

Fast Interactive Image Segmentation Using Bipartite Graph Based Random Walk with Restart

Yunfan Du^(✉), Fei Li, and Rujie Liu

Fujitsu Research and Development Center Co., Ltd., Beijing, China
{duyunfan,lifei,rjliu}@cn.fujitsu.com

Abstract. Although random walk with restart(RWR) has been successfully used in interactive image segmentation, the traditional implementation of RWR does not scale for large images. As the images are usually stored on local disk prior to user interaction, we can preprocess the images to save user time. In this paper, we do an offline precomputation that over-segments the input image into superpixels with different scales and then aggregates superpixels and pixels into one bipartite graph which fuses the high level and low level information. Given user scribbles, we do a realtime RWR on the bipartite graph by applying an approximate method which maps the RWR from pixel level to superpixel level. As the number of superpixels is far more less than the number of pixels in the image, our method reduces the amount of user time significantly. The experimental results demonstrate that our method achieves a similar result compared to original RWR along with outperforming in speed.

Keywords: Interactive image segmentation · Random walk with restart · Superpixel · Bipartite graph

1 Introduction

Image segmentation is one of the fundamental but challenging problems in image processing and computer vision. The approaches of unsupervised image segmentation automatically partition an image into coherent regions without any prior knowledge, such as the stochastic clustering [22], mean shift [5], mixture model [19, 20], and level sets [11, 12]. Unsupervised image segmentation is widely referred as a crucial function of high-level image understanding, which is designed to simulate functionalities of human visual perception such as object recognition [6] and scene parsing [7]. However, the state-of-the-art automatic segmentation methods are still far from the human segmentation performance, which have several problems such as finding the faint object boundaries and separating the highly complicated background in natural images. In order to solve these problems, an interactive method is often preferred when the objects of interest need to be accurately selected and extracted from the background. In this paper, we address this interactive segmentation problem with our fast RWR method.

The task of interactive segmentation is generally to produce a binary segmentation mask of the input image by separating the objects of interest from their background. There is a plenty of literature on the work of interactive image segmentation techniques that have been explored by the investigators during the last decade. The popular graph-based approaches include interactive graph cut [4, 18], geodesic distance [3], level sets [16], random walk [9], and RWR [10]. An input image is usually represented by an undirected graph structure where the vertices denote image pixels and edges connect pairs of vertices. Then the problem of interactive segmentation becomes equivalent to partitioning the vertices into disjoint segments. One very successful graph partition technique is RWR, which usually starts random walk from a seed vertex, iteratively moves to its neighborhood vertex, with a probability that is proportional to the weight of the edge between them. There are many RWR related methods including cross modal correlation discovery [17], generative image segmentation [10] and so on. In [10], RWR was incorporated with naive Bayesian theory, which generated an interactive image segmentation method. Pan et al. [17] used RWR to do automatic image captioning.

An important research challenge of RWR is its speed. The solutions of above methods usually need to compute the inverse of matrix. As we all know that the time complexity of matrix inversion approximates $O(N^3)$, which means that the computation cost may become unaffordable in a realtime application, especially for large matrix.

In this paper, we propose a fast interactive segmentation method. Our idea originally inspired by [13]. Li et al. [13] is an unsupervised image segmentation method which can be seen as a fast version of normalized cut. Li et al. [13]’s acceleration work was based on a bipartite graph structure. Li et al. [13] first segmented the input image into superpixels, then aggregated multi-layer superpixels and image pixels into one bipartite graph, and so that highly efficient spectral clustering could be applied on the unbalanced bipartite graph. We introduce the bipartite graph to accelerate interactive image segmentation by modifying the structure of that bipartite graph, we are able to do fast RWR on this graph.

We also notice that some work has been done on bipartite graph based RWR [15, 23]. Sun et al. [23] constructed a bipartite graph based on RWR to address two issues: neighborhood formation and anomaly detection while [15] applied bipartite graph based RWR on spatial outlier detection. Sun et al. [23] constructed bipartite graph based on the inner correlation of object set such as conferences vs. authors in a scientific publication network. This construction step limited the application of [23] because only some particular datasets has the inner correlation like conferences vs. authors. Lately, the construction of bipartite graph was generalized to ordinary dataset by [15]. Liu et al. [15] first generated a set of clusters from the ordinary spatial object set, then put the set of clusters and original spatial objects into two disjoint sets of bipartite graph respectively. After the construction of bipartite graph, both [15, 23] merged bipartite graph $B_{K \times N}$ and its transpose matrix $B_{K \times N}^T$ into one large adjacency matrix $M_{(K+N) \times (K+N)}$ and then RWR was done on this large adjacency matrix.

Here K is the number of clusters, N is the number of spatial objects. This adjacency matrix is even larger than the adjacency matrix of ordinary RWR, which is usually $M_{N \times N}$. That means that to compute the inverse of large matrix will be more time consuming. Different from their works, we applying an approximate method to map the RWR from pixel level to superpixel level, which significantly reduces the computation cost.

The main contributions of the paper are as follows: 1. We transfer the RWR from pixel level to superpixel level by applying an approximate method. 2. We achieve competitive results compared with original RWR and outperforming in speed.

The rest of the paper is organized as follows: the proposed method is presented in Sect. 2; The experimental results are presented in Sect. 3. Finally, we conclude the paper in Sect. 4.

2 Method

We explain the principle of our fast RWR method and how the bipartite graph accelerates the original RWR in this section. Our method can be divided into two steps: offline and online. In the offline step, we first over-segment the input image into small pixel patches which also known as superpixels, by using some over-segmentation methods such as MeanShift [5], FH [8], Entropy Rate Superpixel [14], Lazy Random Walk [21] and many more. Then we compute the affinity map between each pixel patch and its corresponding superpixels and construct a bipartite graph. In the online step, we do RWR on the constructed bipartite graph.

2.1 Graph Structure

The structure of our bipartite graph is showed in Fig. 1. The bipartite graph can be divided into two sets, one consists of pixels, while the other consists of superpixels with different scales. A pixel will be connected to the superpixels it belongs to. Note that one pixel may be connected to multiple superpixels, because these superpixels are obtained by doing multiple over-segmentation with different scale parameters. The purple arrow in Fig. 1 is one possible random walk path. We can see that the random walk on bipartite graph makes round trip between two sets, so that it is faster than the random walk on ordinary graph which is constructed from pixels.

Given image I with N pixels, to build a bipartite graph, first we need to select some over-segmentation methods to cluster the image pixels into small clusters. The cluster can also be called superpixel in image processing field. After t times of over-segmentation, finally we have $M = \sum_{c=1}^t K_c$ superpixels. Here K_c is the number of superpixels we get in the c -th over-segmentation. We vary the value of K_c to get different scales of superpixels. These superpixels can enforce the local coherence and global relationship in the bipartite graph.

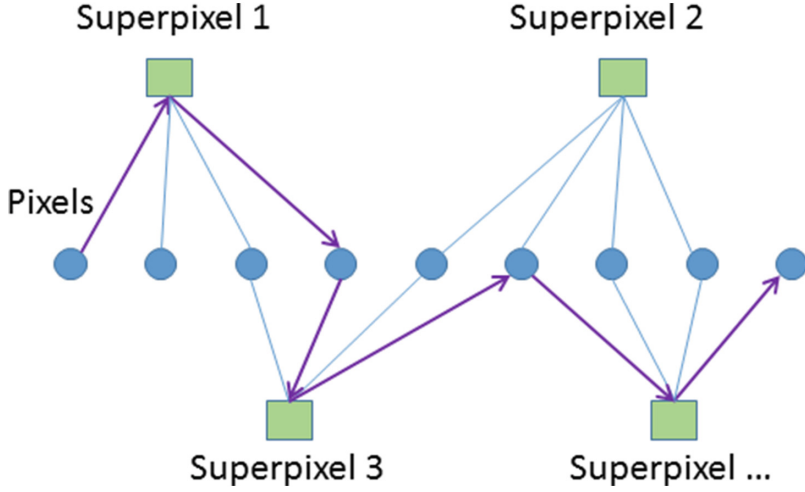


Fig. 1. Structure of our bipartite graph.

After obtaining superpixels, we define a weighted bipartite graph $G = \{V_1 \cup V_2, E\}$. Where $|V_1| = N$ and $|V_2| = M$ are node sets that represent pixels and superpixels respectively and $E \subseteq V_1 \times V_2$. Given two vertices $v_i \in V_1$ and $v_j \in V_2$, the edge that connects the two vertices is defined as $e_{ij} \in E$. The weight of e_{ij} is defined as w_{ij} . Such an edge weight w_{ij} measures the similarity between i -th pixel and the j -th superpixel which the i -th pixel belongs to. Ideally, the superpixels must have some overlapped regions to make sure the RWR can reach every pixel and every superpixel. This is also a limitation for selecting over-segmentation method.

2.2 Edge Weight Measurement

We use the color and spatial information as the feature of image pixels, for example, $p_i = (l_i, a_i, b_i, x_i, y_i)$. Here l_i, a_i, b_i are the color values of p_i , note that the color values are in Lab color space. (x_i, y_i) is the coordinate of p_i in the image. The superpixel can be represented by $q_j = \frac{1}{|C_j|} \sum_{p_i \in C_j} p_i$ and q_j can also be represented by $(l_j, a_j, b_j, x_j, y_j)$. The similarity between the pixels and the superpixels is defined as follow [1]:

$$\begin{aligned}
 d_c &= \sqrt{(l_i - l_j)^2 + (a_i - a_j)^2 + (b_i - b_j)^2}, \\
 d_s &= \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \\
 w_{ij} &= \frac{1}{e^{\sqrt{d_c^2 + \alpha * (\frac{d_s}{T_j})^2}}},
 \end{aligned} \tag{1}$$

where d_c and d_s is the difference between i -th pixel and j -th superpixel in color and spatial space. α is a weight parameter that control the relative importance

between color similarity and spatial proximity. T is a normalization factor that represents the scale of j -th superpixel. If j -th superpixel is obtained from c -th over-segmentation, then $T_j = \sqrt{N/K_c}$.

2.3 Naive RWR on Bipartite Graph

Based on G , we construct an $N \times M$ asymmetric matrix B , with $B_{ij} = w_{ij}$. Then the adjacency matrix of graph G can be represented by:

$$W = \begin{pmatrix} 0 & B^T \\ B & 0 \end{pmatrix}. \quad (2)$$

where W is an $(N + M) \times (N + M)$ matrix, B^T is the transpose of B . By row normalizing matrix W , we get the transition probability matrix $P = D^{-1}W$, where D is a diagonal matrix with $D_{ii} = \sum_j W_{ij}$. The matrix P can also be represented by:

$$P = \begin{pmatrix} 0 & \mathcal{B}^T \\ \mathcal{B} & 0 \end{pmatrix}. \quad (3)$$

where \mathcal{B} and \mathcal{B}^T are row normalized matrix B and B^T . The RWR on G will start from initial seed vertices, then converge to a steady state probability distribution u . RWR is usually defined as equation [10]:

$$u = (1 - \gamma) * P * u + \gamma * v, \quad (4)$$

where v is an initial vector that represents seed vertices, γ is the probability that random walk restarts from seed vertices. We can initialize v from user scribbles. u is the steady state probability vector to be solved. The target of RWR is find the converge vector u . Working directly on P will result in a large consumption of time and storage. Next we propose an approximate method to transfer the RWR from P into a far more smaller matrix, in other words, transfer RWR from pixel level to superpixel level.

2.4 Accelerating RWR

First, we process pixels and superpixels separately in Eq. 4:

$$\begin{pmatrix} u_{sp} \\ u_p \end{pmatrix} = (1 - \gamma) \begin{pmatrix} \mathcal{B}^T \cdot u_p \\ \mathcal{B} \cdot u_{sp} \end{pmatrix} + \gamma \begin{pmatrix} v_{sp} \\ v_p \end{pmatrix} \quad (5)$$

where u_p is an $N \times 1$ vector that means the probability that each pixel belongs to certain class, for example c . Similarly, u_{sp} is an $M \times 1$ vector that represents the probability that each superpixel belongs to certain class c . v_p and v_{sp} are the initial probabilities of each pixel and superpixel belongs to c . We can further partition the above equation into two new equations:

$$u_{sp} = (1 - \gamma) \mathcal{B}^T u_p + \gamma v_{sp} \quad (6)$$

$$\mathbf{u}_p = (1 - \gamma)\mathcal{B}\mathbf{u}_{sp} + \gamma\mathbf{v}_p \quad (7)$$

To accelerate RWR, with the unbalanced bipartite graph B , we approximate \mathbf{u}_p and \mathbf{u}_{sp} by:

$$\begin{cases} \mathbf{u}_p = \mathcal{B}\mathbf{u}_{sp}, \\ \mathbf{u}_{sp} = \mathcal{B}^T\mathbf{u}_p. \end{cases} \quad (8)$$

Note that matrix \mathcal{B} and \mathcal{B}^T represent the similarity between each pixel and their corresponding superpixels. Briefly speaking, the two equations can be seen as a assumption that the probability of each superpixel belongs to certain class is the sum of the probability of its member pixels belong to the same class. We can easily generalize the assumption to the initial vector:

$$\begin{cases} \mathbf{v}_p = \mathcal{B}\mathbf{v}_{sp}, \\ \mathbf{v}_{sp} = \mathcal{B}^T\mathbf{v}_p. \end{cases} \quad (9)$$

With this simple linear approximation, we can rewrite Eq. 6 by left multiplying Eq. 7 with \mathcal{B}^T :

$$\mathcal{B}^T\mathbf{u}_p = (1 - \gamma)\mathcal{B}^T\mathcal{B}\mathbf{u}_{sp} + \gamma\mathcal{B}^T\mathbf{v}_p. \quad (10)$$

Together with Eqs. 8 and 9, the above equation is equal to:

$$\mathbf{u}_{sp} = (1 - \gamma)Q\mathbf{u}_{sp} + \gamma\mathbf{v}_{sp}, \quad (11)$$

where $[Q]_{M \times M} = \mathcal{B}^T\mathcal{B}$, then Q_{ij} means the similarity between two superpixels i and j . In other words, the above equation can be seen as doing RWR on the superpixel level. As $K \ll N$, the RWR on matrix Q is far more efficient than on matrix P . The converged vector \mathbf{u}_{sp} can be mapped back to the pixel level according to Eq. 8, and the final result is $\mathbf{u}_p = \mathcal{B}\mathbf{u}_{sp}$.

In conclusion, based on the special structure of bipartite graph and our assumption, we first transfer the RWR from pixel level to superpixel level which can be seen as a downsampling operation. After solving Eq. 11 in superpixel level, we then map the obtained superpixel level class probability information back to the pixel level which can be seen as an upsampling operation. By working on superpixel level, we achieve significant improvement in speed compared to the traditional pixel level RWR. What makes our downsampling and upsampling different is that our operations are based on superpixels with different scales while conventional downsampling is usually on a fixed scale. Benefit from the over-segmentation methods in the offline step, our downsampling and upsampling method has a good property that preserving the region and boundary information of superpixels. Next we will show the performance of our method by some experiments.

3 Results

In this section, experiments are conducted to evaluate the effectiveness of the proposed framework. We have three parts in this section. We first test our approach for interactive image segmentation on the Berkeley Segmentation Datasets

(BSD) of testing images¹, which includes five hundred test images with human annotations as the ground-truth data [2]. The BSD benchmark includes the natural images with abundant colors and different complicated textures, which makes it challenging to segment them even with the user scribbles. Then we compare the error rate of our method with original RWR method [10]. Finally we do a statistic analysis for the computation cost of the two methods.

All the segmentation results are obtained from official released code and the parameters are set to default values. In this section, the scribbles used in our algorithm are presented in the following format: green scribbles are used to indicate the foreground object, blue scribbles are used to indicate the background parts of images. These pixels marked with scribbles will be used as seed pixels. Then the seed pixels are used to estimate the labels of the unlabeled pixels. We execute all our experiments on an 3.4 Ghz and 24 GB RAM workstation using MATLAB 7.10.

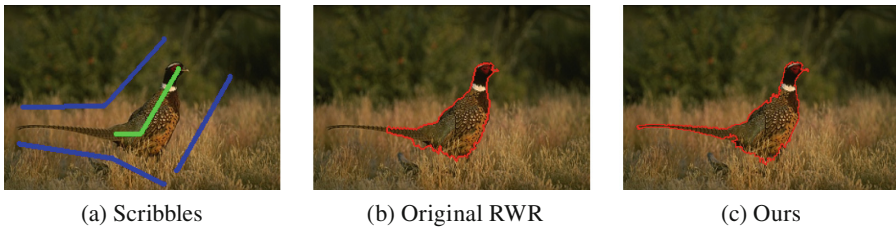


Fig. 2. Qualitative comparison using images with complex texture. From left to right, (a) scribbles; (b) segmentation result of original RWR; (c) result of our method (Color figure online).

3.1 Qualitative Comparison

Original RWR method [10] has a good performance in complex texture regions. In this part, we choose some images which have complex texture regions from the BSD datasets to show that our method achieves similar performance. Here we select MeanShift [5] and ERS [14] to do over-segmentation in the offline step. We use three different scales of parameters to generate hundreds of superpixels. Generally speaking, the more superpixels, the better segmentation results. But the time cost will increase along with the increment of superpixel number. Figure 2 demonstrates the effect of our method compared with original RWR. As showed in Fig. 2(c), even though there is no scribble at the tail of the bird, our method cuts the tail out precisely. By aggregating different scales of superpixels into one bipartite graph, we can obtain different scales of edge and region information at the same time, so that our method has better boundary coherence in narrow regions with same user scribbles, as showed in Fig. 2.

¹ <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html>.

3.2 Quantitative Comparison and Error Estimation

In this part, we will do a quantitative comparison. All images and their ground truth come from BSD datasets. We adopt normalized overlap [10] as accurateness metric:

$$Ac = \frac{S \cap G}{S \cup G}, \quad (12)$$

where S is the segmented foreground area, G is the ground truth foreground area. In BSD datasets every image usually has 5 grouth truth segmentation results. Here we choose the first one as metric. Ac represents the accuracy of segmentation, is in range[0, 1]. The bigger Ac means the higher segmentation accuracy.

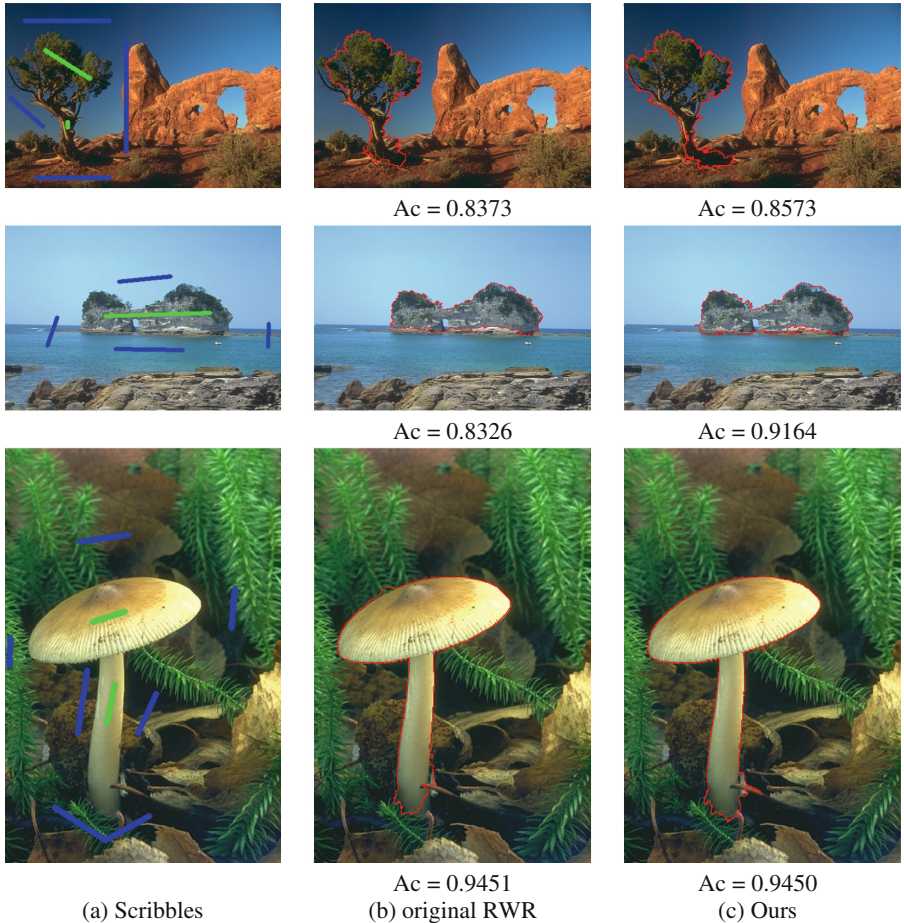


Fig. 3. Quantitative comparison using images with complex texture. From left to right, (a) original image with scribbles; (b) results of original RWR; (c) results of ours (Color figure online).

To do a sufficient comparison, we select images with complex texture. As showed in Fig. 3, our method achieve similar result compared to original RWR.

3.3 Time Analysis

As mentioned before, our method can be divided into two steps. Although the offline processing takes a little time, it can be processed automatically prior to user interaction. In the online step, the actual user interaction only needs very little time. As showed in Table 1, our method achieves $20x$ speed up compared to original RWR. The original RWR suffers from computing the inverse of large scale matrix and become infeasible to do a real-time large image segmentation. When the image resolution reaches 1536×1024 , original RWR needs about 31 s to make a segmentation, which is definitely unacceptable in an interactive setting.

Table 1. Time cost of RWR and our method in different resolutions.

Resolution	RWR	Ours-offline	Ours-online
289×193	0.85 s	1.2 s	0.04 s
481×321	2.5 s	3.1 s	0.11 s
1536×1024	31 s	45 s	1.2 s

First column is the resolution of input image; second column is the time cost of the offline step of our method; third column is the time cost of the online step of our method

4 Conclusions

In this paper, we have presented a fast interactive image segmentation method. By means of selected over-segmentation methods we first construct an unbalanced bipartite graph which consists of pixels and superpixels with different scales. Then we do random walk on this unbalance bipartite graph and transfer the RWR from pixel level to superpixel level with an approximation equation. In particular, our proposed method has produced competitive and qualitative segmentation results on the BSD datasets. And the experiments demonstrate that our method achieves significant improvement in speed compared to the original RWR.

References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Susstrunk, S.: Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(11), 2274–2282 (2012)

2. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. *Trans. Pattern Anal. Mach. Intell.* **33**(5), 898–916 (2011)
3. Bai, X., Sapiro, G.: A geodesic framework for fast interactive image and video segmentation and matting. In: *IEEE 11th International Conference on Computer Vision*, pp. 1–8. IEEE (2007)
4. Boykov, Y.Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In: *8th IEEE International Conference on Computer Vision*, vol. 1, pp. 105–112. IEEE (2001)
5. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(5), 603–619 (2002)
6. Divvala, S.K., Hoiem, D., Hays, J.H., Efros, A., Hebert, M., et al.: An empirical study of context in object detection. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1271–1278. IEEE (2009)
7. Eigen, D., Fergus, R.: Nonparametric image parsing using adaptive neighbor sets. In: *IEEE Conference on Computer vision and pattern recognition*, pp. 2799–2806. IEEE (2012)
8. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. *Int. J. Comput. Vis.* **59**(2), 167–181 (2004)
9. Grady, L.: Random walks for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(11), 1768–1783 (2006)
10. Kim, T.-H., Lee, S.U., Lee, K.M.: Generative image segmentation using random walks with restart. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part III*. LNCS, vol. 5304, pp. 264–275. Springer, Heidelberg (2008)
11. Li, C., Kao, C.Y., Gore, J.C., Ding, Z.: Minimization of region-scalable fitting energy for image segmentation. *IEEE Trans. Image Process.* **17**(10), 1940–1949 (2008)
12. Li, C., Xu, C., Gui, C., Fox, M.D.: Distance regularized level set evolution and its application to image segmentation. *IEEE Trans. Image Process.* **19**(12), 3243–3254 (2010)
13. Li, Z., Wu, X.M., Chang, S.F.: Segmentation using superpixels: a bipartite graph partitioning approach. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 789–796. IEEE (2012)
14. Liu, M.Y., Tuzel, O., Ramalingam, S., Chellappa, R.: Entropy rate superpixel segmentation. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2097–2104. IEEE (2011)
15. Liu, X., Lu, C.T., Chen, F.: Spatial outlier detection: random walk based approaches. In: *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 370–379. ACM (2010)
16. Liu, Y., Yu, Y.: Interactive image segmentation based on level sets of probabilities. *IEEE Trans. Visual. Comput. Graph.* **18**(2), 202–213 (2012)
17. Pan, J.Y., Yang, H.J., Faloutsos, C., Duygulu, P.: Automatic multimedia cross-modal correlation discovery. In: *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 653–658. ACM (2004)
18. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph. (TOG)* **23**(3), 309–314 (2004)
19. Sanjay-Gopal, S., Hebert, T.J.: Bayesian pixel classification using spatially variant finite mixtures and the generalized em algorithm. *IEEE Trans. Image Process.* **7**(7), 1014–1028 (1998)

20. Sfikas, G., Nikou, C., Galatsanos, N.: Edge preserving spatially varying mixtures for image segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–7. IEEE (2008)
21. Shen, J., Du, Y., Wang, W., Li, X.: Lazy random walks for superpixel segmentation. *IEEE Trans. Image Process.* **23**(4), 1451–1462 (2014)
22. Shental, N., Zomet, A., Hertz, T., Weiss, Y.: Learning and inferring image segmentations using the gbp typical cut algorithm. In: 9th IEEE International Conference on Computer Vision, pp. 1243–1250. IEEE (2003)
23. Sun, J., Qu, H., Chakrabarti, D., Faloutsos, C.: Neighborhood formation and anomaly detection in bipartite graphs. In: 5th IEEE International Conference on Data Mining, pp. 418–425. IEEE (2005)