Inference Scene Labeling by Incorporating Object Detection with Explicit Shape Model

Quan Zhou and Wenyu Liu

Dept. of Electronics and Information Engineering,
Huazhong University of Science and Technology, Wuhan, China PR
qzhou.lhi@gmail.com, liuwy@hust.edu.cn

Abstract. In this paper, we incorporate shape detection into contextual scene labeling and make use of both shape, texture, and context information in a graphical representation. We propose a candidacy graph, whose vertices are two types of recognition candidates for either a superpixel or a window patch. The superpixel candidates are generated by a discriminative classifier with textural features as well as the window proposals by a learned deformable templates model in the bottom-up steps. The contextual and competitive interactions between graph vertices, in form of probabilistic connecting edges, are defined by two types of contextual metrics and the overlapping of their image domain, respectively. With this representation, a composite clustering sampling algorithm is proposed to fast search the optimal convergence globally using the Markov Chain Monte Carlo (MCMC). Our approach is applied on both lotus hill institute (LHI) and MSRC public datasets and achieves the state-of-art results.

1 Introduction

As Fig. 1 illustrates, this paper presents an semantic scene understanding (labeling) method, motivated by partitioning or segmenting an entire image in (a) into distinct recognizable regions in (b). This task requires classify all pixels, while preserving accurate segmentation. By generating recognition candidates by superpixel classification and object detection in the bottom-up steps as shown in (c) and (d) respectively, we present a candidacy graphical representation to integrate shape, texture, and context information.

We start by reviewing the literature on two research streams: semantic image segmentation and structural object detection that are related to the bottom-up steps of our work.

(i) Many approaches of image segmentation often explore textural appearance features, and use flat graphical representation to encode local confidence and pairwise consistency. Examples include the methods based on Markov random fields (MRFs) and the conditional random fields (CRFs). The former models the joint probability of the image and its corresponding semantic labels [1], and latter models the conditional probability of the labels [2, 3]. Recently, the global and contextual information based on the graphical representation are

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 382–395, 2011.

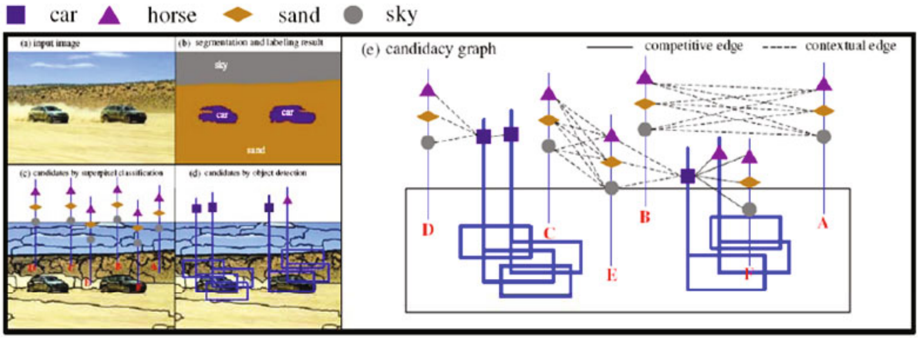

car
horse
sand
sky
(a) input image
(b) segmentation and labeling result
sky
car
car
sand
(c) candidates by superpixel classification
(d) candidates by object detection
(e) candidacy graph
competitive edge
contextual edge
D
C
E
B
F
A

Fig. 1. Illustration of the proposed method. Given an input images in (a), the recognition candidates of over-segmented superpixels (denoted by red letters), as well as the candidates of template-based detector (denoted by blue rectangles), are extracted in (c) and (d) respectively. A candidacy graphical representation is constructed with these candidates as illustrated in (e). The final labeling result is exhibited in (b).

explored by some innovative work [4, 5, 6, 7, 8]. The inference task on graphical models can be formulated as energy minimization problems with soft and hard constraints. The algorithms can be divided into deterministic approximation algorithms, such as the graph cuts [9] and the belief propagation (BP) [10], and stochastic algorithms, like constraint-satisfaction solvers [11], Gibbs sampler [12].

(ii) Some other methods are aimed at detect and localize object-of-interest from cluttered scene by capturing shape information. These methods usually represent structural objects by a spatial or context configuration with a small number of primitives, such as the PAS-based model [13], the shape context [14], and the recent proposed active basis model [15].

Though the research in these two streams has made remarkable progress, it still remains a challenge to combine the two types of method for the entire scene understanding due to the difficulty of integrating shape and texture model. A few pioneer work demonstrates this path with some special cases [17, 18].

In this paper, we study (i) a candidacy graphical representation that incorporates the shape information and textural appearance in a Bayesian framework (ii) a composite cluster sampling algorithm for energy convergence globally.

Given an input image, we first generate a batch of recognition candidates (proposals) by two types of classifiers: superpixel classifier and shape detector. The superpixel classifier is learned by JointBoost method with a bank of low-level features, inspired by [3], and it gives the possible labels to each superpixel. The active basis model [15] is employed as shape detector to learn deformable templates of structural object, and used to generate possible matchings on the testing image, as shown in Fig. 1 (d). We thus build up an adjacency candidacy graphical representation with these candidates, as illustrated in Fig. 1 (e) and Fig. 3, where each graph vertex is equivalent to a recognition candidate. Each two vertices can be linked by a probabilistic edge denoting the competitive or contextual interaction. Thus the semantic parsing can be solved by validating these candidates while accounting for the interactions among them.

With this representation, we present a composite cluster sampling algorithm using the Markov Chain Monte Carlo (MCMC) mechanism [19]. Unlike the traditional single-site sampler [11,12], this algorithm updates large portions of the solution space quickly to minimize constraint energy, by clustering connected components in each sampling step. It can be viewed as an extension of the multiple-site sampler [20] by dealing with the soft (contextual) and hard (competitive) constraints simultaneously. Given the candidacy graphical representation, this algorithm contains two iterative steps: (i) Sampling the competitive and contextual edges to form a composite cluster; (ii) Validating the graph vertices of this cluster following the Markov Chain Monte Carlo (MCMC) mechanism [19].

The remainder of this paper is arranged as follows. We first present the bottom-up proposal and candidacy representation in Sect. 2, and follow with a description of the problem formulation in Sect. 3. The inference algorithm is discussed in Sect. 4. The experimental results are shown in Sect. 5 and the paper concludes with a summary in Sect. 6 .

2 Representation

In this section, we first introduce the recognition candidate generation by two types of classifiers and then discuss a candidacy graphical by these candidates.

2.1 Bottom-Up Candidates Generation

Given an input image I, we first use two types of classifier to generate recognition candidates: one for superpixels with low-level (textural appearance) features and the other for structural objects with shape templates. A candidate is defined as a universal form $c_i = (A_i, l_i)$. $A_i = (X_i, \Gamma_i, \Lambda_i, \omega_i)$ denotes the candidate attributes, including location X_i, outer contour Γ_i and image domain Λ_i, respectively. $\omega_i \in \{0,1\}$ is the binary variable that denotes the validation or not of c_i. $l_i = ('car', 'grass', 'cow', \ldots)$ denotes the semantic label.

We start by discussing generation recognition candidates by these two classifiers.

(i) Superpixel-based candidate.

Superpixels are often used to effectively reduce the solution complexity in image segmentation. In this work, we use an over-segmentation scheme used in [24] to obtain the superpixels of both training and testing images. In practice, each image contains around $30 \sim 50$ superpixels.

Given the annotated training images, we collect a pool of textural features, including texton filters [3], color [22], and location [4], and then learn a discriminative classifier formed by a set of selected features using a boosting framework [23]. In the testing stage, each superpixel receives a recognition score for each semantic label by this classifier, and a batch of superpixel candidates are generated.

Thus the energy cost for each superpixel-based proposal can be computed by

$$E_T(c_i|I) = \sum_{j=1}^{n(T)} \alpha_j f_j(A_i, l_i), \tag{1}$$

where $f_j(A_i, l_i)$ is one selected feature (weak classifier) over superpixel and semantic label, and α_j is the weight parameter. $n(T)$ denotes the number of shape templates.

(ii) Template-based candidate.

A recent proposed active basis model [15] is utilized to capture shape information of structural object categories (car, cow, and horse, etc.). Using the model with a shared sketch algorithm, deformable templates can be learned for each object category on a small set of aligned positive samples in the same pose without negative samples. A template $\mathbf{B}_{l_i}$ for category l_i, consist of a set of active Gabor basis $\{B_j\}$ that are allowed to slightly perturb their locations and orientations before they are linearly combined to generate the image. One can use other object detection approaches [16] without major algorithmic changes.

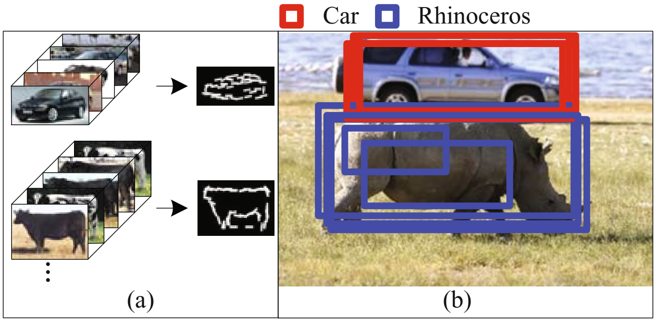


Fig. 2. Template-based recognition candidates generated by active basis model [15]. (a) illustrates the deformable template learning from a set of aligned positive samples and (b) shows an example of template detection.

In Fig. 2 (a), we intuitively illustrate the template learning process from several aligned training images, and Fig. 2 (b) shows an example of detecting the structural object instances by matching the templates in cluttered image, (red rectangle for car detection and blue rectangle for rhinoceros).

We solve the energy cost for each candidate by template matching [15], as

$$E_S(c_i|I) = \sum_{j=1}^{n(S)} [\lambda_j h(| < I(A_i), B_j > |^2) - \log Z(\lambda_j)], \tag{2}$$

where $h(\cdot)$ is transformation function on filter response. λ_j denotes the learned weight parameter for each basis B_j, and $Z(\lambda_j)$ is the normalizing constant and it can be computed using [15]. $n(S)$ denotes the number of superpixels.

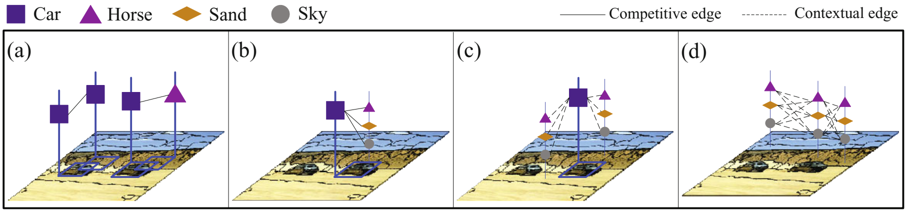


Fig. 3. Incorporating contextual and competitive interactions in the candidacy graphical representation. The blue rectangles indicate the detected object templates. The polygons on thick line denote the template-based candidates and polygons on thin line denote the superpixel-based candidates. The different polygons with different colors denote recognition label. The dashed line and solid line denote the contextual and competitive edge link respectively. The competitive edges exist between two candidates sharing image domain (as shown in (a) and (b)), and the contextual edges for connecting candidates defined by context features (as shown in (c) and (d)).

2.2 Candidacy Graph Construction

With these generated recognition candidates, we establish an adjacency graphical representation $G = (V, E)$, whose vertex v_i is equivalent to a candidate c_i. We thus define the graph vertex set as,

$$V = V_T \bigcup V_S = \{v_i = c_i = (A_i, l_i), i = 1, \dots n(T) + n(S)\}, \qquad (3)$$

where V_T and V_S are candidate sets from superpixel classifier and shape detector, respectively. In this graph, the parsing problem can be formulated as the candidate validating task.

For any two vertices v_i and v_j specified by two adjacent superpixels, a probabilistic edge $e =< v_i, v_j >, e \in E$ is defined to indicate the competitive or contextual interaction between them, this leads to $E = E^+ \cup E^-$. Each **contextual edge** E^+ exists between two vertices that share the contextual correlation, while each **competitive edge** E^- accounts for the mutual exclusion between two vertices that share overlapping in image domain. Fig. 3 illustrates a typical example of the candidacy graphical representation.

Competitive edges are defined for the mutual exclusion constraint that the two vertices should not both be validated if they overlap with each other in image domain. The overlapping often occurs with two neighboring template candidates or one template candidate including a superpixel proposal, as illustrated in Fig. 3 (a) and (b) respectively. The connecting probability ρ_e^- of the competitive edge is thus defined as

$$\rho_e^- = \begin{cases} \exp\{\dfrac{||\Lambda_i \bigcap \Lambda_j||}{||\Lambda_i \bigcup \Lambda_j||}\}, & v_i, v_j \in V_S, \Lambda_i \bigcap \Lambda_j \neq \emptyset \\ 0, & v_i, v_j \in V_S, \Lambda_i \bigcap \Lambda_j = \emptyset \\ 1, & \Lambda_i \subset \Lambda_j \;\; or \;\; \Lambda_j \subset \Lambda_i \end{cases} \qquad (4)$$

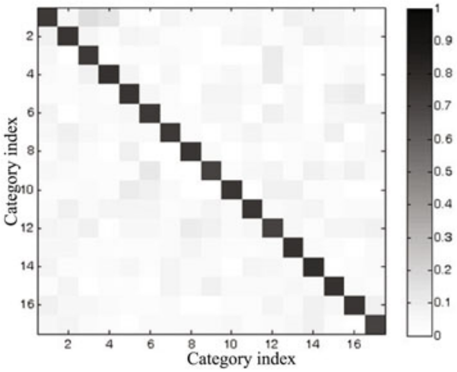


Fig. 4. Pairwise co-occurrence matrix for superpixel candidates

Contextual edges play a key role in our model, which imply contextual information between two graph vertices. We explore two types of contextual metrics: the co-occurrence between two superpixel candidates and the layout between a template candidate and an adjacent superpixel proposal.

- **Co-occurrence context**, constrains the two connected superpixel candidates according to a learned distribution of related object categories, as represented by a pairwise co-occurrence matrix $H^O(\cdot,\cdot)$ in Fig. 4. In this figure, the co-occurrence probabilities are scaled to gray-levels with the diagonals contributing zero energy. The darker gray-scale intensity denotes higher co-occurrence probability. For example, the "car" to "grass" pair has comparatively low probability because they seldom appear adjacently in our dataset. Conversely, the probability between "sky" and "mountain" is intuitively high as a result of their frequent co-occurrence in natural scene images.
- **Layout context**, constrains the relative location of a superpixel candidate, given a template candidate. For each structural category and surrounding superpixels in training set, we learn a 2D probability histogram $H^L_{l_i}(\cdot,\cdot)$ that encodes normalized pixel number with variational quadrant index and category index. Fig. 5 illustrates this layout context and the histogram.

Based on the definition of two context metrics, the probability of contextual edges ρ_e^+ is thus defined as,

$$\rho_e^+ = \begin{cases} H^O(l_i, l_j), & v_i, v_j \in V_T \\ \sum_{k=1}^{n(D)} \#(\Lambda_j \cap D_k) H^L_{l_i}(D_k, l_j), & v_i \in V_S \text{ and } v_j \in V_T \end{cases} \tag{5}$$

where D_k refers the pixel map of the j-th quadrant. $n(D)$ denotes the number of quadrant and we set it as 10. $H^L_{l_i}(\cdot,\cdot)$ is the layout context histogram with respect to template candidate v_i. Λ_j and l_j indicate image domain and semantic label of the superpixel candidate v_j respectively.

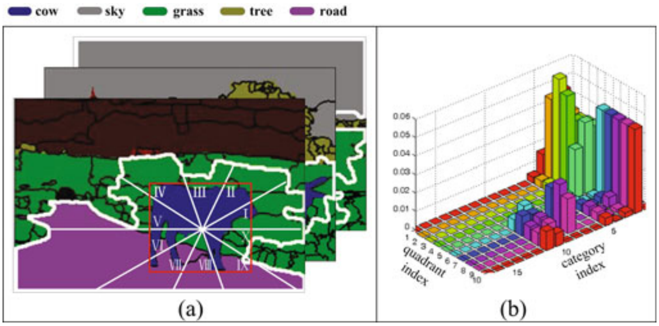


Fig. 5. Layout context probability histogram for the template candidates with surrounding superpixel candidates. (a) illustrates the definition of the layout context histogram. Given a structural object in training images, the annotated surrounding superpixels are projected to a number of quadrant (denoted by the Rome number). Thus the pixel number distribution over the quadrant index and category index can be calculated as shown in (b).

3 Probabilistic Formulation

Assume two sets of candidates are validated from the candidate set V: $Q = \{q_i\} \subset V_T, U = \{u_i\} \subset V_S$ with $\omega = 1$, and sizes of Q, U are N_T, N_S respectively. The solution configuration of parsing is defined as,

$$W = \left\{N_T, Q = \{q_i\}_{i=1}^{N_T}, N_S, U = \{u_i\}_{i=1}^{N_S}\right\}, \tag{6}$$

We further formulate the solution W in Bayesian framework and solve it by maximizing a posterior probability as

$$W^* = \arg\max p(W|I) = \arg\max p(W)p(I|W). \tag{7}$$

Prior term. We define the prior probability over the candidate numbers N_T, N_S and the validating set $\Psi = (Q \bigcup U)$, as,

$$\begin{aligned} P(W) = & P(N_T)P(N_S)P(Q,U) \\ & \propto \exp\{-\alpha_T N_T\} \exp\{-\alpha_S N_S\} \cdot P(Q,U), \end{aligned} \tag{8}$$

where α_T and α_S are tuning parameters and are set as 1 empirically. Since Q and U are two types of candidates sharing the same definition as graph vertices, we define

$$P(Q,U) = \prod_{e \in E^+} \exp\{\beta \mathbf{1}(\omega_i = \omega_j)\} \prod_{e \in E^-} \exp\{\beta \mathbf{1}(\omega_i \neq \omega_j)\}. \tag{9}$$

$\beta \in [0,1]$ is the tuning parameter and it set as 0.5 in practice. $\mathbf{1}(\cdot) \in \{0,1\}$ is an indicator function for a Boolean variable. The probability is maximized when all contextual edges have same vertices state and all competitive edges connect two vertices with differently state labels.

Likelihood term. Using the classifiers for the recognition candidate generation, we define the likelihood probability of our model with the validated proposals, as

$$P(I|W) = P(I|Q,U) \propto \prod_{q_i \in Q} \exp\{-E_T(q_i)\} \prod_{u_i \in U} \exp\{-E_S(u_i)\}, \tag{10}$$

where E_T and E_S are energy costs for each validated candidate given the two types of classifiers, as defined in Eq. 1 and Eq. 2.

4 Inference by Composite Cluster Sampling

Based on the candidacy graph $G =< V, E >$, our algorithm simulates a Markov chain that consist of a sequence of states in the solution space, and travels the space by realizing reversible jumps between any two successive states. For each stochastic jump step, whether a new state is accepted is decided by the Metropolis-Hastings [19] method that guarantees the global convergence of the inference algorithm. Given two successive states A and B, the acceptance rate is defined as,

$$\alpha(A \to B) = \{1, \frac{Q(B \to A)P(B)}{Q(A \to B)P(A)}\}, \tag{11}$$

where $P(A)$ and $P(B)$ are the posterior probability. $Q(B \to A)$ and $Q(A \to B)$ are canidae probability of "jumping" between two states. Following the theoretical analysis reported in [20], $Q(B \to A)/Q(A \to B)$ can be simplified by cluster sampling, which contains two steps: 1) forming a composite cluster, called connected component (CCP), by sampling the probabilistic edge connection; 2) flipping the generated CCP by re-validating graph vertices.

Forming a composite CCP in $G =< V, E >$ is equivalent to sampling the edge probability (defined in Eq. 4 and Eq. 5). For each probabilistic link $e =< v_i, v_j >$, we define the sampling protocol for edge sampling (cutting) as

- **Deterministic cut**, as illustrated by black "×" in Fig. 6, is performed (i) on contextual edges connecting two different state vertices, and (ii) on competitive edges connecting two same state vertices.
- **Probabilistic cut**, is illustrated by black "||" in Fig. 6. (i) The contextual edges connecting two same state vertices are turn off with probability $1 - \rho_e^+$, and (ii) the competitive edges connecting two different state vertices are turn off with probability $1 - \rho_e^-$.

Note we then select one CCP with equal probability if more than one is formed. Thus the generation of the composite cluster can be calculated by the probability of "turning off" the edges (as the black "||" and "×" denote in Fig. 6) around the composite cluster, as

$$Q(CCP) = \prod_{e \in E^+ \bigcap \mathcal{C}} (1 - \rho_e^+) \prod_{e \in E^- \bigcap \mathcal{C}} (1 - \rho_e^-), \tag{12}$$

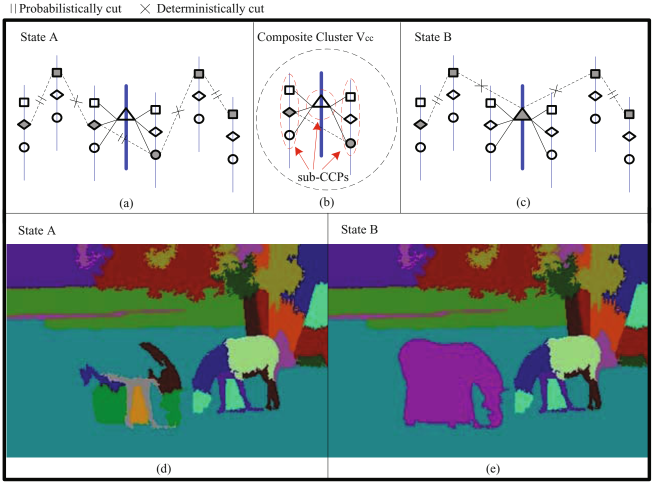


Fig. 6. The illustration of composite cluster sampling. (a) shows a current state and the edge links are cut deterministically or probabilistically denoting by black "×" or "‖"; (b) shows a composite cluster (in the black ellipse) including three conflicting connected components (CCPs) (in the dashed rectangles), and each one includes vertices connected by contextual (dashed) links while any neighboring CCPs are connected by a competitive (solid) link; (c) is the new state resulting from the re-validating in the composite cluster, and all vertices in a CCP are compatibly validated as a whole; (d) and (e) are the real segmentation solutions corresponding to states (a) and (c). Note the solid vertices imply validated candidates.

where $\mathcal{C}$ is a set of the edges that has been "turned off" around CCP. The edge probability ρ_e^+ and ρ_e^- are defined in Eq. 5 and Eq. 4 respectively. Note we then u.a.r select one CCP if more than one is formed.

Flipping the CCP is equivalent to re-validating vertices in the CCP. We split the selected CCP to many sub-CCPs, as showed in Fig. 6 (b), and then simply reverse the state of each vertex thus keeping the current constraints satisfied, since our candidacy representation is a typical Ising model where each site only has two states.

Thus $Q(B \to A)/Q(A \to B)$ only depends on the generation of CCP, and computed by

$$\frac{\prod_{e \in E^+ \bigcap \mathcal{C}_B}(1-\rho_e^+)\prod_{e \in E^- \bigcap \mathcal{C}_B}(1-\rho_e^-)}{\prod_{e \in E^+ \bigcap \mathcal{C}_A}(1-\rho_e^+)\prod_{e \in E^- \bigcap \mathcal{C}_A}(1-\rho_e^-)}. \tag{13}$$

A representative composite cluster CPP with three conflicting connected components (sub-CCPs) is shown in Fig. 6 (b), and all vertices in each sub-CCP should be compatibly validated with the other neighboring ones. Fig. 6 (c) and (d) demonstrate the real segmentation solutions corresponding in a step of reversible jump, taking fully advantage of the composite cluster.

The overall description for this composite cluster sampling algorithm is summarized in Algorithm 1.

Algorithm 1. Inference Algorithm

```
Input: testing image $I$, superpixel-based candidate set $V_T$ , template-based
       candidate set $V_S$
Output: convergence solution $W^* \sim P(W|I)$
Construct graph representation: $G =< V, E >$.
repeat to sample loop for $W$ to get the final solution
    begin Cut edges to form CCP sets $\{V_i, i = 1, 2, \ldots, M\}$ by edge strength
        for each contextual edge do
            if two vertices have the same state then
                cut the edge by probability $1 - \rho_e^+$
            else
                cut the edge deterministically
        for each competitive edge do
            if two vertices have the same state then
                cut the edge deterministically
            else
                cut the edge by probability $1 - \rho_e^-$
    end
    Randomly select a composite $CCP$;
    Revalidated each sub-CCP of $CCP$ to form a new state $W'$;
    Calculate the accept probability $\alpha(W \to W')$ by Eq. 11 to move to next
    solution or not.
until Predefined criterion is satisfied.;
```

5 Experiments

We evaluate our approach on two public data sets: (i) MSRC 21-class database [3] that contains 591 images in total, and (ii) LHI 17-class database [21] including $17 \times 15 = 255$ images. Compared to the LHI database, the MSRC database was published earlier and many researchers reported result on it, however its groundtruth annotation is relatively rough as a benchmark for semantic parsing task. For both data sets, the images are normalized into size of 320×213 and all images are split randomly into roughly 45% for training, 10% for validation and 45% for testing as well as [3] does. The algorithm is implemented by C++ on a PC with Core Duo 2.8 GHZ CPU. The computation cost is comparatively lower and it spends around $40 \sim 60s$ per image. The average sampling cost time for convergence is only around $4 \sim 6$ seconds based on the generated candidates. The speed reported in [3] was 3 minutes and 70 seconds in [8].

MSRC 21-calss databae. We randomly split the dataset into 337 images for training and 254 ones for testing, like in [3]. Some typical results are shown in Fig. 7, and the quantitative overall pixel-wise accuracy with comparison is reported in Table 1.

Table 1. Overall pixel-wise accuracy on MSRC 21-class database [3] and LHI 17-class database

Methods	MSRC	LHI
Proposed	77.6%	77.2%
CRF + Rel.Loc. [4]	76.5%	N/A
Bag of Keypoints [24]	75.1%	N/A
Auto-Context [8]	72.9%	N/A
TextonBoost [3]	72.2%	67.2%
Geodesic-distance [22]	N/A	71.4%

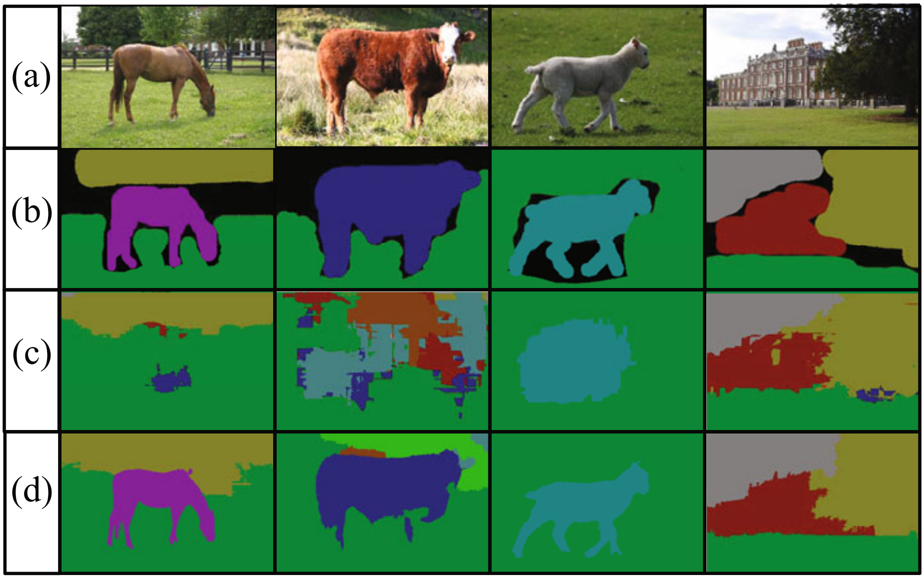
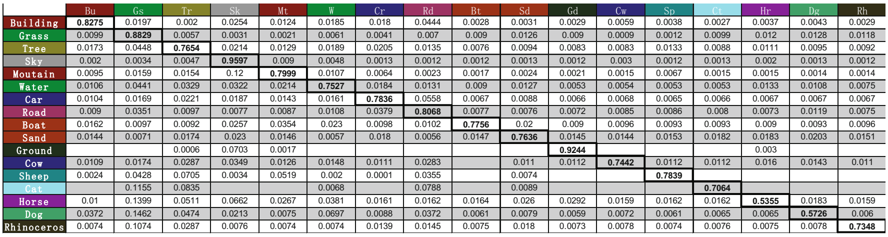


Fig. 7. A few typical results on MSRC 21-class data set. From the top row to the bottom row are: original images, annotated label maps, Result by [3], and our results. The different color denotes the different object category. The overall pixel-wise accuracy is proposed in Table. 1.

LHI 17-class database. We further test our approach on more challenging LHI database that provides more accurate annotated groundtruth. A number of original images, annotation labeling, superpixel-based candidates, template-based candidates, and the final results are presented in Fig. 9. A few examples of iterative sampling are exhibited in Fig. 10(b). The confusion matrix of multi-class recognition for total 17 categories is proposed in Fig. 8, and the overall pixel-wise accuracy on this dataset is 77.2%. The TextonBoost [3] on this dataset outputs 67.2%.

In another comparison, we implement a recently presented geodesic-distance method [22] to achieve segmentation based on our superpixel-based candidates. Like the graph cuts [9], geodesic-distance algorithm deterministically assigns



	Bu	Gs	Tr	Sk	Mt	W	Cr	Rd	Bt	Sd	Gd	Cw	Sp	Ct	Hr	Dg	Rh
Building	**0.8275**	0.0197	0.002	0.0254	0.0124	0.0185	0.018	0.0444	0.0028	0.0031	0.0029	0.0059	0.0038	0.0027	0.0037	0.0043	0.0029
Grass	0.0099	**0.8829**	0.0057	0.0031	0.0021	0.0061	0.0041	0.007	0.009	0.0126	0.009	0.0009	0.0012	0.0099	0.012	0.0128	0.0118
Tree	0.0173	0.0448	**0.7654**	0.0214	0.0129	0.0189	0.0205	0.0135	0.0076	0.0094	0.0083	0.0083	0.0133	0.0088	0.0111	0.0095	0.0092
Sky	0.002	0.0034	0.0047	**0.9597**	0.009	0.0048	0.0013	0.0012	0.0012	0.0013	0.0012	0.003	0.0012	0.0013	0.002	0.0013	0.0012
Moutain	0.0095	0.0159	0.0154	0.12	**0.7999**	0.0107	0.0064	0.0023	0.0017	0.0024	0.0021	0.0015	0.0067	0.0015	0.0015	0.0014	0.0014
Water	0.0106	0.0441	0.0329	0.0322	0.0214	**0.7527**	0.0184	0.0131	0.009	0.0127	0.0053	0.0054	0.0053	0.0053	0.0133	0.0108	0.0075
Car	0.0104	0.0169	0.0221	0.0187	0.0143	0.0161	**0.7836**	0.0558	0.0067	0.0088	0.0066	0.0068	0.0065	0.0066	0.0067	0.0067	0.0067
Road	0.009	0.0351	0.0097	0.0077	0.0087	0.0108	0.0379	**0.8068**	0.0077	0.0076	0.0072	0.0085	0.0086	0.008	0.0073	0.0119	0.0075
Boat	0.0162	0.0097	0.0092	0.0257	0.0354	0.023	0.0098	0.0102	**0.7756**	0.02	0.009	0.0096	0.0093	0.0093	0.009	0.0093	0.0096
Sand	0.0144	0.0071	0.0174	0.023	0.0146	0.0057	0.018	0.0056	0.0147	**0.7636**	0.0145	0.0144	0.0153	0.0182	0.0183	0.0203	0.0151
Ground			0.0006	0.0703	0.0017						**0.9244**				0.003		
Cow	0.0109	0.0174	0.0287	0.0349	0.0126	0.0148	0.0111	0.0283		0.011	0.0112	**0.7442**	0.0112	0.0112	0.016	0.0143	0.011
Sheep	0.0024	0.0428	0.0705	0.0034	0.0519	0.002	0.0001	0.0355		0.0074			**0.7839**				
Cat		0.1155	0.0835			0.0068		0.0788		0.0089				**0.7064**			
Horse	0.01	0.1399	0.0511	0.0662	0.0267	0.0381	0.0161	0.0162	0.0164	0.026	0.0292	0.0159	0.0162	0.0162	**0.5355**	0.0183	0.0159
Dog	0.0372	0.1462	0.0474	0.0213	0.0075	0.0697	0.0088	0.0372	0.0061	0.0079	0.0059	0.0072	0.0061	0.0065	0.0065	**0.5726**	0.006
Rhinoceros	0.0074	0.1074	0.0287	0.0076	0.0074	0.0074	0.0139	0.0145	0.0075	0.018	0.0073	0.0078	0.0074	0.0076	0.0075	0.0078	**0.7348**

Fig. 8. Confusion matrix of labeling for total 17 categories on the LHI 17-class [21] database. The overall accuracy is 77.2%.

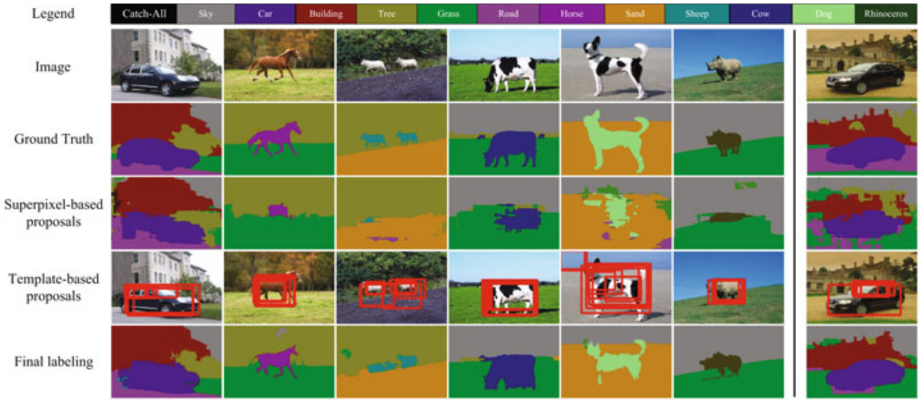


Fig. 9. Example results on LHI 17-calss database [21]. We demonstrate a few original images, annotation labeling, superpixel-based candidates, template-based candidates, and the final results from the top row to the bottom row. The column on the right is a failure example.

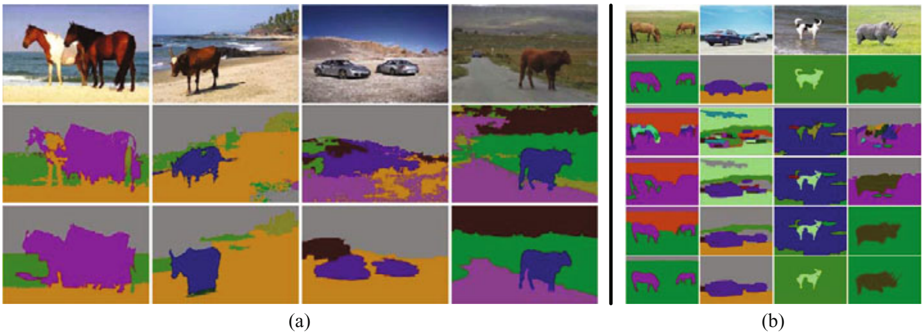


Fig. 10. (a)A comparison with geodesic-distance method (in the middle row). The original images and our results are shown in the top row and the bottom row respectively. (b)An illustrative examples of iterative sampling. The top two rows exhibit the original images and the groundtruth annotation. The other rows (3-rd ∼ 6-th) show the results in iterative sampling.

label to each pixel based on confident initialization. In this experiment, we set the initialization by a few candidates with low energy cost. The overall accuracy reaches 71.4%. However, this deterministic algorithm often depends on confident initialization and may stuck in local minimal. Three typical examples of geodesic-distance method are exhibited in Fig. 10(a) to compare with our method.

6 Summary

For the semantic scene understanding task, this paper studies a candidacy graphical representation of integrating the textural appearance and shape information. In contrast to the current methods using textural appearance and pixel-level context information, we additionally explore the object structural model in the candidacy representation, as well as the competitive and contextual interactions. An efficient composite sampling algorithm based on this representation is proposed in the Bayesian framework. Unlike the traditional single-site sampler, this algorithm updates large portions of the solution space quickly to minimize constraint energy, by clustering connected components in each sampling step. Our approach is test on both LHI and MSRC public data sets and outperforms the state-of-art methods.

Acknowledgement. We also thank Jun Zhu, Tianfu Wu, Lin Liang, Xiang Bai, and Yu Zhou for their helpful discussions. This work was supported by NSFC No.60873127, 60903096 and 60903172.

References

1. Laferte, J.M., Heitz, F., Perez, P., Fabre, E.: Hierarchical statistical methord for the fusion of multiresolution data. In: ICCV (1995)
2. Xuming, H., Zemel, R.S., Carreira-Perpinan, M.A.: Multiscale conditional random fields for image labeling. In: CVPR (2004)
3. Shotton, J., Winn, J., Rother, C., Criminisi, A.: Textonboost: Joint appearnce, shape and context modeling for multiclass object recognition and segmentation. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 1–15. Springer, Heidelberg (2006)
4. Gould, S., Rodgers, J., Cohen, D., Elidan, D., Koller, D.: Multi-class segmentation with relative location prior. IJCV 80, 300–316 (2008)
5. Galleguillos, C., Rabinovich, A., Belongie, S.: Object categorization using co-occurrence, location and appearance. In: CVPR (2008)
6. Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., Belongie, S.: Object in context. In: ICCV (2007)
7. Bastian, L., Ales, L., Bernt, S.: Combined object categorization and segmentation with an implicit shape model. In: ECCV Workshop on Statistical Learning in Computer Vision (2004)
8. Zuowen, T.: Auto-context and its application for high-level vision tasks. In: CVPR (2008)
9. Kolmogorov, V., Zabin, R.: What energy functions can be minimized via graph cuts? PAMI 26, 147–159 (2004)

10. Frey, B.J., Mackay, D.: A revolution: Belief propagation in graphs with cycles. In: NIPS (1997)
11. Apt, K.: The essence of constraint propagation. Theoretical Computer Science 221, 179–210 (1999)
12. Geman, S., Geman, D.: Stochastic relaxation, gibbs distributions and the bayesian restoration of images. PAMI 6, 721–741 (1984)
13. Ferrari, V., Jurie, F., Schmid, C.: Accurate object detection with deformable shape models learnt from images. In: CVPR (2007)
14. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. PAMI 24, 509–522 (2002)
15. Yinnian, W., Zhangzhang, S., Songchun, Z.: Deformable template as active basis. In: ICCV (2007)
16. Xiang, B., Xinggan, W., Longin, J.L., Wenyu, L., Zuowen, T.: Active Skeleton for Non-rigid Object Detection. In: ICCV (2009)
17. Borenstein, E., Ullman, S.: Combined top-down/bottom-up segmentation. PAMI 30, 2109–2125 (2008)
18. Tu, Z.W., Chen, X., Yulle, A., Zhu, S.: Image parsing: Unifying segmentation, detection, and recognition. IJCV 63 (2005)
19. Metropolis, N.: Equation of state calculations by fast computing machines. Journal of Chemical Physics 21, 1087–1092 (1953)
20. Barbu, A., Zhu, S.: Generalizing swendsen-wang for image analysis. Journal of Computational and Graphical Statistics 16, 877–900 (2007)
21. Yao, B., Yang, X., Zhu, S.-C.: Introduction to a large-scale general purpose ground truth database: Methodology, annotation tool and benchmarks. In: Yuille, A.L., Zhu, S.-C., Cremers, D., Wang, Y. (eds.) EMMCVPR 2007. LNCS, vol. 4679, pp. 169–183. Springer, Heidelberg (2007)
22. Bai, X., Sapiro, G.: Geodesic matting: A framework for fast interactive image and video segmentation and matting. IJCV 82, 113–132 (2009)
23. Torralba, A., Murphy, K.P., Freeman, W.T.: Sharing features: efficient boosting procedures for multiclass object detection. In: CVPR (2004)
24. Lin, Y., Meer, P., Foran, D.J.: Multiple class segmentation using a unified framework over mean-shift patches. In: CVPR (2007)