Hough Forest Random Field for Object Recognition and Segmentation

Nadia Payet, *Student Member*, *IEEE*, and Sinisa Todorovic, *Member*, *IEEE*

Abstract—This paper presents a new computational framework for detecting and segmenting object occurrences in images. We combine Hough forest (HF) and conditional random field (CRF) into HFRF to assign labels of object classes to image regions. HF captures intrinsic and contextual properties of objects. CRF then fuses the labeling hypotheses generated by HF for identifying every object occurrence. Interaction between HF and CRF happens in HFRF inference, which uses the Metropolis-Hastings algorithm. The Metropolis-Hastings reversible jumps depend on two ratios of proposal and posterior distributions. Instead of estimating four distributions, we directly compute the two ratios using HF. In leaf nodes, HF records class histograms of training examples and information about their configurations. This evidence is used in inference for nonparametric estimation of the two distribution ratios. Our empirical evaluation on benchmark datasets demonstrates higher average precision rates of object detection, smaller object segmentation error, and faster convergence rates of our inference, relative to the state of the art. The paper also presents theoretical error bounds of HF and HFRF applied to a two-class object detection and segmentation.

Index Terms—Object recognition and segmentation, conditional random field, Hough forest, Metropolis-Hastings algorithm

1 Introduction

This paper presents a new computational framework, called Hough forest random field (HFRF). HFRF provides a principled way to jointly reason about multiple, statistically dependent random variables and their attributes. We derive theoretical performance bounds of HFRF, and demonstrate its utility on a challenging task of conjoint object recognition and segmentation.

Identifying subimage ownership among occurrences of distinct object classes in an image is one of the fundamental problems in computer vision [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11]. Our approach builds on the following common recognition strategies: 1) Objects of interest in the scene occupy image regions, characterized by class-specific appearance, shape, and spatial-layout properties; 2) neighboring image parts are likely to be correlated since they may be occupied by the same object or distinct objects that typically co-occur (e.g., cars and road); and 3) recognized objects provide contextual cues for identifying other objects, their scale, and spatial configuration in the scene. We formalize steps 1-3 by using image regions as basic features and assigning labels of object classes to these regions in the MAP inference of a graphical model. The model is aimed at capturing statistical intrinsic and contextual properties of target object classes in terms of region appearance, shape, spatial-layout, and co-occurrence properties.

Specifically, we use Conditional Random Field (CRF) [12]—one of the most popular graphical models for object recognition and segmentation [2], [3], [4], [5], [6], [7], [8]. A CRF defines a posterior distribution of hidden random variables $\boldsymbol{Y}$ (e.g., object-class labels), given observed image features $\boldsymbol{X}$, in a factored form: $p(\boldsymbol{Y}|\boldsymbol{X};\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})}\exp(-\sum_c \psi_c(\boldsymbol{Y}_c, \boldsymbol{X}; \boldsymbol{\theta}))$. Observable image features, $\boldsymbol{X}$, could be pixels [2], [3], image patches [4], [5], or image regions [6], [7], [8]. Each potential function, ψ_c, accounts for statistical dependencies of a subset of hidden random variables, $\boldsymbol{Y}_c \subseteq \boldsymbol{Y}$, conditioned on observables, $\boldsymbol{X}$, and is parameterized by model parameters $\boldsymbol{\theta}$. The potentials are often defined as linear functions of parameters, $\psi_c(\boldsymbol{Y}_c, \boldsymbol{X}; \boldsymbol{\theta}) = \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\Psi}_c$, where $\boldsymbol{\Psi}_c$ is a descriptor vector associated with image features X [2], [3], [4]. Learning $\boldsymbol{\theta}$ is hard because computation of the partition function $Z(\boldsymbol{\theta})$ is intractable for most graphs (except for chains and trees). Inference of this model amounts to an energy minimization problem, and is typically posed as the joint MAP assignment that minimizes the energy $\sum_c \psi_c(\boldsymbol{Y}_c, \boldsymbol{X}; \boldsymbol{\theta})$. Such an inference is also intractable for general graphs. This requires considering approximate algorithms, e.g., graph cut and loopy belief propagation (LBP). The effect of these approximations on the original semantics of CRF is poorly understood.

We address intractable CRF inference with the Metropolis-Hastings (MH) algorithm [13]. MH draws samples $\boldsymbol{Y}^{(t)}$ from the CRF's posterior, $p(\boldsymbol{Y}|\boldsymbol{X})$, and thus generates a Markov chain in which state $\boldsymbol{Y}^{(t+1)}$ depends only on the previous state $\boldsymbol{Y}^{(t)}$. The jumps between the states are reversible, and governed by a proposal distribution $q(\boldsymbol{Y}^{(t)} \rightarrow \boldsymbol{Y}^{(t+1)})$. The proposal is accepted if the acceptance rate, α, drawn from the uniform distribution, $U(0,1)$, satisfies $\alpha<\min\{1, \frac{q(\boldsymbol{Y}^{(t+1)}\rightarrow\boldsymbol{Y}^{(t)})}{q(\boldsymbol{Y}^{(t)}\rightarrow\boldsymbol{Y}^{(t+1)})}\frac{p(\boldsymbol{Y}^{(t+1)}|\boldsymbol{X})}{p(\boldsymbol{Y}^{(t)}|\boldsymbol{X})}\}$. As can be seen, MH is regulated by two ratios of the proposal and posterior

- *The authors are with the School of Electrical Engineering and Computer Science, Oregon State University, 1148 Kelley Engineering Center, Corvallis, OR 97331.*
  E-mail: payetn@onid.orst.edu, sinisa@eecs.oregonstate.edu.

Manuscript received 31 May 2011; revised 30 Mar. 2012; accepted 22 Aug. 2012; published online 5 Sept. 2012.
Recommended for acceptance by P. Felzenszwalb.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2011-05-0346.
Digital Object Identifier no. 10.1109/TPAMI.2012.194.

distributions. Note that the partition function Z cancels out in these ratios. Consequently, MH does not require computation of Z, thereby addressing the key bottleneck of CRF learning and inference.

However, convergence of MH typically requires exponential time (except for chains models), which hinders the use of MH in many applications. The literature abounds with diverse approaches to improving the MH convergence rate. For example, the Swendsen-Wang algorithm has been successfully used in computer vision [14]. This and related approaches are based on proposing more efficient reversible jumps across the states. But they still require estimation of the proposal and posterior distributions in each proposed state for computing the acceptance rate α.

In this paper, we improve the convergence rate of MH by *directly estimating the two ratios* of the proposal and posterior distributions instead of computing each individual distribution. Related work usually estimates the four distributions: $q(\boldsymbol{Y}^{(t+1)}\rightarrow\boldsymbol{Y}^{(t)})$, $q(\boldsymbol{Y}^{(t)}\rightarrow\boldsymbol{Y}^{(t+1)})$, $p(\boldsymbol{Y}^{(t+1)}|\boldsymbol{X})$, and $p(\boldsymbol{Y}^{(t)}|\boldsymbol{X})$ [14]. In contrast, we directly estimate the two ratios,

$$qr^{(t)} = \frac{q\big(\boldsymbol{Y}^{(t+1)} \rightarrow \boldsymbol{Y}^{(t)}\big)}{q\big(\boldsymbol{Y}^{(t)} \rightarrow \boldsymbol{Y}^{(t+1)}\big)} \text{ and } pr^{(t)} = \frac{p\big(\boldsymbol{Y}^{(t+1)}|\boldsymbol{X}\big)}{p\big(\boldsymbol{Y}^{(t)}|\boldsymbol{X}\big)}, \qquad (1)$$

in a discriminative manner. The Hough forest (HF) [15] is used to estimate $qr^{(t)}$ and $pr^{(t)}$. The HF is trained such that the estimated ratios take high values only over a few states, while for the most of the state space they are close to zero. Consequently, the resulting αs are not sufficiently large for MH to visit most of the proposed states. This improves the convergence rate because MH does not waste computational resources to visit many states.

Given a training set of labeled image regions, HF grows many decision trees. We view the trees as a way of discriminatively structuring evidence about: 1) the class distributions in the training set, and 2) spatial relations of training image regions relative to manually annotated bounding boxes (BBs) of objects in the training images. In particular, image regions (i.e., their descriptors) are "dropped" down every decision tree of HF until they reach leaf nodes. Each leaf of each tree stores: 1) a histogram of the number of training image regions from each class that reached that leaf, and 2) the positions and scales of labeled object bounding boxes that overlap with image regions that reached that leaf. When a new image is encountered, its regions are also "dropped" down every decision tree in the forest until they reach leaf nodes. We use the class histograms and spatial layouts of object bounding boxes of training examples, stored in these leaves, for robust estimation of $qr^{(t)}$ and $pr^{(t)}$.

HF represents an integral part of CRF—i.e., HF is not used as a plug-in to compute the potential functions, $\psi_c(\boldsymbol{Y}_c, \boldsymbol{X}; \boldsymbol{\theta}) = \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\Psi}_c$, of a CRF. Instead, these two modeling paradigms jointly form a unified object representation termed HFRF.

Contributions:

- We combine HF and CRF into a new computational framework HFRF. HFRF is used for object recognition and segmentation in challenging images with occlusions and varying numbers and scales of object occurrences.
- Learning is efficiently conducted by HF which collects the class histograms and spatial information about object bounding boxes in training images. This training evidence is then used for estimation of $qr^{(t)}$ and $pr^{(t)}$, required by MH-based inference of HFRF.
- Our evaluation on benchmark datasets demonstrates higher recall and precision of object detection, smaller object segmentation error, and faster convergence rates of our inference relative to the state of the art.
- For a two-class object recognition problem, we derive theoretical error bounds of estimating $qr^{(t)}$ and $pr^{(t)}$, as well as theoretical performance bounds of HFRF.

Paper organization: Section 2 reviews prior work. Sections 3, 4, and 5 specify the CRF model, its MH-based inference, and HF-based learning. Section 7 presents our experimental evaluation. A theoretical analysis of HFRF is given in Section 8. Our concluding remarks are given in Section 9.

2 Literature Review

This section reviews prior work on random fields and random forests, and points out our contributions.

Random fields have been used with great success for object recognition and segmentation [2], [3], [16]. These problems can be cast as an image labeling problem, where the goal is to predict class labels, Y, of a given set of image features X (e.g., regions). Random fields factorize the joint distribution $p(\mathbf{X}, \mathbf{Y})$ or the posterior distribution $p(\mathbf{Y}|\mathbf{X})$ into a product of local interactions. Conditional random fields [2], [12] have become popular for their improved ability to capture the relationships between object-class labels and the image by conditioning subsets of hidden random variables of the model on the observables.

Inference of CRF is typically formulated as the energy minimization problem, $\boldsymbol{Y}^* = \arg\min_{\boldsymbol{Y}} \sum_c \psi_c(\boldsymbol{Y}_c, \boldsymbol{X}; \boldsymbol{\theta})$ [17], [18], [19], [20], [21]. In a special case, e.g., for trees [22], the energy is submodular, and thus its minimization can be solved efficiently using linear programming relaxation methods, e.g., belief propagation [23]. Graph-cut algorithms have also been demonstrated as efficient in finding global optima for submodular energies [24], [25]. In general, CRF inference is intractable, and requires *approximate* algorithms. Existing approximate algorithms can be broadly grouped as

1. graph-cuts methods [3], [26], [27],
2. message-passing algorithms based on cycle inequalities [28], [29], [30] or LBP [4], [23], [31],
3. variational methods [32],
4. dual decomposition algorithms [33], [34], [35], and
5. approaches based on the roof duality relaxation [25], [36].

Learning CRF parameters, $\boldsymbol{\theta}$, typically uses different algorithms from those for inference [27], [37], [38], [39]. Recently, the structural SVM (i.e., large margin) formulations of learning CRF parameters has shown great promise in object recognition [40].

In this work, we depart from the standard linearization of the potential functions, $\psi_c(\boldsymbol{Y}_c, \boldsymbol{X}; \boldsymbol{\theta}) = \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\Psi}_c$, and also unify CRF inference and learning within the same framework. The unification is based on HF. For the CRF inference, we use the MH algorithm, whose convergence rate is improved by directly computing the two ratios of the proposal and posterior distributions over states in the space of MAP solutions. Our contribution is that we avoid the usual commitment of prior work to estimating the linear potential functions in the nominator and denominator of the ratio

$$\frac{\prod_c \boldsymbol{\theta}^{\mathrm{T}} \boldsymbol{\Psi}_c^{(t+1)}}{\prod_c \boldsymbol{\theta}^{\mathrm{T}} \boldsymbol{\Psi}_c^{(t)}}.$$

Instead, we use HF for directly estimating the two ratios in a nonparametric manner.

Random Forests have been used for: 1) discriminative learning of visual codewords for the Bag-of-Words model of objects [41], [42], [43] and images [44], 2) segmenting and grouping all neighboring instances of the same class [45], and 3) segmenting every object occurrence [16]. Decision trees have also been employed for estimating pairwise potential functions of a CRF [46].

Our work is related to recent work on Hough forests for object detection and localization [15]. Leaf nodes of HF collect the information about the locations and sizes of object bounding boxes in training images. This information, however, is used to predict a spatial distribution of bounding boxes in a test image. We instead use this information to predict the two distribution *ratios*. Evidence trees are also used for image classification [47], but only as a first stage of a stacked-classifier architecture that replaces the standard majority voting of Random Forest.

In our initial work [48], we fuse Random Forest and CRF into a Random-Forest-Random-Field (RFRF) model for labeling every pixel in the image with an appropriate object class. RFRF, however, cannot distinguish between neighboring, distinct instances of the same object class, but simply clumps together all pixels with the same label. The work presented in this paper advances our initial approach by enabling segmentation of individual objects. Instead of Random Forest used in [48], we employ HF, which additionally captures the spatial layout properties of image regions occupied by target objects.

3 CRF Model

Our CRF is defined over a set of multiscale image regions. Regions are used as image features because they are dimensionally matched with 2D object occurrences in the image and thus facilitate modeling of various perceptual-organization and contextual cues (e.g., continuation, smoothness, containment, and adjacency) that are often used in recognition [6], [7], [8], [9], [10], [11]. Access to regions is provided by the state-of-the-art, multiscale segmentation algorithm of [49]. Since the right scale at which objects occur is unknown, we use all regions from all scales.

The extracted regions are organized in a graph, $G = (V, E, \boldsymbol{X})$, where V and E are sets of nodes and edges, and $\boldsymbol{X}$ denotes their descriptors. The nodes $i = 1, \ldots, |V|$ correspond to multiscale segments, and edges $(i, j) \in E$ capture their spatial relations. Each node i is characterized by a d-dimensional descriptor vector, $\boldsymbol{x}_i \in \mathbb{R}^d$, whose elements describe photometric and geometric properties of the corresponding image region, including color, shape, filter responses. The set of all region descriptors is $\boldsymbol{X} = \{\boldsymbol{x}_i : i = 1, \ldots, |V|\}$. In addition, each edge $(i, j) \in E$ is characterized by a descriptor indicating the spatial relationship type between i and j. With a slight abuse of notation, we denote this edge descriptor as $(\boldsymbol{x}_i, \boldsymbol{x}_j)$. A pair of regions can have one of the following relationships: 1) part of, 2) touching, and 3) far. Since the segmentation algorithm of [49] is strictly hierarchical, region i is a descendent of region j if i is fully embedded as subregion within ancestor j. Two regions i and j touch if they share a boundary part. Finally, if i and j are not in the hierarchical and touch relationships, then they are declared as far.

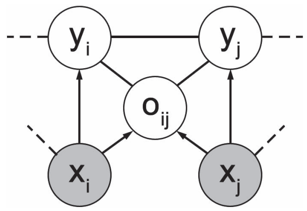


Fig. 1. Our CRF over image regions. Observable region descriptors $\boldsymbol{x}_i, \boldsymbol{x}_j$ and their hidden class labels y_i, y_j represent the nodes of CRF. Node o_{ij} represents the binary indicator if regions i and j belong to the same object. Every region pair is connected in the model (dashed lines), but we show only one pair, for clarity.

CRF is defined as a graphical model over G, and is illustrated in Fig. 1. Our CRF defines a posterior distribution of two types of hidden random variables, $\boldsymbol{M} = \{\boldsymbol{Y}, \boldsymbol{O}\}$, given observables $\boldsymbol{X}$. $\boldsymbol{Y} = \{y_i\}$ represents a set of hidden random variables associated with nodes of G, indicating the class label of the corresponding region, $y_i \in \{1, \ldots, K\}$, where K denotes the total number of object classes. Also, $\boldsymbol{O} = \{o_{ij}\}$ represents a set of binary hidden random variables associated with edges of G, indicating whether two regions belong to the same object. If $o_{ij} = 1$, then regions i and j are occupied by the same object; otherwise, $o_{ij} = 0$ means i and j belong to two distinct objects.

Let p_i and p_{ij} denote posterior distributions over nodes and pairs of nodes in CRF, defined as

$$p_i = p(y_i | \boldsymbol{x}_i), \tag{2}$$

$$p_{ij} = p(y_i, y_j | \boldsymbol{x}_i, \boldsymbol{x}_j) p(o_{ij} | y_i, y_j, \boldsymbol{x}_i, \boldsymbol{x}_j). \tag{3}$$

Then, the posterior of CRF is defined as

$$p(\boldsymbol{M}|G) = p(\boldsymbol{Y}, \boldsymbol{O}|G) \propto \prod_{i \in V} p_i \prod_{(i,j) \in E} p_{ij} \ , \tag{4}$$

where $\propto$ denotes proportionality to a normalizing constant.

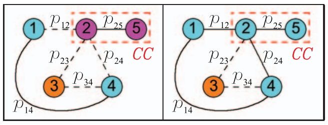
Object recognition and segmentation are formulated as multicoloring of CRF using the joint MAP assignment:

$$\boldsymbol{M}^* = (\boldsymbol{Y}^*, \boldsymbol{O}^*) = \arg\max_{\boldsymbol{Y}, \boldsymbol{O}} \ p(\boldsymbol{Y}, \boldsymbol{O}|G), \tag{5}$$

as further explained in the next section.

4 CRF Inference

For CRF inference, we use the Swendsen-Wang cut algorithm (SW-cut), presented in [14]. SW-cut iterates the



(a) State A (b) State B

$$q(CC|A)=1,$$
$$q(CC|B)=(1-p_{12}^B)(1-p_{24}^B),$$
$$\frac{p(\boldsymbol{M}=B|G)}{p(\boldsymbol{M}=A|G)}=\frac{p_2^B\,p_5^B}{p_2^A\,p_5^A}\cdot\frac{p_{12}^B p_{23}^B p_{24}^B}{p_{12}^A p_{23}^A p_{24}^A}$$

(c) Distibutions for this example

Fig. 2. One iteration of SW-cut illustrated on an example graph of image regions. The figure shows only y_i nodes of CRF, for clarity. (a) Initial state A represents the configuration of connected components, CCs, in the graph after edges have been probabilistically sampled (bold) or "cut" (dashed). SW-cut randomly selects $CC = \{2, 5\}$. (b) The coloring of all nodes in the CC is randomly changed, resulting in new state B. Now, the CC has the same label as nodes 1 and 4, which results in cutting edges $(1, 2)$ and $(2, 4)$. (c) The proposed and posterior distributions of states A and B. Best viewed in color.

Metropolis-Hastings reversible jumps. It iteratively searches for the MAP inference by jointly coloring groups of image regions and cutting their statistical dependencies in every iteration. Unlike [14], the edges in our graphical model are labeled, which helps guide the MH jumps toward the MAP solution by taking into account object instance labeling. This leads to an improved convergence rate of MH, as shown in our results. In this section, we briefly review SW-cut, and illustrate its iterations on a graph whose nodes represent object class labels y_i of CRF.

SW-cut iterates the following two steps: 1) *Graph Clustering* probabilistically samples edges in a graph of image regions based on $p(y_i, y_j|\boldsymbol{x}_i, \boldsymbol{x}_j)$ and probabilistically samples their labels $o_{ij} \in \{0, 1\}$ based on $p(o_{ij}|y_i, y_j, \boldsymbol{x}_i, \boldsymbol{x}_j)$. This forms connected components, CCs. A CC is a subset of neighboring nodes (i.e., image regions) with the same color connected by edges with $o_{ij} = 1$. That is, a CC represents an instance of one of the object classes. If edge (i, j) is not sampled, we say that it has been probabilistically "cut." The cut is a set of edges that would have linked CC to external nodes. 2) *Graph Relabeling* first randomly selects one of the CC's obtained in Graph Clustering, then flips the color of *all* nodes in that CC, randomly selecting one of the K object classes, and, finally, cuts the CC's edges to the rest of the graph nodes having that same color. A particular label assignment to both edges, in Graph Clustering, and nodes, in Graph Relabeling, jointly define one state in a space of inference solutions. In each iteration, SW-cut probabilistically decides whether to accept the new state or to keep the previous state. Unlike other MCMC methods that consider one node at a time (e.g., Gibbs sampler), SW-cut operates on a number of nodes and edges at once. Consequently, SW-cut converges faster and enables inference on relatively large graphs. Fig. 2 shows an illustrative example.

In the following, we define the acceptance rate of MH jumps. Let $q(A \to B)$ be the proposal probability for moving from state A to B, and let $q(B \to A)$ denote the converse. $q(A \to B)$ is defined as

$$q(A{\to}B) = q(CC|A)q(B(CC)|CC, A), \tag{6}$$

where $q(CC|A)$ denotes the probability of generating CC at state A, and $q(B(CC)|CC, A)$ denotes the probability of recoloring CC to state B when CC is obtained in state A. From (6), the acceptance rate, $\alpha(A{\to}B)$, of the MH move from A to B is defined as

$$\alpha(A \to B) = \min\left(1, \frac{q(B \to A)p(\boldsymbol{M} = B|G)}{q(A \to B)p(\boldsymbol{M} = A|G)}\right). \tag{7}$$

Note that complexity of computing (7) is relatively low. This is because computing the ratio $qr(A \to B) = \frac{q(B\to A)}{q(A\to B)}$ in (7) involves only those edges that are "cut" around CC in states A and B—not all edges. Also, computing the ratio $pr(A \to B) = \frac{p(\boldsymbol{M}=B|G)}{p(\boldsymbol{M}=A|G)}$ accounts only for recolored nodes in CC and their adjacent edges—not the entire graph G. All the other probabilities have not changed from state A to state B, so they will be canceled out in the ratio. From (6), the ratio $\frac{q(A(CC)|CC,B)}{q(B(CC)|CC,A)}$ in $qr(A \to B)$ can be canceled out because CCs are assigned colors with the uniform distribution. Thus, from (4), we have

$$qr(A \to B) = \frac{\prod_{(i,j)\in \mathrm{Cut}_B} (1-p_{ij}^B)}{\prod_{(i,j)\in \mathrm{Cut}_A} (1-p_{ij}^A)}, \tag{8}$$

$$pr(A \to B) = \prod_{i\in CC} \frac{p_i^B}{p_i^A} \prod_{j\in\mathcal{N}(i)} \frac{p_{ij}^B}{p_{ij}^A}, \tag{9}$$

where Cut_A and Cut_B denote the sets of "cut" edges in states A and B, and $\mathcal{N}(i)$ is the set of neighbors of node i, $\mathcal{N}(i) = \{j : j \in V, (i, j) \in E\}$. Fig. 2c shows an example of how to compute qr_{AB} and pr_{AB}.

With probability $\alpha(A \to B)$, the algorithm will move to state B. Otherwise, it remains in state A and tries to probabilistically select a different CC or to propose a different coloring scheme for the same CC.

As shown in [14], SW-cut is relatively insensitive to different initializations. In our experiments, we initialize the labels of all nodes and edges using the Bayesian decision on the corresponding posteriors p_i and p_{ij}. As mentioned in Section 1, the ratios in (8) and (9) are computed using HF, whose learning is explained in the following section.

5 Learning

This section presents our training setting, explains how to learn HF from a set of labeled image regions, and specifies three types of statistics stored in HF that are used for estimating the ratios, given by (8) and (9).

5.1 Training Setting

We assume that training images are manually labeled with bounding boxes around target object occurrences, and that the boxes are tagged with the appropriate object class label. Thus, our training dataset consists of m labeled regions from training images, $\{(\boldsymbol{x}_i, y_i, \boldsymbol{b}_i) : i = 1, \ldots, m\}$.

$\boldsymbol{x}_i$ is a d-dimensional descriptor, $\boldsymbol{x}_i \in \mathbb{R}^d$, encoding the photometric and geometric properties of region i.

y_i is the object class label associated with region i. If i falls within an object bounding box labeled with class $y \in \{1, 2, \ldots, K\}$, it receives label y, i.e., $y_i = y$. If i is covered by a number of object bounding boxes of different classes, then i is added to the training set multiple times to account for all distinct class labels it covers.

$\boldsymbol{b}_i$ represents the information about the object bounding box which covers region i. Specifically, $\boldsymbol{b}_i$ is a vector whose elements include: lengths of two sides of the bounding box (a_i, b_i), and the offset vector of i's centroid from the

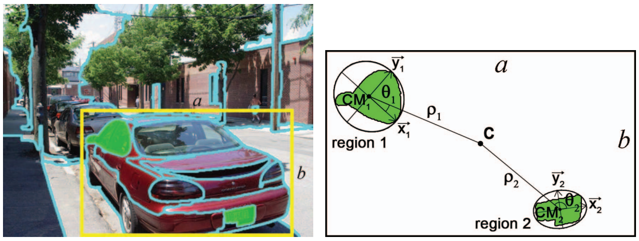


Fig. 3. To achieve scale and rotation invariance, the spatial information about the object bounding box that contains two image regions 1 and 2 is computed separately for each region with respect to their respective reference systems. The right figure presents the bounding box with sides a and b before the transformation to each region's reference coordinate system.

bounding-box center, $\rho_i e^{\theta\sqrt{-1}}$. For scale and rotation invariance, both the side lengths and the offset vector are computed with respect to i's reference system (and thus indexed by i) as follows: The best fitting ellipse of i is transformed to a unit circle, so the principal axes of the ellipse coincide with the x-axis and y-axis of the image. The estimated transformation is applied to the bounding box. In this reference system, we compute the offset and normalized size of the transformed bounding box, $\boldsymbol{b}_i = (\rho_i \cos\theta_i, \rho_i \sin\theta_i, a_i, b_i)$, as illustrated in Fig. 3. Note that if region i belongs to the background class, then there is no bounding box information, i.e., $\boldsymbol{b}_i$ is undefined.

5.2 Constructing the Hough Forest

The training dataset $\{(\boldsymbol{x}_i, y_i, \boldsymbol{b}_i) : i = 1, \dots, m\}$ is used to learn an ensemble of $T = 10$ decision trees representing HF. As is standard in the Random Forest and HF literature [15], [50], each tree $t = 1, \dots, T$ is constructed from a distinct, randomly selected subset of training examples (i.e., image regions in our case) to ensure variability across the trees. A tree is constructed recursively starting from the root. A node of the tree receives a set of training regions. If the node depth is maximum ($= 15$) or the number of received training regions is small (< 30), the node is declared as a leaf. Otherwise, the node is declared as a nonleaf, and an optimal discriminative pair $(\textit{coordinate in } \boldsymbol{x}, \textit{threshold})$ is chosen from a large pool of randomly generated tests to split the received training examples. These examples are then passed to the two newly created children nodes. The recursion stops at leaf nodes.

Below, we explain the criterion for splitting the training examples at a node. The key idea is to find the test which maximally reduces uncertainty in both the class labels and the size and location of bounding boxes of objects occurring in training images. We follow [15] and define two measures of uncertainty for a set of training regions $S = \{(\boldsymbol{x}_i, y_i, \boldsymbol{b}_i)\}$ that reached the node. The *class-label uncertainty* measures the impurity of the class labels, $U_1(S) = |S| \cdot H(\mathbf{Y})$, where $H(\mathbf{Y})$ is the class entropy. The *offset uncertainty* measures the impurity of the bounding-box vectors $\boldsymbol{b}_i$, $U_2(S) = \sum_{i:\exists \boldsymbol{b}_i} \|\boldsymbol{b}_i - \overline{\boldsymbol{b}_S}\|^2$, where $\overline{\boldsymbol{b}_S}$ is the mean vector over S. Note that $U_2(S)$ ignores background regions in S for which $\boldsymbol{b}_i$ is undefined.

For splitting S at tree node k, we generate a pool of tests $\{\tau^k\}$ by uniformly sampling a particular coordinate in all descriptors $\{\boldsymbol{x}_i\}$. The threshold value δ for each test is chosen uniformly at random in the range of values of the selected coordinate. Then, with equal probability, we pick the test τ^{k^*} that either minimizes $U_1(S)$ or $U_2(S)$, as $\tau^{k^*} = \arg\min_k (U_*(\{i|\tau^k(\boldsymbol{x}_i) < \delta\}) + U_*(\{i|\tau^k(\boldsymbol{x}_i) \geq \delta\}))$, where $* = 1$ or $* = 2$. By interleaving nodes that decrease $U_1(S)$ with nodes that decrease $U_2(S)$, the tree construction ensures that training regions which reach the leaves have low variations in both class labels and offsets from object bounding boxes in training images.

HF becomes equivalent to Random Forest when bounding-box annotations of objects in training images are not considered. That is, nodes in the decision trees of RF split training regions such that only $U_1(S)$ is minimized. In our initial work [48], we used RF instead of HF.

5.3 The Three Statistics

After HF is constructed, we compute three types of useful statistics from the training regions collected in every leaf node. The three statistics are used in CRF inference to estimate the two ratios qr and pr.

First, for every leaf node of HF, L, we compute one-class counts $\Phi_L = \{\phi_L(y) : y = 1, \dots, K\}$, where $\phi_L(y)$ is the number of training examples belonging to class y that reached L. Normalizing $\phi_L(y)$ over the total number of regions in L gives an estimate of the posterior $p(y_i|\boldsymbol{x}_i)$.

Second, for every pair of leaves, (L, L'), we compute two-class counts $\Psi_{LL'} = \{\psi_{LL'}(y, y', \boldsymbol{x}, \boldsymbol{x}')\}$, where $\psi_{LL'}(y, y', \boldsymbol{x}, \boldsymbol{x}')$ is the number of training example pairs belonging to classes y and y' that reached L and L', and simultaneously have the spatial relationship type $(\boldsymbol{x}, \boldsymbol{x}') \in$ {"part-of," "touching," "far"}. Normalizing $\psi_{LL'}(y, y', \boldsymbol{x}, \boldsymbol{x}')$ over the number of pairs of regions in L and L' that have relationship $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ would result in an estimate of the posterior $p(y_i, y_j|\boldsymbol{x}_i, \boldsymbol{x}_j)$.

Third, for every pair of leaves, (L, L'), we compute the total area overlap between bounding boxes $\boldsymbol{b}_i$ and $\boldsymbol{b}_{i'}$ associated with all regions i and i' in L and L', respectively, where the regions satisfy the following conditions: 1) They all have the same class label, $\forall (i, i') y = y_i = y_{i'}$, and 2) they all have the same type of spatial relationship, $\forall (i, i') (\boldsymbol{x}_i, \boldsymbol{x}_{i'}) = (\boldsymbol{x}, \boldsymbol{x}') \in$ {"part-of", "touching", "far"}. Let $\Xi_{LL'} = \{\xi_{LL'}(y, \boldsymbol{x}, \boldsymbol{x}')\}$ denote the set of area overlaps of bounding boxes satisfying conditions 1 and 2, where we compute

$$\xi_{LL'}(y, \boldsymbol{x}, \boldsymbol{x}') = \sum_{\substack{i \in L, i' \in L' \\ y = y_i = y_{i'} \\ (\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}_i, \boldsymbol{x}_{i'})}} \frac{\boldsymbol{b}_i \cap \boldsymbol{b}_{i'}}{\boldsymbol{b}_i \cup \boldsymbol{b}_{i'}},$$

where $\boldsymbol{b}_i$ denotes the area of the bounding box i. Recall that $\boldsymbol{b}_i$ and $\boldsymbol{b}_{i'}$ are normalized and computed in the reference coordinate systems of their respective regions i and i'. Normalizing $\xi_{LL'}(y, \boldsymbol{x}, \boldsymbol{x}')$ over the number of pairs of regions in L and L' that have class $y_i = y_j$ and relationship $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ would result in an estimate of the posterior probability $p(o_{ij} = 1|y_i = y_j, \boldsymbol{x}_i, \boldsymbol{x}_j)$ that two regions with the same class label belong to a single object instance in the image. Intuitively, this posterior is directly proportional to the area overlap of the associated object bounding boxes.

In the following section, we explain how to use Φ_L, $\Psi_{LL'}$, and $\Xi_{LL'}$ in CRF inference.

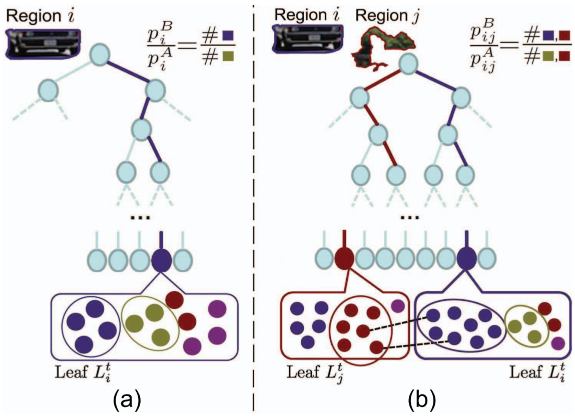

Region i
$\frac{p_i^B}{p_i^A} = \frac{\#🟦}{\#🟩}$
Leaf L_i^t
(a)
Region i
Region j
$\frac{p_{ij}^B}{p_{ij}^A} = \frac{\#🟦,🟥}{\#🟩,🟥}$
Leaf L_j^t
Leaf L_i^t
(b)

Fig. 4. Estimating the following distribution ratios: (a) p_i^B/p_i^A and (b) $p(y_i^B, y_j^B|\boldsymbol{x}_i, \boldsymbol{x}_j)/p(y_i^A, y_j^A|\boldsymbol{x}_i, \boldsymbol{x}_j)$, using HF. When a new image is encountered, its regions are "dropped" down through T decision trees of HF. The figure shows that image regions i and j have reached leaf nodes L_i^t and L_j^t in trees $t = 1, \ldots, T$. The colored circles in L_i^t and L_j^t represent training examples collected in learning, where each color corresponds to one object class. Black dashed lines connect pairs of nodes that have relationship $(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

6 Estimating the Distribution Ratios

Given HF, we are in a position to estimate the distribution ratios in (8) and (9), and thus conduct MH jumps in CRF inference. When a new image is encountered, its regions are first "dropped" down through T decision trees of HF, as illustrated in Fig. 4. Suppose regions i and j reach leaf nodes L_i^t and L_j^t, in trees $t = 1, \ldots, T$, as shown in Fig. 4. Then, we use the three statistics stored in L_i^t and L_j^t—namely, one-class counts $\Phi_{L_i^t}$, two-class counts $\Psi_{L_i^t L_j^t}$, and bounding box overlaps $\Xi_{L_i^t L_j^t}$—to compute the ratio of posteriors p_i^B/p_i^A and p_{ij}^B/p_{ij}^A appearing in (9), and the ratio of proposal distributions $q(B{\to}A)/q(A{\to}B)$ in (8).

In particular, suppose that an MH jump from state A to state B, in CRF inference, involves recoloring two regions i and j from their labels y_i^A and y_j^A in state A to new labels y_i^B and y_j^B in state B. Then, the ratio of their posterior distributions can be estimated using the class histograms, stored in L_i^t and L_j^t, as

$$\frac{p_i^B}{p_i^A} \approx \frac{1}{T} \sum_{t=1}^{T} \frac{\phi_{L_i^t}\big(y_i^B\big)}{\phi_{L_i^t}\big(y_i^A\big)} \ . \tag{10}$$

This is illustrated in Fig. 4a.

Regarding the posterior of region pairs, it is convenient to express it as: $p_{ij} = p(y_i, y_j|\boldsymbol{x}_i, \boldsymbol{x}_j) p(o_{ij}|y_i, y_j, \boldsymbol{x}_i, \boldsymbol{x}_j)$, and then to estimate the ratio p_{ij}^B/p_{ij}^A as a product of two ratios:

$$\frac{p\big(y_i^B, y_j^B|\boldsymbol{x}_i, \boldsymbol{x}_j\big)}{p\big(y_i^A, y_j^A|\boldsymbol{x}_i, \boldsymbol{x}_j\big)} \frac{p\big(o_{ij}^B|y_i^B, y_j^B, \boldsymbol{x}_i, \boldsymbol{x}_j\big)}{p\big(o_{ij}^A|y_i^A, y_j^A, \boldsymbol{x}_i, \boldsymbol{x}_j\big)}.$$

Using the two-class histograms stored in HF, we compute

$$\frac{p\big(y_i^B, y_j^B|\boldsymbol{x}_i, \boldsymbol{x}_j\big)}{p\big(y_i^A, y_j^A|\boldsymbol{x}_i, \boldsymbol{x}_j\big)} \approx \frac{1}{T} \sum_{t=1}^{T} \frac{\psi_{L_i^t L_j^t}\big(y_i^B, y_j^B, \boldsymbol{x}_i, \boldsymbol{x}_j\big)}{\psi_{L_i^t L_j^t}\big(y_i^A, y_j^A, \boldsymbol{x}_i, \boldsymbol{x}_j\big)} \ . \tag{11}$$

This is illustrated in Fig. 4b.

For estimating

$$\frac{p\big(o_{ij}^B|y_i^B, y_j^B, \boldsymbol{x}_i, \boldsymbol{x}_j\big)}{p\big(o_{ij}^A|y_i^A, y_j^A, \boldsymbol{x}_i, \boldsymbol{x}_j\big)},$$

note that it suffices to specify only the ratio

$$\frac{p\big(o_{ij}^B = 1|y_i^B = y_j^B, \boldsymbol{x}_i, \boldsymbol{x}_j\big)}{p\big(o_{ij}^A = 1|y_i^A = y_j^A, \boldsymbol{x}_i, \boldsymbol{x}_j\big)}.$$

This is because $p(o_{ij} = 0|y_i, y_j, \boldsymbol{x}_i, \boldsymbol{x}_j) = 1 - p(o_{ij} = 1|y_i, y_j, \boldsymbol{x}_i, \boldsymbol{x}_j)$, and if $y_i \neq y_j$, then $p(o_{ij} = 1|y_i, y_j, \boldsymbol{x}_i, \boldsymbol{x}_j) = 0$. Thus, using the posterior estimates $\xi_{L_i^t L_j^t}$ stored in HF, we readily have

$$\frac{p\big(o_{ij}^B = 1|y^B = y_i^B = y_j^B, \boldsymbol{x}_i, \boldsymbol{x}_j\big)}{p\big(o_{ij}^A = 1|y^A = y_i^A = y_j^A, \boldsymbol{x}_i, \boldsymbol{x}_j\big)} \approx \frac{1}{T} \sum_{t=1}^{T} \frac{\frac{\xi_{L_i^t L_j^t}(y^B, \boldsymbol{x}_i, \boldsymbol{x}_j)}{N_{L_i^t L_j^t}(y^B, \boldsymbol{x}_i, \boldsymbol{x}_j)}}{\frac{\xi_{L_i^t L_j^t}(y^A, \boldsymbol{x}_i, \boldsymbol{x}_j)}{N_{L_i^t L_j^t}(y^A, \boldsymbol{x}_i, \boldsymbol{x}_j)}} \ , \tag{12}$$

where $N_{LL'}(y, \boldsymbol{x}, \boldsymbol{x}')$ is the number of all training region pairs (k, l), stored in L and L', that satisfy the following conditions: $k{\in}L$, $l{\in}L'$, $y = y_k = y_l$, and $(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}_k, \boldsymbol{x}_l)$.

Finally, to estimate the ratio of proposal distributions $\frac{q(B\to A)}{q(A\to B)}$, we need to compute each individual p_{ij} in (8) rather than simply multiplying the estimates of their ratios. This is because the numerator and denominator of $\frac{q(B\to A)}{q(A\to B)}$ in (8) do not contain the same set of "cut" edges, $\text{Cut}_A \neq \text{Cut}_B$. Since $p_{ij} = p(y_i, y_j|\boldsymbol{x}_i, \boldsymbol{x}_j) p(o_{ij}|y_i, y_j, \boldsymbol{x}_i, \boldsymbol{x}_j)$, we estimate the two posteriors separately. We approximate $p(y_i, y_j|\boldsymbol{x}_i, \boldsymbol{x}_j)$ as an average of two-class histograms:

$$p(y_i, y_j|\boldsymbol{x}_i, \boldsymbol{x}_j) \approx \frac{1}{T} \sum_{t=1}^{T} \frac{\psi_{L_i^t L_j^t}(y_i, y_j, \boldsymbol{x}_i, \boldsymbol{x}_j)}{M_{L_i^t L_j^t}(\boldsymbol{x}_i, \boldsymbol{x}_j)}, \tag{13}$$

where $M_{L_i^t L_j^t}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the number of training region pairs in L_i^t and L_j^t that have relationship $(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

As mentioned above, to approximate $p(o_{ij}|y_i, y_j, \boldsymbol{x}_i, \boldsymbol{x}_j)$, it suffices to estimate $p(o_{ij} = 1|y_i = y_j = y, \boldsymbol{x}_i, \boldsymbol{x}_j)$ as

$$p(o_{ij} = 1|y_i = y_j = y, \boldsymbol{x}_i, \boldsymbol{x}_j) \approx \frac{1}{T} \sum_{t=1}^{T} \frac{\xi_{L_i^t L_j^t}(y, \boldsymbol{x}_i, \boldsymbol{x}_j)}{N_{L_i^t L_j^t}(y, \boldsymbol{x}_i, \boldsymbol{x}_j)}, \tag{14}$$

where $N_{L_i^t L_j^t}(y, \boldsymbol{x}_i, \boldsymbol{x}_j)$ is the number of all training region pairs (k, l), stored in L_i^t and L_j^t, that satisfy the following conditions: $k{\in}L_i^t$, $l{\in}L_j^t$, $y = y_k = y_l$, and $(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_k, \boldsymbol{x}_l)$.

In summary, an MH jump in CRF inference from state A to B requires computation of $qr(A \to B) = \frac{q(B\to A)}{q(A\to B)}$, given by (8), and $pr(A \to B) = \frac{p(\boldsymbol{M}=B|G)}{p(\boldsymbol{M}=A|G)}$, given by (9). We estimate $pr(A \to B)$ as a product of expressions specified in (10), (11), and (12). To compute $qr(A \to B)$, we estimate each posterior distribution p_{ij} in (8) as a product of expressions specified in (13) and (14).

MH provides theoretical guarantees of convergence to the globally optimal solution of a given energy function. However, we approximate the energy function in (10)-(14), and thus the final MH solution is only an approximation with respect to the original energy function.

In the following, we first present our empirical evaluation, and then derive the theoretical performance bounds.

7 Results

Datasets. We evaluate HFRF on three benchmark datasets: the MSRC dataset [3], the Street-Scene dataset [6], [51] and the PASCAL VOC 2007 dataset [52]. The MSRC dataset consists of 591 images showing objects from 21 categories. We duplicate the evaluation setup of [3], i.e., we use the standard random split of MSRC dataset into training and test images. Note that object classes in MSRC images are manually segmented, but bounding boxes around individual instances are not provided. Therefore, instead of HF, we use Random Forest for this dataset, as explained in Section 5.2. That is, for thr MSRC dataset, we use the approach of our initial work, termed RFRF, presented in [48]. The Street-Scene dataset consists of 3,547 images of urban environments, where individual objects (cars, pedestrians, and bicycles) are annotated with bounding boxes. As in [6], one-fifth of the Street-Scene images are used for testing and the rest for training. The PASCAL VOC 2007 dataset consists of 9,963 images depicting 20 object classes, where each object is annotated with a bounding box. The data are split into 50 percent for training and 50 percent for testing, where the distributions of images and objects by class are approximately uniform across the training and test sets.

Evaluation metrics. Object recognition and segmentation are evaluated using the pixel-wise classification accuracy, averaged across all test images and object classes. This metric is suitable because it does not favor object classes that occur in images more frequently. Object detection is evaluated in terms of Average Precision (AP) according to the VOC protocol [52]. The AP summarizes the precision-recall curve, and is specified as the average precision at a set of 11 equally spaced recall values $[0, 0.1, \dots, 1]$. We also evaluate object segmentation error for all objects that are correctly detected, i.e., for true positives (TPs). This is computed as pixel-wise segmentation error with respect to the ground truth object masks. These masks are available for the Street-Scene and PASCAL datasets.

Training setup. Images are segmented using the state-of-the-art multiscale segmentation algorithm of [49]. This algorithm takes the perceptual significance of a region boundary, $P_b \in [0, 255]$, as an input parameter. We vary $P_b =$ 30:10:150, and thus obtain a hierarchy of regions for each image. A region is characterized by a descriptor vector consisting of the following properties:

1. 30-bin color histogram in the CIELAB space,
2. 250D histogram of filter responses of the MR8 filter bank and the Laplacian of Gaussian filters computed at each pixel, and mapped to 250 codewords whose dictionary is obtained by K-means over all training images,
3. 128D region boundary descriptor measuring oriented contour energy along eight orientations of each cell of a 4×4 grid overlaid over the region's bounding box,
4. coordinates of the region's centroid normalized to the image size.

Regions extracted from training images are used to learn the HF. Training images are labeled with bounding boxes around object occurrences, so each region that falls within a bounding box is assigned the label of that box. If a region covers a number of bounding boxes of different classes, it is added to the training set multiple times to account for each distinct label. Each object region is also associated with an offset vector from the region's centroid to the center of the bounding box. This vector is normalized with respect to the reference coordinate system of the region, as explained in Section 5. As is standard in HF literature [15], [50], we use equal-size random splits of training data to train $T = 10$ decision trees of HF. The complexity of this step is $O(m)$, where m is the total number of regions. The growth of each tree is constrained so its depth is less than 15 and its every leaf node contains at least $R = 30$ training examples.

Testing setup. To recognize and segment objects in a new test image, we first extract a hierarchy of regions from the image by the segmentation algorithm of [49]. Then, we build the fully connected CRF graph from extracted regions (Section 3), and run the SW-cut inference (Section 4). We use HF to estimate the distribution ratios required for Metropolis-Hastings jumps. Note that we do not require a threshold as input parameter for inferring the labels of nodes and edges of our CRF.

We examine the following three variants of our approach. HFRF-1: The spatial relationships of regions, $(\boldsymbol{x}_i, \boldsymbol{x}_j)$, are not accounted for when computing p_{ij} in (11)-(14). HFRF-2: The region relationships touching and far are considered, while the part of relationship is not accounted for. HFRF-3: All three types of region layout and structural relationships are modeled. In this paper, we consider HFRF-3 as our default variant and explicitly state when the other two are used instead. Note that considering region layouts and part of relationships changes only the three statistics recorded in leaf nodes of HF, but it does not affect complexity.

Also, we compare HFRF with our initial model RFRF, presented in [48], on a task where this comparison is possible. Since RFRF is not capable of detecting instances of an object class, the comparison is limited to the task of assigning object class labels to image regions.

7.1 Quantitative Results

Convergence rate. We compare the convergence rates of HFRF inference with that of RFRF inference [48], and the standard SW-cut algorithm presented in [14]. The standard SW-cut is used for inference of a random field that defines the Gibbs distribution as

$$\begin{aligned} &p(\boldsymbol{Y}|G) \propto \exp\{-E(\boldsymbol{Y}|G)\}, \\ &E(\boldsymbol{Y}|G) = \sum_i \phi_i(y_i, \boldsymbol{x}_i) + \sum_i \sum_{j \in \mathcal{N}(i)} \psi_{ij}(y_i, y_j, \boldsymbol{x}_i, \boldsymbol{x}_j), \end{aligned} \tag{15}$$

where both the unary potential $\phi_i = \mathbf{w}_1^T \mathbf{f}_1(\boldsymbol{x}_i)$ and the pairwise potential $\psi_{ij} = \mathbf{w}_2^T \mathbf{f}_2(\boldsymbol{x}_i, \boldsymbol{x}_j)$ represent a weighted sum of features $\mathbf{f}_1(\boldsymbol{x}_i)$ associated with region i, and features $\mathbf{f}_2(\boldsymbol{x}_i, \boldsymbol{x}_j)$ associated with pairs of regions i and j. For fair comparison, we use the same set of color, texture, and shape features that we use in our HFRF to define $\mathbf{f}_1(\boldsymbol{x}_i)$. The pairwise features are defined as similarity $\mathbf{f}_2(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\sum_l \beta_l \| f_{1,l}(\boldsymbol{x}_i) - f_{1,l}(\boldsymbol{x}_j)\|^2)$. To learn the parameters $\mathbf{w}_1$, $\mathbf{w}_2$, and $\{\beta_l\}$ of the Gibbs distribution in (15), we use Stochastic Gradient Descent [61]. This formulation of the Gibbs distribution is very similar to that of [14], and thus allows us to fairly compare the convergence rates of our inference with theirs. Fig. 5 shows the convergence rates of HFRF inference, RFRF inference

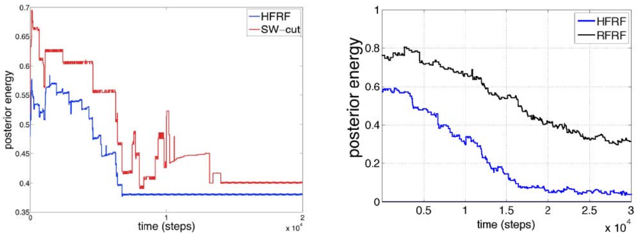


Fig. 5. Average convergence rates: (left) HFRF and SW-cut of a random field with the Gibbs distribution presented in [14] on the Street-Scene test images; (right) HFRF and RFRF [48] on PASCAL VOC 2009 test images.

[48], and the standard SW-cut algorithm presented in [14]. For this comparison, we use test images of the Street-Scene and PASCAL VOC 2009 datasets. As can be seen, HFRF inference converges, on average, twice as fast as the original SW-cut algorithm [14], and to a lower energy. Similarly, HFRF inference converges faster than RFRF inference. One reason for this is that edges of HFRF are labeled, unlike in RFRF. This helps guide the MH jumps faster toward the MAP solution of HFRF inference by taking into account object instance labeling.

Object recognition and segmentation. Tables 1 and 2 show our pixel-wise classification accuracy on PASCAL 2009, Street-Scene, and MSRC images. Table 2 also compares the three variants of HFRF on MSRC and Street-Scene images with the state-of-the-art CRF-based approaches [3], [6], [56], [57]. The standard deviations in this table are computed across classes, as is standard for these datasets—-not over different training/validation/test data splits. In Table 2, we observe higher standard deviations on the MSRC dataset than those on the Street-Scene dataset. This is, in part, because we do not have access to bounding boxes of object instances in training on MSRC. By contrast, the Street-Scene provides the bounding boxes, and thus we improve accuracy by exploiting richer annotations in training. We also observe that HFRF has the lowest standard deviations. As can be seen, the additional consideration of the spatial relationships—touching and far—increases performance relative to that of HFRF-1. Our performance is best when all three types of region relationships are modeled.

TABLE 3
Object Detection Average Precision
on the Street-Scene Dataset

Method	Car	Pedestrian
Patch baseline	40	15
HOG baseline	35	37
[6]	58	35
[48]	31	13
Ours	**63**	**38**

Although we outperform the approaches of [6], [56] in Table 2, note that this comparison is unfair to us since they additionally use higher level cues about the horizon location and 3D scene layout in their object recognition and segmentation. In addition, HFRF outperforms the latest CRF models on both datasets. Table 2 shows that HFRF improves the results of our initial RFRF-based approach [48]. Finally, Table 2 also presents computation times of our and competing methods. As can be seen, the faster convergence rates of MH-based inference of HFRF leads to speed-ups relative to RFRF [48].

Object detection. We run HFRF inference five times with random restarts, and take the inference solution with the maximum posterior distribution $p(\boldsymbol{M}|G)$. Each connected component CC in that solution represents a detected object instance and, simultaneously, its segmentation. The marginal posterior of a CC, given by $\prod_{i,j\in CC} p_i p_{ij}$, is taken as a confidence score of the corresponding object detection. For evaluating the AP, as in the VOC protocol [52], we choose 11 thresholds of the confidence scores to obtain 11 equally spaced recall values $[0, 0.1, \ldots, 1]$. We fit a bounding box to every segmented object. A detected object is considered TP if the largest intersection between BB and a nearby ground-truth bounding box (GT) is greater than one half of their union, TP: $\max_{\mathrm{GT}} \frac{\mathrm{intersect(BB,GT)}}{\mathrm{union(BB,GT)}} > 0.5$; otherwise, a detected object is considered false positive.

Table 3 shows our car and pedestrian AP on the Street-Scene dataset. We present a comparison with two baseline

TABLE 1
Average Pixel-Wise Classification Accuracy (i.e., Object Segmentation Results) on the PASCAL VOC 2009 Dataset

	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	avg
[53]	64.3	21.8	21.7	32.0	**40.2**	**57.3**	49.4	**38.8**	5.2	28.5	22.0	19.6	33.6	45.5	**33.6**	**27.3**	**40.4**	18.1	33.6	**46.1**	**36.3**
[54]	48.3	6.7	19.1	10.0	16.6	32.7	38.1	25.3	5.5	9.4	25.1	13.3	12.3	35.5	20.7	13.4	17.1	18.4	37.5	36.4	24.8
[55]	62.3	**24.1**	**28.3**	30.2	32.7	42.2	48.1	22.8	9.1	30.1	7.9	**21.5**	41.9	**49.6**	31.5	26.1	37.0	**20.1**	39.4	31.1	34.1
Ours	**69.6**	24.0	26.6	**34.4**	38.5	50.5	**54.3**	37.4	**14.8**	**32.1**	**25.7**	20.1	**42.2**	44.4	30.9	26.2	40.0	19.7	**44.6**	38.3	35.7

Our segmentation results are competitive with the best-performing approaches of the 2009 VOC challenge [53], [54], [55].

TABLE 2
Pixel Classification Accuracy, Standard Deviation and Computation Times Averaged across All Object Classes
of the Three Variants of Our Approach on the MSRC and Street-Scene Datasets

Method	StreetScene	Test time
HFRF-1	77.1%±0.6%	13-15s
HFRF-2	89.7%±0.7%	18-20s
HFRF-3	**91.4%±0.6%**	24-27s
[56]	83.0%	N/A
[6]	84.2%	N/A
[48]	89.8%±0.6%	30-32s

Method	MSRC	Test time
RFRF-1	69.5%±13.7%	20-23s
RFRF-2	80.2%±14.4%	25-27s
RFRF-3 [48]	**82.9%±15.8%**	30-32s
[57]	70.0%±25.4%	N/A
[56]	76.4%	N/A
[3]	70.0%±29.7%	10-30s

The standard deviations are computed across classes, as is standard for these datasets—not over different training/validation/test data splits. The MSRC dataset does not provide ground truth bounding boxes around object instances; therefore, instead of HF, we are bound to use Random Forest for MSRC dataset. The comparison is with the state-of-the-art CRF methods presented in [3], [6], [56], [57], and with our RFRF [48].

TABLE 4
Object Detection Average Precision on the PASCAL VOC 2007 Dataset

	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	avg
[58]	29.0	54.6	0.6	13.4	26.2	39.4	46.4	16.1	16.3	16.5	24.5	5.0	43.6	37.8	35.0	8.8	17.3	21.6	34.0	39.0	26.8
[59]	**29.4**	**55.8**	**9.4**	14.3	28.6	44.0	51.3	**21.3**	20.0	19.3	25.2	12.5	50.4	38.4	36.6	**15.1**	19.7	**25.1**	36.8	**39.3**	29.6
[60]	27.7	54.0	6.6	15.1	14.8	44.2	47.3	14.6	12.5	22.0	24.2	12.0	**52.0**	42.0	26.8	10.6	22.9	18.8	35.3	31.1	26.7
Ours	28.9	54.7	5.2	**17.2**	**29.7**	**48.9**	**52.6**	15.9	**22.7**	**24.9**	**26.4**	**13.1**	51.1	**44.5**	**38.7**	10.1	**25.6**	22.0	**37.3**	38.0	**30.4**

We compare HFRF detection results to the state-of-the-art object detectors of Felzenswalb et al. [58], Zhu et al. [59], and Pedersoli et al. [60].

sliding-window approaches, as well as a comparison with [6]. Our method significantly improves over the baselines and outperforms [6] as well. Table 4 shows that the AP of HFRF is comparable to the state-of-the-art object detectors on the PASCAL VOC 2007 dataset. On the Street Scene dataset, the standard deviation of AP across all object classes is 0.2 percent, and on the PASCAL 2007 dataset, it is only 0.03 percent. We evaluate the segmentation error of $54.8\% \pm 4.2\%$ on this dataset, which is mainly due to errors in the low-level segmentation. For example, Fig. 9a shows a region where the tree is merged with its shadow in the low-level segmentation due to low contrast. We also note that the increase in the number of random restarts, beyond five times, does not affect our AP on these two datasets.

7.2 Qualitative Results

Our object detection, recognition, and segmentation results on example images from the MSRC, Street-Scene, and PASCAL datasets are shown in Figs. 6, 7, and 8. Labels of the finest-scale regions are depicted using distinct colors since pixels get labels of the finest-scale regions. Detected object instances are also delineated with colored boundaries, and tagged with the class label. Each tag corresponds to one detected instance. As can be seen, HFRF correctly identifies groups of regions that belong to the same class and is also able to segment individual objects. Fig. 9 shows some failure examples, as well as a comparison to [48]. We note that there are two main reasons for inaccurate detection. First, we cannot recover from errors in the low-level segmentation, e.g., when an object part is merged with the background (the train is merged with its shadow in Fig. 9b), or two objects are merged together (the two pedestrians in Fig. 9a). We also note that most of the detection errors come from overlapping objects being merged into one; see, for example, the sheep in Fig. 9c. This is because when objects are close to each other, their parts vote for a similar location of their center, and thus these regions get merged under the same instance. Note that the methods that use standard nonmaximum suppression for object detection, e.g., [15], [58], suffer from the same problem.

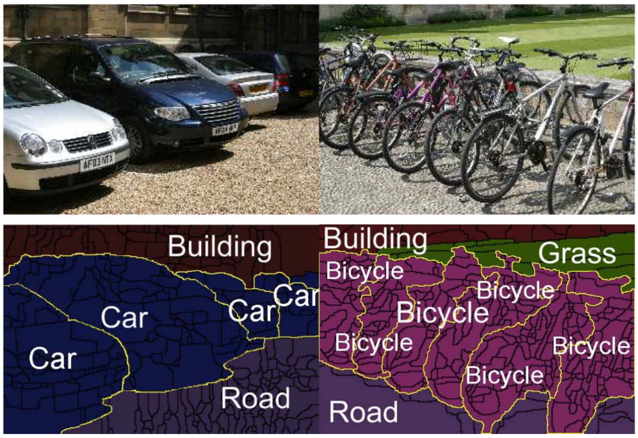


Fig. 6. Object instance detection and segmentation by HFRF on example images from the MSRC dataset. Black boundaries mark the finest-scale regions found by the multiscale segmentation algorithm of [49]. Regions occupied by detected object classes are color coded, where each color corresponds to one class. Detected object instances are also delineated with colored boundaries, and tagged with the class label. Each tag corresponds to one detected instance. The results are good despite occlusion and changes in illumination and scale. (best viewed in color).

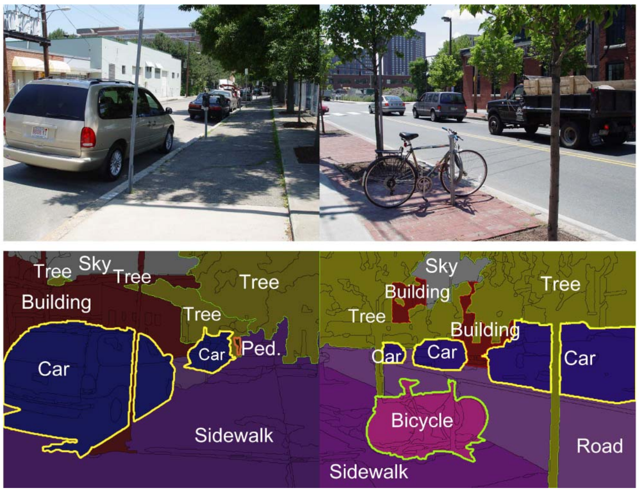


Fig. 7. Object instance detection and segmentation by HFRF on example images from the Street-Scene dataset. See the caption of Fig. 6. Instance detection fails on the textured trees with low contrast regions and the row of buildings. On the left, HFRF correctly merges the two separate car regions into one instance.

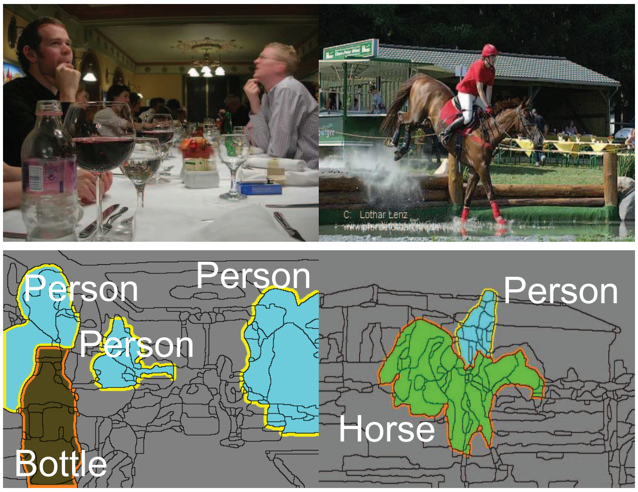


Fig. 8. Object instance detection and segmentation by HFRF on example images from the PASCAL dataset. See the caption of Fig. 6.

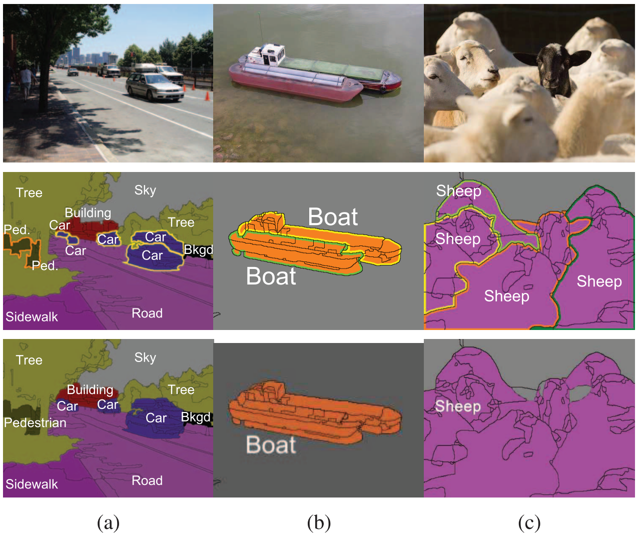


Fig. 9. Comparison HFRF (middle row) versus RFRF of [48] (bottom row). See the caption of Fig. 6. The top row shows original images. (a) Street-scene dataset—RFRF merges the two pedestrians into one detection, whereas HFRF corrects this. (b) The PASCAL VOC 2007 dataset—RFRF merges the two boats into one detection, whereas HFRF corrects this. (c) The PASCAL VOC 2007 dataset—RFRF merges many sheep into one detection because they all vote for a similar center location, whereas HFRF corrects this. HFRF also improves object segmentation results of RFRF by correctly separating object regions from background regions, e.g., the sheep regions labeled as background by RFRF in (c).

7.3 Implementation Details

We chose the optimal configuration of HF in terms of the number of decision trees, $T = 10$, the maximum depth of each tree, $d = 15$, and the minimum number of regions in each leaf node, $R = 30$, using the following grid testing: $T = 5, 10, 20, 50$, $d = 10, 15, 20, 25$, and $R = 10, 30, 50, 100$.

Since the depth of each decision tree in HF is less than 15 (i.e., approximately constant), the complexity of dropping an instance through one tree is $O(1)$, and through HF with T trees is $O(T)$. Our C-implementation of the HF-guided SW-cut inference of CRF takes 10 to 30 s on a 2.40 GHz PC with 3.48 GB RAM for MSRC, Street-Scene, and PASCAL 2007 images. Table 2 shows that our average running times are comparable to those of the other CRF methods that use approximate inference [3], [6], [56], [57].

8 Theoretical Analysis

The results presented in Section 7 demonstrate that HFRF is an efficient and powerful framework to jointly reason about multiple, statistically dependent random variables and their attributes. In this section, we seek a theoretical explanation for such a good performance. In particular, we derive the theoretical performance bounds of HFRF for the two-class learning problem for simplicity.

Random Forest is difficult to analyze [50], [62], as is HF. Regarding the consistency of Random Forest, it is known that their rate of convergence to the optimal Bayes rule depends only on the number of informative variables. It is also shown that Random Forest, which cuts down to pure leaves, uses a weighted, layered, nearest neighbor rule [62]. We are not aware of any theoretical analysis of HF as an estimator of ratios of posterior distributions.

As explained in Section 4, we use the SW-cut for HFRF inference. The SW-cut iterates the Metropolis-Hastings reversible jumps, and thus explores the state space of solutions. An MH jump from state A to another state B is controlled by the acceptance rate $\alpha(A \rightarrow B)$, which depends on the ratios of the proposal and posterior distributions, $\frac{q(B\rightarrow A)p(\boldsymbol{M}=B|G)}{q(A\rightarrow B)p(\boldsymbol{M}=A|G)}$. Below, we show that the error made by the two-class HFRF in estimating these ratios is bounded. Our derivation of the error bounds of HFRF is based on the theoretical analysis of evidence trees, presented in [47].

8.1 An Upper Error Bound of HFRF

In general, MH allows jumps from states with higher posterior distributions to states with lower posteriors. An error occurs when a *balanced* reversible jump, $\frac{q(B\rightarrow A)}{q(A\rightarrow B)} = 1$, is encountered, i.e., when there is no preference between jumping from state A to state B and reverse, and the posterior distribution of state A is lower than that of B. In this case, $\alpha(A \rightarrow B) = 1$ and the SW-cut will erroneously visit state B. We are interested in finding the probability of this error, $P(\epsilon) = P(\frac{p(\boldsymbol{M}=B|G)}{p(\boldsymbol{M}=A|G)} \geq 1)$, specified as

$$P(\epsilon) = P\left(\prod_{i\in CC} \frac{p_i^B}{p_i^A} \prod_{j\in\mathcal{N}(i)} \frac{p_{ij}^B}{p_{ij}^A} \geq 1\right). \tag{16}$$

$P(\epsilon)$ can be computed by estimating the pdf of a product of random variables $C_i = p_i^B/p_i^A \in [0,\infty)$ and $D_{ij} = p_{ij}^B/p_{ij}^A \in [0,\infty)$, within a graph's connected component, $i \in CC$, where $|CC| = n, i = 1,\ldots,n$, and $j \in \mathcal{N}(i)$. In the following, we will show how to compute the distributions $f_C(c)$ and $f_D(d)$ for the products $C = \prod_{i=1}^n C_i$ and $D = \prod_{i=1}^n \prod_{j\in\mathcal{N}(i)} D_{ij}$. From these distributions and (16), we will derive the probability that HFRF makes a wrong prediction, $P(\epsilon) = P(C \cdot D \geq 1)$.

8.2 A Mathematical Model of HFRF Performance

In this section, we derive that the HFRF estimates of the ratios C_i, D_{ij} have the exponential distribution. We consider a binary classification problem, where training and test instances may have positive and negative labels. The two classes are balanced $P(y = +1) = P(y = -1) = 1/2$. We define π to be a fraction of pairs of training examples that have a certain spatial relationship. The learning algorithm that creates HF is not modeled. Instead, we assume that the learned decision trees have the following properties.

Each node of the tree can be either a c-node (for class) or an s-node (for spatial). c-nodes split the training data so as to minimize the class uncertainty, $U_1(\cdot)$, whereas s-nodes split the training examples so as to minimize the spatial offset uncertainty $U_2(\cdot)$, defined in Section 5.

Each leaf node of a tree: 1) stores a total of R training instances, and 2) has a probabilistic margin $\gamma \in [0, 1/2)$ for classification. By margin, we mean that in every leaf reached by R training instances a fraction of $1/2 + \gamma$ of the training instances will belong to one class (e.g., positive) and a fraction of $1/2 - \gamma$ of them will belong to the other class (e.g., negative). We say that a leaf is positive if a majority of the training instances collected by the leaf are positive or, otherwise, we say that the leaf is negative.

Each pair of leaf nodes in a tree also has a probabilistic margin γ_o, i.e., there is a probability $1/2 + \gamma_o$ that HF will correctly label the hidden random variable o_{ij} of two regions i and j that fall in those two leaves.

A new test instance is classified by dropping it through T classification trees, and taking a majority vote of the labels of all $R \cdot T$ training instances stored in the leaves reached by the test instance. The object term o_{ij} of a pair of test instances (i, j) with relationship e is classified by dropping both instances through T classification trees, and taking a majority vote of the labels of all $\pi R^2 \cdot T$ training instances with relationship e stored in the leaves reached by the test instances. We refer to this classification procedure as evidence voting [47], as opposed to decision voting over the leaf labels in the standard HF [50].

In the following, we will first show in Lemma 1 that the probabilistic margin γ is a function of two margins—namely, the margin g_c defined for c-nodes, and the margin g_s defined for s-nodes. Then, we will prove in Propositions 1, 2, and 3 that the random variables C_i, D_{ij} have exponential distributions. Below, we first give formal definitions of the margins g_c and g_s.

Definition 1. *g_c is the probabilistic margin of HF at any c-node. This means that an instance that arrives at a c-node will be correctly routed down the decision tree with probability $1/2 + g_c$.*

Definition 2. *g_s is the probabilistic margin of HF at any s-node. This means that an instance that arrives at a s-node will be correctly routed down the decision tree with probability $1/2 + g_s$.*

A motivating example. This paragraph presents an example specification of g_c and g_s. Thus, g_c can be defined, e.g., as the smallest distance from any data point to a decision boundary defined by the split at that c-node. Also, g_s can be defined, e.g., as the symmetric Kullback-Leibler divergence between the two distributions of bounding-box offset vectors created by the split at that s-node.

Lemma 1. *Given the probabilistic margins g_c and g_s at each c-node and s-node, the probability that HF correctly labels an instance is $1/2 + \gamma$, where $\gamma = g_c^2 + g_c g_s$.*

Proof. An instance dropped down a decision tree falls with equal probability in a c-node, n_c, or in an s-node, n_s, i.e., $p(n_c) = p(n_s) = 1/2$. For n_c, an instance is routed correctly (e.g., to a positive leaf if it is a positive instance) with probability $1/2 + g_c$, and labeled correctly with probability $1/2 + g_c$. It is routed incorrectly with probability $1/2 - g_c$ and labeled incorrectly with probability $1/2 - g_c$. For node n_s, an instance is routed correctly with probability $1/2 + g_s$, and labeled correctly with probability $1/2 + g_c$. It is routed incorrectly with probability $1/2 - g_s$ and labeled incorrectly with probability $1/2 - g_c$. Hence, the probability that at any leaf node, an instance is labeled correctly is

$$\begin{aligned} &1/2[(1/2 + g_c)(1/2 + g_c) + (1/2 - g_c)(1/2 - g_c)] \\ &\quad + 1/2[(1/2 + g_s)(1/2 + g_c) + (1/2 - g_s)(1/2 - g_c)] \\ &= 1/2 + g_c^2 + g_c g_s = 1/2 + \gamma. \end{aligned}$$

□

8.2.1 Distribution of the Random Variables C_i

Denote $P(\epsilon_1) = P(C_i \geq 1)$ the probability that evidence voting misclassifies an instance. The following proposition states that this probability is upper bounded.

Proposition 1. *The probability that HF with T trees, where every leaf stores R training instances, incorrectly classifies an instance is upper bounded, $P(\epsilon_1) \leq \exp(-2RT\gamma^2)$.*

Proof. Evidence voting for labeling an instance can be formalized as drawing a total of $R \cdot T$ independent Bernoulli random variables, with success rate p_1, whose outcomes are $\{-1, +1\}$. $+1$ is received for correct and -1 for incorrect labeling of the instance. The success rate p_1 is the probability that an instance is correctly labeled, i.e., $p_1 = 1/2 + \gamma$ (Lemma 1). Let $\mathcal{S}_1$ denote a sum of these Bernoulli random variables. A positive instance is incorrectly labeled if $\mathcal{S}_1 \leq 0$, and a negative instance is misclassified if $\mathcal{S}_1 > 0$. Since the two classes are balanced, by applying the standard Chernoff bound, we obtain $P(\epsilon_1) = P(\mathcal{S}_1 \leq 0) \leq \exp(-2RT\gamma^2)$. □

From Proposition 1, it follows that the probability that HF makes a wrong prediction about the posterior ratio of an instance is upper bounded, $P(C_i \geq 1) = P(\epsilon_1) \leq \exp(-2RT\gamma^2), \forall i \in CC$. This gives the probability density function $f_{C_i}(c) = \lambda_1 \exp(-\lambda_1 c)$, where $\lambda_1 = 2RT\gamma^2$. Then, it follows that the product $C = \prod_{i=1}^n C_i = (C_i)^n$ has the distribution $f_C(c) = \frac{\lambda_1}{n} c^{\frac{1-n}{n}} \exp(\lambda_1 c^{\frac{1}{n}})$.

8.2.2 Distribution of the Random Variables D_{ij}

We study two cases for the random variables D_{ij}. In case 1, we assume that $p_{ij} = p(y_i, y_j | \boldsymbol{x}_i, \boldsymbol{x}_j)$—which is the setting of our initial work, presented in [48]. Thus, we have $D_{ij} = D_{ij;1} = \frac{p(y_i^B, y_j^B | \boldsymbol{x}_i, \boldsymbol{x}_j)}{p(y_i^A, y_j^A | \boldsymbol{x}_i, \boldsymbol{x}_j)}$. In case 2, we consider the extended definition of p_{ij}, as specified in Section 3, i.e., $p_{ij} = p(o_{ij}, y_i, y_j | \boldsymbol{x}_i, \boldsymbol{x}_j)$. Thus, we have $D_{ij} = D_{ij;1} \cdot D_{ij;2}$, where

$$D_{ij;1} = \frac{p\left(y_i^B, y_j^B | \boldsymbol{x}_i, \boldsymbol{x}_j\right)}{p\left(y_i^A, y_j^A | \boldsymbol{x}_i, \boldsymbol{x}_j\right)} \text{ and } D_{ij;2} = \frac{p\left(o_{ij}^B | y_i^B, y_j^B, \boldsymbol{x}_i, \boldsymbol{x}_j\right)}{p\left(o_{ij}^A | y_i^A, y_j^A, \boldsymbol{x}_i, \boldsymbol{x}_j\right)}.$$

Below, we first consider case 1.

Evidence voting is also used for labeling pairs of instances. The probability that evidence voting misclassifies a pair of instances, $P(\epsilon_2) = P(D_{ij;1} \geq 1)$, is upper bounded, as stated in Proposition 2.

Proposition 2. *The probability that HF with T trees, where every leaf stores R training instances, incorrectly labels a pair of instances with relationship e is upper bounded, $P(\epsilon_2) \leq \exp(-2\pi R^2 T \zeta^2)$, with $\zeta = \gamma^2 + \gamma - 1/4$.*

Proof. The proof is similar to the proof of Proposition 1. The probability that an instance is correctly labeled is $1/2 + \gamma$, so *a pair* of instances is correctly labeled with probability $(1/2 + \gamma)^2 = \frac{1}{2} + \zeta$, with $\zeta = \gamma^2 + \gamma - \frac{1}{4}$. Note that ζ is positive, provided that $\gamma > \frac{\sqrt{2}-1}{2}$, which we enforce in practice when learning HF. Hence, ζ represents the margin of HF for classifying pairs of instances.

There are $R^2 T$ pairs of training instances in the leaves of HF. Since π is the fraction of pairs of instances with a particular relationship e, there are a total of $\pi R^2 T$ pairs of training instances with that relationship. Evidence voting for labeling a pair of instances can now be

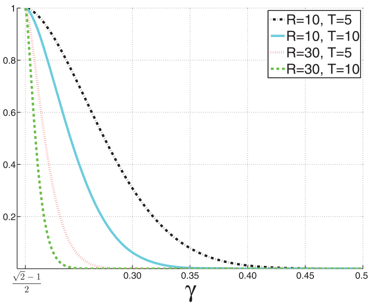


Fig. 10. **Case 1.** Influence of the classifier's probabilistic margin, γ, on the probability of error $P(\epsilon)$, (16). We plot $P(\epsilon)$ for multiple values of T and R, where T is the number of trees in the HF and R is the number of training instances in each leaf node. As mentioned in Proposition 2, $\gamma \in [\frac{\sqrt{2}-1}{2}; \frac{1}{2}]$, so the dashed region is the range where $P(\epsilon)$ is not defined.

formalized as drawing $\pi R^2 T$ independent Bernoulli random variables, with success rate p_2, whose outcomes are $\{-1, +1\}$, where $+1$ is received for correct and -1 for incorrect labeling of the instance pair. The success rate p_2 is the probability that an instance pair is correctly labeled, i.e., $p_2 = \frac{1}{2} + \zeta$. Let $\mathcal{S}_2$ denote a sum of these Bernoulli random variables. By applying the standard Chernoff bound, we obtain that $P(\epsilon_2) = P(\mathcal{S}_2 \leq 0) \leq \exp(-2\pi R^2 T \zeta^2)$. □

From Proposition 2, it follows that the probability that HF makes a wrong prediction about the ratio of a pair of instances is upper bounded, $P(D_{ij;1} \geq 1) = P(\epsilon_2) \leq \exp(-2\pi R^2 T \zeta^2)$, $\forall i \in CC$ and $j \in \mathcal{N}(i)$. This gives the probability density function $f_{D_{ij;1}}(d) = \lambda_2 \exp(-\lambda_2 d)$, where $\lambda_2 = 2\pi R^2 T \zeta^2$. Then, it follows that the product $D_1 = \prod_{i=1}^n \prod_{j\in\mathcal{N}(i)} D_{ij;1} = (D_{ij;1})^{nk} \approx (D_{ij;1})^n$ has the distribution $f_{D_1}(d) = \frac{\lambda_2}{n} d^{\frac{1-n}{n}} \exp(\lambda_2 d^{\frac{1}{n}})$. We here approximate that the number of edges within CC is the same as the number of nodes in CC as a result of "cutting" graph edges by the SW-cut algorithm.

In case 2, we derive the probability that evidence voting misclassifies the binary o_{ij} random variable of a pair of instances, $P(\epsilon_3) = P(D_{ij;2} \geq 1)$. The following proposition states that this probability is upper bounded.

Proposition 3. *The probability that HF with T trees, where every leaf stores R training instances, incorrectly classifies the object term of a pair of instances is upper bounded, $P(\epsilon_3) \leq \exp(-2\pi R^2 T \gamma_o^2)$.*

Proof. Evidence voting for labeling o_{ij} of a pair of instances can be formalized as drawing a total of $\pi R^2 T$ independent Bernoulli random variables, with success rate p_3, whose outcomes are $\{-1, +1\}$. $+1$ is received for correct, and -1 for incorrect labeling of the instance. The success rate p_3 is the probability that the object term of a pair of instances is correctly labeled, i.e., $p_3 = 1/2 + \gamma_o$. Let $\mathcal{S}_3$ denote a sum of these Bernoulli random variables. By applying the standard Chernoff bound, we obtain $P(\epsilon_3) = P(\mathcal{S}_3 \leq 0) \leq \exp(-2\pi R^2 T \gamma_o^2)$. □

From Proposition 3, it follows that the probability that HF makes a wrong prediction about the ratio of the object terms of pairs of instances is upper bounded, $P(D_{ij;2} \geq 1) = P(\epsilon_3) \leq \exp(-2\pi R^2 T \gamma_o^2)$, $\forall i \in CC$. This gives the

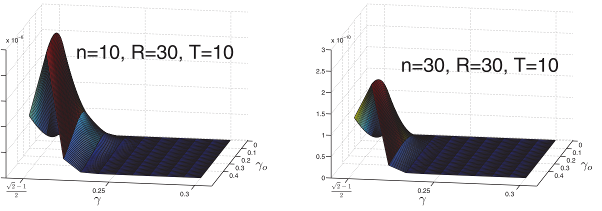


Fig. 11. **Case 2.** Influence of the classifier's probabilistic margins, γ and γ_o, on the probability of error $P(\epsilon)$, (16). We plot $P(\epsilon)$ for two values of n, the number of nodes in a particular connected component ($n = 10$ and $n = 30$). Note that for better visualization, we have cropped the plot to display $\gamma \in [\frac{\sqrt{2}-1}{2}; 0.3]$, but that $P(\epsilon)$ stays equal to zero for $\gamma \in [0.3; 0.5]$ as well.

probability density function $f_{D_{ij;2}}(d) = \lambda_3 \exp(-\lambda_3 d)$, where $\lambda_3 = 2\pi R^2 T \gamma_o^2$. Then, it follows that the product $D_2 = \prod_{i=1}^n \prod_{j\in\mathcal{N}(i)} D_{ij;2} = (D_{ij;2})^{nk} \approx (D_{ij;2})^n$ has the distribution $f_{D_2}(d) = \frac{\lambda_3}{n} d^{\frac{1-n}{n}} \exp(\lambda_3 d^{\frac{1}{n}})$.

8.2.3 Distribution of the Random Variable $H = C \cdot D$

Case 1: $\mathbf{D} = \mathbf{D_1}$. Since the random variable $H = C \cdot D$ is the product of two random variables with exponential distributions, it is possible to analytically derive the probability that HF makes a wrong prediction, $P(\epsilon) = P(C \cdot D \geq 1)$, as stated in the following theorem.

Theorem 1. *The probability that the two-class HF makes a wrong prediction is*

$$P(\epsilon) = P(C \cdot D \geq 1) = 2\sqrt{\lambda_1 \lambda_2} K_1(2\sqrt{\lambda_1 \lambda_2}), \tag{17}$$

where $C \in [0, \infty)$ and $D \in [0, \infty)$ are random variables characterized by the probability density functions $f_C(c) = \frac{\lambda_1}{n} c^{\frac{1-n}{n}} \exp(-\lambda_1 c^{\frac{1}{n}})$ and $f_D(d) = f_{D_1}(d) = \frac{\lambda_2}{n} d^{\frac{1-n}{n}} \exp(-\lambda_2 d^{\frac{1}{n}})$, with parameters λ_1 and λ_2, and where K_1 is the modified Bessel function of the second kind.

Proof. Using the standard derivation steps from the integral theory [63], we derive that $f_H(h) = \int_0^\infty \frac{1}{c} f_C(c) f_{D_1}(\frac{h}{c}) dc = \frac{2\lambda_1\lambda_2}{n} h^{\frac{1-n}{n}} K_0(2\sqrt{\lambda_1\lambda_2} h^{\frac{1}{2n}})$. K_0 is the modified Bessel function of the second kind. It follows that $P(\epsilon) = P(H \geq 1) = 1 - \int_0^1 f_H(h) dh = 2\sqrt{\lambda_1\lambda_2} K_1(2\sqrt{\lambda_1\lambda_2})$, using properties of the Bessel functions of the second kind. □

From Theorem 1, $P(\epsilon)$ decreases when any of the following parameters increases: R, T, γ, and π. Fig. 10 shows the influence of γ on $P(\epsilon)$, when the other parameters are fixed to their typical values: $R = 10$ or 30, $T = 5$ or 10, and $\pi = 0.005$.

Case 2: $D = D_1 \cdot D_2$**.** In this situation, the random variable $H = C \cdot D$ is the product of three random variables with exponential distributions, and there is no closed-form expression for $P(\epsilon)$ [64]. In the following, we will show numerically that $P(\epsilon)$ is upper bounded.

Theorem 2. *The probability that HF makes a wrong prediction is*

$$\begin{aligned} P(\epsilon) &= P(C \cdot D \geq 1) \\ &= \frac{2\lambda_1\lambda_2\lambda_3}{n^2} \int_1^\infty \int_0^\infty h^{\frac{1-n}{n}} \frac{1}{c} \exp\Big(-\lambda_1 c^{\frac{1}{n}}\Big) \cdot \\ &\qquad K_0\Big(2\sqrt{\lambda_2\lambda_3} h^{\frac{1}{2n}} c^{-\frac{1}{2n}}\Big) dc\, dh, \end{aligned}$$

where $C \in [0, \infty)$, $D_1 \in [0, \infty)$, and $D_2 \in [0, \infty)$ are random variables characterized by the probability density functions $f_C(c) = \frac{\lambda_1}{n} c^{\frac{1-n}{n}} \exp(-\lambda_1 c^{\frac{1}{n}})$, $f_{D_1}(d) = \frac{\lambda_2}{n} d^{\frac{1-n}{n}} \exp(-\lambda_2 d^{\frac{1}{n}})$, and $f_{D_2}(d) = \frac{\lambda_3}{n} d^{\frac{1-n}{n}} \exp(-\lambda_3 d^{\frac{1}{n}})$ with parameters λ_1, λ_2, and λ_3.

Proof. We have shown in Theorem 1 how to compute the distribution of a product of random variables with exponential distributions. Thus, we compute $f_D(d) = \int_0^\infty \frac{1}{c} f_{D_1}(c) f_{D_2}(\frac{d}{c}) dc = \frac{2\lambda_2\lambda_3}{n} d^{\frac{1-n}{n}} K_0(2\sqrt{\lambda_2\lambda_3} d^{\frac{1}{2n}})$, where K_0 is the modified Bessel function of the second kind. It follows that

$$f_H(h) = \int_0^\infty \frac{1}{c} f_C(c) f_D\left(\frac{h}{c}\right) dc = \frac{2\lambda_1\lambda_2\lambda_3}{n^2} h^{\frac{1-n}{n}}$$
$$\int_0^\infty \frac{1}{c} \exp\left(-\lambda_1 c^{\frac{1}{n}}\right) K_0\left(2\sqrt{\lambda_2\lambda_3} h^{\frac{1}{2n}} c^{-\frac{1}{2n}}\right) dc.$$

When replacing $f_H(h)$ in the expression $P(\epsilon) = P(H \geq 1) = \int_1^\infty f_H(h) dh$, we obtain the result in Theorem 2. □

From Theorem 2, we can numerically show the influence of the probabilistic margins γ and γ_o on the probability of error $P(\epsilon)$. This is illustrated in Fig. 11.

9 Conclusion

We have addressed the problem of concurrent object detection and segmentation in images. This problem is formulated as maximizing the posterior distribution over object-class labels associated with image regions, forming a conditional random field. Regions with the same class label are classified as belonging to either a single object instance or two distinct instances. Our MAP inference of CRF is based on the Metropolis-Hastings algorithm. In general, MH is controlled by the two ratios of proposal and posterior distributions of states in a space of possible label assignments to image regions. Our key idea is to use a Hough Forest to estimate these two ratios in a nonparametric manner, directly from appearance, geometric, and spatial-layout properties of image regions. Iterative MH jumps fuse CRF and HF in a unified object representation termed HFRF.

Our empirical evaluation demonstrates superior object detection and segmentation results of HFRF relative to the state of the art. In addition, these results are obtained with faster convergence rate than by using a standard formulation of MH that requires a parametric estimation of the nominators and denominators of the two ratios of proposal and posterior distributions. HFRF outperforms a variant of our approach, called RFRF, which uses Random Forest instead of HF and ignores the information about annotated bounding boxes of objects in training, in terms of both average accuracy and computation times. The paper has also presented a theoretical analysis of HF and HFRF applied to a two-class object detection problem. Specifically, we have derived the theoretical upper bounds of classification error for HF and HFRF, and thus proved that these errors are bounded.

References

[1] L.-J. Li, R. Socher, and L. Fei-Fei, "Towards Total Scene Understanding: Classification, Annotation and Segmentation in an Automatic Framework," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2009.

[2] X. He, R.S. Zemel, and M.Á. Carreira-Perpiñán, "Multiscale Conditional Random Fields for Image Labeling," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 695-702, 2004.

[3] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context," *Int'l J. Computer Vision,* vol. 81, pp. 2-23, 2007.

[4] J. Verbeek and B. Triggs, "Scene Segmentation with CRFs Learned from Partially Labeled Images," *Proc. Advances Neural Information Processing Systems,* pp. 1553-1560, 2008.

[5] A.B. Torralba, K.P. Murphy, and W.T. Freeman, "Contextual Models for Object Detection Using Boosted Random Fields," *Proc. Advances Neural Information Processing Systems,* 2004.

[6] S. Gould, T. Gao, and D. Koller, "Region-Based Segmentation and Object Detection," *Proc. Advances Neural Information Processing Systems,* 2009.

[7] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie, "Objects in Context," *Proc. 11th IEEE Int'l Conf. Computer Vision,* 2007.

[8] N. Payet and S. Todorovic, "From a Set of Shapes to Object Discovery," *Proc. 11th European Conf. Computer Vision,* 2010.

[9] S. Todorovic and N. Ahuja, "Unsupervised Category Modeling, Recognition, and Segmentation in Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 12, pp. 1-17, Dec. 2008.

[10] J.J. Lim, P. Arbelaez, C. Gu, and J. Malik, "Context by Region Ancestry," *Proc. 12th IEEE Int'l Conf. Computer Vision,* 2009.

[11] J. Sivic, B.C. Russell, A. Zisserman, W.T. Freeman, and A.A. Efros, "Unsupervised Discovery of Visual Object Class Hierarchies," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2008.

[12] J. Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," *Proc. Int'l Conf. Machine Learning,* pp. 282-289, 2001.

[13] W.K. Hastings, "Monte Carlo Sampling Methods Using Markov Chains and Their Applications," *Biometrika,* vol. 57, no. 1, pp. 97-109, 1970.

[14] A. Barbu and S.-C. Zhu, "Generalizing Swendsen-Wang to Sampling Arbitrary Posterior Probabilities," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 8, pp. 1239-1253, Aug. 2005.

[15] J. Gall and V. Lempitsky, "Class-Specific Hough Forests for Object Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2009.

[16] J.M. Winn and J. Shotton, "The Layout Consistent Random Field for Recognizing and Segmenting Partially Occluded Objects," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 37-44, 2006.

[17] P.F. Felzenszwalb and D.P. Huttenlocher, "Efficient Belief Propagation for Early Vision," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 261-268, 2004.

[18] V. Kolmogorov, "Convergent Tree-Reweighted Message Passing for Energy Minimization," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 10, pp. 1568-1583, Oct. 2006.

[19] N. Komodakis and G. Tziritas, "Approximate Labeling Via Graph-Cuts Based on Linear Programming," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 8, pp. 1436-1453, Aug. 2007.

[20] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 6, pp. 1068-1080, June 2008.

[21] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques.* MIT Press, 2009.

[22] M.J. Wainwright, T.S. Jaakkola, and A.S. Willsky, "MAP Estimation via Agreement on Trees: Message-Passing and Linear Programming," *IEEE Trans. Information Theory,* vol. 51, no. 11, pp. 3697-3717, Oct. 2005.

[23] J. Pearl, *Probabilistic Reasoning in Intelligence Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers, 1988.

[24] V. Kolmogorov and R. Zabin, "What Energy Functions Can Be Minimized via Graph Cuts?" *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 2, pp. 147-159, Feb. 2004.

[25] E. Boros and P.L. Hammer, "Pseudo-Boolean Optimization," *Discrete Applied Math.,* vol. 123, pp. 155-225, 2002.

[26] Y. Boykov, O. Veksler, and R. Zabih, "Fast Approximate Energy Minimization via Graph Cuts," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 11, pp. 1222-1239, Nov. 2001.

[27] M. Szummer, P. Kohli, and D. Hoiem, "Learning CRFs Using Graph Cuts," *Proc. European Conf. Computer Vision,* pp. 582-595, 2008.

[28] N. Komodakis and N. Paragios, "Beyond Loose LP-Relaxations: Optimizing MRFs by Repairing Cycles," *Proc. European Conf. Computer Vision,* pp. 806-820, 2008.

[29] M.P. Kumar and P.H.S. Torr, "Efficiently Solving Convex Relaxations for MAP Estimation," *Proc. Int'l Conf. Machine Learning,* pp. 680-687, 2008.

[30] D. Sontag, T. Meltzer, A. Globerson, T. Jaakkola, and Y. Weiss, "Tightening LP Relaxations for MAP Using Message Passing," *Proc. 24th Ann. Conf. Uncertainty in Artificial Intelligence,* 2008.

[31] J.S. Yedidia, W.T. Freeman, and Y. Weiss, *Understanding Belief Propagation and Its Generalizations,* pp. 239-269. Morgan Kaufmann Publishers, Inc., 2003.

[32] M.J. Beal, "Variational Algorithms for Approximate Bayesian Inference," PhD dissertation, Gatsby Computational Neuroscience Unit, Univ. College London, 2003.

[33] D.P. Bertsekas, *Nonlinear Programming,* second ed. Athena Scientific, Sept. 1999.

[34] N. Komodakis, G. Tziritas, and N. Paragios, "Performance vs Computational Efficiency for Optimizing Single and Dynamic MRFs: Setting the State of the Art with Primal-Dual Strategies," *Computer Visual Image Understanding,* vol. 112, no. 1, pp. 14-29, 2008.

[35] M.P. Kumar and D. Koller, "Efficiently Selecting Regions for Scene Understanding," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2010.

[36] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer, "Optimizing Binary MRFs via Extended Roof Duality," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2007.

[37] D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng, "Discriminative Learning of Markov Random Fields for Segmentation of 3D Scan Data," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 169-176, 2005.

[38] S. Kumar, J. August, and M. Hebert, "Exploiting Inference for Approximate Parameter Learning in Discriminative Fields: An Empirical Study," *Proc. Energy Minimization Methods in Computer Vision and Pattern Recognition,* 2005.

[39] L. Zhang and S.M. Seitz, "Parameter Estimation for MRF Stereo," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2005.

[40] C. Desai, D. Ramanan, and C. Fowlkes, "Discriminative Models for Multi-Class Object Layout," *Proc. 12th IEEE Int'l Conf. Computer Vision,* 2009.

[41] R. Marée, P. Geurts, J. Piater, and L. Wehenkel, "Random Subwindows for Robust Image Classification," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 34-40, 2005.

[42] F. Moosmann, B. Triggs, and F. Jurie, "Fast Discriminative Visual Codebooks Using Randomized Clustering Forests," *Proc. Neural Information Processing Systems,* pp. 985-992, 2007.

[43] F. Schroff, A. Criminisi, and A. Zisserman, "Object Class Segmentation Using Random Forests," *Proc. British Machine Vision Conf.,* 2008.

[44] A. Bosch, A. Zisserman, and X. Munoz, "Image Classification Using Random Forests and Ferns," *Proc. 11th IEEE Int'l Conf. Computer Vision Conf.,* 2007.

[45] J. Shotton, M. Johnson, and R. Cipolla, "Semantic Texton Forests for Image Categorization and Segmentation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2008.

[46] S. Nowozin, C. Rother, S. Bagon, T. Sharp, B. Yao, and P. Kholi, "Decision Tree Fields," *Proc. 11th IEEE Int'l Conf. Computer Vision Conf.,* Nov. 2011.

[47] G. Martinez, W. Zhang, N. Payet, S. Todorovic, N. Larios, A. Yamamuro, D. Lytle, A. Moldenke, E. Mortensen, R. Paasch, L. Shapiro, and T. Dietterich, "Dictionary-Free Categorization of Very Similar Objects via Stacked Evidence Trees," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2009.

[48] N. Payet and S. Todorovic, "$(\mathrm{RF})^2$—Random Forest Random Field," *Proc. Neural Information Processing Systems,* 2010.

[49] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "From Contours to Regions: An Empirical Evaluation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2009.

[50] L. Breiman, "Random Forests," *Machine Learning,* vol. 45, no. 1, pp. 5-32, 2001.

[51] S. Bileschi and L. Wolf, "A Unified System for Object Detection, Texture Recognition, and Context Analysis Based on the Standard Model Feature Set," *Proc. British Machine Vision Conf.,* 2005.

[52] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes," www.pascal-network.org/challenges/VOC/, 2012.

[53] F. Li, J. Carreira, and C. Sminchisescu, "Object Recognition as Ranking Holistic Figure-Ground Hypotheses," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2010.

[54] C. Russell, P.H.S. Torr, and P. Kohli, "Associative Hierarchical CRFs for Object Class Image Segmentation," *Proc. 12th IEEE Int'l Conf. Computer Vision,* 2009.

[55] J. Gonfaus, X. Boix, J. van de Weijer, A. Bagdanov, J. Serrat, and J. Gonzalez, "Harmony Potentials for Joint Classification and Segmentation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 3280-3287, 2010.

[56] S. Gould, R. Fulton, and D. Koller, "Decomposing a Scene into Geometric and Semantically Consistent Regions," *Proc. 12th IEEE Int'l Conf. Computer Vision,* pp. 1-8, 2009.

[57] C. Galleguillos, B. McFee, S. Belongie, and G.R.G. Lanckriet, "Multi-Class Object Localization by Combining Local Contextual Interactions," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2010.

[58] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 32, no. 9, pp. 1627-1645, Sept. 2010.

[59] L. Zhu, Y. Chen, A.L. Yuille, and W.T. Freeman, "Latent Hierarchical Structural Learning for Object Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2010.

[60] M. Pedersoli, A. Vedaldi, and J. Gonzàlez, "A Coarse-to-Fine Approach for Fast Deformable Object Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2011.

[61] N.N. Schraudolph, J. Yu, and S. Günter, "A Stochastic Quasi-Newton Method for Online Convex Optimization," *Proc. Int'l Conf. Artificial Intelligence and Statistics,* vol. 2, pp. 436-443, 2007.

[62] Y. Lin and Y. Jeon, "Random Forests and Adaptive Nearest Neighbors," *J. Am. Statistical Assoc.,* vol. 101, pp. 578-590, 2006.

[63] I.S. Gradshteyn and I.M. Ryzhik, *Table of Integrals, Series and Products,* fifth ed. Academic Press, Inc., 2007.

[64] Z.A. Lomnicki, "On the Distribution of Products of Random Variables," *J. Royal Statistical Soc., Series B (Methodological),* vol. 29, no. 3, pp. 513-524, 1967.

Nadia Payet received the MS degree in electrical and computer engineering from ESCPE Lyon, France, in 2005 and the PhD degree in computer science from Oregon State University in 2011. She is currently a researcher at Amazon. Her research interests include computer vision, machine learning, and computer graphics. She is a student member of the IEEE.

Sinisa Todorovic received the PhD degree in electrical and computer engineering from the University of Florida in 2005. He is an assistant professor in the School of Electrical Engineering and Computer Science at Oregon State University. He was a postdoctoral research associate in the Beckman Institute at the University of Illinois at Urbana-Champaign, between 2005 and 2008. His research interests include computer vision and machine learning problems. He is a member of the IEEE.