Learning Dynamic Hybrid Markov Random Field for Image Labeling

Quan Zhou, Jun Zhu, and Wenyu Liu, *Member, IEEE*

***Abstract*— Using shape information has gained increasing concerns in the task of image labeling. In this paper, we present a dynamic hybrid Markov random field (DHMRF), which explicitly captures middle-level object shape and low-level visual appearance (e.g., texture and color) for image labeling. Each node in DHMRF is described by either a deformable template or an appearance model as visual prototype. On the other hand, the edges encode two types of intersections: co-occurrence and spatial layered context, with respect to the labels and prototypes of connected nodes. To learn the DHMRF model, an iterative algorithm is designed to automatically select the most informative features and estimate model parameters. The algorithm achieves high computational efficiency since a branch-and-bound schema is introduced to estimate model parameters. Compared with previous methods, which usually employ implicit shape cues, our DHMRF model seamlessly integrates color, texture, and shape cues to inference labeling output, and thus produces more accurate and reliable results. Extensive experiments validate its superiority over other state-of-the-art methods in terms of recognition accuracy and implementation efficiency on: 1) the MSRC 21-class dataset, and 2) the lotus hill institute 15-class dataset.**

***Index Terms*— Classification, feature selection, image labeling, image segmentation, Markov random field (MRF).**

I. Introduction

IMAGE labeling plays an important role in object recognition and scene understanding [1]–[3]. It offers a framework to assign a semantic label to each pixel for object recognition and segmentation task. However, great variation of different object classes makes it as an extremely challenge task in computer vision. As shown in Fig. 1, an image is typically associated with two types of object categories: *structured objects* with rigid or deformable shapes, e.g., "cow", "horse" and "car", and *generic objects* without *explicit* shape patterns, e.g., "sky", "tree" and "water". Besides, different objects are usually characterized with heterogeneous features. For example, it is observed that "horse" has distinct texture and shape, yet its color is often undistinguishable from the backgrounds; "grass" and "tree" always have similar texture and color; "sky" can be easily recognized by its geometric attributes, yet it is often confused with "water"; "car" has rigid shape outline, but does not have common texture or color; even within one class, the articulated "cow" has different poses, directions and scales.

Fig. 1. Nature images.

To overcome these visual variations, researchers have built up many successful vision systems on image labeling. Most of these approaches utilize the bottom-up appearance cues [2], [4], [5] and the top-down information [3], [6]–[9]. Recently, the graphical models like Markov random fields (MRFs) [10], [11] and conditional random fields (CRFs) [12], [13], have been widely used to capture the contextual information to improve the performance of labeling results. Many learning algorithms, such as discriminative learning [14], [15] and generative learning [16], [17], have been proposed to learn model parameters. However, these models and algorithms have similar disadvantages:

1) Despite the success of exploring local features [18] or *implicit* shape cues [19], most previous methods are still limited to describe object with *explicit* shape model to improve the performance of image labeling.
2) Traditional graphical models usually have fixed graph structure in labeling literatures [1], [20], [21], which may be inflexible enough to represent pairwise contextual relationships among image elements.
3) For each class, how to effectively learn different features within unified principle remains unknown. The learning algorithm is subject to automatically select most discriminative features, and to compose the selected features into a well normalized probability model.

To address these problems, we present a dynamic hybrid Markov random field (DHMRF) model in this paper. This model *explicitly* captures the middle-level object shape and low-level visual appearance, which is robust to intra-class variance and inter-class similarities. No matter what structured or generic object is, we project image regions into three types

Manuscript received November 28, 2011; revised April 23, 2012; accepted January 10, 2013. Date of publication April 1, 2013; date of current version March 29, 2013. This work was supported in part by the National Natural Science Foundation of China under Grant 61173120, and Grant 61271226. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Arun A. Ross.

Q. Zhou and W. Liu are with the Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: qzhou.lhi@gmail.com; liuwy@mail.hust.edu.cn).

J. Zhu is with the Department of Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: zhujun.lhi@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2013.2246519

of feature descriptors, called primitive, texture and color, and then calculate their responses as features. These features are used to build up likelihood terms in our DHMRF model with two types of prototypes: 1) appearance model composed by a series of histograms for generic object, and 2) deformable template consisted of a set of Gabor wavelets for structured object [22]. For the structured objects with large shape variation, we learn multiple templates through an EM clustering procedure. Embedding *explicit* deformable template not only defines shape mask in guiding more precise segmentation, but also provides consistent labels for associated pixels.

Moreover, the structure of DHMRF model is dynamic since the estimation of deformable templates is driven by [23], that adapts to some particular instances based on the responses of deformable part-based detectors. The dynamic structure allows us more flexible to incorporate spatial layered context and co-occurrence with respect to the labels and prototypes of image elements. As a result, our model can produce more consistent labeling results to significantly improve the performance both on the tasks of image labeling and object detection.

To learn visual prototypes under an unified framework, we design an efficient iterative algorithm based on information projection principle [16]. By collecting marginal statistics of the projected features over the training examples, we start from an initialized reference model to learn a sequence of probability models so as to minimize the Kullback–Leibler divergence (KLD) with respect to the underlying distribution. At each iteration, we choose most discriminative feature, which leads to the maximum reduction of the KLD. Then the corresponding model parameters are estimated based on a branch-and-bound scheme. The algorithm iterates until the gain of induced feature is less than a given threshold. This information projection schema allows us to learn our likelihood model with desirable computational efficiency. Particularly, the branch-and-bound scheme theoretically guarantees that our algorithm converges to the optimal model parameters. The contributions of this paper are mainly summarized as follows:

1) We propose a DHMRF model for image labeling. Compared with existing methods, it seamlessly integrates two types of visual prototypes, namely *deformable templates* and *appearance models*, into an unified inference procedure, which is robust to visual variance.
2) Dynamic structure of our DHMRF model allows us more flexible to encode top-down contextual cues (e.g., spatial layered context and co-occurrence), and thus benefits both for the tasks of image labeling and object detection.
3) We establish an iterative algorithm to learn visual prototypes. By employing a branch-and-bound schema, our algorithm converges to optimal model parameters, and is also computationally efficient.

After a brief discussion of related work in Section II, we describe our DHMRF model in Section III, and entail the learning algorithm in Section IV. Section V briefly introduces the inference algorithm to obtain the final labeling outputs. Experimental results are given in Section VI. Finally, we conclude this paper in VII.

II. Related Work

We discuss related work in two areas of computer vision: probabilistic models and statistic learning for image labeling.

1) Probabilistic Models for Image Labeling: There has been a lot of work proposed for jointly modeling recognition and segmentation, which fuse the bottom-up low-level image statistics and top-down high-level contextual cues in probabilistic models [1]–[3], [19], [24]–[26]. He *et al*. [20] attempt to infer an environment-specific prior to guide segmentation; Hoiem *et al*. [27] present a system combining the interaction between different objects in a loop as mutual support; Gould *et al*. [21] and Belongie [28] prefer to employ the relative location prior and co-occurrence as a contextual feature in their probabilistic construction. Others investigate the contextual information in hierarchical graphic models [29], [30]. However, our DHMRF model differs from previous efforts in several aspects: 1) it is more robust to visual variance by simultaneously modeling generic and structured objects; 2) the *explicit* shape model is benefit for segmentation and consistent labeling; and 3) layered context is more representative than co-occurrence and relative location.

Comparing to traditional MRFs [10], [11], [31], [32] and CRFs [12], [13], our DHMRF model is not limited to a fixed graphic structure. The label/vertex of each region have support from local appearance or object detections [23]. In terms of shape prior, our approach is similar to [9] which uses a part-based model to define a shape mask. However, we focus on incorporating deformable template as a whole shape representation. Moreover, we argue that the layered context is significant in case of segmenting images containing multi-instances drawn from structured objects, which is neglect in [9]. Three approaches directly related to our approach are [6], [7] and [8]. The first two methods bias a hierarchical CRF model using object detection windows across multiple categories, while the final generates segmentations using part-specific shape models. Our work combines appearance models for generic objects and deformable templates for structured objects, using a DHMRF model to construct a globally consistent pixel-level labeling system of an image.

2) Statistic Learning for Image Labeling: In the literature, the learning algorithm has been studied in two families of statistical models: *discriminative* and *generative* learning.

The most *discriminative* learning methods directly target on the posterior distribution in a supervised manner [1], [2], [33]–[35]. Typically, the boosting algorithms, such as regression boost [15] and adaboost [36], automatically select a weak classifier at each iteration, which maximally reduces the empirical error over the weighted training examples. On the other hand, Support vector machine (SVM) [14] selects a linear hyperplane in the feature space that maximizes the margin between the positive and negative samples.

Unlike the *discriminative* learning methods, the *generative* learning approaches iteratively find the most informative feature, where the current model achieves the maximum entropy [37]. Then this feature is updated into the potential function so that the new model can observe the underlying statistics. Due to the high correlation of feature

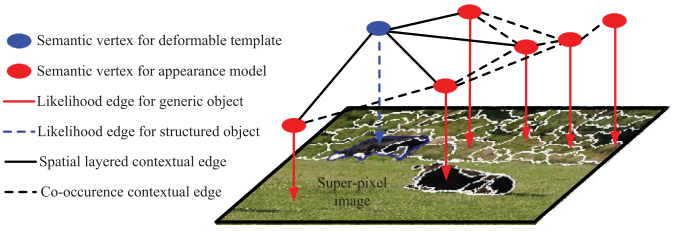


Fig. 2. Illumination of our DHMRF model. Each super-pixel is split by white boundary. Note the highlight blue boundary of deformable "cow" is used to distinguish with super-pixels. (Best viewed in color.)

statistics, the parameters/weights of all previously selected features are required to be updated again when a new feature is selected [16], [37]. Conversely, we carefully de-correlate the feature responses by independence assumption. With this assumption, the normalizing constant can be computed analytically and we do not have to update all parameters at each iterative learning procedure. Thus our learning algorithm achieves high computational efficiency than the traditional MRF/Gibbs learning methods [10], [16], [37]. In addition, we do not have to re-weight the examples in the training set as well as *discriminative* learning algorithm does.

III. Graph Formulation for Image Labeling

The goal of our labeling system is to partition image I into $\mathcal{K}$ disjoint sub-regions υ_r and assign each region a semantic label, such as "grass", "car" and "dog". However, the structured objects may exhibit large shape variations, e.g., the crouching "cow" shown in Fig. 2 is more appropriately described by an appearance model rather than a deformable template. Denote $\ell \in \mathcal{L}$ be a label for one region with a potential ρ to further describe its visual prototype. Then $\mathcal{L}$ is an expanded set of labels, which can be divided into two sets: i) Str used to encode object shape with the prototypes of deformable templates T_ℓ, and ii) Rgn used to represent the prototypes of appearance models $\mathbf{h}_\ell$ for generic classes.

Let W denote the final labeling output that considers both recognition accuracy and contextual constrains with image observation I. We thus have

$$W = (\{\upsilon_r, \ell_r, \rho_r\}_{r=1}^{\mathcal{K}}) \tag{1}$$

where $\mathcal{K}$ is a variable to be determined in inference process.

As shown in Fig. 2, an input image I is first over-segmented into a set of super-pixels using MeanShift technique [38]. Our DHMRF model can be equivalently represented by a graph formulation, in which image I and contextual interaction are encoded by an undirected graph $G = \langle V, E\rangle$ based on these super-pixels. More specifically, the vertex set $V = \{V_S, V_R\}$ is composed of two components: the set of semantic vertices V_S, and the set of image region vertices V_R. From Fig. 2, we can see that each $\upsilon_r \in V_R$ corresponds to a super-pixel, or an image region composed by several super-pixels. Each region υ_r is also associated with a semantic vertex $\upsilon_s \in V_S$ which has two attributes: $\ell \in \mathcal{L}$ indicating its potential label and ρ denoting its prototype. The prototype ρ can be further explained as a deformable template T_ℓ (denoted by blue circle) or an appearance model $\mathbf{h}_\ell$ (denoted by red circle). The edge

Fig. 3. Gabor wavelets with different orientations.

set $E = \{\boldsymbol{\varepsilon}_L, \boldsymbol{\varepsilon}_C\}$ also consists of two parts: the data likelihood edges $\boldsymbol{\varepsilon}_L = \{(\upsilon_r, \upsilon_s)|\upsilon_r \in V_R, \upsilon_s \in V_S\}$ (solid red line for generic classes and dash blue line for structured objects), and the contextual edges $\boldsymbol{\varepsilon}_C = \{(\upsilon_p, \upsilon_q)|(\upsilon_p, \upsilon_q) \in V_S$ and $(p, q) \in \mathcal{N}\}$. Unlike the traditional MRFs [10], [11], [31] that build their 4-connected or 8-connected neighborhood on a rigid image lattice, we construct our neighborhood system $\mathcal{N}$ reflecting the pairwise dependency between semantic vertices of two adjacent image regions (solid line for spatial layered context and dash line for co-occurrence).

In our graphical formulation, the posterior probability of the labeling output W given observed image I is equivalent to the following Gibbs energy:

$$E(W|I;\boldsymbol{\Theta}) = E_L(I|W;\boldsymbol{\Theta}_{\text{lik}}) + E_C(W;\boldsymbol{\Theta}_{\text{coh}}) - \log Z(\boldsymbol{\Theta}, I) \tag{2}$$

where $Z(\boldsymbol{\Theta}, I) = \sum_W \exp\{E_L(I|W;\boldsymbol{\Theta}_{\text{lik}}) + E_C(W;\boldsymbol{\Theta}_{\text{coh}})\}$ is the partition function which normalizes the distribution by marginalizing all possible configurations of labeling output W. $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}_{\text{lik}}, \boldsymbol{\Theta}_{\text{coh}}\}$ represents parameters involved in likelihood and coherent models. $E_L(I|W;\boldsymbol{\Theta}_{\text{lik}})$ is the likelihood energy function to evaluate the goodness of assigning label ℓ and prototype ρ to each image region. $E_C(W;\boldsymbol{\Theta}_{\text{coh}})$ is the coherence energy function encoding the prior of label-based configuration among each element defined in (1). In the following, we entail these two energy terms elaborately.

A. Likelihood Energy $E_L(I|W;\boldsymbol{\Theta}_{\text{lik}})$

Since there is no overlapping between arbitrary two regions and each region is only represented by one prototype, the likelihood models $E_L(I|W;\boldsymbol{\Theta}_{\text{lik}})$ are assumed to be conditionally independent on W and υ_r. Therefore, we have

$$\begin{aligned} E_L(I|W;\boldsymbol{\Theta}_{\text{lik}}) = &\sum_{\substack{(\upsilon_r,\upsilon_p)\in\varepsilon_L \\ \rho_r = T_{\ell_r}}} E_L(I_{\upsilon_r}|\ell_r, T_{\ell_r};\boldsymbol{\theta}_r) \\ &+ \sum_{\substack{(\upsilon_r,\upsilon_p)\in\varepsilon_L \\ \rho_r = \mathbf{h}_{\ell_r}}} E_L(I_{\upsilon_r}|\ell_r, \mathbf{h}_{\ell_r};\boldsymbol{\theta}_r) \end{aligned} \tag{3}$$

where $\boldsymbol{\Theta}_{\text{lik}} = \{\boldsymbol{\theta}_r\}_{r=1}^{\mathcal{K}}$ depends on each image element υ_r via the partition label ℓ_r and prototype ρ_r.

1) Image Representation and Prototypes: Let Δ be a filter bank dictionary which includes a set of Gabor wavelets B_k indexed by $(x, y, \sigma_x, \sigma_y, o)$. The Gabor wavelet of the canonical scale and orientation is defined as:

$$F(x, y) \propto \exp\{-(x/\sigma_x^2) - (y/\sigma_y^2)\}e^{i(x\cos o + y\sin o)} \tag{4}$$

with paramters $\sigma_x^2 = 5, \sigma_y^2 = 10$ and $o = 15$ following [22]. These Gabor wavelets are illustrated in Fig. 3.

For $\ell \in Str$, the deformable template T_ℓ defines an image subspace $\Omega(T_\ell)$ through the sparse coding model with s

number of Gabor wavelets $B_{k,\ell}$ (called image primitives)

$$\Omega(T_\ell) = \left\{ I : I = \sum_{k=1}^{s} c_{k,\ell} B_{k,\ell} + U \right\} \tag{5}$$

where $c_{k,\ell}$ are coefficients and U is the residual image.

For $\ell \in Rgn$, the appearance model $\mathbf{h}_\ell$ defines an image subspace $\Omega(\mathbf{h}_\ell)$ through an implicit function which consists of a series of g histograms

$$\Omega(\mathbf{h}_\ell) = \{ I : \mathbf{H} = \mathbf{h}_\ell + \epsilon \} \tag{6}$$

where $\mathbf{h}_\ell = (h_{1,\ell}, \ldots, h_{g,\ell})$, in which each element $h_{k,\ell}$, encodes the texture or color cues, and ϵ is the residual.

The usage of Gabor wavelets and histograms is a robust description for object shape and category appearance, which makes our likelihood model is invariant to visual variability of scales, orientations, rotations and translations.

2) Image Features: Intuitively, we may project $\Omega(T_\ell)$ and $\Omega(\mathbf{h}_\ell)$ into feature space with different metrics, on which we can calculate the statistic features over the training set. In the following, we discuss the details of computing the distances for the primitives $B_{k,\ell}$ and histograms $h_{k,\ell}$.

a) Calculating distance on primitives: For $B_{k,\ell} \in \Delta$ and $I \in \Omega(T_\ell)$, the distance is calculated as the local maximum over the activity $\delta_{k,\ell} = (\delta_{x,k,\ell}, \delta_{y,k,\ell}, \delta_{o,k,\ell})$ by slightly perturbing its locations and orientations

$$d^{prm}(I, B_{k,\ell}) = \max_{\delta_{k,\ell}} ||I - c_{k,\ell} B_{k,\ell}(\delta_{k,\ell})||^2 \tag{7}$$

b) Calculating distance on texture: In contrast to the primitives, texture region usually contains many small elements which represent texture cues in different directions. Within k^{th} direction, we pool a histogram of filter responses over local region to form texture descriptor $H_k^{\text{txt}}(I)$. Then the distance for texture gradient is defined using histogram intersection kernel (HIK) [39]

$$d^{txt}(I, h_{k,\ell}) = \sum_{b=1}^{\mathbf{b}} \min(H_k^{\text{txt}}(I)[b], h_{k,\ell}[b]) \tag{8}$$

where $\mathbf{b}$ is the number of bins of the histogram. $h_{k,\ell}$ ($k \in \{1, 2, \ldots, 15\}$) is pre-computed histogram by averaging the histograms over all positive example with label ℓ. For the k^{th} filter $B_k \in \Delta$, the histogram $H_k^{\text{txt}}(I)$ is calculated by summing over all pixels in image region I for a number of $\mathbf{b}$ bins, in which the b^{th} bin is indexed by range $[\mathcal{A}_b, \mathcal{B}_b]$

$$H_k^{\text{txt}}(I)[b] = \frac{1}{|I|} \sum_{(x,y)\in I} 1(\mathcal{A}_b < |B_k * I|^2 < \mathcal{B}_b) \tag{9}$$

where $B_k * I$ denotes the convolution between filter B_k and image regions I, and $1(\cdot) \in \{0, 1\}$ is an indicator function.

c) Calculating distance on color: Similar to texture, we calculate a histogram $H_k^{clr}(I)$ on the RGB color space for the rest elements of $\mathbf{h}_\ell$. With the similar definition of (8), the distance $d^{clr}(I, h_{k,\ell}), k = \{16, 17, 18\}$ is defined between the color histogram $H_k^{clr}(I)$ of the observed image and the prototype histogram $h_{k,\ell}$.

Taking into account the statistical fluctuations of different image distance, we use a sigmoid function (denoted by $\text{Sig}(\cdot)$) to compute image features through the transformation of the measured distance

$$r_k = \text{Sig}(d(I, \cdot)) = \tau \times \left(\frac{2}{1 + e^{-2[\eta - d(I,\cdot)]/\tau}} - 1 \right) \tag{10}$$

with parameter $\tau = 6$ controls the upper bound and $\eta = 2$ controls the translation as well as [22] does.

Either for T_ℓ or $\mathbf{h}_\ell$, each prototype is represented by a set of features $\{r_k\}$. Each r_k is a soft measure to describe whether the associated image regions belong to the subspace defined by T_ℓ or $\mathbf{h}_\ell$. We will discuss the form of $E_L(I|\ell, \rho; \mathbf{\Theta}_{\text{lik}})$ and estimate model parameters in Section IV-B.

B. Coherence Energy $E_C(W; \mathbf{\Theta}_{\text{coh}})$

The coherence energy function $E_C(W; \mathbf{\Theta}_{\text{coh}})$ encodes the compatibilities and interaction among the elements in W from the segmentation and labeling perspectives. It can be decomposed into two components: the co-occurrence energy $E_C^{co}(W; \boldsymbol{\theta}^{co})$ and layered context energy $E_C^{ly}(W; \boldsymbol{\theta}^{ly})$

$$E_C(W; \mathbf{\Theta}_{\text{coh}}) = E_C^{co}(W; \boldsymbol{\theta}^{co}) + E_C^{ly}(W; \boldsymbol{\theta}^{ly}) \tag{11}$$

where $\mathbf{\Theta}_{\text{coh}} = \{\boldsymbol{\theta}^{co}, \boldsymbol{\theta}^{ly}\}$ denotes the parameters involved in $E_C^{co}(W; \boldsymbol{\theta}^{co})$ and $E_C^{ly}(W; \boldsymbol{\theta}^{ly})$, which are used to evaluate the consistency between two adjacent labels with respective to their prototypes.

1) Co-occurrence Energy: For two adjacent regions υ_r, υ_r' with label ℓ_r, ℓ_r' and prototype ρ_r, ρ_r', the co-occurrence energy is defined following [28], [40]

$$E_C^{co}(W; \boldsymbol{\theta}^{co}) = E_C^{co}(\ell_r, \ell_r' | [\rho_r = \rho_r']; \boldsymbol{\theta}^{co}) = \theta^{co}_{(\ell_r, \ell_r')} \mathbf{1}(\ell_r, \ell_r') \tag{12}$$

where $\theta^{co}_{(\ell_r,\ell_r')} \in \boldsymbol{\theta}^{co}$ is the parameter for co-occurrence, $[\rho_r = \rho_r']$ denotes υ_r and υ_r' have same type of prototypes. $\mathbf{1}(\ell_r, \ell_r')$ is zero-one indicator function and $\mathbf{1}(\ell_r, \ell_r') = 1$ if ℓ_r and ℓ_r' appear simultaneously, otherwise, $\mathbf{1}(\ell_r, \ell_r') = 0$.

2) Layered Context Energy: Similarly, we define the layered context energy of two adjacent regions υ_r and υ_r' as

$$E_C^{ly}(W; \boldsymbol{\theta}^{ly}) = E_C^{ly}(\ell_r, \ell_r' | [\rho_r \neq \rho_r']; \boldsymbol{\theta}^{ly}) = \theta^{ly}_{(\ell_r, \ell_r')} \mathbf{1}(\ell_r, \ell_r') \tag{13}$$

where $\theta^{ly}_{(\ell_r,\ell_r')} \in \boldsymbol{\theta}^{ly}$ is the parameter for layered context, $[\rho_r \neq \rho_r']$ denotes υ_r, υ_r' have different type of prototypes.

Immediately below, we will present the algorithm to learn likelihood terms and model parameters $\mathbf{\Theta} = \{\mathbf{\Theta}_{\text{lik}}, \mathbf{\Theta}_{\text{coh}}\}$.

IV. Model Learning

In this section, we first briefly describe the method to learn coherent parameters $\mathbf{\Theta}_{\text{coh}}$, and then introduce an iterative algorithm for learning likelihood energy $E_L(I|\ell, \rho; \mathbf{\Theta}_{\text{lik}})$.

A. Learning for Coherence Energy

Ideally, the parameters $\mathbf{\Theta}_{\text{coh}}$ should be learned using maximum likelihood estimation (MLE) [41], which needs to evaluate the partition function. Exact computation, however, of partition function is intractable, since it requires marginalization on all possible configurations of W. In principle, it can be

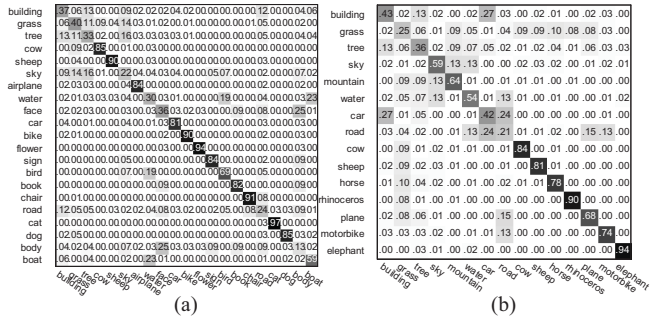


Fig. 4. Learned frequency matrices on (a) MSRC 21-class and (b) LHI 15-class datasets. Note these two matrices are symmetric, nonnegative, and each row is normalized to have summation 1.

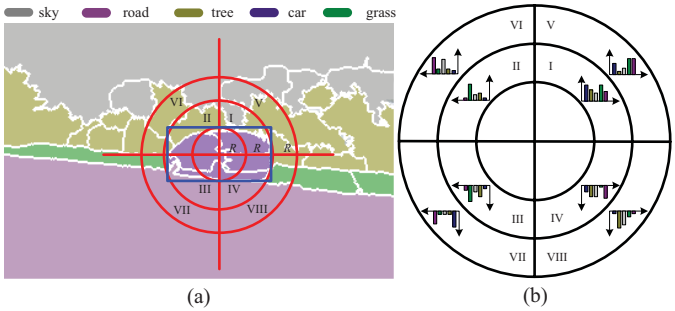


Fig. 5. Characterizing the spatial layered context. (a) The "car" instance is bounded by a blue rectangle, and red circles are used to form sectors. Note each sector is represented by a Roman numeral. (b) We illustrate the frequency distribution of each generic category within different sectors.

approximated by Markov Chain Monte Carlo (MCMC) sampling technique [10], [28], [40] or coding method [31], [32]. However, the former is prohibitively impractical in computation, while the later may converge to local minimum [32] for image labeling. Alternatively, we turn to a pragmatic approach using empirical frequency to estimate coherent parameters, since it achieves high computational efficiency and remains remarkable performance as shown in Section VI.

1) Learning Co-occurrence Energy: The co-occurrence explores the frequency that two semantic labels ℓ_r and ℓ'_r appear together in training images. We thus compute the parameter $\theta^{co}_{(\ell_r,\ell'_r)}$ in (12) as,

$$\theta^{co}_{(\ell_r,\ell'_r)} = \ln[\mathcal{H}(\ell_r,\ell'_r) + \omega_{co}] \tag{14}$$

where ω_{co} is the prior offset parameter to avoid overfitting. An entry $h(\ell_r,\ell'_r) \in \mathcal{H}(\ell_r,\ell'_r)$ counts the frequency that an object with label ℓ_r appears in a training image with an object with label ℓ'_r. The diagonal entries correspond to the frequency of the object in the training set. Fig. 4 shows the learned $\mathcal{H}(\ell_r,\ell'_r)$ matrices on MSRC 21-class and LHI 15-class dataset, respectively.

2) Learning Layered Context Energy: As shown in Fig. 5, unlike co-occurrence energy that addresses the joint appearance of object categories, the layered context energy encodes the *relative* co-occurrence between semantic labels with respect to relative location. In Fig. 5(a), suppose an instance with $\ell_r \in Str$ (e.g., rigid "car") is enclosed by a blue rectangle $B_{W\times H}$. If we use the center location (x_r, y_r) of $B_{W\times H}$ as the origin point, then the image plane is divided into $\mathcal{O}$ sectors by four quadrants and three circles with radius $R, 2R$ and $3R$, where $R = \frac{1}{2}\min\{W, H\}$. Then $\theta^{ly}_{(\ell_r,\ell'_r)}$ in (13) is computed as

$$\theta^{ly}_{(\ell_r,\ell'_r)} = \ln \sum_{o=1}^{\mathcal{O}} \frac{|v'_r \cap A_o|}{|A_o|}[\mathcal{H}(\ell'_r|\ell_r; A_o) + \omega_{ly}] \tag{15}$$

where A_o is the o^{th} sector and ω_{ly} is also the offset parameter as well as ω_{co} does. $\mathcal{H}(\ell'_r|\ell_r; A_o)$ is estimated similar with $\mathcal{H}(\ell_r,\ell'_r)$, but now we measure the spatial layered context with respect to the given structure object ℓ_r and its surrounding generic objects ℓ'_r in o^{th} sector (as shown in Fig. 5(b)).

B. Learning for Likelihood Energy

As the likelihood energy has been decomposed on each image entity $E_L(I_{v_r}|\ell_r,\rho_r;\boldsymbol{\theta}_r)$, they are usually easy to learn from training data. Our learning algorithm selects the most informative features, calculates the prototypes, derives the model forms, and estimates model parameters. From the perspective of DHMRF model, the likelihood energy is equivalent to the probability model $p_L(I_{v_r}|\ell_r,\rho_r;\boldsymbol{\theta}_r)$. We use p to represent it for notation simplicity.

Let f be the underlying probability distribution for a specific label ℓ, our objective is to learn a series of models that approach f from an initial or reference model q (which only need to be specified implicitly in our learning procedure). This procedure sequentially adds the most discriminative feature and matches the observed marginal statistics collected from the samples of f. With more marginal statistics matched between the model p and f, p will approach f in terms of reducing the KLD $KL(f||p)$ monotonically.

1) Training Data: For a given $\ell \in \mathcal{L}$, let $D^+ = \{I_1, \ldots, I_N\}$ be a set of positive examples sampled from f.

i) For learning T_ℓ ($\ell \in Str$), positive examples are roughly aligned after global transformations (e.g., translation, rotation, dilation and constriction), and can be represented by a common shape prototype.

ii) For learning $\mathbf{h}_\ell$ ($\ell \in Rgn$), positive examples refer to the over-segmented super-pixels with label ℓ in training set.

Let $D^- = \{J_1, \ldots, J_M\}$ be a set of negative examples collected from a reference model q. In [22], authors randomly crop image patches from nature images to produce D^-. On the contrary, we generate D^- by collecting the positive example from other categories for discriminative purpose, neither for learning T_ℓ nor $\mathbf{h}_\ell$.

2) Overview of Learning Procedure: Supposing one prototype can be represented by a feature candidate $\Omega^\ell_{\text{cand}} = \{r_{1,\ell}, r_{2,\ell}, \ldots, r_{K,\ell}\}$ ($K = s$ for T_ℓ and $K = g$ for $\mathbf{h}_\ell$), from which we will select the most discriminative features sequentially to form our likelihood model.

By induction, at the k^{th} iteration, we have a prototype with $k-1$ selected features and a model $p = p_{k-1}$. For current feature $r_{k,\ell} \in \Omega^\ell_{\text{cand}}$, we also calculate the feature responses over the N positive examples and the M negative examples

$$\{r^+_{k,\ell,n}, n = 1, \ldots, N\}; \{r^-_{k,\ell,m}, m = 1, \ldots, M\} \tag{16}$$

In following, the subscript ℓ is omitted for notation simplicity, and let $\bar{r}_k^+$ be the sample means over the positive examples.

The gain of inducing feature r_k to the prototype is measured by the KLD between the target marginal distribution $f(r_k)$ and current model $p_{k-1}(r_k)$, as this represents the new information in the training data that is not yet captured in the current model. Among all the candidate features that are not selected from $\Omega^{\ell}_{\text{cand}}$, the one with the largest gain is selected. Thereafter, the parameters are estimated to form new model p_k, that matches the observed marginal statistics.

To estimate model parameters, we need samples from $f(r_k)$ and $p_{k-1}(r_k)$. Obviously, $\{r^+_{k,n}\}$ are fair samples from $f(r_k)$. To get samples from $p_{k-1}(r_k)$, one may generate a number of synthesized images from model p_{k-1} [16], or use importance sampling by re-weighting the samples of $\frac{p_{k-1}(r_k)}{q(r_k)}$. In our learning paradigm, we simplify this problem by employing an independent assumption that a feature r_k is roughly uncorrelated with $r_{k'}$ if one of the following conditions holds:

1) Two image patches I_k and $I_{k'}$, which are ready to compute r_k and $r_{k'}$, have little overlapping;
2) r_k and $r_{k'}$ are feature responses from different filters.

As a result, $p_{k-1}(r_k) = q(r_k)$ and $\{r^-_{k,m}\}$ can be used as samples for $p_{k-1}(r_k)$.

3) Log-Linear Form for Likelihood Model: Denote Ω_p be the model space that agrees statistics constraint

$$\Omega_p = \{p : E_{p_k}[r_k] = E_f[r_k]\}. \tag{17}$$

Based on maximum entropy principle (MEP) [16], [37], the optimal model p_k^* for underlying model f is the distribution that matches certain observed statistics of (17), while should be also close to the learned model p_{k-1}. This is often expressed as a constrained optimization problem:

$$\begin{aligned} p_k^* &= \arg\min_{p\in\Omega_p} \mathrm{KL}(p_k||p_{k-1}) \\ s.t. E_{p_k}[r_k] &= E_f[r_k] \\ \sum_I p(I) &= 1. \end{aligned} \tag{18}$$

We solve this problem with multiplier Euler–Lagrange loss function and have

$$p_k(I) = p_{k-1}(I)\frac{1}{z_k}\exp\{\lambda_k r_k\} \tag{19}$$

where λ_k is the parameter (Lagrange multiplier) fixed by (17). z_k is the normalizing constant. It can be estimated by feature responses of the negative samples,

$$z_k = \int p_{k-1}(I)\exp\{\lambda_k r_k\} dI \approx \frac{1}{M}\sum_{m=1}^{M} e^{\lambda_k r^-_{k,m}}. \tag{20}$$

By recursion, we obtain a form for p,

$$p(I) = q(I)\prod_{k=1}^{K}\left[\frac{1}{z_k}\exp\{\lambda_k r_k\}\right]. \tag{21}$$

Thus, we define the likelihood energy $E_L(I_{\upsilon_r}|\ell_r, \rho_r; \boldsymbol{\theta}_r)$ as a log-linear form by the ratio between $p(I)$ and $q(I)$

$$E_L(I_{\upsilon_r}|\ell_r, \rho_r; \boldsymbol{\theta}_r) = \log\frac{p(I)}{q(I)} = \sum_{k=1}^{K}[\lambda_k r_k - \log z_k] \tag{22}$$

where $\boldsymbol{\theta}_r = \{\lambda_k, z_k\}_{k=1}^K$ denotes model parameters that are required to be estimated.

4) Selecting Features by Information Gain: Each iteration observes the following Pythagorean theorem which is known in information projection [16].

Proposition 1: Given the constraint defined in (17) and model form of (19), the model p_{k-1}, p_k and underlying distribution f satisfy the following equation.

$$\mathrm{KL}(f||p_{k-1}) - \mathrm{KL}(f||p_k) = \mathrm{KL}(p_k||p_{k-1}) > 0. \tag{23}$$

This proposition ensures the convergence of model p_k approaching f in learning process.

According to Proposition 1, we need to select most discriminative feature which largest deceases the KLD between model p_k and p_{k-1}. Thus, we define the gain or improvement by introducing feature r_k as

$$G_{r_k} = \mathrm{KL}(p_k||p_{k-1}) = \lambda_k E_f[r_k] - \log z_k \approx \lambda_k \bar{r}_k^+ - \log z_k. \tag{24}$$

5) Parameter Estimation: For notation clarification, we use $\{\hat{\lambda}_k, \hat{z}_k\}$ to represent the estimation of $\{\lambda_k, z_k\}$. Substitute (19) to (17), we have

$$\begin{aligned} E_{p_k}[r_k] &= E_{p_{k-1}}\left[\frac{1}{\hat{z}_k} r_k \exp\{\hat{\lambda}_k r_k\}\right] \\ &\approx \frac{1}{M}\sum_{m=1}^{M}\left[\frac{1}{\hat{z}_k} r^-_k \exp\{\hat{\lambda}_k r^-_k\}\right] = E_f[r_k] = \bar{r}_k^+ \end{aligned} \tag{25}$$

where $\hat{\lambda}_k$ is estimated by solving this function and $\hat{z}_k$ is calculated based on (20).

Theorem 1: In (25), $E_{p_k}[r_k]$ is a monotone increasing function of λ_k, and the slope at λ_k is σ_g^2, where $g(r_k; \lambda_k) = \frac{1}{z_k}e^{\lambda_k r_k}$. See appendix for prove and discussion.

Theorem 2: If $r^-_k > 0$, let $r^-_{k,\min}$ and $r^-_{k,\max}$ be the minimum and maximum value computed from D^-, then $E_{p_k}[r_k]$ is bounded by $r^-_{k,\min}$ and $r^-_{k,\max}$.
See appendix for prove and discussion.

Unlike the previous parameter estimation methods such as look-up table [22] and iterative scaling (IS) [37], branch-and-bound technique provides a good estimation and efficient computation scheme to calculate $\hat{\lambda}_k$ and $\hat{z}_k$. Theorems 1 and 2 guarantee our learning algorithm converges to the optimal parameters. Denote the low-value, up-value and middle-value of $E_{p_k}[r_k]$ as $E^{low}_{p_k}[r_k]$, $E^{up}_{p_k}[r_k]$ and $E^{mid}_{p_k}[r_k]$, which are calculated by given predefined rang $[\lambda_{k,\min}, \lambda_{k,\max}]$ and $\frac{\lambda_{k,\min}+\lambda_{k,\max}}{2}$. We judge whether $E_f[r_k]$ locates in $[E^{low}_{p_k}[r_k], E^{mid}_{p_k}[r_k]]$ or $[E^{mid}_{p_k}[r_k], E^{up}_{p_k}[r_k]]$ and update the corresponding range of λ_k. The process iteratively performs until $|E_f[r_k] - E^{mid}_{p_k}[r_k]|$ is lower than a threshold ξ, as summarized in Algorithm 1. To avoid over-fitting, we set $\lambda_{k,\max} = -\lambda_{k,\min} = 5$ following [22].[1]

[1]In practice, current setting still appears a bit large for learning our DHMRF model, and can be further reduced. However, too small range of $[\lambda_{k,\min}, \lambda_{k,\max}]$ will lead to infinite loop of Algorithm 1, since $E_f[r_k]$ may not locate in the range of $[E^{low}_{p_k}[r_k], E^{up}_{p_k}[r_k]]$. In spite of this, our algorithm is still computationally efficient by employing a branch-and-bound scheme, while retaining high recognition accuracy.

Algorithm 1: Parameter Estimation

Input: predefined range $[\lambda_{k,\min}, \lambda_{k,\max}]$, threshold ξ, $r^-_{k,\min}$, $r^-_{k,\max}$, $\bar{r}^+_k$ and $r^-_{k,m}, m = \{1, \ldots, M\}$
Output: $\{\hat{\lambda}_k, \hat{z}_k\}$

```
if r̄_k^+ ≥ r_{k,max}^- then
    λ̂_k = λ_{k,max};
end
if r̄_k^+ ≤ r_{k,min}^- then
    λ̂_k = λ_{k,min};
end
if r_{k,min}^- < r̄_k^+ < r_{k,max}^- then
    Compute E_{p_k}^{low}[r_k], E_{p_k}^{mid}[r_k] and E_{p_k}^{up}[r_k]
    while ξ ⩽ |E_f[r_k] − E_{p_k}^{mid}[r_k]| do
        if E_{p_k}^{low}[r_k] ⩽ r̄_k^+ ⩽ E_{p_k}^{mid}[r_k] then
            λ_{k,max} = (λ_{k,min} + λ_{k,max}) / 2
        end
        if E_{p_k}^{mid}[r_k] ⩽ r̄_k^+ ⩽ E_{p_k}^{up}[r_k] then
            λ_{k,min} = (λ_{k,min} + λ_{k,max}) / 2
        end
        Update E_{p_k}^{low}[r_k], E_{p_k}^{mid}[r_k] and [E_{p_k}^{up}[r_k];
    end
    Set λ̂_k = (λ_{k,min} + λ_{k,max}) / 2;
end
Estimate ẑ_k by (20);
```

Algorithm 2: Learning Algorithm

Input: feature candidate Ω_{cand}, global parameter ε
Output: the learned prototype T_ℓ or $\mathbf{h}_\ell$ with selected features and the corresponding parameters $\{\hat{\lambda}_k, \hat{z}_k\}_{k=1}^K$.

```
repeat
    Compute the feature responds according to Section III-A.2;
    foreach feature r_k ∈ Ω_cand do
        Compute its gain by (24);
    end
    Select r_k^* with the greatest gain;
    Estimate model parameter λ̂_k (feature weight) and ẑ_k (normalizing constant) based on Algorithm 1;
    Remove r_k from Ω_cand;
    Update current model p_k.
until Ω_cand = ∅ or gain is smaller than a threshold ε;
```

6) Summary of the Learning Algorithm: The whole algorithm for learning T_ℓ and $\mathbf{h}_\ell$ is described in Algorithm 2, with the stopping criterion that all the features $r_k \in \Omega_{\text{cand}}$ have been selected or the gain of r_k is less than a global parameter ε. In summary, Algorithm 2 iterates the following two steps:

a) Parameter estimation: Suppose we have chosen feature r_k and computed mean feature responds $\bar{r}^+_k \approx E_f[r^+_k]$, then the optimal model p^*_k is the one that is closest to p_{k-1} in Ω_p

$$p^*_k = \arg\min_{p^* \in \Omega_p} \text{KL}(p_k || p_{k-1}). \tag{26}$$

This step solves $\hat{\lambda}_k$ and $\hat{z}_k$ based on Algorithm 1.

b) Feature selection: Among all the candidate features in Ω_{cand}, we are required to choose most discriminative feature which has maximized reduction of KLD between p_k and p_{k-1}

$$p^*_k = \arg\max_{r^*_k \in \Omega_{\text{cand}}} \text{KL}(p_k || p_{k-1}). \tag{27}$$

V. Inference the DHMRF Model

Given the graphical DHMRF model and its learned parameters, we aim to find the optimal labeling result W^* which leads to the maximization of posterior probability $p(W|I)$ or equivalent Gibbs energy defined in (2)

$$W^* = \arg\max_W E(W|I; \boldsymbol{\Theta}). \tag{28}$$

However, using [23] to estimate structured objects leads our DHMRF model has unfixed graphical structure. Consequently, it is very hard to directly optimize energy $E(W|I; \boldsymbol{\Theta})$. Moreover, as mentioned in Section IV-A, exact computation of the partition function in our DHMRF model is intractable. We thus turn to a sampling scheme, in which the optimal labeling W^* can be estimated by the Data-Driven MCMC paradigm [42], [43]. By employing this method, the partition function $Z(\boldsymbol{\Theta}, I)$ does not need to compute in sampling process.

Generally speaking, we start with an initial state W_0 by randomly assigning the prototype ρ and label ℓ to each super-pixel.[2] In each iterative sampling, we form a series of connected components (CCPs) by cutting the contextual edges $\boldsymbol{\varepsilon}_C$ with probability $q_e = 1 - \exp\{E_C(\ell_p, \ell_q|[\cdot]\}$. Each CCP contains $|V_{CCP}|$ number of connected semantic vertices. Denote CCP_0 as one selected CCP and $\mathcal{C}$ as the cut of edges around CCP_0 in current state W. Within CCP_0, we sample new label ℓ' and prototype ρ' for each vertex based on two types of bottom-up proposals: generic and structured proposals, which are generated by [1] and [23] for generic and structured objects, respectively.[3] Each proposal has two attributes: the proposal label ℓ and probability q_v. Denote $q(W \to W') = \prod_{v=1}^{|V_{CCP_0}|} q_v$ as the proposal probability from state W to W'. The move is accepted with an acceptance probability $\alpha(W \to W') = \min(1, \prod_{e' \in \mathcal{C}'} q_{e'} q(W' \to W) \exp\{E(W|I)\} / \prod_{e \in \mathcal{C}} q_e \, q(W \to W') \exp\{E(W'|I)\})$, in which the partition function $Z(\boldsymbol{\Theta}, I)$ is cancelled.

The sampling procedure is computationally efficient, since in each iteration we only update the likelihood and associated coherent energies in CCP_0. The algorithm is guaranteed to converge, and its output is a global optimal, characterized by the property that the reversible moves can explore the solution space effectively [43].

[2]The final optimal solution is not sensitive to the initial configuration.

[3]To handel visual variability of scales and translations for structured objects, a set of detections are generated by scanning multiple potential scales and positions among the image lattice using [23]. The initial scale is set as 0.65 size ratio of the original image, then 10 scales are sampled by incremental ratio of 0.125. Note these detections are produced for every 5 pixels in each sampled scale. The detections, whose scores are higher than a pre-learned threshold, are used to form structured proposals.

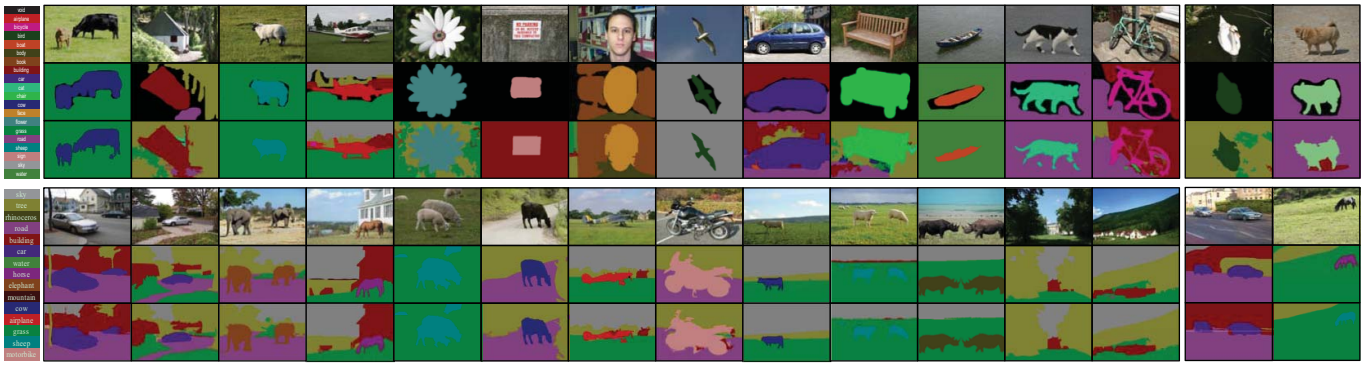


Fig. 6. Some labeling results on the MSRC 21-class (up panel) and LHI 15-class dataset (bottom panel). Each example shows the original image, corresponding ground truth and color-coded output labeling. For clarity, we also provide annotated legends used in two datasets on the left panel. The right panel exhibits examples where recognition works less well. (Best viewed in color.)

VI. Experiments

In this section, we evaluate our system on two challenging datasets: MSRC 21-class datasets [1] and LHI 15-class dataset [44], and compare our results with existing state-of-the-art.

The MSRC 21-class dataset [1] is a very popular benchmark for image labeling, which consists of 591 images including 12 types of structured objects ("flower", "sign", "cow", "sheep", "aeroplane", "face", "car", "bike", "bird", "cat", "dog" and "chair") and 9 types of generic objects ("building", "grass", "tree", "sky", "water", "book", "road", "body" and "boat"). The pixels labeled as "void" are not considered during the training and testing process for direct comparison.

The LHI 15-class dataset consists of 375 images including 8 types of structured objects ("airplane", "cow", "horse", "sheep", "car", "elephant", "rhinoceros" and "motorbike") and 7 types of generic classes ("building", "grass", "tree", "sky", "road", "water" and "mountain"). All images are rescaled to resolution 320×210. To address the problem of "what is the most likely label for each pixel in a given image", we take a voting strategy by labeling a "ground truth" in the image with multiple annotators independently. Compared with MSRC 21-class dataset, images in LHI 15-class dataset are well hand-labeled to achieve accurate segmentation. Each pixel is assigned a color to indicate its label with most annotators. Some examples are illustrated in Fig. 6. Without losing generalization, we select the images by taking into account the following conditions: camera viewpoint, little occlusions, multi-objects, lighting conditions, object pose, deformation, and scale variance. As illustrated in Figs. 1 and 6, each image nearly contains $3 \sim 6$ objects. In our experiments, the two datasets use the same split setting following [1].

A. Overall Labeling Results

This section presents quantitative and qualitative results for our full DHMRF model on two datasets. The parameter settings, learned against the validation set,[4] were $\mathcal{M} = 200$ for super-pixel number, Garbor wavelet number $s = 42$ for structured objects, histogram number $g = 18$ for generic class, $\mathcal{O} = 8$ sectors for layered context definition, offset prior parameters $\omega_{co} = \omega_{ly} = 0.3$, stop criterion $\xi = \varepsilon = 10^{-3}$ and $T = 200$ for iterative sampling number.

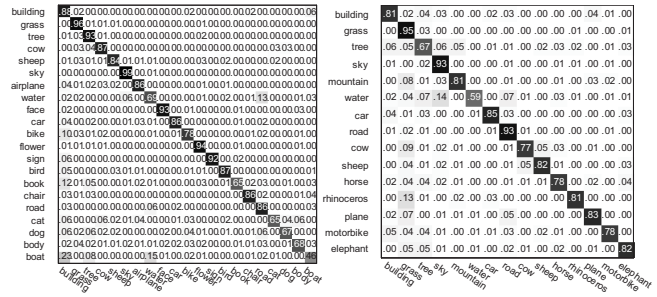
[4]In learning process, we select the optimization for one parameter by fixing others.

TABLE I
Comparison of Pixel-Wise Accuracy and Implement Efficiency on MSRC 21-Class and LHI 15-Class Datasets. Note Training Times Are for the Whole Training Set, Test Times Are Per Image

Methods	Accuracy		Efficiency (Train(h)/Test(s))		Code
	MSRC	LHI	MSRC	LHI	
Ours	**81.7%**	**81.3%**	**0.6** / 8.4	**0.3** / 6.8	C++
HIM [45]	81.2%	—	— / —	— / —	—
AC [2]	77.7%	76.5%	7.7 / 7.4	3.9 / 7.6	C++
RL [21]	76.5%	71.6%	5.5 / 5.7	2.1 / **5.1**	C++
DS [46]	76.4%	71.1%	6.1 / 21.1	3.0 / 21.5	C++
H-CRFs [47]	74.6%	66.5%	5.4 / 16.8	2.7 / 16.6	Matlab
TB [1]	72.2%	62.7%	6.3 / **5.4**	3.8 / 5.2	C#

1) Quantitative Results: Table I shows the comparison of overall pixel-wise accuracy and average implemented efficiency on MSRC 21-class and LHI 15-class datasets. All methods are implemented using a Dual Core 2.6 GHz machine with 2GB memory. It demonstrates our algorithm outperforms other state-of-the-art methods. Training our model is extremely fast which benefits from the independent assumption and branch-and-bound method. Testing time on MSRC dataset only takes 8.4 seconds per image, which is comparable to [1], [2], [21]. The majority of this time is spent performing the bottom-up computation and iterative sampling, which takes 6 seconds and 2.4 seconds, respectively. For each testing image, time for generating generic proposals is under 1 second, while approximately 5 seconds for structured proposals. In practice, we learn a series of pruning thresholds for each ℓ over the validation set. If recognizable outputs of [1], [23] are lower than pre-computed thresholds, we do not generate corresponding proposals. This achieves nearly 5-fold times faster for labeling a testing image without significant performance lost.

Fig. 7 illustrates the confusion matrices by applying our full DHMRF model on two datasets, in which accuracy values are



(a) (b)

Fig. 7. Confusion matrices of our model evaluated on (a) MSRC 21-class and (b) LHI 15-class dataset. The overall pixel-wise accuracy achieves 81.7% and 81.27%, respectively.

TABLE II
COMPARISON OF OUR BRANCH-AND-BOUND ALGORITHM WITH OTHER METHODS ON TESTING SET. NOTE THE REPORTED NUMBER IS AVERAGED OVER ALL FEATURES AND CATEGORIES

Dataset	ξ	10e-1	10e-2	10e-3	10e-4
MSRC	**ours**	**18**	**35**	**79**	**181**
	CG [49]	41	64	142	646
	GA [50]	127	256	461	907
	IS [37]	187	341	817	1334
LHI	**ours**	**16**	**57**	**94**	**196**
	CG [49]	72	162	446	717
	GA [50]	243	304	732	1236
	IS [37]	388	811	1217	1743

computed as the percentage of image pixels assigned to the correct class label. The average pixelwise labeling accuracy of structured objects is 75.6% and 80.75% on two datasets, which implies the shape prior are efficiently captured. Some structured objects in LHI dataset, e.g., "cow", "sheep", and "horse", exhibit relatively large confusions because of their similar deformable template (as illustrated in Fig. 8). We also discover that some generic categories exhibit large errors, e.g., "road" mislabeled as "water", "book" incorrectly recognized as "building", especially "boat" and "building", which due to their inter-class color/texture similarities.

2) Qualitative Results: Example results of simultaneous recognition and segmentation on two datasets are shown in Fig. 6. Our algorithm can handle large visual variability of articulations, rotations, orientations and scales for structured objects (e.g., "flower", "face", "bird", "car" and "elephant"), and of appearance for generic categories (e.g., "building", "grass", and "tree"). An exciting example is the first one on MSRC dataset, in which two cows are well modeled by the deformable templates and appearance models. Another interesting one is the sixth example on MSRC dataset. Although the ground truth does not provide good annotation, our algorithm still correctly recognizes the background as "building".

Some results with poor performance are also displayed in the right panel of Fig. 6. In the first three examples, some instances of "water", "road" and "tree" are misclassified as "tree" and "building", since learned prototypes are not distinct enough in such cases. In the last example, a "horse" is incorrectly labeled as "sheep" due to their inter-class similarities both on shape and appearance. Even when recognition fails, however, segmentation may still be quite accurate.

B. Analysis of Learning Algorithm

In order to better understand the behavior of our labeling system, we highlight some aspects of our learning algorithm.

1) Deformable Templates/Appearance Models: Fig. 8 shows some deformable templates learned from two datasets. We observe that the learned templates capture human intuition and the shape information are well modeled. Note some templates are shared for two datasets, such as "sheep", "car" and "cow". Most structured objects have around $5 \sim 15$ training images. If a structured object can not be well represented by one common shape template, such as orientations and shape variance, we learn multiple templates using an EM procedure to deal with this visual variability [22]. The learned shape models are also robust to some slight deformation of non-rigid structured objects, since the associated Gabor wavelets are allowed to slightly perturb their locations and orientations.

In order to analyze the characteristics of learned $\mathbf{h}_\ell$, we use the algorithm of [48] to synthesize generic models, where the marginal distributions of synthesized image are matched to $\mathbf{h}_\ell$. The left panel of Fig. 9 plots some synthesized prototype $\mathbf{h}_\ell$ on two datasets to gather some intuitively understanding of different categories. For instance, "building" always has horizontal, and vertical edges and "sky" can be represented by flat blue area. We also synthesized the prototypes for structured objects (e.g., "horse", "car" and "cow") in the case that they can not be covered by shape templates. Although these prototypes are not good enough to describe their own generic characteristic, they are discriminative enough to distinguish from each other, which is benefit for recognition.

2) Feature Selection: In order to evaluate how the statistics features are selected for different $\ell \in Rgn$ and how much they contribute to classification, we highlight the iterative learning process of four typical categories: "building", "sky", "road" and "airplane" in the right panel of Fig. 9. In the first sub-image of each category, we plot the gains can be achieved with each selected feature in decreasing order. We also illustrate the corresponding marginal distribution of each selected feature in the following 17 sub-images, where the red solid line is for the current model p_k and the blue dash line is for previous model p_{k-1}. The more difference between these tow curves, the more informative this feature is. To further illustrate this procedure, each selected feature is also displayed in the up-left corner of sub-images. For example, for "sky", which has pure patch, blue color is first selected. For "road", "building" and "airplane", as there are cluttered structures inside objects, texture features make bigger contributions. For most classes, the gains decrease to zero which indicates our 18 features (the first 15 histogram prototypes for describing texture and the last 3 ones for color) are efficient to represent each category.

3) Parameter Estimation: We also compare our parameter estimation method with some widely-used numerical calculation algorithms, such as IS [37], conjugate gradient (CG) [49] and gradient ascent (GA) [50]. For fair comparison, we use

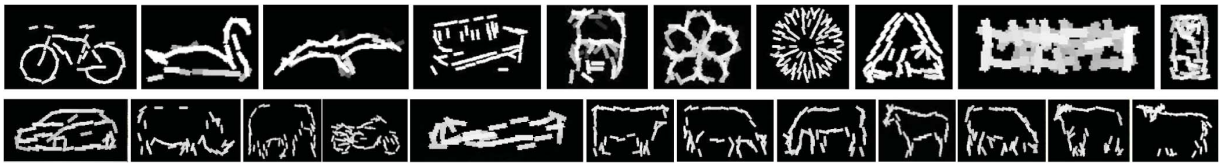


Fig. 8. Some learned deformable templates for MSRC 21-class (up panel) and LHI 15-class dataset (bottom panel). In ordering, the learned templates are "bicycle", "bird", "chair", "face", "flower", "sign", "car", "rhinoceros", "elephant", "motorbike", "airplane", "cow", "horse", and "sheep".

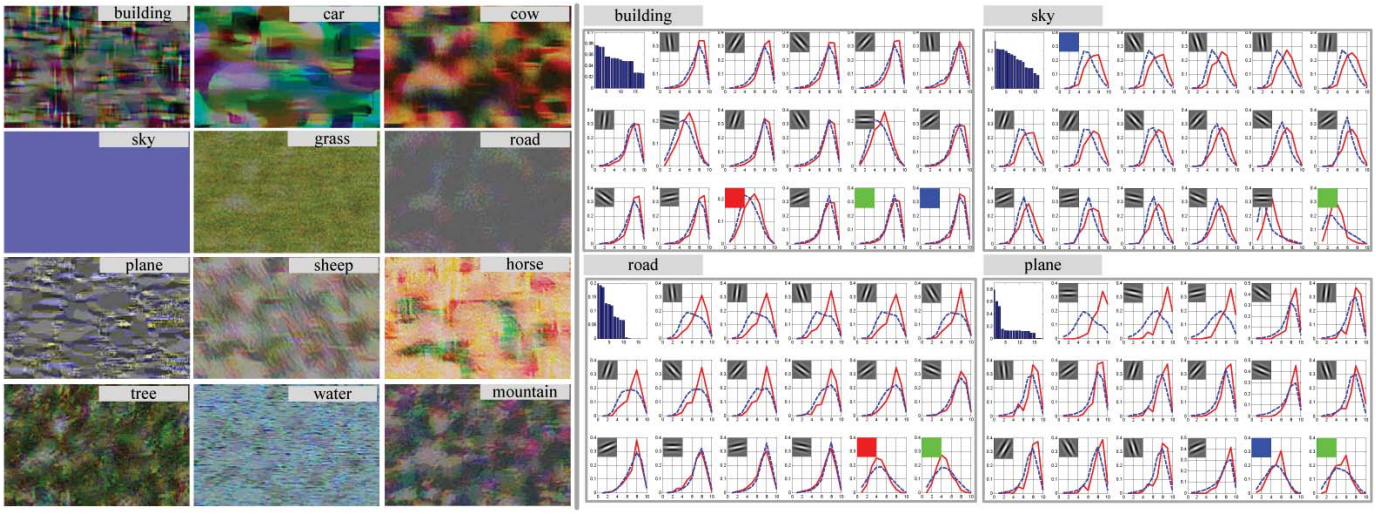
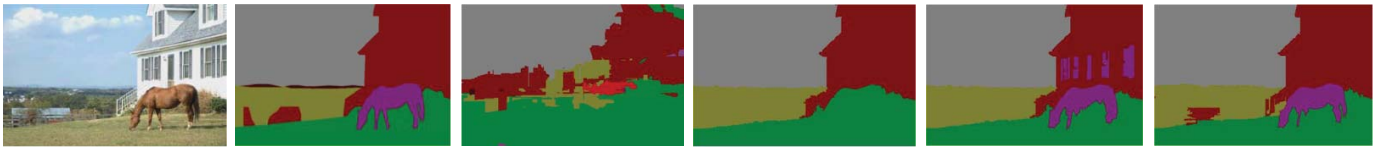


Fig. 9. Some learned prototypes for appearance models (left panel) and texture/color contributions to classification (right panel). The synthesized prototypes of "horse" and "mountain" are from LHI 15-class dataset, while the others are from MSRC 21-class dataset. (Best viewed in color.)



(a) (b) (c) (d) (e) (f)

Fig. 10. Effects of different terms in our DHMRF model. (a) Input image. (b) Ground truth. (c) TextonBoost. (d) Result only using the generic models, without explicit shape templates and contextual coherency. (e) Gives results for the DHMRF model taking deformable templates into account. (f) For our full DHMRF model. (Best viewed in color.)

TABLE III
PIXEL-WISE ACCURACY COMPARISON ON MSRC 21-CLASS AND LHI 15-CLASS DATASETS WITH DIFFERENT LEARNING METHODS

Methods	MSRC 21-Class	LHI 15-Class
Ours	**81.7%**	**81.3%**
Boosting [15] + context	78.3%	79.5%
SVM [14] + context	75.4%	76.1%

the same stopping criterion as $|E_f[r_k] - E_{p_k}^{\hat{\lambda}^t}[r_k]| \leq \xi$, where $\hat{\lambda}^t$ is the parameter estimated in t^{th} iteration. Table II reports the results in terms of iteration number, and shows that using branch-and-bound scheme achieves higher efficiency than other estimation methods.

4) Comparison With Discriminative Learning Methods: To assess how much our learning algorithm help with recognition, Table III gives the comparison results on the accuracy of our system against the discriminative learning approaches, such as SVM [14] and Boosting [15], under the same inference framework.[5] Since we adopt s and g filter responses for structured and generic objects, each training example is represented by a s-dimensional or g-dimensional vector. For two methods, we show the results of directly substituting our likelihood learning algorithm by training linear SVM and regression boosted classifiers, respectively.

We train the classifiers separately in one vs. all fashion among all $\ell \in Rgn$ or $\ell \in Str$. By considering contextual models, the SVM [14] gains 75.4% and 76.1% with penalty parameter $C = 1000$ on two datasets. In training classifiers using [15], we adopt the decision trees as the weak classifier within 400 iterations. It achieves a little higher performance: 78.3% and 79.5%. Discriminative learning methods sometimes incorrectly classify structured objects (e.g., "car", "cow" and "horse") as generic objects, suggesting these approaches have been over-fitted. Furthermore, learning these discrim-

[5]We also use [43] to get final labeling results for direct comparison. Note the trained discriminative models [1], [23] are also used to generate two types of proposals for inference.

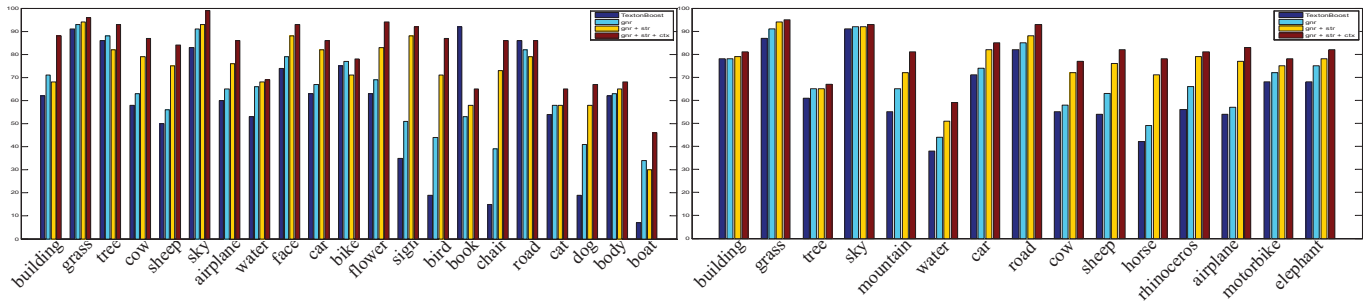


Fig. 11. Quantitative comparison with different energy terms on MSRC 21-class (left panel) and LHI 15-class dataset (right panel).

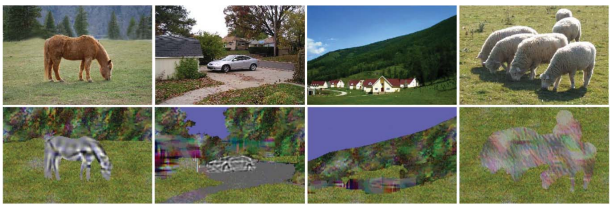


Fig. 12. Some synthesized images based on our labeling results. (Best viewed in color)

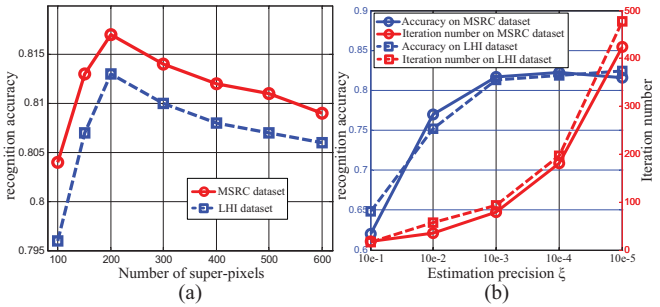


Fig. 13. Effects of: (a) over-segmentation and (b) estimation precision on MSRC 21-class and LHI 15-class datasets.

inative models requires vast computational efforts than our method.

C. Other Aspects

1) Effect of Different Energy Terms: Fig. 10 shows a labeling result of one example in LHI 15-class dataset, which takes account to each energy term defined in (2). For directly comparison, the labeling output in (c) is adopted as baseline using [1]. From (d) to (f), it is evident that imposing appearance models and deformable templates, as well as contextual models, improves the results considerably. Directly incorporating the deformable templates helps to rectify the true labels of misclassified object instances, e.g., the horse in (e). However, it also leads to the lost of performance because of introducing false alarms (note the building windows are labeled as horse). Fortunately, this ambiguity can be resolved by coherent energies as shown in (f).

In Fig. 11, we illustrate recognition accuracy with different energy terms evaluated over the whole test set. These energy terms make effort to capture essential information to improve the labeling performance. Compared with other energy terms, deformable templates achieve the remarkable improvement for most structured objects, e.g., "sign", "bird", and "face" in MSRC dataset and "cow", "sheep", and "horse" in LHI dataset. This demonstrates in some cases the shape prior is more discriminative than texture/color features. Note only one class "book" obtains poor performance compared with [1], which are always confused with "building".

2) Analysis of Image Synthesis: To evaluate the generative property of DHMRF model, we synthesize some images based on labeling results. For $\ell \in Str$, the syntheses are based on the selected Garbor wavelets and some difference of Gaussian filters (DoG) as it was done in [22]. The synthesized results are shown in grey. For $\ell \in Rgn$, we copy the corresponding image pixels from the synthesized prototypes $\mathbf{h}_\ell$ (shown in Fig. 9). Fig. 12 shows some examples of synthesized images based on our labeling results. It is observed that the synthesized parts corresponding to same appearance model look similar within different images.

Fig. 14. Using DHMRF model in images with noise over MSRC (left panel) and LHI (right panel) datasets. From top to bottom: Original images, contaminated images, ground truth, and labeling results.

3) Effect of Over-Segmentation: One factor affecting the performance is the granularity of over-segmentation (i.e. the number of super-pixels $\mathcal{M}$). In practice, we repeated our experiments on two datasets using different numbers of super-pixels. Fig. 13(a) exhibits the plot of recognition accuracy versus number of super-pixels. We observe that the accuracy of our method is insensitive to changes in the number of segments after approximately 200 super-pixels, and any refinement to the over-segmentation will result in slightly decrease of performance. This is also observed by [21].

4) Analysis of Estimation Precision: We also measure the affect of changing the estimation precision ξ involved in Algorithm 1. Fig. 13(b) shows the recognition accuracy and

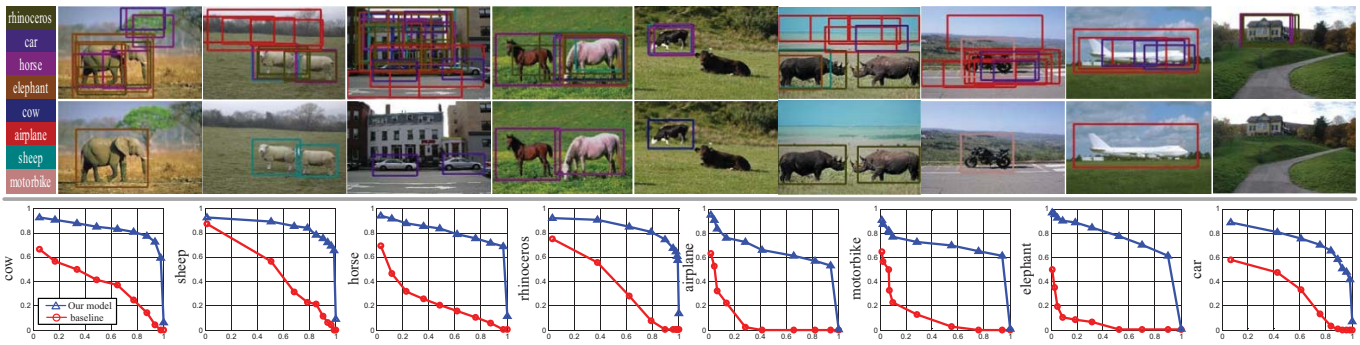


Fig. 15. Comparable quantitative and qualitative results for object detection. The outputs of [23] are represented by a set of bounding box with different colors. In the left column of up panel, we gives the legend for different category. The bottom panel illustrates the precision-recall (PR) curves of the detection results for the eight structured objects on the LHI 15-class dataset, using our method (blue line) and [23] (red line) as baseline. (Best viewed in color.)

TABLE IV
PERFORMANCE OF OUR DHMRF MODEL ON CONTAMINATED IMAGES
(σ RANGES FROM 0.05 TO 0.35)

Dataset	σ	0.05	0.15	0.25	0.35
MSRC	Ours	**80.3%**	**77.2%**	**72.6%**	**67.7%**
	AC [2]	75.4%	73.9%	69.3%	65.5%
	RL [21]	73.7%	70.6%	67.4%	62.2%
	DS [46]	73.1%	69.8%	66.7%	61.6%
	H-CRF [47]	71.6%	66.0%	63.5%	58.7%
	TB [1]	69.9%	64.2%	60.1%	55.3%
LHI	Ours	**79.8%**	**76.4%**	**69.7%**	**65.2%**
	AC [2]	72.4%	70.1%	64.4%	61.8%
	RL [21]	68.8%	66.0%	62.3%	58.5%
	DS [46]	68.3%	65.2%	60.1%	56.7%
	H-CRF [47]	64.4%	60.6%	55.9%	52.2%
	TB [1]	60.5%	57.4%	53.3%	50.6%

iteration number vs. ξ on two datasets. Note the reported number is averaged over all features and categories. As can be seen, smaller value of ξ results in higher recognition accuracy, while the iteration number increases drastically. Note in MSRC dataset, the performance slightly decreases when $\xi = 10e-5$, probably because it leads to over-fitting of the learned model.

5) Analysis of Contaminated Images: In order to further evaluate DHMRF model, we apply our framework on the images contaminated by noise. For each image, its pixel values are corrupted by additive Gaussian noise with zero mean and the standard deviation σ, where σ ranges from 0.05 to 0.35. Table IV shows that our DHMRF model outperforms other methods for handling noise. Fig. 14 further provides some examples by applying our model on the images contaminated with such Gaussian noise. These results illustrate that the mechanism of embedding *explicit* shape priors is more robust to noise than the approaches only based on appearance cues.

6) Improvement for Object Detection: Another highlight property of our approach is the improvement for object detection. By adopting [23] as baseline, we use non-maximum suppression as pre-process step to discard heavy overlapped instances with extremely lower detective probability. The up panel of Fig. 15 shows some comparable results between [23] and ours on LHI 15-class dataset, where the detective structured instances (e.g., "car", "horse" and "airplane") are represented by a series of bounding boxes with different colors. Our layered context, however, gives very low probability when these instances preferentially locates in the backgrounds, such as "building", "sky" and "tree".

For more details, the lower panel of Fig. 15 shows the Precision-Recall curves of object detection for the 8 structured objects in the LHI 15-class dataset. Compared with base-line detection results, our model achieve remarkable improvement to the performance since most detective false alarms have been eliminated by our layered contextual model.

VII. CONCLUSION

In this paper, we present a DHMRF model to address the challenges of incorporating *explicit* shape templates as well as appearance models into a probabilistic multi-class image segmentation framework. The key advantages of our method are: 1) DHMRF model is robust to intra-class variance and inter-class similarities, by integrating two types of visual prototypes: deformable templates and appearance models. 2) The information projection principle allows us to learn these prototypes within unified framework. 3) Due to the independent assumption and branch-and-bound method, our learning algorithm is more computationally efficient than the traditional generative learning approaches, while maintaining good recognition accuracy. 4) The co-occurrence, especially the layered context, plays an important role in improving the performance both on the tasks of image labeling and object detection. In experiments, our method has been tested on two datasets: MSRC 21-class and LHI 15-class dataset. We evaluate the results in terms of pixel-wise segmentation accuracy. Our algorithm achieves extremely fast learning efficiency and outperforms the state-of-the-art methods. We also compare our learning algorithm with other discriminative learning methods and analyze different aspects of our method in details.

Despite achieving state-of-the-art accuracy, we believe that even better results can be obtained by taking appearance cues and shape features *jointly* into account to model structured objects. We are aware of a recent work [51] in this direction. We are also interested in hierarchical part-based shape model, such as [9], [52] for structured objects to cope with inter- or self-occlusion, while retaining high efficiency and accuracy.

APPENDIX

VIII. MATHEMATICAL DETAILS

Prove 1 (Prove of Theorem 1): Substitute (19) to $E_p[r_k]$, then we have

$$E_p[r_k] = \int \frac{1}{z_k} p_{k-1} r_k e^{\lambda_k r_k} dI = E_{p_{k-1}}\left[\frac{1}{z_k} r_k e^{\lambda_k r_k}\right]. \quad (29)$$

Substitute (20) to (29), we get

$$E_{p_{k-1}}\left[\frac{1}{z_k} r_k e^{\lambda_k r_k}\right] \approx \frac{\sum_{m=1}^{M} r_{k,m}^{-} e^{\lambda_k r_{k,m}^{-}}}{\sum_{m=1}^{M} e^{\lambda_k r_{k,m}^{-}}}. \quad (30)$$

Let $f(\lambda_k) = E_{p_{k-1}}[\frac{1}{z_k} r_k e^{\lambda_k r_k}]$, we have

$$\begin{aligned}\frac{\partial(f(\lambda_k))}{\partial \lambda_k} &= \left[\frac{\sum_{m=1}^{M} (r_{k,m}^{-})^2 e^{\lambda_k r_{k,m}^{-}}}{\sum_{m=1}^{M} e^{\lambda_k r_{k,m}^{-}}}\right] \\ &\quad - \left[\frac{\sum_{m=1}^{M} r_{k,m}^{-} e^{\lambda_k r_{k,m}^{-}}}{\sum_{m=1}^{M} e^{\lambda_k r_{k,m}^{-}}}\right]^2 \\ &\approx \frac{1}{M}\sum_{m=1}^{M} \frac{1}{z_k} (r_{k,m}^{-})^2 e^{\lambda_k r_{k,m}^{-}} \\ &\quad - \left[\frac{1}{M}\sum_{m=1}^{M} \frac{1}{z_k} r_{k,m}^{-} e^{\lambda_k r_{k,m}^{-}}\right]^2. \end{aligned} \quad (31)$$

Let $g(r_k; \lambda_k) = \frac{1}{z_k} e^{\lambda_k r_k}$, then we have

$$\frac{\partial f(\lambda_k)}{\partial \lambda_k} = E_g[(r_k^{-})^2] - (E_g[r_k^{-}])^2 = \sigma_g^2 \geq 0 \quad (32)$$

where σ_g^2 is the variance of $g(r_k; \lambda_k)$. Then the result follows. ■

Prove 2 (Prove of Theorem 2): We expand $f(\lambda_k)$ for each component as:

$$\begin{aligned} f(\lambda_k) &= \frac{r_{k,1}^{-}}{\sum_{m=1}^{M} [e^{\lambda_k (r_{k,m}^{-} - r_{k,1}^{-})}]} + \cdots \\ &\quad + \frac{r_{k,j}^{-}}{\sum_{m=1}^{M} [e^{\lambda_k (r_{k,m}^{-} - r_{k,j}^{-})}]} + \cdots. \end{aligned} \quad (33)$$

Then we get

$$\lim_{\lambda_k \to +\infty} f(\lambda_k) = r_{k,1}^{-} \cdot 0 + \cdots + r_{k,\max}^{-} \cdot 1 + \cdots = r_{k,\max}^{-} \quad (34)$$

$$\lim_{\lambda_k \to -\infty} f(\lambda_k) = r_{k,1}^{-} \cdot 0 + \cdots + r_{k,\min}^{-} \cdot 1 + \cdots = r_{k,\min}^{-}. \quad (35)$$

Based on Theorem 1, then the result follows. ■

ACKNOWLEDGMENT

The authors would like to thank all the anonymous reviewer's valuable comments.

REFERENCES

[1] J. Shotton, J. M. Winn, C. Rother, and A. Criminisi, "Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *Int. J. Comput. Vis.*, vol. 81, no. 1, pp. 2–23, 2009.

[2] Z. W. Tu and X. Bai, "Auto-context and its application to high-level vision tasks and 3D brain image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 10, pp. 1744–1757, Oct. 2010.

[3] Z. W. Tu, X. Chen, A. L. Yuille, and S. C. Zhu, "Image parsing: Unifying segmentation, detection, and recognition," in *Int. J. Comput. Vis.*, vol. 63, no. 2, pp. 113–140, 2005.

[4] A. Singhal, J. Luo, and W. Zhu, "Probabilistic spatial context models for scene content understanding," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1. Jun. 2003, pp. 235–241.

[5] J. Verbeek and B. Triggs, "Region classification with markov field aspect models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[6] S. Gould, T. Gao, and D. Koller, "Region-based segmentation and object detection," in *Proc. Adv. Neural Inf. Process. Syst. Conf.*, 2009, pp. 1–17,

[7] L. Ladicky, P. Sturgess, K. Alahari, C. Russell, and P. Torr, "What, Where and how many? combining object detectors and CRFs," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010.

[8] E. Borenstein and S. Ullman, "Combined top-down/bottom-up segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 12, pp. 2109–2125, Dec. 2008.

[9] Y. Yang, S. Hallman, D. Ramanan, and C. Fowlkes, "Layered object detection for multi-class segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3113–3120.

[10] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, no. 6, pp. 721–741, Nov. 1984.

[11] F. Khalifa, G. Beache, G. Gimelfarb, G. Giridharan, and A. El-Baz, "Accurate automatic analysis of cardiac cine images," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 2, pp. 445–455, Feb. 2012.

[12] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th Int. Conf. Mach. Learn.*, Jun. 2001, pp. 282–289.

[13] S. Kumar and M. Hebert, "Discriminative random fields: A discriminative framework for contextual interaction in classification," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 1150–1157.

[14] V. Vapnik, *The Nature of Statistical Learning Theory*. New York, USA: Springer-Verlag, 2000.

[15] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," *Ann. Stat.*, vol. 28, no. 2, pp. 337–407, 2000.

[16] S. C. Zhu, Y. Wu, and D. Mumford, "Minimax entropy principle and its applications to texture modeling," *Neural Comput.*, vol. 9, no. 8, pp. 1627–1660, 1997.

[17] X. Feng, C. Williams, and S. Felderhof, "Combining belief networks and neural networks for scene segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 467–483, Apr. 2002.

[18] L. Yang, P. Meer, and D. Foran, "Multiple class segmentation using a unified framework over mean-shift patches," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[19] B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model," in *Proc. Eur. Conf. Comput. Vis. Workshop Stat. Learn. Comput. Vis.*, 2004, pp. 17–32.

[20] X. He, R. Zemel, and D. Ray, "Learning and incorporating top-down cues in image segmentation," in *Proc. 9th Eur. Conf. Comput. Vis.*, 2006, pp. 338–351.

[21] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller, "Multi-class segmentation with relative location prior," *Int. J. Comput. Vis.*, vol. 80, no. 3, pp. 1239–1253, Dec. 2008.

[22] Y. N. Wu, Z. Z. Si, H. F. Gong, and S. C. Zhu, "Learning active basis model for object detection and recognition," *Int. J. Comput. Vis.*, vol. 90, no. 2, pp. 198–235, 2010.

[23] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[24] A. Levin and Y. Weiss, "Learning to combine bottom-up and top-down segmentation," *Int. J. Comput. Vis.*, vol. 81, no. 1, pp. 105–118, 2009.

[25] X. Ren, C. Fowlkes, and J. Malik, "Cue integration for Figure/Ground labeling," in *Proc. Adv. Neural Inf. Process. Syst. Conf.*, vol. 18. 2006, pp. 1–8.

[26] X. M. He, R. S. Zemel, and D. Ray, "Learning and incorporating top-down cues in image segmentation," in *Proc. 9th Eur. Conf. Comput. Vis.*, May 2006, pp. 338–351.
[27] D. Hoiem, A. Efros, and M. Hebert, "Closing the loop in scene interpretation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[28] C. Galleguillos, A. Rabinovich, and S. Belongie, "Object categorization using co-occurrence, location and appearance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[29] S. Kumar and M. Hebert, "A hierarchical field framework for unified context-based classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2005, pp. 1284–1291.
[30] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky, "Learning hierarchical models of Scenes, Objects, and Parts," in *Proc. 10th IEEE Int. Conf. Comput. Vis.*, Oct. 2005, pp. 1331–1338.
[31] J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *J. Roy. Stat. Soc. Ser. B*, vol. 36, no. 2, pp. 192–236, 1974.
[32] R. Dubes and A. Jain, "Random field models in image analysis," *J. Appl. Stat.*, vol. 16, no. 2, pp. 131–164, 1989.
[33] A. Torralb, K. P. Murphy, and W. T. Freeman, "Contextual models for object detection using boosted random fields," in *Proc. Adv. Neural Inf. Process. Syst. Conf.*, Jun. 2004, pp. 1–10.
[34] J. Verbeek and B. Triggs, "Scene segmentation with conditional random fields learned from partially labeled images," in *Proc. Adv. Neural Inf. Process. Syst. Conf.*, Dec. 2007, pp. 1–8.
[35] C. Wojek and B. Schiele, "A dynamic conditional random field model for joint labeling of object and scene classes," in *Proc. 10th Eur. Conf. Comput. Vis.*, Oct. 2008, pp. 733–747.
[36] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of Online learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.
[37] S. D. Pietra, V. J. D. Pietra, and J. D. Lafferty, "Inducing features of random fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 4, pp. 380–393, Apr. 1997.
[38] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
[39] J. Wu and J. M. Rehg, "Beyond the euclidean distance: Creating effective visual codebooks using the histogram intersection kernel," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep.–Oct. 2009, pp. 630–637.
[40] A. Rabinovich, A. Vedaldi, C. G. E. Wiewiora, and S. Belongie, "Objects in context," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.
[41] S. Li, *Markov Random Field Modeling in Computer Vision*. New York, USA: Springer-Verlag, 1995.
[42] Z. W. Tu and S. C. Zhu, "Image segmentation by data-driven Markov chain Monte Carlo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 657–673, May 2002.
[43] A. Barbu and S. C. Zhu, "Generalizing swendsen-wang to sampling arbitrary posterior probabilities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1239–1253, Aug. 2005.
[44] B. Yao, X. Yang, and S. C. Zhu, "Introduction to a large scale general purpose groundtruth dataset: Methodology, annotation tool, and benchmarks," in *Proc. Int. Conf. Energy Minimization Comput. Vis. Pattern Recognit.*, 2007, pp. 169–183.
[45] L. Zhu, Y. H. Chen, Y. Lin, C. X. Lin, and A. Yuille, "Recursive segmentation and recognition templates for 2D parsing," in *Proc. Adv. Neural Inf. Process. Syst. Conf.*, 2008, pp. 1–8.
[46] S. Gould, R. Fulton, and D. Koller, "Decomposing a scene into geometric and semantically consistent regions," in *Proc. Int. Conf. Comput. Vis.*, 2009, pp. 1–8.
[47] L. Ladicky, C. Russell, and P. Kohli, "Associative hierarchical crfs for object class image segmentation," in *Proc. Int. Conf. Comput. Vis.*, 2009, pp. 1–8.
[48] S. C. Zhu, X. W. Liu, and Y. N. Wu, "Exploring texture enesmbles by efficient markov chain monte carlo-toward a 'trichromacy' theory of texture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 6, pp. 554–569, Jun. 2000.
[49] M. Jamshidian and R. Jennrich, "Acceleration of the em algorithm by using quasi-newton methods," *J. Roy. Stat. Soc. Ser. B*, vol. 59, no. 3, pp. 569–587, Mar. 1997.
[50] R. Malouf, "A comparison of algorithms for maximum entropy parameter estimation," in *Proc. Conf. Natural Lang. Learn.*, 2002, pp. 49–55.
[51] Z. Z. Si, H. F. Gong, Y. N. Wu, and S. C. Zhu, "Learning mixed templates for object recognition," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 272–279.
[52] P. Felzenszwalb and D. Huttenlocher, "Pictorial structures for object recognition," *Int. J. Comput. Vis.*, vol. 61, no. 1, pp. 55–79, 2005.

Quan Zhou received the B.S. degree in electronics and information engineering from the China University of Geosciences, Wuhan, China, in 2002, and the M.S. degree in electronics and information engineering from the Huazhong University of Science and Technology, Wuhan, in 2006, where he is currently pursuing the Ph.D. degree from the Department of Electronics and Information Engineering.

His current research interests include computer vision and pattern recognition.

Jun Zhu received the B.S. degree in communication engineering and the M.S. degree in signal and information processing from Chongqing University, Chongqing, China, in 2004 and 2007, respectively. He is currently pursuing the Ph.D. degree with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China.

His current research interests include image and video analysis, computer vision, and machine learning.

Wenyu Liu received the B.S. degree in computer science from Tsinghua University, Beijing, China, in 1986, and the M.S. and Ph.D. degrees in electronics and information engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 1991 and 2001, respectively.

He is currently a Professor and the Associate Dean with the Department of Electronics and Information Engineering, HUST. His current research interests include multimedia information processing and computer vision.