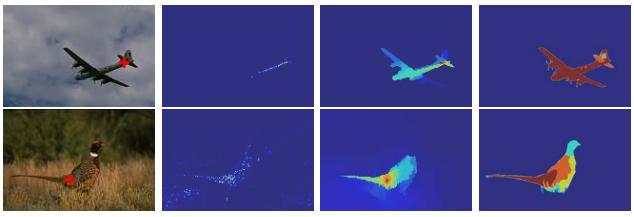
Learning Full Pairwise Affinities for Spectral Segmentation

Tae Hoon Kim Kyoung Mu Lee Sang Uk Lee
Dept. of EECS, ASRI, Seoul National University, 151-742, Seoul, Korea
th33@snu.ac.kr kyoungmu@snu.ac.kr sanguk@ipl.snu.ac.kr

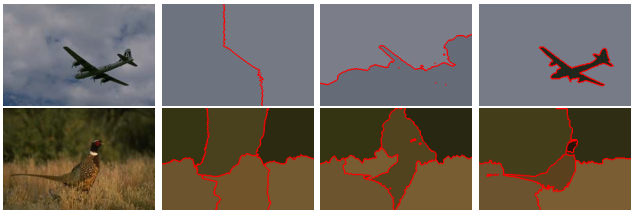
Abstract

This paper studies the problem of learning a full range of pairwise affinities gained by integrating local grouping cues for spectral segmentation. The overall quality of the spectral segmentation depends mainly on the pairwise pixel affinities. By employing a semi-supervised learning technique, optimal affinities are learnt from the test image without iteration. We first construct a multi-layer graph with pixels and regions, generated by the mean shift algorithm, as nodes. By applying the semi-supervised learning strategy to this graph, we can estimate the intra- and inter-layer affinities between all pairs of nodes together. These pairwise affinities are then used to simultaneously cluster all pixel and region nodes into visually coherent groups across all layers in a single multi-layer framework of Normalized Cuts. Our algorithm provides high-quality segmentations with object details by directly incorporating the full range connections in the spectral framework. Since the full affinity matrix is defined by the inverse of a sparse matrix, its eigendecomposition is efficiently computed. The experimental results on Berkeley and MSRC image databases demonstrate the relevance and accuracy of our algorithm as compared to existing popular methods.



(a) (b) (c) (d)

Figure 1. Pixel affinity images. (a) Image with one pixel selected. (b)-(d) Similarities between that pixel and all other pixels in the image, where red represents greater. They are obtained by the conventional color+boundary affinity model used in MNCut [6], our proposed measure, and the groundtruth affinity function from human annotations, respectively.



(a) (b) (c) (d)

Figure 2. Introducing a spectral segmentation. (a) Original image. (b)-(d) Segmentation results by NCut [27], MNCut [6], and our algorithm with the boundaries drawn in red, respectively.

1. Introduction

Unsupervised image segmentation is a fundamental but challenging problem in computer vision. It automatically partitions an image into coherent groups without any prior knowledge. Recently, the main research directions are focused on deterministic annealing [21], stochastic clustering [13][26], mode-seeking methods like mean shift [5][4], mixture model [23][24], rate distortion [22], graph theoretic approaches [27][9][33], and other variational methods [20][8]. Most studies generally defined the segmentation problem as finding the labeling of an image that minimizes a specific energy function. However simple appearance based methods like the mean shift image segmentation algorithm [5] are still generally used to make the over-segmentations with detailed boundaries.

Several problems make unsupervised image segmentation difficult in natural images, such as finding the faint object boundaries and separating the highly cluttered background. To solve these problems, several recent works focused on integration of local grouping cues across long ranges of spatial connections between pairwise pixels and produced impressive segmentation results. Since using the full range connections would require a great computational cost, two conventional approaches to propagate local grouping cues across larger image areas recently became popular. The first approach is a multi-scale image segmentation [25][1][31][10][6] to combine both coarse- and fine-level details. The multi-scale frameworks give efficient approximation to incorporate long-range connections with low complexity. However they typically fail to detect fine-

level details along object boundaries due to coarsening error. The second approach is to define the segmentation as the grouping of non-overlapping regions, instead of pixels [29][22]. It is inspired by the hard constraint whereby pixels in a particular region should have the same label. This region-based segmentation has the benefit of using more informative features extracted from the pixels inside the regions, as well as transferring local information to a larger image area with connections across regions. If, however, the regions are not consistent with the object boundaries, there are radical difficulties in obtaining the exact solutions. In fact, such situations often arise in natural images.

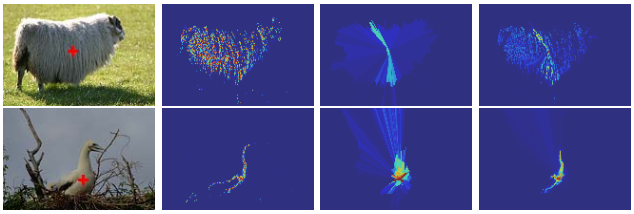
Unlike these conventional approaches, we propose a new segmentation algorithm to directly incorporate a full range of pairwise affinities in the spectral framework. The key contributions of our algorithm are as follows.

1. We introduce a new affinity model for image segmentation. Unlike the conventional affinity models learnt from a large database with manually segmented images [11], or from only local properties of the neighboring pixels [16][6], we propose to use the relevance scores, learnt from the test image by semi-supervised learning [3][34][19], as graph affinities. We first construct a multi-layer graph with pixels and regions generated by the mean shift algorithm [5]. After we assign each node and all other nodes as labeled and unlabeled data respectively, the affinities are efficiently estimated by applying the semi-supervised strategy to this graph. Fig. 1 shows that compared with the general affinity model [6], which heuristically unifies region and boundary information, our proposed affinity model intuitively gives better scores in the highly textured images.

2. We follow the framework of spectral segmentation [30] [27][32][6], a major trend in well-established graph theoretic approaches. In a single multi-layer framework of Normalized Cuts, the proposed full affinities are used to simultaneously cluster all pixel and region nodes into visually coherent groups across all layers.

3. Our algorithm produces high-quality segmentation results by considering all intra- and inter-layer affinities in the spectral framework. Moreover its computation is very efficient by the eigen-decomposition of a sparse matrix. Fig. 2 shows that our algorithm produces much better segmentations with object details than other spectral segmentation methods such as Normalized Cuts (NCut) [27] and Multiscale NCut (MNCut) [6] in natural images.

The rest of the paper is organized as follows. We first review the conventional affinity models in Section 2, and then discuss our algorithm for learning the full pairwise affinities on a multi-layer graph in Section 3. In Section 4, we present the efficient multi-layer spectral segmentation based on these full affinities. The experiments are given in Section 5. Finally, we discuss our approach in Section 6.



(a) (b) (c) (d)

Figure 3. Examples of the conventional affinities encoding local grouping cues. (a) Image with a user-given point. (b)-(d) Similarities between that pixel and all other pixels by (1)-(3), respectively.

2. Conventional Affinity Model

In a graph-theoretic approach, the overall segmentation quality depends mainly on the graph affinities. Therefore, to produce high-quality segmentation results with object-level details, it is important to define the graph affinities gained by integrating local grouping cues.

Now we review the conventional affinity models. Most existing approaches directly define the edge weights W as graph affinities in a graph $G = (V, E)$, where the nodes V consists of N pixels. The pixels within a fixed distance are connected by an edge in E. To compute the weights W, color and boundary cues are generally used as local grouping cues. The conventional models can be defined, according to which grouping cues are used as follows.

Color Cue. Close-by pixels with similar colors are likely to belong to the same image region. The color-based affinity model w_{ij}^{C} is usually formulated as follows [27][2].

$$w_{ij}^{\mathrm{C}} = exp(-\theta_x \|x_i - x_j\|^2 - \theta_g \|g_i - g_j\|^2), \qquad (1)$$

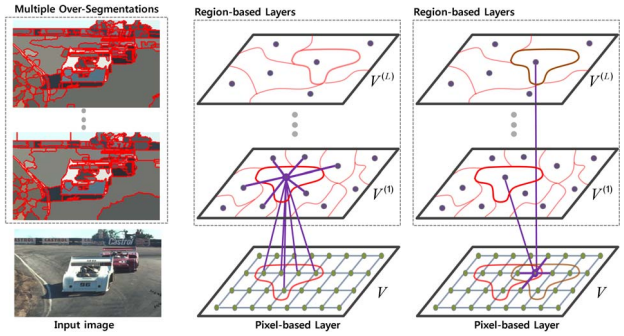
where x_i and g_i denote position and color of a pixel i respectively. Fig. 3(b) shows that the connecting pixels by color are useful when linking disjointed object parts, but make errors if background has the similar color distribution with the object parts.

Boundary Cue. Edgeness is one important boundary cue to detect a potential object boundary. The boundary-based affinity model w_{ij}^{B} is commonly formulated by measuring the magnitude of image edges between two pixels, like [16]:

$$w_{ij}^{\mathrm{B}} = exp(-\max_{i' \in \overline{ij}} \theta_f \|f_{i'}\|^2), \qquad (2)$$

where $\overline{ij}$ is a straight line joining two pixels i and j, and f_i is the edge strength of a pixel i. This boundary-based model is particularly useful when background clutter has a similar color with the object body, like in Fig. 3(c). However, since it uses the only edgeness along the straight line between two pixels without considering all possible paths, texture edges often disturb the long-range affinity estimation.

Combining Color & Boundary Cues. To make the appropriate affinity model in all natural images, it is helpful to



(a) (b) (c)

Figure 4. Our multi-layer graph model. In (a), the graph nodes V^* consist of pixels V and regions $\{V^{(l)}\}_{l=1,...,L}$, generated by varying the parameters of the mean shift algorithm [5]. An undirected edge E^* represents the relation between a pair of nodes. (b) and (c) show the examples of edges (violet lines) connected to one region and to one pixel, respectively.

combine these two grouping cues. They can be simply combined with a parameter α for the combined affinity model w_{ij}^{M} like in [6] as follows.

$$w_{ij}^{\mathrm{M}} = \sqrt{w_{ij}^{\mathrm{C}} \times w_{ij}^{\mathrm{B}}} + \alpha w_{ij}^{\mathrm{B}}, \tag{3}$$

where w_{ij}^{C} in (1) and w_{ij}^{B} in (2). Fig. 3(d) shows some examples of the affinities gained by this combined model. This model still has some weakness in the long-range affinity estimation, since it is formulated by naively mixing two simple color and boundary affinity models.

3. Proposed Full Affinity Model

To estimate the full pairwise affinities by integrating local grouping cues extracted from the entire image, we propose to use semi-supervised learning [3][34][19], which is a very successful technique to learn the global relevance scores between labeled and unlabeled data in a sparse graph. We first design a sparsely connected graph with multi-layers for efficiently combining local grouping cues. We then define the relevance scores between all pairs of these graph nodes by semi-supervised learning [34] as graph affinities.

3.1. Graphical Model

We design a multi-layer graph $G^* = (V^*, E^*)$, where the nodes $V^* = \left\{V \cup V^{(1)} \cup ... \cup V^{(L)}\right\}$ are a set of pixels V and the sets of regions $\left\{V^{(l)}\right\}_{l=1,...,L}$ and the edges E^* are the undirected links as shown in Fig. 4. The node subset $V^{(l)}$ at the l-th region-based layer corresponds to the N_l non-overlapping regions generated by the mean shift algorithm [5]. The use of L over-segmentations by varying the mean shift parameters is helpful in reducing the errors of regions that may contain several object parts [15].

The edges E^* are connected by different criteria according to the node types. An undirected edge $e_{ij}^* \in E^*$ exists, **1)** if two pixels $i, j \in V$ are adjacent, usually 4 neighborhoods, **2)** if two regions $i, j \in V^{(l)}$ share a common boundary at same layer, or **3)** if one pixel $i \in V$ is included in its corresponding region $j \in V^{(l)}$. This edge e_{ij}^* has the following weight $w_{ij}^* \in W^*$:

$$w_{ij}^* = \begin{cases} exp\left(-\theta_g \left\|g_i - g_j\right\|\right) & \text{if } i,j \in V \\ exp\left(-\theta_g \left\|\bar{g}_i - \bar{g}_j\right\|\right) & \text{if } i,j \in V^{(l)} \\ \gamma & \text{if } i \in V, j \in V^{(l)} \end{cases} \tag{4}$$

where $\bar{g}_i$ denotes the mean color of inner pixels of the region i and θ_g is a constant that controls the strength of the weight. The intra-layer weights encode color cues in Lab color space, similar to (1). Since the relationship between the pixels in a given region is more emphasized from the inter-layer connections between that region and its inner pixels in our graph, the region itself gives the implicit boundary information as boundary cue, instead of image edges such as in (2). Therefore the trade-off between our color and boundary cues is captured by a parameter γ.

3.2. Learning Full Affinities

Let $\mathbf{\Pi} = [\pi_{ij}]_{\hat{N}\times\hat{N}}$, where $\hat{N} = N + \sum_{l=1}^{L} N_l$ is the total number of nodes, be the affinity matrix. To estimate the affinities $\mathbf{\Pi}$, we borrow ideas from semi-supervised learning. We assign each node and all other nodes as labeled and unlabeled nodes, and then find the relevance scores between labeled and unlabeled nodes by semi-supervised learning in our graph G^*. In this work, we use these relevance scores as the full affinities between that node and all other nodes. We compare two functions '$\mathcal{F}_1$' and '$\mathcal{F}_2$' in the work of Zhou et al. [34] for semi-supervised learning, since they can make the symmetric affinity matrix whose elements are sufficiently smooth with respect to the intrinsic graph structure for spectral segmentation.

1) '$\mathcal{F}_1$': The first affinity vector $\vec{\pi}_m^{\mathrm{n}} = [\pi_{im}^{\mathrm{n}}]_{\hat{N}\times 1}$ of all nodes V^* from a node m can be formulated:

$$\vec{\pi}_m^{\mathrm{n}} = c\left(\mathbf{I}^* - (1-c)\mathbf{S}^*\right)^{-1}\vec{b}_m, \tag{5}$$

where the identity matrix $\mathbf{I}^*$ of size $\hat{N}$, and the normalized weight matrix $\mathbf{S}^* = \mathbf{D}^{*-\frac{1}{2}}\mathbf{W}^*\mathbf{D}^{*-\frac{1}{2}}$, where $\mathbf{W}^* = [w_{ij}^*]_{\hat{N}\times\hat{N}}$ and $\mathbf{D}^* = \mathrm{diag}([d_1^*, ..., d_{\hat{N}}^*])$, $(d_i^* = \sum_{j=1}^{\hat{N}} w_{ij}^*)$. The vector $\vec{b}_m = [b_{im}]_{\hat{N}\times 1}$ is the $\hat{N}$-dimensional indicating one with $b_{im} = 1$ if labeled node $i = m$ and 0 otherwise. This function can be interpreted as the solution to minimize the following cost function $\mathcal{E}_m^{\mathrm{n}}$:

$$\mathcal{E}_m^{\mathrm{n}} = \sum_{i,j=1}^{\hat{N}} w_{ij}^* \left| \frac{\pi_{im}^{\mathrm{n}}}{\sqrt{d_i^*}} - \frac{\pi_{jm}^{\mathrm{n}}}{\sqrt{d_j^*}} \right|^2 + \mu \sum_{i=1}^{\hat{N}} |\pi_{im}^{\mathrm{n}} - b_{im}|^2, \tag{6}$$

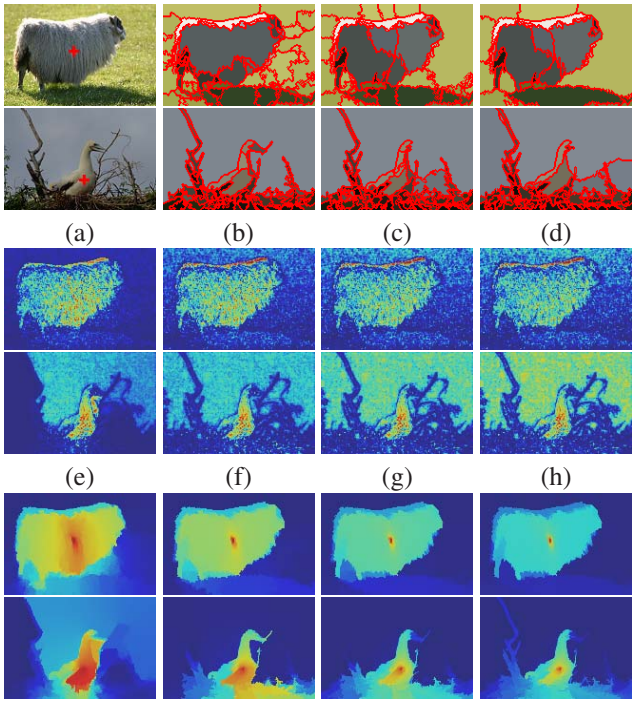


Figure 5. Examples of affinity images by semi-supervised learning [34]. (a) Image with one pixel m selected. (b)-(d) Three over-segmentations, obtained by varying the parameters of the mean shift algorithm [5]. (e)-(h) and (i)-(l) Affinity images $\vec{\pi}^{\mathrm{n}}_m$ in (5) by 'F_1' and $\vec{\pi}^{\mathrm{u}}_m$ in (7) by 'F_2' with none, one (b), two (b)-(c), and three (b)-(d) over-segmentations, respectively.

where $\mu = \frac{c}{1-c}$, $(0 < c < 1)$. The first term in $\mathcal{E}^{\mathrm{n}}_m$ is the smoothness constraint that the good affinities should not change too much between neighboring nodes. However, since this term is defined as local variation by the difference of the normalized affinities between two end nodes of each edge, smooth variation between close-by nodes cannot be guaranteed, as shown in Fig. 5(e)-(h). The second term in $\mathcal{E}^{\mathrm{n}}_m$ is the fitting constraint that the good affinities should not change too much from the initial label assignment. A positive parameter μ (or c) specifies the relative amount of the smoothness and fitting constraints.

2) '$\mathcal{F}_2$'**:** The second affinity vector $\vec{\pi}^{\mathrm{u}}_m = [\pi^{\mathrm{u}}_{im}]_{\hat{N}\times 1}$ of all nodes V^* from a node m are defined to be

$$\vec{\pi}^{\mathrm{u}}_m = c\left(\mathbf{D}^* - (1-c)\mathbf{W}^*\right)^{-1}\vec{b}_m. \tag{7}$$

This function is the same as the solution to minimize the following cost function $\mathcal{E}^{\mathrm{u}}_m$, similar to $\mathcal{E}^{\mathrm{n}}_m$ in (6):

$$\mathcal{E}^{\mathrm{u}}_m = \sum_{i,j=1}^{\hat{N}} w^*_{ij}\left|\pi^{\mathrm{u}}_{im} - \pi^{\mathrm{u}}_{jm}\right|^2 + \mu\sum_{i=1}^{\hat{N}} d^*_i\left|\pi^{\mathrm{u}}_{im} - \frac{b_{im}}{d^*_i}\right|^2. \tag{8}$$

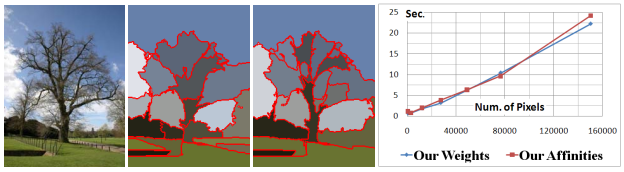


Figure 6. The running time in seconds of our algorithm as a function of image pixels N. (a) Test image. (b) and (c) 20-way partitioning by defining our weight matrix $\mathbf{W}^*$ in (4) and our estimated affinities $\mathbf{\Pi}$ in (9) as graph affinities, respectively (Size: 256×192). (d) Running time graph with respect to N.

The cost function $\mathcal{E}^{\mathrm{u}}_m$ also consists of the smoothness and fitting constraints. Compared with the smoothness constraint in $\mathcal{E}^{\mathrm{n}}_m$, the smoothness constraint in $\mathcal{E}^{\mathrm{u}}_m$ is more helpful in estimating the pairwise affinities with local smooth variation, as shown in Fig. 5(i)-(l). Moreover according to the increase in the number of over-segmentations, the intrinsic image structure with object details is well represented in the affinity image by solving the affinity function in (7).

Although the function '$\mathcal{F}_1$' in (5) has better performance than the other one '$\mathcal{F}_2$' in (7) for classification problems in [34], '$\mathcal{F}_2$' is perceptually more suited for the affinity estimation in image segmentation. We will quantitatively compare the segmentation results by these two functions in the experiments. Therefore, in this work the affinity matrix $\mathbf{\Pi} = [\pi_{ij}]_{\hat{N}\times\hat{N}}$ is defined by using $\vec{\pi}^{\mathrm{u}}_m$ in (7), as follows.

$$\mathbf{\Pi} = [\vec{\pi}^{\mathrm{u}}_1, ..., \vec{\pi}^{\mathrm{u}}_{\hat{N}}] = c\left(\mathbf{D}^* - (1-c)\mathbf{W}^*\right)^{-1}, \tag{9}$$

where the matrix $\mathbf{D}^* - (1-c)\mathbf{W}^*$ is positive definite. Since the matrix $\mathbf{D}^* - (1-c)\mathbf{W}^*$ is sparse but very large, it is difficult to directly inverse it without any approximation techniques. However, this affinity matrix $\mathbf{\Pi}$ can be efficiently used in the spectral segmentation framework.

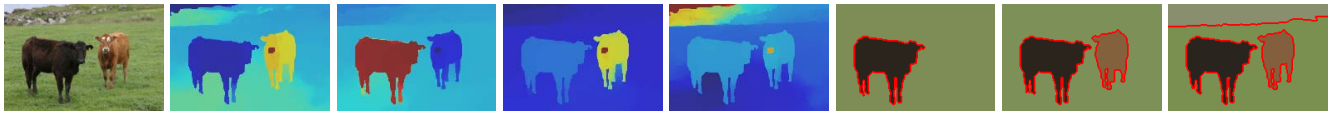
4. Multi-Layer Spectral Segmentation

We consider the image segmentation as a labeling problem in which one label $k \in \{1, ..., K\}$ is assigned to each node i. Let $\vec{y}_k = [y_{ik}]_{\hat{N}\times 1}$ be a partitioning vector with $y_{ik} = 1$ if i belongs to the k-th segment and 0 otherwise. The segmentation criterion follows the Normalized Cuts [27]:

$$\text{maximize } \mathcal{C}(\mathbf{Y}) = \frac{1}{K}\sum_{k=1}^{K}\frac{\vec{y}_k^{\,T}\mathbf{\Pi}\vec{y}_k}{\vec{y}_k^{\,T}\mathbf{D}\vec{y}_k} \tag{10}$$

subject to $\mathbf{Y}\mathbf{Y}^T = \mathbf{I}^*$, where the partitioning matrix $\mathbf{Y} = [\vec{y}_1, ..., \vec{y}_K]$ and the degree matrix $\mathbf{D} = \mathrm{diag}([\mathrm{d}_1, ..., \mathrm{d}_{\hat{N}}])$ with $d_i = \sum_{j=1}^{\hat{N}}\pi_{ij}$ in (9). The diagonal elements of $\mathbf{D}$ can be easily computed by using the following equation:

$$\begin{aligned}[d_1, ..., d_{\hat{N}}]^T &= \mathbf{\Pi}\,\vec{1}_{\hat{N}\times 1} \\ &= c(\mathbf{D}^* - (1-c)\mathbf{W}^*)^{-1}\,\vec{1}_{\hat{N}\times 1}.\end{aligned} \tag{11}$$

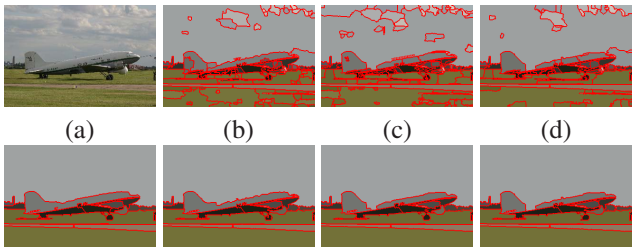


(a) (b) (c) (d) (e) (f) (g) (h)

Figure 7. Overview of our proposed method. (a) Test image. (b)-(e) The four largest eigenvectors of $\mathbf{Z}$ in (12) (the four smallest eigenvectors $\bar{\mathbf{Q}}_4$ of $\mathbf{B}$). The first eigenvector in (b) is very close to a constant vector. (f)-(h) The K-way segmentation results ($K = 2$, 3, and 4).

Algorithm 1 Our K-way Spectral Segmentation

1: Given an image, construct a multi-layer graph G^*, and calculate its weight and degree matrices $\mathbf{W}^*$, $\mathbf{D}^*$ in (4).
2: Compute $\mathbf{B} = \mathbf{D}^{\frac{1}{2}} (\mathbf{D}^* - (1-c)\mathbf{W}^*) \mathbf{D}^{\frac{1}{2}}$ in (12), where the degree matrix $\mathbf{D}$ is calculated in (11).
3: Find $\bar{\mathbf{Q}}_K$, the K smallest eigenvectors of $\mathbf{B}$
4: Discretize $\mathbf{Q}_K = \mathbf{D}^{-\frac{1}{2}} \bar{\mathbf{Q}}_K$ by K-means clustering.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 8. Example of our multi-layer segmentation. (a) Test image. (b)-(d) The three over-segmentations by varying the parameters of the mean shift algorithm. (e)-(h) Our segmentation results at each layer (a)-(d), respectively ($K = 20$).

Since the matrix $\mathbf{D}^* - (1-c)\mathbf{W}^*$ is large but very sparse, the multiplication of its inversion by a single vector typically has an efficient solution. The linear system solver implemented by the MATLAB division operator '\' (which we used in our experiments) is very efficient at finding the solution of (11).

The optimal solution of $\mathcal{C}(\mathbf{Y})$ in (10) is the subspace spanned by the K largest eigenvectors of the matrix $\mathbf{Z}$:

$$\mathbf{Z} = \mathbf{D}^{-\frac{1}{2}} \mathbf{\Pi} \mathbf{D}^{-\frac{1}{2}} = c\mathbf{B}^{-1}, \tag{12}$$

where $\mathbf{B} = \mathbf{D}^{\frac{1}{2}} (\mathbf{D}^* - (1-c)\mathbf{W}^*) \mathbf{D}^{\frac{1}{2}}$. In this work, we find the K smallest eigenvectors $\bar{\mathbf{Q}}_K$ of the very sparse matrix $\mathbf{B}$, instead of the K largest eigenvectors of the very dense and large matrix $\mathbf{Z}$ [14]. Fig. 6(b)-(c) and Fig. 6(d) show one visual comparison and the running time comparison with respect to the number of pixels, respectively. These experiments demonstrate that our segmentation approach has the similar time-complexity to NCut [27] that directly uses a very sparse matrix $\mathbf{W}^*$ as affinities, but produces more high-quality segmentation results. Our segmentation process is summarized in Algorithm 1. In Fig. 7, we present our segmentation results to illustrate the K-way segmentation. The four largest eigenvectors of $\mathbf{Z}$ in (12) are shown in Fig. 7(b)-(e). The 2-4 way partitions are shown in Fig. 7(f)-(h), respectively. Fig. 8 shows an example of our multi-layer segmentation. The three over-segmentations in Fig. 8(b)-(d) contain some incorrect regions, since the airplane has the similar color distribution to the sky in Fig. 8(a). However our algorithm gives an impressive segmentation in Fig. 8(e). Further as depicted in Fig. 8(e)-(h), the segmentation results at different layers are very similar to each other, since the label-continuity between two nodes at different layers is enforced by our segmentation criterion in (10) based on the full affinity matrix between all pixel and region nodes.

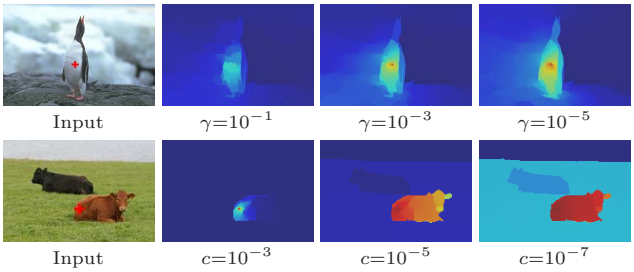


Figure 9. Examples of the affinity images with respect to the variation of the parameters γ (top row, $c = 10^{-5}$) and c (bottom row, $\gamma = 10^{-3}$) in (4) and (9), respectively.

5. Experimental Results

In this work, we initially make the regions by the mean shift algorithm [5], which needs two bandwidth parameters (h_s, h_r) for the spatial and range domains, respectively. In all experiments, we obtained three over-segmentations with the different parameters $(h_s, h_r) = \{(5,7), (7,5), (7,7)\}$. Our segmentation algorithm have two parameters: the inter-layer edge weight γ in (4) and the balanced weight c in (9). Fig. 9 shows variation of affinity images with respect to these two parameters. With a larger γ, the region consistency is more emphasized and the affinities are more discretized on the region boundaries, as shown in the top row of Fig. 9. With a smaller γ, the affinities are more over-smoothed around the user point. With a larger c, the fitting constraint is more emphasized and the only affinities within smaller distance are presented, as shown in the bottom row of Fig. 9. While with a smaller c, long-range connections around the user point are more emphasized. In this work, γ and c were empirically chosen, and we set $\gamma = 10^{-3}$ and

Methods/Score	PRI	VoI	GCE	BDE
MShift [5]	0.7958	1.9725	0.1888	14.41
NCut [27]	0.7242	2.9061	0.2232	17.15
JSEG [7]	0.7756	2.3217	0.1989	14.40
GBIS [9]	0.7139	3.3949	0.1746	16.67
MNCut [6]	0.7559	2.4701	0.1925	15.10
NTP [29]	0.7521	2.4954	0.2373	16.30
Saliency [8]	0.7758	1.8165	0.1768	16.24
TBES [22]	0.80	1.76	-	-
Our algorithm	0.8146	1.8545	0.1809	12.21
$^{\diamond}$NCut	0.7330	2.6137	0.2662	17.19
$^{\diamond}$MNCut	0.7632	2.2789	0.2234	13.17
$^{\diamond}$NCut by $\mathbf{W}^*$	0.8040	1.9537	0.1903	13.24
$^{\diamond}$Ours by '$\mathcal{F}_1$'	0.8076	1.9290	0.1880	13.05

Table 1. Quantitative comparison of our algorithm with other segmentation methods over the Berkeley database. $^{\diamond}$ means that the segment numbers are same to those of our algorithm. The best three results are highlighted in colors: Red, Green, and Blue in descending order.

$c = 10^{-5}$ for all test images.

We first demonstrate the quality of our proposed algorithm on the Berkeley image database [1]. This database contains 300 images and their corresponding ground truth data (at least four human annotations per image). For quantitative evaluation, four different measures are used: **1)** Probabilistic Rand Index (PRI) [28], which counts the number of pixel pairs whose labels are consistent between the segmentation and the ground truth, **2)** Variation of Information (VoI) [18], which measures the amount of randomness in one segmentation that cannot be explained by the other, **3)** Global Consistency Error (GCE) [17], which measures the extent to which one segmentation can be viewed as a refinement of the other, and **4)** Boundary Displacement Error (BDE) [12], which measures the average displacement error of the boundary pixels between two segmented images. The segmentation is viewed better if PRI is larger or the other three are smaller.

Comparison to Conventional Methods: In Table. 1, we report the average scores over the Berkeley database for eight other segmentation methods: Mean Shift (MShift) [5], NCut [27], JSEG [7], Graph-based Segmentation (GBIS) [9], MNCut [6], Normalized Tree Partitioning (NTP) [29], Saliency-based Segmentation (Saliency) [8], and Texture and Boundary Encoding based Segmentation (TBES) [22], taken from [8][22]. To get the optimal performance of our algorithm in Table. 1, we manually chose the different segment number for each image. We rank in the top three for VoI and GCE, and achieve the best performance for PRI and BDE. Our algorithm specially has the overwhelming BDE score. This comparison proves that our segmentations have less errors in terms of boundary displacement with respect to the ground truth. Fig. 10 shows example segmentations compared with reference methods: NTP and TBES.

[1] http://www.cs.berkeley.edu/projects/vision/bsds

Figure 10. Visual comparison of our algorithm with other methods on the Berkeley database. (a) Test images. (b)-(d) Segmentation Results by NTP [29], TBES [22], and our algorithm, respectively.

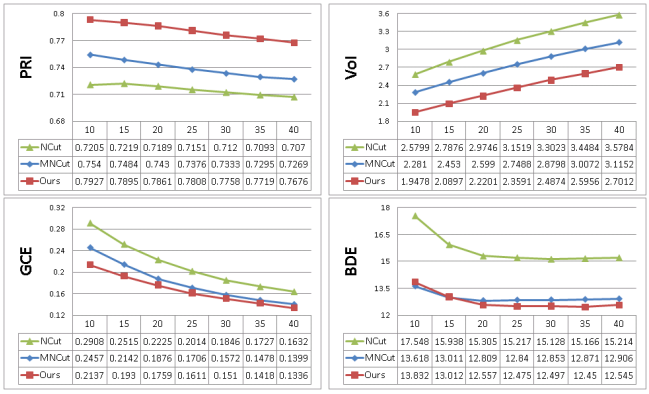


Figure 11. Our statistics on Probabilistic Rand Index (PRI), Variation of Information (VoI), Global Consistency Error (GCE), and Boundary Displacement Error (BDE) over the Berkeley database with $K = \{10, 15, ..., 40\}$, compared with NCut and MNCut.

Comparison to Spectral Methods: We first compared our performance with the average scores for NCut and MNCut, included in Table. 1, over the Berkeley database when the segment numbers of NCut and MNCut are same to those of our algorithm. To make up for this unfair comparison, our algorithm was also compared with NCut and MNCut when

the segment number K is fixed for overall images. Fig. 11 shows the average scores by varying K from 10 to 40, (in steps of five). These comparisons present that our algorithm outperforms MNCut which indirectly considers long-range connections in a multi-scale framework, as well as NCut in all cases except quite small BDE score difference at $K = 10$ in Fig. 11.

Analysis of Our Algorithm: The NCut performance based on the affinity matrix $\mathbf{W}^*$ in the multi-layer graph is included in Table. 1. It is lower than our performance. This comparison shows the advantage of using a full affinity alone. The performance of our approach which uses semi-supervised learning '$\mathcal{F}_1$' in (5) instead of '$\mathcal{F}_2$' in (7), which was used for our segmentation, is also shown in Table. 1. This experiment presents that the function '$\mathcal{F}_2$' is quantitatively more suited than '$\mathcal{F}_1$' for the affinity estimation of spectral segmentation.

Fig. 12 shows a more visual comparison of unsupervised segmentation on the MSRC object recognition database [2]. Compared with two spectral methods: NCut and MNCut, our algorithm perceptually produces high-quality segmentations with detailed object boundaries. Since we consider the full pairwise affinities, our segmentation results can detect larger textured regions as well as elongated object parts.

6. Conclusions

In this paper, the segmentation algorithm based on spectral clustering is proposed. Our work is novel in that it sheds understanding on the full affinities gained in integrating local grouping cues by using semi-supervised learning. Since these well-defined full pairwise affinities are directly used in the multi-layer spectral segmentation framework, our algorithm produces high-quality segmentation results with object details in natural images. Further, our algorithm is computationally very efficient. Our future work is to automatically estimate the number of segments K.

Acknowledgement

This research was supported in part by the IT R&D program of MKE/IITA (2008-F-030-01), and in part by the ITRC program of MKE/NIPA through 3DRC (NIPA-2009-C1090-0902-0018), Korea.

References

[1] A. Barbu and S.-C. Zhu. Graph partition by swendsen-wang cuts. In *ICCV*, 2003.

[2] Y. Boykov and G. Funka-Lea. Graph cuts and efficient n-d image segmentation. *IJCV*, 70(2):109–131, 2006.

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *WWW*, 1998.

[2] http://research.microsoft.com/en-us/projects/objectclassrecognition

[4] M. Á. Carreira-Perpiñán. Fast nonparametric clustering with gaussian blurring mean-shift. In *ICML*, 2006.

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

[6] T. Cour, F. Bénézit, and J. Shi. Spectral segmentation with multiscale graph decomposition. In *CVPR*, 2005.

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

[8] M. Donoser, M. Urschler, M. Hirzer, and H. Bischof. Saliency driven total variation segmentation. In *ICCV*, 2009.

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

[10] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the nyström method. *PAMI*, 26(2):214–225, 2004.

[11] C. Fowlkes, D. Martin, and J. Malik. Learning affinity functions for image segmentation: combining patch-based and gradient-based approaches. In *CVPR*, 2003.

[12] J. Freixenet, X. Muñoz, D. Raba, J. Martí, and X. Cufí. Yet another survey on image segmentation: region and boundary information integration. In *ECCV*, 2002.

[13] Y. Gdalyahu, D. Weinshall, and M. Werman. Stochastic image segmentation by typical cuts. In *CVPR*, 1999.

[14] G. H. Golub and C. F. V. Loan. Matrix computations, 1996. Johns Hopkins University Press.

[15] P. Kohli, L. Ladický, and P. Torr. Robust higher order potentials for enforcing label consistency. In *CVPR*, 2008.

[16] T. Leung and J. Malik. Contour continuity in region-based image segmentation. In *ECCV*, 1998.

[17] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natrual images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001.

[18] M. Meilă. Comparing clusterings: an axiomatic view. In *ICML*, 2005.

[19] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu. Automatic multimedia cross-modal correlation discovery. In *KDD*, 2004.

[20] T. Pock, A. Chambolle, D. Cremers, and H. Bischof. A convex relaxation approach for computing minimal partitions. In *CVPR*, 2009.

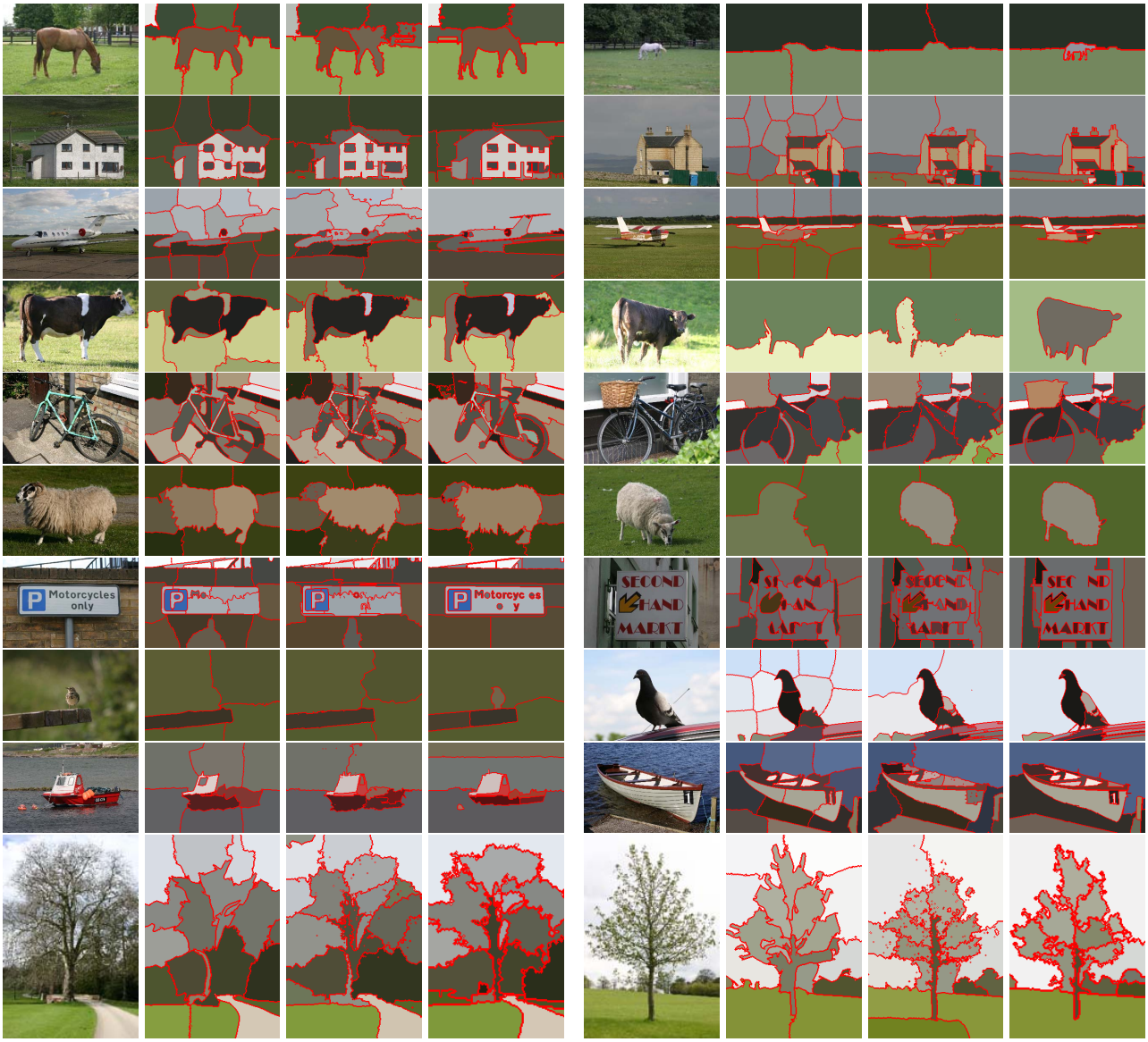
[21] J. Puzicha, T. Hofmann, and J. M. Buhmann. Non-parametric similarity measures for unsupervised texture segmentation and image retrieval. In *CVPR*, 1997.

[22] S. R. Rao, H. Mobahi, A. Y. Yang, S. S. Sastry, and Y. Ma. Natural image segmentation with adaptive texture and boundary encoding. In *ACCV*, 2009.

[23] S. Sanjay-Gopal and T. J. Hebert. Bayesian pixel classification using spatially variant finite mixtures and the generalized em algorithm. *IP*, 7(7):1014–1028, 1998.

[24] G. Sfikas, C. Nikou, and N. Galatsanos. Edge preserving spatially varying mixtures for image segmentation. In *CVPR*, 2008.

[25] E. Sharon, A. Brandt, and R. Basri. Fast multiscale image segmentation. In *CVPR*, 2000.



(a) (b) (c) (d) (a) (b) (c) (d)

Figure 12. Visual comparison of our algorithm with the conventional spectral methods on the MSRC database. (a) Test images. (b)-(d) Segmentation Results by NCut [27], MNCut [6], and our algorithm, respectively.

[26] N. Shental, A. Zomet, T. Hertz, and Y. Weiss. Learning and inferring image segmentations with the gbp typical cut algorithm. In *ICCV*, 2003.

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

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

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

[30] Y. Weiss. Segmentation using eigenvectors: a unifying view. In *ICCV*, 1999.

[31] S. X. Yu. Segmenation using multiscale cues. In *CVPR*, 2004.

[32] S. X. Yu and J. Shi. Multiclass spectral clustering. In *ICCV*, 2003.

[33] R. Zabih and V. Kolmogorov. Spatially coherent clustering using graph cuts. In *CVPR*, 2004.

[34] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *NIPS*, 2003.