Multi-scale Context for Scene Labeling via Dynamic Segmentation Graph

Quan Zhou, Longin Jan Latecki, *Senior Member, IEEE*, and Wenyu Liu, *Member, IEEE*

***Abstract*—Using contextual information for scene labeling has gained substantial attention in the fields of image processing and computer vision. In this paper, a fusing model using dynamic segmentation graph (DSG) is presented to explore multi-scale context for scene labeling problem. Given a family of segmentations, the representation of DSG is established based on the spatial relationship of these segmentations. In the scenario of DSG, the labeling inference process is formulated as a linear fusing model, trained from the discriminative classifiers. Compared to previous approaches, which usually employ Conditional Random Fields (CRFs) or hierarchical models to explore contextual information, our method produces a more powerful representation that captures wide variety of visual context for the task of image labeling. Our approach yields state-of-the-art results on the MSRC dataset (21 classes) and the LHI dataset (15 classes) and near-record results on the SIFT Flow dataset (33 classes), while producing a 320×240 scene labeling in less than a second.**

***Index Terms*—Scene labeling, multi-scale context, dynamic segmentation graph, feature extraction, classification.**

I. Introduction

MULTI-CLASS segmentation plays an important role in the field of computer vision, leading to the challenging problem of *scene labeling* that is an important function for image processing and understanding. From the perspective of human vision system, scene labeling aims to automatically partition an image into semantic regions. On the other hand, from the perspective of computer vision, the goal is to assign each pixel a semantic label which indicates its potential category, achieving synchronous recognition and delineated segmentation for every object in nature image. Scene labeling is related to many applications, such as object detection [1], [2], scene understanding [3], image alignment [4] and image matching [5]. Therefore, this problem has been extensively studied in image processing, computer vision and machine learning literature [6], [7], [8], [9], [10], [11], [12].

Recently, using contextual information has gained increasing attention for the scene labeling problem [9], [13], [14], [15]. According to the usage of contextual information, most of existing models are mainly divided into two categories: short-ranged [9], [15], [16], [17] and long-ranged interactions [18], [19], [20], [21], [22]. Although these successful approaches have achieved promising results, there are still two issues of primary importance to be considered in the contextual modeling for scene labeling:

- How to produce a good and powerful representation to capture the visual context information?
- How to integrate the contextual information into a well-designed model to ensure the self-consistency of labeling results?

This paper presents a scene labeling system based on multi-scale contextual formulation to approach both questions. The main idea consists of two-folds: (a) A common observation is that identifying a larger image region provides strong evidence for classifying the contained smaller ones. For example, if a region is recognized as "grass", it indicates that the covered smaller ones are more likely to be also labeled as "grass". Thus, the containing spatial relationships among regions are considered to investigate this kind of context. (b) However, only using one scale context is difficult to assign semantic category for each pixel. The category of a pixel may rely on relatively short-range intersections, but may also depend on long-range information. For instance, recognizing a gray pixel belonging to a "road" or a "concrete building" requires wide scale contextual clues that show enough of the surroundings to make an informed decision. To address these problems, we propose using the representation of *dynamic segmentation graph (DSG)*, which is able to efficiently investigate covering spatial relationships among the ensemble segmentations. Thereafter, the contextual cues are embedded into a linear fusing model according to the established DSG. The proposed method is able to take into account multi-scale context information more efficiently and flexibly, while keeping the number of free parameters to be minimum.

The traditional approaches [2], [6], [19], [23], [24] for scene labeling usually first produce segmentation hypotheses using over-segmentation algorithms [25], [26]. Candidate segments are then organized in a hierarchical structure to explore visual context [14], [20], [27], [28]. Finally, a conditional random field (CRF), markov random field (MRF) [2] or discriminative random field (DRF) [29] is trained to produce segment categories and to ensure that the labeling results are globally consistent. A striking characteristic of our approach is that the usage of DSG to assign pixel labels reduces the sophisticated inference process of graphical models, while still ensures the consistency of the labeling output.

Specifically, the proposed scene parsing architecture is depicted in Figure 1, which relies on following two components.

Quan Zhou is with the College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing, China 21003.

Longin Jan Latecki is with Department of Computer and Information Science, Temple University, Philadelphia, USA.

Wenyu Liu is with the Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan, China 430074.

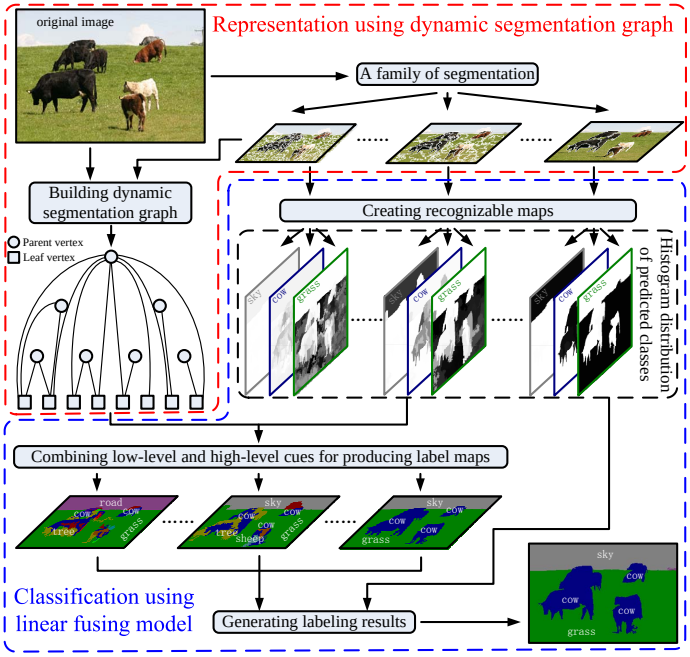

Fig. 1. **Diagram of our scene labeling approach.** The raw input image is partitioned into a series of super-pixels to construct a family of segmentation, which is organized in an incrementally coarse to fine manner. Our dynamic segmentation graph is established based on the spatial interactions between these super-pixels. In parallel, the recognizable maps, as histogram distribution of predicted classes, are created for high level segmentations. In each recognizable map, white represents low confidence, while black indicates high confidence. Thereafter, a series of label maps are produced based on the low-level features and high-level cues. Two types of maps are integrated to produce the final labeling result. (best viewed in color)

A. Multi-level, Dynamic Graph-structure Representation

A family of segmentation in coarse to fine manner is constructed over an input image to investigate contextual cues at multiple levels. In each level, an over-segmentation is performed to partition input image into disjoint regions. This segmentation family might be any set of segmentations, for instance, a collection of super-pixels either produced via different segmentation algorithms [14], [20], [26], [30], [31], [32] or using the same algorithm with different parameter settings. Given a family of segmentation, a DSG is established based on the containing spatial relationship among the segmented super-pixels. The vertex set of DSG consists of two components: leaf vertices and parent vertices. Each leaf vertex is defined in the finest segmentation level, and inherits from some parent vertices, where associated regions cover the smaller one of leaf vertex. Meanwhile, one parent vertex may also connects with several leaf vertices in the finest level. Since the parent vertices may be from inconsecutive high level segmentations, our segmentation graph has dynamic structure so that we can explore contextual information more sufficiently and flexibly. To our best knowledge, DSG has been rarely used in the scenario of scene labeling, and provides a robust and powerful representation, allowing us to detect and recognize all the objects and regions in the given scene.

B. Classification Using Linear Fusing Model

In the scenario of DSG, a linear fusing model is established using Bayesian rule to capture multi-scale contextual information. It treats scene labeling as a hidden variable integration problem, where the hidden variables are the parent vertices and their label hypotheses. Since the corresponding region of each parent vertex is encoded by an aggregated low-level appearance feature vectors (e.g., color and texture), a classifier is then applied to the aggregated features. As a result, a series of recognizable maps are created as the histogram distribution of all object categories presented in the parent vertices. These high-level label hypotheses predictions, once again aggregated with the appearance cues, are fed into another classifier for identifying the leaf vertices. Finally, the output of two types of classifiers are linearly integrated to assign a final single class per leaf vertex. Our linear fusing model is very simple and effective, leading to parse a 320×240 image in less than a second using a conventional personal computer. The bulk of the computation lies in the training procedure of classifiers. Once trained, our linear fusing model is parameter free, and needs no revision of thresholds or other knobs.

The remainder of this paper is organized as follows. After a brief discussion of related work in Section II, we describe the method on establishing DSG in Section III. Section IV elaborates on the details of linear fusing model based on DSG. Experimental results are given in Section V. Finally, we give the concluding remarks in Section VI.

II. Related Work

The scene labeling problem has been addressed with a wide variety of methods in last decade. Most methods mainly employ Markov random fields (MRFs), Conditional random fields (CRFs), or hierarchical models to exploit contextual clues for generating the consistent labeling output [2], [23], [24], [27], [33], [34]. We review the related work from two aspects of contextual modeling for scene labeling problem: short-ranged and long-ranged interactions.

A. Short-ranged Intersections

Many of the labeling approaches [2], [9], [15], [16], [35] construct successful systems that capture short-ranged contextual information based on the image statistics of surrounding patches or regions. As one of the earliest methods, Belonge *et al.* [16] proposed a "looking around" operation to encode local contextual information. Tu and Bai [9] formulated visual context using surrounding patches within rigid position. In [15], the authors computed contextual clues by considering the joint appearance of bottom-up image features. Kumar and Hebert introduced a Discriminative Random Field (DRF) [29] which is defined on a graph within a two-dimensional lattice structure. DRF learns pairwise compatibilities including image information between labels of different nodes. The short-ranged interactions can be also encoded in terms of *generative* manner, such as MRFs [2], [17]. In MRFs, neighboring label variables are connected to each other so that their values are not independent. By combining local pairwise interactions between variables, MRFs impose a global constraint on the

label predictions, leading to more consistent labeling results of an image.

B. Long-ranged Intersections

Besides using short-ranged intersections, vast majority of image labeling methods attempt to explore global-based context using CRFs. He *et al.* [36], for example, attempted to infer an environment-specific prior to guide segmentation; Hoiem *et al.* [3] presented a system combining the interaction between different objects in a loop as mutual support; Belongie *et al.* [37] and Gould *et al.* [38] employed the co-occurrence preference and relative location as contextual features in their probabilistic construction. An alternative approach of using CRFs to capture long-ranged context is to integrate object detection into probabilistic graphical models [1], [6], [11], [39], [40], [41], which combine pixel-based, object-based and scene-based clues for scene labeling problem. Unlike these methods, we employ the covering relationship to capture visual context, and investigate this kind of context in different scales using DSG, without requiring some complex postprocessing to yield cleanly delineated predictions, such as sampling technique [2], [28] or graph cut algorithms [27], [42].

Another better approach for capturing long-ranged context is to build hierarchical models [21], [27], [34], [42], [43]. Some authors employ the families of segmentations to generate the representation of segmentation tree, which is organized within a rigid hierarchical structure via aggregating elementary segments [20], [28]. Other strategies using families of segmentations appeared in [14] and [30]. Compared with these approaches, we establish a robust and powerful representation, DSG, which is more flexible, efficient, and dynamic, allowing a wide variety of long-ranged contextual cues within different scales to contribute to the confidence in each semantic label. Additionally, the proposed method is fully automatic, without any human interactions as well as [21] does.

An early version of this work was first published in [31]. This journal version extends previous one in three aspects: the previous version needs well-designed segmentation hypothesis, still resulting in hierarchical representation, while we proposed DSG, without the hierarchical constraint; besides using texture cues to encode image regions, we also address the cues of color, size, shape and the class distributions to predict region labels; we have implemented more complete experiments, and reported more comparisons and higher results.

III. Image Representation Using Dynamic Segmentation Graph (DSG)

Traditional approaches to investigate global contextual information are to consider a segmentation tree [44], [45], where the segmented regions are hierarchically organized. An alternative technique is to compute a set of segmentations using different merging thresholds [30]. In this section, we propose a method to analyze a family of segmentation, which is used to establish the representation of DSG, without restricted to the hierarchical structure.

As shown in Figure 2(a), an input image $\mathcal{I}$ is first partitioned into a series of super-pixels using mean shift segmentation technique [25] with different parameter tunings. Thus a family of segmentation with $\mathcal{L}$ levels is produced from a coarse to fine manner. Let $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ denotes our DSG, in which image $\mathcal{I}$ and containing spatial interactions are encoded based on these super-pixels. In Figure 2, we can see that the vertex $v \in \mathcal{V}$ corresponds to a super-pixel. The vertex set $\mathcal{V} = \{\mathcal{V}_L, \mathcal{V}_P\}$ is composed of two components: the set of leaf vertex $\mathcal{V}_L$ defined on finest segmentation level, and the set of parent vertex $\mathcal{V}_P$ defined on higher segmentation level. Each leaf vertex $v_l \in \mathcal{V}_L$ is associated with two attributes: the index of i^{th} vertex in $\mathcal{V}_L$ (denoted by v_l^i), and the corresponding super-pixel r_i. On the other hand, each parent vertex $v_p \in \mathcal{V}_P$ is also associated with two attributes: the index of j^{th} segmentation level it belongs to (denoted by v_p^j), and the corresponding super-pixel r_j.

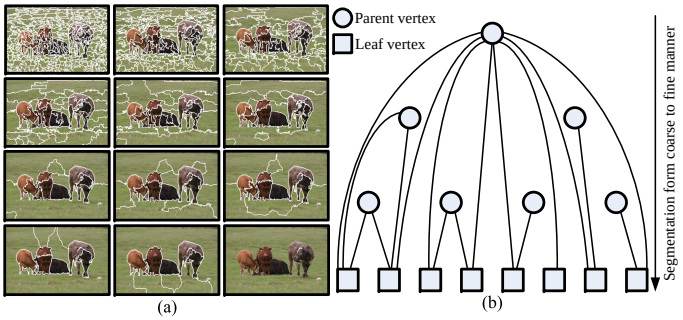

Fig. 2. **Image representation using DSG.** A family of segmentation is illustrated in (a), and (b) shows the sketch map of DSG. In (a), different super-pixels are separated by white boundaries. (best viewed in color)

The edge set $\mathcal{E}$ consists of a set of edges $\varepsilon = \{(v_l, v_p) | v_l \in \mathcal{V}_L, v_p \in \mathcal{V}_P\}$, connecting a leaf vertex v_l and a parent vertex v_p, where the associated super-pixel r_i is covered by the larger super-pixel r_j. Note that there is only one super-pixel r_j containing r_i in j^{th} segmentation level. For leaf vertex v_l^i, we collect all the connected parent vertices, and denote them as $\mathbf{N}_i = \{v_p^j, \forall j \in \{1, 2, \cdots, J\}, 1 \leq J \leq \mathcal{L}\}$. Note that the parent vertex in $\mathbf{N}_i$ may be from inconsecutive segmentations. As shown in Figure 2(b), one parent vertex may connect several leaf vertices and vice versa, leading to the dynamic structure of our DSG.

Ideally, we would evaluate the levels of family segmentation as many as possible so that we can make full use of multi-scale contextual information. However, it requires numerous computational effort in our method, and thus only small level of segmentations can be considered. From Figure 2(a), although those super-pixels tend to be highly irregular in size and shape, the advantage of using [25] is that it can often group large homogeneous regions with similar appearance while dividing heterogeneous regions into many smaller ones. This often produces fewer super-pixels in each level of segmentation. Alternative over-segmentation approaches include hierarchical segmentation [44], graphic-based segmentation [26], normalized cuts [46], and the simple watershed algorithm [47]. These methods require much more processing time to produce super-pixels or the generated super-pixels are always with imprecise boundaries.

IV. Linear Fusing Model

In this section, we first elaborate on the details of the linear fusing model in scenario of DSG, then describe the associated feature and learning algorithm for training this model.

A. Problem Formulation

Given the observation image $\mathcal{I}$ and associated DSG $\mathcal{G}$ with $|\mathcal{V}_L|$ leaf vertices, our formulation contains a set of discrete random variables $\mathbf{Y} = \{Y_1, Y_2, \cdots, Y_{|\mathcal{V}_L|}\}$. The i^{th} element Y_i corresponds to leaf vertex v_l^i, and may take a discrete value from the set of labels: $Y_i = c \in \{1, 2, \cdots, \mathcal{C}\}$. Any possible assignment of labels to the random variables $\mathbf{Y}$ will be called a labeling which takes values from $\mathcal{P} = \mathcal{C}^{|\mathcal{V}_L|}$. Our objective is to compute $\mathbf{Y}^*$ that maximizes a posteriori probability,

$$\mathbf{Y}^* = \arg\max_{\mathbf{Y}\in\mathcal{P}} p(\mathbf{Y}|\mathcal{I}, \mathcal{G}) \tag{1}$$

Due to the non-overlapping of r_i in the lattice defined on image $\mathcal{I}$, the image model $p(\mathbf{Y}|\mathcal{I}, \mathcal{G})$ is assumed to be conditionally independent on $\mathbf{Y}$, then this posteriori probability is further factorized onto each leaf vertex. We thus have,

$$p(\mathbf{Y}|\mathcal{I}, \mathcal{G}) = \prod_{i=1}^{|\mathcal{V}_L|} p(Y_i|v_l^i) \tag{2}$$

In order to get the label posterior for leaf vertex v_l^i, we marginalize over all possible parent vertices defined in $\mathbf{N}_i$ that are connected with v_l^i:

$$\begin{aligned} p(Y_i|v_l^i) &= \sum_{v_p^j \in \mathbf{N}_i} p(Y_i, v_p^j|v_l^i) \\ &\propto \sum_{v_p^j \in \mathbf{N}_i} p(v_p^j|v_l^i) p(Y_i|v_l^i, v_p^j) \end{aligned} \tag{3}$$

where $p(Y_i|v_l^i, v_p^j)$ is the probability to assign a label Y_i to leaf vertex v_l^i given observed parent vertex v_p^j. $p(v_p^j|v_l^i)$ denotes the probability to form the larger super-pixel r_j given the smaller super-pixel r_i. In [30], the authors estimate $p(v_p^j|v_l^i)$ using region homogeneity to merge super-pixels, assuming super-pixels with same region homogeneity should also belong to the same category. Conversely, we assume there is no prior knowledge to guide segmentation, and r_j might be generated by those small super-pixels that are inhomogeneous. For example, as the standing "cow" in the right side of Figure 2, it is quite reasonable to group larger super-pixels containing the "gray leg" and "black neck". As a result, the term $p(v_p^j|v_l^i)$ is considered as a constant in our formulation. Therefore, Equation (3) can be simplified as:

$$p(Y_i|v_l^i) \propto \sum_{v_p^j \in \mathbf{N}_i} p(Y_i|v_l^i, v_p^j) \tag{4}$$

In order to incorporate label hypotheses predictions as high-level semantic contextual clues from parent vertex $v_p^j \in \mathbf{N}_i$, we further marginalize Equation (4) over the potential labels Y_j of v_p^j

$$\begin{aligned} p(Y_i|v_l^i) &\propto \sum_{v_p^j \in \mathbf{N}_i} \sum_{Y_j} p(Y_i, Y_j|v_l^i, v_p^j) \\ &\propto \sum_{v_p^j \in \mathbf{N}_i} \sum_{Y_j} p(Y_j|v_l^i, v_p^j) p(Y_i|Y_j, v_l^i, v_p^j) \end{aligned} \tag{5}$$

Since super-pixel r_j always contains super-pixel r_i, the first term can be treated as independent of v_l^i. Equation (5) thus reduces to

$$p(Y_i|v_l^i) \propto \sum_{v_p^j \in \mathbf{N}_i} \sum_{Y_j} p(Y_j|v_p^j) p(Y_i|Y_j, v_l^i, v_p^j) \tag{6}$$

As can be seen, $p(Y_j|v_p^j)$ denotes the normalized probability that assigns semantic label Y_j for parent vertex v_p^j based on image features abstracted from corresponding super-pixel r_j. This term indicates that classifying larger regions may be helpful to identify smaller ones. From $p(Y_i|Y_j, v_l^i, v_p^j)$, it is clear that besides local clues of v_l^i, identifying leaf vertex v_l^i requires to consider image statistics and high-level label predictions from parent vertex v_p^j. Note the sum is over the parent vertices $v_p^j \in \mathbf{N}_i$, rather than all the parent vertices in DSG, which results in fast computational speed to estimate labeling results of input test image.

The optimal labeling output $\mathbf{Y}^*$ can be achieved by assigning a label with the highest recognizable probability to each leaf vertex. Immediately below, we will entail the associated features and the forms of $p(Y_j|v_p^j)$ and $p(Y_i|Y_j, v_l^i, v_p^j)$ using simple regression boosted classifiers [48].

B. Features

To determine the most probable label for each vertex in DSG, we are required to use all available cues, including low-level image statistics, such as color, size, shape, location, texture, and high-level semantics as the estimated histogram of all object categories. Some of these statistics, however, are only helpful when object category has less visual variance. For example, it is hard to identify "car" that are with different colors. Therefore, our approach computes all cues that might be useful for classification, and allows our classifier (described in Section IV-C) to automatically decide which one should be used and how to use them.

1) Low-level Features: Our low-level features build on those of Barnard *et al.* [49] and Zhang *et al.* [50], consisting of mean, standard deviation, skewness, kurtosis, color histograms and bag of features (BoF) over the super-pixel of:

- RGB color-space components (4×3) and RGB color histogram distributions (10×3)
- CIELab color-space components (4×3) and CIELab color histogram distributions (10 bins for L-channel)
- HSV color-space components (4×3 with additional 5 bin and 3 bin histograms for hue and saturation, respectively)
- Size cues as the ratio of region area to entire scene (1)
- Location cues as the offsets in x and y direction, and distance from image center (3)
- Shape cues as the ratio of the region area to perimeter squared, the moment of inertia about the center of mass, and the ratio of area to bounding rectangle area (3)
- Texture cues drawn from 17 filter responses, including Gaussian, oriented Gaussian, Laplacian-of-Gaussian, and pattern features such as corners and bars (4×17)
- Texture cues drawn from BoF as histogram distribution of learned visual dictionary words (700)

2) High-level Features: Unlike the low-level clues, high-level features provide semantic information for recognizing objects and image regions [18], [51]. According to Equation (6), identifying the label of leaf vertex v_l^i requires to consider the label variable Y_j of parent vertex v_p^j. It is a $\mathcal{C}$-dimensional vector as the distribution of classes present in super-pixel r_j. Given the ground truth segmentation in the training process of $p(Y_i|Y_j, v_l^i, v_p^j)$, the features can be directly computed. At test stage, however, no ground truth segmentation is available, therefore, we need a function that can predict the cost of class distribution for r_j. In practise, we directly apply the trained classifiers $p(Y_j|v_p^j)$ over super-pixel r_j and collect the outputs of classifiers to form this high-level semantic feature.

Let X_i denote the feature vector to describe the super-pixel r_i, we append the additional description vector, considering the weighted average over its neighbors,

$$\frac{\sum_{r_{ik}\in\mathcal{N}(r_i)} |r_{ik}| \cdot X_{ik}}{\sum_{r_{ik}\in\mathcal{N}(r_i)} |r_{ik}|} \tag{7}$$

where $\mathcal{N}(r_i)$ is the set of super-pixels which are adjacent with r_i in the image $\mathcal{I}$, and $|r_{ik}|$ is the number of pixels in super-pixel r_{ik}. The same operation is also applied to X_j associated with super-pixel r_j in j^{th} segmentation level. Denote X be the final feature vector to represent a vertex in DSG, then it is a $(1718+\mathcal{C})\times 2$-dimensional vector for a leaf vertex, and a 1718-dimensional vector for a parent vertex. These features can be quickly computed from super-pixels, and provide sufficient statistics.

Algorithm 1: Training boosted decision trees

Input: $X_1, \cdots, X_m$: training data; $\mathbf{w}^1 = \{\omega_1^1, \cdots, \omega_m^1\}$: initial weights, where $\omega_i^1 = \frac{1}{m}$, $i = \{1, \cdots, m\}$; $y_1, \cdots, y_m \in \{1, -1\}$: labels; $s = 2$: number of nodes per decision trees; K: number of iterations

Output: $h_1, \cdots, h_K$: decision trees; $f_1^1, \cdots, f_K^s$: weighted log-ratio for each node of each tree

for *k = 1 to K* **do**

- Learn k-node decision tree $h_k(X)$ based on weighted distribution $\mathbf{w}^k$.
- Assign to each node of $h_k(X)$: $f_{k,s} = \frac{1}{2}\log\frac{\sum_{i:y_i=1, X_i\in f_{k,s}} \omega_i^k}{\sum_{i:y_i=-1, X_i\in f_{k,s}} \omega_i^k}$, $s = \{left, right\}$
- Update weights: $\omega_i^{k+1} = \frac{1}{1+\exp(y_i \sum_{k'=1}^{k} f_{k',s_{k'}})}$ with $s_{k'} : X_i \in T_{k',s_{k'}}$.
- Normalize weights so that $\sum_i \omega_i^{k+1} = 1$.

end

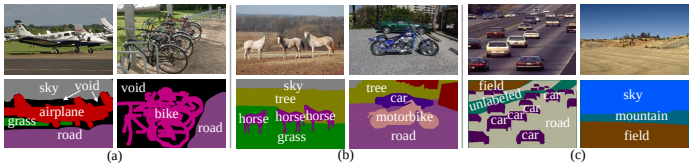

Fig. 3. **Example images of (a) MSRC 21-class (b) LHI 15-class, and (c) SIFT flow 33-class dataset.** The first row displays the original images, and the second row is the corresponding ground truth. For clarity, textual labels have also been superimposed on the ground truth. (best viewed in color)

C. Classifiers and Learning Algorithm

In this paper, both $p(Y_j|v_p^j)$ and $p(Y_i|Y_j, v_l^i, v_p^j)$ can be trained using logistic regression version of Adaboost [48]. For notation simplicity, we use $p(Y|X)$ to represent $p(Y_j|v_p^j)$ and $p(Y_i|Y_j, v_l^i, v_p^j)$, respectively, then the form of $p(Y|X)$ is

$$p(Y = c|X) \propto \exp\{H^c(X)\} \tag{8}$$

where $H^c(X) = \sum_{k=1}^{K} h_k^c(X)$ is an additive model for the c^{th} category by accumulating the classification confidences of K weak learners $h_k^c(X)$. Each weak learner $h_k^c(X)$ adopts a two-terminal node decision tree ("stump") based on input X, and has the following form:

$$h_k^c(X) = f_{k,left}^c \mathbf{1}_{[x_k^m \le \tau_k]} + f_{k,right}^c \mathbf{1}_{[x_k^m > \tau_k]} \tag{9}$$

where x_k^m denotes the m^{th} variable of X selected into the k^{th} weak learner, τ_k is the split-point. $f_{k,left}^c$ and $f_{k,right}^c$ are the weighted log-ratio for the left and right terminal nodes, respectively.

Decision trees make good weak learners, since they provide automatic feature selection and limited modeling of the joint statistics of data. Each decision tree provides a partitioning of the data and outputs a confidence-weighted decision which is the class-conditional log-likelihood ratio for the current weighted distribution. The logistic regression version of Adaboost differs from the original confidence weighted version by only a slight change in the weight update rule, but it results in confidence outputs that tend to be well-calibrated probabilities (after applying the simple sigmoid conversion to the log-ratio output).

In our implementation, all the classifiers are trained in a one vs. all fashion. Here, we take positive examples as the super-pixels which are assigned to that class in the ground truth labeling, and negative examples as all super-pixels which are assigned to a different class in the ground truth. For instance, to distinguish "tree" class, we train the classifiers that estimate the probability of a super-pixel being the remaining semantic labels. The trained classifiers are applied for each category c to the vector of descriptors X, and normalize over all classes by $\frac{\exp\{H^c(X)\}}{\sum_c \exp\{H^c(X)\}}$ to ensure that the estimated probabilities sum to one. The classifier training algorithm is summarized in Algorithm 1.

V. Experiments

The purpose of our experiments is to evaluate the effectiveness of our method, and better understand the behavior of our labeling system. Our analysis includes comparison with recent state-of-the-art methods in the literature, the impact of different cues on recognition accuracy, and the influence on segmentation level.

A. Experimental Setting

1) Dataset: We evaluate our system on three challenging datasets: MSRC 21-class dataset [35], LHI 15-class dataset [52] on which related state-of-the-art methods report labeling

TABLE I
PERFORMANCE OF OUR METHOD OVER MSRC 21-CLASS [35] AND LHI 15-CLASS [52] IN TERMS OF RECOGNITION ACCURACY AND IMPLEMENTAL EFFICIENCY. TRAINING TIMES ARE FOR THE WHOLE TRAINING SET, TEST TIMES ARE PER IMAGE.

	MSRC 21-class dataset [35]				LHI 15-class dataset [52]			
Methods	Average Pixel Acc.	Global Pixel Acc.	Training (h)	Testing (s)	Average Pixel Acc.	Global Pixel Acc.	Training (h)	Testing (s)
Ours	**79.4%**	**86.8%**	6.8	**0.68**	**80.1%**	**83.2%**	2.4	**0.63**
FCRF [53]	78.3%	85.5%	3.4	1.2	-	-	-	-
DHM [2]	76.4%	81.7%	**0.6**	8.4	78.2%	81.3%	**0.3**	6.8
HCRF [27]	75.2%	86.3%	5.4	16.8	66.5%	72.3%	2.7	16.6
AC [9]	68.2%	77.7%	7.7	7.4	66.5%	76.5%	3.9	7.6
RL [38]	64.3%	76.5%	5.5	5.7	64.4%	71.6%	2.1	5.1
TB [35]	57.7%	72.2%	6.3	5.4	62.7%	69.3%	3.8	5.2
HDL [14]	74.6%	80.4%	5.8	0.7	69.3%	77.9%	2.5	0.67
SRT [43]	74.1%	81.2%	20	30	-	-	-	-
PM [42]	72.4%	81.9%	8.4	94	66.1%	78.2%	4.5	87
RA [20]	67.3%	75.4%	6.8	2.86	63.6%	71.8%	3.7	2.88
SHL [34]	66.2%	72.9%	9.1	12	61.8%	70.7%	4.4	15
MS [30]	59.5%	72.5%	6.2	2.6	-	-	-	-

accuracy and implemental efficiency, and a more challenging dataset with a larger number of images and classes: SIFT flow 33-class dataset [4]. All images are rescaled to 320×210 resolution in three datasets, and some examples and associated ground truth are illustrated in Figure 3.

The MSRC 21-class dataset [35] is a very popular benchmark for scene labeling, which consists of 591 images including 21 classes: "building", "grass", "tree", "sky", "water", "book", "road", "body", "boat", "flower", "sign", "cow", "sheep", "aeroplane", "face", "car", "bike", "bird", "cat", "dog" and "chair". The pixels labeled as "void" class are not considered during the training and testing for direct comparison.

The LHI 15-class dataset [52] consists of 370 images gathered from Google image search, and includes 15 object categories: "building", "grass", "tree", "sky", "road", "water", "mountain", "airplane", "cow", "horse", "sheep", "car", "elephant", "rhinoceros" and "motorbike". Compared with MSRC 21-class dataset, images in LHI 15-class dataset are well hand-annotated to achieve accurate segmentation.

The SIFT flow 33-class dataset [4] is composed of 2688 images, that have been thoroughly labeled by LabelMe users. The authors used synonym correction to obtain 33 semantic categories, including: "sky", "building", "mountain", "tree", "road", "sea", "field", "grass", "river", "plant", "car", "sand", "rock", "sidewalk", "window", "desert", "door", "bridge", "person", "fence", "balcony", "crosswalk", "staircase", "awning", "sign", "streetlight", "boat", "pole", "sun", "bus", "bird", "moon" and "cow". It is also a fully annotated dataset, most of which are outdoor scenes including street, beach, mountains and fields. Similar as MSRC 21-class dataset, the pixels labeled as "unlabeled" class are not considered during the training and testing for direct comparison.

2) Baselines: To show the advantages of our approach, we selected 12 state-of-the-art models as baselines. Experimental results of some baseline models are produced using default parameter settings given by the authors, while others are directly borrowed in the literature for comparison. All the baselines are divided into two categories: (1) Modeling scene labeling using CRFs or MRFs, including TextonBoost (TB [35]), relative location prior (RL [38]), auto context (AC [9]), hierarchical CRF (HCRF, [27]), dynamic hybrid MRF (DHM [2]), and full-connected CRF (FCRF [53]); (2) Modeling scene labeling using hierarchical trees, including hierarchical deep learning (HDL, [14]), region ancestry (RA, [20]), multiple segmentation (MS, [30]), pylon model (PM, [42]), stacked hierarchical labeling (SHL, [34]), and segmentation and recognition templates (SRT, [43]).

3) Evaluation Metrics: All the labeling models are evaluated based on the following two widely-used criteria [27]. The *global*-based pixel accuracy pays the most attention to frequently occurring objects and penalizes infrequent objects. It refers to overall accuracy among all categories:

$$\frac{\sum_{m\in\mathcal{C}} N_{mm}}{\sum_{m,n\in\mathcal{C}} N_{mn}} \tag{10}$$

where N_{mm} is the number of pixels correctly labeled, and N_{mn} refers to the number of pixels of category m labeled as class n. On the contrary, the *average*-based pixel accuracy evaluates the recognizable accuracy per category, which is defined as:

$$\frac{1}{\mathcal{C}} \sum_{m\in\mathcal{C}} \frac{N_{mm}}{\sum_{n\in\mathcal{C}} N_{mn}} \tag{11}$$

4) Implemental Details: We use the same split setting [35] for MSRA and LHI datasets that randomly splits all images into three sets: 45% for training, 10% for validation and 45% for testing. While for SIFT flow dataset, we use the evaluation procedure introduced in [14]: 2488 images used for training and 200 images used for testing. A computer with 8 GB memory and 2.6 GHz CPU was used for training and testing.

To build our DSG, we are required to compute the spatial relationship of super-pixel r_i and r_j. Let $\mathcal{O} = \frac{|r_i \cap r_j|}{|r_i|}$ be the overlap ratio between r_i and r_j. If $\mathcal{O}$ is larger than a predefined threshold η (set as 0.95 in our experience), the associated parent vertex becomes one element of $\mathbf{N}_i$. On the other hand, to train the classifiers $p(Y_j|v_p^j)$, we need to assign ground truth to the automatically created super-pixels. If nearly all (at least 90% by area) of the pixels within a super-pixel have the same

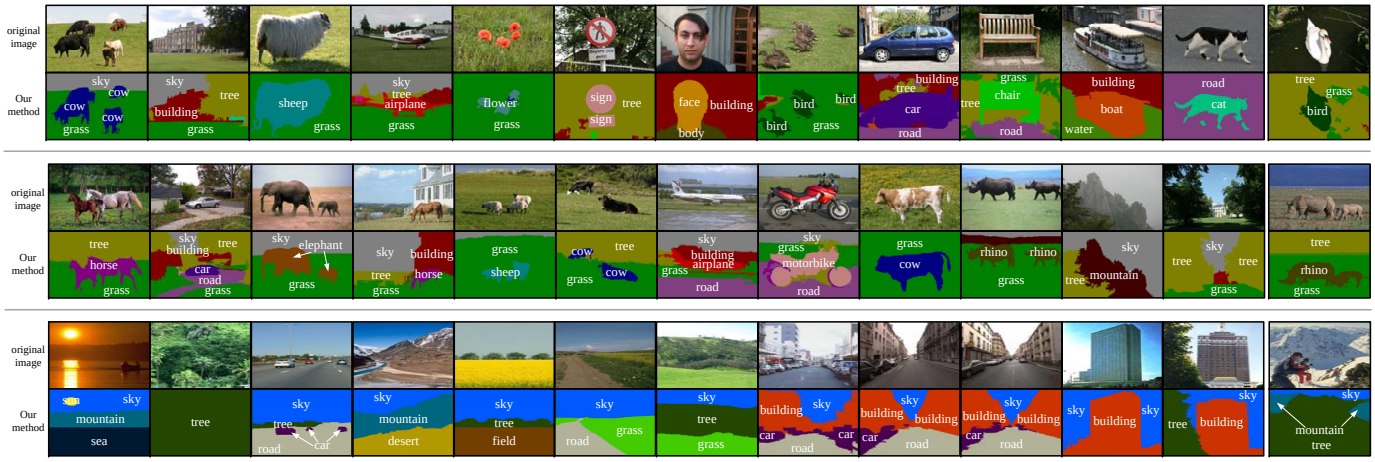

Fig. 4. **Illumination of some labeling results on MSRC 21-class (up panel), LHI 15-class dataset (middle panel) and SIFT flow 33-class dataset (bottom panel).** For clarity, textual labels have also been superimposed on the resulting segmentations and different color denotes different category. (best viewed in color)

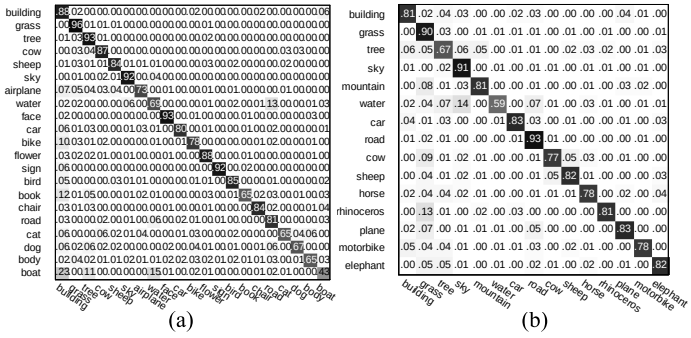

Fig. 5. **Confusion matrices of our method evaluated on (a) MSRC 21-class dataset and (b) LHI 15-class dataset.** The average pixel-wise accuracy achieves 79.4% and 80.1%, respectively.

ground truth label, the super-pixel is assigned that same label. Otherwise, the super-pixel is labeled as "mixed", which is not used in training process of $p(Y_j|v_p^j)$.

B. Results and Analysis

1) Quantitative Results: Table I shows the overall results on MSRA and LHI datasets, and compared with related works. It demonstrates our approach outperforms other state-of-the-art methods both in recognition precision and implementation efficiency, whether using CRFs or hierarchical tree-structure models to capture long-ranged contextual information. This is probably because that our DSG representation, combined with linear fusing model, essentially makes the use of random field model much less useful: The contextual relationships within different scales seem to be efficiently captured by it. Method of [27] also achieves good global-based pixel accuracy on MSRA dataset, and method of [2] implements faster in training process on these two datasets. They are, however, at the price of several seconds to parse one testing image.

Figure 5 illustrates the confusion matrices obtained by applying our approach on MSRC 21-class and LHI 15-class datasets, in which accuracy values are computed as the image pixels assigned to the correct class labels. The results are about

TABLE II
PERFORMANCE OF OUR METHOD OVER SIFT FLOW [4] IN TERMS OF RECOGNITION ACCURACY AND IMPLEMENTAL EFFICIENCY. TRAINING TIMES ARE FOR THE WHOLE TRAINING SET, TEST TIMES ARE PER IMAGE.

	SIFT flow 33-class dataset [4]			
Methods	Average Pixel Acc.	Global Pixel Acc.	Training (h)	Testing (s)
Ours	**52.3%**	71.9%	2.9	**0.64**
HCRF [27]	51.1%	**72.8%**	3.2	16.5
AC [9]	50.7%	72.6%	4.6	7.3
DHM [2]	50.4%	71.2%	**0.4**	8.9
RL [38]	49.5%	70.5%	2.5	5.1
HDL [14]	50.8%	72.3%	3.0	0.72
PM [42]	50.3%	70.4%	5.4	91
RA [20]	48.6%	68.7%	4.4	0.83
SHL [34]	47.8%	67.1%	5.2	15

17 and 12 times better than randomly choosing semantic labels for each pixel on two datasets. We can see that some categories exhibit large errors, e.g., "water" mislabeled as "sky", "book" incorrectly recognized as "building", especially the categories of "boat", "cat", and "dog" on MSRA dataset, which probably due to their extremely inter-class color/texture similarities, or relative small training samples.

We then demonstrate that our method scales nicely when augmenting the number of images and classes on SIFT flow datasets [4] in Table II. Results in Table I and Table II demonstrate the impressive computational advantage of our method over competing models. Establishing dynamic graph-structure representation by computing the spatial relationship of super-pixels allows us to parse an image with resolution 320×240 in less than 1 second, which might be benefit for labeling video sequence or massive image datasets.

2) Qualitative Results: Example results of simultaneous recognition and segmentation over three datasets are shown in Figure 4. Each example shows both the original image and the color coded output labeling. It is evident that our method can handle large appearance variations of object classes. Except the boundary regions that exhibit relative higher confusion,

Fig. 6. **Examples of object segmentation results for MSRC (left panel) and LHI (right panel) datasets using different segmentation levels.** The first collum gives the original image and corresponding ground truth. The rest images show the object recognizable maps by gradually increasing the level $\mathcal{L}$ of DSG. (best viewed in color)

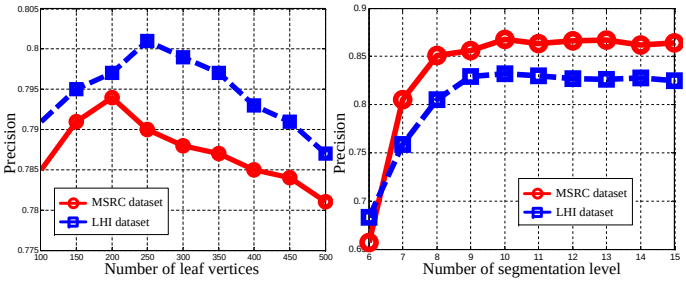

Fig. 7. **Effects on performance of number of leaf vertices (left panel), and the level of segmentation family (right panel).**

TABLE III
CONTRIBUTIONS OF DIFFERENT CUES TO THE PERFORMANCE.

	MSRC [35]		LHI [52]		SIFT flow [4]	
Cues	Average	Global	Average	Global	Average	Global
Txt+Clr+Prd	**78.5%**	**85.2%**	**79.6%**	**85.8%**	**51.1%**	**73.3%**
Txt+Clr	74.7%	80.5%	74.9%	81.7%	49.3%	69.1%
Txt	73.6%	78.4%	72.3%	78.2%	48.9%	67.8%

nearly all super-pixels are correctly classified. On the last column, we show three examples in which the labeling results are not good enough (e.g., "water" and "mountain" incorrectly labeled as "tree", the missed foreground "person"), however, the foreground objects (e.g., "bird" and "rhino") still achieve good segmentation and recognition on MSRC 21-class and LHI 15-class datasets.

C. Other Aspects

1) Analysis of Segmentations: In our experiment, two factors directly affecting the performance are the numbers of leaf vertices $|\mathcal{V}_L|$ and levels of segmentations $\mathcal{L}$. We evaluate the accuracy of our system on MSRC and LHI datasets by changing the values of these two parameters, using all the available cues (texture, color, shape, size, location and label hypotheses predictions). The selection of these two parameters illustrates the trade-off between computational efficiency and recognizable precision.

We first evaluate the effect of the number of leaf vertex on average accuracy given the DSG. In practice, we repeat our experiments to produce different numbers of leaf vertices, and then produce the DSG based on the establishing criteria described in Section III. The left panel of Figure 7 exhibits the plot of average accuracy for increased number of leaf vertices over the testing set. The accuracy of our method peaks at approximately 200 and 250 leaf vertices for MSRC and LHI datasets, respectively, and any refinement to these parameters will result in slightly decrease of performance.

We also measure the effect of changing the number of segmentation levels $\mathcal{L}$. Our implementation uses different levels (from 1 to 15) of segmentations. In the right panel of Figure 7, we display the global accuracy along with the increasing number of segmentation levels. In these experiments, we use the same classifiers trained under our reported results based on DSG but generate new sets of segmentations for testing. As can be seen, more levels of segmentations result in higher accuracy, which shows that using multi-scale context plays a more critical role in object classification. It is observed that the highest performance is achieved when $\mathcal{L} = 10$. Figure 6 displays two examples of all first 10 labeling results for MSRC and LHI datasets.

2) Analysis of Cues: We also wish to evaluate the effectiveness of our three main types of cues: color, texture and label hypotheses predictions as described in Section IV-B. To do this, we train classifiers using different features, and compute the average-based and global-based pixel accuracies of all categories over our testing images. More specifically, we use texture features to train classifiers $p(Y_j|v_p^j)$ and $p(Y_i|Y_j, v_l^i, v_p^j)$ as baseline. Then the color and label hypotheses prediction cues are sequentially introduced. In these experiments, we employ DSG representation, using the same segmentation family as were used to report accuracy. The contributions of different cues are listed in Table III.

From Table III, the simple texture feature proves to be surprisingly effective on three datasets, it achieves 73.6%, 72.3%, and 48.9% average-based, and 78.4%, 78.2%, and 67.8% global-based recognizable accuracy, respectively. We also conclude that the color cues slightly improve the results (only increasing 1.2% and 2.3% for average and global accuracy on three datasets). Similar to color cue, the results reported from Table I, Table II and Table III demonstrate the remaining size, shape and location cues only achieve 1.0% and 0.5% improvement for average and global accuracy. Compared with these low-level cues, the label hypotheses prediction cue by itself seems remarkably effective at discriminating among the classes. Perhaps this high-level feature provides the useful and discriminative information to capture scene-level relationships from different level of DSG. Figure 8 shows two example of visual labeling results on MSRC and LHI datasets, comparing the individual contribution of induced cues and the overall results, respectively. It seems to be that the color feature is highly effective for some specific classes, such as "sky", "grass" and "tree", while not effective for the categories with

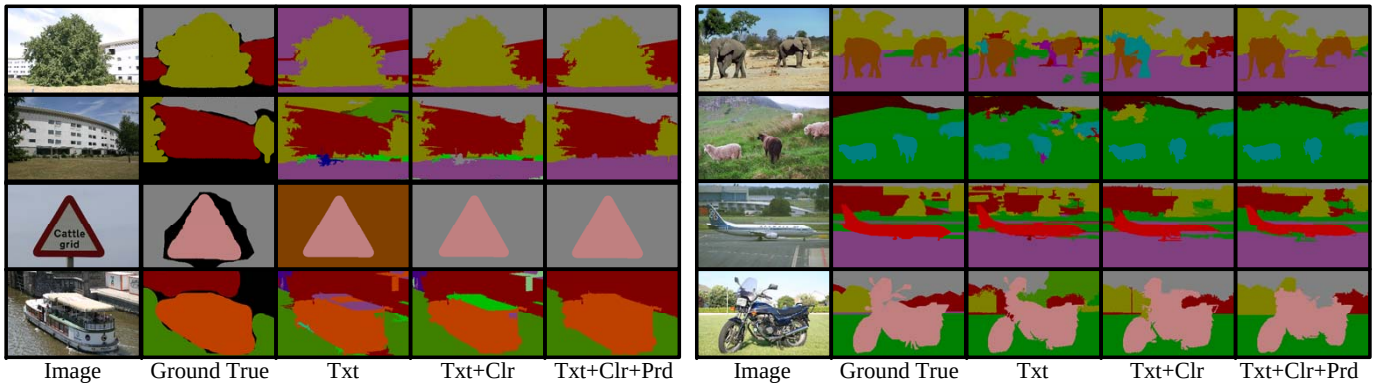

Fig. 8. **Results when performing classification based on sequentially inducing the cues on MSRC (left panel) and LHI (right panel) datasets.** In each case, the same segmentation family and DSG are used, and those super-pixels are described with the given type of cues. (best viewed in color)

great color variance, such as "boat" and "motorbike".

VI. Conclusion

This paper extends our preliminary research [31] and describes a linear fusing model using DSG to explore the multi-scale context information for scene labeling problem. The proposed model is trained from fully labeled images in a supervised manner to learn appropriate low-level features and high-level hypotheses to predict pixel labels. Firstly, a DSG is built based on spatial relationships of segmented super-pixels to represent an entire scene. Then, a linear fusing model, also called integral model, is established to integrate multi-scale visual context for producing consistent recognition and segmentation results. Compared with all previously published results, our method has two main advantages and characteristics for the task of scene labeling problem, as listed below:

- Using the DSG representation is flexible, efficient, and dynamic to capture multi-scale visual context, and yields excellent results when compared with methods that employ CRFs or hierarchical tree-structure models. The labeling accuracy is similar to or better than competing models, even when the segmentation hypothesis generation and the postprocessing module are absent or very simple.
- When a wide scale context is taken into account to predict pixel label, the role of the postprocessing is greatly reduced. This seems to suggest that multi-scale visual context can be taken into account using our linear fusing model based on DSG, perhaps as well as a traditional inference mechanism that propagates vertex label constraints over an entire graphical model, but with a considerably lower computational cost.

Despite obtaining impressive results, we believe that even better results can be achieved by automatically learning the dynamic structure representation. We are aware of a related work [14] in this direction. Additionally, we would like to use other forms of texture features such as clustered SIFT descriptors [54] and incorporate middle-level object shape information [1], [55] as supplementary cues to improve the performance, while retaining high efficiency.

References

[1] Y. Yang, S. Hallman, D. Ramanan, and C. Fowlkes, "Layered object detection for multi-class segmentation," in *CVPR*, 2010, pp. 3113–3120.
[2] Q. Zhou, J. Zhu, and W. Liu, "Learning dynamic hybrid markov random field for image labeling," *TIP*, vol. 22, no. 6, pp. 2219–2232, 2013.
[3] D. Hoiem, A. Efros, and M. Hebert, "Closing the loop in scene interpretation," in *CVPR*, 2008, pp. 1–8.
[4] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing: Label transfer via dense scene alignment," in *CVPR*, 2009, pp. 1972–1979.
[5] ——, "Nonparametric scene parsing via label transfer," *PAMI*, vol. 33, no. 12, pp. 2368–2382, 2011.
[6] L. Ladicky, P. Sturgess, K. Alahari, C.Russell, and P. Torr, "What, where and how many? combining object detectors and crfs," in *ECCV*, 2010, pp. 424–437.
[7] A. Singhal, J. Luo, and W. Zhu, "Probabilistic spatial context models for scene content understanding," in *CVPR*, 2003, pp. 228–235.
[8] J. Verbeek and B. Triggs, "Region classification with markov field aspect models," in *CVPR*, 2007, pp. 1–8.
[9] Z. W. Tu and X. Bai, "Auto-context and its application to high-level vision tasks and 3d brain image segmentation," *PAMI*, vol. 32, no. 10, pp. 1744–1757, 2010.
[10] Z. W. Tu, X. Chen, A. L. Yuille, and S. C. Zhu, "Image parsing: unifying segmentation, detection, and recognition," *IJCV*, vol. 63, no. 2, pp. 113–140, 2005.
[11] S. Gould, T. Gao, and D. Koller, "Region-based segmentation and object detection," in *NIPS*, 2009, pp. 655–663.
[12] E. Borenstein and S. Ullman, "Combined top-down/bottom-up segmentation," *PAMI*, vol. 30, no. 12, pp. 2109–2125, 2008.
[13] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *IJCV*, vol. 42, no. 3, pp. 145–175, 2001.
[14] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *PAMI*, vol. 35, no. 8, pp. 1915–1929, 2013.
[15] L. Yang, P. Meer, and D. Foran, "Multiple class segmentation using a unified framework over mean-shift patches," in *CVPR*, 2007, pp. 1–8.
[16] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *PAMI*, vol. 24, no. 24, pp. 509–522, 2002.
[17] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *PAMI*, vol. 6, no. 6, pp. 721–741, 1984.
[18] M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky., "Exploiting hierarchical context on a large database of object categories," in *CVPR*, 2010, pp. 129–136.
[19] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky, "Learning hierarchical models of scenes, objects, and parts," in *ICCV*, 2005, pp. 1331–1338.
[20] J. J. Lim, P. Arbeláez, C. Gu, and J. Malik, "Context by region ancestry," in *ICCV*, 2009, pp. 1978–1985.
[21] T. Mensink, J. Verbeek, and G. Csurka, "Tree-structured crf models for interactive image labeling," *PAMI*, vol. 35, no. 2, pp. 476–489, 2013.
[22] J. Verbeek and B. Triggs, "Scene segmentation with conditional random fields learned from partially labeled images," in *NIPS*, 2007, pp. 1–8.
[23] J. Tighe and S. Lazebnik, "Superparsing: scalable nonparametric image parsing with superpixels," in *ECCV*, 2010, pp. 352–365.

[24] M. P. Kumar and D. Koller, "Efficiently selecting regions for scene understanding," in *CVPR*, 2010, pp. 3217–3224.
[25] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *PAMI*, vol. 24, no. 5, pp. 603–619, 2002.
[26] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *IJCV*, vol. 59, no. 2, pp. 167–181, 2004.
[27] L. Ladicky, C. Russell, and P. Kohli, "Associative hierarchical crfs for object class image segmentation," in *ICCV*, 2009, pp. 739–746.
[28] S. Kumar and M. Hebert, "A hierarchical field framework for unified context-based classification," in *ICCV*, 2005, pp. 1284–1291.
[29] ——, "Discriminative random fields," *IJCV*, vol. 68, no. 2, pp. 179–201, 2006.
[30] C. Pantofaru, C. Schmid, and M. Hebert, "Object recognition by integrating multiple image segmentations," in *ECCV*, 2008, pp. 481–494.
[31] Q. Zhou, C. Yan, Y. Zhu, X. Bai, and W. Liu, "Image labeling by multiple segmentation," in *ICIP*, 2011, pp. 3129–3132.
[32] J. Carreira and C. Sminchisescu, "Cpmc: Automatic object segmentation using constrained parametric min-cuts," *PAMI*, vol. 34, no. 7, pp. 1312–1328, 2012.
[33] S. Gould, R. Fulton, and D. Koller, "Decomposing a scene into geometric and semantically consistent regions," in *ICCV*, 2009, pp. 1–8.
[34] D. Munoz, J. A. Bagnell, and M. Hebert, "Stacked hierarchical labeling," in *ECCV*, 2010, pp. 57–70.
[35] J. Shotton, J. M. Winn, C. Rother, and A. Criminisi, "Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *IJCV*, vol. 81, no. 1, pp. 2–23, 2009.
[36] X. M. He, R. S. Zemel, and D. Ray, "Learning and incorporating top-down cues in image segmentation," in *ECCV*, 2006, pp. 338–351.
[37] C. Galleguillos, A. Rabinovich, and S. Belongie, "Object categorization using co-occurrence, location and appearance," in *CVPR*, 2008, pp. 1–8.
[38] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller, "Multi-class segmentation with relative location prior," *IJCV*, vol. 80, no. 3, pp. 1239–1253, 2008.
[39] B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model," in *ECCV Workshop on Statistical Learning in Computer Vision*, 2004, pp. 17–32.
[40] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie, "Objects in context," in *ICCV*, 2007, pp. 1–8.
[41] J. Yao, S. Fidler, and R. Urtasun, "Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation," in *CVPR*, 2012, pp. 702–709.
[42] V. Lempitsky, A. Vedaldi, and A. Zisserman, "Pylon model for semantic segmentation," in *NIPS*, 2011.
[43] L. Zhu, Y. Chen, Y. Lin, C. Lin, and A. Yuille, "Recursive segmentation and recognition templates for image parsing," *PAMI*, vol. 34, no. 2, pp. 359–371, 2012.
[44] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *PAMI*, vol. 33, no. 5, pp. 898–916, 2011.
[45] L. Najman and M. Schmitt, "Geodesic saliency of watershed contours and hierarchical segmentation," *PAMI*, vol. 18, no. 12, pp. 1163–1173, 1996.
[46] J. Shi and J. Malik, "Normalized cuts and image segmentation," *PAMI*, vol. 22, no. 8, pp. 888–905, 2000.
[47] L. Y., S. J., T. C.K., and S. H. Y., "Lazy snapping," *ATOG*, vol. 23, no. 3, pp. 303–308, 2004.
[48] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *The annals of statistics*, vol. 28, no. 2, pp. 337–407, 2000.
[49] K. Barnard, P. Duygulu, D. Forsyth, N. De Freitas, D. M. Blei, and M. I. Jordan, "Matching words and pictures," *JMLR*, vol. 3, pp. 1107–1135, 2003.
[50] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *IJCV*, vol. 73, no. 2, pp. 213–238, 2007.
[51] A. Oliva and A. Torralba, "The role of context in object recognition," *Trends in Cognitive Sciences*, vol. 11, no. 12, pp. 520–527, 2007.
[52] B. Yao, X. Yang, and S. C. Zhu, "Introduction to a large scale general purpose groundtruth dataset: Methodology, annotation tool, and benchmarks," in *EMMCVPR*, 2007.
[53] P. Krahenbuhl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *NIPS*, 2011, pp. 109–117.
[54] D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.
[55] P. Felzenszwalb and D. Huttenlocher, "Pictorial structures for object recognition," *IJCV*, vol. 61, no. 1, pp. 55–79, 2005.