Dynamic Trees for Unsupervised Segmentation and Matching of Image Regions

Sinisa Todorovic, *Student Member*, *IEEE*, and Michael C. Nechyba, *Member*, *IEEE*

Abstract—We present a probabilistic framework—namely, multiscale generative models known as Dynamic Trees (DT)—for unsupervised image segmentation and subsequent matching of segmented regions in a given set of images. Beyond these novel applications of DTs, we propose important additions for this modeling paradigm. First, we introduce a novel DT architecture, where multilayered observable data are incorporated at all scales of the model. Second, we derive a novel probabilistic inference algorithm for DTs—Structured Variational Approximation (SVA)—which explicitly accounts for the statistical dependence of node positions and model structure in the approximate posterior distribution, thereby relaxing poorly justified independence assumptions in previous work. Finally, we propose a similarity measure for matching dynamic-tree models, representing segmented image regions, across images. Our results for several data sets show that DTs are capable of capturing important component-subcomponent relationships among objects and their parts, and that DTs perform well in segmenting images into plausible pixel clusters. We demonstrate the significantly improved properties of the SVA algorithm—both in terms of substantially faster convergence rates and larger approximate posteriors for the inferred models—when compared with competing inference algorithms. Furthermore, results on unsupervised object recognition demonstrate the viability of the proposed similarity measure for matching dynamic-structure statistical models.

Index Terms—Generative models, Bayesian networks, dynamic trees, variational inference, image segmentation, image matching, object recognition.

1 Introduction

We present a probabilistic framework for image segmentation and subsequent matching of segmented regions in a given set of images, when only weak or no prior knowledge is available. Thus, we formulate the image-segmentation and matching problems as inference of posterior distributions over pixel random fields given the observed image. Our principal challenge, therefore, lies in choosing a suitable and numerically manageable statistical model, which provides the means for 1) clustering pixels into image regions, which we interpret as objects and 2) detection of instantiated objects across a given set of images. The solution to these two problems can be viewed as critical core components of an integrated computer vision system that is capable of first registering unknown/known objects over an image set and then updating its knowledge base accordingly. While considerations of such a system are beyond the scope of this paper, we point out that the core components introduced here form the basis of many prospective vision systems. Our focus herein is the formulation of a statistical modeling paradigm with the specified capabilities.

Given the assumption that dependencies among image regions occur only through component-subcomponent relationships, multiscale generative models known as *dynamic trees* (DTs) appear very suitable for our goals [1], [2], [3], [4]. In DTs, nodes represent random variables, and arcs between them model causal (Markovian) dependence assumptions through scales, as illustrated in Fig. 1. DTs provide a distribution over image-class labels associated with each node, as well as a distribution over network connectivity. Therefore, for a given image, posteriors of both network structure and image-class labels need to be inferred in the inference algorithm. After inference, the model structure can be determined through Bayesian (MAP) estimation, which gives a topology solution comprising a forest of subtrees, each of which segments the image, as depicted in Fig. 1. Since each root determines a subtree, whose leaf nodes form a detected object, we can assign physical meaning to roots as representing whole objects. Also, each descendant of the root down the subtree can be interpreted as the root of another subtree whose leaf nodes cover only a part of the object. Thus, roots' descendants can be viewed as object parts at various scales. The experimental results over several types of image data sets, which we report herein, show that DTs are capable of capturing important component-subcomponent relationships among objects and their parts, and that DTs perform well in segmenting images into plausible pixel clusters. Hence, the generative (Markovian) property of DTs provides a solution to our first problem of unsupervised image segmentation.

With respect to our second problem, we stress that traditional approaches to object recognition based on statistical modeling of object appearances and subsequent probability-based classification as, for example, in [5], [6], [7], are ill-posed for unsupervised settings. Here, *ill-posed* indicates two difficulties. First, the problem is not uniquely solvable since the absence of prior knowledge on possible image classes renders the design of a classifier ambiguous and, second, the solution does not depend on the data in a continuous way; that is, insufficiently large training data sets lead to very unreliable statistical models. Consequently, in

- *S. Todorovic is with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611. E-mail: sinisha@ufl.edu.*
- *M.C. Nechyba is with Pittsburgh Pattern Recognition, Inc., 40 24th Street, Suite 240, Pittsburgh, PA 15222. E-mail: michael@pittpatt.com.*

Manuscript received 13 Oct. 2003; revised 17 Mar. 2005; accepted 21 Mar. 2005; published online 14 Sept. 2005.
Recommended for acceptance by J. Buhmann.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0316-1003.

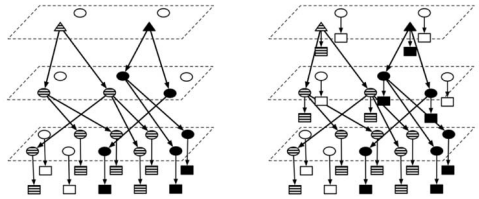

Fig. 1. Two types of DTs: (a) Observable variables present at the leaf level only. (b) Observable variables present at all levels, round and square-shaped nodes indicate hidden and observable random variables, triangles indicate roots, and unconnected nodes in this example belong to other subtrees; each subtree segments the image into regions marked by distinct shading.

unsupervised settings, it is not possible to reliably estimate the variability of data within a class and across various classes, and thereby to set thresholds for subsequent classification. In contrast, it is widely reported in the literature that image matching based on a suitably defined similarity measure is resilient to the outlined problems [8], [9], [10], [11], [12], [13]. As such, herewith, we formulate the object-recognition problem in unsupervised settings as one of computing a *similarity measure* between DTs representing instantiated objects across images. For this purpose, we define a novel similarity measure between two dynamic-structure models. From the given data set, we first choose an arbitrary image to be the reference, and then search for segmented image regions in the rest of the images that are similar to the ones in the reference image. Note that the unsupervised setting precludes the term *reference* signifying *known* image classes (i.e., objects). Therefore, we are not in position to successfully recognize every appearance of a reference object across the data set. Since the appearances of the examined object may differ by many factors including pose, occlusion, lighting, and scale, by matching we can only search for the particular reference appearance of the object in the rest of the images. This limitation of the outlined approach is, however, inherent to unsupervised settings in general, not to our approach in particular. Nevertheless, results of image matching in unsupervised settings, which we report herein, demonstrate the viability of the proposed approach over different image sets.

This paper makes a number of contributions:

1. We introduce multilayered data into the model, as illustrated in Fig. 1b—an approach that has been extensively investigated in fixed-structure quad-trees (e.g., [14], [15], [16]). Throughout, we assume that multiscale data forms a quad-tree structure of dyadic squares, as is the case, for example, in the wavelet transform. The proposed models of data quad-trees have proved rather successful for various applications including image denoising, classification, and segmentation. Hence, it is important to develop a similar formulation for DTs. To our best knowledge, the literature reports only research on DTs whose observables exist at the leaf level, as depicted in Fig. 1a, [1], [2], [3], [4]. This may be so because of a fundamental problem with propagating observables to higher levels in a generative model. Since overlapping image parts are being reused for deriving observables at different levels, a generative model may generate pixel values that are inconsistent with any possible image. Nevertheless, this problem can be alleviated by appropriate normalization or by spanning image data over a set of orthonormal functions (e.g., in the wavelet transform).
2. We develop a novel probabilistic inference algorithm for the proposed model. As is the case for many complex-structure models, exact inference for DTs is intractable. Therefore, we assume that there are averaging phenomena in DTs that may render a given set of variables approximately independent of the rest of the network, and thereby derive a Structured Variational Approximation (SVA) [17], [18], [19]. SVA provides for principled solutions, while reducing computational complexity. Unlike the variational approximation discussed in [4], we explicitly account for the dependence of node positions on the model's structure, which results in the algorithm's faster convergence, when compared to existing approaches.
3. We propose a similarity measure for comparing DTs, which we employ for matching segmented image regions across a given set of images. Standard probabilistic approaches to matching use the log-ratio of model distributions representing the examined image regions [8], [9], [10], [11]. However, as explained before, in unsupervised settings, these distributions may have large variations across images, due to the uninformed estimation of the model parameters. To alleviate this problem, we measure correlation between the cross-likelihoods of the two image regions, normalized by the likelihoods of each individual region.

The remainder of the paper is organized as follows: In Section 2, we first discuss prior research related to tree-structured statistical modeling. Next, in Section 3, we define our dynamic-tree framework. In Section 4, we derive our structured-variational approximation inference algorithm for DTs, and discuss various implementation issues inherent to the algorithm, as well as to unsupervised settings. Then, in Section 5, we consider the problem of comparing DT models and define a similarity measure for that purpose. Finally, in Section 6, we present experimental results on segmentation and matching of image regions for several classes of images.

2 Related Work

Recently, there has been a flurry of research in the field of tree-structured belief networks (TSBNs) [5], [14], [15], [16], [20]. TSBNs are characterized by a fixed, balanced tree-structure of nodes, representing random variables. The edges of TSBNs represent parent-child (Markovian) dependencies between neighboring layers of nodes, while random variables belonging to the same layer are conditionally independent. TSBNs have efficient linear-time inference algorithms, of which, in the graphical-models literature, the best-known is Pearl's λ-π message passing scheme, also known as belief propagation [5], [21]. Cheng and Bouman have used TSBNs for multiscale document segmentation [14]; Schneider et al. have used TSBNs to replace the initial Markovian random field prior and have achieved efficient simultaneous image denoising and segmentation [20]. The aforementioned examples demonstrate the powerful expressiveness of TSBNs

and the efficiency of their inference algorithms, which is critically important for our purposes.

Despite these attractive properties, TSBNs give rise to blocky segmentations, due to their fixed structure. In the literature, there are several approaches to alleviate this problem. Irving's research group has proposed an overlapping tree model, where distinct nodes correspond to overlapping parts in the image [22]. In [23], the authors have discussed two-dimensional hierarchical models, where nodes are dependent both at any particular layer through a Markov-mesh and across resolutions. In both approaches, segmentation results are superior to those when standard TSBNs are used because the descriptive component of the models is improved at some increased computational cost. Ultimately, however, these approaches do not deal with the source of the "blockiness"—namely, the fixed tree structure of TSBNs.

Aside from the work of Williams et al. [1], [2], [3], [4], to which we will refer throughout the paper, we point out other research concerning dynamic-tree structures. Konen et al. have proposed a flexible neural mechanism for invariant pattern recognition based on correlated neuronal activity and the self-organization of dynamic links in neural networks [24]. Also, Montanvert et al. [25] have explored irregular multiscale tessellations that adapt to image content.

3 Dynamic Trees

The model formulation discussed herein is similar to that of the position-encoding dynamic trees in [4], where observables are present only at the lowest model level. However, since we also consider multilayered observable data in DTs, for completeness, we introduce both types of models, emphasizing, where appropriate, the differences.

DTs are directed, acyclic graphs with two disjoint sets of nodes representing hidden and observable random vectors. Graphically, we represent all hidden variables as round-shaped nodes, connected via directed edges indicating Markovian dependencies, while observables are denoted as rectangular-shaped nodes, connected only to their corresponding hidden variables, as depicted in Fig. 1. Below, we first introduce nodes characterized by hidden variables.

There are V round-shaped nodes, organized in hierarchical levels, $V^\ell, \ell = \{0, 1, \ldots, L-1\}$, where V^0 denotes the leaf level and $V' \triangleq V/V^0$. The number of round-shaped nodes is identical to that of the corresponding quad-tree with L levels, such that $|V^\ell| = |V^{\ell-1}|/4 = \ldots = |V^0|/4^\ell$. Connections are established under the constraint that a node at level ℓ can become a root, or it can connect only to the nodes at the next $\ell+1$ level. The network connectivity is represented by random matrix Z, where entry z_{ij} is an indicator random variable, such that $z_{ij}=1$ if $i \in V^\ell$ and $j \in \{0, V^{\ell+1}\}$ are connected. Z contains an additional zero ("root") column, where entries $z_{i0}=1$ if i is a root. Since each node can have only one parent, a realization of Z can have at most one entry equal to 1 in each row. We define the distribution over connectivity as

$$P(Z) \triangleq \prod_{\ell=0}^{L-1} \prod_{(i,j) \in V^\ell \times \{0, V^{\ell+1}\}} \left[\gamma_{ij}\right]^{z_{ij}}, \quad (1)$$

where γ_{ij} is the probability of i being the child of j, subject to $\sum_{j \in \{0, V^{\ell+1}\}} \gamma_{ij} = 1$.

Further, each round-shaped node i (see Fig. 1) is characterized by random position $\boldsymbol{r}_i$ in the image plane. The distribution of $\boldsymbol{r}_i$ is conditioned on the position of its parent $\boldsymbol{r}_j$ as

$$P(\boldsymbol{r}_i | \boldsymbol{r}_j, z_{ij}=1) \triangleq \frac{\exp(-\frac{1}{2}(\boldsymbol{r}_i - \boldsymbol{r}_j - \boldsymbol{d}_{ij})^{\mathrm{T}} \Sigma_{ij}^{-1} (\boldsymbol{r}_i - \boldsymbol{r}_j - \boldsymbol{d}_{ij}))}{2\pi |\Sigma_{ij}|^{\frac{1}{2}}}, \quad (2)$$

where Σ_{ij} is a diagonal matrix that represents the order of magnitude of object size and parameter $\boldsymbol{d}_{ij}$ is the mean of relative displacement $(\boldsymbol{r}_i - \boldsymbol{r}_j)$. In [4], the authors, for simplicity, set $\boldsymbol{d}_{ij}$ to zero, which favors undesirable positioning of children and parent nodes at the same locations. From our experiments, this may seriously degrade the image-modeling capabilities of DTs, and as such some nonzero relative displacement $\boldsymbol{d}_{ij}$ needs to be accounted for. The joint probability of $R \triangleq \{\boldsymbol{r}_i | \forall i \in V\}$, is given by

$$P(R|Z) \triangleq \prod_{i,j \in V} \left[\, P(\boldsymbol{r}_i | \boldsymbol{r}_j, z_{ij}) \,\right]^{z_{ij}}, \quad (3)$$

where for roots i we have $P(\boldsymbol{r}_i | z_{i0}=1) \triangleq \exp(-\frac{1}{2}(\boldsymbol{r}_i - \boldsymbol{d}_i)^{\mathrm{T}} \Sigma_i^{-1} (\boldsymbol{r}_i - \boldsymbol{d}_i))/(2\pi|\Sigma_i|^{\frac{1}{2}})$. At the leaf level, V^0, we fix node positions R^0 to the locations of the finest-scale observables, and then use $P(Z, R' | R^0)$ as the prior over positions and connectivity, where $R^0 \triangleq \{\boldsymbol{r}_i | \forall i \in V \setminus V^0\}$ and $R' \triangleq \{\boldsymbol{r}_i | \forall i \in V \setminus V^0\}$.

Next, each node i is characterized by an image-class label x_i and an image-class indicator random variable x_i^k, such that $x_i^k = 1$ if $x_i = k$, where k is a label taking values in the finite set M. Thus, we assume that the set M of unknown image classes is finite. The label k of node i is conditioned on image class l of its parent j and is given by conditional probability tables P_{ij}^{kl}. The joint probability of $X \triangleq \{x_i^k | i \in V, k \in M\}$ is given by

$$P(X|Z) = \prod_{i,j \in V} \prod_{k,l \in M} \left[P_{ij}^{kl}\right]^{x_i^k x_j^l z_{ij}}, \quad (4)$$

where for roots i ($z_{i0}=1$) we use priors $P(x_i^k)$.

Finally, we introduce nodes that are characterized by observable random vectors representing image texture and color cues. Here, we make a distinction between two types of DTs. The model where observables are present only at the leaf-level is referred to as DT_{V^0}; the model where observables are present at all levels is referred to as DT_V. To clarify the difference between the two types of nodes in DTs, we index observables with respect to their locations in the data structure (e.g., wavelet dyadic squares), while hidden variables are indexed with respect to a node-index in the graph. This generalizes correspondence between hidden and observable random variables of the position-encoding dynamic trees in [4]. We define position $\boldsymbol{\rho}(i)$ to be equal to the center of mass of the ith dyadic square at level ℓ in the corresponding quad-tree with L levels:

$$\boldsymbol{\rho}(i) \triangleq [(n+0.5)2^\ell \;\; (m+0.5)2^\ell]^{\mathrm{T}}, \;\; n, m = 1, 2, \ldots, \quad (5)$$

where n and m denote the row and column in the dyadic square at scale ℓ (e.g., for wavelet coefficients). Clearly, other application-dependent definitions of $\boldsymbol{\rho}(i)$ are possible. Note that while the $\boldsymbol{r}$s are random vectors, the $\boldsymbol{\rho}$s are deterministic values fixed at locations where the corresponding observables are recorded in the image. Also, after fixing R^0 to the locations of the finest-scale observables, we

have $\forall i \in V^0$, $\boldsymbol{\rho}(i)=\boldsymbol{r}_i$. The definition, given by (5), holds for DT_{V^0}, as well, for $\ell=0$.

For both types of DTs, we assume that observables $Y \triangleq \{\boldsymbol{y}_{\boldsymbol{\rho}(i)} | \forall i \in V\}$ at locations R^0 and $\boldsymbol{\rho}' \triangleq \{\boldsymbol{\rho}(i) | \forall i \in V'\}$ are conditionally independent given the corresponding x_i^k:

$$P(Y|X, R^0, \boldsymbol{\rho}') = \prod_{i \in V} \prod_{k \in M} \Big[P(\boldsymbol{y}_{\boldsymbol{\rho}(i)} | x_i^k, \boldsymbol{\rho}(i)) \Big]^{x_i^k}, \quad (6)$$

where for DT_{V^0}, V^0 should be substituted for V. The likelihoods $P(\boldsymbol{y}_{\boldsymbol{\rho}(i)} | x_i^k=1, \boldsymbol{\rho}(i))$ are modeled as mixtures of Gaussians: $P(\boldsymbol{y}_{\boldsymbol{\rho}(i)} | x_i^k=1, \boldsymbol{\rho}(i)) \triangleq \sum_{g=1}^{G_k} \pi_k(g) \mathcal{N}(\boldsymbol{y}_{\boldsymbol{\rho}(i)}; \boldsymbol{\nu}_k(g), \Xi_k(g))$. For large G_k, a Gaussian-mixture density can approximate any probability density [26]. In order to avoid the risk of overfitting the model, we assume that the parameters of the Gaussian-mixture are equal for all nodes. The Gaussian-mixture parameters can be grouped in the set $\theta \triangleq \{G_k, \{\pi_k(g), \boldsymbol{\nu}_k(g), \Xi_k(g)\}_{g=1}^{G_k} \,|\, \forall k \in M\}$.

The joint prior of the model can be written as

$$P(Z, X, R, Y) = P(Y|X, R^0, \boldsymbol{\rho}') P(X|Z) P(Z, R'|R^0) P(R^0). \quad (7)$$

All the parameters of the joint prior can be grouped in the set $\Theta \triangleq \{\gamma_{ij}, \boldsymbol{d}_{ij}, \Sigma_{ij}, P_{ij}^{kl}, \theta\}$, $\forall i, j \in V$, $\forall k, l \in M$.

4 Probabilistic Inference

In order to conduct Bayesian estimation of the model for a given image, as required in our formulation of the image segmentation problem, we need to infer the posterior distributions of Z, X, and R', given Y and R^0. However, due to the complexity of DTs, the exact computation of the posterior $P(Z, X, R'|Y, R^0)$ is intractable. Therefore, we resort to an approximate inference method, of which two broad classes exist: deterministic approximations [17], [18], [19] and Monte-Carlo methods [27], [28], [29]. Generally, in MCMC approaches, with increasing model complexity, the choice of proposals in the Markov chain becomes hard, so that the equilibrium distribution is reached very slowly [27], [28]. To achieve faster inference, we consider variational approximation, a specific type of deterministic approximation [17], [18], [19]. Variational approximation methods have been demonstrated to give good and significantly faster results, when compared to Gibbs sampling ([2], chapter 3). The proposed variational approaches range from a factorized approximating distribution over hidden variables [1] (also known as mean field variational approximation) to more structured solutions [4], where dependencies among hidden variables are enforced. The underlying assumption in those methods is that there are averaging phenomena in DTs that may render a given set of variables approximately independent of the rest of the network. Therefore, the resulting variational optimization of DTs provides for principled solutions, while reducing computational complexity. In the following section, we derive a novel Structured Variational Approximation (SVA) algorithm for DTs.

4.1 Structured Variational Approximation

In variational approximation, the intractable distribution $P(Z, X, R'|Y, R^0)$ is approximated by a simpler distribution $Q(Z, X, R'|Y, R^0)$ closest to $P(Z, X, R'|Y, R^0)$. To simplify notation, below, we omit the conditioning on Y and R^0, and write $Q(Z, X, R')$. The novelty of our approach is that we constrain the variational distribution to the form

$$Q(Z, X, R') \triangleq Q(Z) Q(X|Z) Q(R'|Z), \quad (8)$$

which enforces that both class-indicator variables X and position variables R' are statistically dependent on the tree connectivity Z. Since these dependencies are significant in the prior, one should expect them to remain so in the posterior. Therefore, our formulation appears to be more appropriate for approximating the true posterior than the mean-field variational approximation $Q(Z, X, R')=Q(Z)Q(X)Q(R')$ discussed in [1] and the form $Q(Z, X, R')=Q(Z)Q(X|Z)Q(R')$ proposed in [4]. We define the approximating distributions as follows:

$$Q(Z) \triangleq \prod_{\ell=0}^{L-1} \prod_{(i,j) \in V^\ell \times \{0, V^{\ell+1}\}} \big[\xi_{ij}\big]^{z_{ij}}, \quad (9)$$

$$Q(X|Z) \triangleq \prod_{i,j \in V} \prod_{k,l \in M} \Big[Q_{ij}^{kl}\Big]^{x_i^k x_j^l z_{ij}}, \quad (10)$$

$$Q(R'|Z) \triangleq \prod_{i,j \in V'} \big[Q(\boldsymbol{r}_i | z_{ij})\big]^{z_{ij}} \quad (11)$$

$$Q(R'|Z) \triangleq \prod_{i,j \in V'} \frac{\exp\Big(-\frac{1}{2}(\boldsymbol{r}_i - \boldsymbol{\mu}_{ij})^{\mathrm{T}} \Omega_{ij}^{-1} (\boldsymbol{r}_i - \boldsymbol{\mu}_{ij})\Big)}{2\pi |\Omega_{ij}|^{\frac{1}{2}}}, \quad (12)$$

where parameters ξ_{ij} correspond to the γ_{ij} connection probabilities, and the Q_{ij}^{kl} are analogous to the P_{ij}^{kl} conditional probability tables. For the parameters of $Q(R'|Z)$, note that covariances Ω_{ij} and mean values $\boldsymbol{\mu}_{ij}$ form the set of Gaussian parameters for a given node $i \in V^\ell$ over its candidate parents $j \in V^{\ell+1}$. Which pair of parameters $(\boldsymbol{\mu}_{ij}, \Omega_{ij})$, is used to generate $\boldsymbol{r}_i$ is conditioned on the given connection between i and j—that is, the current realization of Z. We assume that the Ωs are diagonal matrices, such that node positions along the "x" and "y" image axes are uncorrelated. Also, for roots, suitable forms of Q functions are used, similar to the specifications given in Section 3. All the parameters of $Q(Z, X, R')$ can be grouped in the set $\Phi \triangleq \{\xi_{ij}, Q_{ij}^{kl}, \boldsymbol{\mu}_{ij}, \Omega_{ij}\}$.

To find $Q(Z, X, R')$ closest to $P(Z, X, R'|Y, R^0)$ we resort to a standard optimization method, where Kullback-Leibler (KL) divergence between $Q(Z, X, R')$ and $P(Z, X, R'|Y, R^0)$ is minimized ([30], chapter 2, pp. 12-49, and chapter 16, pp. 482-509). The KL divergence is given by

$$KL(Q\|P) \triangleq \int_{R'} dR' \sum_{Z,X} Q(Z, X, R') \log \frac{Q(Z, X, R')}{P(Z, X, R'|Y, R^0)}. \quad (13)$$

It is well-known that $KL(Q\|P)$ is nonnegative for any two distributions Q and P, and $KL(Q\|P)=0$ if and only if $Q=P$; these properties are a direct corollary of Jensen's inequality ([30], chapter 2, pp. 12-49). As such, $KL(Q\|P)$ guarantees a global minimum—that is, a unique solution to $Q(Z, X, R')$. In the following section, we show how to compute $Q(Z, X, R')$.

4.2 Update Equations for Computing $Q(Z, X, R')$

By minimizing the KL divergence, we derive the update equations for estimating the parameters of the variational distribution $Q(Z, X, R')$. Below, we summarize the final derivation results. Detailed derivation steps are reported in the Appendix, where we also provide the list of nomenclature. In the following equations, we use κ to denote an arbitrary normalization constant, the definition of which may change from equation to equation. Parameters on the

right-hand side of the update equations are assumed known, as learned in the previous iteration step.

4.2.1 Optimization of $Q(X|Z)$

$Q(X|Z)$ is fully characterized by parameters Q_{ij}^{kl}, which are updated as

$$Q_{ij}^{kl} = \kappa P_{ij}^{kl} \lambda_i^k, \forall i, j \in V,\ \forall k,\ l \in M, \tag{14}$$

where the auxiliary parameters λ_i^k are computed as

$$\lambda_i^k = \begin{cases} P(\boldsymbol{y}_{\boldsymbol{\rho}(i)} | x_i^k, \boldsymbol{\rho}(i)), & i \in V^0, \\ \prod_{c \in V} \left[\sum_{a \in M} P_{ci}^{ak} \lambda_{ci}^{ak}\right]^{\xi_{ci}}, & i \in V', \end{cases} \tag{15a}$$

$$\lambda_i^k = P(\boldsymbol{y}_{\boldsymbol{\rho}(i)} | x_i^k, \boldsymbol{\rho}(i)) \prod_{c \in V} \left[\sum_{a \in M} P_{ci}^{ak} \lambda_c^a\right]^{\xi_{ci}}, \forall i \in V, \tag{15b}$$

where (15a) is derived for DT_{V^0} and (15b) for DT_V. Since the ξ_{ci} are nonzero only for child-parent pairs, from (15), we note that λs are computed for both models by propagating the λ messages of the corresponding children nodes upward. Thus, Qs, given by (14), can be updated by making a single pass up the tree. Also, note that for leaf nodes, $i \in V^0$, the ξ_{ci} parameters are equal to 0 by definition, yielding $\lambda_i^k = P(\boldsymbol{y}_{\boldsymbol{\rho}(i)} | x_i^k, \boldsymbol{\rho}(i))$ in (15b).

Further, from (9) and (10), we derive the update equation for the approximate posterior probability m_i^k that class $k \in M$ is assigned to node $i \in V$, given Y and R^0, as

$$m_i^k = \int_{R'} dR' \sum_{Z,X} x_i^k Q(Z, X, R'), = \sum_{j \in V'} \xi_{ij} \sum_{l \in M} Q_{ij}^{kl} m_j^l. \tag{16}$$

Note that the m_i^k can be computed by propagating image-class probabilities in a single pass downward. This upward-downward propagation, specified by (15) and (16), is very reminiscent of belief propagation for TSBNs [5], [21]. For the special case when $\xi_{ij}=1$ only for one parent j, we obtain the standard λ-π rules of Pearl's message passing scheme for TSBNs.

4.2.2 Optimization of $Q(R'|Z)$

$Q(R'|Z)$ is fully characterized by parameters $\boldsymbol{\mu}_{ij}$ and Ω_{ij}. The update equations for $\boldsymbol{\mu}_{ij}$ and Ω_{ij}, $\forall (i,j) \in V^\ell \times \{0, V^{\ell+1}\}$, $\ell > 0$, where $\xi_{ij} \neq 0$, are

$$\begin{aligned} \boldsymbol{\mu}_{ij} = & \left[\sum_{p \in V'} \xi_{jp} \Sigma_{ij}^{-1} + \sum_{c \in V'} \xi_{ci} \Sigma_{ci}^{-1}\right]^{-1} \\ & \cdot \left[\sum_{p \in V'} \xi_{jp} \Sigma_{ij}^{-1} (\boldsymbol{\mu}_{jp} + \boldsymbol{d}_{jp}) + \sum_{c \in V'} \xi_{ci} \Sigma_{ci}^{-1} (\boldsymbol{\mu}_{ci} - \boldsymbol{d}_{ij})\right], \end{aligned} \tag{17}$$

$$\begin{aligned} \mathrm{Tr}\{\Omega_{ij}^{-1}\} = & \mathrm{Tr}\{\Sigma_{ij}^{-1}\} \left(1 + \sum_{p \in V'} \xi_{jp} \left[\frac{\mathrm{Tr}\{\Sigma_{ij}^{-1} \Omega_{jp}\}}{\mathrm{Tr}\{\Sigma_{ij}^{-1} \Omega_{ij}\}}\right]^{\frac{1}{2}}\right) \\ & + \sum_{c \in V'} \xi_{ci} \mathrm{Tr}\{\Sigma_{ci}^{-1}\} \left(1 + \left[\frac{\mathrm{Tr}\{\Sigma_{ci}^{-1} \Omega_{ci}\}}{\mathrm{Tr}\{\Sigma_{ci}^{-1} \Omega_{ij}\}}\right]^{\frac{1}{2}}\right), \end{aligned} \tag{18}$$

where c and p denote children and grandparents of node i, respectively. Since the Ωs and Σs are assumed diagonal, it is straightforward to derive the expressions for the diagonal elements of the Ωs from (18). Note that both $\boldsymbol{\mu}_{ij}$ and Ω_{ij} are updated summing over children and grandparents of i and, therefore, should be iterated until convergence.

4.2.3 Optimization of $Q(Z)$

$Q(Z)$ is fully characterized by connectivity probabilities ξ_{ij}, which are computed as

$$\xi_{ij} = \kappa \gamma_{ij}\ \exp(A_{ij} - B_{ij}), \forall \ell, \forall (i,j) \in V^\ell \times \{0, v^{\ell+1}\}, \tag{19}$$

where A_{ij} represents the influence of observables Y, while B_{ij} represents the contribution of the geometric properties of the network to the connectivity distribution. These are defined in the Appendix.

4.3 Inference Algorithm and Bayesian Estimation

For the given set of parameters Θ characterizing the joint prior, observables Y, and leaf-level node positions R^0, the standard Bayesian estimation of optimal $\hat{Z}$, $\hat{X}$, and $\hat{R}'$ requires minimizing the expectation of a cost function $\mathcal{C}$:

$$\begin{aligned} & (\hat{Z}, \hat{X}, \hat{R}') = \\ & \arg \min_{Z,X,R'} \mathbb{E}\{\mathcal{C}((Z, X, R'), (Z^*, X^*, R'^*)) | Y, R^0, \Theta\}, \end{aligned} \tag{20}$$

where $\mathcal{C}(\cdot)$ penalizes the discrepancy between the estimated configuration (Z, X, R') and the true one (Z^*, X^*, R'^*). We propose the following cost function:

$$\begin{aligned} & \mathcal{C}((Z, X, R'), (Z^*, X^*, R'^*)) \triangleq \\ & \sum_{i,j \in V} [1 - \delta(z_{ij} - z_{ij}^*)] + \sum_{i \in V} \sum_{k \in M} [1 - \delta(x_i^k - x_i^{k*})] \\ & + \sum_{i \in V'} [1 - \delta(\boldsymbol{r}_i - \boldsymbol{r}_i^*)], \end{aligned} \tag{21}$$

where * indicates true values, and $\delta(\cdot)$ is the Kronecker delta function. Using the variational approximation $P(Z, X, R'|Y, R^0) \approx Q(Z)Q(X|Z)Q(R'|Z)$, from (20) and (21), we derive:

$$\hat{Z} = \arg \min_Z \sum_Z Q(Z) \sum_{i,j} [1 - \delta(z_{ij} - z_{ij}^*)], \tag{22}$$

$$\hat{X} = \arg \min_X \sum_{Z,X} Q(Z) Q(X|Z) \sum_{i,k} [1 - \delta(x_i^k - x_i^{k*}))], \tag{23}$$

$$\hat{R}' = \arg \min_{R'} \int_{R'} dR' \sum_Z Q(Z) Q(R'|Z) \sum_i [1 - \delta(\boldsymbol{r}_i - \boldsymbol{r}_i^*)]. \tag{24}$$

Given the constraints on connections, discussed in Section 3, minimization in (22) is equivalent to finding parents:

$$(\forall \ell)(\forall i \in V^\ell)(Z_{\cdot i} \neq 0) \hat{\boldsymbol{j}} = \arg \max_{j \in \{0, V^{\ell+1}\}} \xi_{ij},\ \text{for } DT_{V^0}, \tag{25a}$$

$$(\forall \ell)(\forall i \in V^\ell) \hat{\boldsymbol{j}} = \arg \max_{j \in \{0, V^{\ell+1}\}} \xi_{ij},\ \text{for } DT_V, \tag{25b}$$

where ξ_{ij} is given by (19); $Z_{\cdot i}$ denotes the ith column of Z and $Z_{\cdot i} \neq 0$ indicates that there is at least one nonzero element in column $Z_{\cdot i}$; that is, i has children, and thereby is included in the tree structure. Note that due to the distribution over connections, after estimation of Z, for a given image, some nodes may remain without children. To preserve the generative property in DT_{V^0}, we impose an additional constraint on Z that nodes above the leaf level must have children in order to be able to connect to upper levels. On the other hand, in DT_V, due to multilayered observables, all

nodes V must be included in the tree structure, even if they do not have children. The global solution to (25a) is an open problem in many research areas. Therefore, for DT_{V^0}, we propose a stage-wise optimization, where, as we move upward, starting from the leaf level $\ell=\{0,1,\ldots,L-1\}$, we include in the tree structure optimal parents at $V^{\ell+1}$ according to

$$(\forall i\in V^{\ell})(\hat{Z}_{\cdot i}\neq 0)\ \ \hat{j}=\arg\max_{j\in\{0,V^{\ell+1}\}}\xi_{ij}, \tag{26}$$

where $\hat{Z}_{\cdot i}$ denotes ith column of the estimated $\hat{Z}$ and $\hat{Z}_{\cdot i}\neq 0$ indicates that i has already been included in the tree structure when optimizing the previous level V^{ℓ}.

Next, from (23), the resulting Bayesian estimator of image-class labels, denoted as $\hat{\boldsymbol{x}}_i$, is

$$(\forall i\in V)\ \hat{\boldsymbol{x}}_i=\arg\max_{k\in M} m_i^k, \tag{27}$$

where the approximate posterior probability m_i^k that image class k is assigned to node i is given by (16).

Finally, from (24), optimal node positions are estimated $\forall\ell>0$, and $\forall i\in V^{\ell}$ as

$$\hat{\boldsymbol{r}}_i=\arg\max_{\boldsymbol{r}_i}\sum_{Z}Q(\boldsymbol{r}_i|Z)Q(Z)=\sum_{j\in\{0,V^{\ell+1}\}}\boldsymbol{\mu}_{ij}\xi_{ij}, \tag{28}$$

where $\boldsymbol{\mu}_{ij}$ and ξ_{ij} are given by (17) and (19), respectively.

The inference algorithm for DTs is summarized in Fig. 2. The specified ordering of parameter updates for $Q(Z)$, $Q(X|Z)$, and $Q(R'|Z)$ in Fig. 2, Steps 4-10, is arbitrary; theoretically, other orderings are equally valid.

4.4 Specification of Model Parameters

Variational inference presumes that parameters V, L, M, and $\Theta=\{\gamma_{ij},\boldsymbol{d}_{ij},\Sigma_{ij},P_{ij}^{kl},\theta\}$, $\forall i,j\in V$, $\forall k,l\in M$, are available. Due to the lack of example images in unsupervised settings, we are not in a position to learn these parameters on a training image set. This problem has been addressed in the literature with indecisive results (e.g., [31], [32], [33]). In the absence of prior application knowledge, multiple solutions are equally reasonable, as even human interpreters arrive at different answers [33].

First, for the given number of leaf-level nodes $|V^0|$, we set $L=\log_2(|V^0|)$. Next, due to a huge diversity of possible configurations of objects in images, for each node $i\in V^{\ell}$, we set γ_{ij} to be uniform over is candidate parents $\forall j\in\{0,V^{\ell+1}\}$. Then, for all pairs $(i,j)\in V^{\ell}\times V^{\ell+1}$ at level ℓ, we set $\boldsymbol{d}_{ij}=\boldsymbol{\rho}(i)-\boldsymbol{\rho}(j)$—namely, the $\boldsymbol{d}_{ij}$ are initialized to the relative displacement of the centers of mass of the ith and jth dyadic squares in the corresponding quad-tree with L levels, specified in (5). For roots i, we have $\boldsymbol{d}_i=\boldsymbol{\rho}(i)$. Also, we set diagonal elements of $\boldsymbol{\Sigma}_{ij}$ to the diagonal elements of a matrix $\boldsymbol{d}_{ij}\boldsymbol{d}_{ij}^{\mathrm{T}}$. The number of components G_k in a Gaussian mixture for each class k is set to $G_k=3$, which is empirically validated to be appropriate.

Now, the most critical parameters that remain to be specified are the number of image classes $|M|$, conditional probability tables P_{ij}^{kl}, and the parameters of a Gaussian-mixture density θ. For this purpose, we conduct an iterative learning procedure using the EM algorithm on the quad-tree thoroughly discussed in ([16], pp. 399-401). Given V, L, Y, and $|M|$ of the quad-tree, the algorithm readily estimates P_{ij}^{kl} and θ, for a given image. Here, P_{ij}^{kl} and θ are equal for all levels.

Assume that V, L, M, Θ, N_{ε}, ε, and ε_{μ} are given.

(1) Initialization: $t=0$; $t_{\text{in}}=0$; $(\forall i,j\in V)\ (\forall k,l\in M)\ \xi_{ij}(0)=\gamma_{ij}$; $Q_{ij}^{kl}(0)=P_{ij}^{kl}$; $\boldsymbol{\mu}_{ij}(0)$ are set to node locations in the corresponding quad-tree; diagonal elements of $\boldsymbol{\Omega}_{ij}(0)$ are set to the area of dyadic squares in the corresponding quad-tree;

(2) **repeat** Outer Loop

(3) $t=t+1$;

(4) *Bottom-up* pass: for $\ell=0,1,...,L-1$, $\forall i\in V^{\ell}$, $\forall k\in M$, compute $\lambda_i^k(t)$ as in Eq. (13); $Q_{ij}^{kl}(t)$ as in Eq. (12);

(5) *Top-down* pass: for $\ell=L-1,L-2,...,0$, $\forall i\in V^{\ell}$, $\forall k\in M$, compute $m_i^k(t)$ given by Eq. (14);

(6) **repeat** Inner Loop

(7) $t_{\text{in}}=t_{\text{in}}+1$;

(8) Compute $\forall i,j\in V'$,

$\boldsymbol{\mu}_{ij}(t_{\text{in}})$ as in Eq. (15);

$\Omega_{ij}(t_{\text{in}})$ as in Eq. (16);

(9) **until** $|\boldsymbol{\mu}_{ij}(t_{\text{in}})-\boldsymbol{\mu}_{ij}(t_{\text{in}}-1)|/\boldsymbol{\mu}_{ij}(t_{\text{in}}-1)<\varepsilon_{\mu}$;

(10) Compute $\forall i,j\in V'$, $\xi_{ij}(t)$ as in Eq. (17);

(11) **until** $\frac{|Q(Z,X,R';t)-Q(Z,X,R';t-1)|}{Q(Z,X,R';t-1)}<\varepsilon$, for N_{ε} consecutive steps ;

(12) Estimation of $\hat{Z}$:

compute in *bottom-up* pass for $\ell=0,1,...,L-1$,

for DT_{V^0}: $(\forall i\in V^{\ell})(\hat{Z}_{\cdot i}\neq 0)\ \hat{j}=\arg\max_{j\in\{0,V^{\ell+1}\}}\xi_{ij}(t)$,

for DT_V: $(\forall i\in V^{\ell})\ \hat{j}=\arg\max_{j\in\{0,V^{\ell+1}\}}\xi_{ij}(t)$;

(13) Estimation of $\hat{X}$: $(\forall i\in V)\ \hat{x}_i=\arg\max_{k\in M} m_i^k(t)$;

(14) Estimation of $\hat{R}'$:

$(\forall\ell>0)(\forall i\in V^{\ell})\hat{\boldsymbol{r}}_i=\sum_{j\in\{0,V^{\ell+1}\}}\boldsymbol{\mu}_{ij}(t)\xi_{ij}(t)$;

Fig. 2. Inference of the DT given Y, R^0, and Θ; t and t_{in} are counters in the outer and inner loops, respectively; N_{ε}, ε, and ε_{μ} control the convergence criteria for the two loops.

Once estimated, these values can be used to optimize $|M|$. Then, for the new $|M|$ value, we again conduct the EM algorithm on the quad-tree, and so forth. To optimize $|M|$, we assume that $P(|M|)$ is the Poisson distribution, with the mean $\mathbb{E}\{|M|\}=2$—the assumption stemming from our initial guess that each image contains at least two image regions, and that large values of $|M|$ should be penalized due to computational complexity. We optimize $|M|$ by maximizing the function $f(|M|)=P(Y|X,\boldsymbol{\rho},|M|)P(|M|)$ for $|M|=2,3,4,$, where likelihood $P(Y|X,\boldsymbol{\rho},|M|)=\prod_i P(\boldsymbol{y}_{\boldsymbol{\rho}(i)}|x_i,\boldsymbol{\rho}(i),|M|)$ is computed from the results of the EM algorithm on the quad-tree. Since larger $|M|$ values give larger $P(Y|X,\boldsymbol{\rho},|M|)$, $f(|M|)$ increases until some maximum $|M|^*$, when the Poisson distribution of $P(|M|)$ starts to dominate decreasing $f(|M|)$ for $|M|>|M|^*$. Note that $P(|M||X,Y)\propto f(|M|)$, so that the maximum of $f(|M|)$, $|M|^*$, gives the MAP solution to our parameter estimation problem. This iterative learning algorithm stops when $|M|^*$ is reached, yielding also P_{ij}^{kl} and θ parameters. Our experiments show that $|M|^*$ is on average a conservative

estimate of the true number of classes. Since DTs are generalized quad-trees, our experimentation suggests that this optimization with respect to the quad-tree is justified.

4.5 Implementation Issues

In this section, we list algorithm-related details that are necessary for the experimental results, presented in Section 6, to be reproducible. Other specifications, such as, for example, feature extraction, will be detailed in Section 6.

First, direct implementation of (14) would result in numerical underflow. Therefore, we introduce the following scaling procedure: $\tilde{\lambda}_i^k = \lambda_i^k / S_i$, $\forall i \in V$, $\forall k \in M$, where $S_i \triangleq \sum_{k\in M} \lambda_i^k$. Substituting the scaled $\tilde{\lambda}$s into (14), we obtain

$$Q_{ij}^{kl} = \frac{P_{ij}^{kl}\lambda_i^k}{\sum_{a\in M} P_{ij}^{al}\lambda_i^a} = \frac{P_{ij}^{kl}\tilde{\lambda}_i^k}{\sum_{a\in M} P_{ij}^{al}\tilde{\lambda}_i^a}.$$

In other words, computation of Q_{ij}^{kl} does not change when the scaled $\tilde{\lambda}'s$ are used.

Second, to reduce computational complexity, as is done in [4], we consider, for each node i, only the 7×7 box encompassing parent nodes j that neighbor the parent of the corresponding quad-tree. Consequently, the number of possible children nodes c of i is also limited. Our experiments show that the omitted nodes, either children or parents, contribute negligibly to the update equations. Thus, we limit overall computational cost as the number of nodes increases.

Finally, the convergence criterion of the inner loop, where $\boldsymbol{\mu}_{ij}$ and Ω_{ij} are computed, is controlled by parameter ε_μ. When $\varepsilon_\mu = 0.01$, the average number of iteration steps, t_{in}, in the inner loop, is from 3 to 5, depending on the image size, where the latter is obtained for 128×128 images. The convergence criterion of the outer loop is controlled by parameters N_ε and ε. The aforementioned simplifications, which we employ in practice, may lead to suboptimal solutions of SVA. From our experience, though, the algorithm recovers from unstable stationary points for sufficiently large N_ε. In our experiments, we set $N_\varepsilon = 10$ and $\varepsilon = 0.01$.

After the inference algorithm converged, we then estimate the DT structure for a given image, which consists of DT subtrees representing distinct objects in that image. Having found the DT representation of segmented image regions, we are then in a position to measure the similarity of the detected objects across a given set of images.

5 Stochastic Similarity Measure

Recently, similarity measures between two statistical models have been given considerable attention in the literature [8], [9], [10], [11], [12], [13]. To compare a pair of image regions represented by statistical models, in standard probabilistic approaches, one examines the log-ratio $\log P_1/P_2$ or the expected value of the log-ratio $\langle \log P_1/P_2 \rangle$, where P_1 and P_2 are some distributions of the two models (e.g., likelihoods, posteriors, cross probabilities). For instance, Moghaddam [8], for the purposes of face recognition, proposes a similarity measure expressed in terms of probabilities of *intrapersonal* and *extrapersonal* facial image variations. Hermosillo et al. [9] perform matching of images by computing the variational gradient of a hierarchy of statistical similarity measures. These approaches can be viewed as a form of ML or MAP principles. Also, a symmetric function of the KL divergence, $\langle \log P_1/P_2 \rangle_{P_1} + \langle \log P_2/P_1 \rangle_{P_2}$, has been proposed in [10].

Since the size of objects determines the number of random variables in DTs, the log-ratios may turn out to be equal for two different scenarios: When subtrees represent different objects of similar size and when subtrees represent the same object appearing at different size across the images. The same problem has also been discussed in [12], [13], where hidden Markov models of different-length observation sequences have been compared. Therefore, for each pair of image regions, it is necessary to normalize the probability log-ratios. Usually, this is done by multiplying the log-ratio with a suitable constant $\kappa > 0$. Since the κs are different for every pair of compared regions, a decision criterion based on the normalized log-ratios becomes nonprobabilistic.[1] Our experiments on image matching show that this normalization improves performance over matching when probability log-ratios are not normalized.

There are several disadvantages of the outlined approach to image matching. We have observed great sensitivity of $\langle \kappa \log P_1/P_2 \rangle$ to the specification of the optimal κ. Furthermore, matching approaches based on computing $\langle \kappa \log P_1/P_2 \rangle$ implicitly assume that distributions P_1 and P_2 are reliable representations of underlying image processes, which is justifiable for supervised settings. However, this is not the case for unsupervised settings, where P_1 and P_2 may have huge variations over the examined images, due to the uninformed estimation of the prior distribution, that is, in our case, model parameters Θ.

To mitigate the sensitivity to κ, as well as to variations in Θ across images, we propose a novel similarity measure, thereby departing from the outlined approaches where probability log-ratios are used. Thus, in our approach, the impact of unreliable Θ is neutralized by measuring correlation between the cross-likelihoods of the two image regions, which are normalized by the likelihoods of each individual region. Below, we mathematically formulate this idea.

Let Y_t and Y_r denote observables of two DT subtrees $\mathcal{T}_t$ and $\mathcal{T}_r$, respectively, where t in the subscript refers to an image region in the test image, and r, to a region in the reference image, as defined in Section 1. Here, $\mathcal{T}_t$ refers to the estimated configuration $(\hat{Z}_t, \hat{X}_t, \hat{R}_t')$ and the parameter set Θ_t for the test image. Similarly, $\mathcal{T}_r$ refers to the estimated configuration $(\hat{Z}_r, \hat{X}_r, \hat{R}_r')$ and the parameter set Θ_r for the reference image. As discussed above, we normalize the likelihood $P(Y|X, \boldsymbol{\rho}, \Theta)$, given by (6), as $P_{tr} \triangleq P(Y_t|\hat{X}_r, \boldsymbol{\rho}_r, \Theta_r)^{1/C_r}$, where C_r denotes the cardinality of the model $\mathcal{T}_r$. Since the Ys at coarser resolutions affect more pixels than at finer scales, for DT_V, we compute the cardinality as $C \triangleq \sum_{\ell=0}^{L-1} \sum_{i\in\mathcal{T}} \mathcal{K}_i^\ell$, where $\mathcal{K}_i^\ell$ denotes the size of the kernel used to compute observables at level ℓ; for example, $\mathcal{K}_i^\ell = 2^\ell$ for wavelet coefficients. For DT_{V^0}, C is equal to the number of leaf-level nodes. We now define the similarity measure between two models as

$$\sigma_{tr} \triangleq \sqrt{\frac{P_{tr}P_{rt}}{P_{tt}P_{rr}}}. \tag{29}$$

The defined similarity measure exhibits the following properties: 1) by definition $\sigma_{tr} = \sigma_{rt}$, 2) from $0 < P_{tr} \le P_{tt}$ and $0 < P_{rt} \le P_{rr}$ it follows that $0 < \sigma_{tr} \le 1$ and, finally, 3) if $\mathcal{T}_t \equiv \mathcal{T}_r$ than $\sigma_{tr} = 1$. Note that, for property 2), we assume that the inference algorithm guarantees that P_{tt} and P_{rr} are global

1. Scaled P_1 and P_2 do not satisfy the three axioms of probability over the total set of events if the κs vary for different events in that set.

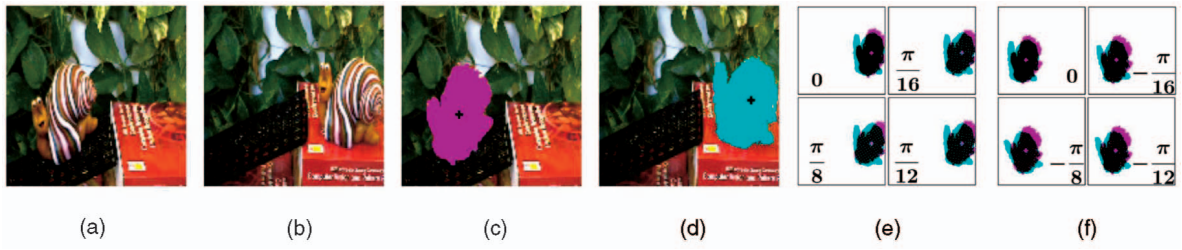

Fig. 3. Alignment tests: (a) and (b) 128×128 test and reference images, (c) segmented region under $\mathcal{T}_t$ using DT_{V^0}, (d) segmented region under $\mathcal{T}_r$ using DT_{V^0}, (e) image regions in the reference image used for substitution $\boldsymbol{y}_{\boldsymbol{\rho}_t(i)} \leftarrow \boldsymbol{y}_{\boldsymbol{\rho}_r(i)}$ for different α_t, and (f) image regions in the test image used for substitution $\boldsymbol{y}_{\boldsymbol{\rho}_r(i)} \leftarrow \boldsymbol{y}_{\boldsymbol{\rho}_t(i)}$ for different α_r. The crosses mark the estimated roots' positions $\hat{\boldsymbol{r}}_t$ and $\hat{\boldsymbol{r}}_r$.

maxima for the test and reference images, respectively. In practice, from our experience, this is not a significant concern, as the algorithm converges to near-optimal solutions, as discussed in Section 4.5.

In computation of the cross probabilities, say, P_{tr}, it is necessary to substitute observables Y_r with Y_t in the estimated subtree structure $(\hat{Z}_r, \hat{X}_r, \hat{R}'_r)$ according to a specified mapping. While a complete treatment of possible mappings is beyond the scope of this paper, below we consider one plausible approach. For this purpose, it is convenient to index observables in terms of their locations in the image. Recall, in Section 3, we define locations of observables $\boldsymbol{\rho}(i)$, given by (5). Thus, the mapping can be conducted as follows: For each observable node i in $\mathcal{T}_r$ that is on location $\boldsymbol{\rho}_r(i)$ in the reference image, we first find the corresponding location in the test image $\boldsymbol{\rho}_t(i)$ and then substitute $\boldsymbol{y}_{\boldsymbol{\rho}_r(i)} \leftarrow \boldsymbol{y}_{\boldsymbol{\rho}_t(i)}$. We define the correspondence between the locations $\boldsymbol{\rho}_r(i)$ and $\boldsymbol{\rho}_t(i)$, as follows:

$$\boldsymbol{\rho}_t(i)=\left\lfloor \hat{\boldsymbol{r}}_t+\begin{bmatrix}\cos(\alpha_r) & -\sin(\alpha_r)\\ \sin(\alpha_r) & \cos(\alpha_r)\end{bmatrix}(\boldsymbol{\rho}_r(i)-\hat{\boldsymbol{r}}_r)\right\rfloor_{2^\ell}, \qquad (30)$$

where α_r is a rotation angle; $\hat{\boldsymbol{r}}_t$ and $\hat{\boldsymbol{r}}_r$ are estimated positions of the roots of $\mathcal{T}_t$ and $\mathcal{T}_r$ in the test and reference images; $\lfloor\cdot\rfloor_{2^\ell}$ finds integer, multiples-of-2 values of the form given by (5).

Pictorially, computation of P_{tr} can be viewed as alignment of $\mathcal{T}_r$ with the location of $\mathcal{T}_t$ in the test image. Thus, according to (30), we first translate $\mathcal{T}_r$ until the root of $\mathcal{T}_r$ coincides with the root of $\mathcal{T}_t$ in the reference image.[2] After the roots are aligned, we then rotate $\mathcal{T}_r$ for several angles α_r about the vertical axis containing the roots. A similar expression to (30) holds for translation and rotation of $\mathcal{T}_t$ in the reference image, when computing P_{rt}. Note that, because of the rotation, we compute two arrays of cross probabilities, $P_{rt}(\alpha_t)$ and $P_{tr}(\alpha_r)$, for each finite rotation increment of α_t and α_r. Although we could eliminate either α_t or α_r, we do not because of finite rotation increments that may differ for the two parameters. We emphasize that the outlined translation/rotation is just a visual interpretation of the mapping, in which one set of observables is substituted by the other; however, this mapping should not be misunderstood as transformation of already estimated dynamic trees.

Due to the mapping, given by (30), when computing $P_{tr}(\alpha_r)$, locations of observables $\boldsymbol{\rho}_t(i)$ may fall outside the boundaries of the test image. In this case, it is necessary to prune that observable node i in $\mathcal{T}_r$ (rectangular-shaped node) and its corresponding node characterized by hidden variables (round-shaped node). This deletion of nodes gives rise to a number of tree-pruning strategies. In our approach, for DT_V, we simply delete outlying rectangular-shaped nodes and their corresponding round-shaped nodes; other nodes are kept intact. For DT_{V^0}, the deletion of nodes at the leaf level, V^0, may leave some higher-level nodes without children. To preserve the generative property, as discussed in Section 4.3, from $\mathcal{T}_r$, we also prune, in the bottom-up sweep, those nodes that happen to lose all their children. A similar pruning procedure is necessary when computing $P_{rt}(\alpha_t)$. In Fig. 3, we illustrate alignment tests pictorially for two sample images.

Having defined our similarity measure, we are now in position to conduct matching of segmented image regions across a given set of images.

6 Experiments

We report experiments on segmentation and matching of image regions for five sets of images. Data Set I contains 50, 4×4, binary images with a total of 50 single object appearances. A sample of Data Set I images is depicted in Fig. 4a (top). Data Set II consists of 50, 8×8, binary images with a total

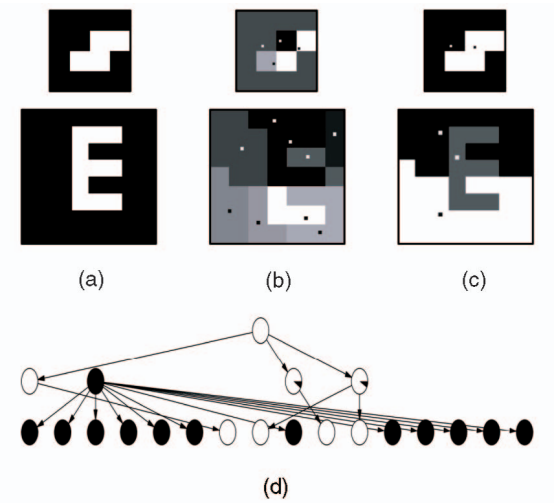

Fig. 4. Image segmentation using DT_{V^0}: (a) sample 4×4 and 8×8 binary images, (b) clustered leaf-level pixels that have the same parent at level $\ell=1$, (c) clustered leaf-level pixels that have the same grandparent at level $\ell=2$; clusters are indicated by different shades of gray; the point in each region marks the position of the parent node, and (d) estimated DT structure for the 4×4 image in (a); nodes are depicted inline representing 4, 2, and 1 actual rows of the levels 0, 1, and 2, respectively; nodes are drawn as pie-charts representing $P(x_i^k=1)$, $k\in\{0,1\}$; note that there are two roots representing two distinct objects.

2. As we demonstrate in Section 6, roots' positions give a good estimate of the center of mass of true object appearances in the image; therefore, we align the roots—not the centers of mass of segmented image regions under $\mathcal{T}_r$ and $\mathcal{T}_t$.

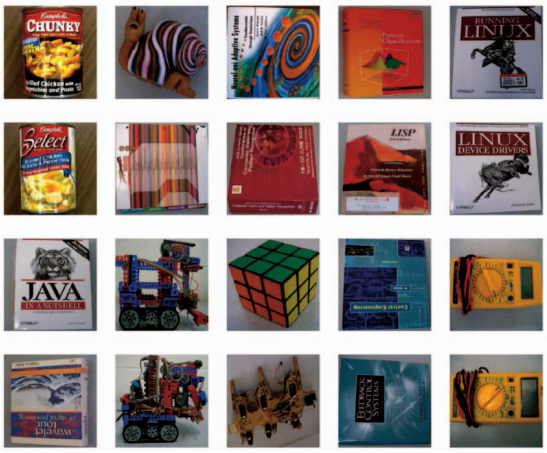

Fig. 5. Twenty image classes in Type III and IV Data Sets.

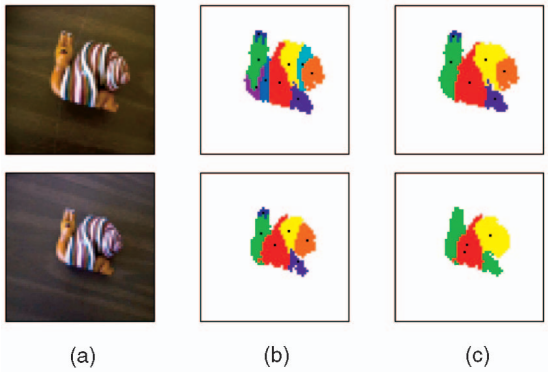

Fig. 6. Image segmentation using DT_{V^0}: (a) Data Set III images, (b) pixel clusters with the same parent at level ℓ=3, (c) pixel clusters with the same parent at level ℓ=4, points mark the position of parent nodes. DT structure is preserved through scales.

of 78 multiple object appearances. A sample of Data Set II images is shown in Fig. 4a(bottom). Data Set III comprises 50, 64×64, simple indoor-scene, color images with a total of 105 object appearances of 20 distinct objects shown in Fig. 5. Samples of Data Set III images are given in Figs. 6, 7, and 9. Data Set IV contains 50, 128×128, challenging indoor-scene, color images with a total of 223 partially occluded object appearances of the same 20 distinct objects as for Data Set III images. Examples of Data Set IV images are shown in Figs. 3 and 10. Note that objects appearing in Data Sets III and IV are carefully chosen to test if DTs are expressive enough to capture very small variations in appearances of some classes (e.g., two different types of cans in Fig. 5), as well as to encode large differences among some other classes (e.g., wiry-featured robot and books in Fig. 5). Finally, Data Set V contains 50, 128×128, natural-scene, color images with a total of 297 object appearances, samples of which are shown in Figs. 8, 11, and 13. Ground truth in images is obtained through hand-labeling of pixels.

For Data Sets I and II, we experiment only with DT_{V^0} models, with observables Y given by binary pixel values. For the other data sets, we test both DT_{V^0} and DT_V. To compute the Ys, we account for both color and texture cues. For texture analysis in DT_V, we choose the complex wavelet transform (CWT) applied to the intensity (gray-scale) image, due to its shift-invariant representation of texture at different scales, orientations, and locations [34]. The CWT's directional selectivity is encoded in six subimages of coefficients oriented

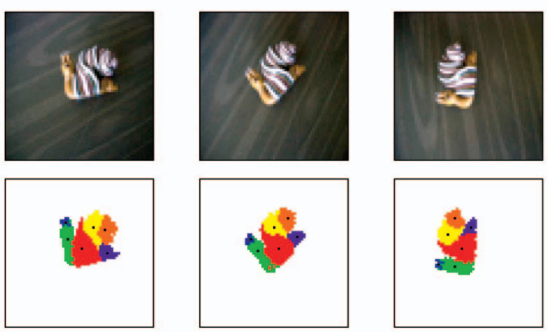

Fig. 7. Image segmentation using DT_{V^0}: (top) Data Set III images and (bottom) pixel clusters with the same parent at level 3. DT structure is preserved over rotations.

at angles $\pm15^\circ$, $\pm45^\circ$, and $\pm75^\circ$. For texture extraction in DT_{V^0}, we compute the difference-of-Gaussian function convolved with the image: $D(x,y,k,\sigma)=(G(x,y,k\sigma)-G(x,y,\sigma))*I(x,y)$, where x and y represent pixel coordinates, $G(x,y,\sigma)=\exp(-(x^2+y^2)/2\sigma^2)/2\pi\sigma^2$ and $I(x,y)$ is the intensity image. In addition to reduced computational complexity, as compared to the CWT, the function D provides a close approximation to the scale-normalized Laplacian of Gaussian, $\sigma^2\nabla^2G$, which has been shown to produce the most stable image features across scales when compared to a range of other possible image functions, such as the gradient and the Hessian [35], [36]. We compute $D(x,y,k,\sigma)$ for three scales $k=\sqrt{2},2,\sqrt{8}$, and $\sigma=2$. For color features in both DT_V and DT_{V^0}, we choose the generalized RGB color space: $r=R/(R+G+B)$ and $g=G/(R+G+B)$, which effectively normalizes variations in brightness; the Ys of higher-level nodes are computed as the mean of the rs and gs of their children nodes of the initial quad-tree structure. Each color observable is normalized to have zero mean and unit variance over the data set. Thus, the $\boldsymbol{y}_{\boldsymbol{\rho}(i)}$s are eight and five-dimensional vectors for DT_V and DT_{V^0}, respectively.

In the following experiments, we compare our SVA inference algorithm with three other inference algorithms: 1) Gibbs sampling discussed in [29], 2) mean-field variational approximation (MFVA) proposed in [1], and 3) variational approximation (VA)[3] discussed in [4]. All the figures in this section illustrate segmentation and matching performance when DTs are inferred using our SVA algorithm.

6.1 Image Segmentation Tests

DT-based image segmentation is tested on all five data sets. Results presented in Figs. 4, 5, 6, 7, and 8 suggest that DTs are able to encode component-subcomponent relationships among objects and their parts in the image. From Fig. 8, we observe that nodes at different levels of the dynamic tree can be interpreted as object parts at various scales. Moreover, from Figs. 6 and 7, we also observe that DTs, inferred through SVA, preserve structure for objects across images subject to translation, rotation and scaling. In Fig. 6, note that the level-4 clustering for the larger-object scale in Fig. 6c (top) corresponds to the level-3 clustering for the smaller-object scale in Fig. 6b. In other words, as the object transitions through scales, the tree structure changes by eliminating the lowest-level layer, while the higher-order structure remains intact.

3. Although the algorithm proposed in [4] is also structured variational approximation, to differentiate that method from ours, we slightly abuse the notation.

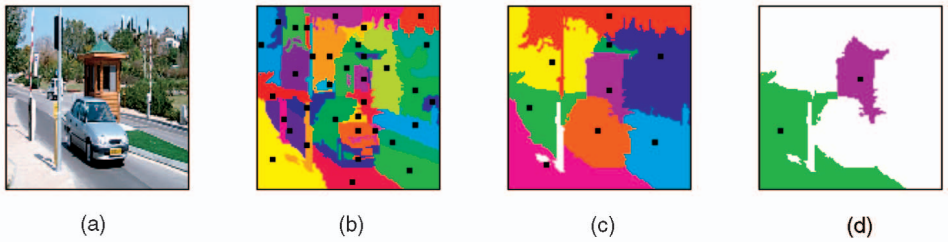

Fig. 8. Image segmentation using DT_V: (a) a Data Set V image, (b), (c), and (d) pixel clusters with the same parent at levels $\ell=3,4,5$, respectively, white regions represent pixels already grouped by roots at the previous scale; points mark the position of parent nodes; nodes at different levels of DT_V can be interpreted as object parts at various scales.

We also note that the estimated positions of higher-level hidden variables are very close to the center of mass of object parts, as well as of whole objects. We compute the error of estimated root-node positions $\hat{\boldsymbol{r}}$ as the distance from the actual center of mass $\boldsymbol{r}_{CM}$ of hand-labeled objects, $d_{err}=||\hat{\boldsymbol{r}}-\boldsymbol{r}_{CM}||$. The averaged error values over the given test images for VA and SVA are reported in Table 1. We observe that the error significantly decreases as the image size increases because in summing node positions over parent and children nodes, as in (17) and (18), more statistically significant information contributes to the position estimates. For example, $d_{err}^{IV}=6.18$ for SVA is only 4.8 percent of the Data Set IV image size, whereas $d_{err}^{III}=4.23$ for SVA is 6.6 percent of the Data Set III image size.

Typical results of the DT-based image segmentation for Data Sets III, IV, and V are shown in Figs. 9, 10, and 11. In Table 2, we report the percentage of erroneously grouped pixels, and, in Table 3, we report the object detection error, when compared to ground truth, averaged over each data set. For estimating the object detection error, the following instances are counted as error: 1) merging two distinct objects into one (i.e., failure to detect an object) and 2) segmenting an object into subregions that are not actual object parts. On the other hand, if an object is segmented into several "meaningful" subregions, verified by visual inspection, this type of error is not included. The averaged pixel error for Gibbs sampling is 6 percent for both type I and II images, while for MFVA, it is 18 percent and 12 percent for the Data Sets I and II, respectively. With regard to object detection error, Gibbs sampling yields no error in the Data Set I, and wrongly segments seven objects in the Data Set II (8 percent). The object detection error for MFVA is 1 undetected object in the Data Set I (2 percent), and 13 merged/undetected objects in the Data Set II (16 percent). With the increase in image size, Gibbs sampling becomes infeasible and MFVA exhibits very poor

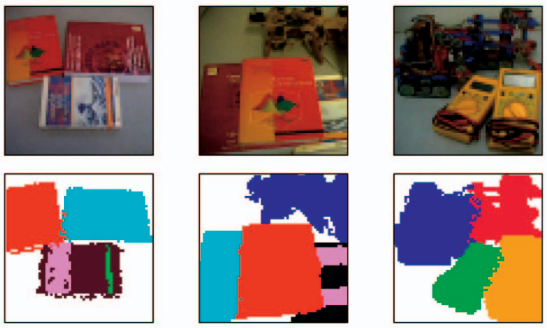

Fig. 9. Image segmentation by DT_{V^0} learned using SVA for Data Set III images; all pixels labeled with the same color are descendants of a unique root.

TABLE 1
Root-Node Distance Error

	DT_{V^0}		DT_V	
	VA	SVA	VA	SVA
II	2.15	1.91	2.11	1.82
III	6.32	4.61	6.14	4.23
IV	9.15	6.87	8.99	6.18
V	11.76	8.03	10.52	7.24

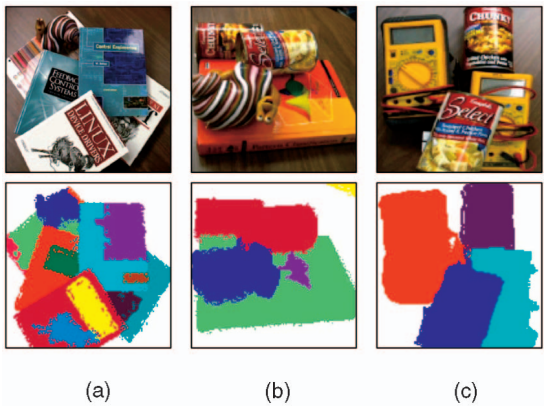

Fig. 10. Image segmentation by DT_{V^0} learned using SVA for Data Set IV images, (b) negative example, where, due to challenging similarity in appearance and occlusion, the DT merges two distinct objects into one; all pixels labeled with the same color are descendants of a unique root.

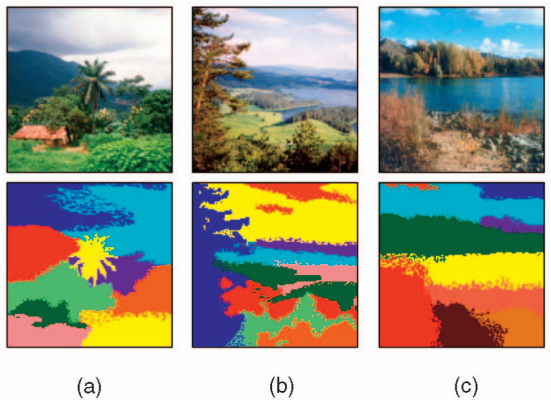

Fig. 11. Image segmentation by DTs learned using SVA for Data Set V: (a) DT_{V^0}, (b) and (c) DT_V, all pixels labeled with the same color are descendants of a unique root.

TABLE 2
Percent of Erroneously Grouped Pixels

		I	II	III	IV	V
DT_{V^0}	VA	6%	8%	7%	10%	14%
	SVA	6%	6%	4%	9%	10%
DT_V	VA	NA	NA	7%	11%	13%
	SVA	NA	NA	4%	7%	8%

TABLE 3
Object Detection Error

		I	II	III	IV	V
DT_{V^0}	VA	2%	8%	4%	13%	12%
	SVA	2%	8%	3%	10%	9%
DT_V	VA	NA	NA	4%	10%	11%
	SVA	NA	NA	2%	6%	7%

performance; therefore, Tables 2 and 3 report results only for VA and SVA. Overall, we observe that SVA outperforms other inference algorithms for image segmentation using DTs. Interestingly, the segmentation results for DT_V models are only slightly better than for DT_{V^0} models.

It should be emphasized that our experiments are carried out in an *unsupervised* setting, and, as such, cannot be equitably evaluated against *supervised* object recognition results reported in the literature. Take, for instance, the segmentation in Fig. 10b, where two overlapping, similar-looking objects are merged into one DT subtree. Given the absence of prior knowledge, the ground-truth segmentation for this image is arbitrary, and the resulting segmentation ambiguous; nevertheless, we still count it toward the object-detection error percentages in Table 3.

Next, in Figs. 12a and 12b, we illustrate the convergence rate of computing $P(Z, X, R'|Y, R^0) \approx Q(Z, X, R')$ for the four inference algorithms, averaged over the given data sets. Numbers above bars represent the mean number of iteration steps it takes for a given algorithm to converge. For all the approaches, we consider the algorithm converged when $|Q(Z, X, R'; t+1) - Q(Z, X, R'; t)| / Q(Z, X, R'; t) < \varepsilon$ for N_ε consecutive iteration steps t, where $N_\varepsilon = 10$ and $\varepsilon = 0.01$ (see Fig. 2, Step (11)). Overall, SVA converges in the fewest number of iterations. The average number of iterations for SVA on Data Set V is 28 and 24 for DT_{V^0} and DT_V, respectively, which takes approximately 6s and 5s on a Dual 2 GHz PowerPC G5. Here, the processing time also includes image-feature extraction.

For the same experiments, in Figs. 12c and 12d, we report the percentage increase in $\log Q(Z, X, R')$ computed using our SVA over $\log Q(Z, X, R')$ obtained by Gibbs sampling, MFVA, and VA, respectively. We note that SVA results in larger approximate posteriors than MFVA and VA. The larger $\log Q(Z, X, R')$ means that the assumed form of the approximate posterior distribution $Q(Z, X, R') = Q(Z)Q(X|Z)Q(R'|Z)$ more accurately represents underlying stochastic processes in the image than VA and MFVA approximations. We note that SVA yields approximately the same $Q(Z, X, R')$ as Gibbs sampling.

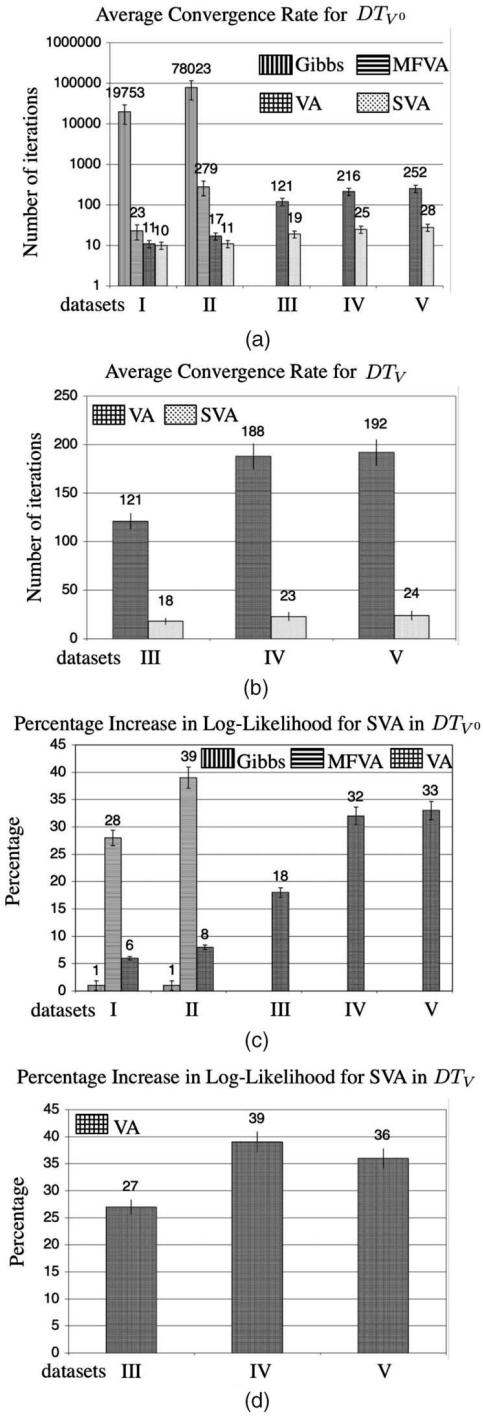

Fig. 12. Comparison of inference algorithms: (a) and (b) convergence rate averaged over the given data sets. (c) and (d) Percentage increase in $\log Q(Z, X, R')$ computed in SVA over $\log Q(Z, X, R')$ computed in Gibbs sampling, MFVA, VA, respectively.

6.2 Tests of Model Matching

We test our approach to model matching for object recognition in unsupervised settings for Data Sets III, IV, and V. As explained in Section 1, we consider a constrained type of object recognition, where we detect a particular appearance of a reference object in a test image. Given the assumption that each test image cannot contain multiple appearances of the same object, we conduct unsupervised object recognition as follows. One at a time, every image in

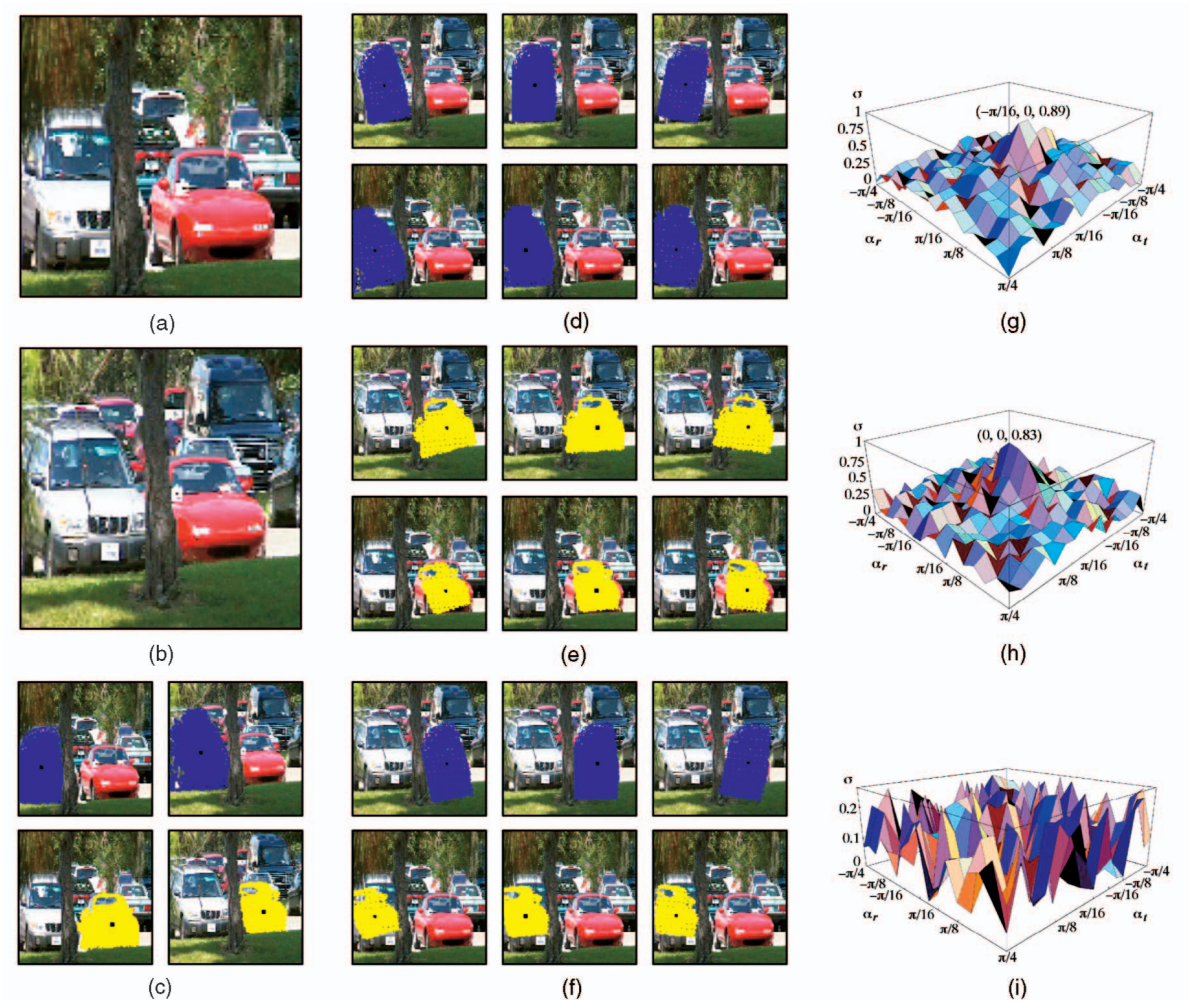

Fig. 13. Image matching for Data Set V images in (a) and (b). (c) Computation of P_{rr} and P_{tt} for a sample of two segmented image regions in the reference and test images, respectively, (d) and (e) computation of $P_{tr}(\alpha_r)$ and $P_{rt}(\alpha_t)$ when $\mathcal{T}_r$ and $\mathcal{T}_t$ represent the same object, (f) computation of $P_{tr}(\alpha_r)$ and $P_{rt}(\alpha_t)$ when $\mathcal{T}_r$ and $\mathcal{T}_t$ represent different objects, (g) and (h) 3D plots of $\sigma_{tr}(\alpha_t, \alpha_r)$ for $\alpha_t, \alpha_r \in \{-\pi/4, \pi/4\}$, where $(\alpha_r, \alpha_t, \sigma)$ marks the maximum. (a) Reference image. (b) Test image. (c) P_{rr} (left), P_{tt} (right). (d) $P_{tr}(\alpha_r)$ (top) $P_{rt}(\alpha_t)$ (bottom). (e) $P_{tr}(\alpha_r)$ (top) $P_{rt}(\alpha_t)$ (bottom). (f) $P_{tr}(\alpha_r)$ (top) $P_{rt}(\alpha_t)$ (bottom). (g) $\sigma_{tr}(\alpha_t, \alpha_r)$ plot for the $(\mathcal{T}_t, \mathcal{T}_r)$ pair in (d). (h) $\sigma_{tr}(\alpha_t, \alpha_r)$ plot for the $(\mathcal{T}_t, \mathcal{T}_r)$ pair in (e). (i) $\sigma_{tr}(\alpha_t, \alpha_r)$ plot for the $(\mathcal{T}_t, \mathcal{T}_r)$ pair in (f).

a given data set is chosen as the reference, while the rest of the images are then marked as test images. After DT-based image segmentation of the reference and test images, for a given $\mathcal{T}_r$, we search for the maximum $\sigma_{tr}(\alpha_t, \alpha_r)$ over all possible image regions under $\mathcal{T}_t$ and rotational alignments (α_t, α_r), as illustrated in Fig. 13. Note that α_t and α_r should be related by $\alpha_t = -\alpha_r$, provided the compared objects are identical. Thus, the test image region under $\mathcal{T}_t$, for which $\sigma_{tr}(\alpha_t, \alpha_r)$ is maximum and $|\alpha_t + \alpha_r| \leq \varepsilon$, where $\varepsilon = \pi/16$ is a rotation increment in the alignment tests, is recognized as the reference image region under $\mathcal{T}_r$. From Figs. 13g and 13h, we observe that σ_{tr} is a "peaky" function, reaching its maximum when the same objects are matched.

To compare our approach with methods which use probability log-ratios for image matching, we repeat the aforementioned set of experiments, but now using the symmetric KL distance d_{tr}, specified in [10] as

$$d_{tr} \approx \frac{1}{N_t} \sum_{i \in \mathcal{T}_t} \log \frac{P(\boldsymbol{y}_{\boldsymbol{\rho}_t(i)} | x_{it})}{P(\boldsymbol{y}_{\boldsymbol{\rho}_t(i)} | x_{ir})} + \frac{1}{N_r} \sum_{i \in \mathcal{T}_r} \log \frac{P(\boldsymbol{y}_{\boldsymbol{\rho}_r(i)} | x_{ir})}{P(\boldsymbol{y}_{\boldsymbol{\rho}_r(i)} | x_{it})},$$

where N_t and N_r are the number of observables in $\mathcal{T}_t$ and $\mathcal{T}_r$, respectively; $\boldsymbol{y}_{\boldsymbol{\rho}_t(i)}$ and $\boldsymbol{y}_{\boldsymbol{\rho}_r(i)}$ are observables in the test and reference images;and x_{it} and x_{ir} are image-class indicator random variables in $\mathcal{T}_t$ and $\mathcal{T}_r$, respectively. In light of the discussion in Section 5 on the necessity to normalize differently sized models, we also carry out the object recognition experiments using the normalized symmetric KL distance, $\tilde{d}_{tr}$, given by

$$\tilde{d}_{tr} \approx \frac{1}{N_t} \frac{C_r}{C_t} \sum_{i \in \mathcal{T}_t} \log \frac{P(\boldsymbol{y}_{\boldsymbol{\rho}_t(i)} | x_{it})}{P(\boldsymbol{y}_{\boldsymbol{\rho}_t(i)} | x_{ir})} + \frac{1}{N_r} \frac{C_t}{C_r} \sum_{i \in \mathcal{T}_r} \log \frac{P(\boldsymbol{y}_{\boldsymbol{\rho}_r(i)} | x_{ir})}{P(\boldsymbol{y}_{\boldsymbol{\rho}_r(i)} | x_{it})},$$

where C_t and C_r are the cardinality of $\mathcal{T}_t$ and $\mathcal{T}_r$, respectively, as defined in Section 5. For both distance measures, the image region under $\mathcal{T}_t$, for which d_{tr}, or $\tilde{d}_{tr}$, is closest to zero compared to the rest of the segmented regions in the test image, is recognized as $\mathcal{T}_r$.

In Table 4, we summarize our object recognition results using both DT_{V^0} and DT_V, inferred using SVA and VA, for σ_{tr}, d_{tr}, and $\tilde{d}_{tr}$. We define the recognition rate as the percentage of correctly detected appearances of an object in the total number of actual appearances of that object in the test images. The false detection rate is defined as the percentage of incorrectly detected appearances of an object in the total number of the detected appearances of that object in the test

TABLE 4
Recognition Rate (R) and False Detection Rate (F)

			dataset III		dataset IV		dataset V	
			R	F	R	F	R	F
σ_{tr}	DT_{V^0}	VA	92%	12%	83%	21%	89%	11%
		SVA	94%	5%	89%	17%	91%	10%
	DT_V	VA	91%	9%	81%	15%	90%	7%
		SVA	94%	4%	90%	18%	92%	7%
$\tilde{d}_{tr}$	DT_{V^0}	SVA	91%	18%	82%	22%	87%	14%
	DT_V	SVA	91%	14%	85%	19%	90%	21%
d_{tr}	DT_{V^0}	SVA	86%	28%	67%	23%	82%	27%
	DT_V	SVA	88%	13%	72%	18%	85%	21%

images. Here, ground truth is established by visual inspection. Recognition and false detection rates are averaged over all segmented regions and all images. Overall, we observe significantly better object recognition performance when σ_{tr} is used as a model-matching measure compared to d_{tr} and $\tilde{d}_{tr}$. Again, DT_V models outperform DT_{V^0} models.

7 Conclusion

In this paper, we presented a probabilistic framework for image segmentation and subsequent matching of segmented regions, when only weak or no prior knowledge is available. We proposed and demonstrated the use of Dynamic Trees (DTs) to address these problems. More precisely, we formulated image segmentation as inference of model posterior distributions, given an image, and subsequent Bayesian estimation of DT structure. Beyond this novel application of DTs, we built on previous DT work to formulate a novel DT architecture that introduces multilayered observable data into the model. For the proposed model, we derived a novel Structured Variational Approximation (SVA) inference algorithm that removes independence assumptions between node positions and model structure, as was done in prior work. Furthermore, we formulated image matching as similarity analysis between two DTs representing examined image regions. To conduct this analysis, we specified a novel similarity measure between two statistical models, which we find more suitable for unsupervised settings than measures based on probability log-ratios. We proposed one possible alignment procedure for comparing two DTs, and developed criteria based on the resulting similarity measure for ultimate unsupervised object recognition.

Through a set of detailed experiments, we demonstrated the significantly improved properties of the SVA algorithm—both in terms of substantially faster convergence rates, and larger approximate posteriors for the inferred models—when compared with competing inference algorithms. Our results show that DTs are capable of capturing important component-subcomponent relationships among objects and their parts, and, hence, that DTs perform well in segmenting images into plausible pixel clusters. Furthermore, we reported results on unsupervised object recognition, demonstrating the viability of the proposed similarity measure for matching statistical models.

This paper opens a number of research issues that need further investigation. First among these is the optimal alignment procedure required for comparing dynamic-structure models. Possible choices of the alignment procedure, ultimately, should look to enhance the discriminative power of the similarity measure—that is, how well the similarity measure distinguishes like objects (i.e., DT models) from dissimilar objects. Second, our experiments show (see Figs. 13g and 13h) that the proposed similarity measure, σ, is a "peaky" function, which suggests that σ can be successfully used in supervised settings. It is likely that using a suitable (learned) threshold for the classifier could improve object recognition results beyond those reported in Table 4. Next, we currently assume that node positions in DTs are uncorrelated (i.e. diagonal covariances) along "x" and "y" image coordinates, in order to facilitate the computation of (18). Often, this may not be an appropriate assumption, and we will further examine how to modify our inference algorithm to accommodate dependencies between coordinates. Finally, although DT_V type of models outperforms DT_{V^0} in every reported experiment, this may have been the result of more expressive texture extraction (i.e., the complex wavelet transform) used in DT_V than that used in DT_{V^0}. Further research is necessary for establishing when and why one model is better than the other.

Appendix

Derivation of Structured Variational Approximation

A.1 Notation

- $V=\{V', V^0\}$: set of all nodes; V^0: set of leaf-level nodes;
- $\boldsymbol{y}_{\boldsymbol{\rho}(i)}$: observable random vector at location $\boldsymbol{\rho}(i)$; $Y \triangleq \{\boldsymbol{y}_{\boldsymbol{\rho}(i)} | \forall i \in V\}$;
- z_{ij}: indicator random variable (RV) denoting a connection between nodes i and j; $Z \triangleq \{z_{ij} | \forall i, j \in V\}$; γ_{ij}: true probability of i being the child of j; ξ_{ij}: approximate probability of i being the child of j given Y and R^0;
- M: set of image classes; x_i^k: indicator RV denoting i is labeled as class $k \in M$; $X \triangleq \{x_i^k | \forall i \in V, k \in M\}$; P_{ij}^{kl}: true conditional probability tables; Q_{ij}^{kl}: approximate conditional probability tables given Y and R^0; m_i^k: approximate posterior that node i is labeled as image class k given Y and R^0;
- $\boldsymbol{r}_i$: position of node i; $R^0 \triangleq \{\boldsymbol{r}_i | \forall i \in V^0\}$; $R' \triangleq \{\boldsymbol{r}_i | \forall i \in V'\}$; Σ_{ij} and $\boldsymbol{d}_{ij}$: true diagonal covariance and mean of a relative child-parent displacement $(\boldsymbol{r}_i - \boldsymbol{r}_j)$; Ω_{ij} and $\boldsymbol{\mu}_{ij}$: approximate diagonal covariance and mean of $\boldsymbol{r}_i$, given that j is the parent of i and given Y and R^0;
- $\langle \cdot \rangle$: expectation value with respect to $Q(Z, X, R')$; κ: normalization constant; $pa(i)$: candidate parents of i; $c(i)$: children of i; $d(i)$: all descendants down the subtree of i.

A.2 Preliminaries

Computation of $KL(Q\|P)$, given by (13), is intractable, because it depends on $P(Z,X,R'|Y,R^0)$. Note, though, that $Q(Z,X,R')$ does not depend on $P(Y|R^0)$ and $P(R^0)$. Consequently, by subtracting $\log P(Y|R^0)$ and $\log P(R^0)$ from $KL(Q\|P)$, we obtain a tractable criterion $J(Q,P)$, whose minimization with respect to $Q(Z,X,R')$ yields the same solution as minimization of $KL(Q\|P)$:

$$\begin{aligned} J(Q,P) &\triangleq KL(Q\|P) - \log P(Y|R^0) - \log P(R^0), \\ &= \int_{R'} dR' \sum_{Z,X} Q(Z,X,R') \log \frac{Q(Z,X,R')}{P(Z,X,R,Y)}. \end{aligned} \tag{31}$$

$J(Q,P)$ is known alternatively as Helmholtz free energy, Gibbs free energy, or free energy [17], [19]. By minimizing $J(Q,P)$, we seek to compute parameters of approximate distributions $Q(Z)$, $Q(X|Z)$, and $Q(R'|Z)$. It is convenient, first, to reformulate (31) as $J(Q,P)=L_Z+L_X+L_R$, where $L_Z \triangleq \sum_Z Q(Z)\log\frac{Q(Z)}{P(Z)}$,

$$L_X \triangleq \sum_{Z,X} Q(Z)Q(X|Z)\log\frac{Q(X|Z)}{P(X|Z)P(Y|X,\boldsymbol{\rho})},$$

and

$$L_R \triangleq \int_{R'} dR'' \sum_{Z} Q(Z)Q(R'|Z)\log\frac{Q(R'|Z)}{P(R|Z)}.$$

To derive expressions for L_Z, L_X, and L_R, we first observe:

$$\begin{aligned} &\langle z_{ij}\rangle = \xi_{ij}, \langle x_i^k\rangle = m_i^k, \left\langle x_i^k x_j^l\right\rangle = Q_{ij}^{kl} m_j^l, \\ &\Rightarrow m_i^k = \sum_{j\in V}\xi_{ij}\sum_{l\in M} Q_{ij}^{kl} m_j^l, \forall i\in V, \forall k\in M, \end{aligned} \tag{32}$$

where $\langle\cdot\rangle$ denotes expectation with respect to $Q(Z,X,R')$. Consequently, from (1), (9), and (32), we have

$$L_Z = \sum_{ij\in V}\xi_{ij}\log[\xi_{ij}/\gamma_{ij}]. \tag{33}$$

Next, from (4), (10), and (32), we derive

$$\begin{aligned} L_X = &\sum_{i,j\in V}\sum_{k,l\in M}\xi_{ij}Q_{ij}^{kl}m_j^l\log[Q_{ij}^{kl}/P_{ij}^{kl}] \\ &-\sum_{i\in V}\sum_{k\in M} m_i^k \log P(\boldsymbol{y}_{\boldsymbol{\rho}(i)}|x_i^k,\boldsymbol{\rho}(i)). \end{aligned} \tag{34}$$

Note that for DT_{V^0}, V in the second term is substituted with V^0. Finally, from (3), (11), and (32), we get

$$\begin{aligned} L_R = \frac{1}{2}\sum_{i,j\in V'}\xi_{ij}\Big(&\log\frac{|\Sigma_{ij}|}{|\Omega_{ij}|} - \mathrm{Tr}\Big\{\Omega_{ij}^{-1}\Omega_{ij}\Big\} \\ &+ \mathrm{Tr}\Big\{\Sigma_{ij}^{-1}\langle(\boldsymbol{r}_i-\boldsymbol{r}_j-\boldsymbol{d}_{ij})(\boldsymbol{r}_i-\boldsymbol{r}_j-\boldsymbol{d}_{ij})^{\mathrm{T}}\rangle\Big\}\Big), \end{aligned} \tag{35}$$

where $\langle\cdot\rangle$ denotes expectation with respect to $Q_1 \triangleq Q(\boldsymbol{r}_i|z_{ij}{=}1)\,Q(\boldsymbol{r}_j|z_{jp}{=}1)Q(z_{jp}{=}1)$. Let us now consider the expectation in the last term:

$$\begin{aligned} &\langle(\boldsymbol{r}_i-\boldsymbol{r}_j-\boldsymbol{d}_{ij})(\boldsymbol{r}_i-\boldsymbol{r}_j-\boldsymbol{d}_{ij})^{\mathrm{T}}\rangle_{Q_1} = \\ &= \langle(\boldsymbol{r}_i-\boldsymbol{\mu}_{ij}+\boldsymbol{\mu}_{ij}-\boldsymbol{r}_j-\boldsymbol{d}_{ij})(\boldsymbol{r}_i-\boldsymbol{\mu}_{ij}+\boldsymbol{\mu}_{ij}-\boldsymbol{r}_j-\boldsymbol{d}_{ij})^{\mathrm{T}}\rangle_{Q_1} = \\ &= \Omega_{ij}+2\langle(\boldsymbol{r}_i-\boldsymbol{\mu}_{ij})(-\boldsymbol{r}_j-\boldsymbol{d}_{ij}+\boldsymbol{\mu}_{ij})^{\mathrm{T}}\rangle_{Q_1} \\ &\quad+\langle(\boldsymbol{r}_j+\boldsymbol{d}_{ij}-\boldsymbol{\mu}_{ij})(\boldsymbol{r}_j+\boldsymbol{d}_{ij}-\boldsymbol{\mu}_{ij})^{\mathrm{T}}\rangle_{Q_1} = \\ &= \Omega_{ij}+2\langle(\boldsymbol{r}_i-\boldsymbol{\mu}_{ij})(\pm\boldsymbol{\mu}_{jp}-\boldsymbol{r}_j-\boldsymbol{d}_{ij}+\boldsymbol{\mu}_{ij})^{\mathrm{T}}\rangle_{Q_1} \\ &\quad+\langle(\pm\boldsymbol{\mu}_{jp}+\boldsymbol{r}_j+\boldsymbol{d}_{ij}-\boldsymbol{\mu}_{ij})(\pm\boldsymbol{\mu}_{jp}+\boldsymbol{r}_j+\boldsymbol{d}_{ij}-\boldsymbol{\mu}_{ij})^{\mathrm{T}}\rangle_{Q_1} = \\ &= \Omega_{ij}+2\sum_{p\in V'}\xi_{jp}\langle(\boldsymbol{r}_i-\boldsymbol{\mu}_{ij})(\boldsymbol{\mu}_{jp}-\boldsymbol{r}_j)^{\mathrm{T}}\rangle_{Q(\boldsymbol{r}_i,\boldsymbol{r}_j|z_{ij},z_{jp})} \\ &\quad+\sum_{p\in V'}\xi_{jp}\langle(\boldsymbol{r}_j-\boldsymbol{\mu}_{jp})(\boldsymbol{r}_j-\boldsymbol{\mu}_{jp})^{\mathrm{T}}\rangle_{Q(\boldsymbol{r}_j|z_{jp}=1)} \\ &\quad+\sum_{p\in V'}\xi_{jp}(\boldsymbol{\mu}_{ij}-\boldsymbol{\mu}_{jp}-\boldsymbol{d}_{ij})(\boldsymbol{\mu}_{ij}-\boldsymbol{\mu}_{jp}-\boldsymbol{d}_{ij})^{\mathrm{T}} = \\ &= \Omega_{ij}+\sum_{p\in V'}\xi_{jp}\big(2\Psi_{ijp}+\Omega_{jp}+\mathcal{M}_{ijp}\big), \end{aligned} \tag{36}$$

where the definitions of auxiliary matrices Ψ_{ijp} and $\mathcal{M}_{ijp}$ are given in the second to the last derivation step above, and *i-j-p* is a child-parent-grandparent triad. It follows from (35) and (36) that

$$\begin{aligned} L_R = \frac{1}{2}\sum_{i,j\in V'}\xi_{ij}\Big(&\log\frac{|\Sigma_{ij}|}{|\Omega_{ij}|} - 2 + \mathrm{Tr}\{\Sigma_{ij}^{-1}\Omega_{ij}\} \\ &+\sum_{p\in V'}\xi_{jp}\mathrm{Tr}\{\Sigma_{ij}^{-1}(2\Psi_{ijp}+\Omega_{jp}+\mathcal{M}_{ijp})\}\Big). \end{aligned} \tag{37}$$

In (37), the last expression left to compute is $\mathrm{Tr}\{\Sigma_{ij}^{-1}\Psi_{ijp}\}$. For this purpose, we apply the Cauchy-Schwartz inequality as follows:

$$\begin{aligned} \mathrm{Tr}\{\Sigma_{ij}^{-1}\Psi_{ijp}\} &= \mathrm{Tr}\{\Sigma_{ij}^{-\frac{1}{2}}\Sigma_{ij}^{-\frac{1}{2}}\langle(\boldsymbol{r}_i-\boldsymbol{\mu}_{ij})(\boldsymbol{\mu}_{jp}-\boldsymbol{r}_j)^{\mathrm{T}}\rangle\}, \\ &= \mathrm{Tr}\{\langle\Sigma_{ij}^{-\frac{1}{2}}(\boldsymbol{r}_i-\boldsymbol{\mu}_{ij})(\boldsymbol{\mu}_{jp}-\boldsymbol{r}_j)^{\mathrm{T}}\Sigma_{ij}^{-\frac{1}{2}}\rangle\}, \\ &\le \mathrm{Tr}\{\Sigma_{ij}^{-1}\Omega_{ij}\}^{\frac{1}{2}}\mathrm{Tr}\{\Sigma_{ij}^{-1}\Omega_{jp}\}^{\frac{1}{2}}, \end{aligned} \tag{38}$$

where we used the fact that the Σs and Ωs are diagonal matrices. Although the Cauchy-Schwartz inequality, in general, does not yield a tight upper bound, in our case it appears reasonable to assume that variables $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$ (i.e., positions of object parts at different scales) are uncorrelated. Substituting (38) into (37), we finally derive the upper bound for L_R as

$$\begin{aligned} L_R \le \frac{1}{2}\sum_{i,j\in V'}\xi_{ij}\Big(&\log\frac{|\Sigma_{ij}|}{|\Omega_{ij}|} - 2 + \mathrm{Tr}\{\Sigma_{ij}^{-1}\Omega_{ij}\} \\ &+\sum_{p\in V'}\xi_{jp}\mathrm{Tr}\{\Sigma_{ij}^{-1}\big(\Omega_{jp}+\mathcal{M}_{ijp}\big)\} \\ &+2\sum_{p\in V'}\xi_{jp}\mathrm{Tr}\{\Sigma_{ij}^{-1}\Omega_{ij}\}^{\frac{1}{2}}\mathrm{Tr}\{\Sigma_{ij}^{-1}\Omega_{jp}\}^{\frac{1}{2}}\Big). \end{aligned} \tag{39}$$

A.3 Optimization of $Q(X|Z)$

$Q(X|Z)$ is fully characterized by parameters Q_{ij}^{kl}. From the definition of L_X, we have $\partial J(Q,P)/\partial Q_{ij}^{kl} = \partial L_X/\partial Q_{ij}^{kl}$. Due to parent-child dependencies in (32), it is necessary to iteratively differentiate L_X with respect to Q_{ij}^{kl} down the subtree of node i. For this purpose, we introduce three

auxiliary terms F_{ij}, G_i, and λ_i^k, which facilitate computation of $\partial L_X/\partial Q_{ij}^{kl}$, as shown below:

$$\begin{aligned} F_{ij} &\triangleq \sum_{k,l\in M} \xi_{ij} Q_{ij}^{kl} m_j^l \log[Q_{ij}^{kl}/P_{ij}^{kl}], \\ G_i &\triangleq \sum_{d,c\in d(i)} F_{dc} - \left\{ \sum_{k\in M} m_i^k \log P(\boldsymbol{y}_{\boldsymbol{\rho}(i)} | x_i^k, \boldsymbol{\rho}(i)) \right\}_{V^0}, \\ \lambda_i^k &\triangleq \exp(-\partial G_i/\partial m_i^k), \\ \Rightarrow \frac{\partial L_X}{\partial Q_{ij}^{kl}} &= \frac{\partial F_{ij}}{\partial Q_{ij}^{kl}} + \frac{\partial G_i}{\partial m_i^k}\frac{\partial m_i^k}{\partial Q_{ij}^{kl}}, \end{aligned} \tag{40}$$

where $\{\cdot\}_{V^0}$ denotes that the term is included in the expression for G_i if i is a leaf node for DT_{V^0}. For DT_V, the term in braces $\{\cdot\}$ is always included. This allows us to derive update equations for both models simultaneously. After finding the derivatives $\partial F_{ij}/\partial Q_{ij}^{kl} = \xi_{ij} m_j^l (\log[Q_{ij}^{kl}/P_{ij}^{kl}]+1)$ and $\partial m_i^k/\partial Q_{ij}^{kl} = \xi_{ij} m_j^l$, we arrive at

$$\partial L_X/\partial Q_{ij}^{kl} = \xi_{ij} m_j^l (\log[Q_{ij}^{kl}/P_{ij}^{kl}] + 1 - \log \lambda_i^k). \tag{41}$$

Finally, optimizing (41) with the Lagrange multiplier that accounts for the constraint $\sum_{k\in M} Q_{ij}^{kl} = 1$ yields the desired update equation: $Q_{ij}^{kl} = \kappa P_{ij}^{kl} \lambda_i^k$, introduced in (14).

To compute $\lambda_i^k = \exp(-\partial G_i/\partial m_i^k)$, we first find

$$\begin{aligned} \frac{\partial G_i}{\partial m_i^k} &= \sum_{c\in c(i)} \left(\frac{\partial F_{ci}}{\partial m_i^k} + \sum_{a\in M} \frac{\partial G_c}{\partial m_c^a}\frac{\partial m_c^a}{\partial m_i^k} \right) \\ &\quad - \{\log P(\boldsymbol{y}_{\boldsymbol{\rho}(i)} | x_i^k, \boldsymbol{\rho}(i))\}_{V^0}, \\ &= \sum_{c\in c(i)} \sum_{a\in M} \xi_{ci} Q_{ci}^{ak} \left(\log \frac{Q_{ci}^{ak}}{P_{ci}^{ak}} + \frac{\partial G_c}{\partial m_c^a} \right) \\ &\quad - \{\log P(\boldsymbol{y}_{\boldsymbol{\rho}(i)} | x_i^k, \boldsymbol{\rho}(i))\}_{V^0}, \end{aligned} \tag{42}$$

and then substitute Q_{ij}^{kl}, given by (14), into (42), which gives (15).

A.4 Optimization of $Q(R'|Z)$

$Q(R'|Z)$ is fully characterized by parameters $\boldsymbol{\mu}_{ij}$ and Ω_{ij}. From the definition of L_R, we observe that $\partial J(Q)/\partial \Omega_{ij} = \partial L_R/\partial \Omega_{ij}$ and $\partial J(Q)/\partial \boldsymbol{\mu}_{ij} = \partial L_R/\partial \boldsymbol{\mu}_i$. Since the Ωs are positive definite, from (39), it follows that

$$\begin{aligned} \frac{\partial L_R}{\partial \Omega_{ij}} = \frac{1}{2}\xi_{ij} \Bigg(&-\mathrm{Tr}\{\Omega_{ij}^{-1}\} + \mathrm{Tr}\{\Sigma_{ij}^{-1}\} + \sum_{c\in V'} \xi_{ci} \mathrm{Tr}\{\Sigma_{ci}^{-1}\} \\ &+ \sum_{p\in V'} \xi_{jp} \mathrm{Tr}\{\Sigma_{ij}^{-1}\} \mathrm{Tr}\{\Sigma_{ij}^{-1}\Omega_{ij}\}^{-\frac{1}{2}} \mathrm{Tr}\{\Sigma_{ij}^{-1}\Omega_{jp}\}^{\frac{1}{2}} \\ &+ \sum_{c\in V\prime} \xi_{ci} \mathrm{Tr}\{\Sigma_{ci}^{-1}\} \mathrm{Tr}\{\Sigma_{ci}^{-1}\Omega_{ij}\}^{-\frac{1}{2}} \mathrm{Tr}\{\Sigma_{ci}^{-1}\Omega_{ci}\}^{\frac{1}{2}} \Bigg). \end{aligned}$$

From $\partial L_R/\partial \Omega_{ij} = 0$, it is straightforward to derive the update equation for Ω_{ij} given by (18).

Next, to optimize the $\boldsymbol{\mu}_{ij}$ parameters, from (36) and (39), we compute

$$\begin{aligned} \frac{\partial L_R}{\partial \boldsymbol{\mu}_{ij}} = \sum_{c,p\in V'} \Big[&\xi_{ij}\xi_{jp} \Sigma_{ij}^{-1} (\boldsymbol{\mu}_{ij} - \boldsymbol{\mu}_{jp} - \boldsymbol{d}_{jp}) \\ &- \xi_{ci}\xi_{ij} \Sigma_{ci}^{-1} (\boldsymbol{\mu}_{ci} - \boldsymbol{\mu}_{ij} - \boldsymbol{d}_{ij}) \Big]. \end{aligned} \tag{43}$$

Then, from $\partial L_R/\partial \boldsymbol{\mu}_{ij} = 0$, it is straightforward to compute the update equation for $\boldsymbol{\mu}_{ij}$ given by (17).

A.5 Optimization of $Q(Z)$

$Q(Z)$ is fully characterized by the parameters ξ_{ij}. From the definitions of L_Z, L_X, and L_R we see that $\partial J(Q)/\partial \xi_{ij} = \partial (L_X + L_R + L_Z)/\partial \xi_{ij}$. Similar to the optimization of Q_{ij}^{kl}, we need to iteratively differentiate L_X as follows:

$$\frac{\partial L_X}{\partial \xi_{ij}} = \frac{\partial F_{ij}}{\partial \xi_{ij}} + \sum_{k\in M} \frac{\partial G_i}{\partial m_i^k}\frac{\partial m_i^k}{\partial \xi_{ij}}, \tag{44}$$

where F_{ij} and G_i are defined as in (40). By substituting the derivatives $\partial G_i/\partial m_i^k = -\log \lambda_i^k$, and $\partial F_{ij}/\partial \xi_{ij} = \sum_{k,l\in M} Q_{ij}^{kl} m_j^l \log[Q_{ij}^{kl}/P_{ij}^{kl}]$, and $\partial m_i^k/\partial \xi_{ij} = \sum_{l\in M} Q_{ij}^{kl} m_j^l$ into (44), we obtain

$$\begin{aligned} \frac{\partial L_X}{\partial \xi_{ij}} &= \sum_{k,l\in M} Q_{ij}^{kl} m_j^l \left(\log \frac{Q_{ij}^{kl}}{P_{ij}^{kl}} - \log \lambda_i^k \right), \\ &= -\sum_{k,l\in M} Q_{ij}^{kl} m_j^l \log \left(\sum_{a\in M} P_{ij}^{al} \lambda_i^a \right), \\ &= -A_{ij}. \end{aligned} \tag{45}$$

Next, we differentiate L_R, given by (39), with respect to ξ_{ij} as

$$\begin{aligned} \frac{\partial L_R}{\partial \xi_{ij}} = &\frac{1}{2}\log\frac{|\Sigma_{ij}|}{|\Omega_{ij}|} - 1 + \frac{1}{2}\mathrm{Tr}\{\Sigma_{ij}^{-1}\Omega_{ij}\} \\ &+ \frac{1}{2}\sum_{p\in V'} \xi_{jp} \Big(\mathrm{Tr}\{\Sigma_{ij}^{-1}(\Omega_{jp} + \mathcal{M}_{ijp})\} \\ &+ 2\mathrm{Tr}\{\Sigma_{ij}^{-1}\Omega_{ij}\}^{\frac{1}{2}} \mathrm{Tr}\{\Sigma_{ij}^{-1}\Omega_{tu}\}^{\frac{1}{2}} \Big) \\ &+ \frac{1}{2}\sum_{c\in V'} \xi_{ci} \big(\mathrm{Tr}\{\Sigma_{ci}^{-1}\big(\Omega_{ij} + \mathcal{M}_{cij}\big)\} \\ &+ 2\mathrm{Tr}\{\Sigma_{ci}^{-1}\Omega_{ci}\}^{\frac{1}{2}} \mathrm{Tr}\{\Sigma_{ci}^{-1}\Omega_{ij}\}^{\frac{1}{2}} \Big), \\ = &\, B_{ij} - 1, \end{aligned} \tag{46}$$

where indexes c, j, and p denote children, parents, and grandparents of node i, respectively. Further, from (33), we get

$$\partial L_Z/\partial \xi_{ij} = 1 + \log \xi_{ij}/\gamma_{ij}. \tag{47}$$

Finally, substituting (45), (46), and (47) into $\partial J(Q)/\partial \xi_{ij} = 0$ and adding the Lagrange multiplier to account for the constraint $\sum_{j\in V'} \xi_{ij} = 1$, we solve for the update equation of ξ_{ij} given by (19).

ACKNOWLEDGMENTS

This work was supported in part by grants from NASA Langley, the US Air Force Office of Sponsored Research, the US Air Force, and the US Special Operations Command. The authors would like to thank anonymous reviewers whose thoughtful comments and suggestions improved the quality of the paper.

REFERENCES

[1] N.J. Adams, A.J. Storkey, Z. Ghahramani, and C.K. I. Williams, "MFDTs: Mean Field Dynamic Trees," *Proc. 15th Int'l Conf. Pattern Recognition,* vol. 3, pp. 147-150, 2002.

[2] N.J. Adams, "Dynamic Trees: A Hierarchical Probabilistic Approach to Image Modeling," PhD dissertation, Division of Informatics, Univ. of Edinburgh, Edinburgh, U.K., 2001.

[3] A.J. Storkey, "Dynamic Trees: A Structured Variational Method Giving Efficient Propagation Rules," *Proc. 16th Conf. Uncertainty in Artificial Intelligence,* pp. 566-573, 2000.

[4] A.J. Storkey and C.K.I. Williams, "Image Modeling with Position-Encoding Dynamic Trees," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 7, pp. 859-871, July 2003.

[5] X. Feng, C.K.I. Williams, and S.N. Felderhof, "Combining Belief Networks and Neural Networks for Scene Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 4, pp. 467-483, Apr. 2002.

[6] Z. Ying and D. Castanon, "Partially Occluded Object Recognition Using Statistical Models," *Int'l J. Computer Vision,* vol. 49, no. 1, pp. 57-78, 2002.

[7] H. Schneiderman and T. Kanade, "Object Detection Using the Statistics of Parts," *Int'l J. Computer Vision,* vol. 56, no. 3, pp. 151-177, 2004.

[8] B. Moghaddam, "Principal Manifolds and Probabilistic Subspaces for Visual Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 6, pp. 780-788, June 2002.

[9] G. Hermosillo, C. Chefd'Hotel, and O. Faugeras, "Variational Methods for Multimodal Image Matching," *Int'l J. Computer Vision,* vol. 50, no. 3, pp. 329-343, 2002.

[10] H. Greenspan, J. Goldberger, and L. Ridel, "A Continuous Probabilistic Framework for Image Matching," *Computer Vision and Image Understanding,* vol. 84, no. 3, pp. 384-406, 2001.

[11] D. DeMenthon, D. Doermann, and M.V. Stuckelberg, "Image Distance Using Hidden Markov Models," *Proc. 15th Int'l Conf. Pattern Recognition,* vol. 3, pp. 143-146, 2000.

[12] B.H. Juang and L.R. Rabiner, "A Probabilistic Distance Measure for Hidden Markov Models," *AT&T Technical J.,* vol. 64, no. 2, pp. 391-408, 1985.

[13] M.C. Nechyba and Y. Xu, "Stochastic Similarity for Validating Human Control Strategy Models," *IEEE Trans. Robotic Automation,* vol. 14, no. 3, pp. 437-451, 1998.

[14] H. Cheng and C.A. Bouman, "Multiscale Bayesian Segmentation Using a Trainable Context Model," *IEEE Trans. Image Processing,* vol. 10, no. 4, pp. 511-525, 2001.

[15] H. Choi and R.G. Baraniuk, "Multiscale Image Segmentation Using Wavelet-Domain Hidden Markov Models," *IEEE Trans. Image Processing,* vol. 10, no. 9, pp. 1309-1321, 2001.

[16] J.-M. Laferté, P. Pérez, and F. Heitz, "Discrete Markov Image Modeling and Inference on the Quadtree," *IEEE Trans. Image Processing,* vol. 9, no. 3, pp. 390-404, 2000.

[17] M.I. Jordan, Z. Ghahramani, T.S. Jaakkola, and L.K. Saul, "An Introduction to Variational Methods for Graphical Models," *Machine Learning,* vol. 37, no. 2, pp. 183-233, 1999.

[18] T.S. Jaakkola, "Tutorial on Variational Approximation Methods," *Advanced Mean Field Methods,* M. Opper and D. Saad, eds., pp. 129-161, Cambridge, Mass.: MIT Press, 2000.

[19] B.J. Frey and N. Jojic, "Advances in Algorithms for Inference and Learning in Complex Probability Models for Vision," *IEEE Trans. Pattern Analysis and Machine Intelligence,* 2004.

[20] M.K. Schneider, P.W. Fieguth, W.C. Karl, and A.S. Willsky, "Multiscale Methods for the Segmentation and Reconstruction of Signals and Images," *IEEE Trans. Image Processing,* vol. 9, no. 3, pp. 456-468, 2000.

[21] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference,* chapter 4. pp. 143-236, San Mateo, Calif.: Morgan Kaufamnn, 1988.

[22] W.W. Irving, P.W. Fieguth, and A.S. Willsky, "An Overlapping Tree Approach to Multiscale Stochastic Modeling and Estimation," *IEEE Trans. Image Processing,* vol. 6, no. 11, pp. 1517-1529, 1997.

[23] J. Li, R.M. Gray, and R.A. Olshen, "Multiresolution Image Classification by Hierarchical Modeling with Two-Dimensional Hidden Markov Models," *IEEE Trans. Information Theory,* vol. 46, no. 5, pp. 1826-1841, 2000.

[24] W.K. Konen, T. Maurer, and C. von der Malsburg, "A Fast Dynamic Link Matching Algorithm for Invariant Pattern Recognition," *Neural Networks,* vol. 7, no. 6-7, pp. 1019-1030, 1994.

[25] A. Montanvert, P. Meer, and A. Rosenfield, "Hierarchical Image Analysis Using Irregular Tessellations," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, no. 4, pp. 307-316, Apr. 1991.

[26] M. Aitkin and D.B. Rubin, "Estimation and Hypothesis Testing in Finite Mixture Models," *J. Royal Statisitcal Soc. B,* vol. 47, no. 1, pp. 67-75, 1985.

[27] D.J.C. MacKay, "Introduction to Monte Carlo Methods," *Learning in Graphical Models (Adaptive Computation and Machine Learning),* M.I. Jordan, ed., pp. 175-204, Cambridge, Mass.: MIT Press, 1999.

[28] R.M. Neal, "Probabilistic Inference Using Markov Chain Monte Carlo Methods," Technical Report CRG-TR-93-1, Connectionist Research Group, Univ. of Toronto, 1993.

[29] D.J.C. MacKay, *Information Theory, Inference, and Learning Algorithms,* chapter 29, pp. 357-386, Cambridge, U.K.: Cambridge Univ. Press, 2003.

[30] T.M. Cover and J.A. Thomas, *Elements of Information Theory.* New York: Wiley Interscience Press, 1991.

[31] M. Mignotte, C. Collet, P. Perez, and P. Bouthemy, "Sonar Image Segmentation Using an Unsupervised Hierarchical MRF Model," *IEEE Trans. Image Processing,* vol. 9, no. 7, pp. 1216-1231, 2000.

[32] C. D'Elia, G. Poggi, and G. Scarpa, "A Tree-Structured Markov Random Field Model for Bayesian Image Segmentation," *IEEE Trans. Image Processing,* vol. 12, no. 10, pp. 1259-1273, 2003.

[33] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A Database of Human Segmented Natural Images and Its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics," *Proc. Eighth Int'l Conf. Computer Vision,* vol. 2, pp. 416-423, 2001.

[34] N.G. Kingsbury, "Complex Wavelets for Shift Invariant Analysis and Filtering of Signals," *J. Applied Computer Harmonic Analysis,* vol. 10, no. 3, pp. 234-253, 2001.

[35] T. Lindeberg, "Scale-Space Theory: A Basic Tool for Analysing Structures at Different Scales," *J. Applied Statistics,* vol. 21, no. 2, pp. 224-270, 1994.

[36] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision,* vol. 60, no. 2, pp. 91-110, 2004.

Sinisa Todorovic received the BS degree in electrical engineering from the University of Belgrade, Serbia, in 1994. He received the MS and PhD degrees at the University of Florida, in 2002, and 2005, respectively. From 1994 to 2001, he worked as a software engineer in the communications industry. In 2001, he enrolled in the electrical and computer engineering graduate program at the University of Florida, Gainesville. There, as a member of the Center for Micro Air Vehicle Research, he conducted research aimed at enabling intelligent mission profiles for small aircraft. His primary research interests encompass statistical image modeling, machine learning, and multiresolution signal processing. He is a student member of the IEEE.

Michael C. Nechyba received the BS degree in electrical engineering from the University of Florida in 1992, and the PhD degree in robotics from Carnegie Mellon University in 1998. Upon completion of his thesis, he joined the Department of Electrical and Computer Engineering at the University of Florida as assistant professor in August 1998. There, he served as an associate director of the Machine Intelligence Laboratory, conducting research in two primary areas: 1) vision-based and sensor-based autonomy for Micro Air Vehicles (MAVs) and 2) direct brain-machine interfaces (BMIs). In October 2004, he resigned his position at the University of Florida to cofound Pittsburgh Pattern Recognition. Pittsburgh Pattern Recognition seeks to commercialize face/object recognition technology for various applications in digital photography, surveillance, and homeland security, and is currently funded through the US intelligence community. He has published approximately 30 journal and refereed conference papers. He is a member of the IEEE and the IEEE Computer Society.