# Recursive Segmentation and Recognition Templates for Image Parsing

Long (Leo) Zhu[1], Yuanhao Chen[2], Yuan Lin[3], Chenxi Lin[4], Alan Yuille[2,5]

[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
[5] Department of Brain and Cognitive Engineering, Korea University

***Abstract*—In this paper, we propose a Hierarchical Image Model (HIM) which parses images to perform segmentation and object recognition. The HIM represents the image recursively by segmentation and recognition templates at multiple levels of the hierarchy. This 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 (similar to how natural language models represent sentence structure in terms of hierarchical representations such as verb and noun phrases). Secondly, the structure of the HIM allows us to design a rapid inference algorithm, based on dynamic programming, which yields the first polynomial time algorithm for image labeling. Thirdly, we learn the HIM efficiently using machine learning methods from a labeled dataset. We demonstrate that the HIM is comparable with the state-of-the-art methods by evaluation on the challenging public MSRC and PASCAL VOC 2007 image datasets.**

## 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, language contains elementary constituents, such as nouns and verbs, that can be recursively composed into a hierarchy (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 because there are no direct analogies to the syntactic categories and phrase structures that occur in language. Secondly, the inference problem is formidable due to the well-known complexity and ambiguity of segmentation and recognition. In most languages the boundaries between different words are well-defined (Chinese is an exception) but, by contrast, the segmentation boundaries between different image regions are usually highly unclear. Exploring all the different image partitions risks a combinatorial explosions because of the two-dimensional nature of images (dynamic programming methods can be used for one-dimensional data such as language). 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 and highly structured problems such as text [4]).

We argue that to make progress at image parsing requires making trade-offs between the complexity of the representation and the complexity of the computation (for inference and learning). Our work builds on three themes in the recent literature. Firstly, the use of stochastic grammars to represent images and objects [5], [6], [7], [8]. This style of research uses generative models for images and pays less attention to the complexity of compu-

tation. Inference is usually performed by MCMC sampling which is only efficient provided effective proposal probabilities can be designed [5][6]. Secondly, the related work on hierarchical image models [9],[10],[11],[12]. This work has paid greater attention to computational complexity, but been largely focussed on image processing applications [13]. Thirdly, the use of discriminative models, such as conditional random fields (CRF's) [14], which have obtained promising results on image labeling [15], [16]. These models use simpler representations than the stochastic grammars but instead use non-local features trained using discriminative training methods (e.g. AdaBoost, MLE) and efficient algorithms (e.g. belief propagation and graph-cuts). But current CRF models have limited ability to represent long-range relationships and contextual knowledge.

In this paper, we introduce Hierarchical Image Models (HIM)'s for image parsing. Our strategy is to combine the representational advantages of stochastic grammars with the effectiveness of discriminative learning techniques. We define a hierarchical model of hidden states where the state variables are *segmentation and recognition templates* which represent complex image knowledge and are similar to the noun and verb phrases used in natural language. Extending this analogy, we can represent image structure coarsely at high levels of the hierarchy and give more detailed descriptions at lower levels. More precisely, each node of the hierarchy corresponds to an image region (whose size depends on the level in the hierarchy). The state of the 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 summary, 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 [17]).

To illustrate the HIM we implement it for parsing images and we evaluate it on the public MSRC image dataset [16] and the PASCAL VOC 2007 dataset [18]. 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 [19].

## II. Background: Image Models, Representations and Computation

We review the background material using the formulation of probabilities defined over graphs. This will give us a set of considerations which gives a classification of probabilistic image models.

### A. Probabilities and Graphs

**Probabilistic models on structured representations** are defined by specifying a probability distribution over a graph structure. The graph structure is represented by its nodes $\mathcal{V}$ and edges $\mathcal{E}$. The edges $\mathcal{E}$ specify the dependency structure of the state variables $w_\mu$ defined at the nodes (e.g., the Markov property). A clique $Cl$ is defined to be a subset of nodes $\mu \in \mathcal{V}$ such that every pair of nodes in the subset is connected by an edge (i.e. $\forall \mu, \nu \in Cl$ we require $(\mu, \nu) \in \mathcal{E}$).

In this paper we will be concerned with hierarchical graph structures where the graph nodes can be organized into levels: $\mathcal{V} = \bigcup_l \mathcal{V}^l$, where $\mathcal{V}^l$ is the set of nodes at level $l$. A node $\mu \in \mathcal{V}^l$ at level $l$ has *vertical edges* connecting it to nodes at other levels – e.g., *child nodes* at level $l-1$ and *parent nodes* at level $l+1$ – or *horizontal edges* connecting it to *sibling nodes* at the same level $l$ (note that the HIM will use vertical edges only).

We let $\mathbf{W} = \{w_\mu : \mu \in \mathcal{V}\}$ denote the states of all the graph variables. We denote the states of all nodes in clique $Cl$ by $\mathbf{w}_{Cl}$. Factor functions of form $\psi_{Cl}(\mathbf{w}_{Cl})$ are defined over the variables in the cliques. There are also factor functions $\psi_\mu(w_\mu, \mathbf{I})$ relating the state variables to the input image $\mathbf{I}$. These factor functions will be weighted by parameters $\alpha_\mu, \alpha_{Cl}$ to yield potentials $\alpha_\mu \cdot \psi_\mu(w_\mu, \mathbf{I})$ and $\alpha_{Cl} \cdot \psi_{Cl}(\mathbf{w}_{Cl})$.

We define probability distributions over the state variables by Gibbs distributions. This can be done in two ways. For *discriminative* models we combine

all the potential terms together to form an energy:

$$E(\mathbf{W};\mathbf{I}) = -\sum_{\mu\in\mathcal{V}} \alpha_\mu \cdot \psi(w_\mu, \mathbf{I}) - \sum_{Cl} \alpha_{Cl} \cdot \psi(\mathbf{w}_{Cl}), \tag{1}$$

and define a conditional distribution:

$$P(\mathbf{W}|\mathbf{I}) = \frac{1}{Z[\alpha, \mathbf{I}]} \times \exp\{-E(\mathbf{W};\mathbf{I})\}. \tag{2}$$

For *generative* models, we define a prior $P(\mathbf{W})$ and likelihood function $P(\mathbf{I}|\mathbf{W})$ as Gibbs distributions using energies $-\sum_{Cl} \alpha_{Cl} \cdot \psi_{Cl}(\mathbf{w}_{Cl})$ and $-\sum_{\mu\in\mathcal{V}} \alpha_\mu \cdot \psi(w_\mu, \mathbf{I})$ respectively.

Using this framework, we can categorize probability models by the following considerations:

1) Hierarchical or Flat: Is the graph structure hierarchical with three or more levels? Are the majority of the edges vertical between levels, or horizontal within levels?
2) Fixed or Variable Topology: Is the graph structure fixed or variable? Can the number of nodes, or the edges, vary between images.
3) State Variables: What do the state variables represent? Do they represent the same quantity at all levels of the hierarchy?
4) Discriminative or Generative: is the model generative or discriminative? – i.e. is the model formulated in terms of a prior $P(\mathbf{W})$ and a likelihood $P(\mathbf{I}|\mathbf{W})$ (generative) or only by a conditional, or posterior, distribution $P(\mathbf{W}|\mathbf{I})$?
5) Inference Algorithm: which algorithms are used to compute the most probable state $\mathbf{W}^* = \arg\max P(\mathbf{W}|\mathbf{I})$? Or to compute other estimates for $\mathbf{W}$? The choice of algorithm is influenced by the nature of the graph structure (e.g., loopy or non-loopy) and the form of the state variables.
6) Learning Algorithm: Are the parameters $\alpha$ of the model set by hand, or are they learnt from the training data? If learnt, what algorithm is used? Is the graph structure also learnt?

These considerations are related. In particular, if the graph structure has no closed loops then efficient algorithms (e.g., dynamic programming) can be used for inference. By contrast, if the graph contains multiple closed loops then inference is usually intractable except in special cases (e.g., if the energy function is submodular which enables max-flow algorithms).

We now briefly review the literature of image segmentation using these criteria as a guide.

### B. Weak Membrane Models

These are a historically influential class of models for image segmentation which assumed that images were piecewise smooth (weak membranes) [20],[21],[22]. These models were *flat* with *fixed* topology. The graph nodes $\{\mu \in \mathcal{V}\}$ corresponded to the pixels of the image (i.e. each $\mu$ corresponded to a lattice site $(i, j)$ on a two-dimensional grid). The graph edges $\mathcal{E}$ link neighboring pixels sites (i.e. lattice site $(i, j)$ to $(i \pm 1, j)$ and $(i, j \pm 1)$).

The *state variables* $\mathbf{W} = \{w_\mu : \mu \in \mathcal{V}\}$ represent smoothed versions of the image intensity $\mathbf{I} = \{I_\mu : \mu \in \mathcal{V}\}$ (which could be augmented by another layer of binary-valued line process variables indicating the presence of edges).

The models were *generative* with prior $P(\mathbf{W})$ and likelihood $P(\mathbf{I}|\mathbf{W})$ of form:

$$P(\mathbf{I}|\mathbf{W}) = \prod_{\mu\in\mathcal{V}} P(I_\mu|w_\mu),$$

$$P(\mathbf{W}) = \frac{1}{Z(\alpha)} \exp\{-\sum_{(\mu,\nu)\in\mathcal{E}} \alpha_{\mu,\nu} \cdot \psi(w_\mu, w_\nu)\}, \tag{3}$$

where the prior $P(\mathbf{W})$ only enforces highly local constraints on the $\mathbf{W}$ because the graph edges $\mathcal{E}$ only link neighboring pixel sites.

A variety of inference algorithms were applied to these models. Geman and Geman [20] used simulated annealing – Markov Chain Monte Carlo (MCMC) while lowering a temperature parameter. Blake and Zisserman [21] developed graduate non-convexity. Geiger and his collaborators described a range of inference algorithms – including mean field theory (an early variational method), and the expectation maximization algorithm [23], [24]. Koch *et al.* [25] conjectured that these algorithms were biologically plausible and might be implemented in cortical area V1.

The weak membrane models were not learnt from data. But subsequent work [26],[27] showed that prior distributions $P(\mathbf{W})$ learnt from natural images were very similar to those assumed by weak membrane models.

The weak membrane models were historically influential but they are only effective on a restricted class of images due to their simplistic assumptions. Studies of natural images show that piecewise

smoothness is only, at best, a first order approximation and fails to capture the texture and appearance qualities required to parse an image and label its components.

### C. Discriminative Models

*Discriminative* models learn distributions $P(\mathbf{W}|\mathbf{I})$ directly. They can be applied to image labeling and hence can parse images. These models tend to be flat with fixed topology and with state variables $\mathbf{W} = \{w_\mu : \mu \in \mathcal{V}\}$ representing the labels at each pixel – e.g., the labels can be "sky", "vegetation", "edge", "road", and "other". The simplest models of this type are factorizable [28]:

$$P(\mathbf{W}|\mathbf{I}) = \prod_{\mu \in \mathcal{V}} P(w_\mu|\mathbf{I}), \tag{4}$$

where $P(w_\mu|\mathbf{I}) \propto \exp\{-\alpha_\mu \cdot \psi(w_\mu, \mathbf{I})\}$ For these factorized models, learning of $P(w_\mu|\mathbf{I})$ can be performed by standard statistical methods (e.g. regression or AdaBoost). Inference is also straightforward since it can be performed independently at each node – e.g. $\mathbf{W}^* = \arg\max_{\mathbf{W}} P(\mathbf{W}|\mathbf{I})$ can be found by computing $w^*_\mu = \arg\max_{w_\mu} P(w_\mu|\mathbf{I})$ for all $\mu \in \mathcal{V}$. Factorizable models are surprisingly successful for labeling certain types of image regions (e.g., sky, vegetation, and road) because these regions tend to have homogeneous intensity properties which can be captured by non-local feature functions [28]. But these factorizable models ignore context and are less successful for non-homogeneous regions (e.g., buildings).

More advanced models use additional information by taking into account the properties of neighboring labels – as described in the classic work on relaxation labeling [29] (which was not formulated probabilistically and did not use learning). This can be done using conditional random fields [14]. It gives models of form:

$$P(\mathbf{W}|\mathbf{I}) = \frac{1}{Z[\alpha, \mathbf{I}]}$$
$$\exp\{-\sum_{\mu \in \mathcal{V}} \alpha_\mu \cdot \psi(w_\mu, \mathbf{I}) - \sum_{Cl} \alpha_{Cl} \cdot \psi(\mathbf{w}_{Cl})\}. \tag{5}$$

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 algorithms like max-flow and belief propagation have been applied for *inference* and maximum likelihood for *learning*. Other work of this type includes [15],[31],[32],[16] which has shown success on image labeling tasks. Also related is work on segmenting and labeling hand-drawn ink figures with a tree-like structure which enabled efficient inference and learning [33]. In summary, discriminative models are effective for image labeling, because they use non-local image features in their data terms $\psi(w_\mu, \mathbf{I})$, but usually have simple local 'prior' terms $\psi(\mathbf{W})$.

### D. Regional Models

Another class of models assume that the image consists of a variable number of regions and the intensity properties within each region can be specified by a parameterized probability model. Examples include [34], [5], [6]. This is a flat model with variable topology – the number of graph nodes corresponds to the number of image regions and the state variables represent the properties of the region (i.e. the shape of the region and the parameters of the model that generates the intensity properties of the region). The graph edges relate the states of neighboring regions (e.g., ensuring that the region boundaries do not overlap). These models are generative with a distribution $P(\mathbf{I}|\mathbf{W})$ and a prior $P(\mathbf{W})$.

More formally, regional models seek to 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. The intensity $\mathbf{I}_a$ within each region $\mathcal{D}_a$ is generated by a distribution $P(\mathbf{I}_{\mathcal{D}_a}|\tau_a, \gamma_a)$, where $\tau_a$ labels the model type and $\gamma_a$ labels its parameters – e.g., $\tau$ could label the region as 'texture' and $\gamma$ would specify the parameters of the texture model. In graphical terms, there are $M$ graph nodes whose state variables $w_a = (\mathcal{D}_a, \tau_a, \gamma_a)$ represent the region, its model type, and the model parameters (with the constraint that $\bigcup_a \mathcal{D}_a = \mathcal{D}$). There is a prior probability on $\mathbf{W}$ specified by a distribution $P(M)$ on the number of regions/nodes, and on the regional properties $P(\{\mathcal{D}_a\}|M) \prod_a P(\tau_a)P(\gamma_a)$. *Inference* is very challenging for these models because of the changes in topology and the high-dimension space of the variables – e.g., there is an exponentially large number of possible boundaries $\partial\mathcal{D}_a$ for the regions. Inference was done using a stochastic algorithm

– data driven Markov Chain Monte Carlo (DDMCMC) [5], [6] – which is guaranteed to converge to the optimal solution. But the convergence rate of this algorithm is unknown. Region competition [34] is an alternative greedy algorithm which is successful for limited classes of problems.

This approach uses *learning* for components of the model, but these are treated independently. For example, Tu *et al.* learnt both generative and discriminative models of faces and text [6] (the discriminative models were used to create proposal probabilities for DDMCMC). But there is no global criterion which is optimized during learning.

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.

### E. Stochastic Grammars and Hierarchical Models

The models we have described so far are limited in their ability to represent images and, in particular, to enforce long-range interactions and structures at multi-scales. We now present two approaches which address these issues.

One approach is stochastic grammars [35], [8] have been proposed to model images and objects at different levels. This is an attractive research program which is described in more detail by Zhu and Mumford [8]. The work by Geman and his collaborators are also related to this framework [7]. Stochastic grammars are a very promising approach since they have great representational power and can model complex knowledge. 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.

There is also related literature on multiscale (i.e. hierarchical) Markov tree models, particularly in the image analysis community, which is reviewed in [13]. These models can capture long-range interactions but do not have as sophisticated representations as the stochastic grammars. These hierarchical models are often designed using quadtree structures [13] (which we will use for HIMs). Other examples includes multiscale random field models [9] and hidden markov models using wavelets [10]. More recent examples of this approach includes [36],[11] and [12]. But these models have simple state variables (not as complex as those used by HIMs) and do not use discriminative learning. Other hierarchical approaches to image segmentation include Sharon *et al.*'s segmentation by weighted aggregation [37] and multiscale spectral segmentation [38], but these are not expressed within a probabilistic framework and have not been applied to image labeling tasks.

## III. The Hierarchical Image Model (HIM)

The Hierarchical Image Model (HIM) combines properties of the models described above. It has a hierarchical graph structure with fixed topology based on the quadtree models for image processing [13]. There are only vertical edges which means that the graph has no closed loops. The state variables are segmentation recognition templates, which represent the segmentation and labeling of image regions, and relate to stochastic grammar models. The inference algorithm is dynamic programming (exploiting the lack of closed loops in the graph structure). The model is discriminative with a set of pre-specified factor functions whose parameters are learnt by the structure perceptron algorithm [17]. The restrictions of using fixed topology are compensated by the greater representation of the state variables).

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

TABLE I
THE TERMINOLOGY USED IN THE HIM 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 levels in this paper) where the top node of the hierarchy represents the whole image. The intermediate nodes represent different subregions of the image and the leaf nodes represent local image patches ($27 \times 27$ in this paper). This is similar to the quadtree representation used in image processing [13] (but the state variables, the learning algorithm, and the application are very different). We note that quadtree representations are known to have boundary artifacts because pixels nearby in the image may be assigned to different branches of the tree. But this does not cause significant problems for HIMs, as we quantify in section (VI), because we use nonlocal feature functions.

The notation for the model is summarized in table I. 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 $\mu$ 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 $\mu$ is fixed and denoted by $R(\mu)$, while pixels within $R(\mu)$ are labeled by $r$.

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* template, 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 $K \leq 3$ non-overlapping sub-regions $R(\mu) = \bigcup_{i=1}^{K} R_i(\mu)$, with $R_i(\mu) \bigcap R_j(\mu) = \emptyset$ $(i \neq j)$, and is selected from a dictionary $D_s$, where $|D_s| = 30$ in this paper. This dictionary of segmentation templates is shown in figure (1) and 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"). We allow neighboring subregions to have the same label. 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 $s_\mu, \vec{c}_\mu$ – $o_\mu^r = c_i(\mu)$, provided $r \in R_i(\mu)$. Note that any two pixels within the same subregion must have the same label. Observe also that each image pixel will have labels $o_\mu^r$ defined at all levels of the hierarchy, which will be encouraged (probabilistically) to be consistent. We do not want to impose complete consistency, because as described below, we want the higher levels of the HIM to represent only the coarse levels structure of the image enabling the lower levels to represent the more finer-scale structure.

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 [16], [15], [31] 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). Note that this means that there are fewer variables used to represent the image at higher levels and so we cannot enforce complete consistency between the representations at different levels, nor do we want to.

### B. The distribution

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

$$p(W|\mathbf{I}; \alpha) = \frac{1}{Z(\mathbf{I}; \alpha)} \exp\{-E(\mathbf{I}, \mathbf{W}, \alpha)\}, \quad (6)$$

where $\mathbf{I}$ is the input image, $\mathbf{W}$ are the state variables, $\alpha$ are the parameters of the model, $Z(\mathbf{I}; \alpha)$ is the partition function and $E(\mathbf{I}, \mathbf{W}, \alpha)$ is the energy.

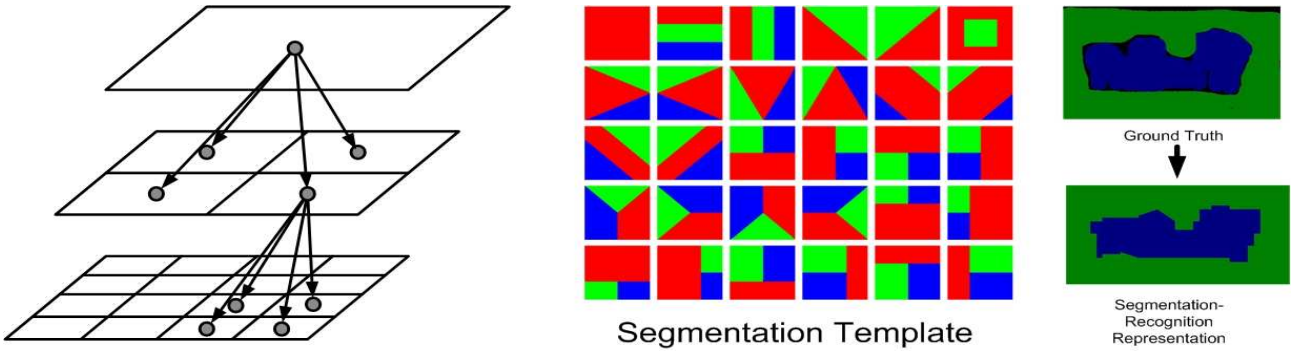


Fig. 1. The left panel shows the structure of the Hierarchical Image Model. The grey circles are the nodes of the hierarchy. All nodes, except the top node, have only one parent nodes. All nodes except the leafs are connected to four child nodes. The middle panel shows a dictionary of 30 segmentation templates (hand designed). 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).

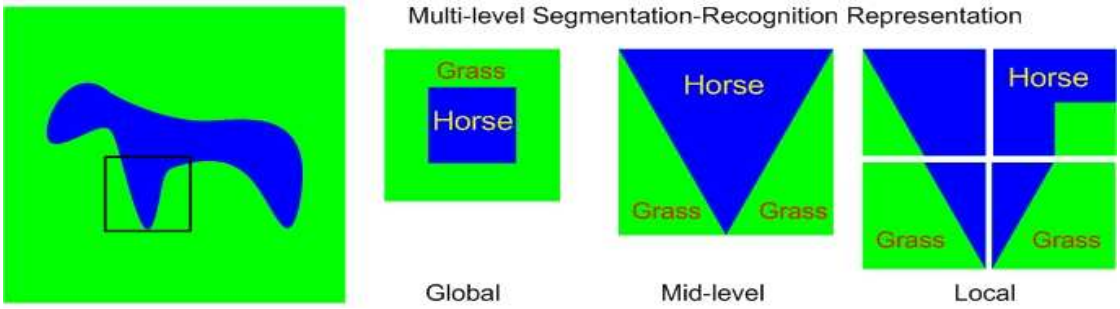


Fig. 2. This figure illustrates how the segmentation templates and labels (i.e. the S-R pairs) represent image regions in a coarse-to-fine way. The left figure is the input image which is followed by global, mid-level and local S-R pairs. The global S-R pair gives a coarse description of the object identity (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.

The energy $E(\mathbf{I}, \mathbf{W}, \alpha)$ is the sum of six energy terms $\{E_i(\mathbf{I}, \mathbf{W}, \alpha) : i = 1, .., 6\}$: (i) $E_1(\mathbf{I}, \mathbf{W}, \alpha_1)$ uses image cues to directly estimate the segmentation and labeling, (ii) $E_2(\mathbf{I}, \mathbf{W}, \alpha_2)$ favors segmentations where pixels in subregions have similar appearance, (iii) $E_3(\mathbf{W}, \alpha_3)$ encourages consistency between segmentation and labeling at adjacent levels of the hierarchy, (iv) $E_4(\mathbf{W}, \alpha_4)$ imposes constraints on the classes of adjacent sub-regions (e.g., cows are unlikely to be in the sky), (v) $E_5(\mathbf{W}, \alpha_5)$ is a segmentation prior (e.g., which segmentation templates are most likely), and (vi) $E_6(\mathbf{W}, \alpha_6)$ models the co-occurrence of the segmentation templates and the labels. We now describe these six terms in more detail.

The first term $E_1(\mathbf{I}, \mathbf{W}, \alpha_1)$ is an object specific data term which represents the image features of regions. We set:

$$E_1(\mathbf{I}, \mathbf{W}, \alpha_1) = -\sum_{\mu \in \mathcal{V}} \alpha_1 \psi_1(\mathbf{I}, s_\mu, \vec{c}_\mu)$$
$$\text{with } \psi_1(\mathbf{I}, s_\mu, \vec{c}_\mu) = \frac{1}{|R(\mu)|} \sum_{r \in R(\mu)} \log p(o_\mu^r | \mathbf{I}^\mathbf{r}) \quad (7)$$

where $\mathbf{I}^r$ is a local image region centered at the location of $r$ (its size scales with the level), $F(\cdot, \cdot)$ is a (strong) classifier learnt by multi-class boosting [39] and $p(o_\mu^r | \mathbf{I}^r)$ is given by:

$$p(o_\mu^r | \mathbf{I}^r) = \frac{\exp\{F(\mathbf{I}^r, o_\mu^r)\}}{\sum_{o'} \exp\{F(\mathbf{I}^r, o')\}} \quad (8)$$

The details of image features and boosting learning will be described in section (VI-A2).

The second term $E_2(\mathbf{I}, \mathbf{W}, \alpha_2)$ is designed to favor segmentation templates for which the pixels belonging to the same sub-regions (i.e., having the

same labels) have similar appearance. We define:

$$E_2(\mathbf{I}, \mathbf{W}, \alpha_2) = -\sum_{\mu \in \mathcal{V}} \alpha_2 \psi_2(\mathbf{I}, s_\mu, \vec{c}_\mu),$$

$$\psi_2(\mathbf{I}, s_\mu, \vec{c}_\mu) = \frac{1}{|\partial R(\mu)|} \sum_{(q,r) \in \partial R(\mu)} \phi(\mathbf{I}^r, \mathbf{I}^q | o^r_\mu, o^q_\mu) \quad (9)$$

where $\partial R(\mu)$ are the set of edges (in the image) connecting pixels $q, r$ in a neighborhood and $\phi(\mathbf{I}^r, \mathbf{I}^q | o^r_\mu, o^q_\mu)$ has form:

$$\phi(\mathbf{I}^r, \mathbf{I}^q | o^r_\mu, o^q_\mu) = \begin{cases} \gamma(r,q) & if \ o^r_\mu = o^q_\mu \\ 0 & if \ o^r_\mu \neq o^q_\mu \end{cases} \quad (10)$$

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 [40] as edge potentials (only in one level) to favors pixels with similar colour having the same labels.

The third term is defined as:

$$E_3(\mathbf{W}, \alpha_3) = -\sum_{\mu \in \mathcal{V}/\mathcal{R}: \nu = Pa(\mu)} \alpha_3 \psi_3(s_\mu, \vec{c}_\mu, s_\nu, \vec{c}_\nu) \quad (11)$$

is used to encourage consistency between the S-R pairs at consecutive levels of the hierarchy (i.e. by relating states of the nodes with those of their parents). The potential $\psi_3(s_\mu, \vec{c}_\mu, s_\nu, \vec{c}_\nu)$ is defined by the Hamming distance:

$$\psi_3(s_\mu, \vec{c}_\mu, s_\nu, \vec{c}_\nu) = \frac{1}{|R(\mu)|} \sum_{r \in R(\mu)} \delta(o^r_\mu, o^r_\nu) \quad (12)$$

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 which is imposed hierarchically to enable long-range interactions.

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

$$E_4(\mathbf{W}, \alpha_4) = -\sum_{\mu \in \mathcal{V}} \sum_{c_i, c_j = 1..M} \alpha_4(c_i, c_j) \psi_4(c_i, c_j, \vec{c}_\mu, \vec{c}_\mu) - \sum_{\mu \in \mathcal{V}/\mathcal{R}: \nu = Pa(\mu)} \sum_{c_i, c_j = 1..M} \overline{\alpha}_4(c_i, c_j) \psi_4(c_i, c_j, \vec{c}_\mu, \vec{c}_\nu) \quad (13)$$

where $\psi_4(c_i, c_j, \vec{c}_\mu, \vec{c}_\nu)$ is an indicator function which equals one if $c_i$ is a component of $\vec{c}_\mu$ and $c_j$ is a component of $\vec{c}_\nu$, and is zero otherwise. Here $\alpha_4$ is a matrix where the entries $\alpha_4(c_i, c_j)$ encodes the compatibility between two classes $c_i$ and $c_j$ at the same level. Similarly $\overline{\alpha}_4(c_i, c_j)$ gives the compatibility between class labels at different levels. 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 $E_5(\mathbf{W}, \alpha_5)$ encodes a prior on the segmentation templates (i.e. on the segmentation into sub-regions):

$$E_5(\mathbf{W}, \alpha_5) = -\sum_{\mu \in \mathcal{V}} \alpha_5 \psi_5(s_\mu), \quad (14)$$

$$\text{where } \psi_5(s_\mu) = \log p(s_\mu). \quad (15)$$

$\psi_5(s_\mu)$ is obtained directly from the training data by label counting. Finally, the sixth term $E_6(\mathbf{W}, \alpha_6)$ models the co-occurrence of the segmentation templates and the object classes.

$$E_6(\mathbf{W}, \alpha_6) = -\sum_{\mu \in \mathcal{V}} \sum_{c_j \in \vec{c}_\mu} \alpha_6 \psi_6(s_\mu, c_j) \quad (16)$$

$$\text{where } \psi_6(s_\mu, c_j) = \log p(s_\mu, c_j), \quad (17)$$

where '$c_j \in \vec{c}_\mu = 1$ if $c_j$ is a component of $\vec{c}_\mu$ and is 0 otherwise. $\psi_6(s_\mu, c_j)$ is directly obtained from training data by label counting.

In summary, we can express the energy as $E(\mathbf{I}, \mathbf{W}, \alpha) = \alpha \cdot \psi(\mathbf{I}, \mathbf{W})$ where there are a total of 676 parameters $\alpha$ obtained as follows: (i) 25 parameters for $\alpha_1, \alpha_2, \alpha_3, \alpha_5, \alpha_6$ (five for each, one for each level), (ii) 210 and 441 parameters of $\alpha_4, \overline{\alpha}_4$ (calculated by $21 \times 20/2$ and $21 \times 21$ respectively).

### C. 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

[41] high resolution buffer hypothesis for the visual cortex.

The summarization principle has four aspects.

(I) The state of $w_\nu$ 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.

## 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:

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} p(\mathbf{W}|\mathbf{I};\alpha) = \arg\min_{\mathbf{W}} E(\mathbf{I}, W, \alpha) \tag{18}$$

This will output state variables $\mathbf{W}^* = \{w^*_\mu = (s^*_\mu, \vec{c}^*_\mu) : \mu \in \mathcal{V}\}$ 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 (18). 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 $\mu$ can only assign labels $o^r_\mu$ 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 term $E(\mathbf{I}, \mathbf{W}, \alpha)$ can be computed recursively which will enable Dynamic Programming with pruning.

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

This can be computed recursively by:

$$E_\nu(\mathbf{I}, \mathbf{w}_{des(\nu)}, \alpha) = \alpha^{data}_\nu \cdot \psi^{data}_\nu(\mathbf{I}, w_\nu) + \sum_{\rho \in Ch(\nu)} \{E_\rho(\mathbf{I}, \mathbf{w}_{des(\rho)}, \alpha) + \alpha^{int}_\nu \cdot \psi^{int}_\nu(w_\nu, w_\rho)\}, \tag{19}$$

where the data terms $\alpha^{data}_\nu \cdot \psi^{data}$ are $\alpha_1\psi_1, \alpha_2\psi_2, \alpha_5\psi_5, \alpha_6\psi_6$ (i.e. the terms in section (III-B) that depend only on node $\nu$ and the data $\mathbf{I}$) and the inter-level terms $\alpha^{int}_\nu \cdot \psi^{int}_\nu$ are $\alpha_3\psi_3, \alpha_4\psi_4$ (i.e. the terms in section (III-B) that depend on node $\nu$ and its children $Ch(\nu)$).

By evaluating $E_\nu(\mathbf{I}, \mathbf{w}_{des(\nu)}, \alpha)$ at the root nodes we obtain the full energy of the model $E(\mathbf{I}, \mathbf{W}, \alpha)$.

### B. Dynamic Programming with Pruning

We can use the recursive formulation of the energy, see equation (19), 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.

DP proceeds by evaluating possible states $w_\nu$ for nodes $\nu$ of the graph. We will refer to these possible states as *proposals* and denote them by $\{w_{\nu,b_\nu}\}$, where $b_\nu$ indexes the proposals for node $\nu$. Each proposal is associated with a *minimum energy* $E_{min}(w_{\nu,b_\nu})$ which corresponds to the lowest possible energy of the subtree whose root node $\nu$ takes state $w_{\nu,b_\nu}$. More precisely, consider the set of all possible state variables $\{\mathbf{w}_{des(\nu),b_\nu}\}$ for the subtree with state $w_{\nu,b_\nu}$ at the root node $\nu$. Then set $E_{min}(w_{\nu,b_\nu}) = \min_{\mathbf{w}_{des(\nu),b_\nu}} E_\nu(\mathbf{I}, \mathbf{w}_{des(\nu),b_\nu}, \alpha)$.

*Recursion for parent nodes*: for each parent node $\mu$ we first access the proposals $\{w_{\rho,b_\rho}\}$ for its child nodes $\rho \in Ch(\mu)$ and their minimum energies $\{E_{min}(w_{\rho,b_\rho})\}$. Then for each state $w_{\mu,b_\mu}$ of $\mu$, we compute its minimum energy by solving:

$$E_{min}(w_{\mu,b_\mu}) = \alpha_\mu^{data} \cdot \psi_\mu^{data}(\mathbf{I}, w_{\mu,b_\mu}) + \sum_{\rho \in Ch(\mu)} \min_{w_{\rho,b_\rho}} \{E_{min}(w_{\rho,b_\rho}) + \alpha_\mu^{int} \cdot \psi_\mu^{int}(w_{\mu,b_\mu}, w_{\rho,b_\rho})\} \quad (20)$$

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

The pruning strategy rejects 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 into three subsets $S_1, S_2, S_3$, where $S_i$ contains partitions with $i$ 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 ('best' means lowest energy). For subset $S_3$, we keep only the best 5 labels of each subregion (hence a total of $5^3 = 125$ labels). In summary, when computing the proposals for node $\mu$, we group the proposals into three sets depending on the partition label $s_\mu$ 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).

The *top-down pass* is used to find the state configurations $\{\mathbf{W}_{\mathcal{R},b}\}$ of the graph nodes $\mathcal{V}_\mathcal{R}$ which correspond to the proposals $\{w_{\mathcal{R},b}\}$ for the root nodes. By construction, the energies of these configurations is equal to the minimum energies at the root nodes, see equation (20). This top-down pass recursively inverts equation (20) to obtain the states of the child nodes that yield the minimum energy – i.e., it solves:

$$\{w_\rho^* : \rho \in Ch(\mu)\} = \arg \min_{\{w_{\rho,b_\rho}\}} \{\alpha_\mu^{data} \cdot \psi_\mu^{data}(\mathbf{I}, w_\mu^*) + \sum_{\rho \in Ch(\mu)} \{E_{min}(w_{\rho,b_\rho}) + \alpha_\mu^{int} \cdot \psi_\mu^{int}(w_\mu^*, w_{\rho,b_\rho})\}\}, \quad (21)$$

which reduces to:

$$w_\rho^* = \arg \min_{w_{\rho,b_\rho}} \{E_{min}(w_{\rho,b_\rho}) + \alpha_\mu^{int} \cdot \psi_\mu^{int}(w_\mu^*, w_{\rho,b_\rho})\}. \quad (22)$$

We then set the optimal solution to be:

$$\hat{\mathbf{W}}^* = \{w_\mu^* : \mu \in \mathcal{V}\} \quad (23)$$

## V. Learning the Model

There are a range of different learning algorithms that we could use to estimate the parameters $\alpha$ of an HIM. These include maximum likelihood, as used in Conditional Random Fields (CRFs) [14], max-margin learning [42], and structure-perceptron learning [17].

In this paper, we use structure perceptron because of its simplicity (and our previous experience of using it in [43]). Structure-perceptron learning is simple to implement and only requires us to calculate the most probable configurations (parses) of the model, see figure (3). By contrast, maximum likelihood learning requires calculating the expectation of features which is difficult due to the large states of HIM. Moreover, Collins [17] proved theoretical results for convergence properties, for both separable and non-separable cases, and also for generalization. The structure-perceptron learning will not compute the partition function $Z(\mathbf{I}; \alpha)$ so 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 $\{\mathbf{W}^i\}$ are a set of parse trees which specify the labels of image regions in a hierarchical form (in practice, the training set only contains the labels of the pixels and we perform an approximation

to estimate the full parse $\mathbf{W}^i$ for the training set, see the implementation details in the Experimental Results section). We use a set of training examples $\{(\mathbf{I}^i, \mathbf{W}^i) : i = 1...N\}$ and the feature functions $\psi(\mathbf{I}, \mathbf{W}) \in R^d$ described in section (III-B).

The basic structure-perceptron algorithm is designed to minimize the loss function:

$$Loss(\alpha) = \alpha \cdot \psi(\mathbf{I}, \mathbf{W}) - \max_{\overline{\mathbf{W}}} \alpha \cdot \psi(\mathbf{I}, \overline{\mathbf{W}}), \quad (24)$$

where $\mathbf{W}$ is the correct parse for input $\mathbf{I}$, and $\overline{\mathbf{W}}$ is a dummy variable. We use *"the averaged parameters"* version whose pseudo-code is given in figure (3). The algorithm proceeds in a simple way (similar to the perceptron algorithm for binary 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 $\alpha$ are updated by an additive term. The most difficult step of the method is finding $\mathbf{W}^* = \arg\max_{\mathbf{W}} \alpha \cdot \psi(\mathbf{I}^i, \mathbf{W})$, which can be solved by the inference algorithm from section (IV). Hence the computational efficiency of structure perceptron (and its practicality) 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.

## VI. Experimental Results

We evaluate the segmentation performance of the HIM on two public datasets, i.e. the MSRC 21-class image datset [16] and the PASCAL VOC 2007 [18].

### A. Experiment I: MSRC

*1) Implementation details:* The MSRC 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 for learning, we estimate the true labels (i.e. the states of the S-R pair ) of the nodes in the five-level 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 – this is for the lowest $27 \times 27$ block – and shows that the quadtree representation does not cause too many artifacts. There are a total of $591$ images. We use the identical splitting as [16], 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. This was used to set the number of steps of the structure perceptron learning algorithm.

For a given image $\mathbf{I}$, the parsing result is obtained by estimating the best configuration $\mathbf{W}^*$ 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 [16]) 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 (II). The confusion matrix of 21 object classes is shown in figure (4) where the diagonal entries give the classification accuracy of individual classes. Figure (5) (best viewed in color) shows some parsing results obtained by the HIM and by the $E_1(\mathbf{I}, \mathbf{W}, \alpha)$ classifier term alone (i.e. $p(o_\mu^r|\mathbf{I})$). Observe that the HIM gives better results than the classifier term alone and hence justifies the use of the hierarchy in order to ensure long-range interactions – see improvements of $20\%$ and $32\%$ in rows 6 and 7. This improvement is quantified in Table (II) showing that HIM improves the results obtained by

**Input:** A set of training images with ground truth $(\mathbf{I}^i, \mathbf{W}^i)$ 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., $\mathbf{W}^* = \arg\max_{\mathbf{W}} \alpha \cdot \psi(\mathbf{I}^i, \mathbf{W})$
- Update the parameters: $\alpha = \alpha + \psi(\mathbf{I}^i, \mathbf{W}^i) - \psi(\mathbf{I}^i, \mathbf{W}^*)$
- Store: $\alpha^{t,i} = \alpha$

**Output:** Parameters $\gamma = \sum_{t,i} \alpha^{t,i}/NT$

Fig. 3. The structure-perceptron learning algorithm. $\alpha$ and $\psi(\mathbf{I}, \mathbf{W})$ represent all the parameters and factor functions from section (III-B).

the classifier by $6.9\%$ for average accuracy and $5.3\%$ for global accuracy. Observe also, from figure (5), that the HIM is able to roughly capture segmentation boundaries of different types of regions (e.g., cow legs, sky boundaries, etc.). Observe that the results look slightly 'blocky' due to the limited number of segmentation templates.

*4) Performance comparisons:* In table (II), we compare the performance of our approach with other successful methods [16], [44], [45]. Our approach outperforms those alternatives by $6\%$ in average accuracy and $4\%$ in global accuracy. Our boosting results are better than Textonboost [16] because of our choice of image features. This raises a question – would we get better results if we used a flat CRF instead of an HIM but with the same image features? We argue that we would not because the CRF only improves TextonBoost's performance by 3% [16], while we gain 5% by using the hierarchy (and we start with a higher baseline). Some other methods [46], [31], [15], which are worse than [44], [45] and evaluated on simpler datasets [15], [31] (less than 10 classes), are not listed here due to lack of space. We also report recent progress [47], [48] on this dataset (only available after our paper was submitted for review). In particular, Ladicky et al.[48] achieves better performance – 86% compared to our 81.4% – probable because they use more powerful unary classifiers, see table (II), and also adaptive image partitioning.

*5) Empirical convergence analysis of perceptron learning.:* The structure-perceptron learning algorithm 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 of $320 \times 200$ (6 seconds for extracting image features, 9 seconds for computing the strong classifier of boosting and 15 seconds for parsing the HIM). Figure (6) plots the convergence curves evaluated by average accuracy and global accuracy on the test set. It shows that structure-perceptron converges in $T = 20$ epochs.

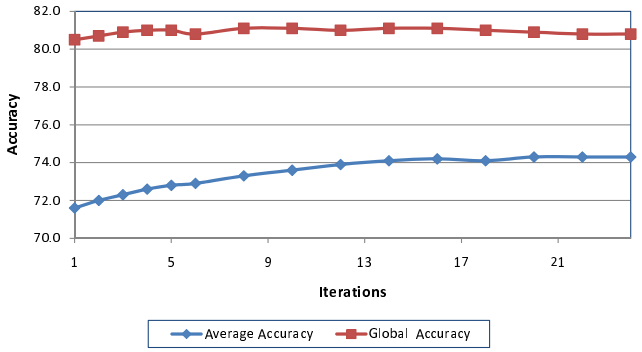


Fig. 6. Empirical Convergence Analysis on the MSRC dataset. The curves plot the average and global accuracy as a function of the number of iterations (of parameter estimation). The accuracy is evaluated on the test dataset

*6) Object part detection by S-R pairs.:* Figure (7) shows how the S-R pairs can be used to detect parts of objects. 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. These results are only approximate but they show proof of concept. In future research we will develop this idea in order to parse objects into their constituent parts.

## B. Experiment II: PASCAL VOC 2007

The PASCAL VOC 2007 dataset [18] 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 is more challenging than the MSRC-21 dataset due to more significant background clutter, illumination

| | 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. 4. The confusion matrix for object classes evaluated on the MSRC dataset [16].

| | Textonboost[16] | PLSA-MRF [44] | Auto-context [45] | Region Ancestry[47] | HCRF[48] | 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 II

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 $E_1(\mathbf{I}, \mathbf{W}, \alpha)$ TERM). THE NUMBERS IN THE BRACKETS ARE THE RESULTS OBTAINED BY THE CLASSIFIER (UNARY POTENTIAL) USED IN HCRF [48].

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

TABLE III

PERFORMANCE COMPARISONS ON THE PASCAL VOC 2007 DATASET. THREE METHODS REPORTED IN THE VOC SEGMENTATION CONTEST 2007 [18] ARE COMPARED.

effects and occlusions. We trained the HIM using the same parameter settings and features as in the experiment on the MSRC-21 dataset. Some parse results are shown in figure (8). 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 [18]. The comparisons in table (III) show that the HIM outperforms most methods and is comparable with TKK. We note that [49] obtains better results – 73.5 % correct – but they only gets 65.0 % correct on MSRC.

## 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 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

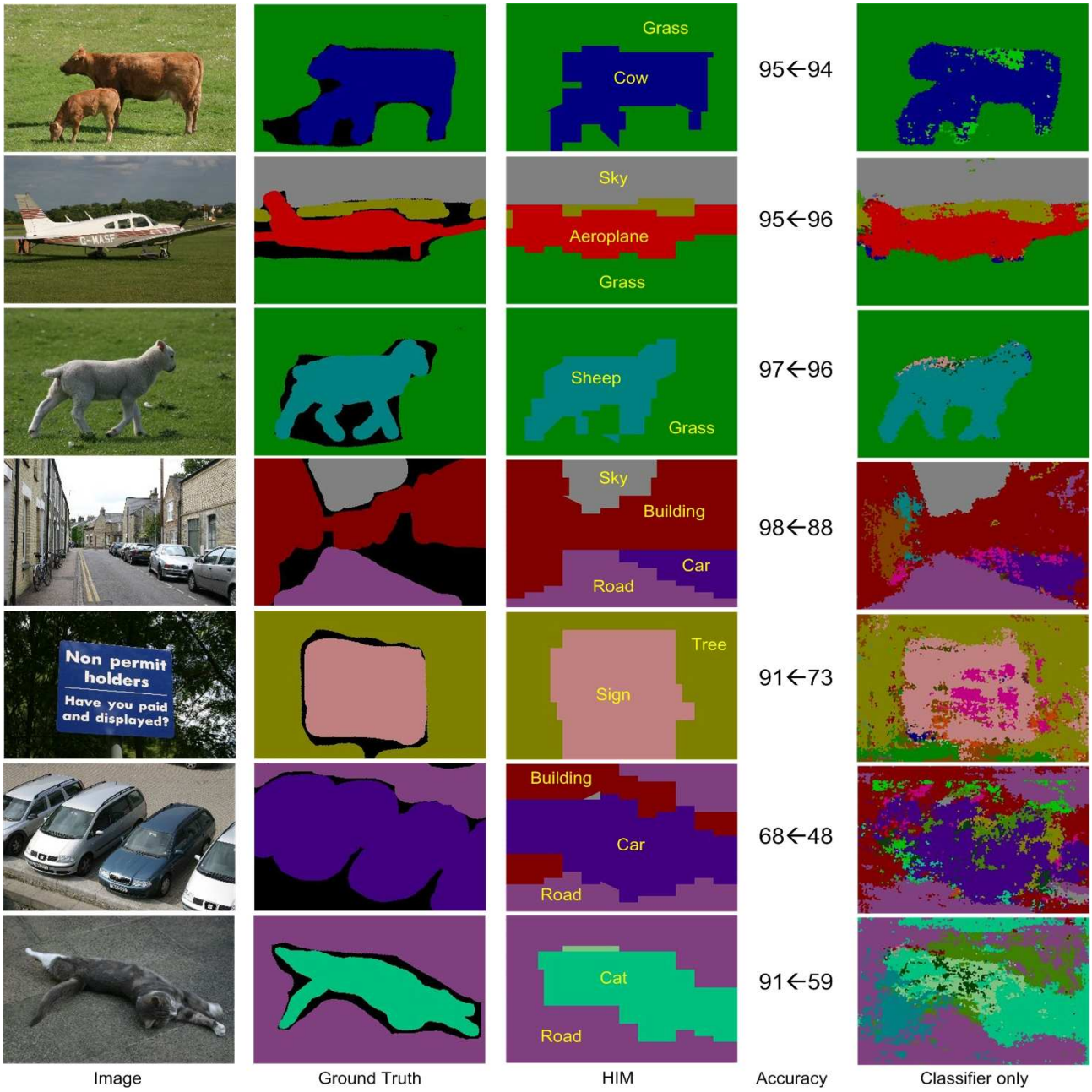


Fig. 5. This figure is best viewed in color. The colors indicate the labels of 21 object classes as in the MSRC dataset [16]. The columns (except the fourth "numerical accuracy" column) show the input images, ground truth, the labels obtained by HIM and the boosting classifier respectively. The "numerical accuracy" column shows the global accuracy obtained by HIM (left) and the classifier alone (right). In these 7 examples, HIM improves the classifier by 1%, -1% (an outlier!), 1%, 10%, 18%, 20% and 32% in terms of global accuracy.

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. Part of this research was supported by WCU (World Class University) program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology(R31-2008-000-10008-0).

## 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.

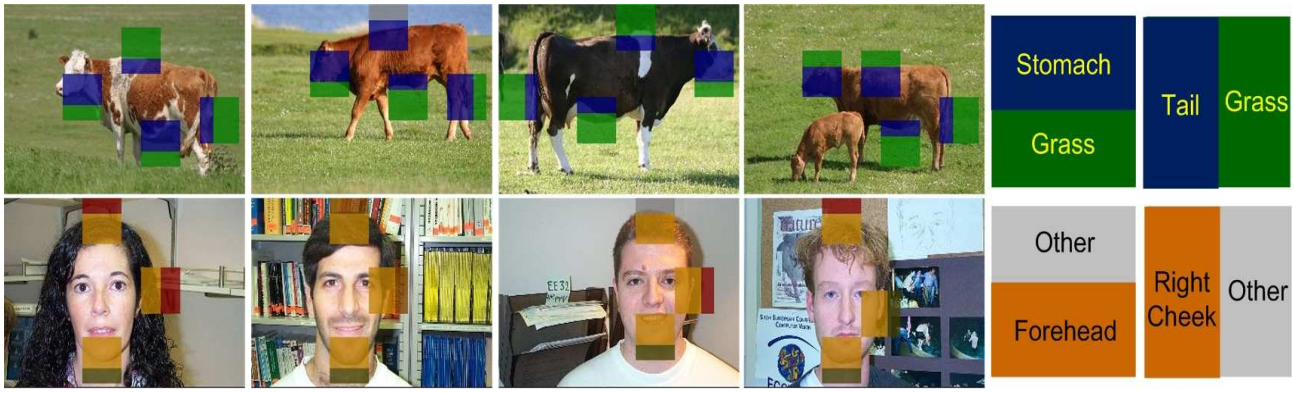


Fig. 7. The S-R pairs can be used to identify (roughly) different parts of objects. Colors indicate the object identities. Observe that the same S-R pairs (e.g., stomach above grass, and tail to the left of grass) correspond to the same object part in different images.

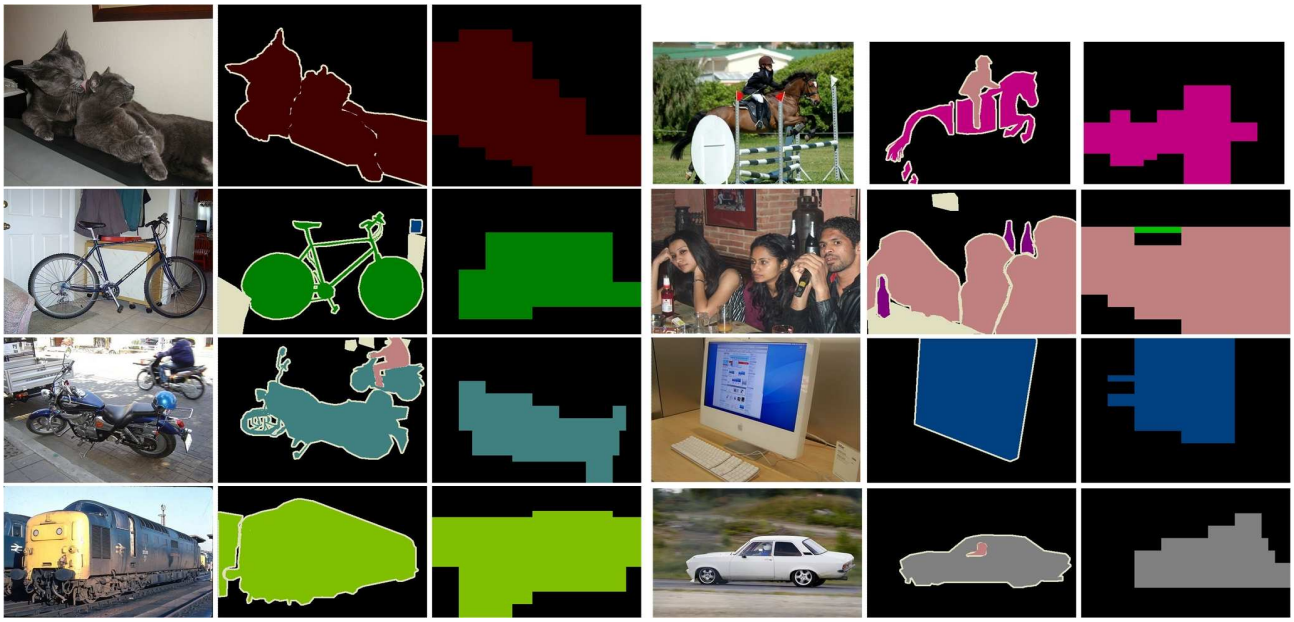

Fig. 8. Parse results on the PASCAL VOC 2007 dataset [18]. The first three columns show the input images, the groundtruth and the parse results of HIM, respectively. The next three columns show the other four examples.


[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] Y. Jin and S. Geman, "Context and hierarchy in a probabilistic image model," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, pp. 2145–2152.

[8] 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.

[9] C. Bouman and M. Shapiro, "A multiscale random field model for Bayesian image segmentation," *IEEE Trans. Image Processing*, vol. 3, no. 2, pp. 162–177, Mar. 1994.

[10] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, "Wavelet–based statistical signal processing using hidden Markov models," *IEEE Trans. Signal Processing*, vol. 46, no. 4, pp. 886–902, Apr. 1998.

[11] C. Spence, L. C. Parra, and P. Sajda, "Varying complexity in tree–structured image distribution models," *IEEE Trans. Image Processing*, vol. 15, no. 2, pp. 319–330, Feb. 2006.

[12] J. J. Kivinen, E. B. Sudderth, and M. I. Jordan, "Learning multiscale representations of natural scenes using Dirichlet processes," in *ICCV*, 2007.

[13] A. S. Willsky, "Multiresolution Markov models for signal and image processing," *Proc. of the IEEE*, vol. 90, no. 8, pp. 1396–1458, Aug. 2002.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

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

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

[21] A. Blake and A. Zisserman, *Visual Reconstruction*. MIT Press, 1987.

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

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

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

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

[26] 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.

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

[32] 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.

[33] P. J. Cowans and M. Szummer, "A graphical model for simultaneous partitioning and labeling," in *Proceedings of AI and Statistics*, 2005.

[34] 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.

[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] 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.

[37] E. Sharon, A. Brandt, and R. Basri, "Fast multiscale image segmentation," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2000, pp. 1070–1077.

[38] T. Cour, F. Benezit, and J. Shi, "Spectral segmentation with multiscale graph decomposition," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

[39] 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.

[40] 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.

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

[42] 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.

[43] 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.

[44] 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.

[45] 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.

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

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

[48] 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.

[49] G. Csurka and F. Perronnin, "A simple high performance approach to semantic segmentation," in *Proceedings of BMVC*, 2008.