RGB-D IMAGE SEGMENTATION BASED ON MULTIPLE RANDOM WALKERS

Se-Ho Lee[1]*, Won-Dong Jang*[1]*, Byung Kwan Park*[2]*, and Chang-Su Kim*[1]

[1] School of Electrical Engineering, Korea University, Seoul, Korea
[2] Video Tech. Lab., SK Telecom, Korea
E-mails: {seholee, wdjang, cskim}@mcl.korea.ac.kr[1], byungkwan.park@sk.com[2]

ABSTRACT

A novel RGB-D image segmentation algorithm is proposed in this work. This is the first attempt to achieve image segmentation based on the theory of multiple random walkers (MRW). We construct a multi-layer graph, whose nodes are superpixels divided with various parameters. Also, we set an edge weight to be proportional to the similarity of color and depth features between two adjacent nodes. Then, we segment an input RGB-D image by employing MRW simulation. Specifically, we decide the initial probability distribution of agents so that they are far from each other. We then execute the MRW process with the repulsive restarting rule, which makes the agents repel one another and occupy their own exclusive regions. Experimental results show that the proposed MRW image segmentation algorithm provides competitive segmentation performances, as compared with the conventional state-of-the-art algorithms.

***Index Terms*—** Multiple random walkers, segmentation, random walk, and RGB-D image segmentation.

1. INTRODUCTION

Image segmentation is a fundamental problem in image processing and computer vision. It has been researched actively [1–8], in order to divide an image into meaningful regions automatically. However, it still remains a difficult problem when objects and background regions have similar colors or textures. To alleviate this problem, we can utilize depth information, since foreground and background regions often yield different depths. Recently, RGB-D sensors can be deployed at moderate costs, *e.g.* Microsoft Kinect. Many computer vision techniques, such as activity recognition [9] and saliency detection [10], exploit depth information. Likewise, we can adopt depth cues to perform image segmentation more accurately.

Various image segmentation algorithms have been proposed. Shi and Malik [1] introduced the spectral graph clustering, which represents an image as a graph and exploits the eigenvectors of the normalized Laplacian of the graph to segment the image. Comaniciu and Meer [2] proposed the mean-shift algorithm, which delineates clusters by finding local modes of a density function. Felzenszwalb and Huttenlocher [3] proposed a graph-based algorithm, which merges two regions by comparing the inter-region difference with the internal difference of each region. Kim *et al.* [4] constructed a multi-layer graph by over-segmenting an image into superpixels and then employed the spectral clustering. Li *et al.* [5] also used the multi-layer structure, but they designed a sparse bipartite graph to segment an image efficiently. Arbeláez *et al.* [6] proposed a contour-based algorithm. To improve the segmentation performance, their algorithm applies the spectral clustering using multiple local cues and employs a learned parameter set.

However, these algorithms [1–6] use only color information to segment images. Recently, several RGB-D image segmentation algorithms have been proposed. For instance, Gupta *et al.* [7] extended the contour-based algorithm in [6] to exploit depth information. They used geometric cues from depth data to detect contours. Silberman *et al.* [8] over-segmented an image and merged those superpixels based on the similarity levels, which were obtained by learned classifiers using RGB, depth, and scene structure data.

We propose a novel RGB-D image segmentation algorithm using the system of multiple random walkers (MRW). Lee *et al.* [11] first introduced the notion of MRW, which simulates movements of multiple random walkers (or agents) on a graph simultaneously. They applied MRW to the co-segmentation problem, which employs two agents only. On the other hand, this is the first attempt to use multiple agents (more than two) in the MRW system to segment a single image. First, we construct a graph using the superpixel techniques in [2, 12]. Each superpixel becomes a node, and adjacent nodes are connected by edges. Also, each edge weight is set to be proportional to the similarity of the color and depth features between the corresponding two superpixels. Then, we perform the MRW simulation to label each node. More specifically, we determine the initial distributions of multiple agents by locating the agents sequentially, so that they are far from one another. We then carry out the MRW simulation with time-varying restarting distributions, which make the agents repel one another and eventually settle in their own regions. Experimental results show that the proposed MRW algorithm provides competitive segmentation performances, as compared with the conventional algorithms [3–7].

The rest of this paper is organized as follows: Section 2 proposes the MRW image segmentation algorithm, Section 3 presents comparative experimental results, and Section 4 concludes this work.

2. PROPOSED ALGORITHM

2.1. Graph Construction

We over-segment an input color image using the superpixel methods [2, 12] to construct the graph $G = (V, E)$. The node set V consists of superpixels $s_i, i = 1, ..., N$, and edge e_{ij} in the edge set E connects superpixels s_i and s_j. We connect superpixels based on a multi-layer structure to achieve reliable clustering. By employing differently over-segmented superpixels in the multiple layers and combining the information systematically, ambiguous regions with weak boundaries can be partitioned accurately in a probabilistic manner [4, 5]. Fig. 1 shows the multi-layer structure, which has a single primary layer and three secondary layers. Superpixels in the primary and secondary layers compose the node set. The primary layer is partitioned into 300 superpixels by the SLIC algorithm [12]. The secondary layers are partitioned, respectively, by

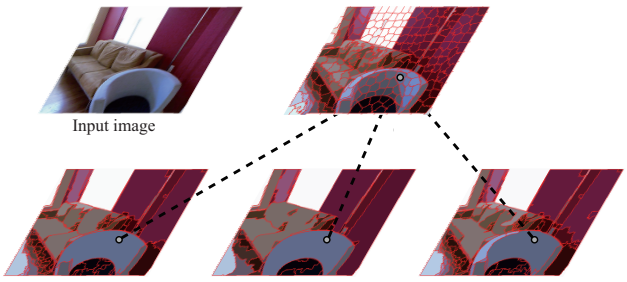


Fig. 1. The multi-layer graph structure. The upper and lower layers represent the primary and secondary layers, respectively. Each superpixel becomes a graph node. Within each layer, adjacent superpixels are connected by edges. A primary node and a secondary node are connected, if they share the same pixel.

the mean-shift algorithm [2] with three sets of the parameters of spatial bandwidth, range bandwidth, and minimum superpixel size: $(7, 7, 200)$, $(7, 9, 200)$, $(9, 7, 200)$.

Each superpixel in the primary and secondary layers is represented by the average LAB color and depth values of the member pixels. Adjacent nodes within each layer are connected by edges. Also, when a primary node and a secondary node share the same pixel, they are connected. However, nodes in different secondary layers are not connected.

We assign weight w_{ij} to edge e_{ij}, representing the affinity between s_i and s_j. Edge weights are obtained from the color distances and the depth distances between nodes. Specifically, we first calculate the color distance $\rho_c(s_i, s_j)$ and the depth distance $\rho_d(s_i, s_j)$ between s_i and s_j as

$$\rho_c(s_i, s_j) = \|\mathbf{c}_i - \mathbf{c}_j\|^2, \quad (1)$$

$$\rho_d(s_i, s_j) = (d_i - d_j)^2, \quad (2)$$

where $\mathbf{c}_i$ and d_i denote the average LAB color and depth of s_i, respectively. Then, we compute the edge weight w_{ij} by

$$w_{ij} = \begin{cases} \exp\left(-\frac{\rho_c(s_i,s_j)}{2\sigma_c^2} - \frac{\rho_d(s_i,s_j)}{2\sigma_d^2}\right) & \text{if } e_{ij} \in E, \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

where the scale parameters are set to $\sigma_c^2 = \sigma_d^2 = 1/60$.

A random walker travels on the graph G according to the transition probability a_{ij} that the walker moves from node j to node i. We obtain the transition probability a_{ij} by dividing w_{ij} by the degree of node j, $a_{ij} = w_{ij} / \sum_k (w_{kj})$. We then construct the transition matrix $\mathbf{A} = [a_{ij}]$.

2.2. MRW Simulation

For the image segmentation, we perform the MRW simulation [11], in which multiple agents move on the graph and interact with one another. Let $\mathbf{p}_k^{(t)} = [p_{k,1}^{(t)}, \ldots, p_{k,N}^{(t)}]^T$ be the probability distribution of agent k on the graph at time t. Then, random movements of agent k are determined by the recursion,

$$\mathbf{p}_k^{(t)} = (1 - \epsilon)\,\mathbf{A}\mathbf{p}_k^{(t-1)} + \epsilon \mathbf{m}_k^{(t)}, \quad k = 1, ..., K, \quad (4)$$

where K is the number of agents on the graph. On the other hand, the random walk with restart (RWR) simulation [13] is given by

$$\mathbf{p}_k^{(t)} = (1 - \epsilon)\,\mathbf{A}\mathbf{p}_k^{(t-1)} + \epsilon \mathbf{r}_k, \quad k = 1, ..., K. \quad (5)$$

In both MRW and RWR processes, agent k traverses the graph based on the transition matrix $\mathbf{A}$ with probability $1 - \epsilon$, and returns to specific nodes according to the restarting distribution with probability ϵ. However, the MRW process adopts the time-varying restarting distribution $\mathbf{m}_k^{(t)} = [m_{k,1}^{(t)}, \ldots, m_{k,N}^{(t)}]^T$ for $1 \le k \le K$, while the RWR process the time-invariant restarting distribution $\mathbf{r}_k$.

In the MRW process, we can make the agents interact with one another, by determining the time-varying restarting distribution $\mathbf{m}_k^{(t)}$ of agent k at time t according to the probability distributions of all agents at time $t - 1$.

2.3. Time-Varying Restarting Distributions

In this work, we determine the time-varying restarting distribution of each agent, so that the agents repel one another. Specifically, we set the ith component $m_{k,i}^{(t)}$ of the restarting distribution $\mathbf{m}_k^{(t)}$ of agent k at time t to

$$m_{k,i}^{(t)} = \beta \cdot \alpha_{k,i}^{(t)} \cdot p_{k,i}^{(t-1)}, \quad (6)$$

where

$$\alpha_{k,i}^{(t)} = \frac{\sum_j a_{ij} \cdot p_{k,j}^{(t-1)}}{\max_l \left(\sum_j a_{ij} \cdot p_{l,j}^{(t-1)}\right)} \quad (7)$$

and β is a constant to normalize $\mathbf{m}_k^{(t)}$ to a probability distribution. Notice that $\sum_j a_{ij} \cdot p_{k,j}^{(t-1)}$ measures the probability distribution of agent k near the ith node at time $t - 1$. Hence, if there are other agents with high probabilities near the ith node at time $t - 1$, $\alpha_{k,i}^{(t)}$ becomes smaller and $m_{k,i}^{(t)}$ also becomes smaller at time t. On the other hand, if agent k has a high probability $p_{k,i}^{(t-1)}$ at node i, $m_{k,i}^{(t)}$ becomes larger. Thus, an agent tends to restart, where it has a high probability but the others have lower probabilities. This enforces the agents to repel one another.

2.4. Initial Probability Distributions

As the iteration goes on, the MRW process in (4) with the restarting rule in (6) converges to stationary distributions. However, the stationary distributions depend on initial distributions $\mathbf{p}_k^{(0)}$, $1 \le k \le K$. In this work, we attempt to locate multiple agents initially to distantly placed modal nodes. Note that a modal node is defined as a node around which the graph has the locally densest distribution of nodes. To find modal nodes, we perform the RWR simulation of a single agent in (5), by employing the uniform restarting distribution, and obtain the stationary distribution $\mathbf{q} = [q_1, ..., q_N]^T$. In general, q_i is high when node i is near a modal node.

Then, we determine the initial distributions $\mathbf{p}_k^{(0)}$ sequentially from $k = 1$ to K. We execute another RWR process to decide $\mathbf{p}_k^{(0)}$, by making agent k restart at the single node, which has a high probability in $\mathbf{q}$ but low probabilities in the previously computed $\mathbf{p}_n^{(0)}$, $1 \le n \le k - 1$. In this way, we can locate agent k far from the previously located agent. More specifically, to determine the single restarting node, we define the vector $\mathbf{h}_k = [h_{k,1}, ..., h_{k,N}]^T$ by

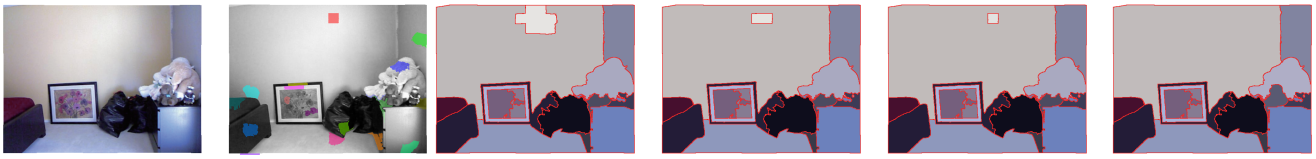
$$h_{k,i} = \frac{q_i}{\sum_{n=1}^{k-1} p_{n,i}^{(0)}}. \quad (8)$$

Then, agent k restarts at the single node i_k^* that maximizes $h_{k,i}$,

$$i_k^* = \arg\max_i h_{k,i}. \quad (9)$$

Table 1. Comparison of image segmentation performances on 200 RGB-D images of NYUDv2 [8] and 50 RGB-D images of KINECTv2D.

Dataset	Method	PRI		VoI		BDE	
		ODS	OIS	ODS	OIS	ODS	OIS
NYUDv2 [8]	FH [3]	0.8490	0.8675	2.2387	2.0935	9.9225	8.4116
	MLSS [4]	0.8405	0.8576	2.0513	1.8968	11.0160	9.9269
	SAS [5]	0.8495	0.8612	1.9733	1.8259	9.6465	8.3747
	UCM [6]	0.8420	0.8533	2.2158	2.0169	10.2206	8.9251
	UCM-RGBD [7]	0.8425	0.8685	**1.8899**	**1.7614**	15.8821	13.7184
	MRW-RGB	0.8448	0.8615	2.0252	1.8344	**9.5180**	**7.9506**
	MRW-RGBD	**0.8515**	**0.8715**	1.9606	1.7720	9.6745	8.2538
KINECTv2D	FH [3]	0.9363	0.9461	1.0974	0.9827	10.0310	8.9796
	MLSS [4]	**0.9558**	0.9648	**0.7387**	0.6407	7.2702	5.0369
	SAS [5]	0.9531	0.9641	0.7482	0.6332	8.2428	5.2309
	UCM [6]	0.9554	0.9593	0.7901	0.7489	**6.3822**	5.1235
	MRW-RGB	0.9488	0.9542	0.7645	0.6921	6.5969	5.1320
	MRW-RGBD	0.9517	**0.9659**	0.7467	**0.6249**	7.2213	**4.7636**



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

Fig. 2. An example of the MRW simulation: (a) an input color image, (b) the initial restarting node of each agent, and (c)∼(f) the segmentation results at $t = 1, 5, 10$, and 100. In this simulation, we set $K = 18$.

With the selected restarting node, we perform the RWR process to obtain the stationary distribution, which is used as the initial probability distribution $\mathbf{p}_k^{(0)}$ in (4).

Fig. 2 shows an example. Colored superpixels in Fig. 2(b) represent the restarting nodes of the 18 agents for the input image in Fig. 2(a). Note that the restarting nodes belong to different objects.

2.5. Label Assignment

From the MRW simulation in (4), we obtain the stationary distributions $\mathbf{p}_k = \lim_{t\to\infty} \mathbf{p}_k^{(t)}$ for $1 \le k \le K$. We assign label k to node i, when agent k has the highest probability at node i. In other words, we determine the label l_i of node i as

$$l_i = \arg\max_k p_{k,i}. \tag{10}$$

In our system, K agents move on the graph to repel one another and occupy their own regions in an input image. Consequently, the input image is partitioned into K regions, where K is manually selected by users.

Figs. 2(c)∼(e) show intermediate segmentation results for the image in Fig. 2(a). We see that agents settle in their own regions, as iteration goes on. After the convergence, the image is partitioned into meaningful segments, as shown in Fig. 2(f).

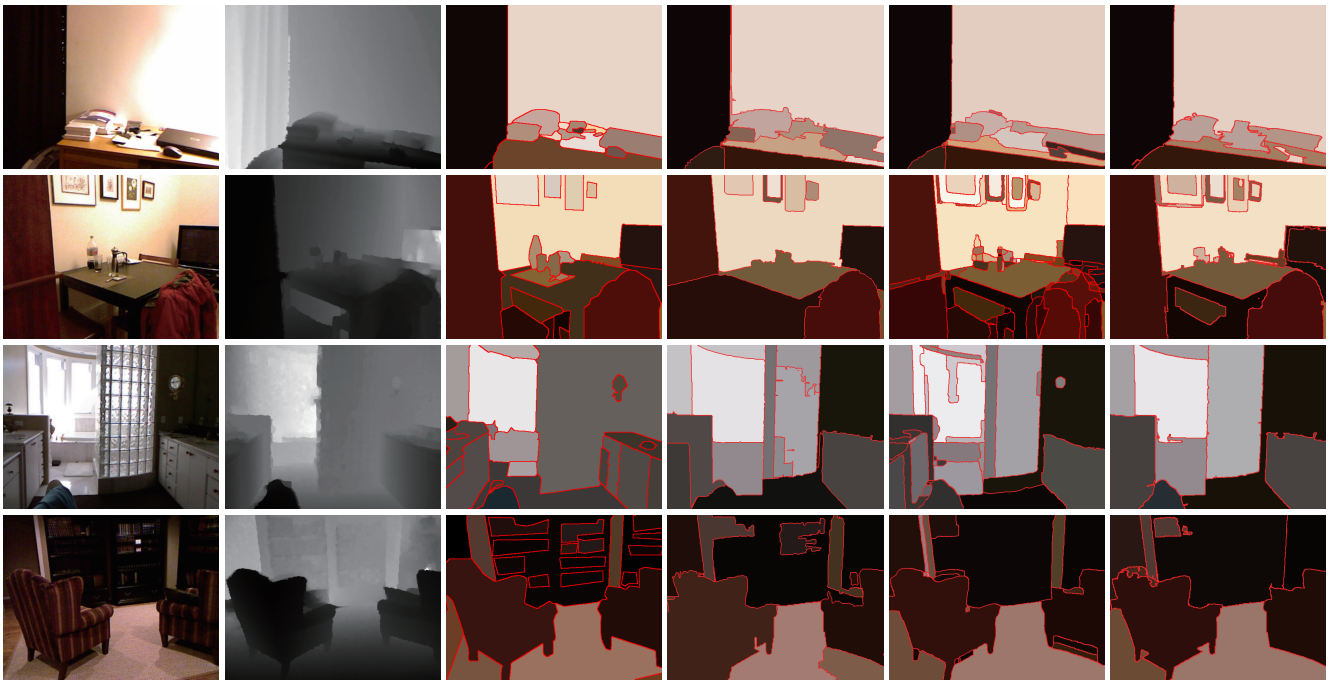
3. EXPERIMENTAL RESULTS

To compare RGB-D image segmentation performances, we use 200 RGB-D images in the NYU depth dataset v2 (NYUDv2) [8] and 50 RGB-D images in our dataset captured from a Microsoft Kinect v2 (KINECTv2D). NYUDv2 contains complex indoor scenes captured from a Microsoft Kinect v1, while KINECTv2D includes relatively simple indoor scenes. We compare the proposed MRW-RGBD segmentation algorithm with five conventional methods: efficient graph-based segmentation (FH) [3], multi-layer spectral segmentation (MLSS) [4], segmentation by aggregating superpixels (SAS) [5], ultrametric contour maps (UCM) [6], and ultrametric contour maps for RGB-D images (UCM-RGBD) [7]. For the UCM-RGBD algorithm, we report the results only on the NYUDv2 dataset, which are available in [7]. Also, we modify the proposed MRW algorithm to use color information only (MRW-RGB). The MRW-RGB results are obtained by replacing the edge weight in (3) with

$$w_{ij} = \begin{cases} \exp\left(-\frac{\rho_c(s_i,s_j)}{2\sigma_c^2}\right) & \text{if } e_{ij} \in E, \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

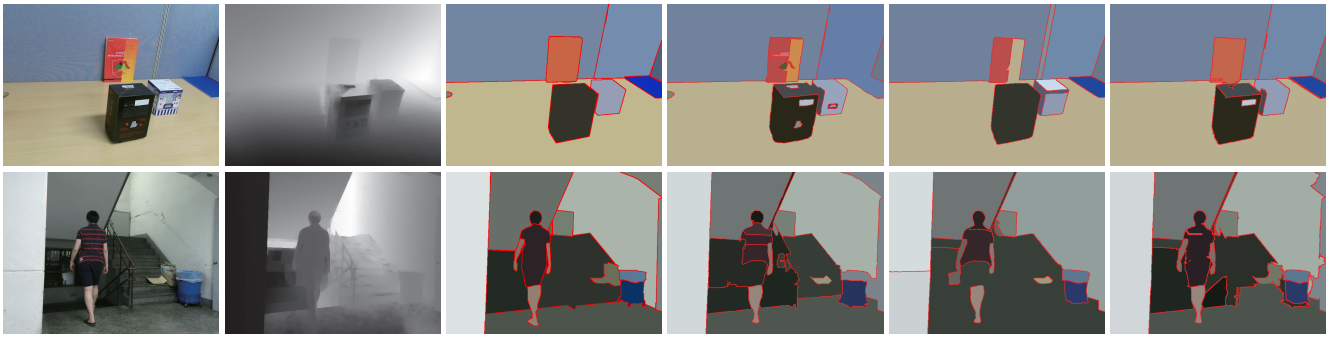
To compare the segmentation performances quantitatively, we adopt three metrics: probabilistic rand index (PRI) [14], variation of information (VoI) [15], and boundary displacement error (BDE) [16]. PRI counts the pairs of pixels that have consistent labels between a human annotated result and an automatic result. VoI measures the amount of irrelevant information between the two results. BDE measures the average displacement of boundaries between the results. Thus, a better segmentation scheme should yield a higher PRI value and lower VoI and BDE values. We evaluate the performances according to the optimal dataset scale (ODS) and the optimal image scale (OIS), respectively, by varying the parameters. The number of segments for each image varies from 4 to 20 when we measure the segmentation performances of MRW-RGB and MRW-RGBD.

Table 1 compares the segmentation performances. Note that the proposed algorithm has competitive image segmentation performances to the state-of-the-art methods on both NYUDv2 and KINECTv2D datasets. Especially, the proposed algorithm provides comparable image segmentation performances to UCM-RGBD on NYUDv2, even though UCM-RGBD is a learning-based method. Also, by comparing the results of MRW-RGB and MRW-RGBD, we see that depth information improves the performance of the pro-



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

Fig. 3. Segmentation results on the NYUDv2 [8]: (a) input images, (b) depth images, (c) ground-truth, (d) SAS [5], (e) UCM-RGBD [7], and (f) MRW-RGBD.



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

Fig. 4. Segmentation results on the KINECTv2D: (a) input images, (b) depth images, (c) ground-truth, (d) SAS [5], (e) UCM [6], and (f) MRW-RGBD.

posed algorithm, except for the case of the BDE test. In the BDE test, the boundary accuracy is important, but the depth noise of the Kinect sensor causes the degradation of boundaries in some cases.

Fig. 3 shows examples of the segmentation results of SAS, UCM-RGBD, and MRW-RGBD on NYUDv2. We set the parameters for each algorithm to obtain the best performance. SAS fails to segment objects, which have similar colors to surrounding regions, since it only exploits color information. For example, in the fourth row in Fig. 3(d), SAS cannot delineate the chair correctly. UCM-RGBD is a contour-based method, which merges superpixels based on the strength of detected contours. Thus, they fail to merge regions, belonging to the same object, when the object has complex color information, as shown in the second row in Fig. 3(e). In contrast, in Fig. 3(f), the proposed algorithm segments input images more accurately. Fig. 4 shows segmentation results of SAS, UCM, and MRW-RGBD on KINECTv2D. We see that the proposed MRW-RGBD algorithm separates foreground objects from background regions successfully.

4. CONCLUSIONS

We proposed the RGB-D image segmentation algorithm, which segments an image using multiple agents in the MRW system. We first constructed a multi-layer graph using the different superpixel methods [2, 12]. We set the initial probability distribution of agents to be far apart from one another. Then, we executed the MRW simulation with the time-varying restarting distributions to make the agents repel one another. Finally, we assigned a label to each node using the stationary distributions of the agents. Experimental results showed that the proposed algorithm provides competitive performances, in comparison with the state-of-the-art algorithms.

5. REFERENCES

[1] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[2] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.

[3] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, Nov. 2004.

[4] T. H. Kim, K. M. Lee, and S. U. Lee, "Learning full pairwise affinities for spectral segmentation," in *Proc. IEEE CVPR*, Jun. 2010, pp. 2101–2108.

[5] Z. Li, X.-M. Wu, and S.-F. Chang, "Segmentation using superpixels: A bipartite graph partitioning approach," in *Proc. IEEE CVPR*, Jun. 2012, pp. 789–796.

[6] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

[7] S. Gupta, P. Arbeláez, and J. Malik, "Perceptual organization and recognition of indoor scenes from RGB-D images," in *Proc. IEEE CVPR*, Jun. 2013, pp. 564–571.

[8] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. ECCV*, Oct. 2012, pp. 746–760.

[9] O. Oreifej and Z. Liu, "HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences," in *Proc. IEEE CVPR*, Jun. 2013, pp. 716–723.

[10] C. Lang, T. V. Nguyen, H. Katti, K. Yadati, M. Kankanhalli, and S. Yan, "Depth matters: Influence of depth cues on visual saliency," in *Proc. ECCV*, Oct. 2012, pp. 101–115.

[11] C. Lee, W.-D. Jang, J.-Y. Sim, and C.-S. Kim, "Multiple random walkers and their application to image cosegmentation," in *Proc. IEEE CVPR*, Jun. 2015, pp. 3837–3845.

[12] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[13] Jia-Yu Pan, Hyung-Jeong Yang, Christos Faloutsos, and Pinar Duygulu, "Automatic multimedia cross-modal correlation discovery," in *Proc. ACM SIGKDD*, 2004, pp. 653–658.

[14] R. Unnikrishnan, C. Pantofaru, and M. Hebert, "Toward objective evaluation of image segmentation algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 929–944, Jun. 2007.

[15] M. Meilă, "Comparing clustering: An axiomatic view," in *Proc. 22nd International Conference on Machine Learning*, Jun. 2005, pp. 577–584.

[16] J. Freixenet, X. Muñoz, D. Raba, J. Martí, and X. Cufí, "Yet another survey on image segmentation: Region and boundary information integration," in *Proc. ECCV*, Apr. 2002, pp. 408–422.