Recursive Segmentation and Recognition Templates for Image Parsing

Long (Leo) Zhu[1], Yuanhao Chen[2], Yuan Lin[3], Chenxi Lin[4], Alan Yuille[2]

[1]CSAIL, MIT
leozhu@csail.mit.edu
[2]Department of Statistics, UCLA
yuille@stat.ucla.edu
[3]Shanghai Jiaotong University
loirey@sjtu.edu.cn
[4]Alibaba Group R&D
chenxi.lin@alibaba-inc.com

Abstract— **Language and image understanding are two major goals of artificial intelligence which can both be conceptually formulated in terms of parsing the input signal into a hierarchical representation. Natural language researchers have made great progress by exploiting the 1D structure of language to design efficient polynomial-time parsing algorithms. By contrast, the two-dimensional nature of images makes it much harder to design efficient image parsers and the form of the hierarchical representations is also unclear. Attempts to adapt representations and algorithms from natural language have only been partially successful.**

In this paper, we propose a Hierarchical Image Model (HIM) for 2D image parsing which outputs image segmentation and object recognition. This HIM is represented by recursive segmentation and recognition templates in multiple layers and has advantages for representation, inference, and learning. Firstly, the HIM has a coarse-to-fine representation which is capable of capturing long-range dependency and exploiting different levels of contextual information. Secondly, the structure of the HIM allows us to design a rapid inference algorithm, based on dynamic programming, which enables us to parse the image rapidly in polynomial time. Thirdly, we can learn the HIM efficiently in a discriminative manner from a labeled dataset. We demonstrate that HIM is comparable with the state-of-the-art methods by evaluation on the challenging public MSRC and PASCAL image datasets. Finally, we sketch how the HIM architecture can be extended to model more complex image phenomena.

I. Introduction

Language and image understanding are two major tasks in artificial intelligence. Natural language researchers have formalized this task in terms of parsing an input signal into a hierarchical representation. They have made great progress in both representation and inference (i.e. parsing). Firstly, they have developed probabilistic grammars (e.g. stochastic context free grammar (SCFG) [1] and beyond [2]) which are capable of representing complex syntactic and semantic language phenomena. For example, speech contains elementary constituents, such as nouns and verbs, that can be recursively composed into a hierarchy of (e.g. noun phrase or verb phrase) of increasing complexity. Secondly, they have exploited the one-dimensional structure of language to obtain efficient polynomial-time parsing algorithms (e.g. the inside-outside algorithm [3]).

By contrast, the nature of images makes it much harder to design efficient image parsers which are capable of simultaneously performing segmentation (parsing an image into regions) and recognition (labeling the regions). Firstly, it is unclear what hierarchical representations should be used to model images and there are no direct analogies to the syntactic categories and phrase structures that occur in speech. Secondly, the inference problem is formidable due to the well-known complexity and ambiguity of segmentation and recognition. Unlike most languages (Chinese is an exception), whose constituents are well-separated words, the boundaries between different image regions are usually highly unclear. Exploring all the different image partitions results in combinatorial explosions because of the two-dimensional nature of images (which makes it impossible to order these partitions to enable dynamic programming). Overall it has been hard to adapt methods from natural language parsing and apply them to vision despite the high-level conceptual similarities (except for restricted problems such as text [4]).

Attempts at image parsing must make trade-offs between the complexity of the models and the complexity of the computation (for inference and learning). Broadly speaking, recent attempts can be divided into two different styles. The first style emphasizes the modeling problem and develops stochastic grammars [5], [6] capable of representing a rich class of visual relationships and conceptual knowledge about objects, scenes, and images. Zhu and Mumford [7] show the representation richness of this approach and discuss the relationship between grammars for images and objects. This style of research pays less attention to the complexity of computation. Parsing is performed by MCMC sampling and is only efficient provided effective proposal probabilities can be designed [5][6]. The second style builds on the success of conditional random fields (CRF's) [8] [9], [10] and emphasizes efficient computation. This yields simpler (discriminative) models which are less capable of representing complex image structures and long range interactions. Efficient inference (e.g. belief propagation and graph-cuts) and learning (e.g. AdaBoost, MLE) are available for basic CRF's and make these methods attractive. But these inference algorithms become less effective, and can fail, if we attempt to make the CRF models more powerful. For example, TextonBoost [10] requires the parameters of the CRF to be tuned manually. Overall, it seems hard to extend the CRF style methods to include long-range relationships and contextual knowledge without significantly altering the models and the algorithms.

In this paper, we introduce Hierarchical Image Models (HIM)'s for image parsing. HIM's balance the trade-off between model and inference complexity by introducing a hierarchy of hidden states. In particular, we introduce *recursive segmentation and recognition templates* which represent complex image knowledge and serve as elementary constituents analogous to those used in speech. As in speech, we can recursively compose these constituents at lower levels to form more complex constituents at higher level. Each node of the hierarchy corresponds to an image region (whose size depends on the level in the hierarchy). The state of each node represents both the partitioning of the corresponding region into segments and the labeling of these segments (i.e. in terms of objects). Segmentations at the top levels of the hierarchy give coarse descriptions of the image which are refined by the segmentations at the lower levels. Learning and inference (parsing) are made efficient by exploiting the hierarchical structure (and the absence of loops). In short, this novel architecture offers two advantages: (I) Representation – the hierarchical model using segmentation templates is able to capture long-range dependency and exploiting different levels of contextual information, (II) Computation – the hierarchical tree structure enables rapid inference (polynomial time) and learning by variants of dynamic programming (with pruning) and the use of machine learning (e.g. structured perceptrons [11]).

To illustrate the HIM we implement it for parsing images and we evaluate it on the public MSRC image dataset [10] and the PASCAL VOC dataset [12]. Our results show that the HIM perform at the state-of-the-art. We discuss ways that HIM's can be extended naturally to model more complex image phenomena. A preliminary version of this work was presented in [13].

II. Background: Image Models, Representations and Computation

The importance of image representations has long been realized and classic representations include the primal sketch [14], the 2-1/2D sketch [14], intrinsic images [15], and the 2.1D sketch [16]. These representations, however, often did not address computational issues such as inference and learning.

In this section we will concentrate on probabilistic models of images which are, to some extent, inspired by these classic representations (we will not deal with models that depend on depth estimates like the 2-1/2D sketch and intrinsic images). These probability models have three major components:

(I) The *representation* which consist of the graph structure, the types of random variables, and the form of the potentials represented at the cliques of the graph.

(II) The algorithms used for *inference*.

(III) The algorithms used to *learn* the probability distribution.

Table (I) gives a taxonomy of the standard probabilistic models for representing images based on these components. The representation is arguably the most important component because it not only determines the representational power of the model, but also the computational complexity of the inference algorithm (by the topology of the graph). In turn, the complexity of learning is strongly dependent on the complexity of the inference algorithm, since all learning algorithms require the ability to do inference. Therefore, all three components are highly related to each other and choice of representation is fundamental because it sets the stage on which inference and learning perform.

More concretely, the probabilistic models are defined over an image *representation* W and the input image $\mathbf{I}$. The image is specified by values $\{I_\mu : \mu \in \mathcal{D}\}$ defined over the image lattice $\mathcal{D}$. The nature of the image representation W differs greatly for different models as described in the following subsection. For some models (e.g., MRFs and CRFs) W is specified over a copy of the image lattice, while for others (e.g, stochastic grammars) it can have arbitrarily complicated, and variable, topology. More generally, W is defined by a set of values $\{W_\nu : \nu \in \mathcal{V}\}$ where $\mathcal{V}$ is the graph structure of the representation (which can have variable topology). The probability distributions relating $\mathbf{I}$ to W are either *generative* – specified by a likelihood function $P(\mathbf{I}|W)$ and a prior $P(W)$ – or are *discriminative* and specified directly by $P(W|\mathbf{I})$. In either case, the distributions are specified over the random variables $\{I_\mu, W_\nu\}$ defined on a graph $\mathcal{D} \bigcup \mathcal{V}$. The probabilities are determined by clique potentials, where the cliques are specified by the edges $\mathcal{E}$ of the graph $\mathcal{D} \bigcup \mathcal{V}$. The *learning* algorithms are required to estimate the distributions – $P(\mathbf{I}|W)$, $P(W)$, or $P(W|\mathbf{I})$ – from training data. Most learning algorithms assume that the graph structure is known and so only learn the distributions, while others attempt to learn the distributions and the graph structure. The *inference* algorithms have an image $\mathbf{I}$ as input and attempt to estimate W by, for example, maximizing $P(W|\mathbf{I})$. The computational complexity of inference and learning depends largely on the graph structure although the nature of the variables is also important.

In the last three decade, researchers in computer vision have strived to develop rich representations which are capable of encoding visual entities (textures, regions, boundaries, objects, scenes, etc.) and their relations while making computation (learning and inference) tractable. Probabilistic models of images in the literature can be classified by the following three questions (see table II): (i) are they generative or discriminative (or some combination)? (ii) do they represent image regions or only edges? and (iii) are they hierarchical (e.g., have deep topological structure) or shallow (like standard MRFs)? This classification is only rough since, for example, the distinction between generative and discriminative models is increasingly being blurred as more sophisticated discriminative models are developed. We will use this classification to review the relevant literature to give context before we present our approach.

Classification of Probabilistic Models of Images
Generative vs. Discriminative
Region vs. Edges
Shallow vs. Deep

TABLE II

A ROUGH CLASSIFICATION FOR MODELS OF IMAGES. THE MODELS CAN BE GENERATIVE – I.E. SPECIFY A DISTRIBUTION FOR GENERATING THE IMAGE – OR DISCRIMINATIVE (THIS DISTINCTION IS BLURRED BECAUSE, FOR EXAMPLE, SOME DISCRIMINATIVE MODELS CAN BE CONSIDERED AS GENERATIVE MODELS FOR IMAGE FEATURES). MODELS CAN BE DISTINGUISHED BETWEEN WHETHER THEY TRY TO DECOMPOSE THE IMAGE INTO DISJOINT REGIONS OR SIMPLY LABEL INTENSITY EDGES AND OTHER SIGNIFICANT IMAGE "EVENTS". MODELS CAN HAVE SHALLOW TOPOLOGY (ONE OR TWO LAYERS) OR BE DEEP AND HAVE MANY LAYERS. DEEP MODELS ARE BETTER ABLE TO REPRESENT LONG RANGE INTERACTIONS.

A. A Taxonomy of Probabilistic Models of Images

Markov Random Field models (MRF's) [17] are among the earliest probabilistic models of images (see also Blake and Zisserman [20]). They *represent* the image $\mathbf{I}$ by a smoothed image f with horizontal h and vertical edges v, hence we set $W = (f, h, v)$. f, h, v are defined on lattices $\mathcal{D}_f, \mathcal{D}_h, \mathcal{D}_v$ which are copies of the image lattice $\mathcal{D}$, so we write $f = \{f_\mu : \mu \in \mathcal{D}_f\}$, $h = \{h_\mu : \mu \in \mathcal{D}_h\}$, $v = \{v_\mu : \mu \in \mathcal{D}_v\}$, defined on a graph $\mathcal{V} = \mathcal{D}_f \bigcup \mathcal{D}_h \bigcup \mathcal{D}_v$. The generative model is of factorized form $P(\mathbf{I}|W) = \prod_{i \in \mathcal{D}} P(I_i|f_i)$ and the prior distribution $P(W) \propto \exp \sum_c \phi(f_c, h_c, v_c)$ where the $\phi(.)$ are the clique potentials defined over local neighborhoods. As shown by Geiger and collaborators [21], [22] the horizontal and vertical edges

Models	Representation	Learning	Computation
MRF [17]	one layer; generative	maximum entropy	MCMC, Mean Field Theory
Conditional Random Fields [9], [10]	one layer; discriminative	boosting + MLE	belief propagation / graph Cut
primal sketch [18]	one layer; generative	sparse learning	greedy search
region competition [19] and image parsing [6]	shallow; generative	separate learning	MCMC
stochastic grammar [7]	deep; generative	MCMC	MCMC
recursive templates	deep; discriminative	structure-perceptron	dynamic programming

TABLE I

A TAXONOMY OF PROBABILISTIC MODELS OF IMAGES

can be summed out (for certain $\phi(.,.,.)$) to yield a model where W corresponds only to the smoothed image f, and where the edges can be inferred from the estimates of f. A variety of *inference* algorithms have been defined for MRF models. These include stochastic sampling and deterministic annealing [17], mean field methods [23] (now renamed as variational methods), EM algorithms [22], and graduated non-convexity [20]. Attempts to learn these models have made the fundamental assumption that we can directly observe W and learn a probability distribution over it by applying minimax entropy theory [24], which applies the maximum entropy principle to histograms $\phi(W)$ of feature statistics computed from W. The maximum entropy principle leads to an exponential distribution of form $P(W) = \frac{1}{Z\alpha} \exp\{\alpha \cdot \phi(W)\}$, where the parameters α are chosen such that $\sum_W P(W)\phi(W) = \psi$, and ψ are the observed statistics. For minimax entropy, there is a dictionary of features and the theory selects those which best fit the data by the maximum likelihood criteria. If the region is sufficiently big, then the only possible states of W with non-zero probability are those that have the exact histogram – i.e. $\phi(W) = \psi$ (like the asymptotic partition criterion from information theory [25]). In practice, W is not directly observable so the distributions are learnt by using images $\mathbf{I}$ as input. A related theory for learning probability distributions from features was developed at the same time by Della Pietra *et al* [26] but not applied to images. See Roth and Black [27] for more recent attempts to learn image priors by mixtures of experts.

By contrast to generative MRF's, *discriminative methods* encode the posterior distribution $P(W|\mathbf{I})$ directly, by ignoring the generative process. Discriminative models have been used for labeling problems where $W = \{w_\mu : \mu \in \mathcal{D}_w\}$ is a set of discrete labels w_μ defined on a copy $\mathcal{D}_w$ of the image lattice $\mathcal{D}$. The labels can correspond, for example, to indicating a pixel $\mu \in \mathcal{D}$ is an "edge" or "non-edge", or they can be used to label pixels as "sky", "vegetation", "road", and "other". Discriminative models attempt to model $P(W|\mathbf{I})$ directly without having explicit models for $P(\mathbf{I}|W)$ and $P(W)$. The simplest models of this type can be expressed as: $P(\{w_\mu\}|\mathbf{I}) = \prod_{\mu \in \mathcal{D}} P(w_\mu|\mathbf{I})$, where the distribution of the label of pixel $\mu \in \mathcal{D}$ is conditionally independent of the label of its neighborhood pixels. For such models, *inference* and *learning* are easy since they can be performed at each pixel independently [28]. But this independence assumption is very restrictive and can be relaxed to include interactions between the labels at different pixels. This can be done using the technology of conditional random fields [8] though the relaxation labeling theories of Rosenfeld and his collaborators had similar intuitions [29]. This leads to models of form $P(\{w_\mu\}|\mathbf{I}) = \frac{1}{Z} \prod_{\mu \in \mathcal{D}} \phi(w_\mu|\mathbf{I}) \prod_{(\mu,\nu) \in \mathcal{E}} \phi(w_\mu, w_\nu|\mathbf{I})$, where Z is a normalization term. Such models have been applied to labeling images and detecting buildings [30]. For these models inference and learning are more difficult and, as for MRFs, a variety of techniques have been proposed. Since the labels are discrete (by contrast, the f in the Geman and Geman model are continuous) discrete algorithms like max-flow and belief propagation have been applied for *inference* and maximum likelihood for *learning*.

The generative and discriminative MRFs discussed in the previous two paragraphs model local image properties – i.e. image appearance of pixels – and make local prior assumptions, such as weak smoothness. The next two classes of model – the primal sketch [18] and regional models [19], [5], [6] – attempt to model richer image properties and have longer range interactions.

The Primal Sketch model [18] was partially motivated by Marr's primal sketch, whose goal was to make explicit important information about image properties including spatial organization. Its starting point is the sparse coding model of images proposed by Olshausen and Field [31] which represents an image $\mathbf{I}$ by a linear sum of basis functions $\{B_i(\mu)\}$ plus a noise term: $\mathbf{I}(\mu) = \sum_{i=1}^{N} \alpha_i B_i(\mu) + \epsilon(\mu)$, with a "sparseness prior" put on the coefficients α so that $\sum_{i=1}^{N} |\alpha_i|$ is small, and the noise term $\epsilon(\mu)$ is assumed to be independent zero mean additive Gaussian noise. Guo *et al* [18] developed this into a model of images by putting a more sophisticated prior on the α including interactions between the coefficients of neighboring basis functions. The basis functions correspond to "sketchable" parts of the image such as edges, peaks, and valleys (which can be identified by thresholding the α). "Non-sketchable" regions can be represented by Julesz ensembles [32] (i.e. histograms of image features). Inference and learning are harder for the full primal sketch model. Projection pursuit [33] can be used if the interactions between the α's can be neglected, see [18] for details.

Regional models decompose the image domain $\mathcal{D}$ into disjoint regions $\mathcal{D} = \bigcup_{a=1}^{M} \mathcal{D}_a$, with $\mathcal{D}_a \bigcap \mathcal{D}_b = \emptyset \ \forall \ a \neq b$ and where the number of regions M is a random variable. An early example is Mumford and Shah [34] where the W corresponds to a smoothed version of the image (somewhat similar to [17]). A more general formulation ([19], [5], [6]) specifies a class of probability models $P(\mathbf{I}_{\mathcal{D}_a}|\tau_a, \gamma_a)$ for representing the image intensity $\mathbf{I}_{\mathcal{D}_a}$ within each subregion $\mathcal{D}_a$, where τ_a labels the model type and γ_a labels its parameters – e.g., τ could label the region as 'texture' and γ would specify the parameters of the texture model. Hence the *representation* is $W = (M, \{(\mathcal{D}_a, \tau_a, \gamma_a) : a = 1, ..., M\})$. The likelihood model is of form $P(\mathbf{I}|M, \{\mathcal{D}_a, \tau_a, \gamma_a\}) = \prod_{a=1}^{M} P(\mathbf{I}_a|\tau_a, \gamma_a)$. There is a prior probability on W specified by $P(M)P(\{\mathcal{D}_a\}|M) \prod_a P(\tau_a)P(\gamma_a)$. *Inference* is difficult for these models. For the simpler versions [34], [19] a variety of algorithms will work (because this model assumes that the intensity properties within all regions only obey smoothness). For the more complex versions [5],[6] stochastic sampling by Data Driven Markov Chain Monte Carlo (DDMCMC) is required. *Learning* the distributions $P(\mathbf{I}_{\mathcal{D}_a}|\tau_a, \gamma_a)$ is easier once images have been hand-labelled.

Although the regional models are able to have some long range interactions (due to the size of the image regions) they are of fairly simple form, partially because of their shallow graph structure. They cannot, for example, represent the spatial relations between windows in a building. *Probabilistic Grammars* [35], [7] have been proposed to model visual entities at different levels and their short range and long range interactions. These models are attractive and we refer to Zhu and Mumford [7] for details of the *representations* which may be achieved by these methods. The *inference* relies on stochastic sampling by Data Driven Markov Chain Monte Carlo (DDMCMC) [5] and the *learning* uses minimax entropy methods with hand-labelled data. Several of the concepts of probabilistic grammars can be illustrated by the stochastic context free grammars (SCFG) used in natural language processing [36]. A SCFG consists of sets of non-terminal and terminal nodes, a set of production rules, and a probability distribution defined over the rules. For natural languages, the non-terminal nodes can be $S, NP, VP, AT, NNS, VBD, PP, IN, DT, NN$ where S is a sentence, VP is a verb phrase, VBD is a verb,NP is a noun phrase, NN is a noun, and so on [36]. The terminal nodes are words from a dictionary (e.g. "the", "cat", "sat", "on", "mat".) The production rules are applied to non-terminal nodes to generate child nodes (e.g. $S \mapsto NP, VP$ or $NN \mapsto$ "cat") and probabilities are defined over these rules. To generate a sentence we start from the root node S, sample to select a production rule and apply it to generate child nodes. We repeat this process on the child nodes and stop when all the nodes are terminal (i.e. all are words). To parse an input sentence, we use dynamic programming to compute the most probable way the sentence could have been generated by the production rules.

Probabilistic grammars seems promising by offering the rich representational power by modeling complex knowledge in a deep structure. However, applying probabilistic grammars to images is not straightforward. The major challenges are : (i) what are the corresponding syntactic categories and phrase structures in the image domain? (ii) can we design an efficient inference algorithm on 2D image space to make model learning and computing tractable? Our recursive segmentation and recognition templates are proposed to address these two critical issues.

Notation	Meaning
$\mathbf{I}$	input image
W	parse tree
μ, ν	node index
$Ch(\mu)$	child nodes of μ
s	segmentation template
c	object class
$\psi_1(\mathbf{I}, s_\mu, c_\mu)$	object class appearance potential
$\psi_2(\mathbf{I}, s_\mu, c_\mu)$	appearance homogeneity potential
$\psi_3(s_\mu, c_\mu, s_\nu, c_\nu)$	layer-wise labeling consistency potential
$\psi_4(i, j, c_\mu, c_\nu)$	object class co-occurrence potential
$\psi_5(s_\mu)$	segmentation template potential
$\psi_6(s_\mu, j)$	co-occurrence of segment and class potential

TABLE III

THE TERMINOLOGY USED IN THE HIM MODEL.

III. HIERARCHICAL IMAGE MODEL

A. The Representation

We represent an image by a hierarchical graph $\mathcal{V}$ with edges $\mathcal{E}$ defined by parent-child relationships, see figure (1). The hierarchy corresponds to an image pyramid (with 5 layers in this paper) where the top node of the hierarchy represents the whole image. The intermediate nodes represent different sub-regions of the image and the leaf nodes represent local image patches (27×27 in this paper).

The notations we used to describe the model are summarized in table III. We use $\mu \in \mathcal{V}$ to index nodes of the hierarchy. $\mathcal{R}$ denotes the root node, $\mathcal{V}^{LEAF}$ are the leaf nodes, $\mathcal{V}/\mathcal{V}^{LEAF}$ are all nodes except the leaf nodes, and $\mathcal{V}/\mathcal{R}$ are all nodes except the root node. A node μ has a unique parent node denoted by $Pa(\mu)$ and four child nodes denoted by $Ch(\mu)$. Thus, the hierarchy is a quad tree and $Ch(\mu)$ encodes all its vertical edges $\mathcal{E}$. The image region represented by node μ is fixed and denoted by $R(\mu)$, while pixels within $R(\mu)$ are labeled by r. The set of pairs of neighbor pixels in $R(\mu)$ is denoted by $\mathcal{E}(\mu)$.

A configuration of the hierarchy is an assignment of state variables $W = \{w_\mu\}$ to the nodes $\mu \in \mathcal{V}$ (all state variables are unobservable and must be inferred). The state variables are of form $w_\mu = (s_\mu, \vec{c}_\mu)$, where s and $\vec{c}$ specify the *segmentation template* and the object *label* respectively. We call $(s, \vec{c})$ a *Segmentation and Recognition* pair, which we abbreviate to an *S-R pair.* They provide a description of the image region $R(\mu)$. Each segmentation template partitions a region into K non-overlapping sub-regions and is selected from a dictionary D_s, where $K \leq 3$ and $|D_s| = 30$ in this paper. The dictionary of segmentation templates, see figure (1), was designed by hand to cover the taxonomy of shape segmentations that happen in images, such as T-junctions, Y-junctions, and so on. We divide the segmentation templates into three disjoint subsets S_1, S_2, S_3, where $\bigcup_{K=1}^{3} S_K = D_s$, so that templates in subset S_K partition the image into K subregions. The variable $\vec{c} = (c_1, ..., c_K)$, where $c_K \in \{1, ..., M\}$, specifies the labels of the K subregions (i.e. labels one subregion as "horse" another as "dog" and another as "grass"). The number M of labels is set to 21 in this paper. The label of a pixel r in region $R(\mu)$ is denoted by $o_\mu^r \in \{1..M\}$ and is computed directly from $s_\mu, \vec{c}_\mu$, hence any two pixels within the same subregion must have the same label. Observe that each image pixel will have labels o_μ^r defined at all levels of the hierarchy, which will be encouraged (probabilistically) to be consistent.

We emphasize that these hierarchical *S-R pairs* are a novel aspect of our approach. They explicitly represent the segmentation and the labeling of the regions, while more traditional vision approaches [10], [9], [37] use labeling only. Intuitively, the hierarchical S-R pairs provide a coarse-to-fine representation which capture the "gist" (e.g., semantical meaning) of the image regions at different levels of resolution. One can think of the S-R pairs at the highest level as providing an "executive summary" of the image, while the lower S-R pairs provided more detailed (but still summarized) descriptions of the image subregions. This is illustrated in figure (2), where the top-level S-R pair shows that there is a horse with grass background, mid-level S-R pairs give a summary description of the horses leg as a triangle, and lower-level S-R pairs give more accurate descriptions of the leg. We will show this approximation quality empirically in section (VI).

B. The distribution

The conditional distribution over the state variables $W = \{w_\mu : \mu \in \mathcal{V}\}$ is given by:

$$\begin{aligned} p(W|\mathbf{I}; \alpha) = \frac{1}{Z(\mathbf{I}; \alpha)} \exp\{ & -E_1(\mathbf{I}, s, c; \alpha_1) \\ & - E_2(\mathbf{I}, s, c; \alpha_2) - E_3(s, c; \alpha_3) - E_4(c; \alpha_4) \\ & - E_5(s; \alpha_5) - E_6(s, c; \alpha_6)\} \qquad (1) \end{aligned}$$

where $\mathbf{I}$ is the input image, W is the parse tree, α are the parameters to be estimated, $Z(\mathbf{I}; \alpha)$ is the partition function and $E_i(\mathbf{I}, W)$ are energy terms defined

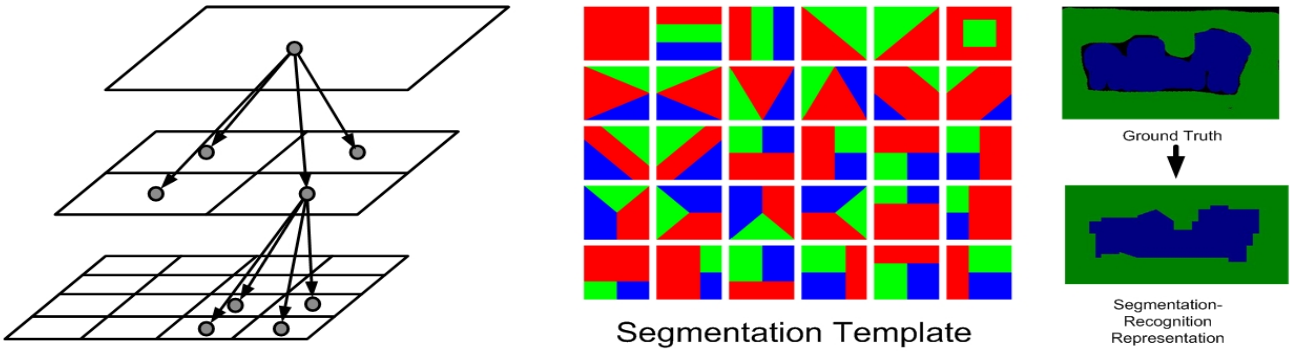


Fig. 1. The left panel shows the structure of the Hierarchical Image Model. The grey circles are the nodes of the hierarchy. All nodes, except the top node, have only one parent nodes. All nodes except the leafs are connected to four child nodes. The middle panel shows a dictionary of 30 segmentation templates. The color of the sub-parts of each template indicates the object class. Different sub-parts may share the same label. For example, three sub-parts may have only two distinct labels. The last panel shows that the ground truth pixel labels (upper right panel) can be well approximated by composing a set of labeled segmentation templates (bottom right panel).

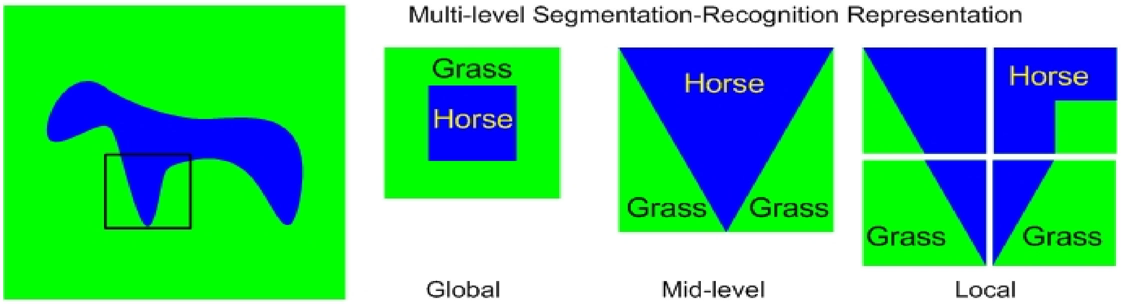


Fig. 2. This figure illustrates how the segmentation templates and object labels (S-R pair) represent image regions in a coarse-to-fine way. The left figure is the input image which is followed by global, mid-level and local S-R pairs. The global S-R pair gives a coarse description of the object identity (horse), its background (grass), and its position in the image (central). The mid-level S-R pair corresponds to the region bounded by the black box in the input image. It represents (roughly) the shape of the horse's leg. The four S-R pairs at the lower level combine to represent the same leg more accurately.

below. Equivalently, the conditional distribution can be reformulated in a log-linear form:

$$\log p(W|\mathbf{I};\alpha) = \alpha \cdot \psi(\mathbf{I}, W) - \log Z(\mathbf{I};\alpha) \qquad (2)$$

Each energy term is expressed in linear form, $E_i(\mathbf{I}, W) = -\alpha_i \cdot \psi_i(\mathbf{I}, W)$, where the inner product is between a parameter α (which will be learnt) and a potential functions ψ. There are six types of energy terms defined as follows.

The first term $E_1(\mathbf{I}, s, c)$ is an object specific data term which represents the image features of regions. We set $E_1(\mathbf{I}, s, c) = -\sum_{\mu \in \mathcal{V}} \alpha_1 \psi_1(\mathbf{I}, s_\mu, \vec{c}_\mu)$ where $\sum_{\mu \in \mathcal{V}}$ is the sum over all nodes at all levels of the hierarchy, and $\psi_1(\mathbf{I}, s_\mu, \vec{c}_\mu)$ is of form:

$$\psi_1(\mathbf{I}, s_\mu, \vec{c}_\mu) = \frac{1}{|R(\mu)|} \sum_{r \in R(\mu)} \log p(o_\mu^r | \mathbf{I}^r) \qquad (3)$$

where $\mathbf{I}^r$ is a local image region centered at the location of r, $F(\cdot,\cdot)$ is a (strong) classifier learnt by multi-class boosting [38] and $p(o_\mu^r|\mathbf{I}^r)$ is given by:

$$p(o_\mu^r | \mathbf{I}^r) = \frac{\exp\{F(\mathbf{I}^r, o_\mu^r)\}}{\sum_{o'} \exp\{F(\mathbf{I}^r, o')\}} \qquad (4)$$

The details of image features and boosting learning will be described in section (VI-A.2).

The second term $E_2(\mathbf{I}, s, c) = -\sum_{\mu \in \mathcal{V}} \alpha_2 \psi_2(\mathbf{I}, s_\mu, c_\mu)$ is designed to favor segmentation templates for which the pixels belonging to the same partitions (i.e., having the same labels) have similar appearance. We define:

$$\psi_2(\mathbf{I}, s_\mu, \vec{c}_\mu) = \frac{1}{|E(\mu)|} \sum_{(q,r) \in E(\mu)} \phi(\mathbf{I}^r, \mathbf{I}^q | o_\mu^r, o_\mu^q) \qquad (5)$$

where $E(\mu)$ are the set of edges connecting pixels q, r in a neighborhood and $\phi(\mathbf{I}^r, \mathbf{I}^q | o_\mu^r, o_\mu^q)$ has the form of

$$\phi(\mathbf{I}^r, \mathbf{I}^q | o^r_\mu, o^q_\mu) = \left\{ \begin{array}{ll} \gamma(r,q) & if\ o^r_\mu = o^q_\mu \\ 0 & if\ o^r_\mu \neq o^q_\mu \end{array} \right. \quad (6)$$

where $\gamma(r,q) = \lambda \exp\{-\frac{g^2(r,q)}{2\gamma^2}\}\frac{1}{dist(r,q)}$, $g(.,.)$ is a distance measure on the colors $\mathbf{I}^r, \mathbf{I}^q$ and $dist(r,q)$ measures the spatial distance between r and q. $\phi(\mathbf{I}^r, \mathbf{I}^q | o^r_\mu, o^q_\mu)$ is so called the contrast sensitive Potts model which is widely used in graph-cut algorithms [39] as edge potentials (only in one level) to favors pixels with similar colour having the same labels.

The third term, defined as:

$$E_3(s,c) = -\sum_{\mu \in \mathcal{V}/\mathcal{R}: \nu = Pa(\mu)} \alpha_3 \psi_3(s_\mu, \vec{c}_\mu, s_\nu, c_\nu) \quad (7)$$

(i.e. we sum over all nodes μ – except the root node – with ν being the parent of μ) is used to encourage consistency between the S-R pairs at consecutive levels of the hierarchy. The potential $\psi_3(s_\mu, c_\mu, s_\nu, c_\nu)$ is defined by the Hamming distance:

$$\psi_3(s_\mu, \vec{c}_\mu, s_\nu, \vec{c}_\nu) = \frac{1}{|R(\mu)|} \sum_{r \in R(\mu)} \delta(o^r_\mu, o^r_\nu) \quad (8)$$

where $\delta(o^r_\mu, o^r_\nu)$ is the Kronecker delta, which equals one whenever $o^r_\mu = o^r_\nu$ and zero otherwise. The hamming function ensures to glue the segmentation templates (and their labels) at different levels together in a consistent hierarchical form. This energy term is a generalization of the interaction energy in the Potts model. However, $E_3(s,c)$ has a hierarchical form which allows multi-level interactions.

The fourth term $E_4(c)$ is designed to model the co-occurrence of two object classes (e.g., a cow is unlikely to appear next to an aeroplane):

$$E_4(c) = -\sum_{\mu \in \mathcal{V}} \sum_{i,j=1..M} \alpha_4(i,j) \psi_4(i,j,c_\mu,c_\mu) - \sum_{\mu \in \mathcal{V}/\mathcal{R}: \nu = Pa(\mu)} \sum_{i,j=1..M} \overline{\alpha}_4(i,j) \psi_4(i,j,c_\mu,c_\nu) \quad (9)$$

where $\psi_4(i,j,c_\mu,c_\nu)$ is an indicator function which equals one while $i \equiv c_\mu$ and $j \equiv c_\nu$ ($i \equiv c_\mu$ means i is a component of c_μ) is true and is zero otherwise. α_4 is a matrix where each entry $\alpha_4(i,j)$ encodes the compatibility between two classes i and j at the same level. Similarly $\overline{\alpha}_4(i,j)$ gives the compatibility between classes at different levels. In other words, the first term on the right hand side encodes the class co-occurrences within a single template while the second term encodes the class co-occurrence between parent and child templates. Note that class co-occurrence is encoded at all levels to capture both short-range and long-range interactions.

The fifth term $E_5(s) = -\sum_{\mu \in \mathcal{V}} \alpha_5 \psi_5(s_\mu)$, where $\psi_5(s_\mu) = \log p(s_\mu)$ encode the generic prior of the segmentation template.

Similarly the sixth term $E_6(s,c) = -\sum_{\mu \in \mathcal{V}} \sum_{j \equiv c_\mu} \alpha_6 \psi_6(s_\mu, j)$, where $\psi_6(s_\mu, j) = \log p(s_\mu, j)$, models the co-occurrence of the segmentation templates and the object classes. $\psi_5(s_\mu)$ and $\psi_6(s_\mu, j)$ are directly obtained from training data by label counting. The parameters α_5 and α_6 are both scalars.

C. How to classify HIMs?

We now describe how HIM fits into our classification shown in table (II). Firstly, HIM is a discriminative model because it specifies $P(W|\mathbf{I})$ directly and contains no model for generating the image. The model contains energy terms – E_3, E_4, E_5, E_6 – which are independent of the image $\mathbf{I}$ and can be loosely considered to specify a prior on the segmentation templates and class labels. More specifically, E_3 encourages consistent between the templates and labels at different levels of the hierarchy, E_4 captures the statistics of the co-occurrence of labels, E_5 gives a prior for the templates, and E_6 describes the co-occurrence of the templates and the classes. More standard models for discriminative classification only include models for the co-occurrence of classes, since they do not use segmentation templates (even though some use hierarchies [9], but use fewer than the five levels used by HIMs). Note that in HIM spatial smoothness of object labels is imposed by the hierarchy meaning that neighboring nodes have similar labels because they are encouraged to be consistent with their parents, instead of being encouraged to directly have similar labels to their neighbours (as in more standard "shallow" models). The energy terms E_1, E_2 specify how HIM interacts with the image. More specifically, E_1 models the image labelling (using machine learning classifiers) and E_2 is like a "data dependent prior" which encourages neighboring regions to have similar labels unless there is a large intensity discontinuity. HIM will be learnt by a discriminative learning method (see section V). Secondly, HIM explicitly represents both regional and edge properties by the segmentation templates as shown in figure (1). Note that some of the templates represent triple points which are used to indicate occlusion and foreground/background relationships, see fourth row of the middle panel of figure (1). Thirdly, the structure of HIM's is deep (with five levels) so that short, medium, and long range interactions between the object regions of different sizes are encoded at different levels of the hierarchy.

D. The Summarization Principle

An important aspect of our Hierarchical Image Model (HIM), which distinguishes it from most other models, is the summarization principle. This design principle is important both for representation and to make computation tractable. It is partially based on the intuition of *executive summary* that nodes at the upper levels of the hierarchy need only provide coarse descriptions of the image because more detailed descriptions can be obtained at lower levels. This intuition relates to Lee and Mumford's [40] high resolution buffer hypothesis for the visual cortex.

The summarization principle has four aspects.

(I) The state of w_ν the random variable at node $\nu \in \mathcal{V}/\mathcal{V}^{LEAF}$ is a summary of the state of its child nodes $\mu \in ch(\nu)$, and hence summarizes their states, see figure (2).

(II) The representational complexity of a node is the same at all levels of the tree – the random variables are restricted to take the same number of states.

(III) The clique potentials for a node $\nu \in \mathcal{V}$ depends on its parent nodes and it child nodes, but not on its grandparents or grandchildren. This is a Markov property on the hierarchy. But, as will be described later, all nodes can receive input directly from the input image.

(IV) The potentials defined over the cliques depend only on simple statistics which also summarize the states of the child nodes.

The executive summary intuition is enforced by aspects (I) and (II) – the upper levels nodes can only give coarse descriptions of the large image regions that they represent and these descriptions are based on the, more detailed, descriptions given by the lower level nodes. The other two aspects – (III) and (IV) – help reduce the number of cliques in the graph and restrict the complexity of the potentials defined over the cliques. Taken all together, the four aspects make learning and inference computationally practical because of (i) the small clique size, (ii) the simplicity of the potentials, and (iii) the limited size of the state space.

E. Comparisons with other work

The HIM has several partial similarities with other work. HIM is a coarse-to-fine representation which captures the "gist" of image regions by using the S-R pairs at multiple levels. But the traditional concept of "gist" [41] relies only on image features and does not include segmentation templates. Levin and Weiss [42] use a segmentation mask which is more object-specific than our segmentation templates (and they do not have a hierarchy). It is worth nothing that, in contrast to TextonBoost [10], we do not use "location features" in order to avoid the dangers of overfitting to a restricted set of scene layouts. Our approach has some similarities to some hierarchical models (which have two-layers only) [9],[37] – but these models also lack segmentation templates. The hierarchial model proposed by [43] is an interesting alternative but which does not perform explicit segmentation.

IV. Inference: Parsing by Dynamic Programming

Parsing an image is performed as inference of the HIM. We parse the image by inferring the maximum a posterior (MAP) estimator of the HIM:

$$W^* = \arg\max_W p(W|\mathbf{I};\alpha) = \arg\max_W \alpha \cdot \psi(\mathbf{I}, W) \tag{10}$$

This will output state variables $\{w^*_\mu = (s^*_\mu, \vec{c}^*_\mu) : \mu \in \mathcal{V}\}$ at all levels of the hierarchy. But we only use the state variables at the lowest level of the graph when we evaluate the HIM for labeling.

The graph of the HIM has no closed loops so Dynamic Programming (DP) can be applied to calculate the best parse tree W^* from equation (10). But the computational complexity is high because of the large size of the state space. To see this, observe that the number of states at each node is $O(M^K|D_s|)$ (where $K = 3, M = 21, |D_s| = 30$) and so the computational complexity is $O(M^{2K}|D_s|^2 H)$ where H is the number of edges in the hierarchy.

Note that the choice of our representation, in particular the segmentation-recognition template, has restricted the size of the state space by requiring that node μ can only assign labels o^r_μ consistent with the state $w_\mu = (s_\mu, \vec{c}_\mu)$. Nevertheless, the computational complexity means that DP is still impractical on standard PCs. We hence use a pruned version of DP which will describe below.

A. Recursive Energy Formulation

The hierarchical form of the HIM (without closed loops) means that the energy terms can be computed recursively which will enable Dynamic Programming and motivate pruning.

More formally, an HIM is defined over a hierarchical graph $\mathcal{V}$ with edges $\mathcal{E}$ specified by the parent-child relations. The graph has no closed loops since each child is constrained to have a single parent. The state variables are $w_\mu = (s_\mu, \vec{c}_\mu)$ for node $\mu \in \mathcal{V}$. The distribution is of Gibbs form with an energy function that can be expressed as:

$$E(w|\mathbf{I}) = \sum_{\mu\in\mathcal{V}} \alpha_\mu \cdot \psi_\mu(\mathbf{I}, w_\mu)$$
$$+ \sum_{\mu\in\mathcal{V}/\mathcal{V}^{LEAF}} \alpha^{int}_\mu \cdot \psi^{int}_\mu(w_\mu, w_{ch(\mu)}), \qquad (11)$$

where: (I) the first term depends only on the states of the nodes at each level and on the input image $\mathbf{I}$, which includes the data terms E_1, E_2 and two of the prior terms E_5, E_6. and (II) the second term E_3, E_4 depends on the states of nodes and their children, and the superscript ^{int} stands for "inter-layer".

We express this energy function recursively, exploiting the tree structure, by defining an energy function $E_\nu(w_{des(\nu)}|\mathbf{I})$ over the subtree with root node ν in terms of the state variables $w_{des(\nu)}$ of the subtree, where $des(\nu)$ stands for the set of descendent nodes of ν – i.e. $w_{des(\nu)} = \{w_\mu : \mu \in \mathcal{V}_\nu\}$, where $\mathcal{V}_\nu$ is the subtree with root node ν. We define $E_\nu(w_{des(\nu)}|\mathbf{I})$ by:

$$E_\nu(w_{des(\nu)})|\mathbf{I}) = \sum_{\mu\in\mathcal{V}_\nu} \alpha^{int}_\mu \cdot \psi^{int}_\mu(w_\mu, w_{ch(\nu)})$$
$$+ \sum_{\mu\in\mathcal{V}_\nu} \alpha_\mu \cdot \psi_\mu(\mathbf{I}, w_\mu) \qquad (12)$$

which can be computed recursively by:

$$E_\nu(w_{des(\nu)}|\mathbf{I}) = \sum_{\rho\in ch(\nu)} E_\rho(w_{des(\rho)}|\mathbf{I})$$
$$+\alpha^{int}_\nu \cdot \psi^{int}_\nu(w_\nu, w_{ch(\nu)}) + \alpha_\nu \cdot \psi_\nu(\mathbf{I}, w_\nu). \qquad (13)$$

Observe that the full energy $E(W|\mathbf{I})$ is obtained by evaluating $E_\nu(.)$ at the root node $\mathcal{R}$.

B. Dynamic Programming with Pruning

We can use the recursive formulation of the energy, see equation (13), to perform Dynamic Programming. But to ensure rapid inference we will need to perform pruning by not exploring partial state configurations which seem unpromising. We first describe DP and then give our pruning strategy. The pseudocode for the algorithm is given in figure (3).

DP proceeds by evaluating possible states w_ν for nodes ν of the graph. We will refer to these possible states as *proposals* and denote them by $\{p_{\nu,b}\}$, where b indexes the proposal. These proposals, and their energies $E_\nu(p_{\nu,b}|\mathbf{I})$, are computed recursively as follows.

Recursion for parent nodes: to obtain the proposals for a parent node μ at a higher level of the graph $\mu \in \mathcal{V}/\mathcal{V}^{LEAF}$ we first access the proposals for all its child nodes $\{p_{\mu_i,b_i}\}$ where $\{\mu_i : i = 1,...,|ch(\mu)|\}$ denotes the set of child nodes of μ and their energies $\{E_{\mu_i}(p_{\mu_i,b_i}|\mathbf{I}) : i = 1,...,|ch(\mu)|\}$. Then we compute the states $\{p_{\mu,b}\}$ such that $E_\mu(p_{\mu,b}|\mathbf{I}) \le E_\mu(w_\mu|\mathbf{I})$ where:

$$E_\mu(p_{\mu,b}|\mathbf{I}) = min_{\{b_i\}}\{ \sum_{i=1}^{|ch(\mu)|} \{E_{\mu_i}(w_{des(\mu_i,b_i)}|\mathbf{I})$$
$$+\alpha^{int}_\mu \cdot \psi^{int}_\mu(p_{\mu,b}, \{p_{\mu_i,b_i}\})\} + \alpha_\mu \cdot \psi_\mu(\mathbf{I}, p_{\mu,b})\} \qquad (14)$$

The initialization is performed at the leaf nodes using the data terms only (E_1 and E_2).

The pruning strategy if to reject proposals whose energies are too high and which hence are unlikely to lead to the optimal solution. To understand our pruning strategy, recall that the set of of region partitions is divided in three subsets S_1, S_2, S_3, where S_i contains i regions. There are $|C|^i$ possible labels c for each region partition which gives a very large state space (since $|C| = 30$). Our pruning strategy is to restrict the set of labels $\vec{c}$ allowed for each of these subsets. For subset S_1, there is only one region so we allow all possible labels for it $c^1 \in C$ and perform no pruning. For subset S_2, there are two subregions and we keep only the best 10 labels for each subregion (i.e. a total of $10 \times 10 = 100$ labels). For subset S_3, we keep only the best 5 labels

```
Input: $\{p^1_{\nu^1}\}$. Output:$\{p^L_{\nu^L}\}$
  • Bottom-Up($p^1$)
    Loop : $l = 1$ to $L$, for each node $\nu$ at level $l$
      1) Composition: $\{p^l_{\nu,b}\} = \oplus_{\rho \in ch(\nu)} p^{l-1}_{\rho,a}$
      2) Pruning: divide the proposals $\{p^l_{\nu,b}\}$ into three sets based on the partition
         variable $s_\mu \in S_i$ $(i = 1, 2, 3)$. Rank all these proposals by energy. Keep the
         lowest $k_i$ proposals for each set $(k_1 = 30, k_2 = 100, k_3 = 125)$.
```

Fig. 3. The inference algorithm. $\oplus$ denotes the operation of combining proposals from the child nodes to make proposals for parent nodes. See text for more details about the pruning.

of each subregion (hence a total of $5^3 = 125$ labels). In summary, when computing the proposals for node μ, we group the proposals into three sets depending on the partition label s_μ of the proposal. If $s_\mu \in S_1$, then the proposal is kept. If $s_\mu \in S_2$ or $s_\mu \in S_3$, we keep the top 100 and 125 proposals respectively. (We experimented with changing these numbers – 100 and 125 – but noticed no significant difference in performance for small changes).

V. Learning the Model

Since HIM is a conditional model, in principle, estimation of its parameters can be achieved by any discriminative learning approach, such as maximum likelihood learning as used in Conditional Random Field (CRF) [8], max-margin learning [44], and structure-perceptron learning [11]. In this paper, we adopt the structure-perceptron learning which has been applied for learning the recursive deformable template (see paper [45]). Note that structure-perceptron learning is simple to implement and only needs to calculate the most probable configurations (parses) of the model. By contrast, maximum likelihood learning requires calculating the expectation of features which is difficult due to the large states of HIM. Therefore, structure-perceptron learning is more flexible and computationally simpler. Moreover, Collins [11] proved theoretical results for convergence properties, for both separable and non-separable cases, and for generalization.

The structure-perceptron learning will not compute the partition function $Z(\mathbf{I}; \alpha)$. Therefore we do not have a formal probabilistic interpretation. The goal of structure-perceptron learning is to learn a mapping from inputs to output structures. In our case, the inputs $\{\mathbf{I}^i\}$ are a set of images, and the outputs $\{W^i\}$ are a set of parse trees which specify the labels of image regions in a hierarchical form. We use a set of training examples $\{(\mathbf{I}^i, W^i) : i = 1...n\}$ and a set of functions ψ which map each $(\mathbf{I}, W))$ to a feature vector $\psi(\mathbf{I}, W) \in R^d$ (in practice, the training set only contains the labels of the pixels and we perform an approximation to estimate the full parse W^i for the training set – see 1) Implementation details in the Experimental Results section). The learning task is to estimate a parameter vector $\alpha \in R^d$ for the weights of the features. The feature vectors $\psi(\mathbf{I}, W)$ can include arbitrary features of parse trees, as we discussed in section (III-A). The loss function used in structure-perceptron learning is usually of form:

$$Loss(\alpha) = \alpha \cdot \psi(\mathbf{I}, W) - \max_{\overline{W}} \alpha \cdot \psi(\mathbf{I}, \overline{W}), \quad (15)$$

where W is the correct structure for input $\mathbf{I}$, and $\overline{W}$ is a dummy variable.

The basic structure-perceptron algorithm is designed to minimize the loss function. We adapt *"the averaged parameters"* version whose pseudo-code is given in figure (4). The algorithm proceeds in a simple way (similar to the perceptron algorithm for classification). The parameters are initialized to zero and the algorithm loops over the training examples. If the highest scoring parse tree for input $\mathbf{I}$ is not correct, then the parameters α are updated by an additive term. The most difficult step of the method is finding $W^* = \arg\max_W \alpha \cdot \psi(\mathbf{I}^i, W)$. This is precisely the parsing (inference) problem. Hence the practicality of structure-perceptron learning, and its computational efficiency, depends on the inference algorithm. As discussed earlier, see section (IV), the inference algorithm has polynomial computational complexity for an HIM which makes structure-perceptron learning practical for HIM. The averaged parameters are defined to be $\gamma = \sum_{t=1}^{T} \sum_{i=1}^{N} \alpha^{t,i}/NT$, where T is the number of epochs, NT is the total number of iterations. It is straightforward to store these averaged parameters and output them as the final estimates.

```
Input: A set of training images with ground truth $(\mathbf{I}^i, W^i)$ for $i = 1..N$. Initialize parameter vector $\alpha = 0$.
For $t = 1..T, i = 1..N$
    • find the best state of the model on the i'th training image with current parameter setting, i.e., $W^* = \arg\max_W \alpha \cdot \psi(\mathbf{I}^i, W)$
    • Update the parameters: $\alpha = \alpha + \psi(\mathbf{I}^i, W^i) - \psi(\mathbf{I}^i, W^*)$
    • Store: $\alpha^{t,i} = \alpha$
Output: Parameters $\gamma = \sum_{t,i} \alpha^{t,i}/NT$
```

Fig. 4. The structure-perceptron learning algorithm.

VI. Experimental Results

We evaluate the segmentation performance of the HIM on two public datasets, i.e. the MSRC 21-class image datset [10] and the PASCAL VOC 2007 [12].

A. Experiment I: MSRC

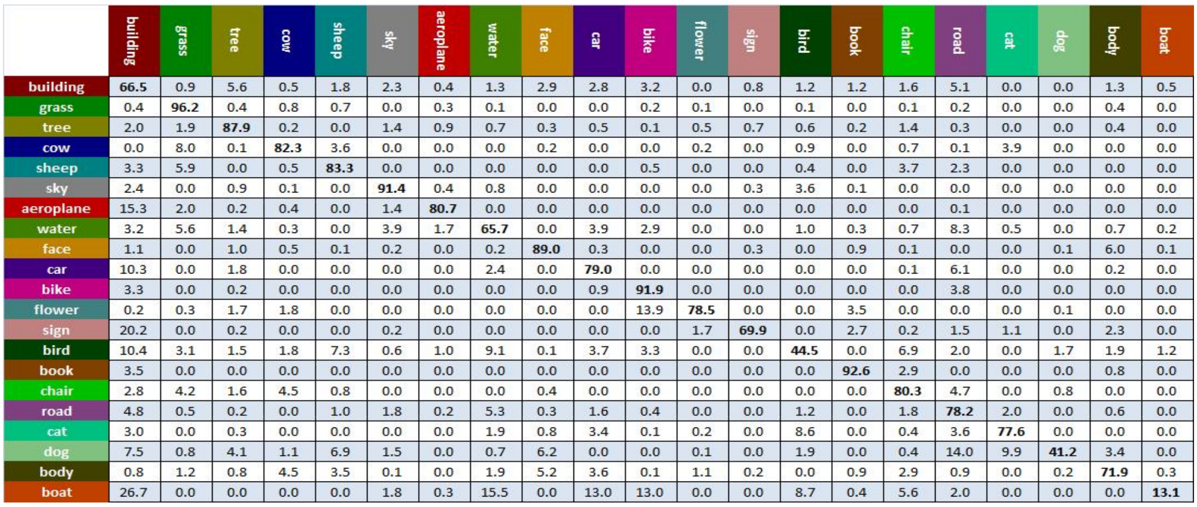
1) Implementation details: We use a standard public dataset, the MSRC 21-class Image Dataset [10], to perform experimental evaluations for the HIM. This dataset is designed to evaluate scene labeling including both image segmentation and multi-class object recognition. The ground truth only gives the labeling of the image pixels. To supplement this ground truth (to enable learning), we estimate the true labels (states of the S-R pair) of the nodes in the five-layer hierarchy of HIM by selecting the S-R pairs which have maximum overlap with the labels of the image pixels. This approximation only results in 2% error in labeling image pixels. There are a total of 591 images. We use the identical splitting as [10], i.e., 45% for training, 10% for validation, and 45% for testing. The parameters learnt from the training set, with the best performance on validation set, are selected.

For a given image $\mathbf{I}$, the parsing result is obtained by estimating the best configuration W^* of the HIM. To evaluate the performance of parsing we use the global accuracy measured in terms of all pixels and the average accuracy over the 21 object classes (global accuracy pays most attention to frequently occurring objects and penalizes infrequent objects). A computer with 8 GB memory and 2.4 GHz CPU was used for training and testing.

2) Image features and potential learning: The image features used by the classifier (47 in total) are the greyscale intensity, the color (R,G, B channels), the intensity gradient, the Canny edge, the response of DOG (difference of Gaussians) and DOOG (Difference of Offset Gaussian) filters at different scales (13*13 and 22*22) and orientations (0,30,60,...), and so on. We use 55 types of shape (spatial) filters (similar to [10]) to calculate the responses of 47 image features. There are $2585 = 47 * 55$ features in total. For each class, there are around $4,500$ weak classifiers selected by multi-class boosting. The boosting learning takes about 35 hours of which 27 hours are spent on I/O processing and 8 hours on computing.

3) Parsing results: The segmentation performance of the HIM on the MSRC dataset is shown in table (IV). The confusion matrix of 21 object classes is shown in table (5) where the diagonal is the classification accuracy of individual classes. Figure (6) (best viewed in color) shows several parsing results obtained by the HIM and by the classifier by itself (i.e. $p(o^r_\mu|\mathbf{I})$ learnt by boosting). The colors used in the segmentation correspond to the 21 object classes encoded in the confusion matrix shown in table (5). One can see that the HIM is able to roughly capture different shaped segmentation boundaries (see the legs of the cow and sheep in rows 1 and 3, and the boundary curve between sky and building in row 4). Table (IV) shows that HIM improves the results obtained by the classifier by 6.9% for average accuracy and 5.3% for global accuracy. In particular, in rows 6 and 7 in figure (6), one can observe that boosting gives many incorrect labels. It is impossible to correct such large mislabeled regions without the long-range interactions in the HIM, which improves the results by 20% and 32%. Figure 7 shows more segmentation results from different scenes.

4) Performance comparisons: In table (IV), we compare the performance of our approach with other



	building	grass	tree	cow	sheep	sky	aeroplane	water	face	car	bike	flower	sign	bird	book	chair	road	cat	dog	body	boat
building	**66.5**	0.9	5.6	0.5	1.8	2.3	0.4	1.3	2.9	2.8	3.2	0.0	0.8	1.2	1.2	1.6	5.1	0.0	0.0	1.3	0.5
grass	0.4	**96.2**	0.4	0.8	0.7	0.0	0.3	0.1	0.0	0.0	0.2	0.1	0.0	0.1	0.0	0.1	0.2	0.0	0.0	0.4	0.0
tree	2.0	1.9	**87.9**	0.2	0.0	1.4	0.9	0.7	0.3	0.5	0.1	0.5	0.7	0.6	0.2	1.4	0.3	0.0	0.0	0.4	0.0
cow	0.0	8.0	0.1	**82.3**	3.6	0.0	0.0	0.0	0.2	0.0	0.0	0.2	0.0	0.9	0.0	0.7	0.1	3.9	0.0	0.0	0.0
sheep	3.3	5.9	0.0	0.5	**83.3**	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.0	0.4	0.0	3.7	2.3	0.0	0.0	0.0	0.0
sky	2.4	0.0	0.9	0.1	0.0	**91.4**	0.4	0.8	0.0	0.0	0.0	0.0	0.3	3.6	0.1	0.0	0.0	0.0	0.0	0.0	0.0
aeroplane	15.3	2.0	0.2	0.4	0.0	1.4	**80.7**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0
water	3.2	5.6	1.4	0.3	0.0	3.9	1.7	**65.7**	0.0	3.9	2.9	0.0	0.0	1.0	0.3	0.7	8.3	0.5	0.0	0.7	0.2
face	1.1	0.0	1.0	0.5	0.1	0.2	0.0	0.2	**89.0**	0.3	0.0	0.0	0.3	0.0	0.9	0.1	0.0	0.0	0.1	6.0	0.1
car	10.3	0.0	1.8	0.0	0.0	0.0	0.0	2.4	0.0	**79.0**	0.0	0.0	0.0	0.0	0.0	0.1	6.1	0.0	0.0	0.2	0.0
bike	3.3	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.9	**91.9**	0.0	0.0	0.0	0.0	0.0	3.8	0.0	0.0	0.0	0.0
flower	0.2	0.3	1.7	1.8	0.0	0.0	0.0	0.0	0.0	0.0	13.9	**78.5**	0.0	0.0	3.5	0.0	0.0	0.0	0.1	0.0	0.0
sign	20.2	0.0	0.2	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	1.7	**69.9**	0.0	2.7	0.2	1.5	1.1	0.0	2.3	0.0
bird	10.4	3.1	1.5	1.8	7.3	0.6	1.0	9.1	0.1	3.7	3.3	0.0	0.0	**44.5**	0.0	6.9	2.0	0.0	1.7	1.9	1.2
book	3.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**92.6**	2.9	0.0	0.0	0.0	0.8	0.0
chair	2.8	4.2	1.6	4.5	0.8	0.0	0.0	0.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	**80.3**	4.7	0.0	0.8	0.0	0.0
road	4.8	0.5	0.2	0.0	1.0	1.8	0.2	5.3	0.3	1.6	0.4	0.0	0.0	1.2	0.0	1.8	**78.2**	2.0	0.0	0.6	0.0
cat	3.0	0.0	0.3	0.0	0.0	0.0	0.0	1.9	0.8	3.4	0.1	0.2	0.0	8.6	0.0	0.4	3.6	**77.6**	0.0	0.0	0.0
dog	7.5	0.8	4.1	1.1	6.9	1.5	0.0	0.7	6.2	0.0	0.0	0.1	0.0	1.9	0.0	0.4	14.0	9.9	**41.2**	3.4	0.0
body	0.8	1.2	0.8	4.5	3.5	0.1	0.0	1.9	5.2	3.6	0.1	1.1	0.2	0.0	0.9	2.9	0.9	0.0	0.2	**71.9**	0.3
boat	26.7	0.0	0.0	0.0	0.0	1.8	0.3	15.5	0.0	13.0	13.0	0.0	0.0	8.7	0.4	5.6	2.0	0.0	0.0	0.0	**13.1**

Fig. 5. Confusion Matrix for different object classes evaluated on the MSRC dataset [10].

successful methods [10], [46], [47]. Our approach outperforms those alternatives by 6% in average accuracy and 4% in global accuracy. Our boosting results are better than Textonboost [10] because of image features. Would we get better results if we use a flat CRF with our boosting instead of a hierarchy? We argue that we would not because the CRF only improves TextonBoost's performance by 3 percent [10], while we gain 5 percent by using the hierarchy (and we start with a higher baseline). Some other methods [48], [37], [9], which are worse than [46], [47] and evaluated on simpler datasets [9], [37] (less than 10 classes), are not listed here due to lack of space. We also report recent progress ([49], [50]) on this dataset. Ladicky et al. [50] achieves better performance, but they use better classifiers (more powerful unary potentials, see table IV) .

5) Empirical convergence analysis of perceptron learning.: The structure-perceptron learning takes about 20 hours to converge in $5520(T = 20, N = 276)$ iterations. In the testing stage, it takes 30 seconds to parse an image with size of 320×200 (6s for extracting image features, 9s for computing the strong classifier of boosting and 15s for parsing the HIM). Figure 8 plots the convergence curves evaluated by average accuracy and global accuracy on the test set. It shows that the structure-perceptron learning converges in T=20 epochs.

6) Diagnosis on the function of S-R pair.: Figure (9) shows how the S-R pairs (including the segmentation templates) can be used to (partially) parse an object into its constituent parts, by the correspondence between S-R pairs and specific parts of objects. We plot the states of a subset of S-R pairs for some images. For example, the S-R pair consisting of two horizontal bars labeled "cow" and "grass" respectively indicates the cow's stomach consistently across different images. Similarly, the cow's tail can be located according to the configuration of another S-R pair with vertical bars. In principle, the whole object can be parsed into its constituent parts which are aligned consistently. Developing this idea further is an exciting aspect of our current research.

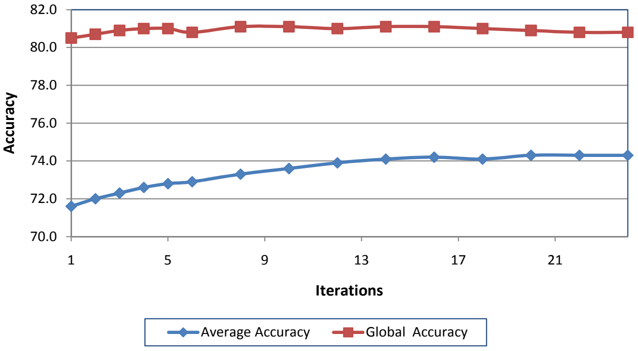


Fig. 8. Empirical Convergence Analysis on the MSRC dataset. The curves plot the average and global accuracy as a function of the number of iterations (of parameter estimation). The accuracy is evaluated on the test dataset

B. Experiment II: PASCAL VOC 2007

The PASCAL VOC 2007 dataset [12] was used for the PASCAL Visual Object Category segmentation contest 2007. It contains 209 training, 213 validation and 210 segmented test images of 20 foreground (object) and 1 background classes. It

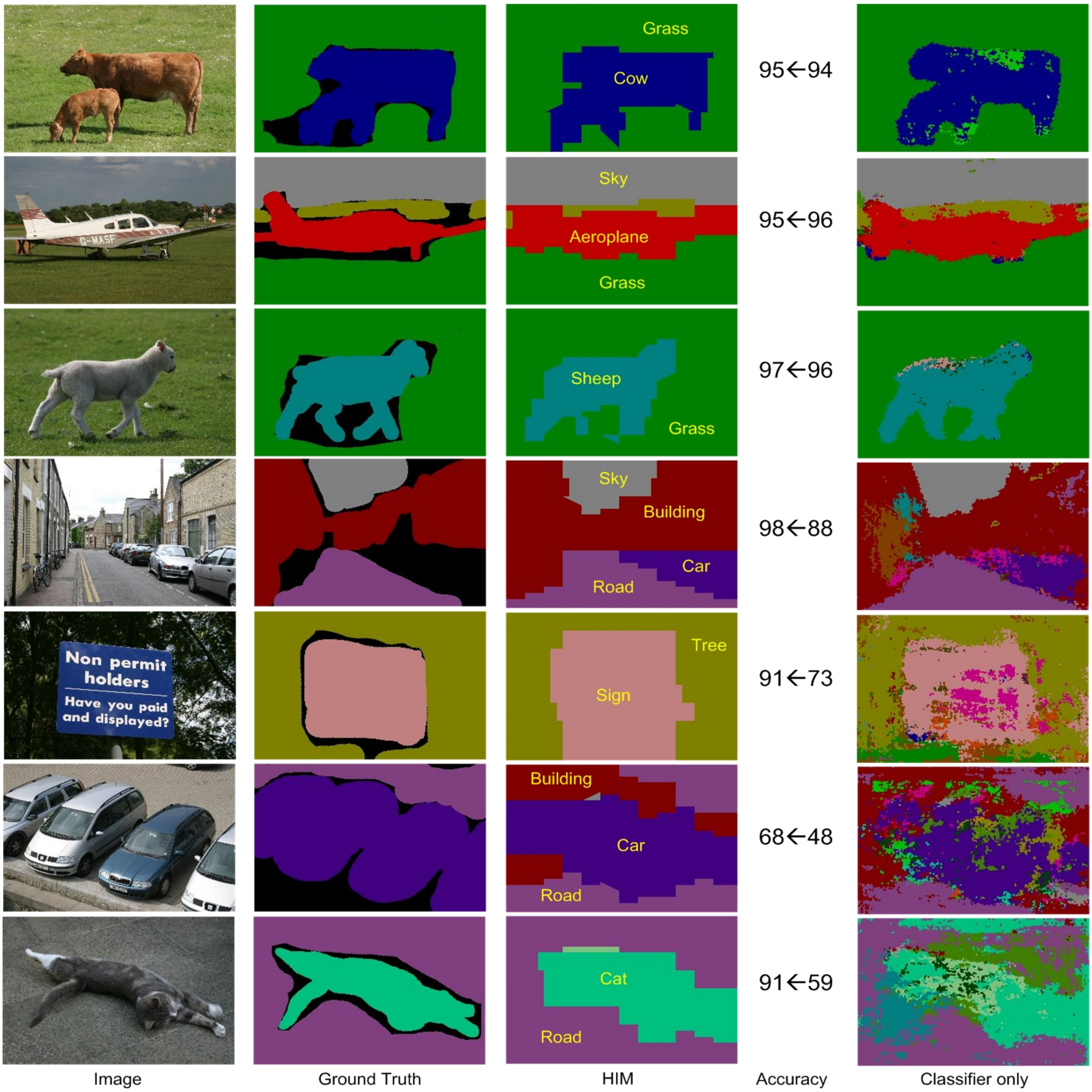


Fig. 6. This figure is best viewed in color. The colors indicate the labels of 21 object classes as in the MSRC dataset [10]. The columns (except the fourth "accuracy" column) show the input images, ground truth, the labels obtained by HIM and the boosting classifier respectively. The "accuracy" column shows the global accuracy obtained by HIM (left) and the boosting classifier (right). In these 7 examples, HIM improves boosting by 1%, -1% (an outlier!), 1%, 10%, 18%, 20% and 32% in terms of global accuracy.

	Textonboost[10]	PLSA-MRF [46]	Auto-context [47]	Region Ancestry[49]	HCRF[50]	Classifier only	HIM
Average	57.7	64.0	68	67	75 (72)	67.2	74.1
Global	72.2	73.5	77.7	–	86 (81)	75.9	81.2

TABLE IV

Performance Comparisons for average accuracy and global accuracy on the MSRC dataset. "Classifier only" are the results where the pixel labels are predicted by the classifier obtained by boosting only. The numbers in the brackets are the results obtained by the classifier (unary potential) used in HCRF [50].

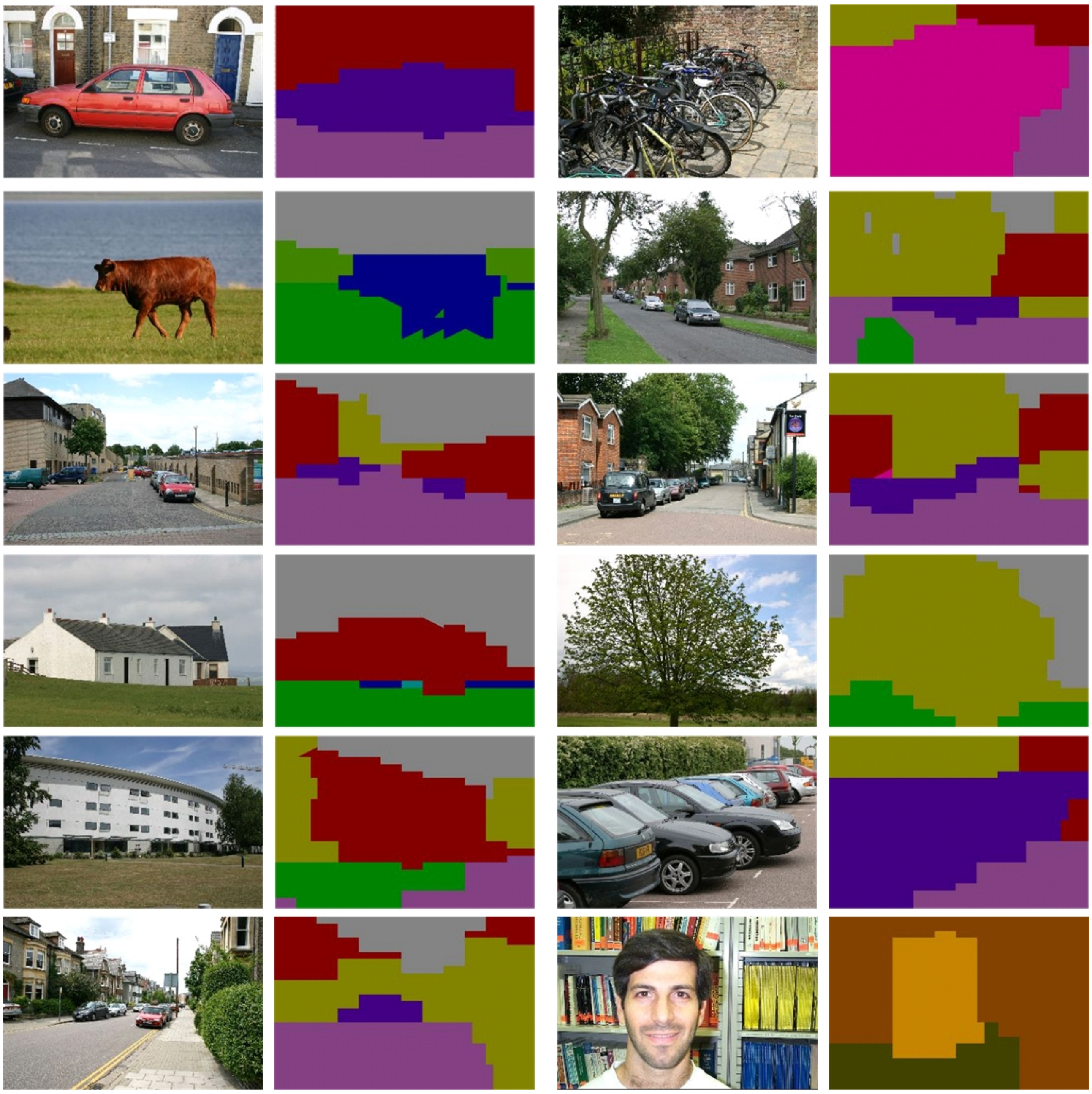


Fig. 7. More parse results on the MSRC dataset. The correspondence between the color and the object class is defined in figure 5.

is more challenging than the MSRC-21 dataset due to more significant background clutter, illumination effects and occlusions. We trained the HIM using the same parameter settings and features as in the experiment on the MSRC-21 dataset. The parse results are shown in figure 10. The segmented results look visually worse than those on the MSRC dataset because in the PASCAL dataset, a single “background” class covers several object classes, such as sky, grass, etc. while more accurate labeling is imposed in the MSRC dataset. We compared our approach with other representative methods reported in the PASCAL VOC segmentation contest 2007 [12]. The comparisons in table V show that the HIM outperforms most methods and is comparable with TKK.

VII. Conclusion

This paper describes a novel hierarchical image model (HIM) for 2D image parsing. The hierarchical nature of the model, and the use of recursive segmentation and recognition templates, enables the

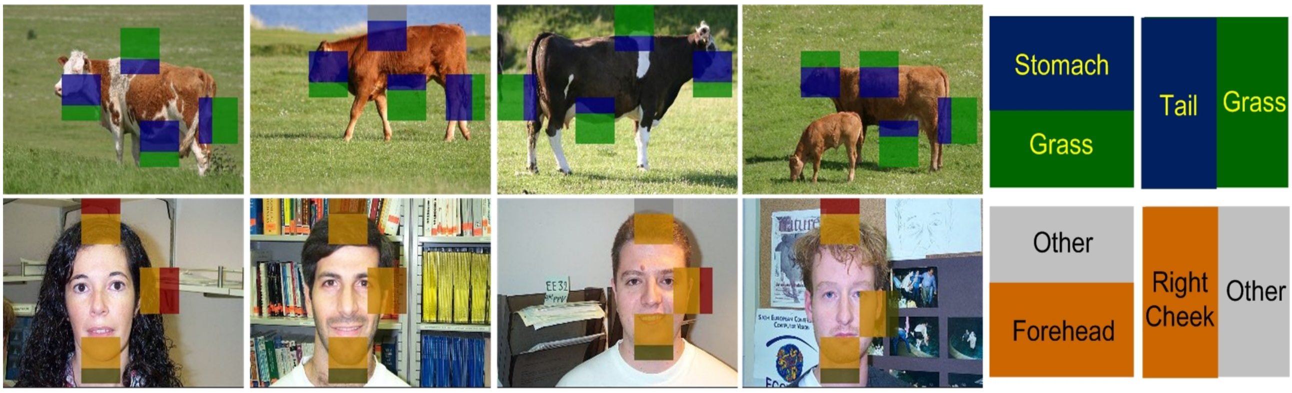


Fig. 9. The S-R pairs can be used to parse the object into parts. The colors indicate the identities of objects. The shapes (spacial layout) of the segmentation templates distinguish the constituent parts of the object. Observe that the same S-R pairs (e.g. stomach above grass, and tail to the left of grass) correspond to the same object part in different images.

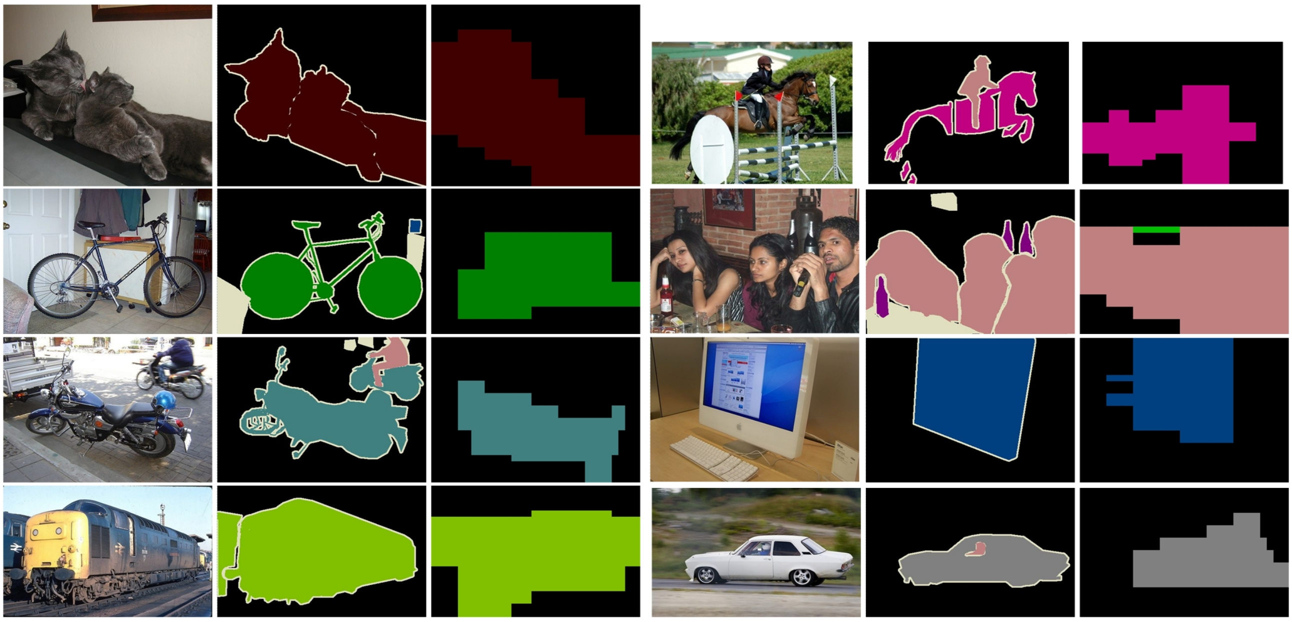


Fig. 10. Parse results on the PASCAL VOC 2007 dataset [12]. The first three columns show the input images, the groundtruth and the parse results of HIM, respectively. The next three columns show the other four examples.

	Brookes	TKK	UoCTTI	HIM
Average	8.5	30.4	21.2	26.5
Global	58.4	24.4	–	67.2

TABLE V
PERFORMANCE COMPARISONS ON THE PASCAL VOC 2007 DATASET. THREE METHODS REPORTED IN THE VOC SEGMENTATION CONTEST 2007 [12] ARE COMPARED.

HIM to represent complex image structures in a coarse-to-fine manner. We can perform inference (parsing) rapidly in polynomial time by exploiting the hierarchical structure. Moreover, we can learn the HIM probability distribution from labeled training data by adapting the structure-perceptron algorithm. We demonstrated the effectiveness of HIM's by applying them to the challenging task of segmentation and labeling of the public MSRC and PASCAL VOC 2007 image databases. Our results show that we perform competitively with state-of-the-art approaches.

The design of the HIM was motivated by drawing parallels between language and vision processing. We have attempted to capture the underlying spirit of the successful language processing approaches – the hierarchical representations based on the recursive composition of constituents and efficient inference and learning algorithms. Our current work attempts to extend the HIM's to improve their representational power while maintaining computational efficiency.

Acknowledgments

This research was supported by the W.M. Keck foundation, NSF grants 0413214 and 613563, and by the Air Force FA9550-08-1-0489.

References

[1] F. Jelinek and J. D. Lafferty, "Computation of the probability of initial substring generation by stochastic context-free grammars," *Computational Linguistics*, vol. 17, no. 3, pp. 315–323, 1991.

[2] M. Collins, "Head-driven statistical models for natural language parsing," *Ph.D. Thesis, University of Pennsylvania*, 1999.

[3] K. Lari and S. J. Young, "The estimation of stochastic context-free grammars using the inside-outside algorithm," in *Computer Speech and Languag*, 1990.

[4] M. Shilman, P. Liang, and P. A. Viola, "Learning non-generative grammatical models for document analysis," in *Proceedings of IEEE International Conference on Computer Vision*, 2005, pp. 962–969.

[5] Z. Tu and S. C. Zhu, "Image segmentation by data-driven markov chain monte carlo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 657–673, 2002.

[6] Z. Tu, X. Chen, A. L. Yuille, and S. C. Zhu, "Image parsing: Unifying segmentation, detection, and recognition," in *Proceedings of IEEE International Conference on Computer Vision*, 2003, pp. 18–25.

[7] S. Zhu and D. Mumford, "A stochastic grammar of images," *Foundations and Trends in Computer Graphics and Vision*, vol. 2, no. 4, pp. 259–362, 2006.

[8] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of International Conference on Machine Learning*, 2001, pp. 282–289.

[9] X. He, R. S. Zemel, and M. Á. Carreira-Perpiñán, "Multiscale conditional random fields for image labeling," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004, pp. 695–702.

[10] J. Shotton, J. M. Winn, C. Rother, and A. Criminisi, "TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation," in *Proceedings of European Conference on Computer Vision*, 2006, pp. 1–15.

[11] M. Collins, "Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms," in *Proceedings of Annual Meeting on Association for Computational Linguistics conference on Empirical methods in natural language processing*, 2002, pp. 1–8.

[12] E. M., L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results," http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[13] L. Zhu, Y. Chen, Y. Lin, C. Lin, and A. Yuillek, "Recursive segmentation and recognition templates for 2d parsing," in *Advances in Neural Information Processing Systems*, 2008.

[14] D. Marr, *Vision*, 1982.

[15] H. G. Barrow and J. M. Tenenbaum, "Recovering intrinsic scene characteristics from images," *Computer Vision Systems*, 1978.

[16] M. Nitzberg and D. Mumford, "The 2.1d sketch," in *In Proc. Int. Conf. on Computer Vision*, 1990.

[17] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1984.

[18] C. Guo, S. Zhu, and Y. Wu, "Primal sketch: Integrating texture and structure," *Computer Vision and Image Understanding*, 2007.

[19] S. Zhu and A. Yuille, "Region competition: Unifying snake/balloon, region growing and bayes/mdl/energy for multi-band image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1996.

[20] A. Blake and A. Zisserman. MIT Press, 1987.

[21] D. Geiger and F. Girosi, "Parallel and deterministic algorithms from mrfs: Surface reconstruction," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1991.

[22] D. Geiger and A. Yuille, "A common framework for image segmentation," *International Journal of Computer Vision*, 1991.

[23] C. Koch, J. Marroquin, and A. Yuille, "Analog neuronal networks in early vision," in *Proceedings of the National Academy of Science*, 1986.

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

[25] T. M. Cover and J. A. Thomas. New York: Wiley, 1991.

[26] S. D. Pietra, V. J. D. Pietra, and J. D. Lafferty, "Inducing features of random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 380–393, 1997.

[27] S. Roth and M. Black, "Fields of experts: A framework for learning image priors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[28] S. Konishi, A. L. Yuille, J. M. Coughlan, and S. C. Zhu, "Statistical edge detection: Learning and evaluating edge cues," *IEEE Trans.on Pattern Analysis and Machine Intelligence*, 2003.

[29] A. Rosenfeld, R. A. Hummel, and S. W. Zucker, "Scene labeling by relaxation operations," *IEEE Trans. Syst., Man, Cybern*, 1976.

[30] S. Kumar and M. Hebert, "Discriminative random fields: A discriminative framework for contextual interaction in classification," in *International Conference on Computer Vision*, 2003.

[31] B. Olshausen and D. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?" *Vision Research*, 1997.

[32] Y. Wu, S. Zhu, and X. Liu, "Equivalence of julesz ensembles and frame models," *International Journal of Computer Vision*, vol. 38, pp. 247–265, 2000.

[33] S. Mallat and Z. Zhang, "Matching pursuit in a time-frequency dictionary," in *IEEE Signal Processing*, 1993.

[34] D. Mumford and J. Shah, "Optimal approximations of piecewise smooth functions and associated variational problems," in *Comm. in Pure and Appl. Math.*, 1989.

[35] H. Chen, Z. Xu, Z. Liu, and S. C. Zhu, "Composite templates for cloth modeling and sketching," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, pp. 943–950.

[36] C. Manning and H. Schuetze, *Foundations of statistical natural language processing*. Cambridge, Mass, USA: MIT Press, 1999.

[37] S. Kumar and M. Hebert, "A hierarchical field framework for unified context-based classification," in *Proceedings of IEEE International Conference on Computer Vision*, 2005, pp. 1284–1291.

[38] E. L. Allwein, R. E. Schapire, and Y. Singer, "Reducing multiclass to binary: A unifying approach for margin classifiers,"

Journal of Machine Learning Research, vol. 1, pp. 113–141, 2000.

[39] Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images," in *Proceedings of IEEE International Conference on Computer Vision*, 2001, pp. 105–112.

[40] T. Lee and D. Mumford, "Hierarchical bayesian inference in the visual cortex," *Journal of the Optical Society of America*, 2003.

[41] A. Oliva and A. Torralba, "Building the gist of a scene: the role of global image features in recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 155, pp. 23–36, 2006.

[42] A. Levin and Y. Weiss, "Learning to combine bottom-up and top-down segmentation," in *Proceedings of European Conference on Computer Vision*, 2006, pp. 581–594.

[43] E. B. Sudderth, A. B. Torralba, W. T. Freeman, and A. S. Willsky, "Learning hierarchical models of scenes, objects, and parts," in *Proceedings of IEEE International Conference on Computer Vision*, 2005, pp. 1331–1338.

[44] B. Taskar, D. Klein, M. Collins, D. Koller, and C. Manning, "Max-margin parsing," in *Proceedings of Annual Meeting on Association for Computational Linguistics conference on Empirical methods in natural language processing*, 2004.

[45] L. Zhu, Y. Chen, X. Ye, and A. L. Yuille, "Structure-perceptron learning of a hierarchical log-linear model," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

[46] J. Verbeek and B. Triggs, "Region classification with markov field aspect models," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.

[47] Z. Tu, "Auto-context and its application to high-level vision tasks," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

[48] J. Verbeek and B. Triggs, "Scene segmentation with crfs learned from partially labeled images," in *Advances in Neural Information Processing Systems*, vol. 20, 2008.

[49] J. J. Lim, P. Arbelaez, C. Gu, and J. Malik, "Context by region ancestry," in *IEEE International Conference on Computer Vision*, 2009.

[50] L. Ladicky, C. Russell, P. Kohli, and P. Torr, "Associative hierarchical crfs for object class image segmentation," in *IEEE International Conference on Computer Vision*, 2009.