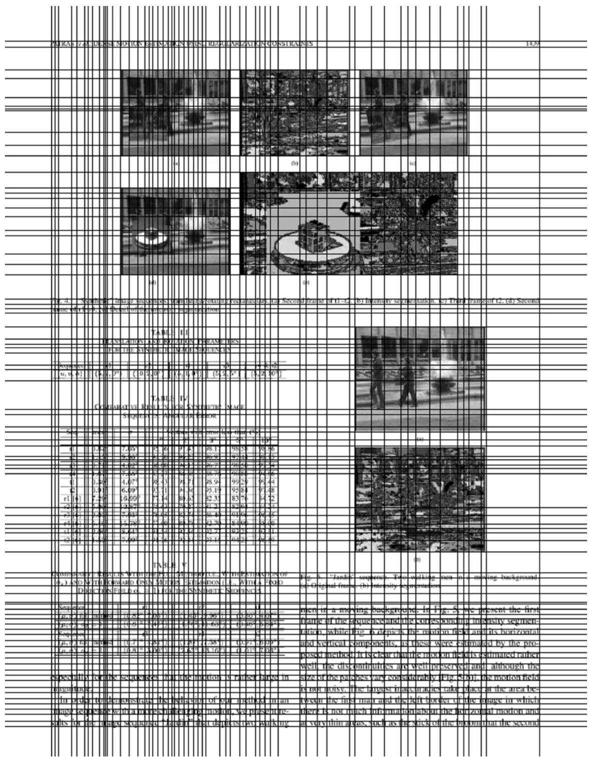


Fig. 9. The 81×61 nonuniform grid Q produced by our preprocessing algorithm in Section VI-C.

computational advantages stemming from the fact that many problems can be solved more efficiently by organizing computations in a multiscale or hierarchical manner. This has motivated a large body of research on multiresolution representations, linear and nonlinear scale spaces, and multiscale statistical models [56]. Our hierarchical models are unique in that they are both statistical and adaptive (i.e., able to adjust the tree structure to the data), yet computationally tractable. We, therefore, anticipate that they will be useful in a wide variety of applications where it is important to extract an optimal hierarchical structure of the data. We have already successfully applied a deterministic version of our model to image compression [29], and are currently applying such models to document image processing [7] and video [53].

APPENDIX
DERIVATION OF THE CENTER-SURROUND ALGORITHM FOR THE EM UPDATES

We prove the fact that the righthand side of (15) can be exactly computed using the righthand side of the update equation for $\hat{\rho}_{[a]}([\alpha])$, (18), together with the center and surround recursions (22), (23). We also prove the likelihood calculation formula of (24).

We start with the following definitions.

- $\mathbf{P}$ denotes any marginal or conditional distribution induced by $\textsc{joint-prob}$.
- $\mathbf{P}(f^{(i)}, a)$ is the joint distribution of the image $f^{(i)}$ and the set of all trees containing the symbol a.
- The *center variable* is defined as $\textsc{center}_a^{(i)} \triangleq \mathbf{P}(f_R^{(i)}|a)$, for any symbol $a = (j, R)$, where $f_R^{(i)}$ is the restriction of the image $f^{(i)}$ to the region R.
- The *surround variable* is defined as $\textsc{surround}_a^{(i)} \triangleq \mathbf{P}(f_{R^c}^{(i)}, a)$, for any symbol $a = (j, R)$, where $R^c = Q \backslash R$ is the complement of R in Q [see Fig. 11(a)].

To prove the likelihood calculation formula given in (24), note that, for the root symbol $a = (0, Q)$, we have $\textsc{center}_a^{(i)} = \mathbf{P}(f_Q^{(i)}|(0,Q)) = \mathbf{P}(f_Q^{(i)}) = \textsc{image-prob}(f^{(i)})$.

Note also that, for any nonroot symbol $a = (j, R)$, the definitions of the center and surround variables have the following interpretations: $\textsc{center}_a^{(i)}$ is the conditional probability distribution of the subimage $f_R^{(i)}$ given the set of all trees containing the symbol a; and $\textsc{surround}_a^{(i)}$ is the joint probability distribution of the subimage $f_{R^c}^{(i)}$ and the set of all trees containing the symbol a. We now prove the recursive center and surround formulas of (22) and (23).

If $a = (j, R)$ is a terminal, then

$$\textsc{center}_a^{(i)} = \mathbf{P}(f_R^{(i)}|a) = \mathbf{p}_a(f_R^{(i)}) \tag{25}$$

otherwise

$$\begin{aligned}\textsc{center}_a^{(i)} &= \mathbf{P}(f_R^{(i)}|a) \\ &= \sum_{\alpha \in S(a)} \mathbf{P}(f_R^{(i)}, a \to \alpha | a) \\ &= \sum_{\alpha \in S(a)} \mathbf{P}(f_R^{(i)} | a \to \alpha) \bar{\mathbf{p}}_a(\alpha). \end{aligned} \tag{26}$$

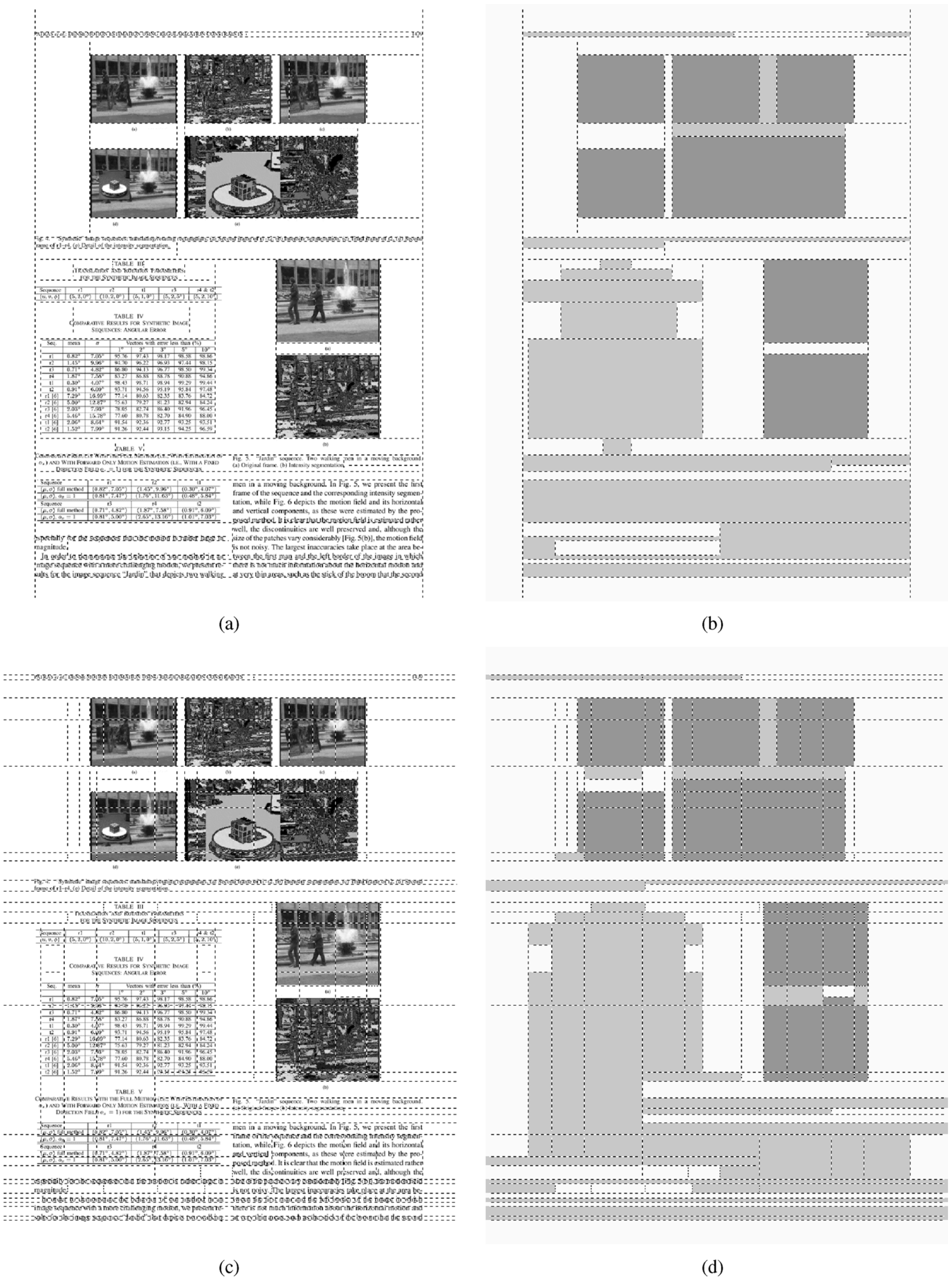


Fig. 10. (a) Document image with the optimal arbitrary rectangular tiling extracted using our MAP estimation algorithm, running time approximately ten minutes on an Athlon. (b) Class labels for the optimal arbitrary tiling: (dark) image, (light) text, and (white) background. (c) Document image with the optimal dyadic rectangular tiling extracted using our MAP estimation algorithm, running time approximately five minutes on an Athlon. (d) Class labels for the optimal dyadic tiling: (dark) image, (light) text, and (white) background.

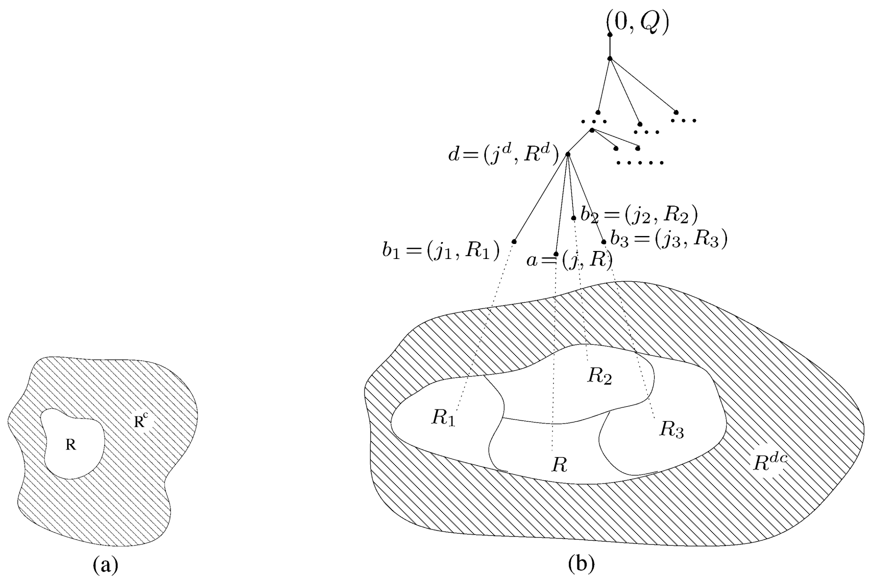


Fig. 11. Illustrations for the derivation of the EM update formulas. (a) Region R and its complement $R^c = Q \backslash R$. (b) Illustration of the surround recursion.

Using the notation $\alpha = \{b_1, b_2, \ldots, b_{|\alpha|}\}$ and $b_k = (j_k, R_k)$, as well as the Markov property of the SRTs, we have

$$\begin{aligned}\mathbf{P}(f_R^{(i)} | a \to \alpha) &= \mathbf{P}(f_{R_1}^{(i)}, f_{R_2}^{(i)}, \ldots, f_{R_{|\alpha|}}^{(i)} | a \to \alpha) \\ &= \prod_{k=1}^{|\alpha|} \mathbf{P}(f_{R_k}^{(i)} | b_k) = \prod_{k=1}^{|\alpha|} \text{CENTER}_{b_k}^{(i)}.\end{aligned}$$

Substituting this back into (26), we obtain

$$\text{CENTER}_a^{(i)} = \sum_{\alpha \in S(a)} \bar{\mathbf{p}}_a(\alpha) \prod_{k=1}^{|\alpha|} \text{CENTER}_{b_k}^{(i)}. \tag{27}$$

Equations (27) and (25) form the center recursion formula of (22).

To develop the surround recursion, note that when $a = (0, Q)$ is the root symbol, then $\text{SURROUND}_a^{(i)} = 1$. Otherwise, if $a = (j, R)$ and $R^c = Q \backslash R$, then

$$\begin{aligned}\text{SURROUND}_a^{(i)} &= \mathbf{P}(f_{R^c}^{(i)}, a) \\ &= \sum_{(d \to \alpha) \in P(a)} \mathbf{P}(f_{R^c}^{(i)}, d \to \alpha) \\ &= \sum_{(d \to \alpha) \in P(a)} \mathbf{P}(f_{R^c}^{(i)} | d \to \alpha) \bar{\mathbf{p}}_d(\alpha) \mathbf{P}(d).\end{aligned}$$

Let $d = (j^d, R^d)$, $\alpha = \{a, b_1, b_2, \ldots, b_{|\alpha|-1}\}$, $b_k = (j_k, R_k)$, and $R^{dc} = Q \backslash R^d$—see Fig. 11(b). Notice that, given these definitions, we have

$$R^c = R^{dc} \cup R_1 \cup R_2 \cup \cdots \cup R_{|\alpha|-1}.$$

$$\begin{aligned}\text{SURROUND}_a^{(i)} &= \sum_{(d \to \alpha) \in P(a)} \mathbf{P}(f_{R^{dc}}^{(i)}, f_{R_1}^{(i)}, f_{R_2}^{(i)}, \ldots, f_{R_{|\alpha|-1}}^{(i)} | d \to \alpha) \bar{\mathbf{p}}_d(\alpha) \mathbf{P}(d) \\ &= \sum_{(d \to \alpha) \in P(a)} \bar{\mathbf{p}}_d(\alpha) \mathbf{P}(f_{R^{dc}}^{(i)} | d \to \alpha) \mathbf{P}(d) \mathbf{P}(f_{R_1}^{(i)}, f_{R_2}^{(i)}, \ldots, f_{R_{|\alpha|-1}}^{(i)} | d \to \alpha, f_{R^{dc}}^{(i)}) \\ &= \sum_{(d \to \alpha) \in P(a)} \bar{\mathbf{p}}_d(\alpha) \mathbf{P}(f_{R^{dc}}^{(i)} | d \to \alpha) \mathbf{P}(d) \mathbf{P}(f_{R_1}^{(i)}, f_{R_2}^{(i)}, \ldots, f_{R_{|\alpha|-1}}^{(i)} | d \to \alpha) \\ &= \sum_{(d \to \alpha) \in P(a)} \bar{\mathbf{p}}_d(\alpha) \mathbf{P}(f_{R^{dc}}^{(i)} | d \to \alpha) \mathbf{P}(d) \prod_{k=1}^{|\alpha|-1} \mathbf{P}(f_{R_k}^{(i)} | b_k) \\ &= \sum_{(d \to \alpha) \in P(a)} \bar{\mathbf{p}}_d(\alpha) \mathbf{P}(f_{R^{dc}}^{(i)} | d \to \alpha) \mathbf{P}(d) \prod_{k=1}^{|\alpha|-1} \text{CENTER}_{b_k}^{(i)}.\end{aligned} \tag{28}$$

Therefore, we have (28), derived and shown at the bottom of the previous page. Because of the Markov property of the SRTs

$$\mathbf{P}(f^{(i)}_{R^{dc}}|d \to \alpha)\mathbf{P}(d) = \mathbf{P}(f^{(i)}_{R^{dc}}|d)\mathbf{P}(d) = \mathbf{P}(f^{(i)}_{R^{dc}}, d) = \text{SURROUND}^{(i)}_{d}.$$

Substituting this back into (28), results in the surround recursion of (23).

To derive the parameter update formulas, we rewrite the right-hand side of (15) as follows:

$$\begin{aligned}\frac{\sum_{i=1}^{I} E[h_{[a\to\alpha]}(T^{(i)})|F^{(i)}=f^{(i)}]}{\sum_{i=1}^{I} E[h_{[a]}(T^{(i)})|F^{(i)}=f^{(i)}]} &= \frac{\sum_{i=1}^{I}\sum_{t\in\mathcal{D}} \text{POSTERIOR-PROB}(t|f^{(i)})h_{[a\to\alpha]}(t)}{\sum_{i=1}^{I}\sum_{t\in\mathcal{D}} \text{POSTERIOR-PROB}(t|f^{(i)})h_{[a]}(t)}\\ &= \frac{\sum_{i=1}^{I}\frac{\sum_{t\in\mathcal{D}}\text{JOINT-PROB}(t,f^{(i)})h_{[a\to\alpha]}(t)}{\text{IMAGE-PROB}(f^{(i)})}}{\sum_{i=1}^{I}\frac{\sum_{t\in\mathcal{D}}\text{JOINT-PROB}(t,f^{(i)})h_{[a]}(t)}{\text{IMAGE-PROB}(f^{(i)})}}.\end{aligned} \qquad (29)$$

The term IMAGE-PROB$(f^{(i)})$ is easily computed from the center variables using (24). We now show how to use the center and surround variables to calculate the remaining two quantities in the righthand side of (29). We start with the denominator

$$\begin{aligned}&\sum_{t\in\mathcal{D}}\text{JOINT-PROB}(t,f^{(i)})h_{[a]}(t)\\ &\quad= \sum_{a'\in[a]}\sum_{t\in\mathcal{D}}\text{JOINT-PROB}(t,f^{(i)})h_{a'}(t)\\ &\quad= \sum_{a'\in[a]}\mathbf{P}(f^{(i)},a')\\ &\quad= \sum_{a'\in[a]}\mathbf{P}(f^{(i)}_R, f^{(i)}_{R^c}, a') \qquad (\text{if } a'=(j,R))\\ &\quad= \sum_{a'\in[a]}\text{SURROUND}^{(i)}_{a'}\text{CENTER}^{(i)}_{a'}\\ &\qquad (\text{due to Markovianity of SRTs.}).\end{aligned} \qquad (30)$$

We now proceed to the numerator of (29)

$$\begin{aligned}&\sum_{t\in\mathcal{D}}\text{JOINT-PROB}(t,f^{(i)})h_{[a\to\alpha]}(t)\\ &\quad= \sum_{(a'\to\alpha')\in[a\to\alpha]}\sum_{t\in\mathcal{D}}\text{JOINT-PROB}(t,f^{(i)})h_{a'\to\alpha'}(t)\\ &\quad= \sum_{(a'\to\alpha')\in[a\to\alpha]}\mathbf{P}(f^{(i)},a'\to\alpha').\end{aligned} \qquad (31)$$

Let $a'=(j,R)$, $\alpha'=\{b_1,b_2,\ldots,b_{|\alpha'|}\}$, and $b_k=(j_k,R_k)$. Then

$$\begin{aligned}&\mathbf{P}(f^{(i)},a'\to\alpha')\\ &\quad= \mathbf{P}(f^{(i)}_R|a'\to\alpha')\mathbf{P}(f^{(i)}_{R^c},a'\to\alpha')\\ &\quad= \mathbf{P}(f^{(i)}_{R_1},f^{(i)}_{R_2},\ldots,f^{(i)}_{R_{|\alpha'|}}|a'\to\alpha')\mathbf{P}(f^{(i)}_{R^c},a')\bar{\mathbf{p}}_{a'}(\alpha')\\ &\quad= \prod_{k=1}^{|\alpha'|}\mathbf{P}(f^{(i)}_{R_k}|b_k)\mathbf{P}(f^{(i)}_{R^c},a')\bar{\mathbf{p}}_{a'}(\alpha')\\ &\quad= \bar{\mathbf{p}}_{a'}(\alpha')\text{SURROUND}^{(i)}_{a'}\prod_{b\in\alpha'}\text{CENTER}^{(i)}_{b}.\end{aligned}$$

Substituting this back into (31) results in

$$\begin{aligned}&\sum_{t\in\mathcal{D}}\text{JOINT-PROB}(t,f^{(i)})h_{[a\to\alpha]}(t)\\ &\quad= \sum_{(a'\to\alpha')\in[a\to\alpha]}\bar{\mathbf{p}}_{a'}(\alpha')\text{SURROUND}^{(i)}_{a'}\prod_{b\in\alpha'}\text{CENTER}^{(i)}_{b}.\end{aligned} \qquad (32)$$

Finally, substituting (30) and (32) into (29), we obtain the parameter update formula of (18). Similarly, we can derive the update equations for the parameters $\mu_{[a]}$ and $\Lambda_{[a]}$.

Acknowledgment

The authors would like to thank Prof. E. J. Delp and Dr. Y. Huang for their help and suggestions during the preparation of this manuscript.

References

[1] J. Baker, "Trainable grammars for speech recognition," in *Proc. Speech Communications Papers 97th Meeting Acoustic Society of America*, D. Klatt and J. Wolf, Eds., 1979, pp. 547–550.

[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. Inf. Theory*, vol. 38, no. 2, pp. 766–784, Mar. 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. Statist.*, vol. 41, no. 1, pp. 164–171, 1970.

[4] N. N. Bennett, "Fast algorithm for best anisotropic Walsh bases and relatives," *J. Appl. Comput. Harmon. Anal.*, vol. 8, no. 1, pp. 86–103, Jan. 2000.

[5] M. Bicego, V. Murino, and M. A. T. Figueiredo, "A sequential pruning strategy for the selection of the number of states in hidden Markov models," *Pattern Recognit. Lett.*, vol. 24, no. 9–10, pp. 1395–1407, Jun. 2003.

[6] I. J. Bienaymé, "De la loi de multiplication et de la durée des familles," *Soc. Philomat. Paris Extraits*, vol. 5, pp. 37–39, 1845. Reprinted as an Appendix to [32].

[7] B. Bitlis, X. Feng, J. L. Harris, C. A. Bouman, I. Pollak, M. P. Harper, and J. P. Allebach, "A hierarchical document description and comparison method," in *Proc. IS&T Archiving Conf.*, San Antonio, TX, Apr. 20–23, 2004, pp. 195–198.

[8] G. Blanchard, C. Schäfer, and Y. Rozenholc, "Oracle bounds and exact algorithm for dyadic classification trees," in *Proc. 17th. Conf. Learning Theory*, 2004, vol. 3120, Springer Lecture Notes in Artificial Intelligence, pp. 378–392.

[9] C. A. Bouman, CLUSTER: An Unsupervised Algorithm for Modeling Gaussian Mixtures 1997 [Online]. Available: www.ece.purdue.edu/~bouman/software/cluster/manual.pdf

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

[11] A. Buzo, A. H. Gray Jr., R. M. Gray, and J. D. Markel, "Speech coding based upon vector quantization," *IEEE. Trans. Acoust. Speech Signal Process.*, vol. 28, no. 5, pp. 562–574, Oct. 1980.

[12] P. Campisi, A. Neri, G. Panci, and G. Scarano, "Robust rotation-invariant texture classification using a model based approach," *IEEE Trans. Image Process.*, vol. 13, no. 6, pp. 782–791, Jun. 2004.

[13] H. Choi and R. G. Baraniuk, "Multiscale image segmentation using wavelet-domain hidden Markov models," *IEEE Trans. Image Process.*, vol. 10, no. 9, pp. 1309–1321, Sep. 2001.

[14] K. C. Chou, A. S. Willsky, A. Benveniste, and M. Basseville, "Recursive and iterative estimation algorithms for multiresolution stochastic processes," in *Proc. 29th IEEE Conf. Decision Control*, Honolulu, HI, Dec. 13–15, 1989, vol. 2, pp. 1184–1189.

[15] S. C. Clippingdale and R. Wilson, "Least squares estimation on a multiresolution pyramid," in *Proc. ICASSP*, Glasgow, U.K., 1989, pp. 1409–1412.

[16] R. R. Coifman, Y. Meyer, and M. V. Wickerhauser, "Wavelet analysis and signal processing," in *Wavelets and their Applications*, M. B. Ruskai, G. Beylkin, R. Coifman, I. Daubechies, S. Mallat, Y. Meyer, and L. Raphael, Eds. Boston, MA: Jones and Bartlett, 1992, pp. 153–178.

[17] R. R. Coifman and M. V. Wickerhauser, "Entropy based algorithms for best basis selection," *IEEE Trans. Inf. Theory*, vol. 38, no. 2, pp. 713–718, Mar. 1992.

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

[19] D. Donoho, "CART and best-ortho-basis: A connection," *Ann. Statist.*, vol. 25, pp. 1870–1911, 1997.

[20] M. A. T. Figueiredo, J. M. N. Leitão, and A. K. Jain, "On fitting mixture models," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*, E. Hancock and M. Pellilo, Eds. New York: Springer-Verlag, 1999, pp. 54–69.

[21] K. S. Fu, *Syntactic Pattern Recognition and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1982.

[22] S. Geman and M. Johnson, "Probability and statistics in computational linguistics: A brief review," in *Mathematical Foundations of Speech and Language Processing*, M. Johnson, S. Khudanpur, M. Ostendorf, and R. Rosenfeld, Eds. New York: Springer, 2004, vol. 138, IMA Volumes in Mathematics and Its Applications, pp. 1–26.

[23] S. Geman, D. F. Potter, and Z. Chi, Composition Systems, Tech. Rep., Division Appl. Math., Brown Univ., Providence, RI, 1998.

[24] M. M. Hertz, "Entropy of languages generated by automata or context-free grammars with unique output," *Nauchno-Tekhnicheskaia Informatsiia, Seriia 2, Informatsionnye Protsessy i Sistemy*, no. 1, pp. 29–34, Jan. 1968.

[25] G. E. Hinton, Z. Ghahramani, and Y. W. Teh, "Learning to parse images," in *Advances Neural Information Processing Systems*, S. A. Solla, T. K. Leen, and K.-R. Müller, Eds. Cambridge, MA: MIT Press, 2000, vol. 12, pp. 463–469.

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

[27] Y. Huang and I. Pollak, "MLC: A novel image coder based on multitree local cosine dictionaries," *IEEE Signal Process. Lett.*, vol. 12, no. 12, pp. 843–846, Dec. 2005.

[28] Y. Huang, I. Pollak, M. N. Do, and C. A. Bouman, "Optimal tilings and best basis search in large dictionaries," in *Proc. 37th Asilomar Conf. Signals, Systems, Computers*, Pacific Grove, CA, Nov. 9–12, 2003, pp. 327–331.

[29] ——, "Fast search for best representations in multitree dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 7, pp. 1779–1793, Jul. 2006.

[30] T. Kanungo and S. Mao, "Stochastic language models for style-directed layout analysis of document images," *IEEE Trans. Image Process.*, vol. 12, no. 5, pp. 583–596, May 2003.

[31] D. G. Kendall, "Branching processes since 1873," *J. Lond. Math. Soc.*, vol. 41, pp. 385–406, 1966.

[32] ——, "The genealogy of genealogy: Branching processes before (and after) 1873," *Bull. Lond. Math. Soc.*, vol. 7, pp. 225–253, 1975.

[33] J.-M. Laferté, P. Pérez, and F. Heitz, "Discrete Markov image modeling and inference on the quadtree," *IEEE Trans. Image Process.*, vol. 9, no. 3, pp. 390–404, Mar. 2000.

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

[35] T. J. Li and K. S. Fu, Automata Games, Stochastic Automata, and Formal Languages, Tech. Rep. TR-EE 69-1, School Elect. Eng., Purdue Univ., West Lafayette, IN, 1969.

[36] J. Li, R. M. Gray, and R. A. Olshen, "Multiresolution image classification by hierarchical modeling with two-dimensional hidden Markov models," *IEEE Trans. Inf. Theory*, vol. 46, no. 5, pp. 1826–1841, Aug. 2000.

[37] M. R. Luettgen, W. C. Karl, A. S. Willsky, and R. R. Tenney, "Multiscale representations of Markov random fields," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3377–3395, Dec. 1993.

[38] C. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press, 1999.

[39] J. Marroquin, S. Mitter, and T. Poggio, "Probabilistic solution of ill-posed problems in computational vision," *J. Amer. Statist. Assoc.*, vol. 82, no. 397, pp. 76–89, Mar. 1987.

[40] I. Pollak, J. M. Siskind, M. P. Harper, and C. A. Bouman, Spatial Random Trees and the Center-Surround Algorithm, Tech. Rep. TR-ECE-03-03, School Elect. Eng., Purdue Univ., West Lafayette, IN, 2003.

[41] ——, "Modeling and estimation of spatial random trees with application to image classification," in *Proc. ICASSP*, Hong Kong, Apr. 6–10, 2003, pp. 305–308.

[42] ——, "Parameter estimation for spatial random trees using the EM algorithm," in *Proc. Int. Conf. Image Processing*, Barcelona, Spain, Sep. 14–17, 2003, pp. 257–260.

[43] D. F. Potter, "Compositional Pattern Recognition," Ph.D. dissertation, Div. Appl. Math., Brown Univ., Providence, RI, 1999.

[44] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[45] J. K. Romberg, H. Choi, and R. G. Baraniuk, "Bayesian tree-structured image modeling using wavelet-domain hidden Markov models," *IEEE Trans. Image Process.*, vol. 10, no. 7, pp. 1056–1068, Jul. 2001.

[46] A. Rosenfeld, *Picture Languages*. New York: Academic, 1979.

[47] D. Sankoff, "Branching processes with terminal types: Application to context-free grammars," *J. Appl. Probl.*, vol. 8, pp. 233–240, 1971.

[48] C. Scott and R. D. Nowak, "Minimax-optimal classification with dyadic decision trees," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1335–1353, Apr. 2006.

[49] A. C. Shaw, "A formal picture description scheme as a basis for picture processing systems," *Inf. Control*, vol. 14, pp. 9–52, 1969.

[50] J. M. Siskind, J. Sherman, I. Pollak, M. P. Harper, and C. A. Bouman, "Spatial random tree grammars for modeling hierarchal structure in images," *IEEE Trans. Pattern Anal. Mach. Intell.*, submitted for publication.

[51] A. Stolcke and S. M. Omohundro, Best-First Model Merging for Hidden Markov Model Induction, Tech. Rep. TR-94-003, Int. Comput. Sci. Inst., Berkeley, CA, 1994.

[52] A. J. Storkey and C. K. I. Williams, "Image modelling with position-encoding dynamic trees," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 7, pp. 859–871, Jul. 2003.

[53] C. M. Taskiran, I. Pollak, C. A. Bouman, and E. J. Delp, "Stochastic models of video structure for program genre detection," in *Visual Content Processing and Representation*. New York: Springer, 2003, vol. 2849, Lecture Notes in Computer Science, pp. 84–92.

[54] H. W. Watson and F. Galton, "On the probability of extinction of families," *J. Anthropol. Inst. Great Brit. Ireland*, vol. 4, pp. 138–144, 1875.

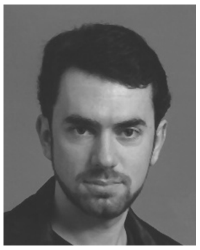
[55] C. K. I. Williams and N. J. Adams, "DTs: Dynamic trees," in *Advances in Neural Information Processing Systems*, M. J. Kearns, S. A. Solla, and D. A. Cohn, Eds. Cambridge, MA: MIT Press, 1999, vol. 11.

[56] A. S. Willsky, "Multiresolution Markov models for signal and image processing," *Proc. IEEE*, vol. 90, no. 8, pp. 1396–1458, Aug. 2002.

[57] R. Wilson and C.-T. Li, "A class of discrete multiresolution random fields and its application to image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 1, pp. 42–56, Jan. 2003.

Wiley Wang received the B.S. and M.Eng. degrees in electronic engineering from Tsinghua University, China, in 1999 and 2001, respectively. He is currently pursuing the Ph.D. degree at the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN.

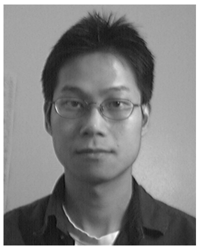
His research interests are in image and signal processing. He was a summer intern at Xerox Innovation Group, Webster, NY, in 2006.



Ilya Pollak (S'99–M'99–SM'05) received the B.S. and M.Eng. degrees in 1995 and the Ph.D. degree in 1999, from the Massachusetts Institute of Technology, Cambridge, in electrical engineering.

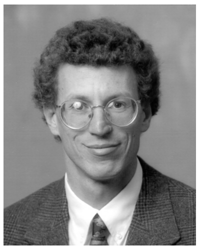
From 1999 to 2000, he was a postdoctoral researcher at the Division of Applied Mathematics, Brown University, Providence, RI. In 2000, he joined the faculty of Purdue University, West Lafayette, IN, where he is currently an Associate Professor of Electrical and Computer Engineering.. He has held short-term visiting positions at the Institut National de Recherche en Informatique et en Automatique, Sophia Antipolis, France, and at Tampere University of Technology, Finland. His research interests are in image and signal processing, specifically hierarchical statistical models, fast estimation algorithms, nonlinear scale-spaces, and adaptive representations.

Dr. Pollak received a CAREER award from the National Science Foundation in 2001. He is an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING, a member of the IEEE Signal Processing Society's Technical Committee on Signal Processing Theory and Methods, and the Chair of the Signal Processing Chapter of the Central Indiana Section of the IEEE. He is a Co-Chair of the SPIE/IS&T Conference on Computational Imaging.



Tak-Shing Wong received the B.Eng. degree in computer engineering and the M.Phil. degree in electrical and electronic engineering from the Hong Kong University of Science and Technology in 1997 and 2000, respectively. He is currently pursuing the Ph.D. degree at the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN.

His research interests are in image processing and document image analysis.



Charles A. Bouman (S'86–M'89–SM'97–F'01) received the B.S.E.E. degree from the University of Pennsylvania, Philadelphia, in 1981, the M.S. degree in electrical engineering from the University of California, Berkeley, in 1982, and the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ.

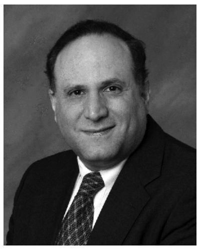
From 1982 to 1985, he was a full staff member at the Massachusetts Institute of Technology's Lincoln Laboratory, Lexington. In 1989, he joined the faculty of Purdue University, West Lafayette, IN, where he holds the rank of Professor with a primary appointment in the School of Electrical and Computer Engineering and a secondary appointment in the School of Biomedical Engineering. His research focuses on the use of statistical image models, multiscale techniques, and fast algorithms in applications including medical and electronic imaging.

Dr. Bouman is a Fellow of the American Institute for Medical and Biological Engineering (AIMBE), a Fellow of the society for Imaging Science and Technology (IS&T), a member of the SPIE professional society, a recipient of IS&T's Raymond C. Bowman Award for outstanding contributions to digital imaging education and research, and a University Faculty Scholar of Purdue University. He is currently the Editor-in-Chief Elect of the IEEE TRANSACTIONS ON IMAGE PROCESSING and a member of the Steering Committee for the IEEE TRANSACTIONS ON MEDICAL IMAGING. He has been an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING and the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. He has also been Co-Chair of the 2006 SPIE/IS&T Symposium on Electronic Imaging, Co-Chair of the SPIE/IS&T conferences on Visual Communications and Image Processing 2000 (VCIP), a Vice President of Publications and a member of the Board of Directors for the IS&T Society, and he is the founder and co-chair of the SPIE/IS&T conference on Computational Imaging.

Mary P. Harper (SM'02) received the M.Sc. and Ph.D. degrees in computer science from Brown University, Providence, RI, in 1986 and 1990, respectively.

In 1989, she joined the faculty of Purdue University, West Lafayette, IN, where she currently holds the rank of Professor in the School of Electrical and Computer Engineering. Dr. Harper's research focuses on computer modeling of human communication with a focus on methods for incorporating multiple types of knowledge sources, including lexical, syntactic, prosodic, and, most recently, visual sources. Recent research involves the integration of speech and natural language processing systems, the integration of speech, gesture, and gaze, and the utilization of hierarchical structure to improve the classification accuracy of documents and images.

Dr. Harper was a speaker for the IEEE Computer Society Distinguished Visitors Program and Chapter Tutorial Program from 1997 through 2000. She is currently an Associate Editor for the IEEE TRANSACTIONS ON SPEECH, AUDIO, AND LANGUAGE PROCESSING and sits on the IEEE Speech Technical Committee of the IEEE Signal Processing Society. She also sits on the Board for the North American Chapter of the Association for Computational Linguistics (NAACL).



Jeffrey M. Siskind received the B.A. degree in computer science from the Technion—Israel Institute of Technology, Haifa, in 1979, and the M.S. and Ph.D. degrees in computer science from the Massachusetts Institute of Technology, Cambridge, in 1989 and 1992, respectively.

He did a postdoctoral fellowship at the Institute for Research in Cognitive Science, University of Pennsylvania, Philadelphia, from 1992 to 1993. He was an Assistant Professor at the Department of Computer Science, University of Toronto, Toronto, ON, Canada, from 1993 to 1995; a Senior Lecturer at the Department of Electrical Engineering, the Technion, in 1996; a Visiting Assistant Professor at the Department of Computer Science and Electrical Engineering, University of Vermont, Burlington, from 1996 to 1997; and a Research Scientist at NEC Research Institute, Inc., from 1997 to 2001. He joined the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, in 2002, where he is currently an Associate Professor. His research interests include machine vision, artificial intelligence, cognitive science, computational linguistics, child language acquisition, and programming languages and compilers.