MODELING AND ESTIMATION OF SPATIAL RANDOM TREES
WITH APPLICATION TO IMAGE CLASSIFICATION

I. Pollak, J.M. Siskind, M.P. Harper, and C.A. Bouman

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

ABSTRACT

A new class of multiscale multidimensional stochastic processes
called spatial random trees is introduced. The model is based
on multiscale stochastic trees with stochastic structure as well as
stochastic states. Procedures are developed for exact likelihood
calculation, MAP estimation of the process, and estimation of the
parameters of the process. The new framework is illustrated through
a simple binary image classification problem.

1. INTRODUCTION

In this work, we develop a new class of multiscale stochastic mod-
els for multidimensional signals that we call spatial random trees
(SRTs). Similarly to [1, 2], our models are stochastic processes on
trees with each leaf corresponding to a single sample. Our key in-
novation, however, is that the tree structure itself is random and is
generated by a probabilistic grammar [3].

Probabilistic grammars have been widely used in natural-lang-
guage processing, for example, to model the structure of sentences
[4]. The concept of probabilistic grammar is based on the notion of
branching stochastic processes which have been used in studying
population dynamics since 1845 [5–7]. These problems have been
posed either in 1-D where the objects under consideration, for ex-
ample, words in sentences, are arranged linearly; or even in “0-D”
where the arrangement of objects, such as molecules of different
types in a population of particles, does not matter. Recently, there
have been efforts to apply probabilistic grammars to 2-D problems
such as optical character recognition [8].

These developments have motivated SRTs—our new general
framework for modeling multidimensional signals with probabilis-
tic grammars. This framework is described in Section 2 and is the
central contribution of this paper. For simplicity, we restrict our
exposition of SRTs to 2-D, but the generalization to an arbitrary
number of dimensions is straightforward.

With our framework, we obtain exact algorithms for perform-
ing the three fundamental tasks required of such models: comput-
data likelihoods; finding the MAP estimate of both the tree
structure and the tree states; and computing the parameter up-
dates required for each iteration of the EM algorithm [9] used to
train the model. These resulting algorithms—collectively termed
the Center-Surround algorithm—are described in Section 3. They
are an extension of—and were inspired by—the Forward-Backward
algorithm [4, 10, 11] for 1-D probabilistic grammars.

While extensive experiments with real data are beyond the
scope of this paper, we include a simple synthetic example in Sec-
2. SPATIAL RANDOM TREES

We consider images defined on an $M_1 \times M_2$ rectangular domain
illustrated in Fig. 1. In other words, an image $\mathbf{u}$ is an $M_1 \times M_2$
matrix of numbers. The rectangular domain whose upper left cor-
nor is $p = (p_1, p_2)$ and whose lower right corner is $q = (q_1, q_2)$ is
denoted $\Box_{M_1}$. For $p = (p_1, p_2)$, we write $u_p$ and $\Box_{pp} = \Box_p = p$
to denote the value and location, respectively, of the pixel at the
intersection of row $p_1$ and column $p_2$. We abbreviate $\mathbf{z} = (1, 1)$
and $M = (M_1, M_2)$, so that the whole domain of definition of
image $\mathbf{u}$ is $\Box_{1, M}$.

2.1. Probabilistic Grammars and Spatial Random Trees

SRTs model images with binary (dyadic) trees whose leaves are
image pixel locations, as illustrated in Fig. 2(a,b). A sample path
of an SRT is a (deterministic) tree, i.e., a triple $(\mathcal{V}, E, x)$ consist-
ing of a set $\mathcal{V}$ of all vertices, a set $E$ of all edges, and a mapping
$x$ which associates a state $x_v$ to every vertex $v$. We distinguish
between two types of states: the states corresponding to the im-
age pixel values which can only appear at the leaf vertices of the

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig1.png}
\caption{An illustration of our notation for images.}
\end{figure}
tree, and the “hidden” states corresponding to the remaining vertices of the tree. Any state which can occur at a leaf vertex (i.e., any possible pixel value) is called a terminal state, and the set of all terminal states is denoted by $T$. Any possible state for an internal vertex (i.e., a vertex which is not a leaf) is called a nonterminal state, and the set of all nonterminal states is denoted by $N$.

The yield of any internal vertex $\alpha$, denoted $Y(\alpha)$, is the set of all leaf descendants of $\alpha$. In our model, the yield of every internal vertex of a tree is a rectangular region of the image. Every internal vertex whose yield is a single pixel $\square_p$ is required to have a single child–pixel location $\square_{p1}$ with a terminal state which is the image value at that pixel, $u_p$. If the parent of $\square_p$ has state $j$, we describe this transition as $j \rightarrow u_p$. Following the terminology of natural-language processing, we call any transition of the form $j \rightarrow u$ with $j \in N$ and $u \in T$, a terminal production.

We moreover impose that unless the yield $\square_{p1}$ of an internal vertex $\alpha$ is a single pixel, $\alpha$ must have two children which are internal vertices with disjoint yields such that the union of the yields is equal to the yield of $\alpha$. In this case, one further restriction is that the two children be an ordered pair, with the upper left corner $\square_{p1}$ falling into the yield of the first child and the lower right corner $\square_{p2}$ falling into the yield of the second child. An equivalent explanation of these requirements is that there are the following possibilities for the yields of the children $\beta$ and $\gamma$ of $\alpha$:

(i) $Y(\beta) = \square_{p1(d+1),q2}$ and $Y(\gamma) = \square_{p1,d+1,q2}$ for some $d \in \{p1, \ldots, q1 - 1\}$, as illustrated in Fig. 3(a).

(ii) $Y(\beta) = \square_{p1(d+1),q2}$ and $Y(\gamma) = \square_{p1,d+1,q2}$ for some $d \in \{p1, \ldots, q1 - 1\}$, as illustrated in Fig. 3(b).

Each nonterminal production $j \rightarrow k, \ell$ is assigned probability $P_{\text{prod}}(j \rightarrow k, \ell)$, and each terminal production $j \rightarrow u$ is assigned probability $P_{\text{prod}}(j \rightarrow u)$, in such a way that the following normalization equations are satisfied:

$$\sum_{\alpha, k, \ell} P_{\text{prod}}(j \rightarrow k, \ell) + \sum_{u} P_{\text{prod}}(j \rightarrow u) = 1, \quad \forall j \in N.'$$

In our model, the state of the root vertex can be any nonterminal state $j \in N$ with probability $P_{\text{root}}(j)$ where

$$\sum_{j \in N'} P_{\text{root}}(j) = 1.$$

The probability of any tree $T$ is then defined to be the product of the root state probability and the probabilities of all the productions that are involved in generating $T$. Denoting the set of all internal vertices of $T$ by $V_{\text{int}}$, the root of $T$ by $\rho$, and the production applied at $\alpha$ by $\Lambda_\alpha$, we have:

$$P(T) \triangleq P_{\text{root}}(\rho_0) \prod_{\alpha \in V_{\text{int}}} P_{\text{prod}}(\Lambda_\alpha).$$

**Definition 1.** The stochastic process defined by the probabilistic grammar with productions (1,2), is called a spatial random tree (SRT).
2.2. Generating Images from the Grammar of Eqs. (1,2)

Note that a sequence of productions from Eqs. (1,2) may generate a tree whose leaves are not arranged in an \( M_1 \times M_2 \) rectangle. For example, while the tree of Fig. 2(c) is consistent with Eqs. (1,2), the “image” it produces is not defined on an \( M_1 \times M_2 \) grid, for any \( M_1 \) and \( M_2 \). In addition to some desired trees such as the tree of Fig. 2(a), our grammar generates undesired trees. It is moreover unclear whether there may be several images corresponding to the same desired tree. In the previous section, we defined a probability for each tree; what we would like, however, is a probabilistic model for images. We therefore need to resolve the issue of unambiguously associating an image with every tree.

Fortunately it turns out that if a tree does produce an image, that image is unique.

**Definition 2 (Admissible trees).** Let \( T \) be a tree generated by the grammar of Eqs. (1,2). Let \( \mathcal{V}_{\text{int}} \) be the set of its internal vertices, and let \( \rho \) be its root vertex. Suppose that there exist a pair of positive integers \( M = (M_1, M_2) \), and a bijective function
\[
\mathfrak{g} : \mathcal{Y}(\rho) \rightarrow \square_{1,M}
\]
which uniquely maps each leaf of the tree to a location in an \( M_1 \times M_2 \) grid, and which has the following property:

*The yield of each internal vertex of the tree is mapped to a rectangular region. More formally,*
\[
\forall \alpha \in \mathcal{V}_{\text{int}} \quad \exists p, q \text{ such that } [\mathfrak{g}(\beta) | \beta \in \mathcal{Y}(\alpha)] = \square_{p,q}.
\]
*We then say that \( T \) is an admissible tree, and \( \mathfrak{g} \) is an associated admissibility function.*

The following theorem, which we state without proof, shows that if the yield of a tree can be mapped to an image grid in a manner described above and illustrated in Fig. 2(a,b), such a mapping is unique.

**Theorem 1 (Admissibility Theorem).** If a tree \( T \) is admissible, there is a unique admissibility function for \( T \).

2.3. Probability Model for Images

Suppose now that we have an \( M_1 \times M_2 \) image \( \mathbf{u} = \mathbf{u}_{1,M} \), and an admissible tree \( T \). If the yield of \( T \) is \( \square_{1,M} \) and the states of the leaves are \( \mathbf{u}_{1,M} \), we say that the tree \( T \) generates the image \( \mathbf{u} \). We define the event \( \Omega_\alpha \), to be the set of all admissible trees that generate the image \( \mathbf{u} \). The term **probability of image** \( \mathbf{u} \) (denoted \( P(\mathbf{u}) \)) is shorthand for the probability of the set \( \Omega_\alpha \). Note that \( P(\mathbf{u}) \) does not, in general, define a probability distribution on the set of all images since the set of all admissible trees is not required to have unit probability.

3. SPATIAL RANDOM TREES AND INFERENCE

Our framework of SRTs admits recursive algorithms for likelihood calculation and for the estimation of the MAP (maximum a posteriori probability) tree. The EM algorithm [9] can moreover be adapted to search for the parameter values which maximize the likelihood of an image or a set of images. These algorithms are collectively termed the **Center-Surround algorithm**. The Center-Surround algorithm is based on recursive calculations involving center and surround probabilities which we presently describe.

For every rectangular region \( \square_{p,q} \) of an image \( \mathbf{u} \), we define the center probability \( c^j_{p,q} \) to be the probability of all admissible trees that generate the subimage \( \mathbf{u}_{p,q} \) and whose root state is \( j \). In other words, the center probability is the conditional probability of subimage \( \mathbf{u}_{p,q} \) given the event \( \Omega_j \) where \( \Omega_j \) is the set of all trees with root state \( j \): \( c^j_{p,q} = P(\mathbf{u}_{p,q}, \Omega_j) \). In particular, the conditional probability of the whole image given \( \Omega_j \) is \( c^j_{1,M} \). Therefore, the probability of image \( \mathbf{u} \) can be easily computed if the center probabilities \( c^j_{1,M} \) are known for all possible root states \( j \in N \):

\[
P(\mathbf{u}) = \sum_{j \in N} c^j_{1,M} P_{\text{root}}(j).
\]  

(3)

The following proposition, illustrated in Fig. 3, gives a recursive algorithm for the computation of \( c^j_{1,M} \). It takes advantage of the fact that any center probability for a rectangle containing multiple pixels can be expressed in terms of the center probabilities for smaller rectangles. Note that the first term of the recursion formula below corresponds to summing over all possible horizontal splittings (Fig. 3(a)), and the second term corresponds to the vertical splittings (Fig. 3(b)).

**Proposition 1.** For any nonempty rectangular domain \( \square_{p,q} \subset \square_{1,M} \) with \( p \neq q \), and any \( j \in N \),

\[
c^j_{p,q} = \sum_{d=p}^{q-1} \sum_{h \in N} \sum_{l \in N} P_{\text{root}}(j \rightarrow k, l) c^{h,k}_{p+d+1,p+q} c^{l}_{d+1,q} + \sum_{d=p}^{q-1} \sum_{h \in N} \sum_{l \in N} P_{\text{root}}(j \rightarrow k, l) c^{h,k}_{p,q} c^{l}_{p+1,d} c^{l}_{p+1,q}.
\]

For any \( p \in \square_{1,M} \) and any \( j \in N \),

\[
c^j_{p} = P_{\text{root}}(j \rightarrow u_p).
\]

Combining Proposition 1 with Eq. (3) gives a recursive algorithm for computing the probability \( P(\mathbf{u}) \) of image \( \mathbf{u} \).

The probability \( P(\mathbf{u}) \) can also be recursively calculated using the surround probabilities \( s^j_{p,q} \). Each surround probability gives the probability of the image region surrounding \( \square_{p,q} \). The combination of these two recursions makes it possible to perform one iteration of the EM procedure for estimating the parameters of the SRT from data. Due to space constraints, we are unable to describe the details of the training algorithm in this paper. It will be published elsewhere.

There also exists a dynamic programming algorithm for MAP tree estimation—i.e. for extracting the most probable tree \( \Omega_\alpha \) for a given image \( \mathbf{u} \). The recursive formulas are a simple variant of the center recursion of Proposition 1, with “\( \sum \)” replaced by “\( \max \)”. The probability of the most probable tree in \( \Omega_\alpha \) with root state \( j \) is denoted \( c^j_{p,q} \). The base case is:

\[
g^j_{p,q} = P_{\text{root}}(j \rightarrow u_p).
\]

We recursively calculate \( g^j_{p,q} \) for any rectangle in terms of probabilities associated with smaller rectangles:

\[
g^{j,k}_{p,q} = \max_{d,l} P_{\text{root}}(j \rightarrow k, l) g^{h,k}_{p+d+1, p+q} g^{l}_{d+1,q},
\]

\[
g^{j,k}_{p,q} = \max_{d,l} P_{\text{root}}(j \rightarrow k, l) g^{h,k}_{p,q} g^{l}_{p+1,d} g^{l}_{p+1,q},
\]

\[
g^{j}_{p,q} = \max(g^{j,k}_{p,q}, s^j_{p,q}).
\]
We in addition store the four-tuple of parameters \((o, k, l, d)\) which have led to the maximal \(g_{pq}^o\) where \(o \in \{h, v\}\) stands for the split orientation. We call this four-tuple \(f_{pq}^i\). If \(g_{pq}^o > g_{pq}^v\), then

\[
\begin{align*}
  f_{pq}^i &= (h, \arg\max_{k,l,d} P_{\text{pred}}(j \xrightarrow{k,\ell} h) g_{pq}^{h(k,l,d)} \gamma_{d+1 \times g_{pq}^{1}}(j)).
\end{align*}
\]

Otherwise,

\[
\begin{align*}
  f_{pq}^i &= (v, \arg\max_{k,l,d} P_{\text{pred}}(j \xrightarrow{k,\ell} v) g_{pq}^{v(k,l,d)} \gamma_{d+1 \times g_{pq}^{1}}(j)).
\end{align*}
\]

If the maximum is not unique, we can choose an arbitrary maximizing four-tuple. The probability of the MAP tree \(\hat{T}\) is calculated from the \(g\) variables,

\[
P(\hat{T}) = \max_{j \in N} g_{1}^{j} P_{\text{pred}}(j),
\]

and \(\hat{T}\) itself is constructed from the list of the \(f\) variables.

## 4. EXPERIMENTAL EXAMPLE

We now apply our likelihood computation algorithm of Section 3 to classifying binary images of noisy digits. Our data set consists of the ten digits from the X WINDOWS 9x15 font whose characters are 10 x 7 pixel images, placed at various locations on a white 14 x 11 background. These images are corrupted by synthetic noise which independently flips every pixel with probability \(\varepsilon\).

For each level of noise \(\varepsilon\) and each digit \(k = 0, 1, \ldots, 9\), a probabilistic grammar \(\mathcal{G}_{h,\varepsilon}\) was obtained through a combination of automatic training via the EM algorithm and manually writing down certain productions and their probabilities. For several noise levels in the range \(0 \leq \varepsilon \leq 0.2\), we conducted 900 classification experiments with noisy digit images. Each of the 900 images was classified by calculating its likelihoods with respect to the ten grammars \(\mathcal{G}_{0,\varepsilon}, \ldots, \mathcal{G}_{9,\varepsilon}\). Classifying each image took about 3 seconds on an 800 MHz Pentium III processor.

Our experiments are summarized in Fig. 4 which shows a plot of our estimates of the correct classification probability as a function of the noise level \(\varepsilon\), from the noise-free case \(\varepsilon = 0\) to the extremely noisy case of \(\varepsilon = 0.2\). This latter case corresponds to an average of about 31 incorrect pixels per 14 x 11 image, which, as shown in Fig. 5, makes some images difficult to recognize for a human. The plot in Fig. 4 demonstrates excellent performance of our algorithm and graceful degradation for very noisy images.

## 5. CONCLUSIONS

We have presented general methods for computing the likelihood of an observation of a multidimensional random field, and for estimating both the structure and the states of a stochastic tree from such an observation. We refer to the associated new class of multiscale processes as spatial random trees. These models can be used to classify and interpret images, and they can be trained using the EM algorithm. A simple experiment illustrates their potential value in signal-processing applications.

## 6. ACKNOWLEDGMENTS

We would like to thank Yan Huang, Bill Nagel, James Sherman, and Eric Théa for developing the code for the Center-Surround algorithm and running the experiment of Section 4.

## 7. REFERENCES


