Sparse Coding and Mid-Level Superpixel-Feature for ℓ_0-Graph Based Unsupervised Image Segmentation

Xiaofang Wang[1], Huibin Li[1], Simon Masnou[2], and Liming Chen[1,⋆]

[1] Ecole Centrale de Lyon, LIRIS UMR5205, F-69134, France
{xiaofang.wang,huibin.li,liming.chen}@ec-lyon.fr
[2] Université de Lyon, CNRS UMR 5208, Université Lyon 1, Institut Camille Jordan,
43 bd du 11 novembre 1918, F-69622 Villeurbanne cedex, France
masnou@math.univ-lyon1.fr

Abstract. We propose in this paper a graph-based unsupervised segmentation approach that combines superpixels, sparse representation, and a new mid-level feature to describe superpixels. Given an input image, we first extract a set of interest points either by sampling or using a local feature detector, and we compute a set of low-level features associated with the patches centered at the interest points. We define a low-level dictionary as the collection of all these low-level features. We call superpixel a region of an oversegmented image obtained from the input image, and we compute the low-level features associated with it. Then we compute for each superpixel a mid-level feature defined as the sparse coding of its low-level features in the aforementioned dictionary. These mid-level features not only carry the same information as the initial low-level features, but also carry additional contextual cue. We use the superpixels at several segmentation scales, their associated mid-level features, and the sparse representation coefficients to build graphs at several scales. Merging these graphs leads to a bipartite graph that can be partitioned using the Transfer Cut algorithm. We validate the proposed mid-level feature framework on the MSRC dataset, and the segmented results show improvements from both qualitative and quantitative viewpoints compared with other state-of-the-art methods.

Keywords: image segmentation, sparse coding, superpixels, mid-level features, ℓ_0-graph.

1 Introduction

Most unsupervised image segmentation methods, which are frequently used for high-level vision tasks like object recognition or image annotation, involve low level features such as color, boundary or texture. In particular, several methods using graphs and spectral clustering have been proposed in recent years [13] [8], however it remains challenging for those methods to provide desirable visually semantic partitions.

Generally, for those methods, building a faithful graph is critical to the final quality. The graph nodes can be pixels or regions, and the graph affinity matrix encodes the

[⋆] Thanks to Chinese Scholarship Council (CSC) for funding. This work is supported in part by the French research agency, l'Agence Nationale de Recherche (ANR), through the Visen project within the ERA-NET CHIST-ERA program, under the grant ANR-12-CHRI-0002-04.

R. Wilson et al. (Eds.): CAIP 2013, Part II, LNCS 8048, pp. 160–168, 2013.

similarity between either low level features or top down features associated with the nodes. Low level features capture object basic properties and they can be obtained with various descriptors or operators, such as color histograms, histogram of oriented gradients (HOG), scale invariant feature transform (SIFT), local binary patterns (LBP), etc. Despite progresses in the design of more informative low-level features, performances remain limited. Top down features usually convey semantic or prior knowledge about the segmented regions or objects. Many works treat the output of trained classifiers and object detectors [7], or semantic segmentation algorithm [5] as top down information to guide the low level unsupervised segmentation. However, all these top-down semantic methods require non-trivial amounts of human-labeled training data, which is unrealistic in practical situation.

In recent years, successful applications of mid-level features (e.g., bag of features) to content-based image retrieval and object categorization have motivated their introduction for other computer vision tasks such as image segmentation. Yu et al.[17] proposed bag of textons combined with clustering for image segmentation. The baseline of a mid-level feature mainly involves low-level feature extraction, representation (using hard assignments with k-means, or soft assignments via sparse coding) and pooling. In this paper, we focus on mid-level features based on sparse coding, as in [18] where first a dictionary is built by learning or human labeling, then the coefficients of the sparse representation in this dictionary are used to define mid-level features for classification or grouping. In contrast to [18], we build the dictionary from informative patches centered at interest points detected without any supervision, and each mid-level feature is the sparse coding in the dictionary of the low level feature associated with a superpixel. This way, the contextual information, which has been proved an efficient cue to discriminate two objects or images [6], is added to the original low-level features to improve the robustness of the similarity coefficient between two superpixels in the graph construction, whose quality plays a critical role to the segmentation result.

More precisely, the whole segmentation model starts by extracting interest points from the image, associating with them a set of low-level features whose collection forms a dictionary, and over-segmenting the input image into multi-layer superpixels. Then, each superpixel is associated with a sparse representation of its low level feature in the previously built dictionary. This proposed feature inherits of the original descriptors' property and covers also adaptive contextual information. Compared with related works and other benchmark algorithms on the MSRC dataset [14], the key contribution of this paper is that our new mid-level feature is able to describe better the superpixels. The similarities between superpixels are then computed based on ℓ_0 graph construction in the spirit of [16] (where only low-level features were used). Finally, the constructed graph is plugged into a robust unsupervised segmentation framework introduced in [8]. The proposed method can segment visually semantic regions, and can be used in many high-level computer vision tasks.

The organization of the paper is as follows: in Section 2 we introduce the proposed mid-level features based on the sparse coding and the segmentation framework, and in Section 3 we present and comment a few segmentation results on the MSRC dataset. We conclude in Section 4.

2 Superpixels, Mid-Level Features, and Sparse Representation

Our approach consists of three steps: 1) interest points extraction, low-level features computation, and dictionary building; 2) over-segmentation of the original image, extraction of superpixels (defined as the over-segmented regions), computation of a low-level feature for each superpixel, and sparse representation in the dictionary of step 1; 3) graph construction and partitioning.

2.1 Low-Level Features Detection and Extraction

We use low-level features extraction to build a meaningful dictionary to represent a given image. First, we extract a set of key points from the image. The meaningfulness of the low-level dictionary is highly dependent on the choice of the key points. If they capture the main structural information of the input image, then the derived dictionary will be highly meaningful. In practice, we have tested various approaches, see Fig. 1: either the interest points are randomly or densely sampled, or they are obtained using a feature descriptor, e.g., the Harris detector, the Difference of Gaussians (DoG), or the Hessian detector. The respective performances are discussed in Section 3.

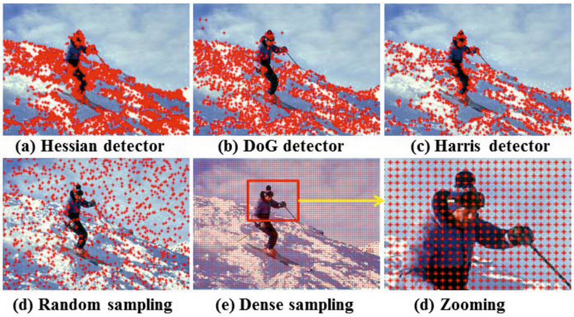

Fig. 1. Illustration of different types of interest points

Once interest points have been extracted, we consider the local image patches around them, from which low-level features can be computed (we use in this paper RGB color histograms for its strong discriminative skill, but other features as LBP histogram or SIFT may be used). Finally, our low-level dictionary is defined as the collection of all these low-level features, see Fig. 2.

2.2 Mid-Level Features Extraction over Superpixels

We call superpixel a region of an over-segmentation of the original image. In practice, we compute several over-segmentations, and we associate with each superpixel a low-level feature (in our experiments, we used RGB color histograms for its strong discriminative skill). Then we define the mid-level feature associated with a superpixel as the sparse representation of its low-level feature in the dictionary built previously, see Fig. 3 for an illustration of the whole process. More precisely, given a superpixel,

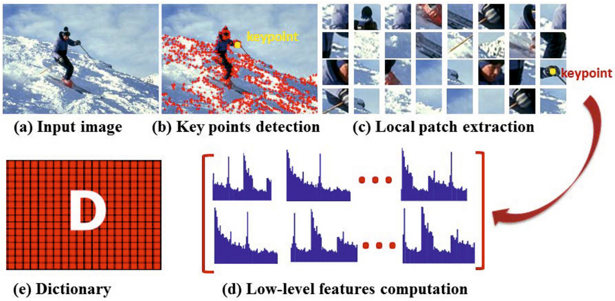

Fig. 2. Illustration of low-level features computation

suppose $x \in \mathbb{R}^m$ is the low-level feature associated with it, and let $D = [d_1 \cdots d_n] \in \mathbb{R}^{m \times n}$ be the low-level dictionary built in section 2.1. The sparse representation of x in D is obtained by solving the following optimization problem:

$$\min_{\alpha} ||x - D\alpha||_2^2 \quad s.t. \quad ||\alpha||_0 \leq L, \tag{1}$$

where $\alpha \in \mathbb{R}^n$, and $\|\alpha\|_0 := \|\alpha\|_{\ell_0}$ is the number of its non-zero coefficients. Suppose $\hat{\alpha}$ is a solution of the problem and $\Lambda_{\hat{\alpha}} = \{j|\hat{\alpha}(j) \neq 0\}$ is the index set of non-zero coefficients of $\hat{\alpha}$, then the mid-level feature associated with the low-level feature x is defined as

$$\hat{x} = D\hat{\alpha} = \sum_{j \in \Lambda_{\hat{\alpha}}} d_j \hat{\alpha}(j). \tag{2}$$

Therefore, the mid-level feature $\hat{x}$ is a linear combination of several low-level features, thus not only carries the same information as the original low-level features, but also carries additional contextual cue.

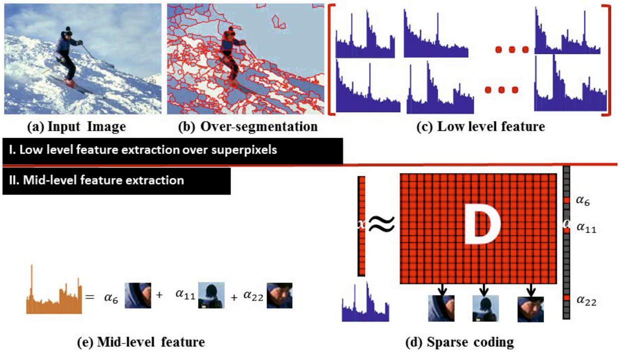

Fig. 3. Illustration of mid-level features computation

2.3 Graph Construction and Partitioning

Once mid-level features have been computed, we build the graph that will be plugged into a spectral clustering algorithm to perform image segmentation. This is done as follows: For each scale of over-segmentation (i.e. for each instance of over-segmentation), we construct a graph whose nodes are the superpixels at that scale, and whose graph edges and weights are computed using ℓ_0-sparse representation. More precisely, we consider as dictionary the mid-level features associated with the superpixels. Then, as in Equation (2), each mid-level feature $\hat{x}_i$ can be represented as a sparse linear combination $\hat{x}_i = \sum_j \alpha^i_j \hat{x}_j$ of the other mid-level features. The similarity coefficient of any pair $\hat{x}_i, \hat{x}_j$ of superpixels is defined as $w_{ij} = \begin{cases} 1 & \text{if } i = j \\ 1 - (r_{ij} + r_{ji})/2 & \text{if } i \neq j. \end{cases}$ where r_{ij} is the sparse representation error of $\hat{x}_i$ and $\hat{x}_j$, i.e. $r_{ij} = \|\hat{x}_i - \alpha^i_j \hat{x}_j\|_2^2$.

We collect all ℓ_0 affinity matrices obtained from all over-segmented images, and we concatenate them diagonally into a unique matrix denoted as W_{SS}, together with the pixel-superpixels affinity matrix W_{IS}. Then we consider the bipartite graph associated with the matrix $B = \begin{bmatrix} W_{IS} \\ W_{SS} \end{bmatrix}$ and the Transfer Cut algorithm [8] is applied to partition the bipartite graph into K clusters by solving the following generalized eigenvalue problem over superpixels only $L_V \mathbf{f} = \lambda D_V \mathbf{f}$, where $L_V = D_V - W_V$, $D_V = diag(B^\top \mathbf{1})$, and $W_V = B^\top D_U^{-1} B$, $D_U = diag(B\mathbf{1})$, see [8] for more details.

3 Experimental Results

3.1 Database and Parameter Settings

We evaluate our approach on the Microsoft Research Cambridge (MSRC) database, which contains 591 images from 23 object classes, and we use for the evaluation the accurate ground-truth segmentations of [9]. To quantitatively evaluate the performance, we apply four popular measurements : 1) Probabilistic Rand Index (PRI) [15]; 2) Variation of Information (VOI) [11]; 3) Global Consistency Error (GCE) [10]; and 4) Boundary Displacement Error (BDE) [4]. A segmentation result is better if PRI is higher and the other three ones are lower. For low-level features extraction, we only use the color feature in RGB space, and the feature dimension is reduced from 256×3 to 64 by PCA. For mid-level dictionary building via sparse coding, we use the Orthogonal Matching Pursuit (OMP) algorithm [12] to solve Eqn. 1 and set the sparsity number $L = 4$ according to the experimental results.

On the step of graph construction and partitioning, we proceed as in our previous work [16], i.e. we derive from the original image 5 or 6 oversegmented images (this number of scales being experimentally satisfactory) obtained by the Mean Shift (MS) method [2] and by the FH method [3]. More precisely, we derive three images by the MS method using the sets of parameters (*hs*, *hr*, *M*)= $\{(7,7,100)$, $(7,9,100)$, and $(7,11,100)\}$, respectively, where *hs* and *hr* are bandwidth parameters in the spatial and range domains, and *M* is the minimum size of each segment. Either two of three over-segmented images are provided by the FH method using as parameters (σ, *c*, *M*) either $\{(0.5,100,50),(0.8,200,100)\}$, or $\{(0.8,150,50),(0.8,200,100),(0.8,300,100)\}$.

Table 1. Comparison of different feature detectors on the whole MSRC database (red color indicates the best result)

Detector	**PRI↑**	**VoI↓**	**GCE↓**	**BDE↓**
Harris detector	0.8195	1.4214	0.1694	9.4530
Hessian detector	0.8177	1.4366	0.1691	9.9951
DoG detector	0.8226	1.3900	0.1670	9.3955
Random sampling	0.8069	1.5578	0.1781	10.1746
Dense sampling	0.8280	1.3452	0.1633	9.4403

To build the ℓ_0 graph, the sparsity number $L = 3$ is used for all the experiments, see [16] for more details. We organize our experimental results as follows: first, we compare the performances of the five different kinds of low-level feature detectors introduced in section 2.1; then, we list the quantitative results of our proposed method on different subsets of MSRC database and compare it with several state-of-the-art methods; finally, we show some visual examples of our method.

Table 2. Performances of our method on MSRC and comparison with state-of-the-art methods

Metric	**PRI↑**		**VoI↓**		**GCE↓**		**BDE↓**	
Object class	**baseline**	**new**	**baseline**	**new**	**baseline**	**new**	**baseline**	**new**
1. grass, cow	0.8889	0.8978	0.7927	0.8417	0.1006	0.1059	4.8316	4.9181
2. tree, grass, sky	0.7865	0.7963	1.2569	1.3664	0.1727	0.1990	18.6141	13.6065
3. building, sky	0.8429	0.8697	1.2660	1.3768	0.1670	0.1755	8.0268	8.3904
4. aeroplane, grass, sky	0.9083	0.9202	1.3133	1.2662	0.1463	0.1649	4.1802	4.3369
5. cow, grass, mount	0.9038	0.8647	0.5641	0.7804	0.0752	0.0889	4.2286	4.8817
6. face, body	0.7176	0.7277	2.2429	2.3892	0.2601	0.2669	16.1357	15.2383
7. car, building	0.7423	0.7624	2.2676	2.1879	0.2044	0.2546	12.3907	12.3268
8. bike, building	0.7037	0.7196	2.0662	2.1575	0.2729	0.2854	10.7725	10.9580
9. sheep, grass	0.8837	0.8867	0.7287	0.7166	0.0853	0.0874	4.7323	4.9983
10. flower	0.8712	0.8766	0.6368	0.7172	0.0836	0.0927	6.8501	5.7331
11. sign	0.8581	0.8839	0.7668	0.7591	0.0929	0.0940	6.4911	6.3972
12. bird, sky, grass, water	0.8820	0.8932	0.6977	0.7215	0.0963	0.0831	5.6918	5.9985
13. book	0.6714	0.6613	1.7574	1.9669	0.1596	0.1633	18.9275	17.7393
14. chair	0.7395	0.7806	1.3144	1.6839	0.1862	0.1807	11.7096	7.7027
15. cat	0.7532	0.7483	1.3479	1.2819	0.1272	0.1240	12.0134	11.8589
16. dog	0.8030	0.8029	1.2856	1.2436	0.1394	0.1613	9.7475	9.5381
17. road, building	0.8439	0.8610	1.6346	1.7412	0.2002	0.2025	9.0031	8.4299
18. water, boat	0.8548	0.8424	1.0310	1.0947	0.0935	0.1088	9.1329	12.4533
19. body, face	0.8376	0.8275	1.6961	1.9347	0.1931	0.2124	7.4399	8.8790
20. water, boat, sky, mount	0.8884	0.9154	1.1942	1.0002	0.1602	0.1279	6.3682	5.6792
Average performance								
Method	**PRI↑**		**VoI↓**		**GCE↓**		**BDE↓**	
Our new method	0.8269		1.3614		0.1590		9.0032	
Baseline [16]	0.8190		1.2930		0.1508		9.3644	
NCut [13]	0.8052		1.2516		-		-	
LRR(CH)[1]	0.7912		1.3002		-		-	
MS[2]	0.7307		1.7472		-		-	

3.2 Experimental Results

As mentioned in section 2.1, the property of the low-level dictionary is highly dependent on the selection of the key points. Therefore, we compared the Harris detector, Difference of Gaussian (DoG), Hessian detector, random sampling, and the dense sampling (see Fig. 1). The results are shown in Tab. 1, from which we can deduce that dense sampling is the most efficient way to extract interest points. The main reason is that dense sampling can capture almost all information of the image and is well-suited for sparse coding that requires an over-complete dictionary.

We compare in Table 2 the performances of our method on the MSRC database and the performances of the method we proposed in [16] (limiting to RGB histogram as superpixel feature, and calling *baseline* this reference algorithm). Obviously, our new method can achieve excellent performances on segmenting object classes such as *cow*, *building*, *sheep*, *flower*, *sign*, *bird*, *road*, and *boat*, but is less efficient for *tree*, *face*, *cat*, *dog*, *bike*, etc. The visual results are also shown in Fig.4. The reasons for the

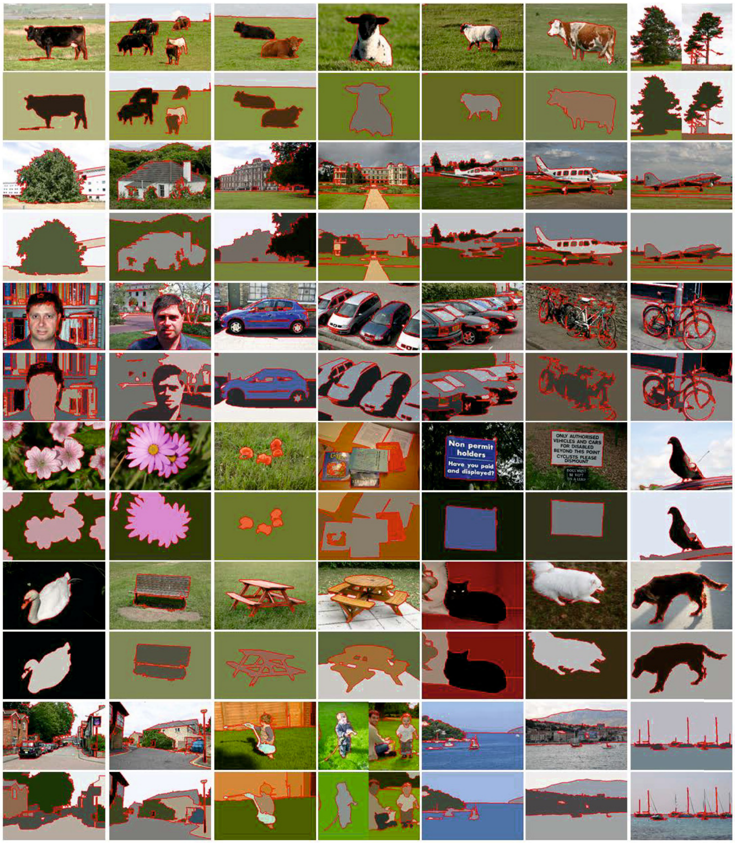
Non permit holders
Have you paid and displayed?
ONLY AUTHORISED VEHICLES AND CARS FOR DISABLED BEYOND THIS POINT CYCLISTS PLEASE DISMOUNT

Fig. 4. Examples of segmented results on the MRSC dataset (for each experiment, we show the segmentation result, and the segmentation superimposed with the original image)

difference performances are various: **1)** objects like *face*, *cat*, and *dog* usually have complex backgrounds mainly associated with indoor scene which makes the evaluation unfair for the machine algorithms since the ground-truth does not label the indoor objects. On the other side, in the case of objects without complex backgrounds, our method can segment them correctly even if the object itself presents obvious color variations like on *cow*, *building* and *flower*; **2)** objects like *face* or *bike* can be subject to strong illumination changes which prevent the machine algorithms from grouping object correctly if only color is used as low level descriptor. Results should be improved if other descriptors as LBP were used, and this is the purpose of future work. **3)** the quality of segmentation can also be influenced greatly by the way superpixels are extracted.

We compare the performances of our approach with other state-of-the-arts algorithms in Tab. 2. We used the scores given in [1], observing that GCE and BDE were not reported. Our method ranks first according to PRI and BDE, which makes it one of the most competitive algorithms.

4 Conclusion

We introduced a new unsupervised image segmentation method based on ℓ_0-graph, superpixels, mid-level features, and sparse coding. An nice property of the mid-level feature we propose is that it can capture adaptive contextual information and carries as well the original low level feature information. Quantitative comparison with the state-of-art methods, as well as visual results, indicate that our new algorithm is a competitive image segmentation method.

References

1. Cheng, B., Liu, G., Wang, J., Huang, Z., Yan, S.: Multi-task low-rank affinity pursuit for image segmentation. In: ICCV, pp. 2439–2446 (2011)
2. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. PAMI 24(5), 603–619 (2002)
3. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. IJCV 59(2), 167–181 (2004)
4. Freixenet, J., Muñoz, X., Raba, D., Martí, J., Cufí, X.: Yet another survey on image segmentation: Region and boundary information integration. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part III. LNCS, vol. 2352, pp. 408–422. Springer, Heidelberg (2002)
5. Fu, H., Qiu, G.: Integrating low-level and semantic features for object consistent segmentation. In: Int. Conf. on Image and Graphics (ICIG), pp. 39–44 (2011)
6. Lee, Y.J., Grauman, K.: Object-graphs for context-aware visual category discovery. PAMI 34(2), 346–358 (2012)
7. Li, L.J., Su, H., Xing, E.P., Fei-Fei, L.: Object bank: A high-level image representation for scene classification and semantic feature sparsification. Advances in Neural Information Processing Systems (2010)
8. Li, Z., Wu, X.M., Chang, S.F.: Segmentation using superpixels: A bipartite graph partitioning approach. In: CVPR, pp. 789–796 (2012)
9. Malisiewicz, T., Efros, A.A.: Improving spatial support for objects via multiple segmentations. In: BMVC (2007)

10. Martin, D.R., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV, pp. 416–425 (2001)
11. Meila, M.: Comparing clusterings: an axiomatic view. In: ICML, pp. 577–584 (2005)
12. Pati, Y., Rezaiifar, R., Krishnaprasad, P.: Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In: 27th Asilomar Conference on Signals, Systems and Computers, pp. 40–44 (1993)
13. Shi, J., Malik, J.: Normalized cuts and image segmentation. PAMI 22(8), 888–905 (2000)
14. Shotton, J., Winn, J.M., Rother, C., Criminisi, A.: Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 1–15. Springer, Heidelberg (2006)
15. Unnikrishnan, R., Pantofaru, C., Hebert, M.: Toward objective evaluation of image segmentation algorithms. IEEE Trans. Pattern Anal. Mach. Intell. 29(6), 929–944 (2007)
16. Wang, X., Li, H., Masnou, S., Chen, L.: A graph-cut approach to image segmentation using an affinity graph based on ℓ_0- sparse representation of features. In: IEEE Int. Conf. on Image Proc. (2013) (accepted)
17. Yu, Z., Li, A., Au, O., Xu, C.: Bag of textons for image segmentation via soft clustering and convex shift. In: CVPR, pp. 781–788 (2012)
18. Zou, W., Kpalma, K., Ronsin, J.: Semantic segmentation via sparse coding over hierarchical regions. In: ICIP, pp. 2577–2580 (2012)