Noname manuscript No.
(will be inserted by the editor)

Scene Parsing by Data Driven Cluster Sampling

Quan Zhou · Tianfu Wu · Wen-yu Liu · Song-Chun Zhu

Received: date / Accepted: date

Abstract This paper presents a data-driven cluster sampling framework for parsing scene images into generic regions (such as the sky, mountain and water) and objects (such as cows, horses and cars). We adopt generative models for both generic regions and objects, thus their likelihood probabilities are comparable and are learned under a common information projection principle. The inference algorithm follows the data-driven Markov Chain Monte Carlo (DDMCMC) paradigm where the object and generic region models cooperate and compete for an optimal interpretation of the scene in a Bayesian framework. The algorithm has two phases: (i) Bottom-up computation for generating data-driven proposals. There are two types of proposals: proposals for regular-shape objects using the active basis models and proposals for both generic regions and irregular-shape objects (such as crouching cows) by training a set of discriminative models on the appearance. A candidacy graph is constructed to summarize all the bottom-up information by treating proposals as nodes and cooperative/competitive contextual relations among proposals as +/- edges. (ii) Top-down computation by cluster sampling for seeking the optimal solution that maximizes the Bayesian posterior probability. The cluster sampling algorithm consists of reversible jumps to explore the solution space effectively. At each step, it samples the +/- edge probabilities on the candidacy graph and divides the candidacy graph into a set of composite connected-components (CCCP's) on which the reversible jumps are carried out. In experiments, our algorithm outperforms the state-of-the-art methods on the LHI 15-class dataset and obtains comparable results on the MSRC 21-class dataset (the LHI 15-class dataset has more accurate annotations and we release it with this publication).

Keywords Scene Parsing · Candidacy Graph · Cluster Sampling · DDMCMC

Q. Zhou[†,*], T.F. Wu[‡,*], W.-Y. Liu[†] and S.-C. Zhu[‡,⋆,*]
Department of [†]Electronic and Information Engineering, Hua Zhong University of Science and Technology, Wuhan, China
Department of [‡]Statistics and ⋆Computer Science, University of California, Los Angeles, USA
*Lotus Hill Research Institute (LHI), Ezhou, China
E-mail: qzhou.lhi@gmail.com, {tfwu, sczhu}@stat.ucla.edu, liuwy@mail.hust.edu.cn

1 Introduction

1.1 Motivation and overview

Real world scene images consist of two types of visual constituents: (i) objects with rigid or deformable shapes, such as cows, horses and cars, and (ii) generic regions which do not have pre-defined shapes, such as the sky, mountains and water. To overcome the recognition difficulties due to large geometric and appearance variations, researchers have explored various contextual information in the recent literature. There are two types of contextual relations: (i) Cooperative relations expressing the co-occurrence and spatial layout context. For example, cows and grass often appear together in an image, and the sky often appear on the top half of an image. (ii) Competitive constraints encoding either the exclusive relations (hard) among different interpretations for the same entity or largely overlapped entities (e.g., atomic regions or sliding windows), or the inhibitive relations (soft) among the candidates for different neighbouring disjoint entities. For example, water and the sky are not distinguishable in some atomic regions and rhinoceros and cows are ambiguous in some sliding windows.

The objective of this paper is to present a unified framework for scene parsing, which is addressed in three parts: (i) Adopting generative models for both objects and generic regions in a principled way; (ii) Studying a candidacy graph representation integrating candidates (as vertices) from various bottom-up computing processes and cooperative/competitive contextual relations (as +/- edges); (iii) Solving the inference on the candidacy graph by a clustering sampling algorithm in the process of maximizing a Bayesian posterior probability.

As illustrated in Fig.1, our framework consists of three components and we introduce them from the bottom to top as follows.

(i) *A data-driven process generating bottom-up proposals.* Given an input image, it generates two types of proposals (as illustrated in the second panel):

(a) Proposals for objects by the active basis models (Wu et al, 2010). Each object category has 2 to 4 templates depending on the shape variations. We call them the template-based proposals and represent them by diamonds.
(b) Proposals for both generic regions and objects which do not have regular shapes (e.g., the crouching cow) by training a set of state-of-the-art discriminative models on the appearance. We call them the appearance-based proposals and represent them by circles. We used the Textonboost classifiers (Shotton et al, 2009) in our experiments.

The data-driven process is to approximate the marginal posterior probabilities. These proposals are selected as candidate labels for the atomic regions after pruning some weak proposals. The threshold is set so that false negative rates (FNR) equal to zero in the validation dataset.

(ii) *A candidacy graph representation summarizing bottom-up information.* In the candidacy graph (as illustrated in the third panel), each node represents a candidate for a certain entity (atomic regions in our experiments) according to the bottom-up proposals, and then positive/negative edges are added to account for the following cooperative and competitive relations:

(a) Due to local ambiguities, an atomic region may have multiple competitive interpretations, so *hard negative edges* (represented by zigzag line segments) are assigned between those candidates, so that only one of them can be confirmed (turned "on").
(b) An object often occupies multiple atomic regions, so for atomic regions inside a template-based candidate (i.e. share a same detection bounding box), *hard positive edges* (represented by solid line segments) are assigned between their template-based candidates, so that those candidates will be confirmed (turned "on") or rejected (turned "off") together.
(c) *Soft positive/negative edges* (represented by dashed line segments) are assigned for any two candidates from adjacent atomic regions accounting for the co-occurrence and spatial layout context (but no edges between a template-based candidate and its adjacent appearance-based candidates inside the bounding box of the template-based candidate).

(iii) *A cluster sampling algorithm seeking globally optimal solutions.* Each node in the candidacy graph is either turned "on" or turned "off" such that a consistent solution is achieved to maximize the Bayesian posterior probability. In the posterior probability, generative image models are adopted for both objects and generic regions under a common framework and these models are learned using the information projection principle (Wu et al, 2010; Si et al, 2009). Our cluster sampling algorithm takes advantage of the coupling strength encoded by the positive/negative edges to construct the scene parsing results on-the-fly (as illustrated in the top panel), and has the ability of traversing any two states in the solution space through the reversible jumps. Each reversible jump is designed in three steps:

(a) A set of connected components (CCP's) is obtained by sampling the positive edges and each CCP represents a local coherent interpretation. Nodes in a CCP will be turned "on" or "off" together.
(b) A set of composite CCP's (CCCP's) is obtained by sampling the negative edges and each CCCP represents some conflicting interpretations of CCP's. So, any two linked CCP's in a CCCP must have different "on"/ "off" states.
(c) A CCCP is selected probabilistically and its state configuration is proposed to change to a new valid one. Whether the new solution is accepted or not is based on the Metropolis-Hastings method. So, the reversible jump observes detailed balance.

In addition, because we adopt the generative image models, our scene parsing results can be synthesized (e.g., the synthesized image shown in the top panel) in the spirit of analysis-by-synthesis.

We evaluate our scene parsing framework in terms of the pixel-level accuracy of the labeling and segmentation on two datasets. Our algorithm outperforms the state-of-the-art methods on the LHI 15-class dataset (8 objects and 7 generic regions) and obtains comparable and competitive results on the MSRC 21-class dataset (14 objects and 7 generic regions) (Shotton et al, 2009; L.Zhu et al, 2008). We will release the LHI 15-class dataset with this publication which consists of similar scenes but has more accurate annotations.

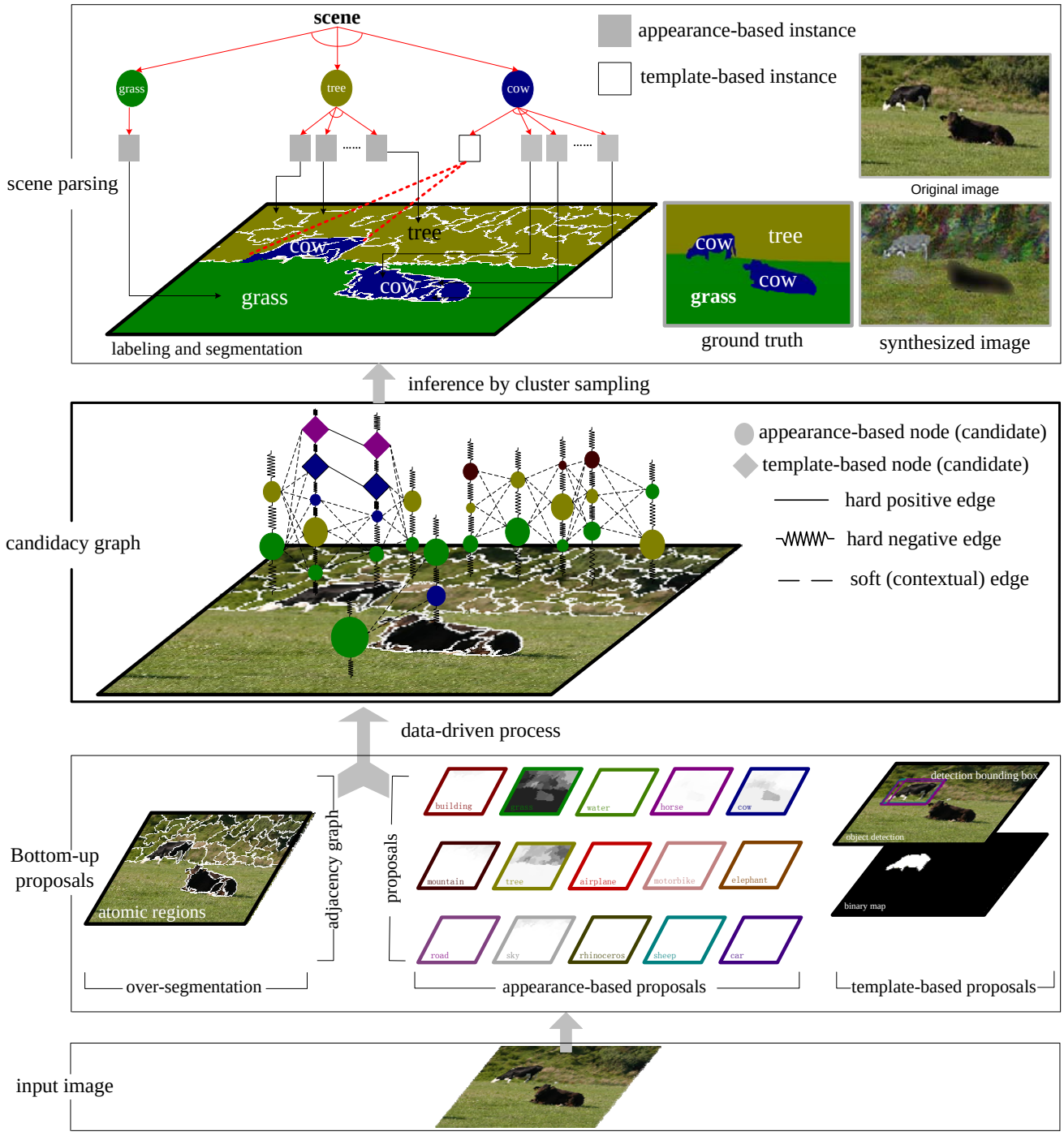


Fig. 1 Illustration of scene parsing by data-driven cluster sampling. It consists of three components: (a) A data-driven process generating bottom-up proposals. (b) A candidacy graph representation summarizing bottom-up information and then activating top-down inference. (c) A cluster sampling algorithm seeking the globally optimal solution. (Best viewed in color)

1.2 Literature review and our contributions

In the literature, there are two streams of research for scene parsing:

(i) *Discriminative methods.* Conditional random fields (CRF) (Lafferty et al, 2001) have been widely used in recent years with two components formulated in an energy function: (a) A local data term encoding pixel-based or atomic region based classification results (i.e. labels). There are two types of cues used in classification: one is purely based on bottom-up local image features (Shotton et al, 2009; Kumar and Hebert, 2005; Wojek et al, 2008; Kumar and Hebert, 2006) and the other combines bottom-up cues and top-down class-

specific features (but often limited to one class at a time) (Borenstein and Ullman, 2008; Leibe et al, 2004; Levin and Weiss, 2009; He et al, 2006; Gould et al, 2009b; Ladicky et al, 2010b; Yang et al, 2010). (b) Some pairwise relation terms expressing local or long-range context between labels such as co-occurrence (Choi et al, 2010; Rabinovich et al, 2007; Galleguillos et al, 2008; Gould et al, 2008; Tu and Bai, 2010; Desai et al, 2009; Verbeek and Triggs, 2007; Torralb et al, 2004) and geometric context (Hoiem et al, 2005, 2008) and global scene template (He et al, 2004). There are also some work using the Graph Cuts (Boykov et al, 2001; Ladicky et al, 2010a) to do the inference for the CRF models. Despite their success, these CRF-based methods only represent objects implicitly through the context defined by the label co-occurrence statistics of neighbouring pixels. When the appearance of objects has large variations or exhibits large inter-class similarities, the context terms are inefficient in expressing shape information.

(ii) *Generative methods.* A typical work is the DDMCMC algorithm. DDMCMC integrates various bottom-up computing processes to compute discriminative probabilities which approximate the marginal posterior probabilities and then drive a set of reversible jumps to explore the solution space in the process of maximizing a Bayesian posterior probability. The DDMCMC algorithms for segmentation (Tu and Zhu, 2002; Barbu and Zhu, 2005) and for image parsing (Tu et al, 2005; Corso et al, 2008) are limited in three aspects: (a) the lack of the contextual models in labeling, especially the competitive relations (i.e. the negative edges); (b) the restricted expressive power of the shape models (they only encode the boundary-based shape descriptors in the prior probability); and (c) the speed of the sampling algorithms slows down by the strongly coupled local interpretations. There are also some work integrating context into generative models (Chang et al, 2011; Jin and Geman, 2006). This paper extends DDMCMC in all the three aspects mentioned above: (a) We exploit the candidacy graph to account for both positive and negative dependencies and contexts. (b) In the posterior probability, we represent the shape information of objects explicitly through generative image models. (c) We design a cluster sampling algorithm to advance the speed of the sampling which can traverse any two different states in the solution space through reversible jumps by taking into account locally coherent sub-solutions and conflicting sub-solutions in a larger range simultaneously.

Contributions. This paper makes the following contributions to the scene parsing problem.

- It adopts generative models for both objects and generic regions, learned by a common information projection principle.
- It constructs a candidacy graph which summarizes all the bottom-up candidates as vertices and has positive/negative edges for contextual relations between objects and generic regions.
- It designs an effective cluster sampling algrithm on the candidacy graph consisting of a set of reversible jumps to explore the solution space and seek the globally optimal solutions that maximize the Bayesin posterior probability.
- It introduces a new benchmark dataset, the LHI 15-class dataset, for scene parsing with more accurate annotations.

1.3 Paper organization

The remainder of this paper is organized as follows. In Sec.2, we present a Bayesian formulation for scene parsing and the contextual models. In Sec.3, we discuss a unified generative model for both objects and generic regions, and the learning procedure using the information projection principle. In Sec.4, we present the cluster sampling algorithm on the candidacy graph. In Sec.5, we show a series of experiments on the LHI 15-class dataset and the MSRC 21-class dataset. We conclude this paper in Sec.6 with a discussion.

2 Problem Formulation

2.1 A Bayesian formulation for scene parsing

Let Λ be the image lattice (e.g. 320×210 pixels) and I_Λ an input image defined on Λ. Our objective is to parse I_Λ into objects and generic regions, represented by,

$$W = (K,\ \{\Lambda_i,\ \ell_i,\ \rho_i,\ \theta_i\}_{i=1}^{K}) \tag{1}$$

W represents a full generative interpretation including the following elements.

(i) An unknown number ($K = K_{Obj} + K_{Rgn}$) of objects (K_{Obj}) and generic regions (K_{Rgn}).

(ii) A K-partition of the image lattice Λ, denoted by

$$\pi_K = (\Lambda_1, \cdots, \Lambda_K),\ \Lambda = \cup_{i=1}^{K} \Lambda_i,\ \Lambda_i \cap \Lambda_j = \emptyset \tag{2}$$

Each Λ_i is occupied by an object or a generic region. For fast computing, our scene parsing starts with atomic regions from the mean-shift over-segmentation (Comaniciu and Meer, 2002). For I_Λ, suppose there are M atomic regions denoted by $\Lambda_1^{(a)}, \cdots, \Lambda_M^{(a)}$.

The number M depends on the granularity parameter in mean-shift algorithm and we estimate the granularity parameter in training images such that all atomic regions are pure (i.e. pixels inside an atomic region have the same label). Often, $M \in [30, 40]$ in our experiments. So, we have $1 \leq K \leq M$ and each $\Lambda_i \in \pi_K$ may be composed of one or more atomic regions.

(iii) ℓ_i is a semantic label assigned to Λ_i. Let $\mathcal{L} = Obj \cup Rgn$ be the set of semantic labels with $Obj = \{`cow', `horse', `car', \cdots\}$ and $Rgn = \{`sky', `water', \cdots\}$ ($|Obj| = 14$ and $|Rgn| = 7$ in the MSRC 21-class dataset and $|Obj| = 8$ and $|Rgn| = 7$ in the LHI 15-class dataset, see Sec.5.1 for the label list).

(iv) ρ_i is a model prototype which explains the image data I_{Λ_i}. There are two types of model prototypes, active basis model and appearance model (to be specified in Sec.2.2).

(v) A set of parameters $\theta_i = (\boldsymbol{\lambda}_{\ell_i}, \mathbf{z}_{\ell_i})$ of a probabilistic model $p(I|\rho_i; \theta_i)$ learned for the category ℓ_i given the prototype ρ_i (to be defined in Sec.3).

Under the Bayesian framework, our objective is to compute W^* that maximizes a posterior probability,

$$W^* = \arg\max_{W \in \Omega} p(W|I_\Lambda) = \arg\max_{W \in \Omega} p(W)p(I_\Lambda|W) \quad (3)$$

where Ω is the solution space for all possible W's.

In the following, we discuss the likelihood, prior probability and solution space.

2.2 The likelihood $p(I_\Lambda|W)$

2.2.1 Representation of objects and generic regions

As Fig.2 illustrates, we use two types of representations.

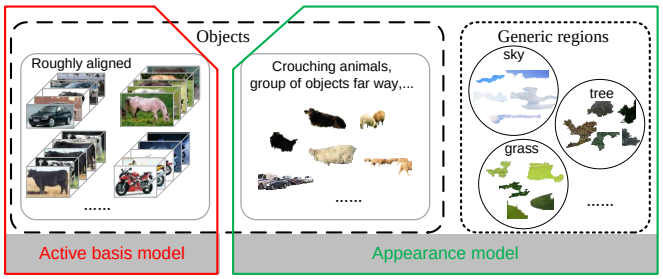


Fig. 2 Illustration of the two types of image representations.

(i) *Active basis model* (Wu et al, 2010) for objects with certain shape configuration, denoted by T_ℓ for a category $\ell \in Obj$. T_ℓ is specified by a small number n of Gabor wavelet elements at selected locations and orientations,

$$T_\ell = (B_1, \cdots, B_n) \quad (4)$$

Let $\Delta = \{B_{x,y,s,o}\}$ be the dictionary of Gabor wavelets and then $B_j \in \Delta$ with the index $j = (x, y, s, o)$. We use one scale $s = 0.7$ and 15 orientations in experiments. The image domain Λ_i occupied by an object instance in detection is decided by a segmentation mask of T_ℓ, denoted by Msk_{T_ℓ}. The mask for T_ℓ is decided by the annotations of training images.

T_ℓ defines a subspace of images through the sparse coding model,

$$\Omega(T_\ell) = \{I_i : I_i = \sum_{j=1}^{n} c_{i,j} B_{i,j} + U_i\} \quad (5)$$

where $c_{i,j}$ are coefficients, $B_{i,j}$ is B_j with local deformation in I_i and U_i the residual image. In experiments, we learn multiple active basis models (2 to 4) to handle viewpoint changes for each object category.

(ii) *Appearance model* for both generic regions and objects which do not have regular shapes (e.g., crouching animals and group of objects at distance), denoted by $\mathbf{h}_\ell$ for a category $\ell \in \mathcal{L}$. $\mathbf{h}_\ell$ is specified by a small number m of histograms pooled from selected filter responses in a filter bank (denoted by $\mathbb{F}$),

$$\mathbf{h}_\ell = (h_1, \cdots, h_m) \quad (6)$$

where h_j's characterize the texture or color. In our experiments, $m = 18$ and we pool $(h_1, \cdots, h_{15})$ from the responses of the 15 orientations of Gabor filters and pool (h_{16}, h_{17}, h_{18}) from the R-, G-, B-space in the RGB color format.

$\mathbf{h}_\ell$ also defines a subspace of images but through an implicit function of images,

$$\Omega(\mathbf{h}_\ell) = \{I_i : H_j(I_i) = h_j + \epsilon_i;\ 1 \leq j \leq m\} \quad (7)$$

where ϵ_i is the residual. In our experiments, we use one appearance model for each object and generic region category.

So, for the model prototype ρ_i in Eqn.1, we have,

$$\rho_i = \begin{cases} \mathbf{h}_{\ell_i} \text{ or } T_{\ell_i} & \text{if } \ell_i \in Obj \\ \mathbf{h}_{\ell_i} & \text{if } \ell_i \in Rgn \end{cases} \quad (8)$$

2.2.2 Factorizing the likelihood

Due to the non-overlapping Λ_i's, the image models $p(I_{\Lambda_i}|\rho_i; \theta_i)$ are assumed to be independent conditioning on W and for I_{Λ_i} described by the appearance

model (i.e. $\rho_i = \mathbf{h}_{\ell_i}$), the likelihoods are further factorized onto atomic regions ($\Lambda_j^{(a)} \subseteq \Lambda_i$). So, we have,

$$p(I_\Lambda|W) = \prod_{i:\rho_i=T_{\ell_i}} p(I_{\Lambda_i}|T_{\ell_i};\theta_i) \times \prod_{i:\rho_i=\mathbf{h}_{\ell_i}} [\prod_{\Lambda_j^{(a)}\subseteq\Lambda_i} p(I_{\Lambda_j^{(a)}}|\mathbf{h}_{\ell_i};\theta_i)] \quad (9)$$

We will discuss the learning of $p(I_{\Lambda_i}|T_{\ell_i},\theta_i)$ and $p(I_{\Lambda_j^{(a)}}|\mathbf{h}_{\ell_i},\theta_i)$ using the information projection principle (Pietra et al, 1997; Wu et al, 2010; Si et al, 2009) in Sec.3.

2.3 The prior probability $p(W)$

The prior probability $p(W)$ encodes the preference and compatibilities among the elements in W, including four aspects.

(i) An exponential model for K_{Obj} and K_{Rgn},

$$p(K_{Obj}, K_{Rgn}) \propto \exp\{-\beta_{Obj}K_{Obj} - \beta_{Rgn}K_{Rgn}\} \quad (10)$$

where β_{Obj} and β_{Rgn} are the parameters ($\beta_{Obj} = \beta_{Rgn} = 1$ in our experiments).

(ii) A Gaussian model for the preferred locations of objects. Let (x_i, y_i) be the center of the bounding box of $(\Lambda_i|\ell_i \in Obj, \rho_i = T_{\ell_i})$, and (w, h) as the width and height of the image lattice Λ. Let $\mathbf{x}_i = x_i/w$ and $\mathbf{y}_i = y_i/h$ representing the relative location. We have,

$$p(\mathbf{x}_i, \mathbf{y}_i) = \mathcal{N}(\mathbf{x}_i, \mathbf{y}_i; \mu_{\ell_i}, \Sigma_{\ell_i}) \quad (11)$$

where $\mathcal{N}()$ represents the Gaussian distribution, μ_{ℓ_i} is mean of the relative location of objects ℓ_i, and Σ_{ℓ_i} the covariance matrix. We estimate μ_ℓ and Σ_ℓ $(\forall \ell \in Obj)$ based on the annotations in the training dataset.

(iii) A contextual model for the pairwise co-occurrence between two semantic labels ℓ_i and ℓ_j of adjacent atomic regions $\Lambda_i^{(a)}$ and $\Lambda_j^{(b)}$,

$$p(\ell_i, \ell_j) = \frac{h(\ell_i)h(\ell_j|\ell_i) + h(\ell_j)h(\ell_i|\ell_j)}{2} \quad (12)$$

where $h(\ell_i)$ and $h(\ell_j)$ are the frequencies of ℓ_i and ℓ_j appearing in the images respectively and $h(\ell_j|\ell_i)$ is the frequency of ℓ_j appearing as an adjacent neighbour of ℓ_i (and vice verse for $h(\ell_i|\ell_j)$). The four terms are estimated in the training dataset based on the atomic regions and the annotations. Fig.3 illustrates the learned co-occurrence.

(iv) A contextual model for the spatial layout surrounding an object. As illustrated in left-top panel in Fig.4, for $\ell_i \in Obj$, we use the center (x_i, y_i) of the bounding box of Λ_i as the origin point and then divide

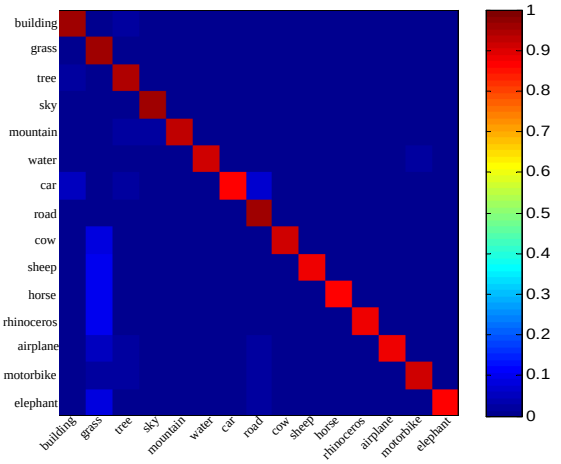


Fig. 3 Illustration of the learned pairwise co-occurrence on the LHI 15-class dataset. The probabilities are scaled to color-levels from blue (low probability) to red (high probability).

the image lattice Λ into ten sectors equally (denoted by Q_o, $1 \le o \le 10$ and represented by Roman numbers I, $\cdots$, X). In each quadrant Q_o, we measure the co-occurrence between ℓ_i and ℓ_j individually and independently (where $\Lambda_j \in Q_o$ and $\ell_j \in \mathcal{L}$) as illustrated in the right-top panel in Fig.4 and we have

$$p(\ell_i|Q_{o=1}^{10}) = \prod_{o=1}^{10} \prod_{\Lambda_j \in Q_o} h(\ell_j|\ell_i, Q_o) \quad (13)$$

where $h(\ell_j|\ell_i, Q_o)$ is estimated in the similar way for $h(\ell_j|\ell_i)$ in Eqn.12. The bottom panel in Fig. 4 shows the learned spatial layout context models for the 8 objects in LHI 15-class dataset.

In summary, we have the prior probability,

$$p(W) \propto p(K_{Obj}, K_{Rgn}) \times \prod_{i:\rho_i=T_{\ell_i}} [\underbrace{p(\mathbf{x}_i, \mathbf{y}_i)}_{\text{Rel-Loc}} \times \underbrace{p(\ell_i|Q_1^{10})}_{\text{SpatialLayout}}] \times \prod_{<i,j>} \underbrace{p(\ell_i, \ell_j)}_{\text{Co-occ}} \quad (14)$$

2.4 Anatomy of the solution space Ω

The solution space Ω for W's (Eqn.1) is a mixture of many subspaces depending on the partition of image lattice Λ. Each subspace in turn is a product space of image models describing all the sub-lattice Λ_i's.

Partition space. Denote by $\Omega_{\pi_K} \ni \pi_K$ the K-partition space of all possible K partitions (Eqn.2). Let Ω_π be the partition space, we have,

$$\Omega_\pi = \cup_{K=1}^{M} \Omega_{\pi_K} \quad (15)$$

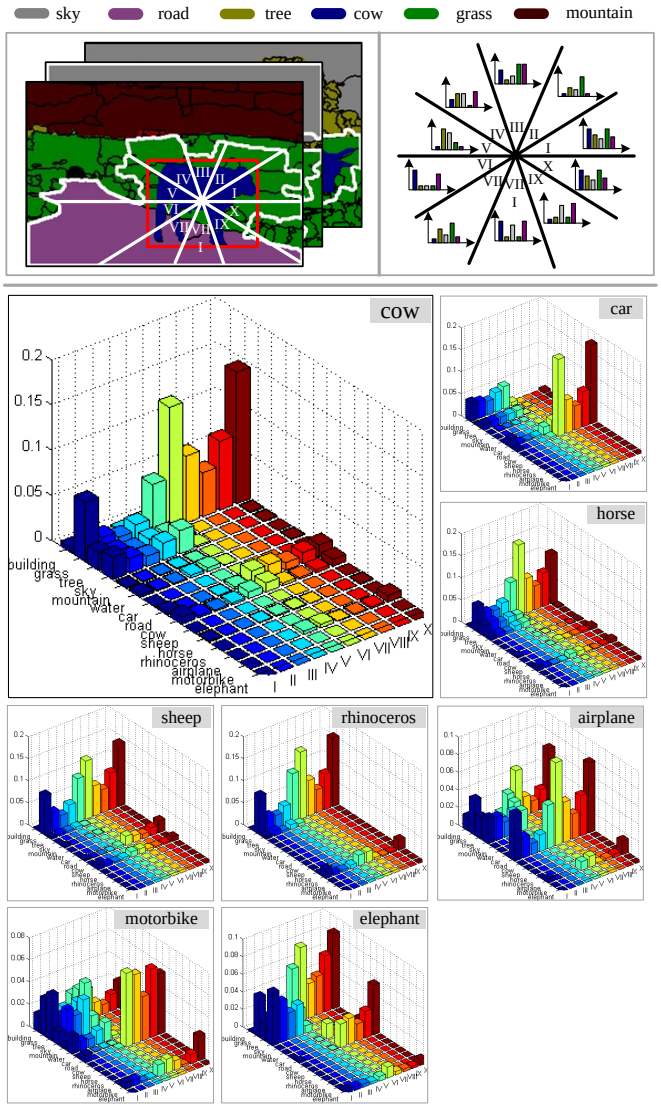


Fig. 4 The top panel illustrates the spatial layout context. The bottom panels shows the learned spatial layout context for the 8 objects in LHI 15-class dataset.

Model space. For a given K-partition π_K, each element Λ_i is characterized by $\omega_i = (\ell_i, \rho_i, \theta_i)$. Denote by $\Omega_\omega \ni \omega_i$ the model space. We have $|\mathcal{L}|$ appearance models and about $2|Obj|$ to $4|Obj|$ active basis models.

Now, the solution space Ω is specified based on the partition space and the model space,

$$\Omega = \cup_{K=1}^{M} \{\Omega_{\pi_K} \times \underbrace{\Omega_\omega \times \cdots \times \Omega_\omega}_{K}\} \tag{16}$$

Designing Markov chain to traverse Ω. The posterior probability $p(W|I_\Lambda)$ not only has possibly enormous number of local maxima but is distributed over subspaces of varying dimensions. To seek globally optimal solutions, we design the Markov chain to traverse Ω by a number of reversible jumps which observe three properties: (i) irreducibility to ensure any two states in the solution space are reachable in finite steps, (ii) aperiodicity ensured by using the jumps at random, and (iii) detailed balanced ensuring $p(W|I_\Lambda)$ is the stationary probability. The solution space Ω is too huge to allow exhaustive search using MCMC. So, we need to use the bottom-up data-driven information to activate and guide the search as it was in DDMCMC.

3 Learning the image models

In this section, we learn the likelihood models in Sec.2.2 by the information projection principle (Wu et al, 2010). The active basis model $p(I_{\Lambda_i}|\rho_i = T_{\ell_i}; \theta_i)$ and appearance model $p(I_{\Lambda_j^{(a)}}|\rho_i = \mathbf{h}_{\ell_i}; \theta_i)$ follow the same learning procedure. So, for notation simplicity, we use $p(I|\rho; \theta)$ in the following.

Training data. For a given category $\ell \in \mathcal{L}$, let $D^+ = \{I_1, \cdots, I_N\}$ be a set of positive images which are samples from a underlying probability $f(I|\rho)$.

(i) In learning the active basis model T_ℓ ($\ell \in Obj$) for objects, positive images I_i's are image patches cropped from the annotated images, and are aligned roughly by similarity transformations. The segmentation mask Msk_{T_ℓ} is computed from the annotated label maps.

(ii) In learning the appearance model $\mathbf{h}_\ell$ ($\ell \in Obj \cup Rgn$) for both generic regions and objects, the histograms h_j's are pooled based on atomic regions belonging to the category ℓ according to the annotated images. So, each positive image I_i refers to an atomic region obtained from the over-segmentation for each category in the annotated label map.

Let D^- be a collection of images used to represent a reference model $q(I)$ (which only need to be specified implicitly in our learning procedure), and we use the whole training dataset as the reference in experiments.

Our learning procedure will proceed as follows: (i) we first specify the distance metrics for a basis prototype $B_j \in \Delta$ in the image space $\Omega(T_\ell)$ (Eqn.5) and a histogram prototype h_j according to the filter bank $\mathbb{F}$ in the image space $\Omega(\mathbf{h}_\ell)$ (Eqn.7) respectively, and then (ii) derive a log-linear form model of $p(I|\rho; \theta)$ starting from the reference model $q(I)$ in term of information projection and (iii) estimate the parameters θ using maximum likelihood estimation (MLE).

(i) For $B_j \in \Delta$ and $I_i \in \Omega(T_\ell)$. Let $\Lambda_{(j)}$ be the domain of B_j and $\Lambda_{(i,j)}$ be the domain of the deformed $B_{i,j}$ in I_i, the distance is measured in the image space,

$$\mathbf{d}_j^{ex}(I_{\Lambda_{(i,j)}}) = ||I_{\Lambda_{(i,j)}} - c_{i,j} B_{i,j}||^2 \tag{17}$$

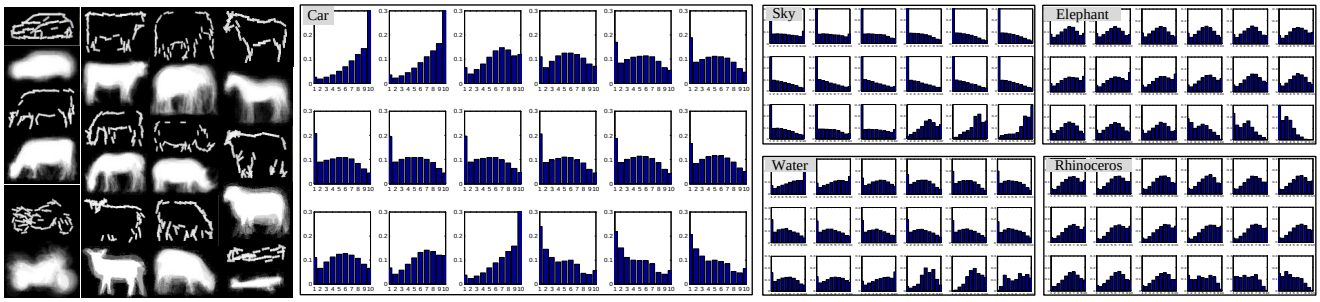


Fig. 5 The left-top panel show some learned active basis models (T_ℓ) and their associated binary segmentation masks (Msk_{T_ℓ}) for objects. The right-top and the bottom panel show some learned appearance models ($\mathbf{h}_\ell$, the first 15 histogram prototypes for describing texture and the last 3 ones for color) for both objects and generic regions.

(ii) For h_j and $I_i \in \Omega(\mathbf{h}_\ell)$, the distance is measured in the projected histogram space and we use the histogram intersection kernel,

$$\mathbf{d}_j^{im}(I_i) = \sum_{b=1}^{\mathbf{b}} \min(H_j(I_i)[b], h_j[b]) \tag{18}$$

where $\mathbf{b}$ is the number of bins of the histogram.

In learning, we need to take into account the statistical fluctuations of $\mathbf{d}_j^{ex}$ and $\mathbf{d}_j^{im}$ in the training datasets D^+ and D^-. So, we use a sigmoid function (denoted by Sigmoid$(\cdot)$) to do the soft transformation of the measured distance (we use $\mathbf{d}_j$ for simplicity),

$$r_j(I) = \text{Sigmoid}(\mathbf{d}_j(I)) \tag{19}$$

and the sigmoid function is defined as,

$$\text{Sigmoid}(x) = \tau(\frac{2}{1+e^{-2(\eta-x)/\tau}} - 1) \tag{20}$$

where τ is the saturation parameter and η the translation parameter.

Learning by information projection. Our objective of learning the model $p(I|\rho;\theta)$ from D^+ is to minimize the Kullback-Leibler divergence (KL) between the underlying probability $f(I|\rho)$ and our model $p(I|\rho;\theta)$ in the model space Ω_p. The learning starts from the reference model $p_0(I|\rho;\theta) = q(I)$. This procedure is equal to maximize the KL divergence between $p(I|\rho;\theta)$ and the reference model $q(I)$. The model space Ω_p is defined as,

$$\Omega_p = \{p(I|\rho;\theta) : E_p[r_j(I)] = E_f[r_j(I)], \forall j\} \tag{21}$$

Then, as addressed in (Zhu et al, 2010; Si et al, 2009), we have,

$$\begin{aligned} p^*(I|\rho;\theta) &= \arg\min_{p\in\Omega_p} \text{KL}(f||p) \\ &= \arg\max_{p\in\Omega_p} \text{KL}(f||q) - \text{KL}(f||p) \\ &= \arg\max_{p\in\Omega_p} \text{KL}(p||q) \end{aligned} \tag{22}$$

By solving the optimization problem, we have the following log-linear form model for both shape and appearance,

$$p(I|\rho;\theta) = q(I)\frac{1}{Z}\exp\{\sum_{j=1}^{J}\lambda_{\ell,j}r_j(I)\} \tag{23}$$

where $J = n$ for the active basis model or $J = m$ for the appearance model and Z is the partition function which is not easy to calculate in general.

To simplify the model, we apply the conditional independence assumption among $r_j(I)$'s in that (i) for learning the active basis model T_ℓ, the selected basis prototypes B_j's are approximately spatially disjoint (i.e. independent) by applying the local inhibition at each step in learning, and (ii) for learning a histogram prototype h_j in the appearance model $\mathbf{h}_\ell$, we already use the disjoint atomic regions (but different h_j's can share some atomic regions). So, we have a factorized log-linear form,

$$p(I|\rho;\theta) = q(I)\prod_{j=1}^{J}[\frac{1}{z_{\ell,j}}\exp\{\lambda_{\ell,j}r_j(I)\}] \tag{24}$$

Then, the total information gain for a model is,

$$\text{IG}(I|\rho;\theta) = \log\frac{p(I|\rho;\theta)}{q(I)} = \sum_{j=1}^{J}[\lambda_{\ell,j}r_j(I) - z_{\ell,j}] \tag{25}$$

Parameter estimation of $\theta = (\boldsymbol{\lambda}, \mathbf{z})$. Given D^+ and D^-, for B_j or h_j, we can obtain a set of positive responses $r_j^+ = \{r_j(I_1), \cdots, r_j(I_N)\}$ from D^+ and a set of reference responses r^- from D^-. Firstly, the reference mode $q(r_j)$ is estimated by pooling a histogram from r^-. Then, we estimate (λ_j, z_j) by fitting the density $p(r_j)$ to r^+ based on MLE. More details are referred to (Si et al, 2009).

In Fig.5, the left panel shows some learned active basis models and its associated binary segmentation masks and the right panel shows the histogram prototypes.

4 Inference by cluster sampling

4.1 Generating bottom-up proposals for atomic regions

In DDMCMC (Tu et al, 2005; Tu and Zhu, 2002), data-driven processes approximate the marginal posterior probabilities in various model spaces Ω_ω. In our framework, we generate proposals for each atomic region $\Lambda_j^{(a)}$,

$$p(\ell, \rho | I_{\Lambda_j^{(a)}}) = p(\rho | I_{\Lambda_j^{(a)}}) p(\ell | I_{\Lambda_j^{(a)}}, \rho) \quad (26)$$

where $p(\rho | I_{\Lambda_j^{(a)}})$ follows the uniform distribution over the set $\{\mathbf{h}_\ell; \ell \in \mathcal{L}\} \cup \{T_\ell; \ell \in Obj\}$. So, we approximate $p(\ell | I_{\Lambda_j^{(a)}}, \mathbf{h}_\ell)$ for both objects and generic regions ($\ell \in \mathcal{L}$) and $p(\ell | I_{\Lambda_j^{(a)}}, T_\ell)$ for objects only ($\ell \in Obj$).

(i) To approximate $p(\ell | I_{\Lambda_j^{(a)}}, \mathbf{h}_\ell)$, we train the Textonboost classifiers (Shotton et al, 2009), which directly estimate $p(\ell | I_{(x,y)}, \mathbf{h}_\ell)$ for each pixel (x, y) through the joint boosting based on the image features extracted in the local image patch and we have $\sum_{\ell \in \mathcal{L}} p(\ell | I_{(x,y)}, \mathbf{h}_\ell) = 1$. In order to prune weak proposals, we need to estimate a threshold (denoted by $\tau_{app}^{(\ell)}$) for each category ℓ. By using the annotated label maps in a validation dataset, we obtain a positive validation dataset for category ℓ, denoted by D_ℓ^+, and then we set the threshold $\tau_{app}^{(\ell)}$ such that the FNR equals to zero,

$$\tau_{app}^{(l)} = \min_{I_{(x,y)} \in D_\ell^+} p(\ell | I_{(x,y)}, \mathbf{h}_\ell)$$

In Tabel.1, the second column shows the estimated thresholds for the LHI 15 categories. We call the outputs from Textonboost classifiers the appearance-based proposal maps (or saliency maps), denoted by $S_{app}^{(\ell)}$,

$$S_{app}^{(\ell)}(x, y) = p(\ell | I_{(x,y)}, \mathbf{h}_\ell) \mathbf{1}(p(\ell | I_{(x,y)}, \mathbf{h}_\ell) \geq \tau_{app}^{(\ell)}) \quad (27)$$

where $\mathbf{1}(\cdot)$ is the boolean function which equals to 1 if the condition is satisfied and 0 otherwise. So, for $\Lambda_j^{(a)}$, we have the average score from appearance,

$$\bar{S}_{app}^{(\ell)}(\Lambda_j^{(a)}) = \frac{1}{|\Lambda_j^{(a)}|} \sum_{(x,y) \in \Lambda_j^{(a)}} S_{app}^{(\ell)}(x, y) \quad (28)$$

(ii) To approximate $p(\ell | I_{\Lambda_j^{(a)}}, T_\ell)$, we use the active basis models, which output the so-called SUM2 map for each object category by calculating the total information gain of the template centered at each pixel (x, y),

$\text{SUM2}^{(\ell)}(x, y) = \text{IG}(I_{(x,y)} | T_\ell; \theta)$ (see Eqn.25)

We normalize $\text{SUM2}^{(\ell)}$ to estimate $p(\ell | I_{(x,y)}, T_\ell)$ individually for each object category. Similarly, by using a validation dataset and the ground-truth bounding boxes of object instances, we estimate the thresholds for each object category, denoted by $\tau_{obj}^{(\ell)}$ (see the third column in Table.1, note that $\tau_{app}^{(\ell)}$ is relatively smaller than $\tau_{obj}^{(\ell)}$ for $\ell \in Obj$ because of their different normalizations). Denote by $S_{obj}^{(\ell)}$ the template-based proposal map, which is generated based the normalized SUM2 maps and the segmentation mask Msk_{T_ℓ} iteratively. For each object category, we first obtain a current best location $(x^*, y^*) = \arg\max p(\ell | I_{(x,y)}, T_\ell)$. Then, by centering the mask Msk_{T_ℓ} at (x^*, y^*), we do the inhibition setting $p(\ell | I_{(x,y)}, T_\ell) = 0$ if $\text{Msk}_{T_\ell}(x, y | x^*, y^*) = 1$. So, we obtain the template-based proposal map,

$$\begin{aligned} S_{obj}^{(\ell)}(x, y) =& p(\ell | I_{(x^*,y^*)}, T_\ell) \mathbf{1}(p(\ell | I_{(x^*,y^*)}) \geq \tau_{obj}^{(\ell)}) \times \\ & \mathbf{1}(\text{Msk}_{T_\ell}(x, y | x^*, y^*) = 1) \end{aligned} \quad (29)$$

Similarly, we have the average score from template,

$$\bar{S}_{obj}^{(\ell)}(\Lambda_j^{(a)}) = \frac{1}{|\Lambda_j^{(a)}|} \sum_{(x,y) \in \Lambda_j^{(a)}} S_{obj}^{(\ell)}(x, y) \quad (30)$$

Let $\mathbf{S} = \sum_{\ell \in \mathcal{L}} \bar{S}_{app}^{(\ell)}(\Lambda_j^{(a)}) + \sum_{\ell \in Obj} \bar{S}_{obj}^{(\ell)}(\Lambda_j^{(a)})$, we estimate the marginal posterior probabilities

$$p(\ell | I_{\Lambda_j^{(a)}}, \mathbf{h}_\ell) = \frac{\bar{S}_{app}^{(\ell)}(\Lambda_j^{(a)})}{\mathbf{S}} \quad (31)$$

and

$$p(\ell | I_{\Lambda_j^{(a)}}, T_\ell) = \frac{\bar{S}_{obj}^{(\ell)}(\Lambda_j^{(a)})}{\mathbf{S}} \quad (32)$$

Table 1 The estimated thresholds on the LHI 15-class dataset

Class	Textonboost ($\tau_{app}^{(\ell)}$)	Active basis ($\tau_{obj}^{(\ell)}$)
building	0.24	-
grass	0.13	-
tree	0.27	-
sky	0.21	-
mountain	0.18	-
water	0.10	-
road	0.14	-
car	0.17	0.75
cow	0.19	0.54
sheep	0.12	0.57
horse	0.18	0.51
rhinoceros	0.25	0.67
airplane	0.14	0.73
motorbike	0.12	0.71
elephant	0.14	0.64

4.2 Candidacy graph construction

A candidacy graph is constructed to summarize all the bottom-up proposals and the contextual relations, and is built up on the adjacency graph.

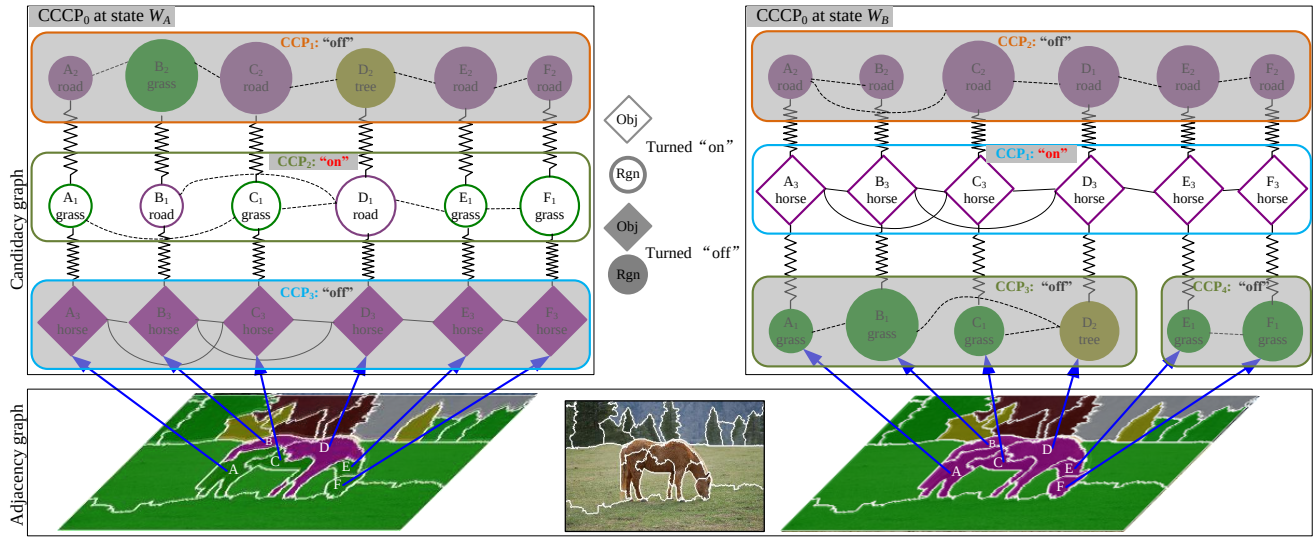


Fig. 6 Illustration of the adjacency graph of atomic regions, the candidacy graph (only a portion is shown for clarity), CCP and CCCP, and the reversible jumps between two states W_A and W_B. See texts for details. (Best viewed in color)

Adjacency graph. Let $V = \{\Lambda_1^{(a)}, \cdots, \Lambda_M^{(a)}\}$ be the set of atomic regions for the image I_Λ. As illustrated in the bottom panel in Fig.6, the adjacency graph of atomic regions is defined as,

$$G_{adj} =< V,\ E_{adj} > \tag{33}$$

where $E_{adj} = \{e_{j,k}^{(adj)} =< \Lambda_j^{(a)}, \Lambda_k^{(a)} >\}$ is the set of undirected edges linking adjacent atomic regions.

Candidates of atomic regions. Denoted by $\mathcal{C}_j$ the set of candidates for the atomic region $\Lambda_j^{(a)}$ and $\mathcal{C} = \cup_{j=1}^M \mathcal{C}_j$ all the candidates from all the atomic regions. Each $\mathcal{C}_j$ consists of a number t of template-based candidates (denoted by $\mathcal{C}_j^{obj}$, represented by diamonds, see Fig.6) from the template-based proposal map $S_{obj}^{(\ell)}$'s and a number h of distinct appearance-based candidates (denoted by $\mathcal{C}_j^{app}$, represented by circles) from the appearance-based proposal map $S_{app}^{(\ell)}$'s,

$$\begin{aligned} \mathcal{C}_j &= \mathcal{C}_j^{obj} \cup \mathcal{C}_j^{app} \\ &= \{O_{j,s} = (\ell_{j,s}, p_{j,s}, f_{j,s}, b_{j,s}); 1 \le s \le t\} \cup \\ &\quad \{R_{j,g} = (\ell_{j,t+g}, p_{j,t+g}, f_{j,t+g}); 1 \le g \le h\} \end{aligned} \tag{34}$$

where $0 \le t \le |Obj|$ and $0 \le h \le |\mathcal{L}|$, and the weight $p_{j,s} = p(\ell_{j,s}|I_{\Lambda_j^{(a)}}, T_{\ell_{j,s}})$ (Eqn.32) and $p_{j,t+g} = p(\ell_{j,t+g}|I_{\Lambda_j^{(a)}}, \mathbf{h}_{\ell_{j,t+g}})$ (Eqn.31) (illustrated by the sizes of the circles and diamonds, see Fig.6), and $f_{j,c} \in \{0,1\}$ is a binary flag (i.e. "off" and "on") for the candidate label $\ell_{j,c}$ which will be assigned during the cluster sampling algorithm, and $b_{j,s}$ is the index of the bounding box which cover the atomic region $\Lambda_j^{(a)}$.

Candidacy graph. As illustrated in the top panel in Fig.6, the candidacy graph is defined as,

$$G_{cand} =< \mathcal{C},\ E >,\ \ E = E^+ \cup E^- \tag{35}$$

where E^+ is the positive edge set indicating the cooperative contextual relations and E^- is the negative edge set representing the competitive constraints. The candidates in $\mathcal{C}$ inherit the adjacency relations based on the adjacency graph.

On each edge $e \in E$, we define an edge probability $q(e)$ accounting for the coupling strength, and an auxiliary binary variable $\mu_e \in \{0,1\}$ (i.e. "off" and "on"). The edge probabilities play the computational role for generating CCP's and CCCP's later on (against the modeling role of context models in the prior model).

(i) **The positive edge set** ($E^+ = E_{hrd}^+ \cup E_{sft}^+$).

E_{hrd}^+ is a set of hard positive edges (represented by solid line segments, see Fig.6) between any two adjacent template-based candidates for atomic regions inside the same bounding box from the active basis detection,

$$E_{hrd}^+ = \{< O_{j,s}, O_{k,s'} >;\ \exists\ e_{j,k}^{(adj)} \in E_{adj}\} \tag{36}$$

where $\ell_{j,s} = \ell_{k,s'}$ and $b_{j,s} = b_{k,s'}$ are also satisfied. The edge probability is,

$$q(e \in E_{hrd}^+) = 1 \tag{37}$$

which means that the nodes $O_{j,s}$ and $O_{k,s'}$ are coupled and always turn on or off together.

$E_{sft}^+ = E_{sft,1}^+ \cup E_{sft,2}^+ \cup E_{sft,3}^+$ is a set of soft positive edges (represented by dashed line segments, see Fig.6) which further consists of three cases:

(a) $E^+_{sft,1}$ is the set of soft positive edges between any two adjacent appearance-based candidates whose co-occurrence probability is greater than a threshold τ_{occ} ($\tau_{occ} = 0.15$ in our experiments),

$$E^+_{sft,1} = \{< R_{j,g}, R_{k,g'} >;\ \exists\ e^{(adj)}_{j,k} \in E_{adj}\} \quad (38)$$

where $p(\ell_{j,g}, \ell_{k,g'}) > \tau_{occ}$ (Eqn.12).

(b) $E^+_{sft,2}$ is the set of soft positive edges between any two adjacent template-based candidates which do not share a same bounding box and whose co-occurrence probability is greater than the threshold τ_{occ},

$$E^+_{sft,2} = \{< O_{j,s}, O_{k,s'} >; \exists e^{(adj)}_{j,k},\ b_{j,s} \neq b_{j,s'}\} \quad (39)$$

where $p(\ell_{j,s}, \ell_{k,s'}) > \tau_{occ}$ (Eqn.12).

(c) $E^+_{sft,3}$ is the set of soft positive edges between one template-based candidate and one adjacent but not inside its bounding box appearance-based candidate whose quadrant co-occurrence probability is greater than the threshold τ_{occ} (from the bounding box index $b_{j,s}$ of the template-based candidate we obtain the quadrant number o of the appearance-based candidate),

$$E^+_{sft,3} = \{< O_{j,s}, R_{k,g} >;\ \exists\ e^{(adj)}_{j,k} \in E_{adj}\} \quad (40)$$

where $p(\ell_{k,g}|\ell_{j,s}, Q_o) > \tau_{occ}$ (Eqn.13).

The edge probability is,

$$q(e \in E^+_{sft}) \quad (41)$$
$$= \begin{cases} p(\ell_{j,g}, \ell_{k,g'}), \text{ if } e =< R_{j,g}, R_{k,g'} >\in E^+_{sft,1} \\ p(\ell_{j,s}, \ell_{k,s'}), \text{ if } e =< O_{j,s}, O_{k,s'} >\in E^+_{sft,2} \\ p(\ell_{k,g}|\ell_{j,s}, Q_o), \text{ if } e =< O_{j,s}, R_{k,g} >\in E^+_{sft,3} \end{cases}$$

which account for the cooperative contextual relations.

(ii) **The negative edge set** ($E^- = E^-_{hrd} \cup E^-_{sft}$).

E^-_{hrd} is a set of hard negative edges (represented by zigzag line segments, see Fig.6) between any two candidates from the same atomic region,

$$E^-_{hrd} = \{ < O_{j,s}, O_{j,s'} >, < O_{j,s}, R_{j,h} >, \quad (42)$$
$$< R_{j,h}, R_{j,h'} >;\ 1 \leq j \leq M\}$$

The edge probability is,

$$q(e \in E^-_{hrd}) = 1 \quad (43)$$

which means that only one candidate for a certain atomic region can turn "on" at each time.

$E^-_{sft} = E^-_{sft,1} \cup E^-_{sft,2} \cup E^-_{sft,3}$ is a set of soft negative edges (represented by dashed line segments, see Fig.6) complementary to the soft positive edge set E^+_{sft} in terms of the threshold τ_{occ}. We have,

$$E^-_{sft,1} = \{< R_{j,g}, R_{k,g'} >;\ \exists e^{(adj)}_{j,k} \in E_{adj}\} \quad (44)$$

where $p(\ell_{j,g}, \ell_{k,g'}) \leq \tau_{occ}$.

$$E^-_{sft,2} = \{< O_{j,s}, O_{k,s'} >; \exists e^{(adj)}_{j,k},\ b_{j,s} \neq b_{j,s'}\} \quad (45)$$

where $p(\ell_{j,s}, \ell_{k,s'}) \leq \tau_{occ}$.

$$E^-_{sft,3} = \{< O_{j,s}, R_{k,g} >;\ \exists e^{(adj)}_{j,k} \in E_{adj}\} \quad (46)$$

where $p(\ell_{k,g}|\ell_{j,s}, Q_o) \leq \tau_{occ}$.

And, the edge probability is,

$$q(e \in E^-_{sft}) \quad (47)$$
$$= \begin{cases} 1 - p(\ell_{j,g}, \ell_{k,g'}), \text{ if } e =< R_{j,g}, R_{k,g'} >\in E^-_{sft,1} \\ 1 - p(\ell_{j,s}, \ell_{k,s'}), \text{ if } e =< O_{j,s}, O_{k,s'} >\in E^-_{sft,2} \\ 1 - p(\ell_{k,g}|\ell_{j,s}, Q_o), \text{ if } e =< O_{j,s}, R_{k,g} >\in E^-_{sft,3} \end{cases}$$

which account for the competitive contextual constraints.

4.3 Clustering candidates by sampling edge probabilities $q(e)$

The candidacy graph summarizes data-driven information about the solution space Ω (i.e. the partition space and the model space). Intuitively, according to positive edges (i.e. cooperative relations), some local coherent solutions can be generated (e.g., for some adjacent atomic regions in a local range), which correspond to connected components (CCP's) of candidates in the candidacy graph (see the three CCP's shown in Fig.6). At the same time, based on negative edges (i.e. competitive relations), Some CCP's conflict each other (e.g., the CCP's compete for the interpretation of the same image domain or some largely overlapping image domains). These competitive CCPs are represented by composite CCPs (CCCP's, see one example in Fig.6). Our clustering sampling algorithm design a set of reversible jumps on CCCP's to traverse the solution space effectively.

In the candidacy graph, at a current state, we generate CCP's and CCCP's in two steps: (i) *Deterministic cuts.* We remove all the positive edges which link two nodes with different "on"/ "off" states, and all the negative edges which link two nodes with same "on"/ "off" states. (ii) *Probabilistically cuts.* By sampling the remaining positive edge probabilities, we divide candidates into a set of CCP's, and then by sampling the remaining negative edge probabilities, we obtain a set CCCP's. Formally, we have,

(i) On each positive edge $e \in E^+$ with the current states of the two linked node being $f_{j,c}$ and $f_{k,d}$ (see Eqn.34), $\mu_e = 1$ (i.e. "on") follows a Bernoulli probability,

$$\mu_e \sim \text{Bern}(q(e)\mathbf{1}(f_{j,c} = f_{k,d})),\ e \in E^+ \quad (48)$$

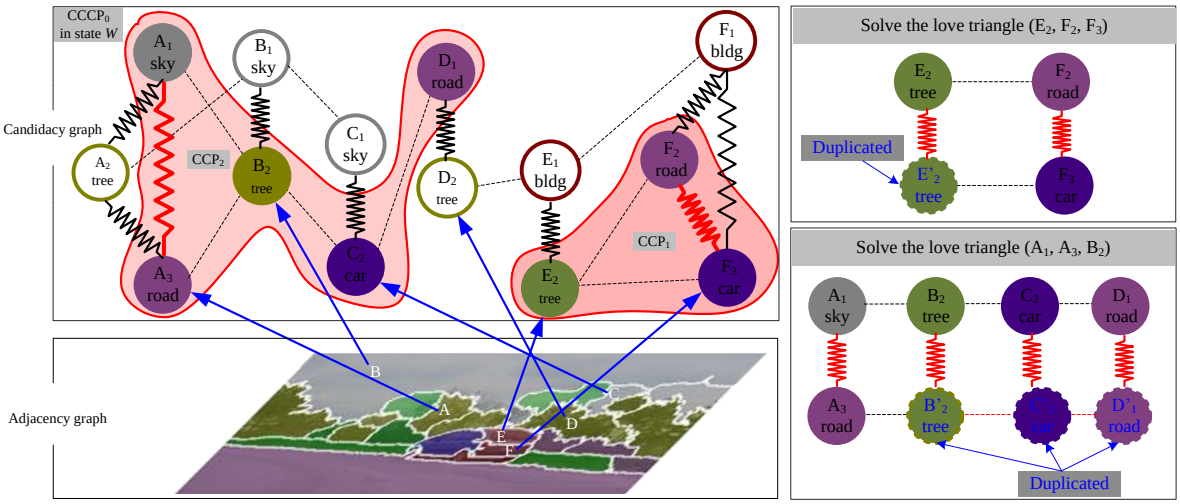


Fig. 7 Illustration of the love triangle in candidacy graph and solving the love triangle by duplication. (Best viewed in color)

Let $E^+_{on} = \{e; e \in E^+, \mu_e = 1\}$ be the set of "on" positive edges.

(ii) On each negative edge $e \in E^-$ (also suppose $f_{j,c}$ and $f_{k,d}$ are the current states of the two linked nodes), $\mu_e = 1$ follows a Bernoulli probability,

$$\mu_e \sim \text{Bern}(q(e)\mathbf{1}(f_{j,c} \neq f_{,d})), \ e \in E^- \tag{49}$$

Let $E^-_{on} = \{e; e \in E^-, \mu_e = 1\}$ be the set of "on" negative edges.

Definition 1: (CCP) In the candidacy graph, a CCP is defined as a set of nodes which are reachable each other by positive edges in E^+_{on}. All candidates in a CCP will be turned "on" or "off" together.

Definition 2: (CCCP) A CCCP is defined as a set of CCP's which are reachable each other by negative edges in E^-_{on} (an isolated CCP is also treated as a CCCP). All CCP's in a CCCP with negative edges should have different "on"/ "off" states.

Due to the randomness in sampling edge probabilities, we may obtain some invalid CCP's, so called "love triangles" (see two examples in Fig.7) in the candidacy graph (Porway and Zhu, 2010), to be resolved.

4.4 Resolving the "love triangles"

As Fig.7 illustrates, for a selected CCCP_0 in current state W, we want to assign new valid state configuration for all CCP's. Consider the two CCP's (CCP_1 and CCP_2), the two negative edges ($< F_2, F_3 >$ and $< A_1, A_3 >$ shown in red) have been cut in state W, but now we want to assign new states for CCP_1 and CCP_2 (e.g., turned "on"), and we need to take into account the two negative edges. A "love triangle" occurs if for a given three nodes (e.g. (E_2, F_2, F_3) in the left-top panel which is an invalid CCP), there are one negative edge $< F_2, F_3 >$ (which requires the "on"/ "off" states $f_{F_2} \neq f_{F_3}$) and two positive edges $< E_2, F_2 >$ and $< E_2, F_3 >$ (which require $f_{F_2} = f_{F_3} = f_{A_3}$ and lead to the conflict).

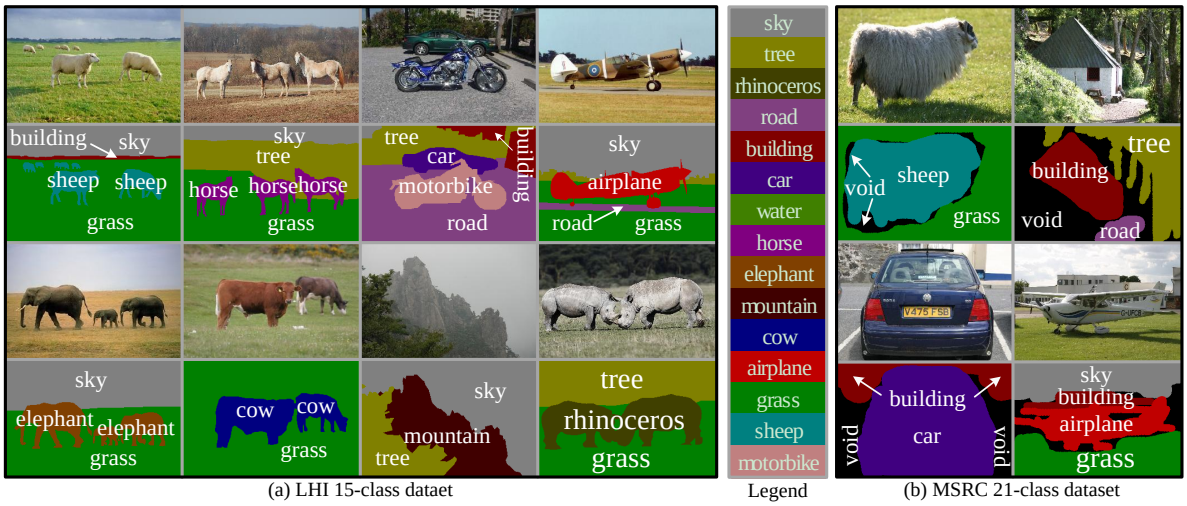
As proposed in (Porway and Zhu, 2010), a "love triangle" can be resolved by duplicating the node with two positive edges (e.g., E_2) into two copies (E_2 and $E_2^{'}$, illustrated in the right-top panel), and then adding a negative edge between the two copied nodes (e.g. $< E_2, E_2^{'} >$). So, we convert an invalid CCP into a CCCP.

Furthermore, for the "love triangle" (A_1, A_3, B_2), as illustrated in the right-bottom panel in Fig.7, we need check and resolve the "love triangles" iteratively. For example, after adding $B_2^{'}$, we get a new "love triangle" $(B_2, B_2^{'}, C_2)$ to be resolved and so on.

4.5 The cluster sampling algorithm

Our cluster sampling algorithm on the candidacy graph is in the same paradigm of the C^4 algorithm (Porway and Zhu, 2010). In each iteration, the candidacy graph is divided into a set of CCP's by sampling soft positive edges and then forms a set of CCCP's by sampling negative edges. A CCCP is selected probabilistically and the labels of its CCP's are reassigned such that all internal negative edges (i.e. competitive relations) are satisfied. The new state will be accepted or rejected based on the Metropolis-Hastings mehod.

The algorithm starts by randomly initializing the state $f_{j,c}$ for each node in the candidacy graph while we assign the same state (randomly "on" or "off") to the template-based candidates which share a same bounding box. We first summarize our clustering sampling



(a) LHI 15-class dataet Legend (b) MSRC 21-class dataset

Fig. 8 Some example images and the corresponding ground truth annotations in the LHI 15-class dataset (the left panel) and MSRC 21-class dataset (the right panel). Note that there are pixels indicated as "void" class in the MSRC dataset.

Algorithm 1: The cluster sampling algorithm

Input: An input image I_Λ, a adjacency graph $G_{adj} = (A, E_{adj})$, a candidacy graph $G_{cand} = (\mathcal{C}, E)$, the edge probabilities $q(e \in E)$ and posterior probability $p(W|I_\Lambda)$

Output: optimal solution $W^* = \arg\max p(W|I)$

1 Initializing the states $(f_{j,c})$ of nodes in $\mathcal{C}$.

2 **repeat**

3 Denote the current state as W_A.

4 **Step I**: Divide the candidacy graph into a set of CCP's denoted by $\{CCP\}$ and then form a set of CCCP's denoted by $\{CCCP\}$ (see Sec.4.3).

5 **Step II**: Selecting a CCCP (denoted by $cccp_0$ and denote by $l(cccp_0) = L_A$ the labels of $cccp_0$ under W_A) from $\{CCCP\}$ based on the probability $q(cccp_0|W_A)$ and then assigning new valid states denoted by L_B to the CCP's in $cccp_0$ and we have $l(cccp_0) = L_B$. Denote the new state as W_B.

6 **Step III**: Calculating the acceptance probability $\alpha(W_A \to W_B) = \min(1, \frac{q(W_B \to W_A)}{q(W_A \to W_B)} \cdot \frac{p(W_B|I_\Lambda)}{p(W_A|I_\Lambda)})$

7 **until** *the posterior probability does not increase any more during a predefined number of iterations*;

algorithm in the Algorithm.1 and then elaborate the details below.

The detailed balance equation. For each move between two different states W_A and W_B, the acceptance probability in our algorithm is based on the Metropolis-Hasting design,

$$\alpha(W_A \to W_B) = \min(1, \frac{q(W_B \to W_A)}{q(W_A \to W_B)} \cdot \frac{p(W_B|I_\Lambda)}{p(W_A|I_\Lambda)}) \quad (50)$$

where $q(W_A \to W_B)$ is the probability for proposing the state W_B from the state W_A to be designed to ensure the detailed balance.

Provided the proposal probability $q(W_A \to W_B)$, let $K(W_A \to W_B) = q(W_A \to W_B)\alpha(W_A \to W_B)$ be the Markov chain kernel and then we can see that in our cluster sampling algorithm the detailed balance equation is observed,

$$p(W_A|I_\Lambda)K(W_A \to W_B) = p(W_B|I_\Lambda)K(W_B \to W_A)$$

The proposal probability. To design the proposal probability, we adopt the same idea proposed in the C^4 algorithm (Porway and Zhu, 2010) and the SWC algorithm (Barbu and Zhu, 2005) and we have,

$$\frac{q(W_B \to W_A)}{q(W_A \to W_B)} = \frac{q(cccp_0|W_B)}{q(cccp_0|W_A)} \cdot \frac{q(l(cccp_0) = L_A|cccp_0, W_B)}{q(l(cccp_0) = L_B|cccp_0, W_A)} \quad (51)$$

$q(cccp_0|W_B)$ and $q(cccp_0|W_A)$ are the probabilities for choosing $cccp_0$ at states W_B and W_A respectively, which depend on two aspects: (a) the probabilities of generating $cccp_0$ under the two states W_A and W_B by sampling the edge probabilities $q(e)$'s (addressed in Sec.4.3), and then (b) the probabilities of select $cccp_0$ from the set of CCCP's $\{CCCP\}$ under the two states W_A and W_B, which are based on the weights $p_{j,c}$ of candidates in the $cccp_0$ to take into account the data-driven proposals. It turns out that,

$$\frac{q(cccp_0|W_B)}{q(cccp_0|W_A)} = \frac{\prod_{e \in Cut(cccp_0|W_B)}(1 - q(e))}{\prod_{e \in Cut(cccp_0|W_A)}(1 - q(e))} \quad (52)$$

where $Cut(cccp_0|W_A)$ is the cut of $cccp_0$ under the state W_A and is defined as the set of all negative (or positive) edges connecting nodes in $cccp_0$ and their neighbouring nodes with different labels (or same labels). More details are referred to (Porway and Zhu, 2010) and (Barbu and Zhu, 2005).

Table 2 Overall pixel-wise accuracy on the LHI 15-class and MSRC 21-class datasets using 5-fold cross validations.

Methods	LHI 15-class	MSRC 21-class
Our DD cluster sampling	**84.44%** $\pm$ **4.03%**	**79.5%** $\pm$ **4.27%**
HIM (L.Zhu et al, 2008)	-	81.2%
Auto-Context (Tu and Bai, 2010)	-	77.7%
CRF + Rel.Loc. (Gould et al, 2008)	-	76.5%
DecomposingScene (Gould et al, 2009a)	71.08% $\pm$ 3.22%	76.4%
Bag of Keypoints (Yang et al, 2007)	68.73% $\pm$ 4.56%	75.1%
Associative H-CRFs (Ladicky et al, 2009)	66.51% $\pm$ 4.27%	74.6%
TextonBoost+MS	64.32% $\pm$ 3.64%	73.5%
TextonBoost (Shotton et al, 2009)	62.70% $\pm$ 3.15%	72.2%

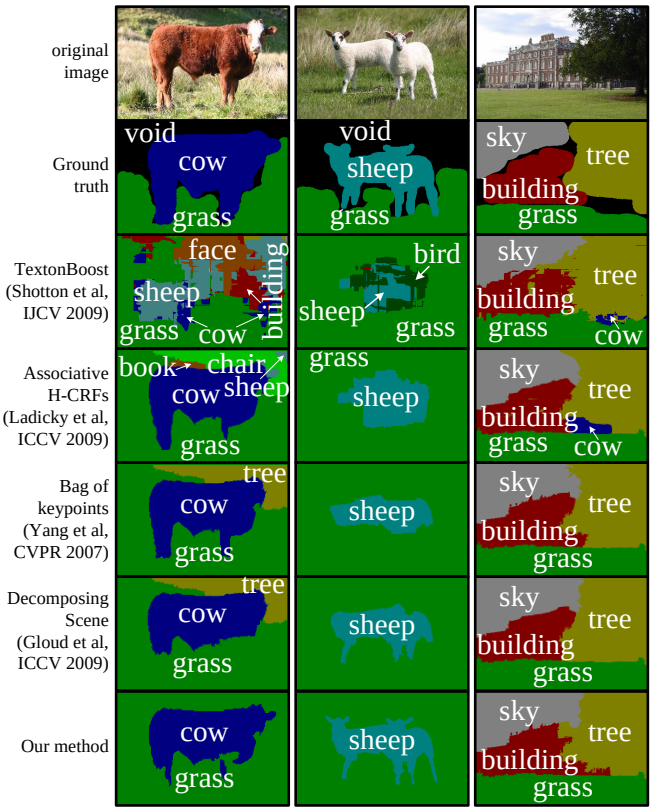


Fig. 9 Result comparisons of different methods. (Best viewed in color)

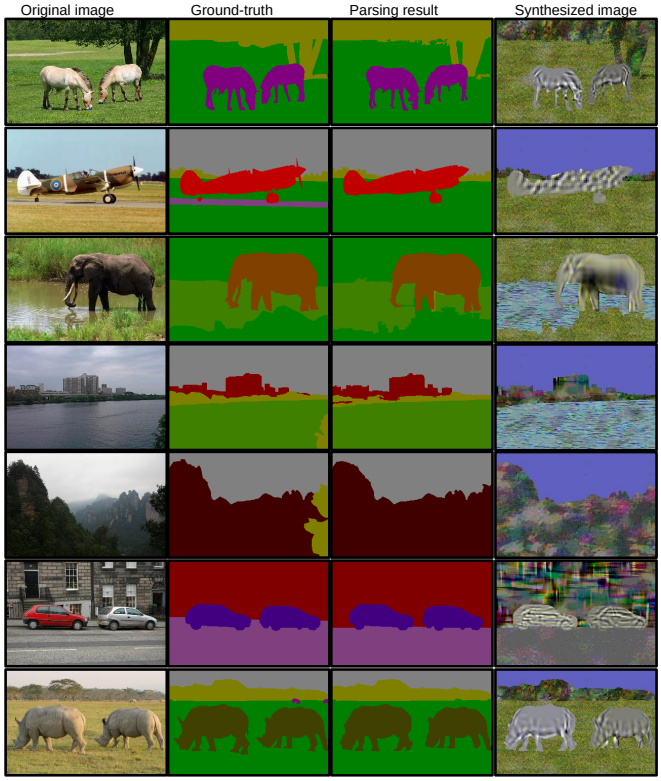


Fig. 10 Some synthesized images based on our parsing results. See texts for details. (Best viewed in color)

5 Experiments

5.1 The datasets

The LHI 15-class dataset[1] consists of 375 images including 7 types of generic regions (building, grass, tree, sky, road, water and mountain) and 8 object categories (airplane, cow, horse, sheep, car, elephant, rhinoceros and motorbike). The MSRC 21-class dataset (Shotton et al, 2009) consists of 591 images including 7 types of generic regions (building, grass, tree, sky, water, book and road) and 14 object categories (cow, sheep, aeroplane, face, car, bike, bird, flower, sign, chair, cat, dog, body and boat, and we did not learn the active basis models for cat, dog and body due to their extreme large variations in the dataset). Fig. 8 shows some example images, where we can see that the LHI 15-class dataset has finer annotations and the MSRC 21-class dataset include the "void" pixels near the boundary. In addition, in the MSRC 21-class dataset, some objects nearly occupy 70% area of the image in the center in most of the collected images, which leave too much information to the contextual models, especially the preferred location prior (Gould et al, 2008).

In our experiments, the LHI 15-class dataset is randomly split into roughly 40% for training, 10% for eval-

[1] www.imageparsing.com/LHI_SceneParsing15Classes/index.html

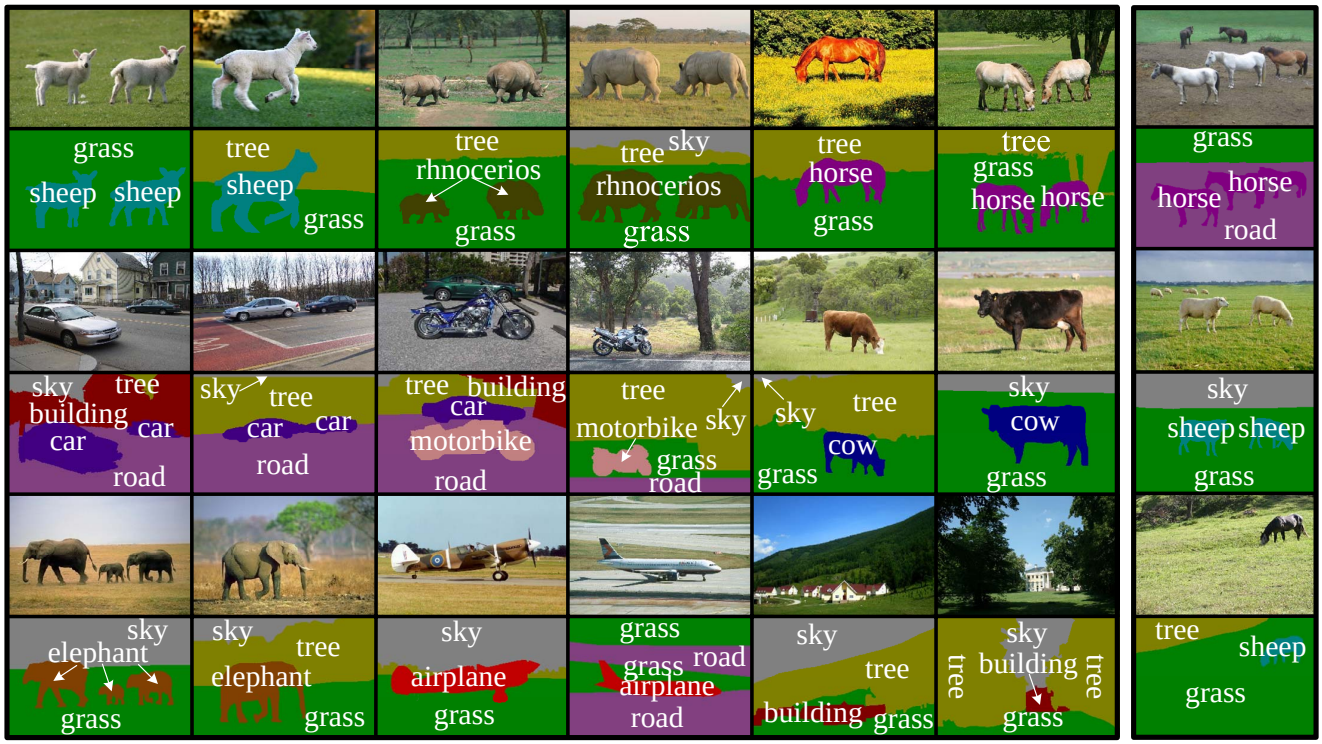


Fig. 11 Some parsing results on the LHI 15-class dataset.

	Bu	Gs	Tr	Sk	Mt	Wt	Cr	Rd	Cw	Sp	Hr	Rn	Pl	Mb	Et
building	**0.85678**	0.015275	0.045349	0.025763	0.00064011	0.00010704	0.020551	0.030512	0.00037055	3.91E-06			0.0037994	0.00085099	
grass	0.004031	**0.92659**	0.025728	0.0027853	0.0016919	0.00062109	0.00071657	0.021715	0.00186	0.0021823	0.0035808	0.0011303	0.0022423	0.0032119	0.0019153
tree	0.065897	0.05252	**0.7855**	0.057189	0.007801	0.0010428	0.0060764	0.0099507	0.00046618	6.16E-05	0.0019858	0.00022186	0.0038697	0.006203	0.001211
sky	0.012853	0.003258	0.015646	**0.96628**	0.0010623	3.88E-05		0.00017632	5.85E-06	1.46E-06	1.39E-05		0.00020925	0.00044886	6.58E-06
mountain	0.0039808	0.083406	0.013363	0.035647	**0.8388**	0.0057379	9.76E-06	0.012259			0.0038227	1.95E-06	0.00048223	0.0024931	
water	0.023443	0.03435	0.067292	0.0956	0.0026731	**0.6652**	1.20E-05	0.076369	0.0086643		0.0087603			0.010623	0.0070142
car	0.042325	0.0052329	0.029666	9.04E-05		0.0050256	**0.88188**	0.029026						0.0067507	
road	0.0089967	0.022284	0.0093855	0.00019735	0.00039443	0.001013	0.0109	**0.93528**	0.00060661	0.00020317	0.00041403		0.002103	0.0080772	0.00014675
cow	0.0048688	0.0931	0.014415	0.015582	0.0065357	0.0041174		0.010275	**0.80056**	0.0501	0.0035552				
sheep	0.001888	0.04656	0.014407	0.017743	0.0080354			0.011293	0.04692	**0.85112**	0.0088268				
horse	0.0083735	0.087718	0.043118	0.016397	0.00010687	0.0043649		0.011122	0.0053036	0.011432	**0.81349**				
rhinoceros		0.13126	0.0051173	0.00082611	0.021594							**0.8412**			
airplane	0.015147	0.067638	0.0020364	0.0041624	0.00011399	0.00012276		0.049505					**0.86064**		
motorbike	0.045	0.040861	0.039181	0.012117	0.001883	0.006148	0.0063131	0.034353						**0.81414**	
elephant		0.091936	0.048045	0.0065974		0.0068686		0.015417							**0.83113**

Fig. 12 Confusion matrix of our data-driven cluster sampling algorithm evaluated on the LHI 15-class dataset. The overall pixel-wise accuracy is 84.44%.

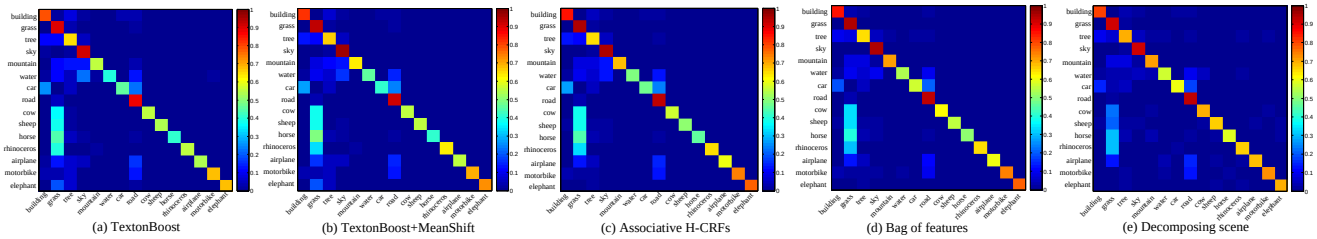
uation and 50% for testing, and the MSRC 21-class dataset uses the same split setting in (Shotton et al, 2009). We conduct 5-fold cross validations to generate the quantitative results in the experiments.

5.2 Overall results

Table.2 shows the overall pixel-wise accuracy results on the LHI 15-class and MSRC 21-class datasets by using our data-driven cluster sampling algorithm and the state-of-the-art methods in the literature. On the LHI 15-class dataset, our algorithm outperforms the state-of-the-art methods. On the MSRC 21-class dataset, our algorithm obtains comparable results with the state-of-the-art method. For visual comparisons, Fig.9 shows the results of different methods for three example images in the MSRC 21-class dataset.

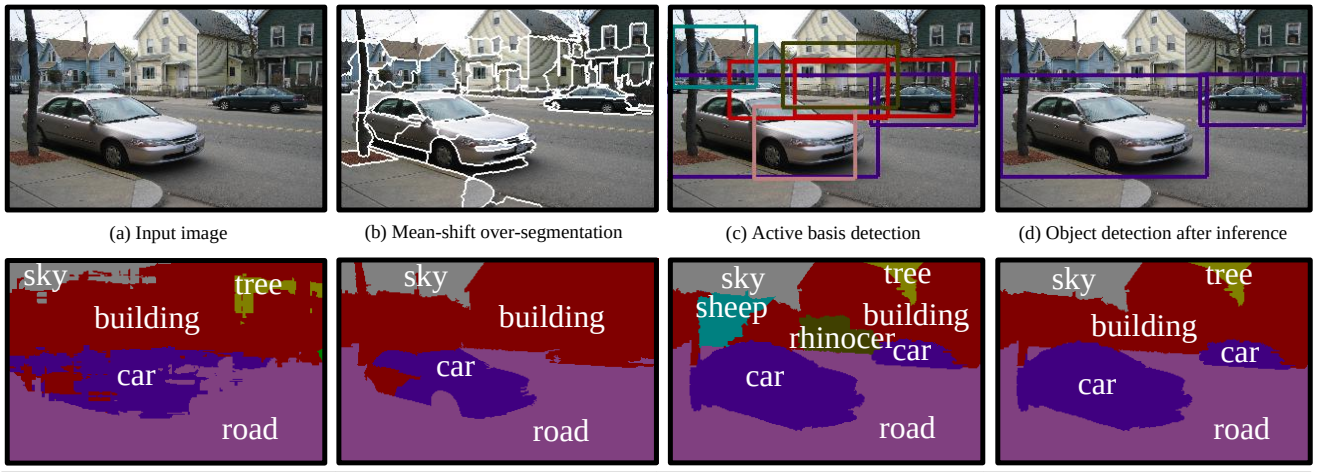
5.3 Results of synthesized images

Because we adopt generative models for both objects and generic regions, we can synthesize the parsing results. For object instances explained by the active basis models, the syntheses are based on the matching pursuit of the selected basis prototypes (i.e. Gabor wavelet elements) and some difference of Gaussian filters (DoG)



(a) TextonBoost (b) TextonBoost+MeanShift (c) Associative H-CRFs (d) Bag of features (e) Decomposing scene

Fig. 13 Comparisons of the confusion matrices. From (a) to (e), the confusion matrix is evaluated by using Textonboost (Shotton et al, 2009), Textonboost+Mean Shift, Associative CRF (Ladicky et al, 2009), Bag of features (Yang et al, 2007), Decomposing scenes (Gould et al, 2009a) respectively.



(a) Input image (b) Mean-shift over-segmentation (c) Active basis detection (d) Object detection after inference

	(e) Textonboost	(f) adding generic regions models	(g) adding active basis models	(h) adding contextual models
	Parsing results and accuracy of our framework by adding different terms of generative image models and contextual models sequentially			
Overall accuracy on LHI 15-calss	62.7%	76.21%	77.62%	84.44%

Fig. 14 Illustration of the effects of different terms in our generative model (see texts in Sec.5.5.1 for details).

as it was done in (Wu et al, 2010), and the inside appearances of the objects in synthesized results are not good enough (which entail the HiT(Si et al, 2009) models for object categories in the future work). For generic regions and object instances explained by appearance models, we use the algorithm "Sampling the Julesz Ensemble" proposed in (Zhu et al, 2000) based on the learned histogram prototypes (which defined the Julesz ensemble based on Eqn.7). Fig.10 shows some examples of synthesized images for our scene parsing results. Because we only learn one appearance model for each generic region category and object category, the corresponding synthesized parts in the synthesize images looks similar. Due to the fact that we used only one appearance model for generic regions, the synthesized results do not look good for categories such as buildings in our experiments.

5.4 Detailed results on the LHI 15-class dataset

Fig. 11 shows some experimental results for images in the LHI 15-class dataset. Our algorithm can handle large variations of view-point and scale of objects(e.g., see "cars" and "elephants") and large variations in the appearance of generic regions (e.g., see "building" and "road", and "tree" and "grass" which have the very similar appearance). On the last column, we show 3 examples in which the labeling are not good enough. In the first two images, some instances of "horse" and "sheep" are missed and merged to "grass". Due to the their extreme low resolutions, the active basis models didn't detect the object instances in bottom-up. In the last image, an instance of "horse" is parsed as "sheep" due to their intra-class similarities on both shape and appearance.

Fig. 12 shows the confusion matrix on the LHI 15-class dataset, in which accuracy values are computed as the percentage of image pixels assigned to the correct class label. The overall pixel-wise labeling accuracy is 84.44%. The average pixel-wise labeling accuracy of objects is 83.7%. From the confusion matrix, we can see that for generic regions, "sky" and "water" exhibit large confusions (about 10% of "water" pixels are labeled as "sky") due to their intra-class appear-

Fig. 15 Illustration of the improvement on object detections by using our framework. In each panel, the first row shows the detection results by only using the active basis model (Wu et al, 2010) (bounding boxes in different colors represent different categories) the second row show results after inference. The improvement mainly come from the advantage the our framework can integrate generative image models and the contextual models for inference.

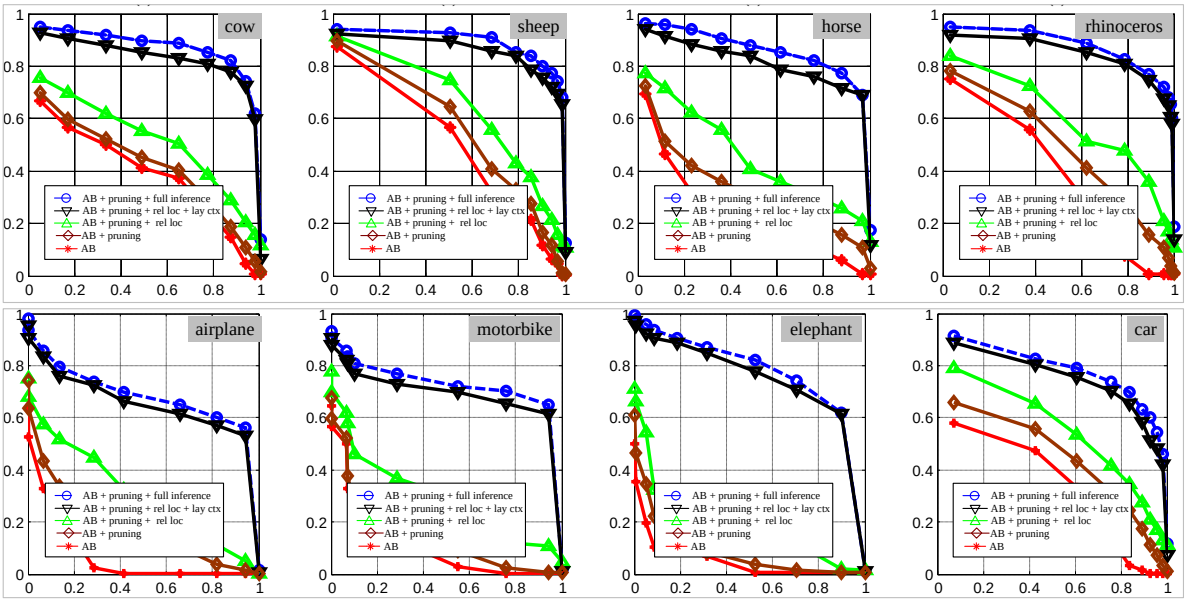


Fig. 16 Precision-Recall (PR) curves of the object detections for the 8 structured objects in the LHI 15-class dataset by using different aspects of our methods.

ance similarities, and for objects, "cow", "sheep" and "horse" exhibit relatively large confusions due to their intra-class shape similarities. For comparisons, Fig.13 also shows the confusion matrices evaluated on the LHI

15-class dataset by using some state-of-the-art methods in the literature.

5.5 Analysis of our cluster sampling algorithm

5.5.1 The contributions of different components

Fig. 14 illustrates the contributions of different components. (c) shows the object detections of the data-driven component by using active basis models (bounding boxes in different colors represent different categories) and (d) shows the object detection results after inference using our cluster sampling algorithm. (e) shows the labeling results using bottom-up Textonboost (Shotton et al, 2009). (f) shows the parsing result of our algorithm by only incorporating models of generic regions. (g) shows the result by further including models of structured objects. (h) shows the final result by considering the contextual models. We can see the average pixel-wise accuracy (calculated on the whole testing dataset, not for the single image in this figure) increases with more aspects being added in our framework.

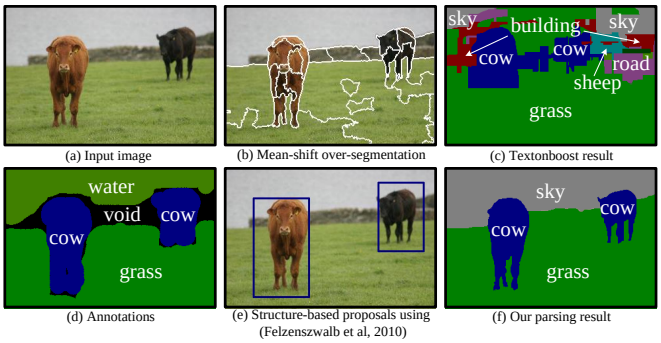
Furthermore, we evaluate the improvements on object detections over the pure active basis models. Fig. 15 shows the comparisons of object detections on the LHI 15-class dataset. For example, the active basis detection results of the right-most image in the top panel in Fig. 15 have some "car" instances, but the preferred location prior has very low probability allowing a "car" to appear around the right-top portion of the image lattice. In more details, Fig. 16 shows the Precision-Recall curves of object detections for the 8 structured objects in the LHI 15-class dataset by using different aspects of our framework (see the legends in the figure).

5.5.2 The effects of proposal generating models

In the data-driven component, we can replace the models by other methods to generate proposals. For example, instead of using the active basis model to generate template-based proposals, we can use the method proposed in (Felzenszwalb et al, 2010). Fig.17 shows a running example in which (e) shows the results of the template-based proposals. Our inference framework is flexible enough to still obtain good results as shown in (f) compared with the Textonboost result in (c).

5.5.3 The effects of thresholds in selecting candidates

As we mentioned in Sec.4.1, when selecting proposals as candidates for each atomic region, we use a set of thresholds estimated by using a validation dataset (see Table.1). We study the effects of the threshold pruning in terms of the accuracy and running time.



(a) Input image (b) Mean-shift over-segmentation (c) Textonboost result

(d) Annotations (e) Structure-based proposals using (Felzenszwalb et al, 2010) (f) Our parsing result

Fig. 17 Illustration of effects of proposal generating models. We generate template-based proposals by using the object detection methods proposed in (Felzenszwalb et al, 2010). The figure shows the parsing result of a testing image in MSRC 21-class dataset.

Table.3 shows the overall accuracy and the running time per image without and with the threshold pruning. Without pruning, there are often many false positives and false negatives incorporated into the candidacy graph and the size of candidacy graph increases largely (both the nodes and the edges). With pruning, we obtain better overall accuracy and much less running time per image.

Fig.16 shows PR curves evaluated on the 8 object categories in the LHI 15-class dataset. It shows the effects of the threshold pruning on the template-based proposals by using the active basis model and we obtain better results with pruning.

Fig. 18 shows PR curves evaluated on the LHI 15-class (all treated as generic regions). It shows the effects of the threshold pruning on the appearance-based proposals by using the Textonboost classifier and we also obtain better results with pruning.

5.5.4 The effects of granularity of over-segmentation and the number of iterations in sampling

Another two factors affecting the accuracy and running time are the granularity of over-segmentation (i.e. the number of atomic regions) and the number of iterations in sampling.

In our experiments, each testing image often has 30 to 40 atomic regions. The right panel in Fig.19 shows the plots of accuracy v.s. atomic regions and running time v.s. atomic regions based on 5-fold cross validations. We observe that after the number of atomic regions increases greater than 30, the accuracy is not improved much but the running time increases due to the increasing size of the candidacy graph.

For the number of iterations, empirically, we use 150 iterations in our experiments. The left panel in Fig.19

Table 3 The effects of the threshold pruning on the LHI 15-class dataset.

Method	Overall accuracy (%)	Inference time (min/per image)
With pruning	84.44	3
Without pruning	83.06	57

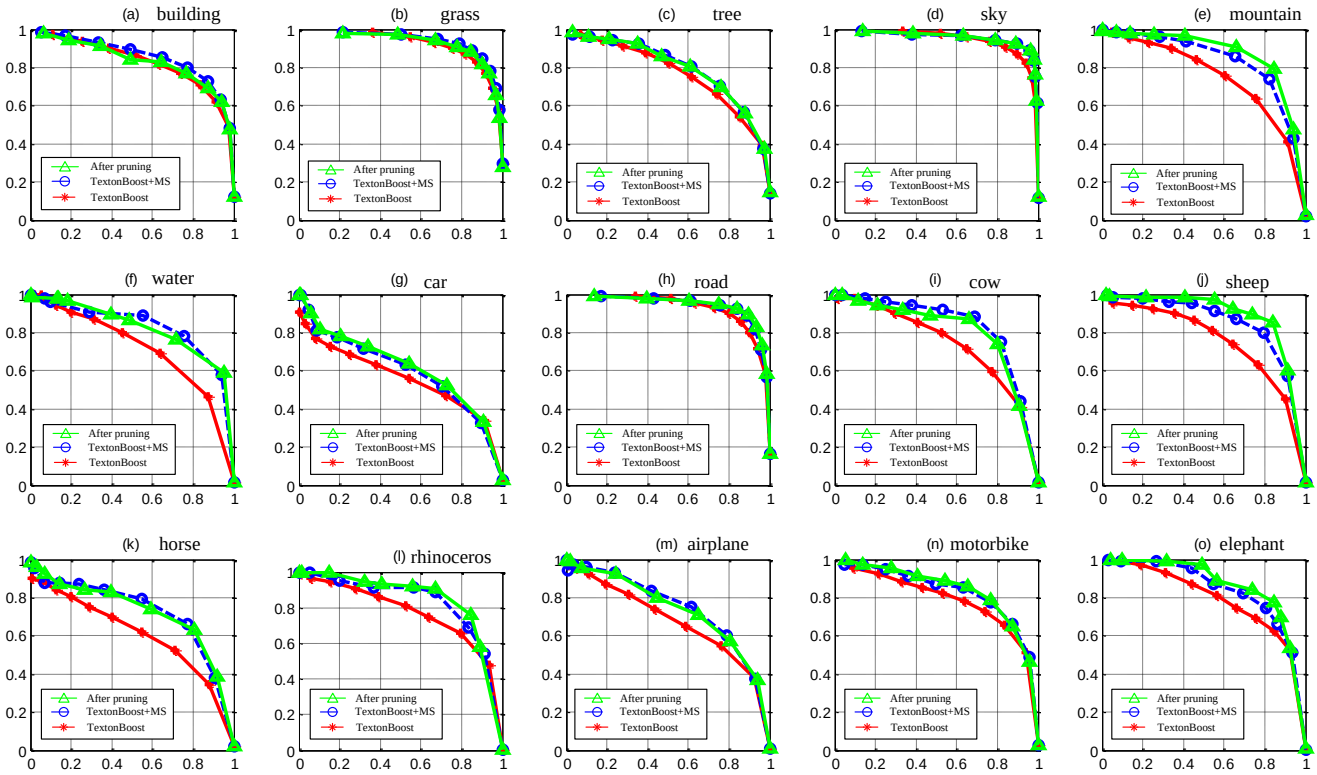


Fig. 18 PR curves before and after applying the thresholds to the Textonboost classifier proposals for the LHI 15-class (see the thresholds in Table.1).

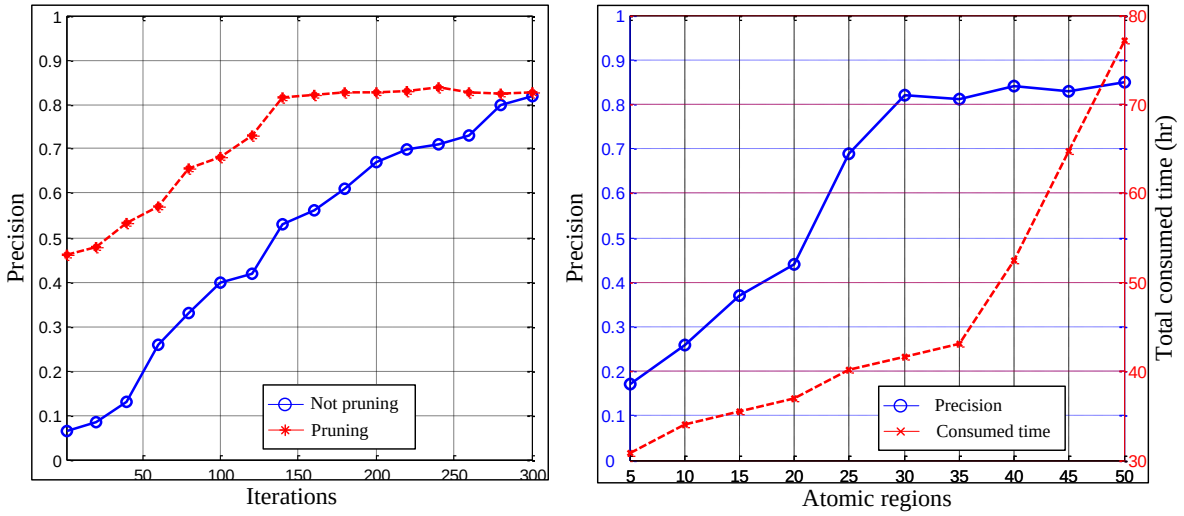


Fig. 19 Effects of the granularity of over-segmentation and the number of iterations in sampling. The plots are based on the results of the average of 5-fold cross validations.

shows the plot of accuracy v.s. iterations (with and without pruning). By pruning the proposal numbers, after 150 iterations, the accuracy is not improved much.

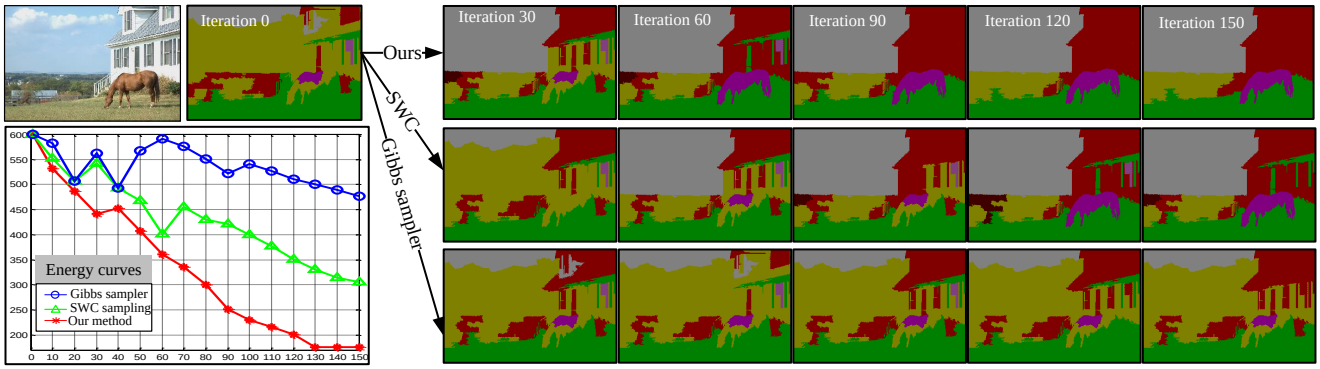


Fig. 20 Comparisons of convergence speed between our cluster sampling algorithm with the classic Gibbs sampler and the SWC algorithm (Barbu and Zhu, 2005).

Without pruning, we need about 300 iterations to get the comparable accuracy.

5.5.5 Comparisons of convergence speed

We also compare our cluster sampling algorithm with the classic Gibbs sampler (Geman and Geman, 1984) and the SWC algorithm (Barbu and Zhu, 2005). Fig. 20 shows the energy curves of the three methods in 150 iterations. From the energy curves, we can see that our algorithm can converge much faster than both Gibbs sampler and SWC due to the fact our algorithm takes advantage of both the positive edges and the negative edges and can swap large components at each iteration.

6 Summary

In this paper, we presented a data-driven cluster sampling framework for parsing scene images into objects and generic regions. Our framework extends the DDMCMC algorithm in three aspects: (i) We take into account the cooperative and competitive contextual relations in labeling and integrate them with the bottom-up proposals in a candidacy graph representation; (ii) We used the active basis models explicitly expressing object shapes; The two types of contextual relations and the active basis models for object shape improve the performance. (iii) We designed a cluster sampling algorithm based on the CCCP's in the candidacy graph to advance the speed of traversing the solution space. In experiments, our framework is tested on two datasets: the LHI 15-class dataset and the MSRC 21-class dataset. We evaluate scene parsing in terms of the pixel-level accuracy of the labeling and segmentation. Our algorithm outperforms the state-of-the-art methods on the LHI 15-class dataset and obtain comparable and competitive results on the MSRC 21-class dataset with the state-of-the-art methods (Shotton et al, 2009; L.Zhu et al, 2008). We also analyse different aspects of our framework in details.

In our on-going work, we are studying generative models for scenes (such as street scenes and open country scenes, classroom, and bedroom) in addition to the generative models for objects and generic regions. We are also trying to use hierarchical active basis models for objects in our algorithm to improve performance. Under a similar framework, we are also exploring the temporal context in video scene parsing.

Reproducibility

We have set up a project webpage (www.stat.ucla.edu/∼tfwu/project/SceneParsing.htm) where we release the LHI 15-class dataset for scene parsing used in this paper and the code for the experiments.

References

Barbu A, Zhu SC (2005) Generalizing swendsen-wang to sampling arbitrary posterior probabilities. PAMI 27(8):1239–1253

Borenstein E, Ullman S (2008) Combined top-down/bottom-up segmentation. PAMI 30(12):2109–2125

Boykov Y, Veksler O, Zabih R (2001) Fast approximate energy minimization via graph cuts. PAMI 23(11):1222–1239

Chang LB, Jin Y, Zhang W, Borenstein E, Geman S (2011) Context, computation, and optimal roc performance in hierarchical models. IJCV (to appear)

Choi MJ, Lim JJ, Torralba A, Willsky AS (2010) Exploiting hierarchical context on a large database of object categories. In: CVPR

Comaniciu D, Meer P (2002) Mean shift: A robust approach toward feature space analysis. PAMI 24(5):603–619

Corso JJ, Yuille AL, Tu ZW (2008) Graph-shifts: Natural image labeling by dynamic hierarchical computing. In: CVPR

Desai C, Ramanan D, Fowlkers C (2009) Discriminative models for multi-class object layout. In: ICCV

Felzenszwalb P, Girshick R, McAllester D, Ramanan D (2010) Object detection with discriminatively trained part based models. PAMI 32(9):1627–1645

Galleguillos C, Rabinovich A, Belongie S (2008) Object categorization using co-occurrence, location and appearance. In: CVPR

Geman S, Geman D (1984) Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. PAMI 6(6):721–741

Gould S, Rodgers J, Cohen D, Elidan G, Koller D (2008) Multi-class segmentation with relative location prior. IJCV 80(3):1239–1253

Gould S, Fulton R, Koller D (2009a) Decomposing a scene into geometric and semantically consistent regions. In: ICCV

Gould S, Gao T, Koller D (2009b) Region-based segmentation and object detection. In: NIPS

He XM, Zemel RS, Carreira-Perpinan MA (2004) Multiscale conditional random fields for image labeling. In: CVPR

He XM, Zemel RS, Ray D (2006) Learning and incorporating top-down cues in image segmentation. In: ECCV

Hoiem D, Efros AA, Hebert M (2005) Geometric context from a single image. In: ICCV

Hoiem D, Efros A, Hebert M (2008) Closing the loop in scene interpretation. In: CVPR

Jin Y, Geman S (2006) Context and hierarchy in a probabilistic image model. In: CVPR, pp 2145–2152

Kumar S, Hebert M (2005) A hierarchical field framework for unified context-based classification. In: ICCV

Kumar S, Hebert M (2006) Discriminative random fields. IJCV 68:179–201

Ladicky L, Russell C, Kohli P (2009) Associative hierarchical crfs for object class image segmentation. In: ICCV

Ladicky L, Russell C, Kohli P, Torr PHS (2010a) Graph cut based inference with co-occurrence statistics. In: ECCV(5), pp 239–253

Ladicky L, Sturgess P, Alahari K, CRussell, Torr P (2010b) What, where and how many? combining object detectors and crfs. In: ECCV

Lafferty J, McCallum A, Pereira F (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: ICML

Leibe B, Leonardis A, Schiele B (2004) Combined object categorization and segmentation with an implicit shape model. In: ECCV Workshop

Levin A, Weiss Y (2009) Learning to combine bottom-up and top-down segmentation. IJCV 81(1):105–118

LZhu, Chen YH, Lin Y, Lin CX, Yuille A (2008) Recursive segmentation and recognition templates for 2d parsing. In: NIPS

Pietra SD, Pietra VJD, Lafferty JD (1997) Inducing features of random fields. PAMI 19(4):380–393

Porway J, Zhu SC (2010) C4 : Computing multiple solutions in graphical models by cluster sampling. PAMI (to appear)

Rabinovich A, Vedaldi A, Wiewiora CGE, Belongie S (2007) Objects in context. In: ICCV

Shotton J, Winn JM, Rother C, Criminisi A (2009) Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. IJCV 81(1):2–23

Si ZZ, Gong HF, Wu YN, Zhu SC (2009) Learning mixed templates for object recognition. In: CVPR

Torralb A, Murphy KP, Freeman WT (2004) Contextual models for object detection using boosted random fields. In: NIPS

Tu Z, Bai X (2010) Auto-context and its application to high-level vision tasks and 3d brain image segmentation. PAMI 32(10):1744–1757

Tu ZW, Zhu SC (2002) Image segmentation by data-driven markov chain monte carlo. PAMI 24(5):657–673

Tu ZW, Chen X, Yuille AL, Zhu SC (2005) Image parsing: unifying segmentation, detection, and recognition. In: IJCV

Verbeek J, Triggs B (2007) Scene segmentation with conditional random fields learned from partially labeled images. In: NIPS

Wojek C, , Schiele B (2008) A dynamic conditional random field model for joint labeling of object and scene classes. In: ECCV

Wu YN, Si ZZ, Gong HF, Zhu SC (2010) Learning active basis model for object detection and recognition. IJCV 90(2):198–235

Yang L, Meer P, Foran D (2007) Multiple class segmentation using a unified framework over mean-shift patches. In: CVPR

Yang Y, Hallman S, Ramanan D, Fowlkes C (2010) Layered object detection for multi-class segmentation. In: CVPR

Zhu SC, Liu X, Wu YN (2000) Exploring texture ensembles by efficient markov chain monte carlo-toward a 'trichromacy' theory of texture. PAMI 22(6):554–569

Zhu SC, Shi K, Si Z (2010) Learning explicit and implicit visual manifolds by information projection. Pattern Recognition Letters 31(8):667–685