Task-Specific Image Partitioning

Sungwoong Kim*, *Member, IEEE,* Sebastian Nowozin, Pushmeet Kohli, and Chang D. Yoo, *Senior Member, IEEE*

***Abstract*—Image partitioning is an important preprocessing step for many of the state-of-the-art algorithms used for performing high-level computer vision tasks. Typically, partitioning is conducted without regards to the task in hand. We propose a task-specific image partitioning framework to produce a region-based image representation that will lead to a higher task performance than that reached using any task-oblivious partitioning framework and existing supervised partitioning framework, albeit few in number. The proposed method partitions the image by means of correlation clustering, maximizing a linear discriminant function defined over a superpixel graph. The parameters of the discriminant function that define task-specific similarity/dissimilarity among superpixels are estimated based on structured support vector machine (S-SVM) using task-specific training data. The S-SVM learning leads to a better generalization ability while the construction of the superpixel graph used to define the discriminant function allows a rich set of features to be incorporated to improve discriminability and robustness. We evaluate the learnt task-aware partitioning algorithms on three benchmark datasets. Results show that task-aware partitioning leads to better labeling performance than the partitioning computed by the state-of-the-art general-purpose and supervised partitioning algorithms. We believe that the task-specific image partitioning paradigm is widely applicable to improve the performance in high-level image understanding tasks.**

EDICS Category: ARS-RBS, ARS-IIU

I. Introduction

Region-based image representations (RBIRs) have been shown to be effective in improving the performance of algorithms for high-level image/scene understanding, which encompasses tasks such as object class segmentation, scene segmentation, surface layout labeling, and single view 3D reconstruction [1]–[5]. The effectiveness comes as a result of promoting the following three merits of using the RBIRs. *First*, the coherent support of a region, commonly assumed to be of a single label, serves as a good prior for many labeling tasks. *Second*, these coherent regions allow a more consistent feature extraction that can incorporate surrounding contextual information by pooling many feature responses over the region. *Third*, compared to pixels, a small number of larger homogeneous regions can significantly reduces the computational cost in the successive labeling task. In this paper, we propose an image partitioning framework for obtaining RBIRs that realizes these benefits and improves the task-specific labeling performance.

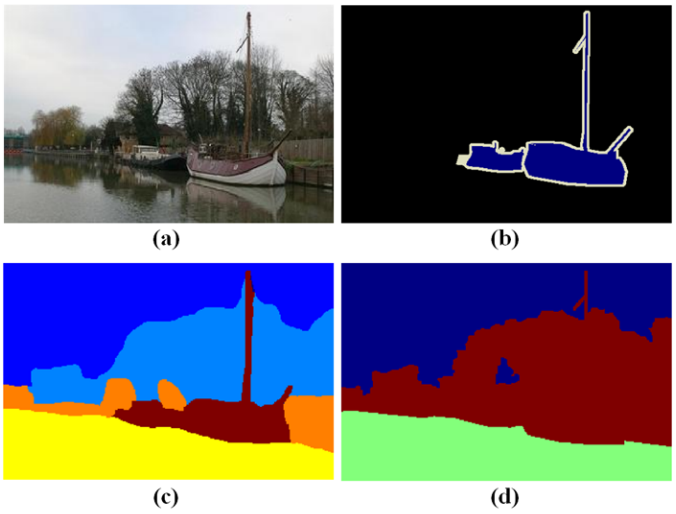


Fig. 1. Example of different ground-truth (GT) labelings for the same image according to different tasks. (a) Original image. (b) The GT on the object class segmentation. (c) The GT on the semantic scene segmentation. (d) The GT on the surface layout labeling.

Up until now, using RBIRs in an image labeling system is a two-stage process: first, a general-purpose image partitioning method is used to obtain the RBIR, and second, this RBIR is used by a model that is trained with task-specific supervision. The first stage is task-oblivious but has direct influence on the performance of the model in the second stage. Ignoring the task at hand during creation of the RBIR is therefore a limitation of current systems that we address. For instance, consider Fig. 1 that shows ground-truth labelings for different labeling tasks for a given image. The ideal partitioning for the object-specific segmentation task would group each boat into one region and the remaining part into one background [6]. For the task of semantic scene segmentation [2], [3], the preference is to segment each distinct class – sky, tree, water, boat – into a region of its own. In surface layout labeling which does not distinguish between object classes, the preference is to segment the image into regions of coherent surface normals [2], [4].

From the example explained above, it is obvious that a task-specific image partitioning algorithm would lead to partitioning that would be more conducive to the particular labeling

Manuscript received September 10, 2011; revised July 6, 2012; accepted August 19, 2012. This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST) (No.2012-0005378 and No.2012-0000985).

S. Kim is with Qualcomm Research Korea, 15th FL., POBA Gangnam Tower, 119 Nonhyeon-dong, Gangnam-gu, Seoul, South Korea (Phone + 82-2-530-6984, Fax + 82-2-530-6996, E-mail: sungwoong.kim01@gmail.com)

S. Nowozin and P. Kohli are with Microsoft Research Cambridge, (E-mail: Sebastian.Nowozin@microsoft.com; pkohli@microsoft.com)

C. Yoo is with Department of EE, Korea Advanced Institute of Science and Technology, 373-1 Guseong Dong, Yuseong Gu, Daejeon 305-701, South Korea (Phone + 82-42-350-5470, Fax + 82-42-350-8590, E-mail: cdyoo@ee.kaist.ac.kr)

task in hand than the general-purpose partitioning algorithm. With this goal in mind, we explicitly address the *task-specific image partitioning problem* as follows: given an image and labeling task, produce a partitioning of the image into disjoint regions such that each region is homogeneous with respect to the desired labeling of the task, and the labels of its neighboring regions are different. Note that this is different from image labeling in that we aim to produce a partitioning without region-labeling. For example, for the object-specific class segmentation, the object class does not have to be known during creating the partitioning of the image.

There are several general-purpose unsupervised image partitioning algorithms for region-based image understanding. For instance, in the superpixel-based conditional random fields (CRFs) models [5], [7], [8], mean-shift [9], normalized cuts [10], graph-based local variation algorithm [11], and their variants such as quick-shift [12] are used to obtain small coherent image regions, called superpixels. These *a priori* over-segmentations are not related to any task and maybe limited in capturing accurate global information for the successive region-labeling step. To enhance its ability, some recent CRFs are based on either a hierarchy of regions [1], [13] or a set of partitionings [2]–[4]. These multiple partitionings are obtained using mean-shift segmentation with different kernel sizes, multiscale normalized cuts [14], a hierarchical segmentation with increasing edge strength [13], and a simple region-merging algorithm [4]. These algorithms – while empirically successful to a certain extent – use task-oblivious partitionings and therefore do not address the task-specific image partitioning problem.

In this paper, we address the *task-specific image partitioning* problem using correlation clustering [15] which is a graph-partitioning algorithm that simultaneously maximizes intra-cluster similarity and inter-cluster dissimilarity. Here, the similarity and dissimilarity must be defined differently according to the task, and this is achieved by learning parameters using task-specific training data. Since correlation clustering assigns a label to each edge, in contrast to other image partitioning algorithms, correlation clustering does not require a pre-specified number of clusters and distance threshold for clustering. Furthermore, correlation clustering leads to linear discriminant functions which allow for large margin training based on structured support vector machine (S-SVM) [16]. Within our proposed framework, we learn optimal task-specific parameters for partitioning using the task-specific ground-truth partitionings of the training data; the proposed task-specific image partitioning is a supervised image partitioning in which a homogeneous region is determined by the desired labeling of the task.

Although a number of supervised learning algorithms for graph-based image partitioning such as supervised spectral clustering and pairwise affinity learning [17]–[21] have been proposed, none have exploited the use of task-specific training data to produce partitioning that is substantially more beneficial than unsupervised partitioning in terms of addressing the task-specific image partitioning problem mentioned above. Furthermore, these supervised image partitioning algorithms suffer from a number of problems in learning task-specific partitionings. *First*, inference is slow and difficult especially with increasing graph size, which restricts experimentation to small data sets. *Second*, the learning criterion does not take into account inference on the full graph; instead, is is based on a local cost by treating each pairwise relations between adjacent nodes as independent samples and sometimes requires a complex and unstable eigenvector approximation which must be differentiable. *Third*, the learning criterion is based on either the maximum likelihood or the minimum square error and leads to generalization problems on unseen data when the number of parameters is relatively large in comparison to the number of training data. The proposed correlation clustering for task-specific image partitioning, on the other hand, overcomes all of these problems.

A framework that uses the S-SVM for training the parameters in correlation clustering has been considered previously by Finley *et al.* [22]; however, the framework was used for noun-phrase and news article clusterings. Taskar derived a max-margin formulation for learning the edge scores for correlation clustering [23] without experimental or quantitative results. This learning criterion is different from the S-SVM and is limited to applications involving two different segmentations of a single image.

The proposed correlation clustering algorithm starts from a fine superpixel graph to reduce computational cost and also to extract a more meaningful discriminative features from larger consistent regions. To start with a fine superpixelization is typically not a limitation in practice as the number of fine superpixels is much larger (hundreds) than the final number of regions (tens). A rich pairwise feature vector on neighboring superpixels based on several visual cues is defined, and the correlation clustering problem is approximately solved using a linear programming (LP) relaxation technique. Correlation clustering is in general NP-hard, and therefore, the relaxation provides a polynomial-time approximation to its maximum a posteriori (MAP) solution. Moreover, recent research suggests that relaxations can be favorable within max-margin learning procedures [24]–[26]. For supervised training of the parameter vector, we apply a decomposable structured loss function to handle imbalanced classes. We incorporate this loss function into the cutting plane procedure for S-SVM training [16]. We will show by means of experimental results on various datasets that the proposed correlation clustering outperforms other state-of-the-art image partitioning algorithms owing to the task-specific image partitioning.

To summarize, our main contributions are: (1) a study on task-specific image partitioning that is suitable for any particular labeling problem at hand, (2) a supervised correlation clustering on a superpixel graph for task-specific image partitioning; a rich feature vector is taken for robust partitioning, the LP relaxation is used for fast inference, and the SSVM with a modified label loss is used for task-specific training of the parameter vector, and (3) an empirical validation of the proposed task-specific image partitioning that is more conducive to the successive labeling task in comparison to existing state-of-the-art partitioning algorithms.

The rest of the paper is organized as follows. Section II presents the proposed correlation clustering for image

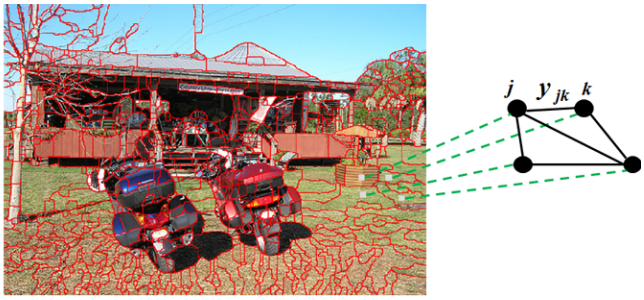


Fig. 2. Illustration of a part of the graph built on superpixels.

partitioning. Section III describes large margin training for task-specific image partitioning based on the S-SVM and the cutting plane algorithm. A number of experimental and comparative results are presented and discussed in Section IV, followed by a conclusion in Section V.

II. Correlation Clustering for Image Partitioning

The proposed image partitioning is based on superpixels, which can significantly reduce computational cost and allow feature extraction to be conducted from a larger homogeneous region. As shown in Fig. 2, superpixels preserve almost all boundaries between different regions, independent of the task. The proposed correlation clustering merges superpixels into disjoint regions of homogeneity over a superpixel graph.

A. Correlation Clustering over Superpixel Graph

Define an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where a node corresponds to a superpixel and a link between adjacent superpixels corresponds to an edge (see Fig. 2). A binary label y_{jk} for an edge $(j,k) \in \mathcal{E}$ between nodes j and k is defined such that

$$y_{jk} = \begin{cases} 1, & \text{if nodes } j \text{ and } k \text{ belong to the same region,} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

A discriminant function, which is the negative energy function, is defined over image $\mathbf{x}$ and all edge labels $\mathbf{y}$ as

$$\begin{aligned} F(\mathbf{x},\mathbf{y};\mathbf{w}) &= \sum_{(j,k)\in\mathcal{E}} \text{Sim}_{\mathbf{w}}(\mathbf{x},j,k) y_{jk} \\ &= \sum_{(j,k)\in\mathcal{E}} \langle \mathbf{w}, \phi_{jk}(\mathbf{x}) \rangle y_{jk} \\ &= \langle \mathbf{w}, \sum_{(j,k)\in\mathcal{E}} \phi_{jk}(\mathbf{x}) y_{jk} \rangle = \langle \mathbf{w}, \Phi(\mathbf{x},\mathbf{y}) \rangle, \quad (2) \end{aligned}$$

where the similarity measure between nodes j and k, $\text{Sim}_{\mathbf{w}}(\mathbf{x},j,k)$, is parameterized by $\mathbf{w}$ and takes values of both signs such that a large positive value means strong similarity while a large negative value means high degree of dissimilarity. Note that the discriminant function $F(\mathbf{x},\mathbf{y};\mathbf{w})$ is assumed to be linear in both the parameter vector $\mathbf{w}$ and the joint feature map $\Phi(\mathbf{x},\mathbf{y})$, and $\phi_{jk}(\mathbf{x})$ is a pairwise feature vector which reflects the correspondence between the jth and the kth superpixels. An image segmentation is to infer the edge label, $\hat{\mathbf{y}}$, over the pairwise superpixel graph $\mathcal{G}$ by maximizing F such that

$$\hat{\mathbf{y}} = \underset{\mathbf{y}\in\mathcal{Y}(\mathcal{G})}{\text{argmax}}\, F(\mathbf{x},\mathbf{y};\mathbf{w}), \quad (3)$$

where $\mathcal{Y}(\mathcal{G})$ is the set of $\{0,1\}^{\mathcal{E}}$ that corresponds to a *valid segmentation*. This set is the set of multicuts [27]. However, solving (3) with this $\mathcal{Y}(\mathcal{G})$ is generally NP-hard. Therefore, we approximate $\mathcal{Y}(\mathcal{G})$ by means of a common multicut LP relaxation [27], [28] with the following two constraints: (1) cycle inequality and (2) odd-wheel inequality. Indeed, the LP relaxation to approximately solve (3) can be formulated as

$$\begin{aligned} \underset{\mathbf{y}}{\text{argmax}} \quad & \sum_{(j,k)\in\mathcal{E}} \langle \mathbf{w}, \phi_{jk}(\mathbf{x}) \rangle y_{jk} \quad (4) \\ \text{s.t.} \quad & \mathbf{y} \in \mathcal{Z}(\mathcal{G}), \end{aligned}$$

where $\mathcal{Z}(\mathcal{G}) \supset \mathcal{Y}(\mathcal{G})$ is the relaxed polytope defined by the two LP constraints.

1) Cycle inequality: Let $\text{Path}(j,k)$ be the set of paths between nodes j and k. The cycle inequalities, which are generalizations of the triangle inequality [27], are defined as

$$(1 - y_{jk}) \le \sum_{(s,t)\in p} (1 - y_{st}), \quad p \in \text{Path}(j,k). \quad (5)$$

2) Odd-wheel inequality: Let a q-wheel be a connected subgraph $\mathcal{S} = (\mathcal{V}_s, \mathcal{E}_s)$ with a central vertex $j \in \mathcal{V}_s$ and a cycle of the q vertices in $\mathcal{C} = \mathcal{V}_s \setminus \{j\}$. For every odd $q(\ge 3)$-wheel, a valid partitioning $\mathbf{y}$ satisfies

$$\sum_{(s,t)\in\mathcal{E}(\mathcal{C})} (1 - y_{st}) - \sum_{k\in\mathcal{C}} (1 - y_{jk}) \le \lfloor \frac{1}{2} q \rfloor, \quad (6)$$

where $\mathcal{E}(\mathcal{C})$ denotes the set of all edges in the outer cycle $\mathcal{C}$.

Note that the cycle inequalities and odd-wheel inequalities are separable (possible to seek a violated inequality) in polynomial time, which is important to solve the LP relaxation in polynomial time.

The relation between the solution to (3) and the solution to (4) is as follows: if the LP solution to (4) is integral, that is for all $(j,k) \in \mathcal{E}$ we have $y_{jk} \in \{0,1\}$, then the solution $\mathbf{y}$ is the exact solution to (3). If instead it is fractional, then our floor-rounding provides a feasible but potentially sub-optimal solution to (3).

B. Pairwise Feature Vector

We construct a rich pairwise feature vector based on different quantization levels and thresholds. The magnitude of $\mathbf{w}$ determines the importance of each feature, and this importance is task-dependent. Here, $\mathbf{w}$ is estimated by supervised training described in Section III.

We extract several visual cues from a superpixel, including brightness (intensity), color, texture, and shape. Based on these visual cues, we construct a 321-dimensional pairwise feature vector ϕ by concatenating a color difference feature ϕ^c, texture difference feature ϕ^t, shape/location difference feature ϕ^s,

edge strength feature ϕ^e, joint visual word posterior feature ϕ^v, and bias ϕ^b as follows:

$$\phi_{jk}(\mathbf{x}) = [\phi^c_{jk}(\mathbf{x}); \phi^t_{jk}(\mathbf{x}); \phi^s_{jk}(\mathbf{x}); \phi^e_{jk}(\mathbf{x}); \phi^v_{jk}(\mathbf{x}); \phi^b_{jk}(\mathbf{x})]. \quad (7)$$

- Color difference feature ϕ^c: The color difference feature ϕ^c is composed of 26 color distances between two adjacent superpixels based on RGB and HSV channels. Specifically, we calculate 18 earth mover's distances (EMDs) [29] between two color histograms extracted from each superpixel with various numbers of bins and thresholds for ground distance. In addition, six absolute differences (one for each color channel) between the means of the two superpixels and two χ^2-distances between hue/saturation histograms of the two superpixels are concatenated in ϕ^c.
- Texture difference feature ϕ^t: The 64-dimensional texture difference feature ϕ^t is composed of 15 absolute differences (one for each texture-response) between the means of two superpixels using 15 Leung-Malik (LM) filter banks [30] and one χ^2-distance and 48 EMDs (from various numbers of bins and thresholds for ground distance) between texture histograms of the two superpixels.
- Shape/location difference feature ϕ^s: The 5-dimensional shape/location difference feature ϕ^s is composed of two absolute differences between the normalized (x/y) center positions of the two superpixels, the ratio of the size of the smaller superpixel to that of the larger superpixel, the percentage of boundary with respect to the smaller superpixel, and the straightness of boundary [4].
- Edge strength feature ϕ^e: The 15-dimensional edge strength feature ϕ^e is a 1-of-15 coding of the quantized edge strength proposed by Arbelaez *et al.* [13].
- Joint visual word posterior feature ϕ^v: The 210-dimensional joint visual word posterior feature ϕ^v is defined as the vector holding the joint visual word posteriors for a pair of neighboring superpixels using 20 visual words [31] as follows.
  First, a 52-dimensional raw feature vector x_j based on color, texture, location, and shape features described in [4] is extracted from the jth superpixel. Then, the visual word posterior distribution $P(v_i|x_j)$ is computed using the Gaussian RBF kernel where v_i denotes the ith visual word. Let $V_{jk}(\mathbf{x})$ be a 20-by-20 matrix whose elements are the joint visual word posteriors between nodes j and k defined such that

$$V_{jk}(\mathbf{x}) = \begin{bmatrix} P(v_1|x_j)P(v_1|x_k) & \cdots & P(v_1|x_j)P(v_{20}|x_k) \\ P(v_2|x_j)P(v_1|x_k) & \cdots & P(v_2|x_j)P(v_{20}|x_k) \\ \vdots & \ddots & \vdots \\ P(v_{20}|x_j)P(v_1|x_k) & \cdots & P(v_{20}|x_j)P(v_{20}|x_k) \end{bmatrix}. \quad (8)$$

  The joint visual word posterior feature between nodes j and k, $\phi^v_{jk}(\mathbf{x})$, is defined as

$$\phi^v_{jk}(\mathbf{x}) = \mathrm{vec}\Big(V_{jk}(\mathbf{x})\Big) + \mathrm{vec}\Big(V^T_{jk}(\mathbf{x})\Big), \quad (9)$$

  where $\mathrm{vec}(V)$ be the $210 (= 20 \times 21/2)$-dimensional vector whose elements are from the upper triangular part of V.

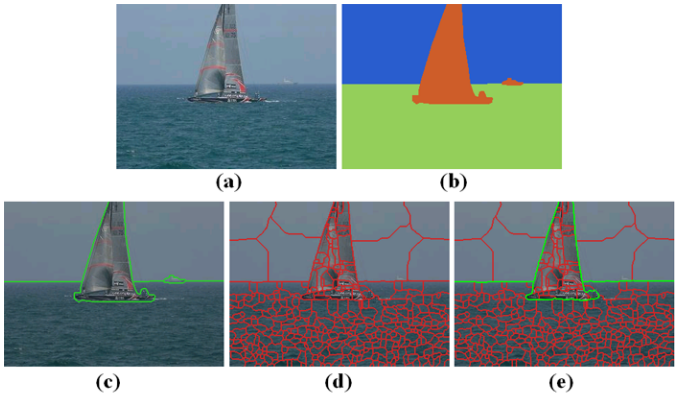


Fig. 3. Example of ground-truth (GT) edge labels on the superpixels from the GT task labels. (a) Original image. (b) GT task labels on the semantic scene segmentation. (c) GT partitioning from the GT task labels. (d) Superpixels. (e) GT edge labels on the superpixels (green: edge label is 0, red: edge label is 1).

  This joint visual word posterior feature could overcome the weakness of class-agnostic features and incorporate the contextual information.
- Bias ϕ^b: We augment the bias $\phi^b = 1$ for a proper similarity measure which can be either positive or negative.

III. Supervised Training for Task-Specific Partitioning

For task-specific image partitioning, the parameter vector $\mathbf{w}$ is estimated from the training data for each task. The proposed discriminant function is defined over the superpixel graph, and therefore, the ground-truth task labels of the pixels need to be transformed to the ground-truth edge labels of the superpixel graph. Note that different from the ground-truth edge-labeling over the superpixel graph, the ground-truth partitioning is directly defined by the ground-truth task labels as illustrated in Fig. 3. First, we assign a single dominant task label to each superpixel by majority voting over the superpixel's constituent pixels and then obtain the ground-truth edge labels on the superpixel graph according to whether dominant labels of neighboring superpixels are equal or not (see Fig. 3).

Using this ground-truth edge labels of the training data, we use the S-SVM to estimate the parameter vector for task-specific correlation clustering. We use the cutting plane algorithm with LP relaxation (4) for loss-augmented inference is used to solve the optimization problem of the S-SVM, since fast convergence and high robustness of the cutting plane algorithm in handling a large number of margin constraints are well-known [16].

A. Structured Support Vector Machine

Given N training samples $\{(\mathbf{x}^n, \mathbf{y}^n)\}_{n=1}^N$ where $\mathbf{y}^n$ is the ground-truth edge labels for the nth training image, the S-SVM [16] optimizes $\mathbf{w}$ by minimizing a quadratic objective

function subject to a set of linear margin constraints:

$$\min_{\mathbf{w},\xi} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n \tag{10}$$

$$\text{s.t. } \langle \mathbf{w}, \delta\Phi(\mathbf{x}^n, \mathbf{y})\rangle \geq \Delta(\mathbf{y}^n, \mathbf{y}) - \xi_n, \ \forall n, \mathbf{y} \in \mathcal{Z}(\mathcal{G})\backslash\mathbf{y}^n, \tag{11}$$

$$\xi_n \geq 0, \quad \forall n, \tag{12}$$

where $\delta\Phi(\mathbf{x}^n, \mathbf{y}) = \Phi(\mathbf{x}^n, \mathbf{y}^n) - \Phi(\mathbf{x}^n, \mathbf{y})$, and $C > 0$ is a constant that controls the trade-off between margin maximization and training error minimization. In the S-SVM, the margin is scaled with a loss $\Delta(\mathbf{y}^n, \mathbf{y})$, which is the difference measure between prediction $\mathbf{y}$ and ground-truth label $\mathbf{y}^n$ of the nth image. The S-SVM offers good generalization ability as well as the flexibility to choose any loss function [16].

B. Cutting Plane Algorithm

The exponentially large number of margin constraints (11) and the intractability of the loss-augmented inference problem make it difficult to solve the constrained optimization problem of (10). Therefore, we apply the cutting plane algorithm [16], [28], also known as the column generation algorithm, to approximately solve the constrained optimization problem. The cutting plane algorithm is summarized in Algorithm 1. In each iteration, the most violated constraint for each training sample is approximately found by performing the loss-augmented inference using the LP relaxation. The computational cost for inference can be greatly reduced when a decomposable loss such as the Hamming loss is used; if the loss function is decomposed in the same manner as the joint feature map, we can add the loss function to each edge score in the inference. We then check if the constraint found tightens the feasible set of (10), and if it does, then the parameter vector $\mathbf{w}$ and ξ are updated by solving the restricted problem of (10) on the current set of active constraints that includes it. The theoretical convergence and robustness of the cutting plane algorithm was studied by Tsochantaridis *et al.* [16]. The LP relaxations for loss-augmented inferences are considered to be well suited to structured learning [24]–[26].

C. Label Loss

A loss function $\Delta : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}_+$ is defined as a non-negative function satisfying the following properties for all n,

$$\begin{cases} \Delta(\mathbf{y}^n, \mathbf{y}) > 0, & \text{if } \mathbf{y} \neq \mathbf{y}^n, \\ \Delta(\mathbf{y}^n, \mathbf{y}) = 0, & \text{if } \mathbf{y} = \mathbf{y}^n. \end{cases} \tag{13}$$

A loss function should be decomposable to effectively perform loss-augmented inference in the cutting plane algorithm. The most popular decomposable loss function is the Hamming distance which is equivalent to the number of mismatches between $\mathbf{y}^n$ and $\mathbf{y}$. Unfortunately, the number of edges with label 1 in the proposed correlation clustering is considerably higher than that of edges with label 0 (see Fig. 3). This imbalance makes other learning methods such as the perceptron algorithm inappropriate, since it leads to the clustering of the whole image as one segment. This imbalance occurs when we use the Hamming loss in the S-SVM; therefore, we use the following adjusted loss function:

$$\begin{aligned} \Delta(\mathbf{y}^n, \mathbf{y}) &= \sum_{(j,k)\in\mathcal{E}} \Delta_{jk}(y^n_{jk}, y_{jk}) \\ &= \sum_{(j,k)\in\mathcal{E}} R\ y^n_{jk} + y_{jk} - (R+1)y^n_{jk}y_{jk} \end{aligned} \tag{14}$$

where Δ_{jk} is the label loss on the edge between nodes j and k, and R is the relative weight of the false negative to that of the false positive[1]. Note that the additive decomposition of the loss allows us to cast the loss into the additive edge score when performing the loss-augmented inference. Moreover, R controls the relative importance between the incorrect merging of the superpixels and the incorrect separation of the superpixels by imposing different weights to the false negative and the false positive, as shown in Table I. Here, we set R to be less than 1 to overcome the problem due to the imbalance. This loss is similar to the loss proposed by Cour *et al.* [32], however, the proposed loss is appropriate for fractionally-predicted labels during LP-relaxed inference while their loss is appropriate for only integer solutions.

Algorithm 1 Cutting Plane Algorithm

Choose: $\mathbf{w}_0$, C, R, ϵ
$S_n \leftarrow \emptyset, \quad \forall n, \quad \mathbf{w} \leftarrow \mathbf{w}_0, \quad \xi \leftarrow \mathbf{0}$
repeat
  for $n = 1, ..., N$ **do**
    Perform the loss-augmented inference by LP relaxation:

$$\hat{\mathbf{y}}^n = \underset{\mathbf{y}\in\mathcal{Z}(\mathcal{G})}{\operatorname{argmax}} \Big(\langle \mathbf{w}, \Phi(\mathbf{x}^n, \mathbf{y})\rangle + \Delta(\mathbf{y}^n, \mathbf{y}) \Big)$$

    if $-\langle \mathbf{w}, \delta\Phi(\mathbf{x}^n, \hat{\mathbf{y}}^n)\rangle + \Delta(\mathbf{y}^n, \hat{\mathbf{y}}^n) > \xi_n + \epsilon$ **then**
      $S_n \leftarrow S_n \cup \{\hat{\mathbf{y}}^n\}$
    end if
  end for
  Solve the restricted problem of (10) on the current set of constraints:

$$(\mathbf{w}^*, \xi^*) = \underset{\mathbf{w}',\xi'}{\operatorname{argmin}} \quad \frac{1}{2}\|\mathbf{w}'\|^2 + C\sum_{n=1}^{N}\xi'_n$$

$$\text{s.t.} \quad \langle \mathbf{w}', \delta\Phi(\mathbf{x}^n, \mathbf{y})\rangle \geq \Delta(\mathbf{y}^n, \mathbf{y}) - \xi'_n, \ \forall n, \mathbf{y} \in S_n,$$
$$\xi'_n \geq 0, \quad \forall n$$

  Update: $\mathbf{w} \leftarrow \mathbf{w}^*, \quad \xi \leftarrow \xi^*$
until no S_n has changed

TABLE I
LABEL LOSS AT THE EDGE LEVEL.

y^n_{jk}	0	1	0	1
y_{jk}	0	1	1	0
Δ_{jk}	0	0	1	R

[1]Here, the positive label refers to the label assigned as 1 while the negative label refers to the other.

IV. EXPERIMENTS

The purpose of the following experiments is to demonstrate for various labeling tasks the proposed task-specific partitioning can lead to higher task performance than that reached using task-oblivious partitioning. For this purpose, we conducted image partitionings for two tasks: semantic scene segmentation and surface layout labeling.

For task-specific image partitioning based on correlation clustering, we initially obtain baseline superpixels (an average of 367 superpixels per image) by the *gPb* contour detector and the oriented watershed transform (gPb-owt) [13] and then construct a superpixel graph with pairwise feature vectors. On both tasks, the function parameters are initially set to zero, and then based on the S-SVM, the structured output learning is used to estimate the parameter vectors. Note that the relaxed solutions in loss-augmented inference are used during training, while in testing, as described in Section II, our simple rounding method is used to produce valid partitioning results. Rounding is only necessary in case we obtain fractional solutions from LP-relaxed correlation clustering.

We compared the proposed task-specific correlation clustering to the following three unsupervised image partitioning algorithms and two supervised image partitioning algorithms:

- Mean-shift: Comaniciu and Meer [9] devised the mean-shift algorithm that is a mode-seeking algorithm to locate points of locally-maximal density in feature space.
- Multiscale NCut: Cour *et al.* [14] devised a multiscale spectral image partitioning algorithm by decomposing an image partitioning graph into different scales in the normalized cut framework.
- gPb-owt-ucm: The oriented watershed transform - ultrametric contour map algorithm [13] produced hierarchical regions using the gPb contour detector as input.
- gPb-Hoiem: Hoiem *et al.* [4] grouped superpixels based on pairwise same-label likelihoods. The superpixels were obtained by the same gPb contour detector, and the pairwise same-label likelihoods based on the same 321-dimensional pairwise feature vector were independently learnt from the task-specific training data.
- Supervised NCut: We applied a supervised learning algorithm for parameter estimation under the normalized cut framework. For this, first, the affinity matrix on the same pairwise superpixel graph was defined as

$$A_{jk} = \begin{cases} \min(1, \exp\{-\langle \mathbf{w}, \phi_{jk} \rangle\}), & \text{if } (j,k) \in \mathcal{E}, \\ 0, & \text{otherwise,} \end{cases}$$

  where the same 321-dimensional pairwise feature vector ϕ_{jk} was used. Then, the standard pairwise affinity learning with the square-square loss function and the gradient descent algorithm [21] was used for task-specific training.

To quantitatively evaluate partitionings obtained by various algorithms against the ground-truth partitioning that is associated with the ground-truth task labels (see Fig. 3.(c)), we consider four performance measures: *Probabilistic Rand index* (PRI) [33], *segmentation covering* (SCO) [13], *variation of information* (VOI) [34], and *boundary displacement error* (BDE) [35]. As the predicted partitioning is close to the ground-truth partitioning, the PRI and SCO are increased while the VOI and BDE are decreased. The performance varies with different numbers of regions, and for this reason, we designed each algorithm to produce multiple partitionings (10 to 40 regions). For example, when using the codes publicly released by the authors for (multiscale) NCut and gPb-Hoiem, we explicitly set the number of regions as an input parameter or while for the mean-shift and gPb-owt-ucm, we applied different kernel bandwidths and level-thresholds in a hierarchy of regions to produce multiple partitionings. Specifically, multiple partitionings in the proposed algorithm were obtained by varying R from 0.005 to 0.2 in the loss function during training. As R increases, the number of partitioned regions of a test image tends to decrease, since the false negative error is penalized more compared to the false positive error. However, in testing, in contrast to other algorithms which pre-fix the number of regions or threshold of level in a hierarchy of regions equally across all images, the proposed correlation clustering automatically determines the proper number of partitioned regions in each image. To perform image partitioning for the tasks of semantic scene segmentation and surface layout labeling, we used the Stanford background dataset [2], which consists of 715 outdoor images with corresponding pixel-wise annotations. We employed 5-fold cross-validation with the dataset randomly split into 572 training images and 143 test images for each fold.

Image partitioning performances. The goal of semantic scene segmentation is to generate pixel-wise segmentations such that each pixel is labeled with either one of 7 background classes or a generic foreground class. From the given pixel-wise ground-truth annotations, we obtain ground-truth task-specific partitionings for each image. We train our proposed task-specific correlation clustering algorithm, the gPb-Hoiem, and the supervised NCut on the training set and compare all image partitioning algorithms on the separate test set. Fig. 4.(a) shows the obtained four measures from partitioning results according to the average number of regions. The proposed task-specific image partitioning for the task of semantic scene segmentation (Corr-Cluster-Semantic) performed better than other algorithms. Especially, in producing image partitioning for the task of semantic scene segmentation, correlation clustering using the parameters learnt for the task of surface layout labeling (Corr-Cluster-Geometric) gave worse results than those obtained by Corr-Cluster-Semantic. These results show improvements obtained by the task-specific training of parameters within the proposed framework.

The task of surface layout labeling is to label each pixel in an image into one of three geometric classes (horizontal, vertical, and sky). As shown in Fig. 4.(b), the proposed task-specific image partitioning (Corr-Cluster-Geometric) achieved the best results.

Compared to previous supervised image partitioning algorithms, the proposed task-specific correlation clustering algorithm enables easier construction of rich pairwise feature vectors from visual cues, and it can scalably estimate high-

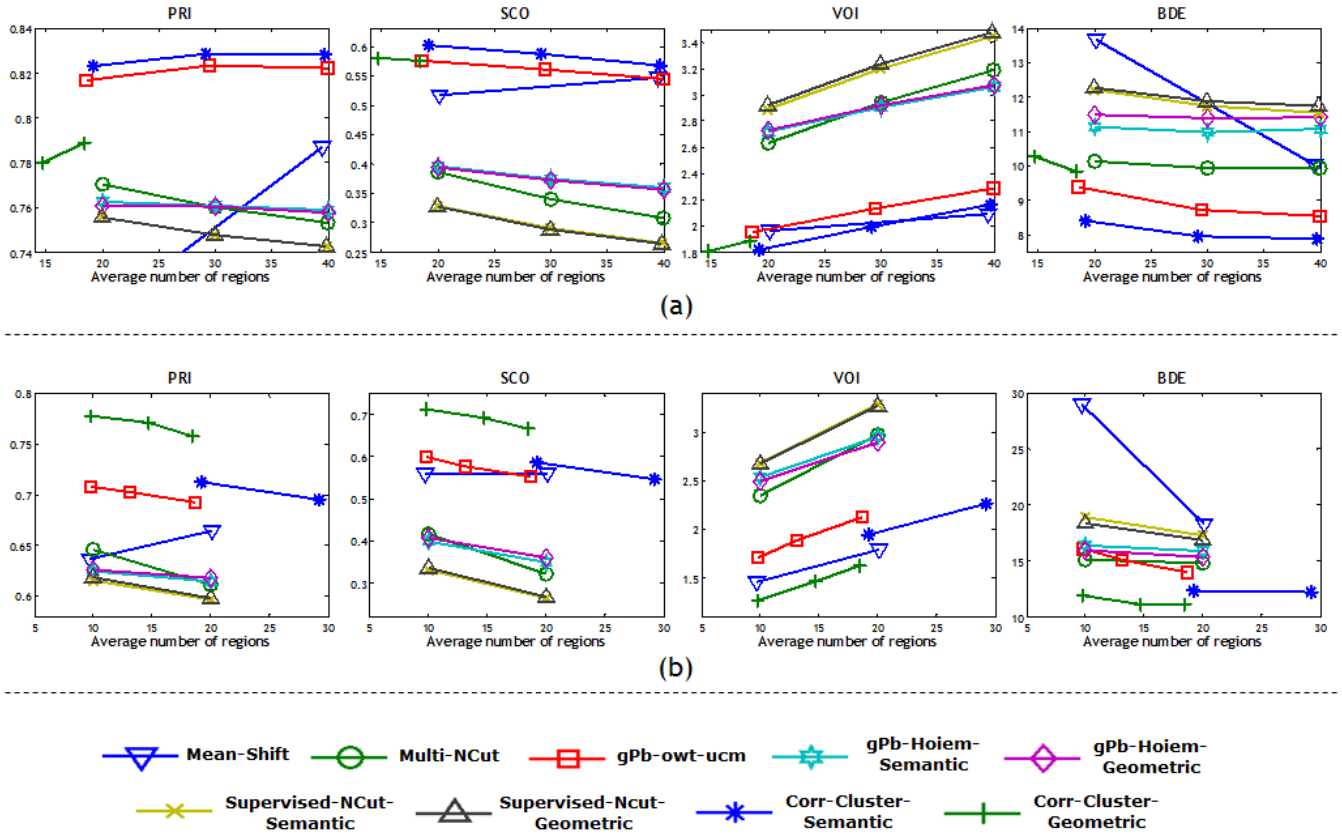


Fig. 4. Obtained evaluation measures from partitioning results on the test set for semantic scene segmentation (a) and surface layout labeling (b).

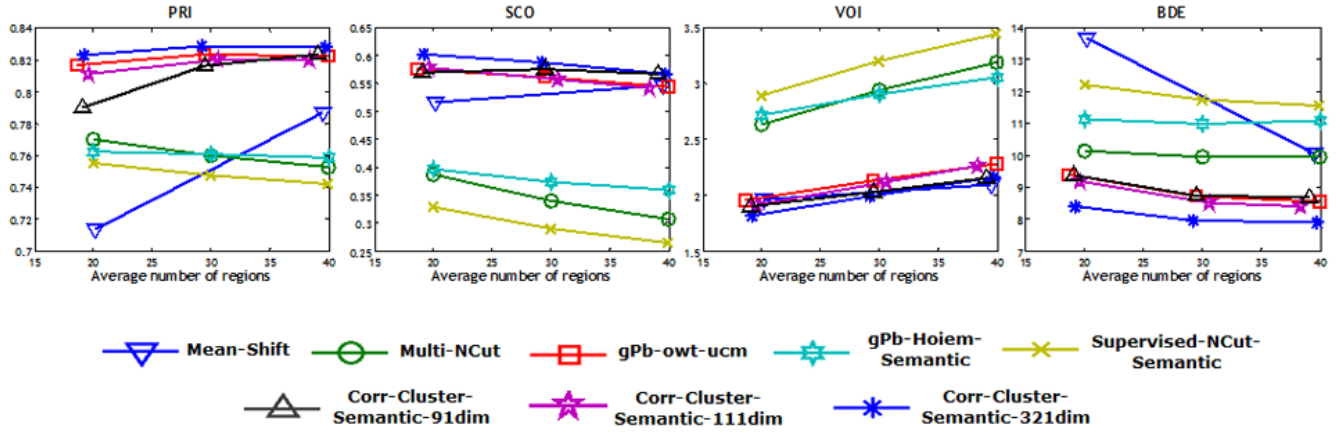


Fig. 5. Obtained evaluation measures from partitioning results according to the different set of features on the test set for semantic scene segmentation.

dimensional feature weight vectors. To evaluate the effectiveness of various feature sets within the proposed framework, we obtained evaluation measures for different set of features:

1) Color difference + Texture difference (91-dim):

$$\phi_{jk} = [\phi^c_{jk};\ \phi^t_{jk};\ \phi^b_{jk}]. \tag{15}$$

2) Color difference + Texture difference + Shape/location difference + Edge strength (111-dim):

$$\phi_{jk} = [\phi^c_{jk};\ \phi^t_{jk};\ \phi^s_{jk};\ \phi^e_{jk};\ \phi^b_{jk}]. \tag{16}$$

3) Color difference + Texture difference + Shape/location difference + Edge strength + Joint visual word posterior (321-dim):

$$\phi_{jk} = [\phi^c_{jk};\ \phi^t_{jk};\ \phi^s_{jk};\ \phi^e_{jk};\ \phi^v_{jk};\ \phi^b_{jk}]. \tag{17}$$

As shown in Fig. 5, the color and texture features influenced the result most followed by shape/location and edge strength features. Including the joint visual word posterior feature to the color and texture features improved the performance significantly.

The proposed correlation clustering is based on the superpixel graph. Therefore, performances might be influenced by baseline superpixels. Fig. 7 shows the performance dependency of the proposed correlation clustering algorithm on

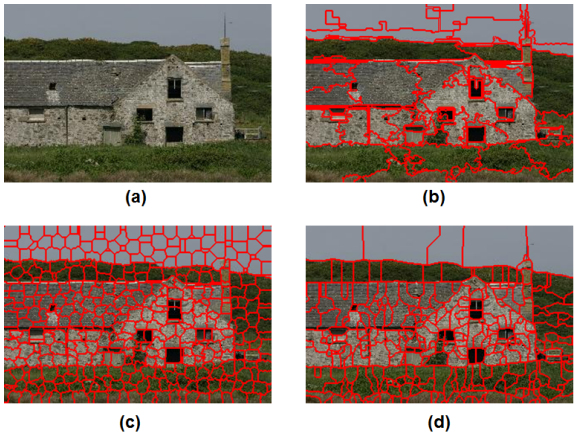


Fig. 6. Examples of different baseline superpixels. (a) Original image. (b) Superpixels obtained by FH. (c) Superpixels obtained by GC. (d) Superpixels obtained by (gPb-owt).

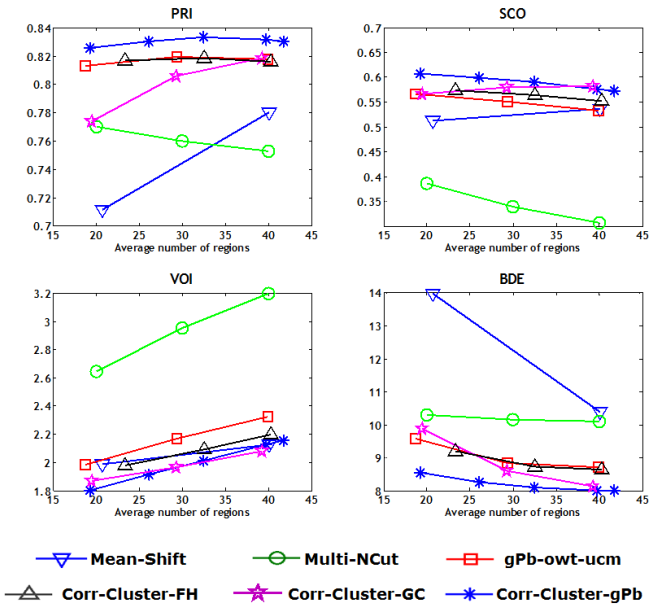


Fig. 7. Obtained evaluation measures from partitioning results according to the different baseline superpixels on the test set for semantic scene segmentation.

the choice of the baseline superpixelization algorithm. The performance of the proposed correlation clustering algorithm was evaluated using three different baseline superpixelization algorithms that include the graph-based local variation algorithm referred to as Felzenszwalb-Huttenlocher (FH) algorithm [11], the graph-cut based over-segmentation algorithm (GC) [36], and the gPb-owt. In comparison to the segmentation results obtained by the gPb-owt, the FH algorithm produced more irregular superpixels while the GC algorithm produced more regular superpixels (see Fig. 6). Note that for this empirical evaluation, we employed a random split of 50% for training and 50% for testing for the task of semantic scene segmentation. Regardless of the choice of the baseline superpixelization algorithm, the proposed correlation clustering algorithm performed better than previous partitioning algorithms. It should be noted that there was a slight performance difference depending on the baseline superpixelization, and the gPb-owt baseline superpixelization performed the best among all algorithms compared for the task of semantic scene segmentation.

We can also construct "task-specific" superpixelization algorithm based on the proposed framework. To do this, a pixel-based graph, a pairwise feature vector between neighboring pixels, and (approximately) ground-truth superpixelizations on training images are necessary.

For our learned predictors, we observed that 81 percent of the test-instances were solved exactly by our relaxation. For the instances that were not solved exactly, our rounding heuristic provided feasible solutions.

Regarding the runtime of our algorithm, we observed that for test-time inference it took on average around 10 seconds per image on a 2.67GHz processor, whereas the overall training took 10 hours on the training set. Note that other partitioning algorithms such as the multiscale NCut and the gPb-owt-ucm took on average a few minutes per image.

Region-labeling performances. To validate that our task-specific image partitioning is conducive to the specific labeling task, we estimated a single label for each region independently by a one-vs-one multi-class support vector machine (SVM) with an RBF-kernel using libsvm [37]. For this, we extracted a 449-dimensional region-feature vector, which includes color histograms, gradient histograms, spatial location histograms, and SIFT descriptors, using VLFeat [38]. In order to train the region-labeling classifier, each partitioning algorithm produced regions from the training images, and a single ground-truth label for each region was assigned by majority voting on the constituent pixels. Note that we designed each partitioning algorithm to generate 40 regions for semantic scene segmentation and 20 regions for surface layout labeling, on average. We compared labeling performances from different image partitionings by measuring the mean pixel accuracies.

TABLE II
MEAN PIXEL ACCURACIES (%) OBTAINED BY REGION-LABELING ON THE TEST SET (SEMANTIC: SEMANTIC SCENE SEGMENTATION, GEOMETRIC: SURFACE LAYOUT LABELING).

	Semantic	Geometric
Our superpixels	65.27	84.73
Mean-shift	70.70	76.95
Multi-NCut	75.60	85.32
gPb-owt-ucm	76.06	87.17
gPb-Hoiem-Semantic	73.84	85.12
gPb-Hoiem-Geometric	72.66	85.46
Supervised-NCut-Semantic	75.73	85.78
Supervised-NCut-Geometric	75.34	86.53
Corr-Cluster-Semantic	**77.01**	87.45
Corr-Cluster-Geometric	70.14	**88.15**

Table II shows that for both tasks, the task-specific partitioning improved the region-labeling performances. Even though the proposed partitioning with our simple region-labeling method in the experiment did not produce the highest mean pixel accuracy of 79.42% in the task of semantic scene

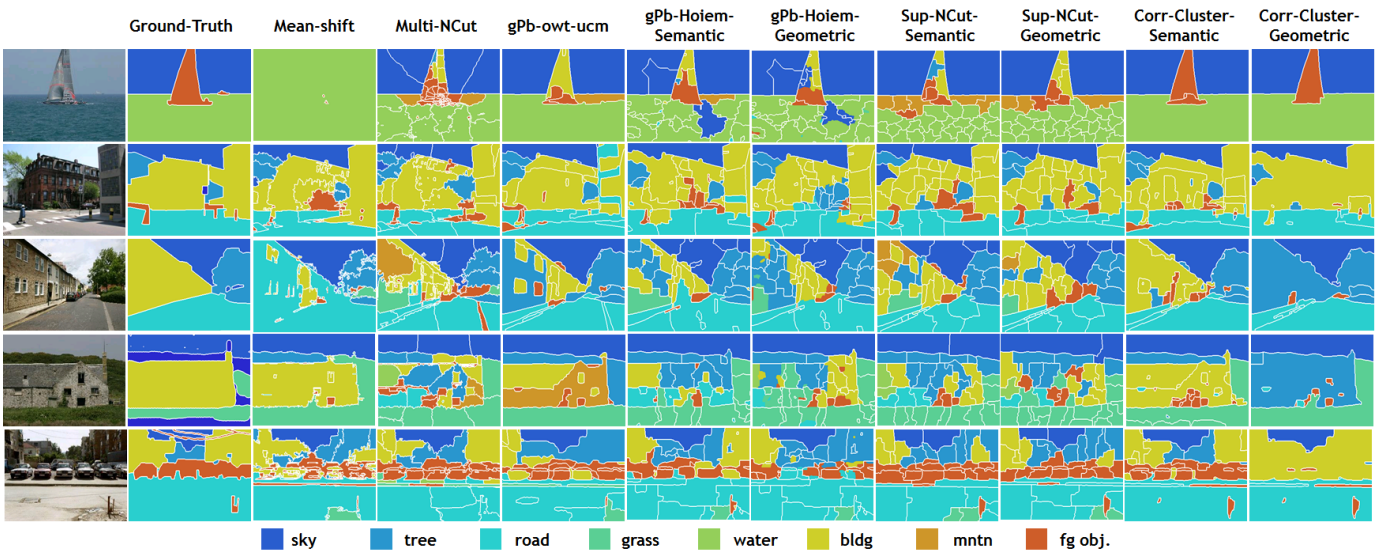


Fig. 8. Examples of image partitionings and region-labelings for semantic scene segmentation. White colors indicate region boundaries.

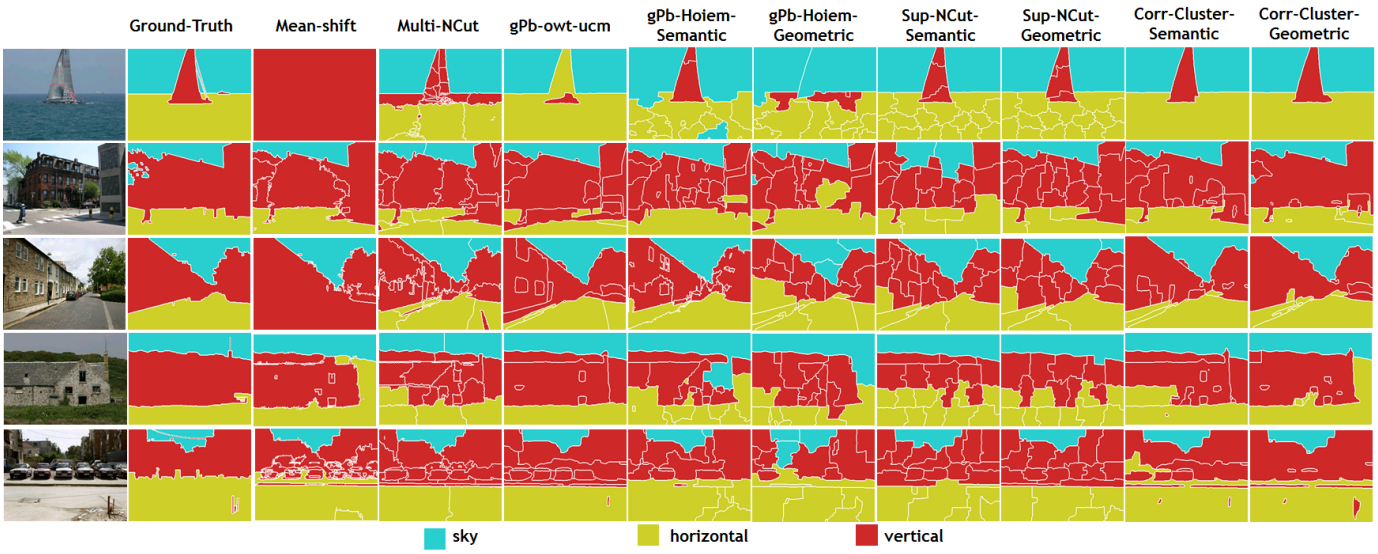


Fig. 9. Examples of image partitionings and region-labelings for surface layout labeling. White colors indicate region boundaries.

segmentation reported in [3][2] and 91.0% in the task of surface layout labeling reported in [2], the proposed task-specific correlation clustering is significant in that in comparison to the task-oblivious partitioning algorithms, it improves the partitioning to be closer to the ideal partitioning and can help improve the labeling performance easily. In the realm of semantic scene segmentation and surface layout labeling on the Stanford background dataset, in which significant effort is needed to achieve even improvements well under a percent [2], [3], 1% improvement achieved by the proposed algorithm in Table II is a significant improvement.

Qualitative results. Fig. 8 and 9 show some example partitionings and region-labelings on test images obtained by various partitioning algorithms for the tasks of semantic scene segmentation and surface layout labeling, respectively. For semantic scene segmentation, partitioning results by Corr-Cluster-Semantic are closer to the ground-truth partitionings, and these lead to qualitatively better labeling results. For surface layout labeling, Corr-Cluster-Geometric similarly yielded the best results in terms of both image partitionings and region-labelings. According to the task, the task-specific correlation clustering partitioned an image differently: Corr-Cluster-Geometric appears to produce broader regions than Corr-Cluster-Semantic.

The gPb-Hoiem treats each edge as an independent pairwise instance, therefore, the partitioning results are not stable (producing inconsistent local regions) even though it uses additional features. On the other hand, the gPb-owt-ucm uses combination of multiscale cues with a globalization machinery that reduces clutter edges and completes a set of

[2]Their algorithm is very computational-demanding (a few minutes per image).

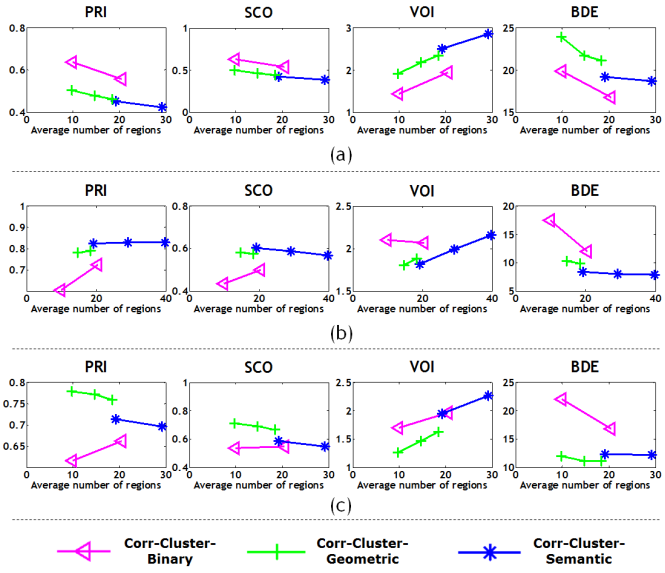


Fig. 10. Obtained evaluation measures from partitioning results on the test set for binary foreground-background segmentation (a), semantic scene segmentation (b), and surface layout labeling (c).

TABLE III
MEAN PIXEL ACCURACIES (%) OBTAINED BY REGION-LABELING ON THE TEST SET (BINARY: BINARY FOREGROUND-BACKGROUND SEGMENTATION, SEMANTIC: SEMANTIC SCENE SEGMENTATION, GEOMETRIC: SURFACE LAYOUT LABELING).

	Binary	Semantic	Geometric
Corr-Cluster-Binary	**90.25**	61.45	76.84
Corr-Cluster-Semantic	89.50	**77.01**	87.45
Corr-Cluster-Geometric	88.32	70.14	**88.15**

closed contours.

Another task. In order to reconfirm the capability of the task-specific correlation clustering to produce effective partitioning specific to the binary foreground-background segmentation task, the proposed task-specific partitioning was applied to the dataset where the 7 background classes were grouped into a single background class. As shown in Fig. 10 and Table III[3], for the task of binary foreground-background segmentation, Corr-Cluster-Binary, which indicates correlation clustering using the parameters learnt for the task of binary foreground-background segmentation, performed better than Corr-Cluster-Geometric and Corr-Cluster-Semantic. Partitionings obtained by Corr-Cluster-Binary were also evaluated for other tasks, and it was reconfirmed that task-specific partitioning is more conducive to the task in hand.

Other datasets. The proposed algorithm also has the potential to improve the performance of *generic* image partitioning as a supervised learning framework for image partitioning. We evaluate the proposed algorithm on the Berkeley segmentation dataset (BSDS) [39] for supervised generic partitioning. The BSDS contains 300 natural images

[3]Here, we designed each algorithm to produce 20 regions for binary foreground-background segmentation, on average.

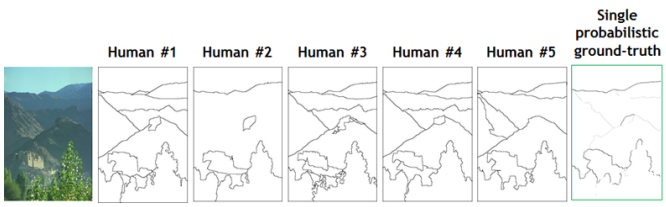


Fig. 11. Examples of partitionings by multiple human subjects and single probabilistic (real-valued) ground-truth partitioning.

TABLE IV
QUANTITATIVE RESULTS ON THE BSDS TEST SET.

Test set	PRI	SCO	VOI	BDE
Mean-shift	0.60	0.47	2.04	29.93
Multi-NCut	0.73	0.31	3.04	14.26
gPb-owt-ucm	0.80	0.58	1.85	11.46
gPb-Hoiem	0.72	0.32	3.19	14.80
Supervised-NCut	0.72	0.26	3.41	16.61
Corr-Cluster	**0.81**	**0.60**	**1.83**	**11.19**

which was split into the 200 training images and 100 test images. Since each image is partitioned by multiple human subjects, we defined a single probabilistic (real-valued) ground-truth partitioning of each image only for training by the proposed algorithm (see Fig. 11). The gPb-Hoiem and the supervised NCut used a different ground-truth for training on the BSDS: declare two superpixels to lie in the same segment only if all human subjects declare them to lie in the same segment.

Table IV shows the obtained results on test images when all partitioning algorithms were set to produce 30 disjoint regions per image, on average. Note that the level-threshold in producing segmentation results at a universal fixed scale in Table 2 in [13] was optimized differently for each performance measure while in our experiment the same level-threshold (0.155) was used for all performance measures in evaluating gPb-owt-ucm.

Irrespective of the measure, the proposed algorithm (Corr-Cluster) gave the best results. Moreover, it is noticed that these results are similar or even better than the state-of-the-art results on the BSDS [13], [40], [41].

We changed the level-threshold for gPb-owt-ucm and R for correlation clustering to produce different numbers of regions per image, on average, and observed that the correlation clustering always performed better than the gPb-owt-ucm (see Fig. 12), as on the Stanford background dataset. As the number of regions increased, the PRI and VOI increased while the SCO and BDE decreased for both algorithms. We set the level-threshold of 0.155 for gPb-owt-ucm and R of 0.15 for correlation clustering, since for both algorithms these values produced on average 30 regions per image and gave the best results with regards to the four measures. Improvement of 1% in PRI, 2% in SCO, 0.02 in VOI, and 0.3 pixel in BDE on the BSDS test set is comparable to the improvements reported in [13], [40] (1% in PRI, 2% in SCO, 0.08 in VOI, and 1 pixel in BDE). We observed that in comparison to the gPb-owt-ucm, by the proposed correlation clustering, 63 segmentation results were improved, 10 results did not change, and the rest

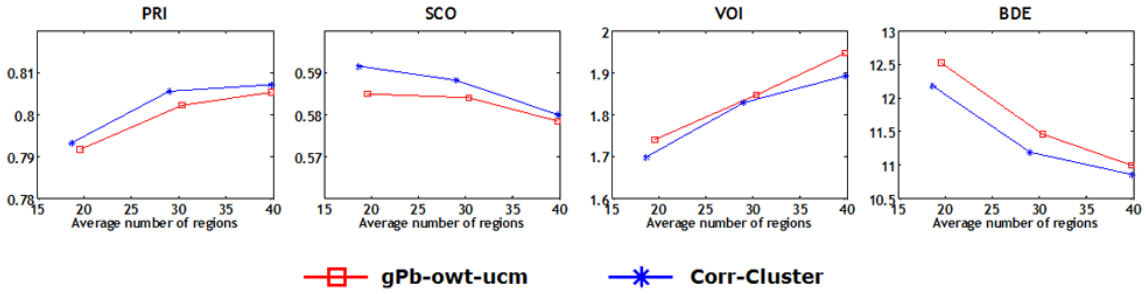


Fig. 12. Obtained evaluation measures from partitioning results of gPb-owt-ucm and Corr-Cluster on the BSDS test set according to the average number of regions.

TABLE V
OBTAINED F-MEASURES ON THE BSDS TEST SET.

Test set	F-measure
Mean-shift	0.50
Multi-NCut	0.59
gPb-owt-ucm	0.69
gPb-Hoiem	0.62
Supervised-NCut	0.53
Corr-Cluster	**0.71**

TABLE VI
QUANTITATIVE RESULTS ON THE MSRC TEST SET.

Test set	PRI	SCO	VOI	BDE
Mean-shift	0.734	0.606	1.649	13.944
Multi-NCut	0.628	0.341	2.765	11.941
gPb-owt-ucm	0.779	0.628	1.675	9.800
gPb-Hoiem	0.614	0.353	2.847	13.533
Supervised-NCut	0.601	0.287	3.101	13.498
Corr-Cluster	**0.773**	**0.632**	**1.648**	**9.194**

27 results got worse on the BSDS test set.

On the BSDS benchmark dataset, the F-measure has been popularly used for evaluation of segment boundaries obtained by image partitioning algorithms. Therefore, we also computed the F-measure on the BSDS test set, and as shown in Table V, the proposed correlation clustering gave the best score.

Fig. 13 shows some example partitionings on test images obtained by various partitioning algorithms. The proposed correlation clustering (Corr-Cluster) yielded the best partitioning results.

We also conducted image partitionings on the MSRC dataset [42] that is composed of 591 natural images. We split the data into 45% training, 10% validation, and 45% test sets, following [42]. The performance was evaluated using the clean ground-truth object instance labeling of [43]. On average, all partitioning algorithms were set to produce 15 disjoint regions per image on the MSRC dataset. As shown in Table VI, the proposed correlation clustering gave the best results on the test set. Note that as on the BSDS dataset mentioned above, we report the results not on the whole set but on the test set, and we observed the same tendency on the MSRC dataset as on the BSDS dataset.

We also trained on the MSRC dataset and tested on the BSDS dataset. This decreases the performance over training and testing on the BSDS dataset. This observation is also true in the reverse direction, i.e. when training on the BSDS dataset and testing on the MSRC dataset. Overall this suggests that these two datasets have different statistics. Therefore, we believe that our framework is helpful even for regular image segmentation applications, because it allows the partitioning to be tuned to the particular dataset at hand.

V. CONCLUSION

This work addressed the problem of task-specific image partitioning by supervised training. We proposed the correlation clustering model which aims to merge superpixels into regions of homogeneity with respect to the solution of any particular image labeling problem. The LP relaxation was used to approximately solve the correlation clustering over a superpixel graph where a rich pairwise feature vector was defined based on several visual cues. The S-SVM was used for supervised training of parameters in correlation clustering, and the cutting plane algorithm with LP-relaxed inference was applied to solve the optimization problem of S-SVM. Experimental results showed that the proposed task-specific correlation clustering outperformed other image partitioning algorithms on semantic scene segmentation and surface layout labeling. The proposed framework is applicable to a broad variety of other high-level vision tasks.

REFERENCES

[1] L. Ladickỳ, C. Russell, P. Kohli, and P. H. S. Torr, "Associative hierarchical crfs for object class image segmentation," in *Proc. IEEE International Conference on Computer Vision*, 2009.
[2] S. Gould, R. Fulton, and D. Koller, "Decomposing a scene into geometric and semantically consistent regions," in *Proc. IEEE International Conference on Computer Vision*, 2009.
[3] M. P. Kumar and D. Koller, "Efficiently selecting regions for scene understanding," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2010.
[4] D. Hoiem, A. A. Efros, and M. Hebert, "Recovering surface layout from an image," *International Journal of Computer Vision*, vol. 75, pp. 151–172, 2007.
[5] B. Liu, S. Gould, and D. Koller, "Single image depth estimation from predicted semantic labels," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2010.
[6] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2009 Results," http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html.

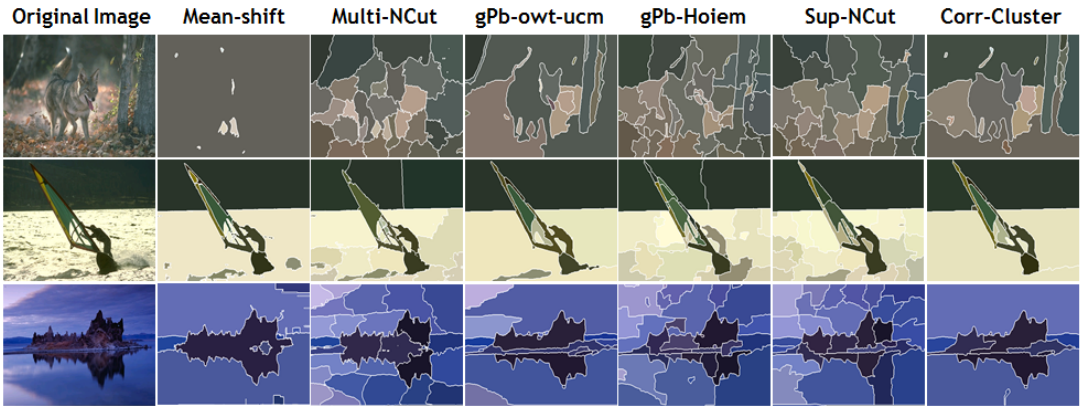


Fig. 13. Examples of image partitionings on the BSDS test set.

[7] B. Fulkerson, A. Vedaldi, and S. Soatto, "Class segmentation and object localization with superpixel neighborhoods," in *Proc. IEEE International Conference on Computer Vision*, 2009.

[8] P. Kohli, L. Ladickỳ, and P. H. S. Torr, "Robust higher order potentials for enforcing label consistency," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[9] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 603–619, 2002.

[10] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 888–905, 2000.

[11] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, pp. 167–181, 2004.

[12] A. Vedaldi and S. Soatt, "Quick shift and kernel methods for mode seeking," in *Proc. European Conference on Computer Vision (ECCV)*, 2008.

[13] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 898–916, 2011.

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

[15] N. Bansal, A. Blum, and S. Chawla, "Correlation clustering," *Machine Learning*, vol. 56, pp. 89–113, 2004.

[16] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and independent output variables," *Journal of Machine Learning Research*, vol. 6, pp. 1453–1484, 2005.

[17] C. Fowlkes, D. Martin, and J. Malik, "Learning affinity functions for image segmentation: combining patch-based and gradient-based approaches," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[18] F. Bach and M. I. Jordan, "Learning spectral clustering," in *Proc. Neural Information Processing Systems*, 2003.

[19] N. Shental, A. Zomet, T. Hertz, and Y. Welss, "Pairwise clustering and graphical models," in *Proc. Neural Information Processing Systems*, 2004.

[20] T. Cour, N. Gogin, and J. Shi, "Learning spectral graph segmentation," in *Proc. International Conference on Artificial Intelligence and Statistics*, 2005.

[21] S. Turaga, K. Briggman, M. Helmstaedter, W. Denk, and H. Seung, "Maximin affinity learning of image segmentation," in *Proc. Neural Information Processing Systems*, 2009.

[22] T. Finley and T. Joachims, "Supervised clustering with support vector machines," in *Proc. International Conference on Machine Learning*, 2005.

[23] B. Taskar, "Learning structured prediction models: a large margin approach," *Ph.D. thesis, Stanford University*, 2004.

[24] T. Finley and T. Joachims, "Training structural SVMs when exact inference is intractable," in *Proc. International Conference on Machine Learning*, 2008.

[25] A. Kulesza and F. Pereira, "Structured learning with approximate inference," in *Proc. Neural Information Processing Systems*, 2007.

[26] A. F. T. Martins, N. A. Smith, and E. P. Xing, "Polyhedral outer approximations with application to natural language parsing," in *Proc. International Conference on Machine Learning*, 2009.

[27] S. Chopra and M. R. Rao, "The partition problem," *Math. Program*, vol. 59, pp. 87–115, 1993.

[28] S. Nowozin and S. Jegelka, "Solution stability in linear programming relaxations: Graph partitioning and unsupervised learning," in *Proc. International Conference on Machine Learning*, 2009.

[29] O. Pele and M. Werman, "Fast and robust earth mover's distances," in *Proc. IEEE International Conference on Computer Vision*, 2009.

[30] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *International Journal of Computer Vision*, vol. 43, pp. 29–44, 2001.

[31] D. Batra, R. Sukthankar, and T. Chen, "Learning class-specific affinities for image labelling," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[32] T. Cour, B. Sapp, A. Nagle, and B. Taskar, "Talking pictures: temporal grouping and dialog-supervised person recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[33] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, vol. 66, pp. 846–850, 1971.

[34] M. Meila, "Computing clusterings: An axiomatic view," in *Proc. International Conference on Machine Learning*, 2005.

[35] J. Freixenet, X. Munoz, D. Raba, J. Marti, and X. Cufi, "Yet another survey on image segmentation: Region and boundary information integration," in *Proc. European Conference on Computer Vision (ECCV)*, 2002.

[36] O. Veksler, Y. Boykov, and P. Mehrani, "Superpixels and supervoxels in an energy optimization framework," in *Proc. European Conference on Computer Vision (ECCV)*, 2010.

[37] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[38] A. Vedaldi and B. Fulkerson, *VLFeat: An open and portable library of computer vision algorithms.*, 2008, http://www.vlfeat.org/.

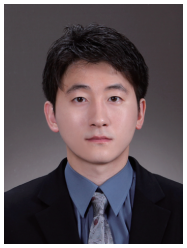
[39] C. Fowlkes, D. Martin, and J. Malik, *The Berkeley Segmentation Dataset and Benchmark (BSDB)*, http://www.cs.berkeley.edu/projects/vision/grouping/segbench/.

[40] T. Kim, K. Lee, and S. Lee, "Learning full pairwise affinities for spectral segmentation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

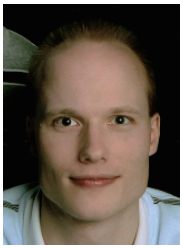
[41] S. R. Rao, H. Mobahi, A. Y. Yang, S. S. Sastry, and Y. Ma, "Natural image segmentation with adaptive texture and boundary encoding," in *Proc. Asian Conference on Computer Vision (ACCV)*, 2009.

[42] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost: joint appreearence, shape and context modeling for multi-class object recognition and segmentation," in *Proc. European Conference on Computer Vision (ECCV)*, 2006.

[43] T. Malisiewicz and A. A. Efros, "Improving spatial support for objects via multiple segmentations," in *Proc. British Machine Vision Conference (BMVC)*, 2007.



Sungwoong Kim (S'07-M'12) received the B.S. and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2004 and 2011, respectively. Since 2012 he is with Qualcomm Research Korea where he is a senior engineer. His research interests include machine learning for multimedia signal processing, discriminative training, and graphical modeling.



Sebastian Nowozin received his computer science diploma from the Technical University of Berlin in 2006, his MEng degree from the Shanghai Jiaotong University in 2006, and his PhD degree (Dr.rer.nat) in 2009 from the Technical University of Berlin, in cooperation with the Max Planck Institute for Biological Cybernetics. Since 2009 he is with Microsoft Research at Cambridge, UK, where he currently is a researcher. His research interest include machine learning, computer vision, graphical models, and numerical optimization.

Pushmeet Kohli is a research scientist in the Machine Learning and Perception group at Microsoft Research Cambridge, and an associate of the Psychometric Centre, University of Cambridge. His PhD thesis written at Oxford Brookes University was the winner of the British Machine Vision Associations Sullivan Doctoral Thesis Award, and a runner-up for the British Computer Society's Distinguished Dissertation Award. Pushmeets research revolves around Intelligent Systems and Computational Sciences with a particular emphasis on algorithms and models for scene understanding and human pose estimation. His papers have appeared in SIGGRAPH, NIPS, ICCV, AAAI, CVPR, PAMI, IJCV, CVIU, ICML, AISTATS, AAMAS, UAI, ECCV, and ICVGIP and have won best paper awards in ECCV 2010, ISMAR 2011 and ICVGIP 2006, 2010.

Chang D. Yoo (S'92-M'96-SM'11) received the B.S. degree in Engineering and Applied Science from California Institute of Technology in 1986, the M.S. degree in Electrical Engineering from Cornell University in 1988 and the Ph.D. degree in Electrical Engineering from Massachusetts Institute of Technology in 1996. From January 1997 to March 1999 he worked at Korea Telecom as a Senior Researcher. He joined the Department of Electrical Engineering at Korea Advanced Institute of Science and Technology in April 1999. From March 2005 to March 2006, he was with Research Laboratory of Electronics at MIT. His current research interests are in the application of machine learning and digital signal processing in multimedia.