A graph-cut approach to image segmentation using an affinity graph based on l0–sparse representation of features

Xiaofang Wang, Huibin Li, Charles-Edmond Bichot, Simon Masnou, Liming Chen

A graph-cut approach to image segmentation using an affinity graph based on ℓ_0-sparse representation of features

Xiaofang Wang[1], Huibin Li[1], Charles-Edmond Bichot[1],
Simon Masnou[2], and Liming Chen[1]

[1]*Ecole Centrale de Lyon, LIRIS UMR5205, F-69134, France*
[2]*Université de Lyon, CNRS UMR 5208, Université Lyon 1, Institut Camille Jordan,
F-69622 Villeurbanne, France*

Abstract

We propose a graph-cut based image segmentation method by constructing an affinity graph using ℓ_0 sparse representation. Computing first over-segmented images, we associate with all segments, that we call superpixels, a collection of features. We find the sparse representation of each set of features over the dictionary of all features by solving a ℓ_0-minimization problem. Then, the connection information between superpixels is encoded as the non-zero representation coefficients, and the affinity of connected superpixels is derived by the corresponding representation error. This provides a ℓ_0 affinity graph that has interesting properties of long range and sparsity, and a suitable graph cut yields a segmentation. Experimental results on the BSD database demonstrate that our method provides perfectly semantic regions even with a constant segmentation number, but also that very competitive quantitative results are achieved.

Keywords: Image segmentation, sparse representation, ℓ_0 affinity graph, spectral clustering.

1   Introduction

Image segmentation is a fundamental low-level image processing problem, which plays a key role in many high-level computer vision tasks, such as scene understanding [8], object recognition [9], etc. In the literature, unsupervised spectral segmentation algorithms have been intensively studied [15] [3] [10], and many works focus more particularly on constructing a reliable affinity graph [7] [11].

Clearly, the quality of the segmentation results depend on the choice of a particular affinity graph, which depends on the neighborhood topology and pairwise affinities between nodes, which can be pixels or superpixels (i.e. groups of pixels or pixel features). Due to computer storage limit and computational complexity of the eigenvalue problem, a basic requirement for a desirable affinity graph is sparsity. Thus, most of the existing benchmark algorithms use a predefined range of local neighborhood topology, e.g., 4-connected neighbors [10], or a fixed neighborhood radius [15] [3]. However, this kind of fixed local neighborhood topology fails to capture long-range connections and usually causes over-segmentation, see for instance the results of the method, that will be referred to as Segmentation by Aggregating Superpixels (SAS) in the sequel, proposed in [10] with 4-connected neighbors (see the third column of Fig. 1). As pointed out in [3], a larger neighborhood topology radius usually outcomes a better segmentation. Unfortunately, long-range affinity graphs built with a larger neighborhood radius produce dense graphs, thus yields a heavier computational cost.

To meet the requirements of sparsity and long range simultaneously, we propose in this paper a novel unsupervised spectral segmentation algorithm by constructing an affinity graph using ℓ_0-sparse representation, inspired by the work of Wright et al., see [19] and [18]. The so-called ℓ_0 affinity graph is constructed over a set of superpixels. The basic idea is to find the sparse representation of each superpixel over a large dictionary containing all the other superpixels by solving a ℓ_0-minimization problem. Then, considering the superpixels as the vertices of a graph, the edges are encoded in the non-zero representation coefficients issued from the optimization step, whereas the affinity between two superpixels can be derived from the corresponding representation error. Benefiting from the global searching and representation strategy of sparse representation, the derived ℓ_0 affinity graph has the characteristics of long range neighborhood topology and sparsity. Furthermore, we propose to refine the sparse representation by considering the spatial location information between superpixels as a penalty of the global searching and representation strategy.

To the best of our knowledge, there exists only one recent paper [21] which
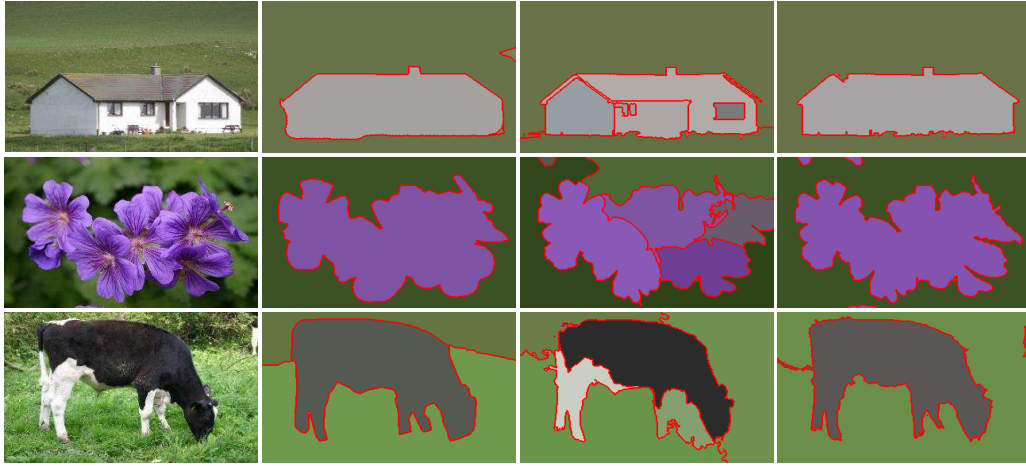
Figure 1: Illustration of the superiority of building ℓ_0 affinity graphs: for each row, from left to right, original image, ground-truth result, result of SAS [10], and result using our method with a ℓ_0 affinity graph.

makes use of sparse representation for the purpose of SAR image segmentation. As compared to that technique, the proposed method proposes to operate over multi-scale super-pixels through ℓ_0 and makes use of both refined reconstruction error and distance penalty for the construction of the affinity graph. This results in an effective method for a more general task of segmenting natural images from a large image dataset.

To improve the discriminative power of the ℓ_0 graph, both mean value in Lab color space (mLab) and Color Local Binary Pattern(CLBP) features [22] are used to build the graph. The final ℓ_0 affinity graph is obtained by merging multiple ℓ_0 graphs built over different features and different superpixel scales. Then, it is used to build a bipartite graph for final image segmentation as introduced in SAS [10].

Comparing to the existing benchmark spectral segmentation algorithms, the proposed method has the following two advantages:

1. Benefiting of the characteristics of the ℓ_0 graph, our algorithm outcomes semantic segmentation results. As shown in the last column of Fig.1, the proposed algorithm can provide meaningful segmentation results, in particular, the whole object can be segmented correctly even when there are significant color variations within the object (e.g., the cow image).

2. Most of the existing algorithms require, for each image, a careful and manual tuning of the number of segments K (usually from 2 to 40 [7]) in order to obtain

3

a desirable result. In contrast, our algorithm can produce meaningful results with $K = 2$ for most images in the Berkeley Segmentation Database (BSD) [1], which is more realistic in practical applications.

The organization of the paper is as follows: in Section 2 we introduce the proposed method and the construction of the ℓ_0 affinity graph, and we present in Section 3 comparison experiments with SAS and other methods on the BSD.

2 Construction of the graph and segmentation

2.1 Extracting multi-features over multi-scale superpixels

As pointed in [10], superpixels generated by different methods with varying parameters can capture various and multiscale visual contents of a natural image. By superpixel, we mean here a connected maximal region in a segmented image. As shown in Fig. 2, we first oversegment the input image into multiscale superpixels with either the Mean Shift algorithm (MS) [2] or the Felzenszwalb-Huttenlocher (FH) graph-based method [5], and using the same parameters as in the SAS algorithm [10]. Then, to obtain a discriminative affinity graph, we compute for each superpixel different features. Actually, any kind of region-based feature can be used. In this paper, we consider two types of features: mean value in Lab color space (mLab), and CLBP. Color is a very basic yet powerful cue to distinguish objects, whereas LBP is robust to monotonic light changes and can be used to capture texture characteristics. The parameters of these features will be introduced in section 3.1.
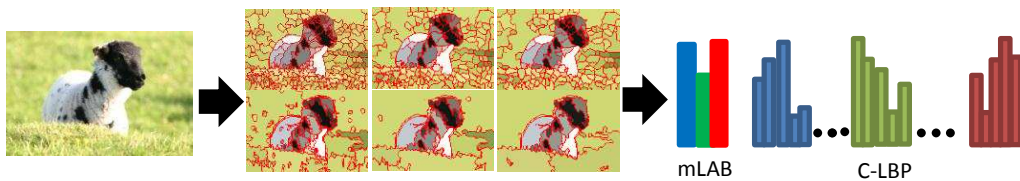

Figure 2: Illustration of extracting multi-features over multi-scale superpixels.

4

2.2 Construction of the ℓ_0 affinity graph

Let I denote an oversegmented image obtained from the initial image at the previous step, either by Mean Shift (MS) or Felzenszwalb-Huttenlocher (FH) approach. We denote as $S = \{s_i\}_{i=1}^{N}$ the collection of its superpixels, i.e. individual regions. For each superpixel s_i, $x_i^{mLab} \in \mathbb{R}^3$ and $x_i^{CLBP} \in \mathbb{R}^{256}$ are single feature normalized vectors extracted from s_i and associated with mLab and CLBP, respectively. Therefore, if N denotes the number of regions in the oversegmented image, we get two feature vectors $\{x_1^{mLab}, \cdots, x_N^{mLab}\}$ and $\{x_1^{CLBP}, \cdots, x_N^{CLBP}\}$. For each feature vector $\{x_1, \cdots, x_N\}$, we define the sparse representation dictionary $D = [x_1, \cdots, x_N] \in \mathbb{R}^{m \times N}$ ($m = 3$ for mLab and $m = 256 \times 3$ for CLBP, or possibly less by dimension reduction, see Section 3). For each $i \in \{1, \cdots, N\}$, we consider the following ℓ_0-minimization problem

$$\hat{\alpha}^i = \operatorname{argmin}_{\alpha} \{ \|x_i - D\alpha\|_2^2, \ \alpha \in \mathbb{R}^N, \ \|\alpha\|_0 \leq L, \ \alpha_i = 0 \} \tag{1}$$

where $\alpha \in \mathbb{R}^N$ runs over all sparse representation vectors, the ℓ_0 norm $\|\alpha\|_0$ is the number of non-zero coefficients in α, and the parameter L controls the sparsity of the representation. In other words, the vector $\hat{\alpha}^i$ is associated with the best representation of x_i, in the ℓ_2 nom, as a linear combination of at most L elements among $x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_N$. This vector can be computed with the Orthogonal Matching Pursuit (OMP) algorithm [14], and it provides a link between the superpixel s_i (associated with the feature x_i) and the other superpixels.

Clearly, this representation is only feature-based and does not really incorporate spatial constraints, which may be a drawback for segmentation purposes where objects are supposed to be connected. We therefore consider an additional step where we discard, in the representation above, the farthest superpixels (i.e., far from a spatial viewpoint).

In practice, for the case $L = 3$ that appeared in our experiments to yield good results, if more than two superpixels are selected by Eq. (1) to represent s_i, we find the farthest one from s_i according to the distance between centroids, and we remove it from the sparse representation. In the case $L > 3$, other methods can be used, for instance thresholding above the average distance or a fraction of it. If $k_1, \cdots, k_h$ denote the indices of the removed superpixels, we recompute the sparse representation in Eq. (1) in a very constrained way, i.e. by restricting to all α such that $\alpha_j = 0$ whenever $\hat{\alpha}_j^i = 0$ and, in addition, $\alpha_{k_1} = \cdots = \alpha_{k_h} = 0$. We denote as $\mathring{\alpha}^i$ the updated sparse representation vector, and we define

$$r_{ij} = \|x_i - \mathring{\alpha}_j^i x_j\|_2^2 \tag{2}$$

5

Finally, the similarity coefficient w_{ij} between superpixel s_i and superpixel s_j is defined as

$$w_{ij} = \begin{cases} 1 & \text{if } i = j \\ 1 - (r_{ij} + r_{ji})/2 & \text{if } i \neq j. \end{cases} \quad (3)$$

and we denote as $W = (w_{ij})$ the similarity matrix.

All steps above are applied to both feature vectors $\{x_1^{mLab}, \cdots, x_N^{mLab}\}$ and $\{x_1^{CLBP}, \cdots, x_N^{CLBP}\}$, and yield two matrices W^{mLab} and W^{CLBP}, and therefore two ℓ_0 graphs $G^{mLab} = \{S, W^{mLab}\}$ and $G^{CLBP} = \{S, W^{CLBP}\}$. These two graphs can be merged into a single graph $G = \{S, W\}$ where $W = (w_{ij})$ is defined by

$$w_{ij} = \sqrt{(w_{ij}^{mLab})^2 + (w_{ij}^{CLBP})^2}. \quad (4)$$

So far, we dealt with a single oversegmented image only. In order to benefit of the advantages of using various oversegmented images, as in the SAS method, the same procedure can be applied to each oversegmented image I_k, $k = 1, \cdots, M$ and yields the graphs $G_k = \{S, W_k\}_{k=1}^M$. The final ℓ_0-affinity graph $G = \{S, W\}$ is obtained by a simple concatenation, i.e. $W = diag(W_1, W_2, \cdots, W_M)$.

2.3 Transfer cuts and image segmentation

To perform image segmentation, we use the Transfer Cuts method (Tcuts) [10], that has proven to be fast and efficient. First, we build a bipartite graph over the input image I and its superpixel set S. Recall that our final fused ℓ_0 affinity graph $G = \{S, W\}$ is constructed over the superpixel set S. The bipartite graph also incorporates the relationship information between pixels and superpixels, and is defined as $G_B = \{U, V, B\}$, where $U = I \cup S$, $V = S$, and $B = \begin{bmatrix} W_{IS} \\ W_{SS} \end{bmatrix}$, with $W_{IS} = (b_{ij})_{|I| \times |V|}$, and $b_{ij} = $ a positive constant b if pixel i belongs to superpixel j, 0 otherwise (in our experiments, we set $b = 10^{-3}$). W_{SS} is the affinity graph between superpixels computed in section 2.2. The Tcuts method yields a partition of the bipartite graph into K clusters. More precisely, it provides the bottom K eigenpairs $\{\lambda_i, f_i\}_{i=1}^K$ of the following generalized eigenvalue problem over superpixels only:

$$L_V \mathbf{f} = \lambda D_V \mathbf{f}, \quad (5)$$

where $L_V = D_V - W_V$, $D_V = diag(B^\top \mathbf{1})$, and $W_V = B^\top D_U^{-1} B$, $D_U = diag(B\mathbf{1})$.

3 EXPERIMENTAL RESULTS

3.1 Database and parameter settings

We evaluate our method on a standard benchmark image segmentation database, the BSD [1]. The BSD contains 300 images, each one provided with at least 4 or 5 ground truth segments labeled by several people. Four measurements are used for quantitative evaluation: Probabilistic Rand Index (PRI) [16], Variation of Information (VoI) [13], Global Consistency Error (GCE) [12], and Boundary Displacement Error (BDE) [6]. A segmentation result is better if PRI is higher and the other three ones are lower.

For feature extraction, we use the LBP(1,8) operator in the RGB color space, and the feature dimension is reduced from 256×3 to 64 by PCA. For building the ℓ_0 graph, we use the Orthogonal Matching Pursuit (OMP) algorithm [14] to solve Eq. (1), in which the sparsity number $L = 3$ is used for all the experiments.

We organize our experimental results as follows: first, we show some visual comparison results with SAS; then, quantitative comparison with SAS and other algorithms are listed; finally, we show more visual examples of our method with a fixed segment number $K = 2$.

3.2 Visual comparison with SAS

Our work follows a similar, yet not identical, strategy as the SAS algorithm, i.e., building a bipartite graph over multiple superpixels and pixels, then use Tcuts for image segmentation. The main difference between the two methods is the affinity graph construction. In SAS, 4-connected neighborhoods of superpixels are used, and the pairwise superpixel similarity is computed by the Gaussian weighted Euclidean distance in the color feature space. In our method, we build a ℓ_0 affinity graph using sparse representation over multiple types of features and multi-scale superpixels, making the constructed graph having the characteristics of long range neighborhood topology, yet with sparsity and high discriminative power.

In this section, we show some visual comparison results with SAS. As shown in Fig.3, four groups of visual examples are reported: the first two columns display the results of SAS and the last two columns show our results. Notice that the results of SAS are the best results reported by the authors, where the segmentation number K for the owl, leopard, people, and landscape are 5, 4, 40, and 9 respectively. For our method, all the results are obtained by setting $K = 2$.

For the first example, although the owl is in a highly clustered background
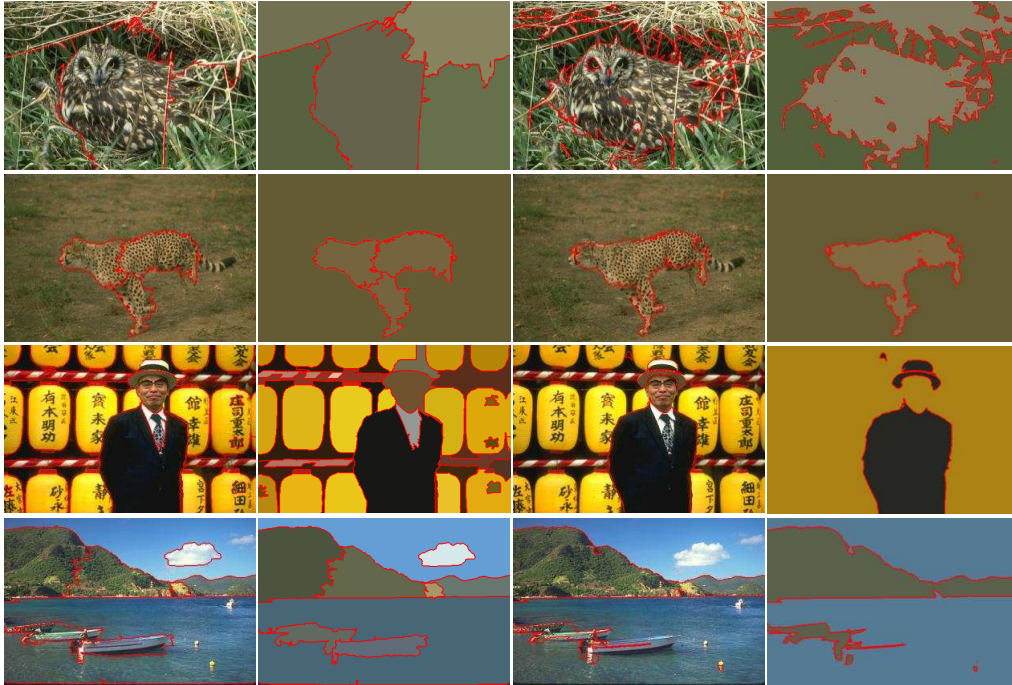
Figure 3: Four examples of visual comparison with SAS: first two columns: results of SAS, last two columns: results with our method.

and has itself strong color variations, our method segments it correctly while the segmentation provided by SAS is not meaningful. For the second example, the leopard texture is very similar to the background. The SAS algorithm oversegments the leopard into three parts. In contrast, our method provides the whole body of the leopard. Similar results are achieved for the third example: in contrast with SAS where K has been carefully tuned at $K = 40$, our method can segment correctly the main object (i.e., the people) by setting $K = 2$. In the last example as well, SAS over-segments the hill into several parts.

3.3   Quantitative comparison with SAS and other algorithms

In this section, we report quantitative comparison with SAS and other standard benchmarks: Ncut [15], JSEG [20], MNcut [3], NTP [17], SDTV [4], LFPA [7] and SAS [10]. The results are shown in Table 1, where we highlight the best result of each measurement in bold. The average scores of the benchmark methods are

8

collected from [10] and [7]. We can see that our method ranks in the first place with PRI and BDE, and is competitive with others in terms of VoI and GCE. However, all these scores are collected by tuning K manually for each image and choosing the best results, which is unrealistic in practical applications (for our method, we set K from 2 to 40). Thus, to demonstrate the obvious advantage of our method related to K, we compare the average scores of SAS and our method by fixing $K = 2$ for all images on the BSD. We can see that in this case, our method outperforms SAS with PRI, GCE and BDE (the gain being really significant for BDE).

Methods	PRI↑	VoI↓	GCE↓	BDE↓
NCut	0.7242	2.9061	0.2232	17.15
JSEG	0.7756	2.3217	0.1989	14.40
MNCut	0.7559	2.4701	0.1925	15.10
NTP	0.7521	2.4954	0.2373	16.30
SDTV	0.7758	1.8165	**0.1768**	16.24
LFPA	0.8146	1.8545	0.1809	12.21
SAS	0.8319	**1.6849**	0.1779	11.29
Ours	**0.8355**	1.9935	0.2297	**11.1955**
SAS (K=2)	0.6197	**2.0119**	0.1106	42.2877
Ours (K=2)	**0.6270**	2.0299	**0.1050**	**23.1298**

Table 1: Quantitative comparison of our method with other state-of-the-art methods over BSD.

3.4 More visual examples

To demonstrate the advantage of our algorithm in practical applications, we show more visual segmentation results of our method with $K = 2$. As can be seen in Fig. 4, all the objects are correctly segmented even in the following cases where: 1) The detected object is quite tiny (as seen in the first two rows); 2) Multiple objects are needed to segment in the same image (as in both middle rows); 3) The colors of background and object are quite similar (as in both last rows).
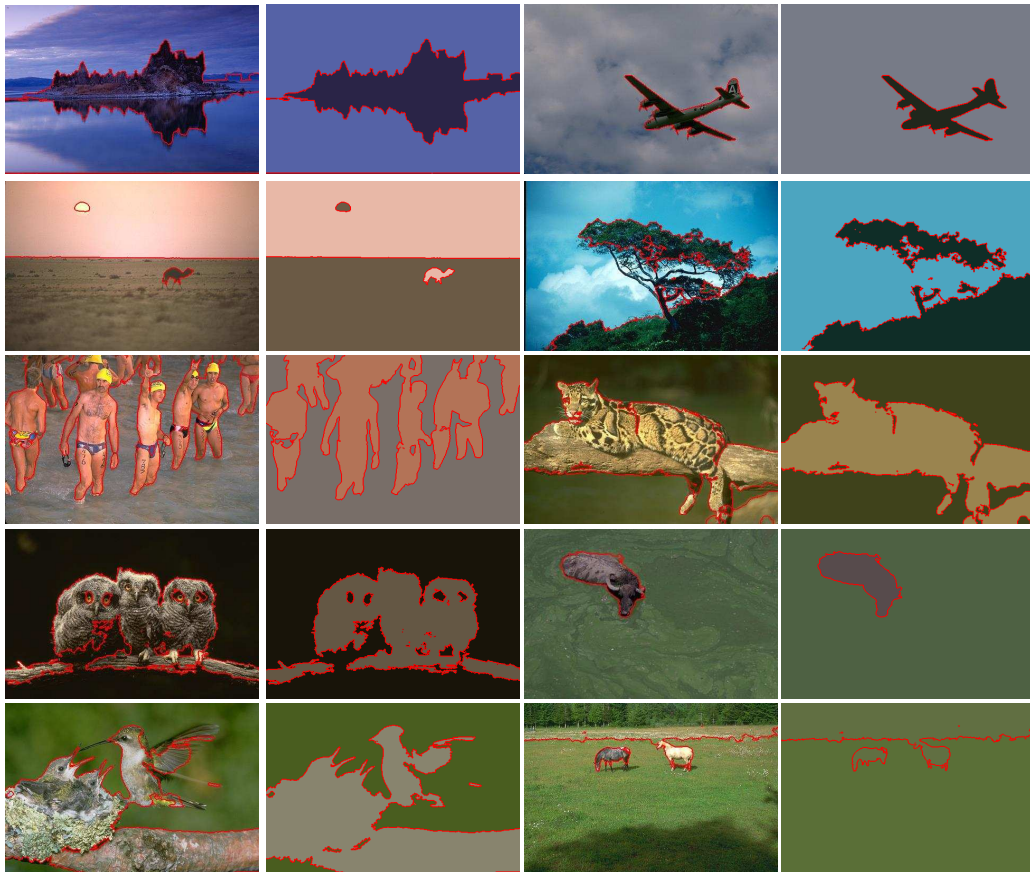
Figure 4: More visual results of our method with $K = 2$.

4 CONCLUSION

We proposed a graph-cut method for unsupervised image segmentation based on a ℓ_0 affinity graph using sparse representation. By solving several ℓ_0 minimization problems, the neighborhood topology structures and the affinities among super-pixels can be derived simultaneously. In addition, the ℓ_0 affinity graph has nice properties of sparsity and long range neighborhood topology. The ℓ_0 graph is then refined by slightly forcing the spatial locality of the representation. The discriminative power of the ℓ_0 affinity graph is then enhanced by fusing mLab and CLBP features over multi-scale superpixels. Experimental results on the BSD database show that our method yields very competitive qualitative and quantitative segmentation results compared to other state-of-the-art methods.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *PAMI*, 33(5):898–916, 2011.

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

[3] T. Cour, F. Bénézit, and J. Shi. Spectral segmentation with multiscale graph decomposition. In *CVPR*, pages 1124–1131, 2005.

[4] M. Donoser, M. Urschler, M. H., and H. Bischof. Saliency driven total variation segmentation. In *ICCV*, pages 817–824, 2009.

[5] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, 2004.

[6] J. Freixenet, X. Muñoz, D. Raba, J. Martí, and X. Cufí. Yet another survey on image segmentation: Region and boundary information integration. In *ECCV*, pages 408–422, 2002.

[7] T. H. Kim, K. M. Lee, and S. U. Lee. Learning full pairwise affinities for spectral segmentation. In *CVPR*, pages 2101–2108, 2010.

[8] M. P. Kumar and D. Koller. Efficiently selecting regions for scene understanding. In *CVPR*, pages 3217–3224, 2010.

[9] Y. J. Lee and K. Grauman. Object-graphs for context-aware category discovery. In *CVPR*, pages 1–8, 2010.

[10] Z. Li, X. Wu, and S. Chang. Segmentation using superpixels: A bipartite graph partitioning approach. In *CVPR*, pages 789–796, 2012.

[11] H. Liu, X. Yang, L. J. Latecki, and S. Yan. Dense neighborhoods on affinity graph. *International Journal of Computer Vision*, 98(1):65–82, 2012.

[12] D. R. Martin, C. Fowlkes, D. Tal, , and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, pages 416–425, 2001.

[13] M. Meila. Comparing clusterings: an axiomatic view. In *ICML*, pages 577–584, 2005.

[14] Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *27th Asilomar Conference on Signals, Systems and Computers*, pages 40 –44, 1993.

[15] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 22(8):888–905, 2000.

[16] R. Unnikrishnan, C. Pantofaru, and M. Hebert. Toward objective evaluation of image segmentation algorithms. *PAMI*, 29(6):929–944, 2007.

[17] J. Wang, Y. Jia, X. Hua, C. Zhang, and L. Quan. Normalized tree partitioning for image segmentation. In *CVPR*, 2008.

[18] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan. Sparse representation for computer vision and pattern recognition. *Proceedings of the IEEE*, 98(6):1031–1044, 2010.

[19] J. Wright, A.Y. Yang, A. Ganesh, S.S.Sastry, and M.Yi. Robust face recognition via sparse representation. *PAMI*, 31(2):210 –227, 2009.

[20] D. Yining and B. S. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *PAMI*, 23(8):800–810, 2001.

[21] X.G. Zhang, Z.L.Wei, J. Feng, and L.C. Jiao. Sparse representation-based spectral clustering for sar image segmentation. *SPIE*, pages 08–06, 2011.

[22] C. Zhu, C. Bichot, and L. Chen. Multi-scale color local binary patterns for visual object classes recognition. In *ICPR*, pages 3065–3068, 2010.