Varying Complexity in Tree-Structured Image Distribution Models

Clay Spence, Lucas C. Parra, and Paul Sajda

***Abstract*—Probabilistic models of image statistics underlie many approaches in image analysis and processing. An important class of such models have variables whose dependency graph is a tree. If the hidden variables take values on a finite set, most computations with the model can be performed exactly, including the likelihood calculation, training with the EM algorithm, etc. Crouse *et al.* developed one such model, the *hidden Markov tree* (*HMT*). They took particular care to limit the complexity of their model. We argue that it is beneficial to allow more complex tree-structured models, describe the use of information theoretic penalties to choose the model complexity, and present experimental results to support these proposals. For these experiments, we use what we call the hierarchical image probability (HIP) model. The differences between the HIP and the HMT models include the use of multivariate Gaussians to model the distributions of local vectors of wavelet coefficients and the use of different numbers of hidden states at each resolution. We demonstrate the broad utility of image distributions by applying the HIP model to classification, synthesis, and compression, across a variety of image types, namely, electrooptical, synthetic aperture radar, and mammograms (digitized X-rays). In all cases, we compare with the HMT.**

***Index Terms*—Bayesian network, classification, compression, hidden Markov tree (HMT), hidden variables, image model, minimum description length (MDL), synthesis, tree-structured belief network (TSBN).**

I. Introduction

ONE of the primary goals of computational vision is to develop mathematical descriptions of the visual world. Relatively recent work on statistical regularities in natural images [1]–[3] has led to these mathematical descriptions being cast as probabilistic models. There are several fundamental reasons why such models are an attractive framework for describing natural images. First, theories about the ecological basis of biological vision need to take into account the statistical regularities in visual scenes [4], [5]. A compact probabilistic description of visual scenes may, therefore, provide important insight into the representations and underlying mechanisms used in biological vision. We will focus on a second reason: essentially all image analysis can be formulated within the context of a probabilistic model. Assume one can construct a model for the probability distribution $\Pr(I \mid C)$ of images I given that they belong to class C. Such a model is often referred to as a *generative* model, as opposed to a *discriminative* model of the probability $\Pr(C \mid I)$ of the class given the image. Generative models enable a wide range of applications, including, but not limited to the following.

- Classification: Using Bayes' rule, we can compute the class conditional likelihood of new images under the model $\Pr(C \mid I) = \Pr(I \mid C) \Pr(C) / \Pr(I)$. Thus, if we have a set of generative models, one for each of a set of image classes, a discriminative model is implied.[1]
- Synthesis: By sampling the model for $\Pr(I \mid C)$, we can generate new images for class C.
- Compression: In principle, a good model of $\Pr(I \mid C)$ for the class of images provides a good code for compressing images of this class.

We do not mean merely that one *class* of models can be used for all of these applications. Rather, we can fit a single model to one set of images and use that model for a variety of applications. This flexibility is possible because a generative model allows us to both evaluate the likelihood of the image and sample from the distribution. While generative models are often used for other data types, their use for images is challenging due to the high dimensionality and rich structure of the data.

An important class of generative image models captures multiscale, hierarchical dependencies of structure in images through the use of tree-structured arrangements of the dependencies between the model's variables, with the root at the coarsest resolution and the leaves at the finest. These have been called tree-structured belief networks (TSBNs) [6]. Trees have the disadvantage that they are not invariant to translations or rotations, but they make the graph of dependencies acyclic, so that computations are greatly simplified. In addition, the tree naturally captures the persistence of structure across scale that is commonly observed in images, and provides relatively direct dependencies between distant parts of the image.

If the hidden variables in a tree-structured model take values in a finite set, most computations can be carried out exactly. The earliest such model specifically for distributions of natural images was the hidden Markov tree (HMT) of Crouse *et al.*[7].[2] The HMT uses binary-valued hidden "state" variables at the nodes of the tree. These states determine the range of coefficient

Manuscript received June 1, 2004; revised January 14, 2005. This work was supported in part by the U.S. Army Medical Research and Material Command (DAMD17-98-1-8061) and in part by the DoD Multidisciplinary University Research Initiative (MURI) program administered by the Office of Naval Research under Grant N00014-01-1-0625. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Theirry Blu.

C. Spence is with the Sarnoff Corporation, Princeton, NJ 08540 USA (e-mail: cspence@sarnoff.com).

L. C. Parra is with the Department of Biomedical Engineering, City University of New York, New York, NY 10031 USA (e-mail: parra@ccny.cuny.edu).

P. Sajda is with the Department of Biomedical Engineering, Columbia University, New York, NY 10027 USA (e-mail: ps629@columbia.edu).

Digital Object Identifier 10.1109/TIP.2005.860601

[1]To evaluate this expression, we also need to know $\Pr(C)$ for all C, which is merely a finite set of real numbers. We can then compute $\Pr(I)$ as $\sum_C \Pr(I \mid C) \Pr(C)$. The converse operation, computing $\Pr(I \mid C)$ as $\Pr(C \mid I) \Pr(I) / \Pr(C)$, requires that we learn $\Pr(I)$.

[2]Although Crouse *et al.* first coined the term "hidden Markov tree" for their model, it is a good literal description for many of the tree-structured models in the literature.

magnitudes by specifying which of two variances is used in a normal distribution of the coefficient at that node. Thus, the distribution of a coefficient is a two-component mixture of Gaussians model, with state probabilities that depend on the state at the parent node in the tree. For images, the HMT uses a separate hidden variable tree for each type of wavelet subband, i.e., the horizontal, vertical, and diagonal high-pass bands. These models have been successfully applied to several problems, such as image enhancement and texture segmentation [8], [9].

In designing the HMT, Crouse *et al.* took care to limit the complexity of the model, especially the number of parameters, in order to avoid over-fitting. The hidden variables are strictly a means to represent wavelet coefficient magnitude and its persistence across image scale. We suggest that the hidden variables can represent more local image structure. Using more than two values (as suggested by Crouse *et al.*) would allow the hidden variables to better fit the marginal distributions of single coefficients. If we use a single hidden variable at a location to condition the entire vector of local coefficients, that variable could also represent orientation.[3] In addition to orientation and scale, other local image structure can be expressed as relationships between elements of the coefficient vector, such as the presence of edges, lines, corners, more general parts of a texture, etc. Beyond local structure, a hidden label that is not at the leaf of the tree can represent common information among all of its descendants. Thus, much more of the inherent complexity in natural images could by represented in the hidden labels of TSBNs.

We contend that images can support the use of models with this additional complexity, since an image effectively presents us with many examples. A 256×256 pixel image has 65 536 pixels, 16 384 positions in the first level of a wavelet decomposition, 4 096 in the second, and so on. Although not entirely independent, each of these positions is another example for learning model parameters.

To explore fitting the complexity of a model to a set of images, we present what we call the hierarchical image probability (HIP) model. We allow HIP models to have different mixture components and different numbers of components at each level in the tree. However, we use the same components everywhere within a given level, a constraint that is often referred to as "tying" in the HMM literature [10], and is also used in the original HMT. So we tie across position but not scale. We choose the number of mixture components within each level with the minimum description length (MDL) criterion [11], [12]. Even with tying to reduce the number of parameters, we obtain more than ten thousand parameters when fitting to data sets with several tens of images. The test set results of Section III and the MDL criterion agree that the models with these very large numbers of parameters are optimal. The additional complexity allows HIP models to capture subtleties of images that are missed by the HMT.

In the following, we present the HIP model and its application to a variety of image analysis problems. In Section II, we present the model in detail, including training and MDL-based algorithms for choosing the numbers of mixture components. In Section III, we present the results of experiments with HIP models. For each of three very different data sets, we fit a model to images of a class and then use that model for several different applications. We show that MDL is effective in choosing a model's complexity so that it generalizes well. In all of the applications we compare with the HMT.

[3]The HMT's set of three binary variables at a node can represent some orientation information, though it is difficult without dependencies between these variables.

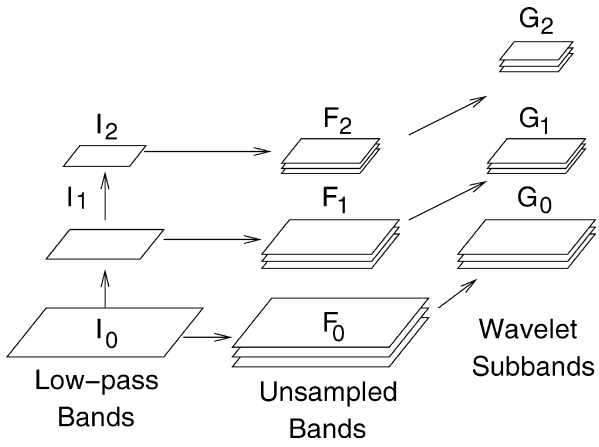

Fig. 1. Notation for variables in the HIP model. Shown are images from a wavelet decomposition, including low-pass bands I, subbands $\mathbf{G}$, and unsampled subbands $\mathbf{F}$.

II. HIP Model

In this section, we present the HIP model in some detail. In Section II-A, we give a basic presentation of the HIP model. We then discuss other tree-structured models used for image processing, and give a detailed comparison of the HIP model and the HMT (Section II-B). Finally, we present an EM algorithm for training HIP models (Section II-C) and procedures for architecture selection using MDL (Section II-D).

A. Basic Structure

To motivate the HIP model's structure we present it in terms of the conditional independence of the variables within it. A similar description would apply to most TSBNs.

We first represent structure in the image from different scales by applying a wavelet decomposition. The wavelets filters are applied to the image I_0 to obtain a set of images $\mathbf{F}_0$ (Fig. 1). These filters are band or high-pass, so the $\mathbf{F}_0$ have the low frequency content suppressed. The $\mathbf{F}_0$ and a low-pass filtered version of I_0 (filtered with the scaling function) are then subsampled to construct a set of smaller images or subbands $\tilde{\mathbf{G}}_0$, which consists of a low-pass subband I_1 and bandpass or high-pass subbands $\mathbf{G}_0$. We write this mapping from I_0 to $\tilde{\mathbf{G}}_0$ as $\tilde{\mathcal{G}}_0$.

If the wavelet transform is critical and invertible, this is a change of variables.[4] Consequently, we can write the distribution over images I as $\Pr(I_0) = |\tilde{\mathcal{G}}_0| \Pr(\tilde{\mathbf{G}}_0)$, since the determinant $|\tilde{\mathcal{G}}_0|$ is the Jacobian for the change of variables. We factor this and obtain $\Pr(I_0) = |\tilde{\mathcal{G}}_0| \Pr(\mathbf{G}_0 \,|\, I_1) \Pr(I_1)$. Repeating this change of variables on successive I_l and factoring gives

$$\Pr(I) = \left[\prod_{l=0}^{L} |\tilde{\mathcal{G}}_l| \Pr(\mathbf{G}_l \,|\, I_{l+1}) \right] \Pr(I_{L+1}). \tag{1}$$

[4]Note that there may be nonwavelet transforms that also satisfy these criteria and could be used instead.

We choose L so that $\mathbf{G}_L$ and I_{L+1} have a single pixel. Note that there are no assumptions needed to derive (1) other than that the wavelet transform is a change of variables. However, conditioning finer-scale structure on coarser structure is an intuitively appealing choice common to all TSBNs.

The factors in (1) are still likely to be very complex distributions. To further simplify, we factor each over position and reduce the conditioning variables to some local part of I_{l+1}. For the latter, we choose the vector of coefficients $\mathbf{f}_{l+1}(x)$ from position x in the unsampled images $\mathbf{F}_{l+1}$.[5] This vector describes local image structure at a coarser scale than $\mathbf{g}_l(x)$.[6] This gives

$$\Pr(\mathbf{G}_l \,|\, I_{l+1}) = \prod_{x \in I_{l+1}} \Pr(\mathbf{g}_l(x) \,|\, \mathbf{f}_{l+1}(x)) \tag{2}$$

$$\equiv \prod_x \Pr(\mathbf{g}_l \,|\, \mathbf{f}_{l+1}, x). \tag{3}$$

Here, we have introduced several notations. To indicate the set of positions within level l of the decomposition, we have written $x \in I_{l+1}$. We use I_{l+1} because that low-pass image has the same dimensions as $\mathbf{G}_l$. We will often write this simply as a product over x, since the range of x should be clear from context. For the vector of wavelet coefficients from $\mathbf{G}_l$ at position x, we write $\mathbf{g}_l(x)$. Finally, to simplify notation, we have moved the dependence on position x to the right in (3), writing it as if it were a conditioning variable. We will not be discussing the distribution of $\mathbf{g}(x)$ conditioned on $\mathbf{f}(x')$ for $x \neq x'$, so this should not cause confusion.

The factoring over position in (3) represents two assumptions. First, it assumes that the $\mathbf{g}_l(x)$ for different x are independent of each other given I_{l+1}, i.e., $\Pr(\mathbf{G}_l \,|\, I_{l+1}) = \prod_x \Pr(\mathbf{g}_l(x) \,|\, I_{l+1})$. Second, it assumes that $\mathbf{g}_l(x)$ depends only on $\mathbf{f}_{l+1}$ out of all of I_{l+1}.[7] These assumptions are not strictly correct for any but artificial images. For example, we can argue that the $\mathbf{g}_l(x)$ for different x are not independent, though the dependence between pairs decreases with distance [15]. Such dependencies can be caused by the presence of an object that is covered by a single texture. Observing this texture at one location tells us that other locations within the object's typical size are likely to share that texture. This will appear as a dependency between the coefficient vectors $\mathbf{g}_l$ at distant locations. If the object is not identifiable in I_{l+1}, this lower resolution image does not imply the presence of that texture. Therefore, conditioning on I_{l+1} does not make the coefficient vectors at different locations independent.

To reintroduce some of the dependencies that would be lost by these assumptions, we add a discrete hidden variable or *label* $a_l(x)$ at each position x in every level l. We use A_l to denote the image of these labels at level l. So, we are assuming that the $\mathbf{g}_l(x)$ are *conditionally* independent at different x given the hidden variable $a_l(x)$ and the observed $\mathbf{f}_{l+1}(x)$ (Figs. 2 and 3).

[5]Note that, for $l = L$, there is no $\mathbf{F}_{l+1}$, since I_{L+1} is a single pixel. In this case, we either model the joint distribution $\Pr(\mathbf{g}_L, I_{L+1})$, or use $\Pr(\mathbf{G}_L \,|\, I_{L+1}) = \Pr(\mathbf{g}_L)$, choosing not to condition on the single value of I_{L+1}.

[6]We could have used $\mathbf{g}_{l+1}$ at the parent of x instead, but chose $\mathbf{f}_{l+1}$ because it avoids some of the aliasing from which wavelets suffer [13], [14].

[7]We have sometimes included the value of of I_{l+1} at x, so $\mathbf{g}_l$ is also allowed to depend on local brightness as well as $\mathbf{f}_{l+1}$.

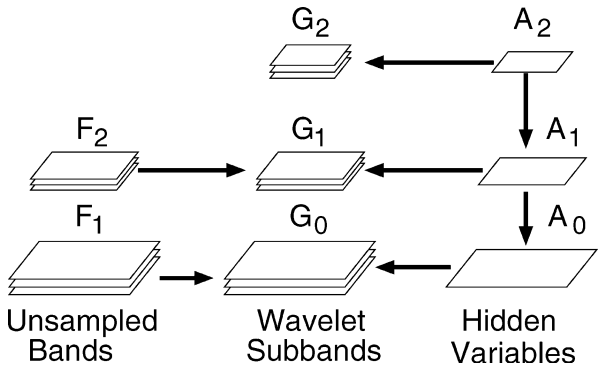

Fig. 2. Conditional dependencies between images in the HIP model. For simplicity we are not including the deterministic dependency of $\mathbf{F}_l$ on $\mathbf{G}_l$ and I_{l+1}.

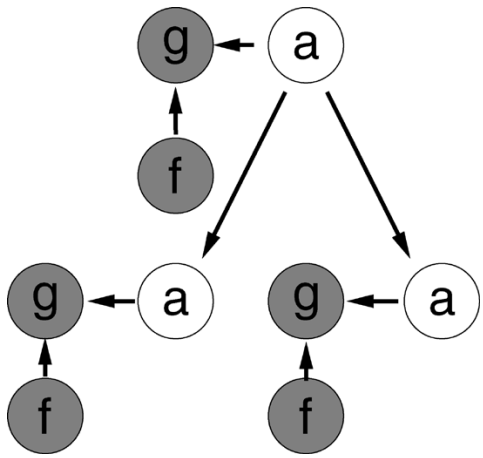

Fig. 3. Conditional dependencies between node variables in the HIP model. For simplicity we include only one parent and its children in a one-dimensional (1-D) model, and we do not include the deterministic dependency of $\mathbf{f}_l$ on $\mathbf{g}_l$ and i_{l+1}.

To carry the information about the dependencies between the gs across scale and position, we make $a_l(x)$ depend on the label at the parent position $\mathrm{Par}(x)$ in a quad-tree arrangement in the wavelet decomposition.[8] This introduces the factors $\Pr(a_l(x) \,|\, a_{l+1}(\mathrm{Par}(x))) \equiv \Pr(a_l \,|\, a_{l+1}, x)$. By this expression, we mean the probability of the label at x taking on the value a_l given that the value of the label at the parent position $\mathrm{Par}(x)$ is a_{l+1}. The probability does not depend on the position x, but we include it to make it clear that a_l depends only on the label a_{l+1} at the parent position. Furthermore, at each level we allow the labels to take on a different number of values N_{a_l}, which we will choose by fitting to the data. Within a level the same set of labels are used everywhere, that is, we tie across position. However, we do not tie across scale, so we allow the parameters at each level to be different.

Combining (1) and (3), and adding the hidden labels, gives

$$\Pr(I) = \left[\prod_{l=0}^{L-1} |\tilde{\mathcal{G}}_l| \sum_{A_l} \prod_{x \in I_{l+1}} \Pr(\mathbf{g}_l \,|\, \mathbf{f}_{l+1}, a_l, x) \Pr(a_l \,|\, a_{l+1}, x)\right] \times |\tilde{\mathcal{G}}_L| \sum_{A_L} \Pr(\mathbf{g}_L \,|\, a_L) \Pr(a_L) \Pr(I_{L+1}). \tag{4}$$

[8]In the quad-tree graph, the node at location $\mathrm{Par}(x) = (i, j)$ in level $l + 1$ has children at $x \in \{(2i, 2j), (2i + 1, 2j), (2i, 2j + 1), 2i + 1, 2j + 1)\}$ in level l.

With some assumptions and abuses of notation we can simplify this to

$$\Pr(I) \propto \sum_{A} \prod_{l=0}^{L} \prod_{x} \Pr(\mathbf{g}_l \,|\, \mathbf{f}_{l+1}, a_l, x) \Pr(a_l \,|\, a_{l+1}, x). \quad (5)$$

To obtain the latter equation, we drop the determinants, since for many applications we only need relative likelihoods. Also, when interpreting this expression, we need to remember that there are no quantities $\mathbf{f}_{L+1}$ or a_{L+1}, so we do not condition on these. Finally, we drop $\Pr(I_{L+1})$. Ignoring I_{L+1} should be adequate for some applications, or we could include it in the distribution for $\mathbf{g}_L$, giving $\Pr(\mathbf{g}_L, I_{L+1} \,|\, a_L) \Pr(a_L)$ for the $l = L$ factor.

Finally, we need a model for $\Pr(\mathbf{g}_l \,|\, \mathbf{f}_{l+1}, a_l, x)$. Following other TSBNs, we choose a multivariate Gaussian with a mean that depends linearly on $\mathbf{f}_{l+1}$, so that

$$\Pr(\mathbf{g}_l \,|\, \mathbf{f}_{l+1}, a_l, x) = \mathcal{N}(\mathbf{g}_l(x);\ \bar{\mathbf{g}}_{a_l} + M_{a_l} \mathbf{f}_{l+1}(x),\ \Lambda_{a_l}). \quad (6)$$

This makes the model distribution of $\mathbf{g}$ a mixture of Gaussians, with mixing components that depend on the label at the parent node via $\Pr(a_l \,|\, a_{l+1}, x)$. This choice is popular both because Gaussians are tractable and because mixtures of Gaussians are very general.

Thus, the set of the parameters of a HIP model includes the label probabilities $\Pr(a_l \,|\, a_{l+1})$ at each level, for each label pair a_l and a_{l+1}, and each child position relative to the parent. There are also the Gaussian mixture component parameters for each label value a_l at each level. These are the means $\bar{\mathbf{g}}_{a_l}$, correlations M_{a_l} with the unsampled parent coefficients, and covariance matrices Λ_{a_l}. All of these parameters can be different at each level, though they are common within a level, i.e., they are *tied* across position, but not scale.

Equations (5) and (6) specify the HIP model. In principle, sufficiently complex hidden variables could model any distribution, so that these models are completely general. In practice their generality is limited because there is only so much that can be learned from a finite data set, and the tree structure has limitations for images, as discussed earlier.

B. Other Image Models

Before focusing on other tree-structured models, note that there are several models without a tree structure. Examples include Markov random fields (MRFs) [16], [17], and the maximum entropy approach of Zhu *et al.* [18]. Many algorithms for texture analysis and synthesis are related to modeling image probability distributions [19]–[21], since the ability to synthesize implies a distribution. All of these models focus on sampling, so it is not straightforward to compute the probability of a given image. This can make applications like classification difficult.[9] In addition, it is not clear that these algorithms can deal with images that are not texture, e.g., those that have long-range variations in local texture.

[9]Note, however, that De Bonet and Viola [21] can apply their approach to classification, since they explicitly model the distribution of feature vectors of an image.

An early example of a tree-structured image model is the multiscale stochastic process (MSP) [22], [23]. MSPs capture dependency across scale by conditioning the feature values at one scale on values at the next lower resolution of the feature pyramid. Luettgen and Willsky [24], for example, apply a scale-space auto-regression (AR) model for the problem of texture discrimination. The continuous variables in the process are hidden and the observations are sums of these hidden variables, plus noise. The chief drawback of the MSP as an image distribution is that the resulting joint distribution is Gaussian. This is clearly not the case in natural images. Buccigrossi and Simoncelli [25], for example, have shown that the conditional distribution of neighboring image features can have high kurtosis, i.e., the joint distribution of the features are non-Gaussian, so the marginals are as well (see Fig. 8).

The random cascades of Wainwright *et al.* [3], [15] account for non-Gaussian feature distributions along with nontrivial dependencies across image scales. Wainwright *et al.* construct a multiscale auto-regressive model of hidden scale factors on a tree. The typical magnitudes or scales of the wavelet coefficients are given by these factors, since a coefficient at a node is the product of the factor and a normally-distributed random variable. This successfully reproduces a number of marginal and joint statistics of wavelet coefficients. A difficulty with this model is the need for approximate methods to fit the model parameters and evaluate likelihoods.

A number of tree-structured models have been developed in which the variables with dependencies between parent and child nodes take values on a finite set. In such models the computations can be performed exactly. Much of the work in this area is concerned with image segmentation [6], [26]–[28]. Although some of these models could be modified to work as distributions of the observed images, they are all mainly concerned with obtaining accurate segmentations. Typically the segmentation labels have the tree-structured dependency from one resolution to the next, and they condition the observed image values locally. In some cases, training is accomplished with given segmentation labels, so that these are observed rather than hidden.

The HIP model can be viewed as an elaboration of the original HMT model. The common attributes of both models include the following.

- Local dependencies are captured through models of the distributions of the local variables in a decomposition of the image.
- Nonlocal and interscale dependencies are captured with a set of discrete hidden variables whose dependency graph is a tree.
- Model parameters are optimized to match the natural image statistics using strict Maximum Likelihood.
- The models allow both evaluation of the likelihood and sampling from the distribution.

The HIP model differs from the original HMT in the following ways.

- In the HIP model, the coefficients of the different subbands at each node are modeled jointly, using a mixture of multivariate normal distributions. The original HMTs

model each coefficient separately with a mixture of univariate Gaussians.[10]

- The HIP model has one tree of hidden variables, whereas the HMT has a separate tree for each subband type.
- In the HIP model, the number of hidden states in each level is adjusted separately in an attempt to better fit the image distribution. The original HMTs have two states for each of the subband types.
- In the HIP model, each (parent) state conditions the likelihood of a (child) state with $\Pr(a_l \mid a_{l+1})$. We assign different such conditional probabilities to the different children depending on their location relative to the parent (upper-left, lower-right, etc.)
- The mean of each normal distribution depends on the corresponding coefficient vector in the unsampled wavelet coefficient subbands from the next coarser level (the HIP model resembles an MSP in this way.)

We feel that the use of a single tree of hidden variables and multivariate Gaussians for the vectors of coefficients at each location makes the HIP model conceptually simpler than the original HMT image model, at least for images as opposed to 1-D signals. However, HIP models typically have many more parameters, so by this measure, they can be much more complex.

C. Training the HIP Model With an EM Algorithm

We adjust the parameters of our model to match the statistics of a given set of images by using maximum likelihood (ML) parameter estimation. Like other tree-structured models, the structure of the model in (5) and illustrated in Fig. 1 permits the exact and efficient computation of all marginal probabilities required for the expectation-maximization (EM) algorithm [30]. Note that much of the previous work on tree-structured models cited in Section II-B also use an EM algorithm. It is, of course, especially similar to the EM algorithm used for the original HMT. A theoretical discussion of the fitting of some tree-structured likelihoods, including complexity-penalized fitting, was given by Kolaczyk and Nowak [31]. We, therefore, give a basic description and the full equations, without giving a complete derivation.

The algorithm first computes the expectations, over the hidden variables, of the log-likelihood for a given set of parameters and observations (E-step). Then, using these expectations, the likelihood is maximized with respect to the parameters of the model (M-step).

$$\text{E-step}: \quad Q(\theta \mid \theta^t) = \sum_{I,A} \Pr(A \mid I, \theta^t) \ln \Pr(I, A \mid \theta) \tag{7}$$

$$\text{M-step}: \quad \theta^{t+1} = \arg\max_{\theta} Q(\theta \mid \theta^t). \tag{8}$$

[10]However, Fan *et al.* use a single hidden variable tree in which the hidden state is a triplet of binary values, one for each subband type [29].

Here, we have summarized all parameters of the model in θ, and θ^t represents the values of the parameters in the current iteration step t. The sum over I is over all images in the training set.

The main challenge for this model lies in computing the expectations over the unknown labels. In this section only the resulting equations will be given. For the derivation of the probability propagation in this hierarchical model readers are referred to [32].

1) Maximization: We start with the M-step by inserting (5) into (7), as shown in (9) and (10), at the bottom of the page. Here, $\Pr(a_l, a_{l+1} \mid I, x, \theta^t)$ represents the marginal probabilities of pairs of labels from neighboring layers at position x for given image data and the current parameter values. The additive constant is due to the proportionality factors of (5). Assuming we know the probability $\Pr(a_l, a_{l+1}, \mid I, x, \theta^t)$ for all parent/child label pairs, a_l, a_{l+1}, we can search for the optimal parameters.

At this point we must insert expressions for $\Pr(a_l \mid a_{l+1}, x)$ and $\Pr(\mathbf{g}_l \mid \mathbf{f}_{l+1}, a_l, x)$ in order to make the set of parameters explicit and derive expressions for the M-step. As mentioned earlier we use the same parameters for all positions so that we obtain homogeneous behavior across the image, tying across position. However, we allow our model to have different parameters at different pyramid levels — we tie across position but not scale. We choose to parameterize $\Pr(a_l \mid a_{l+1}, x)$ as

$$\Pr(a_l \mid a_{l+1}) = \frac{\pi_{a_l, a_{l+1}}}{\sum_{a_l} \pi_{a_l, a_{l+1}}} \tag{11}$$

where the parameters $\pi_{a_l, a_{l+1}}$ are computed through the parameter updates in the M-step of the EM algorithm. This is a means for keeping $\Pr(a_l \mid a_{l+1})$ properly normalized (which could also have been done using Lagrange multipliers). The πs have an arbitrary scale factor, but the expressions given by the EM algorithm are well-defined in spite of this. Note also that we omit x in this notation as the parameterization is independent of the position within a layer (it does depend on the position of x relative to the parent node, but we drop this for the sake of brevity).

We have already given the expression for $\Pr(\mathbf{g} \mid \mathbf{f}, a)$ in (6). The parameter set is now defined as

$$\theta = \bigcup_{l=0}^{L} \{\pi_{a_l, a_{l+1}}, M_{a_l}, \bar{\mathbf{g}}_{a_l}, \Lambda_{a_l} | 0 \le a_l < N_a^l, 0 \le a_{l+1} < N_a^{l+1}\} \tag{12}$$

where N_a^l is the number of mixture components in level l.

With the choices (11) and (6) the M-step is easily solved. The maximum of (10) with respect to θ can be found by setting the derivatives with respect to the different parameters equal to zero

$$Q(\theta \mid \theta^t) = \sum_{I,A} \Pr(A \mid I, \theta^t) \sum_{l=0}^{L} \sum_{x} \ln \Pr(\mathbf{g}_l, a_l \mid \mathbf{f}_{l+1}, a_{l+1}, x, \theta) + \text{const} \tag{9}$$

$$= \sum_{I,l,x,a_l,a_{l+1}} \{\Pr(a_l, a_{l+1} \mid I, x, \theta^t) \ln \Pr(\mathbf{g}_l, a_l \mid \mathbf{f}_{l+1}, a_{l+1}, x)\} + \text{const} \tag{10}$$

and solving for the corresponding parameter. For $\pi^{t+1}_{a_l,a_{l+1}}$, we find

$$\pi^{t+1}_{a_l,a_{l+1}} = \sum_{I,x} \Pr(a_l, a_{l+1} \,|\, I, x, \theta^t). \tag{13}$$

For the remaining update equations, we define the following weighted average:

$$\langle X \rangle_{t,a_l} = \frac{\sum_{I,x} \Pr(a_l \,|\, I, x, \theta^t) X(I,x)}{\sum_{I,x} \Pr(a_l \,|\, I, x, \theta^t)}. \tag{14}$$

The weights $\Pr(a_l \,|\, I, x, \theta^t)$ represent the the marginal probabilities of finding label value $a_l(x)$ at position x given I and the current parameter values. The update equations are

$$M^{t+1}_{a_l} = \left(\langle \mathbf{g}_l \mathbf{f}^T_{l+1} \rangle_{t,a_l} - \langle \mathbf{g}_l \rangle_{t,a_l} \langle \mathbf{f}^T_{l+1} \rangle_{t,a_l} \right) \times \left(\langle \mathbf{f}_{l+1} \mathbf{f}^T_{l+1} \rangle_{t,a_l} - \langle \mathbf{f}_{l+1} \rangle_{t,a_l} \langle \mathbf{f}^T_{l+1} \rangle_{t,a_l} \right)^{-1} \tag{15}$$

$$\bar{\mathbf{g}}^{t+1}_{a_l} = \langle \mathbf{g}_l \rangle_{t,a_l} - M^{t+1}_{a_l} \langle \mathbf{f}_{l+1} \rangle_{t,a_l} \tag{16}$$

and

$$\Lambda^{t+1}_{a_l} = \langle \left(\mathbf{g}_l - M^{t+1}_{a_l} \mathbf{f}_{l+1} \right) \left(\mathbf{g}_l - M^{t+1}_{a_l} \mathbf{f}_{l+1} \right)^T \rangle_{t,a_l} - \bar{\mathbf{g}}^{t+1}_{a_l} (\bar{\mathbf{g}}^{t+1}_{a_l})^T. \tag{17}$$

At coarse scales, there may not be enough data to fit full covariance matrices. In this case, we can assume diagonal M and Λ. The densities $\mathcal{N}(\mathbf{g}, M_a \mathbf{f} + \bar{\mathbf{g}}_a, \Lambda_a)$ then factor into individual densities for each component of $\mathbf{g}$, and we can replace (15)–(17) with scalar versions for each component of $\mathbf{g}$.

2) Expectation: As is the case for other tree-structured image models, the expectation step is an elaboration of the basic forward-backward algorithm that has appeared in several fields. This has been presented by Baum *et al.* [33] and in terms of belief networks by Pearl [34]. We need to compute the marginal probabilities of pairs of labels from neighboring layers $\Pr(a_l, a_{l+1} \,|\, I, x, \theta^t)$.

This computation will be essentially the same as propagating the probabilities of observations of the entire pyramid to a particular junction of label pairs. Probabilities first propagate upward, and then downward to a particular label pair. During the propagation, we marginalize over the other labels. We recursively define quantities u and d, representing the upward and downward propagating probabilities

$$u_l(a_l, x) = \Pr(\mathbf{g}_l \,|\, \mathbf{f}_{l+1}, a_l, x) \prod_{x' \in Ch(x)} \tilde{u}_{l-1}(a_l, x') \tag{18}$$

$$\tilde{u}_l(a_{l+1}, x) = \sum_{a_l} \Pr(a_l \,|\, a_{l+1})\, u_l(a_l, x) \tag{19}$$

$$d_l(a_l, x) = \sum_{a_{l+1}} \Pr(a_l \,|\, a_{l+1})\, \tilde{d}_l(a_{l+1}, x) \tag{20}$$

$$\tilde{d}_l(a_{l+1}, x) = \frac{u_{l+1}(a_{l+1}, \mathrm{Par}(x))}{\tilde{u}_l(a_{l+1}, x)} d_{l+1}(a_{l+1}, \mathrm{Par}(x)). \tag{21}$$

The upward recursion (18) and (19) is initialized at $l = 0$ with $u_0(a_0, x) = \Pr(\mathbf{g}_0 \,|\, \mathbf{f}_1, a_0, x)$ and ends at $l = L$ with $\tilde{u}_L = \sum_{a_L} \Pr(a_L)\, u_L(a_L)$. Remember there is no label a_{L+1}, or one can think of it as a label with a single value. Also we have assumed that this level has only one location x. The image distribution is then $\Pr(I) \propto \tilde{u}_L$. If we choose level L with more than one location, the total image probability is the product over x of all $\tilde{u}_L(x)$

$$\Pr(I \,|\, \theta^t) \propto \prod_{x \in I_{L+1}} \tilde{u}_L(x) = u_{L+1}. \tag{22}$$

This actually applies whether there is one or more pixel in I_{L+1}, so we will use this.

The downward recursion (20) and (21) starts with (21) at $l = L$ with $d_{L+1}(a_{L+1}, x) = d_{L+1}(x) = 1$, and ends at $l = 0$ with (20).

With these quantities we can compute the marginal of a parent-child label pair as

$$\Pr(a_l, a_{l+1} \,|\, I, x, \theta^t) = u_l(a_l, x) \tilde{d}_l(a_{l+1}, x) \Pr(a_l \,|\, a_{l+1}) \tag{23}$$

$$\Pr(a_l \,|\, I, x, \theta^t) = u_l(a_l, x) d_l(a_l, x) \tag{24}$$

where the computations (18)–(24) in the E-step at iteration t are performed with fixed parameters θ^t.

Note that several quantities in these computations can take on extreme values, since we are dealing with densities in a very high-dimensional space. Therefore, when implementing these algorithms all of the probability densities and the us and ds need to be represented as logarithms.

D. MDL for Architecture Selection

We now need some criterion for choosing the number of mixture components at each level. We have found that the HIP model is well-suited for use with information-theoretic criteria like MDL or Akaike's information criterion (AIC). In the experiments in Section III we use the MDL cost to select the number of mixture components at each level. This MDL cost [11], [12], C is given as

$$-\log \Pr(I \,|\, H) + \frac{d}{2} \log(N) \tag{25}$$

where $\log \Pr(I \,|\, H)$ is the likelihood of the training data under the model H, d is the number of parameters in H, and N is the number of images in the training set. We use this cost in a search for an optimal *architecture*, by which we mean the number of mixture components at each level in the model.

The search algorithm we use in the experiments of Section III proceeds as follows. We begin with only one mixture component at each level. We then duplicate each component along with the associated parameters, i.e., $\Pr(a_l \,|\, a_{l+1})$, $\bar{\mathbf{g}}_{a_l}$, M_{a_l}, and Λ_{a_l}, introducing small random perturbations in the parameters of the duplicate component. We retrain the new, larger model and compare its MDL cost with the previous model. We repeat this, successively duplicating labels and retraining, until the MDL cost increases. The model with the lowest MDL cost is then used in the applications presented below.

Note that we do not always duplicate (or split) the labels at all levels. Since each label value has an associated mixture component, there must be sufficient pixels in the training set to fit all

TABLE I
NUMBERS OF HIDDEN LABEL VALUES

	Layer				
Class	0	1	2	3	4
Aircraft Positives	64	**16**	**4**	2	-
Aircraft Negatives	**128**	**16**	**2**	**1**	-
SAR BMP2	32	16	**16**	**8**	-
SAR BTR70	32	16	**16**	**8**	-
SAR T72	64	16	**16**	**8**	-
Mass Positives	32	32	**16**	**4**	**1**
Mass Negatives	64	**64**	16	**4**	**4**

Bold numbers are at upper bound.

of the mixture components at that level. It is possible to use too many mixture components, in which case the average matrices used in (15) and (17) become rank deficient for some components. Accordingly, we specify a maximum numbers of labels for each level and stop duplicating labels at a level when the resulting number would exceed the maximum. Finally, we constrain the mixture components at the coarsest level to have diagonal covariance matrices, once again because of the lack of training data at that level.

III. APPLICATIONS AND EXPERIMENTAL RESULTS

To demonstrate its broad applicability, we train a HIP model for each of three sets of images: EO aerial images of aircraft, SAR aerial images of vehicles, and mammographic images of malignant masses. For each of the data sets we apply the model to classification, synthesis and compression. In all cases, we compare the HIP model with the original HMT model.

A. Preprocessing and Training Methods

We divide each of the three data sets into training and test sets of approximately equal size. We use a set of orthogonal wavelets to decompose the images into features. For the HIP model, we use orthogonal wavelets with subsampling by three (Appendix) since these enabled an explicit center pixel to be defined, which we feel may be an advantage for classification. For the HMT we use eight-tap Daubechie wavelets, since these are commonly used in the HMT literature. When applying the wavelet decompositions we wrap image borders, so that for an image of width W, a pixel with horizontal index i can also be referenced by horizontal index $i + nW$, with n an integer. This effectively treats the images as toroidal, and is needed for compression and synthesis in order to get perfect reconstruction with non-Haar wavelets. We crop images so that they are square, with objects approximately centered. We apply the wavelet decomposition for a maximum number of pyramid levels, resulting in the coarsest-level feature images consisting of a single pixel. As discussed earlier, for the HIP models we include I_{L+1} in the set of features $\mathbf{G}_L$ at the lowest resolution level, which forces the model to account for overall image brightness. We also include the low-pass band (without subsampling) at each level in the set of parent feature images $\mathbf{F}_l$, allowing the features $\mathbf{G}_l$ to depend on local image brightness.

We train the HIP model using the EM algorithm described in Section II-C. The number of labels was chosen through the MDL-based splitting procedure described in Section II-D. The number of hidden label values in the MDL optimal HIP models are given in Table I. Label numbers that reached the maximum value we allowed for a given task and level are shown in bold. The HMT models were trained using software downloaded from http://www-dsp.rice.edu/software/whmt.shtml.

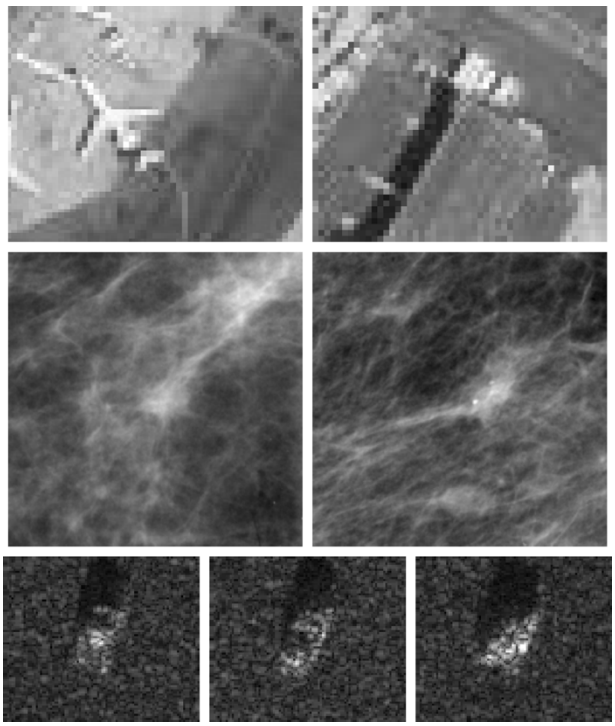

Fig. 4. Examples of data used in experiments. Top row: Two image classes for EO dataset. Left: Aircraft. Right: False positives from template matching, usually buildings. Middle row: Examples of two image classes for X-ray mammography dataset. Left: Malignant mass. Right: False positive generated by the UofC CAD system. Many of the false positives had structure very similar to the malignant masses. Bottom row: Examples of three image classes in SAR dataset. Left: BMP2. Center: BTR70. Right: T72.

For the aircraft and mammography data sets, we performed a jack-knife study of classifier performance. That is, we randomly split the data sets into training and test sets, then trained and tested models on these sets, and repeated this procedure ten times with different random splits of the data. The final test performance figures are averages of the test results over jack-knife splits. The spread of performance gives information on sensitivity to the split of the data set, and reduces the variance in the performance figures, i.e., accidental high or low values. As an important but secondary issue, it also tests variations in local minima of the likelihood during EM training. We did not perform a jack-knife study on the SAR data, since it is supplied as explicit training and test sets.

B. Data Sets

Example images of the different classes in each of the three datasets are shown in Fig. 4. Note that the difference between the classes is often very subtle.

1) Aircraft: This dataset was constructed from large EO overhead imagery of Logan and San Francisco International Airports. We selected regions of interest (ROIs), each containing an aircraft or a false positive (a negative example), by applying a simple template matching algorithm. We collect 40 positive and 40 negative ROI images, each cropped to 81×81 pixels for the HIP model and 64×64 pixels for the HMT.

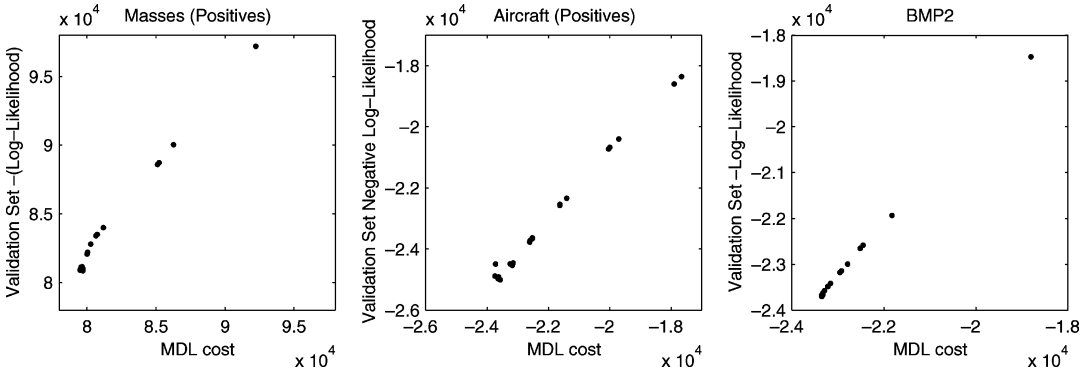

Fig. 5. Negative log-likelihood on validation data versus MDL cost (on training data). Left: Models trained on positive mass images. Middle: Models trained on positive aircraft images. Right: Models trained on BMP2 images.

2) SAR Images of Vehicles: This dataset was taken from the "MSTAR Targets" data set.[11] We chose separate training and test sets for each of three vehicle classes: BMP2, BTR70, and T72. Each vehicle was imaged at fifteen degree intervals in azimuth. The training and test sets were gathered at different depression angles (fifteen and seventeen degrees, respectively) introducing a small systematic variation between the sets. The images were 128×128 pixels, which we cropped to 81×81 pixels for the HIP model. There were 233 images in the BMP2 and BTR70 training sets, 232 images in the T72 training set, 195 images in the BMP2 test set, and 196 images in the BTR70 and T72 test sets.

3) Mammography: This dataset consisted of ROIs selected from mammograms by a computer-aided diagnosis (CAD) system developed at the Rossmann Laboratories of the University of Chicago (UofC) [35], [36]. Of these ROIs, 72 contained malignant masses and 169 were false positives of the CAD system. The detected objects (apparent lesions) are not necessarily centered in the ROI, since they may lie close to the edge of the mammogram. The original ROIs were 512×512 pixels, but to save computer time we sub-sampled them by two in each direction after applying a standard five-tap binomial blurring filter. These smaller ROIs were cropped to 243×243 pixels for the HIP models.

C. MDL Results

Since fitting a HIP model to data means maximizing the likelihood, the appropriate measure of generalization performance is the likelihood of new data. So, to test the effectiveness of the MDL cost for choosing a model architecture, we compared the MDL cost on the training data with the likelihood of a validation set. Scatter plots are shown in Fig. 5. Though the relationship is not always perfectly linear, it is monotonic with very little scatter. Near the optimum there is some disagreement between the two measures, but the difference between the criteria for these models is very small. The MDL cost was an excellent guide to choosing a model that generalizes well.

D. Classification

For two-class problems, we classified an image by the ratio of its likelihoods under models trained on each of the classes. The standard measure of performance in such a case is A_z, the area under the receiver operating characteristic (ROC) curve [37]. For the SAR vehicle dataset we have three classes and, therefore, cannot use A_z as a performance metric. Instead we classify examples according to the model that gives the highest likelihood, measuring overall performance by the percent correctly classified.

TABLE II
SUMMARY OF CLASSIFICATION PERFORMANCE, HIP VERSUS HMT

	HIP	HMT
Aircraft (A_z)	0.889 ± 0.031	0.556 ± 0.047
Mammo Masses (A_z)	0.798 ± 0.027	0.558 ± 0.051
SAR Vehicles (% correct)	84.7%	32.7%

A summary of classification results is shown in Table II. For the aircraft and mammography problems, the results are reported in terms of the area under the ROC curve as measured on the test sets. The errors given are standard deviations across the jack-knife results. For all three problems the HIP models performed significantly better than the HMT models. The poor performance of the HMT, which is little better than chance, is surprising given its good performance in other applications. It may be that the HMT captures important structures in some image classes but lacks the flexibility to capture subtler features that distinguish between similar classes.

E. Image Synthesis

Since HIP models are generative, we can sample them to synthesize new images. These synthesized images can provide qualitative insight into what features the model is extracting and representing for both positive and negative ROIs. The sampling procedure begins at the coarsest resolution, where the hidden labels are randomly sampled from the distribution $\Pr(A_L)$. The feature images $\mathbf{G}_L$ are then sampled from $\Pr(\mathbf{G}_L \,|\, A_L)$. The $\mathbf{G}_L$ are used to construct I_{L-1}, from which the $\mathbf{F}_L$ are constructed. We then sample A_{L-1} from $\Pr(A_{L-1} \,|\, A_L)$, and then $\mathbf{G}_{L-1}$ from $\Pr(\mathbf{G}_{L-1} \,|\, \mathbf{F}_L, A_{L-1})$. This is repeated until the finest resolution is reached and I_0 is constructed. Figs. 6 and 7 show examples of synthetic images for all three datasets generated by the HIP and HMT models (since the HMT models do not model the mean image intensity, we scaled each individual HMT synthetic image so that black and white are the minimum and maximum pixel values within the image; this was also done

[11] Available at http://www.mbvlab.wpafb.af.mil/public/sdms/ datasets/mstar/.

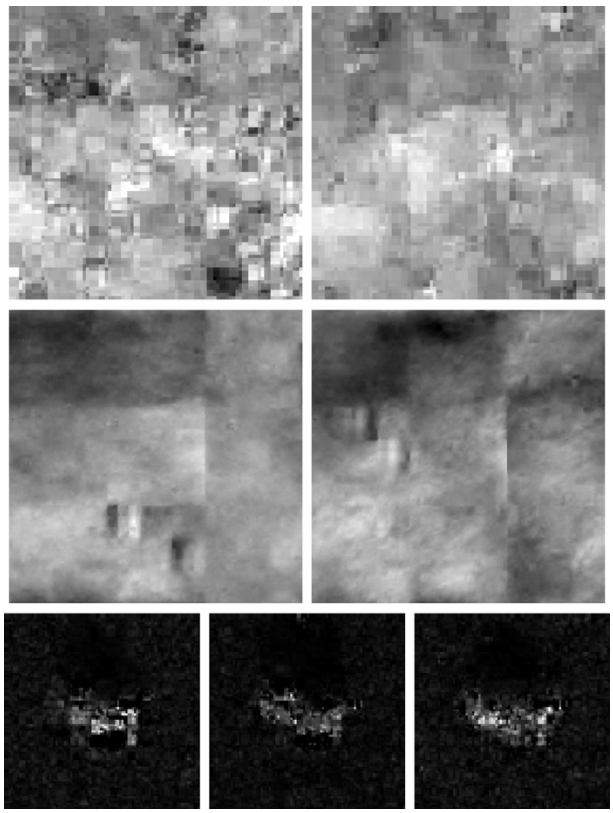

Fig. 6. Synthetic images generated by HIP models. Top row: Models trained on (left) positive and (right) negative examples of aircraft images. Middle row: Mammographic ROI images synthesized from (left) positive and (right) negative models. Bottom row: Synthetic images generated by HIP models trained on examples of the three SAR vehicle types.

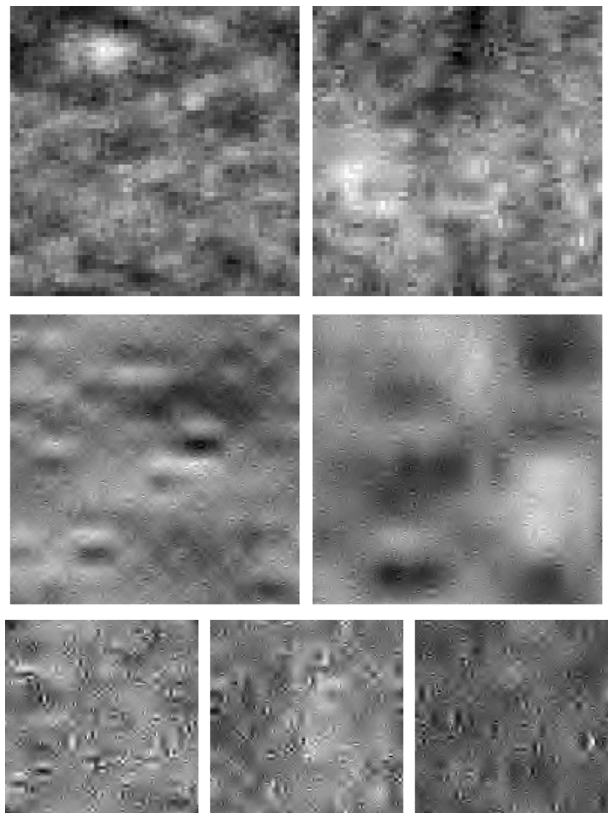

Fig. 7. Synthetic images generated by HMT models. Top row: Models trained on (left) positive and (right) negative examples of aircraft images. Middle row: Mammographic ROI images synthesized from (left) positive and (right) negative models. Bottom row: Synthetic images generated by HMT models trained on examples of the three SAR vehicle types.

for the HIP mass images, but not for the HIP aircraft or SAR images). Though the images synthesized using the HIP model do not capture all the detail indicative of both the image type and class, they clearly are better matches to the data compared to those synthesized using the original HMT. For example, the HIP models synthesize images that capture the statistical dependencies that distinguish the three modalities (EO, X-ray, and SAR), while those synthesized by the original HMT do not. In addition the synthesized images from the HIP model demonstrate subtle differences between positive and negative classes for a given image modality. For example, the two synthesized images for the EO imagery show that the model for negative examples results in a synthesized square blob in the center of the image. This is presumably because many of the negative examples in the dataset consisted of aerial views of building. In addition, the image synthesized for x-ray mammography positive examples shows a central white blob, indicative that this HIP model represents centralized mass structure. Such class-dependent structure is not seen in the images synthesized using the original HMT models.

1) Conditional Distribution of Features With HIP: To further test the ability of the HIP model to learn an accurate representation of an image distribution, we measure several distributions of individual features $g_l(x)$ conditioned on the parent feature $f_{l+1}(x)$ for both real and synthesized images. A typical example (aircraft, level 0, horizontal intermediate band) is shown in Fig. 8. The conditional distributions we examined all have similar appearance, and in all cases the real and synthetic distributions agree.

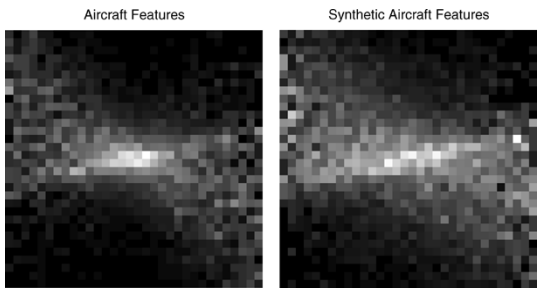

Fig. 8. Distribution of a feature $g_l(x)$ conditioned on its parent feature $f_{l+1}(x)$ for (left) real and (right) synthetic images.

Buccigrossi and Simoncelli [25] have reported such "bow-tie" shape conditional distributions for a variety of features. We point out that such conditional distributions are naturally obtained for any mixture of Gaussian distributions with varying scales and zero means.[12] The present HIP model learns such conditionals, in effect describing the features as nonstationary Gaussian variables [38].

[12]In our case, the means are not constrained to be zero, but often have very small values after training.

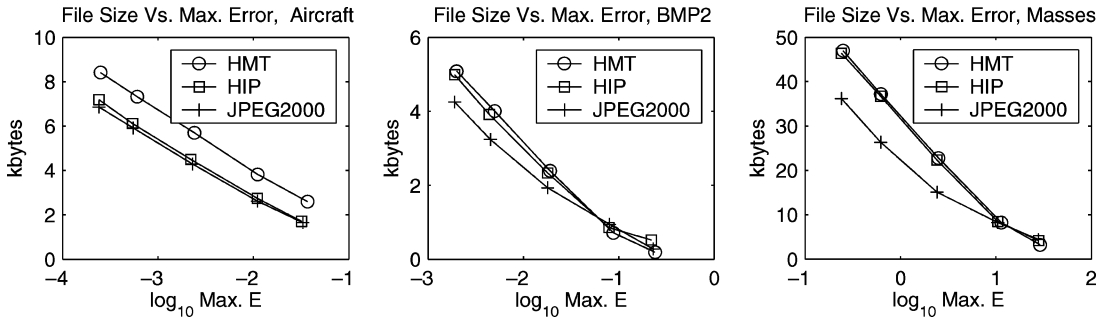

Fig. 9. Size of compressed files versus maximum pixel error for HMT models, MDL optimal HIP models and JPEG 2000. Left: Aircraft images. Center: BMP2 SAR images. Right: Mass images.

F. Compression

A stream of random values can be optimally compressed if we know the distribution of these values. A model of a source of images should, therefore, allow us to compress examples of those images with high efficiency. Here we demonstrate compression with HIP and HMT models using a simple technique.

Given an image and a HIP model, we compress the image as follows. First, we compute the most likely value of each hidden label, $a_l{}^*(x) = \arg\max_{a_l} \Pr(a_l \,|\, I, x, \theta^t)$ using (24). These most likely values are then encoded with arithmetic coders [39], [40], which require a probability distribution for the symbols they are to encode. For this, we use the HIP model distributions $\Pr(a_l{}^*(x) \,|\, a_{l+1}{}^*(x))$.

Given the label value $a_l{}^*(x)$, we then encode the feature vector $\mathbf{g}_l(x)$ using $\Pr(\mathbf{g}_l|\mathbf{f}_{l+1}, a_l{}^*, x)$. The latter is used by decomposing $\mathbf{g}_l(x) - (\bar{\mathbf{g}}_{a_l{}^*} + M_{a_l{}^*}\mathbf{f}_{l+1}(x))$ into its components along the eigenvectors of the covariance matrix $\Lambda_{a_l{}^*}$. These components are independent under $\Pr(\mathbf{g}_l|\mathbf{f}_{l+1}, a_l{}^*)$ so they can be encoded independently. Each component is encoded with a specified precision by dividing the real line into intervals of width equal to twice the precision. We then encode the index of the interval containing the component using an arithmetic coder. The probability of each interval is provided by the integral of the univariate Gaussian distribution of the component implied by $\Pr(\mathbf{g}_l|\mathbf{f}_{l+1}, a_l{}^*)$. This procedure is computationally expensive, and is not necessarily optimal even if the HIP model exactly matches the image distribution, but it serves to demonstrate the capability. We used the analogous procedure for compression with the HMT models, except the DC residual is stored separately, without compression (note that the precision could change with level in the pyramid, depending on how the wavelets are normalized).

We compress the images at several values of the precision. For each of the data sets, which have single-precision floating point pixels, the range of precisions was chosen by finding the maximum pixel value in the set of images and choosing precisions between approximately 10^{-4} and 0.015 times this maximum. Note that this gives rather different ranges for the three data sets. The images in the aircraft set all have maxima near one and the maximum over all images is one; the maximum over all of the BMP2 images is around eight while the maxima of individual BMP2 images vary considerably from this; and the mass images have individual maxima that vary somewhat while the overall maximum is a little more than one thousand.

Note that we are not including the parameters of either the HIP or HMT models in the code lengths of the images. This is because there is one set of parameters used for all of the images. In the limit of large numbers of images, this extra code length is negligible. One can imagine new images of a class, say mammographic masses, compressed and decompressed with a specialized program that includes the model parameters.

To compare with JPEG 2000, we first convert the images to integer pixels. We divided the images by $2\epsilon_{\mathrm{HIP}}$, where ϵ_{HIP} is the maximum error on the entire data set caused by compression and decompression with the HIP model at a given precision. The rounding introduces a maximum error of 0.5 in the scaled pixels, so that uncompressing the image and multiplying by $2\epsilon_{\mathrm{HIP}}$ gives back the maximum error ϵ_{HIP}. We then compressed the scaled integer images losslessly with JPEG 2000.

The results are shown in Fig. 9 for both HMT, HIP, and for JPEG 2000. Although our somewhat naïve method of compressing images with HMT and HIP models is not quite as good as JPEG 2000, it does perform reasonably well. The results show that the hidden labels do capture useful information, allowing better compression.

G. Segmentation

Although we do not have segmentation data for any of the three datasets, we can determine the label probabilities at each location in an image (at several scales) to obtain a qualitative estimate of how well the HIP model segments a given image. Fig. 10 shows probability images for two of the labels from level one of the MDL optimal HIP model for the BMP2 image shown in Fig. 4. The left image shows a label that appears to represent the radar "shadow" behind the vehicle, while the label in the right image appears to represent some of the edges of the vehicle (note that it has low probability in the interior and at other boundaries). Of the other labels that we do not show, many appear to represent various details of the background clutter or speckle, and others represent internal details of the vehicle. Taken with the synthesis results for the SAR images, this result shows that the hidden labels represent different textures and the presence of these textures in definite spatial regions. Interestingly, all three vehicle classes share these properties. Since the HIP model can distinguish between these classes with some success, it must learn some of the more subtle distinctions between the spatial regions, or possibly the structure internal to these regions.

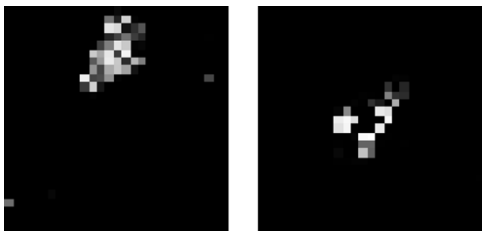

Fig. 10. Two label probability images for the BMP2 image of Fig. 4. Brightness indicates posterior probability of the particular label value at each pixel, given the image and the HIP model.

IV. Conclusion

We have presented a TSBN for natural image distributions that we call the HIP model. It is similar to the HMT of Crouse *et al.* [7]. We demonstrated the HIP model's effectiveness in several applications, comparing with the HMT.

Although there are several differences between HIP models and HMTs, we have emphasized the importance of fitting the HIP model's complexity to the data, varying the complexity by choosing the number of values that the model's hidden variables can take on. We showed that the MDL criterion provides a simple and effective means for making these choices.

In our experiments, the MDL-optimal HIP models have thousands of parameters, yet apparently do not overfit the data. This is due to the nature of the HIP model and many other TSBNs, which reuse their components (in this case, Gaussian distributions of wavelet coefficients) many times within a single image. With models of this sort a single image effectively presents many examples.

We may well ask what these hidden variables learn to represent. The hidden variables in the HMT are designed to represent the range of magnitudes of the wavelet coefficients, in order to give high-kurtosis marginal distributions. They also capture the persistence of magnitude across scale. By contrast, the hidden states in the HIP model are not predefined to have any particular meaning. Inevitably they model coefficient magnitudes, but they can also learn to represent other aspects of image structure, such as spatial groupings of different textures. They do not necessarily result in a physically meaningful segmentation, as they are driven by image statistics. However, this representation of image statistics does sometimes correspond to real properties of the scene, as demonstrated in the previous section. We are not arguing that one should ignore prior knowledge of relevant image structure. In fact, elsewhere, [41] we have modified the HIP model to explicitly represent the range of coefficient magnitudes, as in the HMT or the random cascades of Wainwright *et al.* [3], [15]. However, there is always other image structure that one cannot foresee, so it is worthwhile to include sufficient flexibility in an image model to learn this extra structure.

Another point we emphasized is the flexibility of generative image models. After training such a model once on a set of images we can use it for a variety of applications, as we demonstrated in the experiments. To enable this the structure of the model itself must be such that it is easy to apply forward or backward, i.e., $\Pr(I \mid C)$ or $\Pr(C \mid I)$. Though it is clear that other methods might obtain similar, or even superior, performance on the individual applications presented, we believe that flexible tree-structured models like the HIP model provide a practical unified approach for the modeling and analysis of natural images.

TABLE III
Tap Weights for Eleven-Tap Orthogonal Wavelet Filters With Subsampling by Three

Tap index	Even low-pass	Odd band-pass
0	0.402369	0
1	0.286596	0.405309
2	0.048816	0
3	-0.034518	-0.048816
4	0	0
5	-0.002079	-0.002940

Appendix

The wavelets we used for the HIP model use subsampling by three. The two-dimensional filters are, as usual, separable, and are products of 1-D wavelets. For the 1-D filters we solved for appropriate tap weights subject to the following constraints.

1) One filter is even-symmetric and low-pass (taps sum to one, zero response at the Nyquist frequency).
2) A second filter is odd-symmetric (therefore, high or bandpass).
3) The third filter is even-symmetric and high-pass (taps sum to zero).
4) The resulting transform is orthogonal.

For the even-symmetric filters these constraints automatically give zero first derivatives of the frequency response at zero frequency and the Nyquist frequency. To solve these constraints numerically seemed to require eleven taps. In addition, the high-pass filter had taps equal to those of the low-pass filter, except for an alternating sign. Using this to reduce the number of parameters that characterize the three filters, we found that the constraints can be solved exactly by symbolic manipulation software, though the results are somewhat complex. Numerical values for the tap weights of the low and bandpass filters are given in Table III. The central tap has index 0.

Acknowledgment

The authors would like to thank A. Gerson for assistance in running the HMT experiments.

References

[1] D. J. Field, "Relations between the statistics of natural images and the response properties of cortical cells," *J. Opt. Soc. Amer.*, vol. 4, no. 12, pp. 2379–2394, 1987.
[2] D. L. Ruderman and W. Bialek, "Statistics of natural images:Scaling in the woods," *Phys. Rev. Lett.*, vol. 73, no. 6, pp. 814–817, Aug. 1994.
[3] M. J. Wainwright and E. P. Simoncelli, "Scale mixtures of Gaussians and the statistics of natural images," in Advances in Neural Information Processing Systems, S. A. Solla, T. K. Leen, and K.-R. Müller, Eds. Cambridge, MA: MIT Press, 1999, vol. 12, to be published.
[4] F. Attneave, "Some informational aspects of visual perception," *Psychol. Rev.*, vol. 61, pp. 183–193, 1954.
[5] H. B. Barlow, "Possible principles underlying the transformation of sensory messages," in *Sensory Communication*, W. A. Rosenblith, Ed. Cambridge, MA: MIT Press, 1961, pp. 217–234.
[6] X. Feng, C. K. I. Williams, and S. N. Felderhof, "Combining belief networks and neural networks for scene segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 467–483, Apr. 2002.
[7] 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. 3, pp. 886–902, Mar. 1998.
[8] J. K. Romberg, H. Choi, and R. G. Baraniuk, "Bayesian tree-stuctured image modeling using wavelet domain hidden Markov models," *IEEE Trans. Image Process.*, vol. 10, no. 8, pp. 1056–1068, Aug. 2001.
[9] H. Choi and R. G. Baraniuk, "Multiscale image segmentation using wavelet-domain hidden Markov models," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1309–1321, Oct. 2001.

[10] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–285, Feb. 1989.
[11] J. A. Rissanen, "Information theory and neural nets," in *Math. Perspectives Neural Netw.*, Smolensky, Mozer, and Rumelhart, Eds., 1996, pp. 567–602.
[12] G. Deco and D. Obradovic, *An Information-Theoretic Approach to Neural Computing*. New York: Springer Verlag, 1996.
[13] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, "Shiftable multiscale transforms," *IEEE Trans. Inf. Theory*, vol. 38, no. 2, pp. 587–607, Mar. 1992.
[14] J. Magarey and N. Kingsbury, "Motion estimation using a complex-valued wavelet transform," *IEEE Trans. Signal Processing*, vol. 46, no. 4, pp. 1069–1084, Apr. 1998.
[15] M. J. Wainwright, E. P. Simoncelli, and A. S. Willsky, "Random cascades on wavelet trees and their use in analyzing and modeling natural images," *Appl. Comput. Harmon. Anal.*, vol. 11, pp. 89–123, 2001.
[16] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 6, pp. 194–207, Nov. 1984.
[17] R. Chellappa and S. Chatterjee, "Classification of textures using Gaussian Markov random fields," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-33, no. 8, pp. 959–963, Aug. 1985.
[18] S. C. Zhu, Y. N. Wu, and D. Mumford, "Minimax entropy principle and its application to texture modeling," *Neural Comput.*, vol. 9, no. 8, pp. 1627–1660, 1997.
[19] J. Portilla and E. P. Simoncelli, "A parametric texture model based on joint statistics of complex wavelet coefficients," *Int. J. Comput. Vis.*, vol. 40, no. 1, pp. 49–71, Dec. 2000.
[20] J. S. De Bonet and P. Viola, "Texture recognition using a nonparametric multi-scale statistical model," presented at the Conf. Computer Vision and Pattern Recognition, 1998.
[21] J. S. De Bonet, P. Viola, and J. W. Fisher III, "Flexible histograms: A multiresolution target discrimination model," presented at the SPIE Conf., vol. 3370, E. G. Zelnio, Ed., 1998.
[22] K. C. Chou, A. S. Willsky, A. Benveniste, and M. Basseville, "Recursive and iterative estimation algorithms for multi-resolution stochastic processes," in *Proc. 28th Conf. Decision and Control*, 1989, pp. 1184–1189.
[23] K. C. Chou, A. S. Willsky, and A. Benveniste, "Multiscale recursive estimation, data fusion and regularization," *IEEE Trans. Automat. Control*, vol. 39, no. 3, pp. 464–478, Mar. 1994.
[24] M. R. Luettgen and A. S. Willsky, "Likelihood calculation for a class of multiscale stochastic models, with application to texture discrimination," *IEEE Trans. Image Process.*, vol. 4, no. 2, pp. 194–207, Feb. 1995.
[25] R. W. Buccigrossi and E. P. Simoncelli, "Image Compression via Joint Statistical Characterization in the Wavelet Domain," GRASP Lab., Univ. Pennsylvania, Philadelphia, Tech. Rep. 414, 1998.
[26] C. A. Bouman and M. Shapiro, "A multiscale random field model for Bayesian image segmentation," *IEEE Trans. Image Process.*, vol. 3, no. 3, pp. 162–177, Mar. 1994.
[27] 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.
[28] H. Cheng and C. A. Bouman, "Multiscale Bayesian segmentation using a trainable context model," *IEEE Trans. Image Process.*, vol. 10, no. 4, pp. 511–525, Apr. 2001.
[29] G. Fan and X.-G. Xia, "Wavelet-based texture analysis and synthesis using hidden Markov models," *IEEE Trans. Circuits Syst. I*, vol. 50, no. 1, pp. 106–120, Jan. 2003.
[30] N. M. Dempster, A. P. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc. B*, vol. 39, pp. 185–197, 1977.
[31] E. D. Kolaczyk and R. D. Nowak, "Multiscale likelihood analysis and complexity penalized estimation," Ann. Stat., Feb. 2003, submitted for publication.
[32] C. D. Spence and L. C. Parra, "Hierarchical Image Probability (HIP) Models," Tech. Rep., Sarnoff Corp., Princeton, NJ, 2000.
[33] L. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occuring in the statistical analysis of probabilistic functions on Markov chains," *Ann. Math. Stat.*, vol. 41, pp. 164–171, 1970.
[34] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann, 1988.
[35] R. M. Nishikawa, R. C. Haldemann, J. Papaioannou, M. L. Giger, P. Lu, R. A. Schmidt, D. E. Wolverton, U. Bick, and K. Doi, "Initial experience with a prototype clinical intelligent mammography workstation for computer-aided diagnosis," in *Proc. Medical Imaging*, vol. 2434, M. H. Loew and K. M. Hanson, Eds., Bellingham, WA, 1995, pp. 65–71.
[36] R. M. Nishikawa, R. A. Schmidt, R. B. Osnis, M. L. Giger, K. Doi, and D. E. Wolverton, "Two-year evaluation of a prototype clinical mammographic workstation for computer-aided diagnosis," *Radiology*, vol. 201, no. (P), p. 256, 1996.
[37] D. M. Green and J. A. Swets, *Signal Detection Theory and Psychophysics*. New York: Wiley, 1966.
[38] L. C. Parra, C. D. Spence, and P. Sajda, "Higher order statistical properties arising from the nonstationarity of natural signals," in *Advances in Neural Information Processing Systems*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds. Cambridge, MA: MIT Press, 2001, vol. 13, pp. 786–792.
[39] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, no. 6, pp. 520–540, Jun. 1987.
[40] P. G. Howard and J. S. Vitter, "Arithmetic coding for data compression," *Proc. IEEE*, vol. 82, no. 6, pp. 857–865, Jun. 1994.
[41] P. Sajda, C. Spence, and L. C. Parra, "A multi-scale probabilistic network model for detection, synthesis, and compression in mammographic image analysis," *Med. Imag. Anal.*, vol. 7, pp. 187–204, 2003.

Clay Spence received the Ph.D. degree in physics from the University of California, Irvine, in 1987.

He is a member of the technical staff of the Sarnoff Corporation, Princeton, NJ. His expertise includes machine learning and pattern recognition, blind source separation and other array processing of a variety of signal types, simulations of the auditory nervous system of the barn owl, and Monte Carlo simulations of physical systems. Besides image modeling, his current research includes pattern recognition in three-dimensional data sets.

Lucas C. Parra received the Ph.D. degree in physics from the Ludwig-Maximilian University, Germany, in 1996.

He is an Associate Professor of biomedical engineering at the City College of New York, New York. Previously, he was Technology Leader for Adaptive Image and Signal Processing at Sarnoff Corporation, Princeton, NJ (from 1997 to 2003), and a Member of the Technical Staff at the Machine Learning and the Imaging Departments, Siemens Corporate Research (from 1995 to 1997). From 2002 to 2003, he was also Adjunct Assistant Professor of Biomedical Engineering, Columbia University, New York. His expertise includes machine learning and pattern recognition, acoustic array processing, emission tomography, and encephalography. His current research in biomedical signal processing and medical imaging focuses on functional brain imaging.

Paul Sajda received the B.S. degree in electrical engineering from the Massachusetts Institute of Technology (MIT), Cambridge, in 1989, and the M.S. and Ph.D. degrees in bioengineering from the University of Pennsylvania (UPenn), Philadelphia, in 1992 and 1994, respectively.

In 1994, he joined Sarnoff Corporation, Princeton, NJ, where he went on to become the Head of the Adaptive Image and Signal Processing Group. He is currently an Associate Professor of biomedical engineering and radiology at Columbia University, New York, where he is the Director of the Laboratory for Intelligent Imaging and Neural Computing. Currently, he has over 80 publications and holds five U.S. patents. His research focuses on neural engineering, neuroimaging, computational neural modeling, and machine learning applied to image understanding.

Dr. Sajda has received several awards for his research, including an NSF CAREER Award (2002), the Sarnoff Technical Achievement Award (1996), The Pollack Award for outstanding dissertation research in Bioengineering (UPenn, 1994), The Flexner Award for outstanding thesis research in the Neurosciences (UPenn, 1993), and the Adler Award for outstanding undergraduate research in Electrical Engineering (MIT, 1989). He is an Associate Editor for IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING.