# Saliency Detection Based on Manifold Ranking and Refined Seed Labels

Shan Su, Ziguan Cui(✉), Yutao Yao, Zongliang Gan, Guijin Tang, and Feng Liu

Image Processing and Image Communication Lab,
Nanjing University of Posts and Telecommunications, Nanjing 210003, China
cuizg@njupt.edu.cn

**Abstract.** Graph-based manifold ranking has been exploited for saliency detection with seed labels. However, when the selected labels are not accurate, these methods can't emphasize the foreground and suppress the background effectively. In this paper, we propose a novel saliency detection approach through manifold ranking and refined seed labels. We first construct a half-two layers graph based on the nodes after superpixel segmentation, which is generated by connecting each node to neighboring nodes and the half of the most similar nodes that share common boundaries with neighboring nodes. Then we compute superpixel saliency using manifold ranking with refined labels by two-step manner. After clustering superpixel with K-means, the background-based detection is obtained by refined background labels, which are those clusters containing boundary. The foreground-based detection is acquired with the refined foreground labels which are the complete cluster after thresholding the background-based detection. The proposed method has been tested on four universal datasets: ASD, CSSD, ECSSD and SOD. Experimental results show that our method performs better than prior similar state-of-the-art methods in various assessment indexes.

**Keywords:** Saliency detection · Manifold ranking · K-means · Graph model

## 1 Introduction

Recently, salient object detection has acquired much research interest, which aims to locate interesting and important regions in an image [1]. The output of saliency can be benefit to numerous applications such as object recognition, object tracking, image segmentation, image compression, image retrieval, and image quality assessment.

Generally, based on data processing mechanisms, saliency detection can be categorized as either bottom-up [1–4] or top-down [5–7] schemes. The bottom-up model is a fast, unconscious, data-driven and open-loop visual attention mechanism which base on the characteristics of the visual scene. In contrast, top-down model is a slow, conscious, task-driven and closed-loop visual attention mechanism which relies on the observer's expectations. Saliency detection methods can also be classified as salient region detection and eye fixation prediction. In this paper, we focus on the bottom-up salient object detection task.


Y. Zhao et al. (Eds.): ICIG 2019, LNCS 11901, pp. 284–296, 2019.
https://doi.org/10.1007/978-3-030-34120-6_23

Most bottom-up saliency detection methods are based on low-level features, such as color contrast, Euclidean distance and orientation. Itti et al. [1] proposed a conceptual model for saliency detection by performing multi-feature extraction and multi-scale decomposition of the input image, then fused the feature map linearly. Cheng et al. [3] presented a histogram contrast-based (HC) method, which considered the regional contrast with respect to the entire image and pixel-wise color separation to produce saliency map. Zhai et al. [8] calculated the global luminance contrast (LC) of pixel over the entire image to detect saliency. Hou et al. [9] established a spectral residual (SR) model of the image to obtain the saliency map. Achanta et al. [10] computed the saliency likelihood of each pixel by a frequency-tuned method based on luminance and color. By combining color uniqueness and spatial distribution, Perazzi et al. [11] applied a high-dimensional Gaussian filter to generate pixel-map. Zhou et al. [12] generated pixel saliency map by integrating diffusional compactness and local contrast (DCLC) cues.

However, those low-level features based methods maybe ignore the intrinsic connection between pixels and regions in images. To solve this problem, the graph-based methods are put forward. Harel et al. [13] explored a graph based visual saliency algorithm, which uses certain features to form activation map and then highlights the area of interest by normalizing. Gopalakrishnan et al. [2] detected seed nodes by Markov random walk model, which is carried out with the sparse k-regular graph and the complete graph, then the estimated location of the most notable region in an image is determined by seed nodes. By graph-based manifold ranking (MR) method, Yang et al. [4] utilized the boundary regions as background labels to generate initial saliency map and extracted foreground labels from initial map to obtain the final saliency map. In [14], a co-transduction algorithm is devised to fuse both boundary and objectness labels based on inter propagation scheme (LPS). Zhang et al. [15] adopted a linear scheme to fuse texture saliency map and color saliency map (TC) by manifold ranking. Zhou et al. [16] detected salient regions via diffusion process on sparse graph (DSG), and calculated background seed vectors by a compactness measure. Yuan et al. [17] removed foreground labels from background prior by reversion correction and built the regularized random (RCRR) walk ranking model to generate pixel-wise saliency map.

Among the graph-based methods, the boundary-based model outperforms most of the state-of-the-art saliency detection methods and is more computationally efficient. However, there still are some drawbacks that prevent from optimal performance. Firstly, most constructed graphs such as proposed in [4, 17] are full connected, each node connects to those nodes neighboring it as well as sharing common boundaries with its neighboring nodes. However, if the nodes of salient objects are inhomogeneous or incoherent, the full connected graph may lead to errors and seldom detect complete foreground. Secondly, background regions usually have a wider distribution over the entire image. The four boundaries of the image are treated as background labels for background-based saliency detection in [4, 17]. It's insufficient and maybe fail due to the negative influence when foreground objects touch the boundary.

In order to overcome above-mentioned problems, we propose half-two layers graph and select accurate seed labels by clustering for saliency detection. Firstly, we construct a half-two layers graph model, which is generated by connecting each node to

neighboring nodes and the half of the most similar nodes that share common boundaries with neighboring nodes. This method effectively removes redundant nodes and fully uses the local spatial information. Then we apply the K-means to cluster image superpixels and those clusters containing boundary are regarded as background. Due to foreground objects may touch the boundary, we employ reversion correction method [17] to remove foreground in these background labels. The background saliency map is obtained based on background labels by manifold ranking. Finally, we binarize the background saliency map and use those complete clusters as the foreground labels. And we use foreground labels based manifold ranking method to get the final saliency map.

The residual of this paper is organized as follows. Section 2 shows the overall flow of our algorithm, including the construction of the graph model, the selection of foreground labels and background labels. The experimental results for ASD, CSSD, ECSSD and SOD datasets are shown in Sects. 3, and 4 is conclusion.

## 2 The Proposed Method

The framework of our proposed algorithm is shown in Fig. 1.

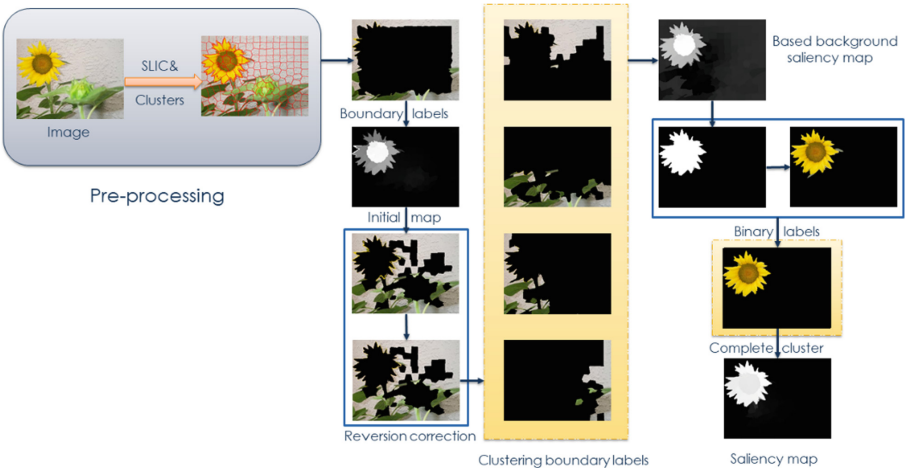



**Fig. 1.** Principal steps of our method.

Firstly, we perform the SLIC algorithm [18] to generate superpixels and construct a half-two layers graph. Secondly, we employ the K-means to cluster the superpixels. Thirdly, we select the background labels that those clusters contain boundary and remove the foreground labels. Finally, the complete cluster is regarded as foreground label after using an adaptive threshold, and then we apply the manifold ranking [16] to obtain the final saliency map.

### 2.1 Graph Construction and Clustering

In order to improve the performance of salient object detection, we use the SLIC algorithm to divide the input image into homogeneous and compact superpixels using the color means. Then we construct a graph $\mathrm{G} = (\mathrm{V}, \mathrm{E})$ depend on the superpixels of image, where each node V denotes a superpixel produced by the SLIC algorithm and edge E denote that $V_i$ connects to $V_j$. The node set V consists of superpixels $\mathrm{X} = \{x_1, \ldots, x_q, x_{q+1}, \ldots, x_n\} \in \mathbb{R}^m$. Some nodes are used as queries, and the remaining nodes need to be ranked according to their relevance to the queries. Let $\mathrm{f} : \mathrm{X} \rightarrow \mathbb{R}$ denote a ranking function, which assigns a ranking value $f_i$ to each block $x_i$, and f can be regarded as a vector $\mathrm{f} = [f_1, \ldots, f_n]^T$. Let $\mathrm{y} = [y_1, \ldots, y_n]^T$ denotes an indication vector, where $y_i = 1$ if $x_i$ is a query, and $y_i = 0$ otherwise. We use manifold ranking [4] as the ranking function, which is written as:

$$\mathrm{f} = (D - \alpha W)^{-1} y \tag{1}$$

where $\alpha$ denote a constant, the affinity matrix is denoted by $W = \{w_{ij}\}_{N \times N}$, and $D = diag\{d_{11}, d_{22}, \ldots, d_{NN}\}$ is the degree matrix, where $d_{ii} = \sum_j w_{ij}$. More manifold ranking details could be found in [4, 19].

We define the weight $w_{ij}$ between two nodes as

$$w_{ij} = e^{\frac{-\left\|c_i - c_j\right\|}{\sigma^2}} \tag{2}$$

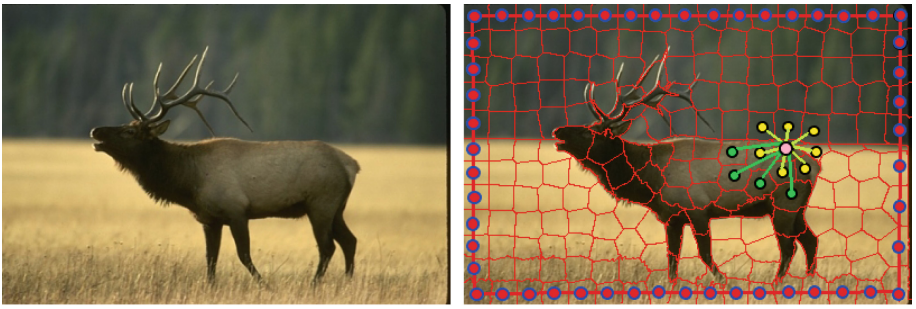
where $c_i$ and $c_j$ denote the mean of color of nodes $V_i$ and $V_j$ in Lab color space, $\sigma$ is constant factor which controls the weight.

Generally, most graph-based methods construct a full connection, each node connects to those neighboring nodes $D_1(\mathrm{j})$ as well as those nodes sharing common boundaries with its neighboring nodes $D_2(\mathrm{j})$, which may obtain erroneous local relation. Thus, in this paper, we propose a half-two layers graph for calculating saliency. As shown in Fig. 2, the half-two layers graph generated by connecting each node to its neighboring nodes and the half of the most similar nodes $p$ that share common boundaries with neighboring nodes. It's well known that the second layer contains some local information, and some redundant information is adulterated in. To reduce redundancy and retain more local information, we retain the half of the most similar nodes, which is denoted as:

$$\mathrm{D(p)} = \left\{\mathrm{q} \in D_2(\mathrm{j}) : w_{ij} > v\right\} \tag{3}$$

where $v$ is the weight means of the second layer nodes $D_2(\mathrm{j})$, $q$ is the node in $D_2(\mathrm{j})$, and $p$ is the node whose weight larger than $v$.

Moreover, each node of the four boundaries of the image must be connected in pairs, and we describe the image as a closed-loop graph. Thus, the constructed graph model effectively removes redundant nodes and fully uses the local spatial distribution feature, which shows the obvious advantages compared with others graph models.



(a) (b)

**Fig. 2.** The two-half layer graph model. (a) Input image. (b) Edge connection between nodes. A node (illustrated by a pink dot) connects to both its adjacent nodes (yellow dot) and the half of the most similar nodes (green dot) sharing common boundaries with its adjacent nodes. Each pair of boundary nodes are connected to each other (red dot and connection). (Color figure online)

We then employ K-means algorithm to cluster the N superpixels of the image into K clusters. Considering Lab color space is more related to human perception, we use three-dimensional Lab color feature to cluster.

### 2.2 Background-Based Saliency Estimation

Usually most of background regions are near the boundary, which are sparse and have a wider spatial distribution over the entire image compared with foreground regions. However, it's not adequate that simply utilizes the boundary labels as background labels. Therefore, we extend the background labels by clustering the image, each cluster contains one superpixel at least, and those clusters that contain boundary background are regarded as background labels. With the increase of the background labels, when calculating the background prior of the image, it's more effective to detect the foreground saliency object and uniformly highlight the entire salient region.

To select the background labels more accurately, we first calculate the initial saliency map using the boundary regions as [4] and remove the boundary-adjacent foreground regions from the boundary clusters by reverse correction method [17]. The initial map is generated via the separation and combination (SC) scheme, that is, we construct four background prior maps with boundary labels and then multiply them each other as the initial map. Then we use reverse correction method to mark the foreground regions with 1 and the background regions with 2. Specifically, for each boundary, the mean of the cluster that contains boundary background is called $L_{label}$. Given pre-defined threshold Th1 = 1, if Th1 smaller than $L_{label}$, we will repute that those clusters contain foreground regions in background regions, and then we will remove those regions and acquire exact background labels. Figure 3 shows examples of background labels, we can see that compare with general background labels (Fig. 3 (b)) and undoing reverse correction background labels (Fig. 3(c)), our background labels (Fig. 3(d)) are more precise.

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

**Fig. 3.** Examples of background labels. From left to right: (a) Input image. (b) General background labels. (c) Not reverse correction background labels. (d) Our background labels.

After, we calculate background saliency maps by the manifold ranking. Taking top labels as an example, the queries are the exact background labels and the remaining regions are ranked. Thus, the indication vector $y_i$ is obtained, and all the nodes are ranked based on Eq. (1) in $f_b$, which means each superpixel relevance to the exact background labels. The background saliency $S_b$ based on top labels is calculated as:

$$S_b(i) = 1 - f_b(i) \tag{4}$$

where $f_b(i)$ denotes the normalize vector, and the range of $f_b(i)$ is between 0 and 1.

We generate the other three saliency maps using the queries that selected via the similar method. And then the background-based saliency $S_B$ is obtained by the following procedure:

$$S_B(i) = \prod_{b=1}^{k} S_b(i) \tag{5}$$

Where $k$ denotes the number of boundary.

### 2.3 Foreground-Based Saliency Estimation

Through the above steps, the most saliency regions are highlighted. However, there are some background regions which may not be inhibited. By the adaptive threshold method could diminish this problem, but the picked foreground labels may adulterate some background labels, as is shown in Fig. 4(b). To select the foreground labels more reasonable, we regard the extracted labels belonging to the complete clusters as foreground labels.

(a) (b) (c)

**Fig. 4.** Example of foreground labels. From left to right: (a) Input Image. (b) Adaptive threshold labels. (c) Adaptive threshold labels and the same cluster labels.

We separate the background saliency map by binary threshold, which exploits the adaptive threshold Th2 defined as the mean saliency over the whole saliency map. If $S_B(\mathrm{i}) > \mathrm{Th2}$, the $S_B(i)$ is treated as foreground labels. The K-means algorithm divides the image into three categories: intra-object, intra-background and object-background, so we deem that those complete clusters are final foreground labels after adaptive threshold, as is shown in Fig. 4(c). Then we calculate the saliency map with final queries in each superpixel using Eq. (1). The foreground-based saliency map $S_F$ is defined:

$$S_F(i) = \bar{f}(i) \tag{6}$$

where $\bar{f}(i)$ denote the normalized vector.

By the above method, the final saliency map will be greatly improved. As shown in Fig. 5. We notice that our method can stress the foreground evenly and suppress the background in effect.

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

**Fig. 5.** An saliency example by our method. (a) Input image. (b) GT. (c) Saliency map based on half-two layers, (d) Saliency map based on background labels. (e) Saliency map based on foreground labels.

## 3 Experimental Results

### 3.1 Experimental Setup

We test the proposed method on four datasets. The ASD dataset [10] contains 1000 images. The second one is SOD dataset [20], which contains 300 images with multiple objects. The CSSD [21] is the third dataset, which contains diversified patterns in both

the foreground and background. And the last one is ECSSD dataset [21], which is an extension of CSSD to express natural circumstances.

There are four parameters in the experiment which need to be set. In all experiments, we empirically set the number of superpixel nodes N = 200. $\sigma$ is the edge weight, which controls the fall-off rate of the exponential function. In manifold ranking algorithm, $\alpha$ balances the smooth and fitting constraints. We empirically set $\sigma = 0.1$, and $\alpha = 0.99$. The parameter K is the number of cluster in K-means, through experiment we set K = 70. As shown in Fig. 6, we varied it from 30 to 90 in intervals of 10 to determine an appropriate value for K with ASD dataset.

To evaluate the performance of different methods, we use the average precision-recall curve and the F-measure as evaluation criterion. We vary the threshold from 0 to 255 and compute the precision and recall at each threshold by comparing the binary mask and the ground truth to compare the accuracy of the different saliency maps. Then we apply the sequence of precision-recall pairs to plot the precision-recall curve. The F-measure is calculated using:

$$F_\beta = \frac{\left(1+\beta^2\right) Precision \times Recall}{\beta^2 Precision + Recall} \tag{7}$$

Following [4], we set $\beta^2 = 0.3$.

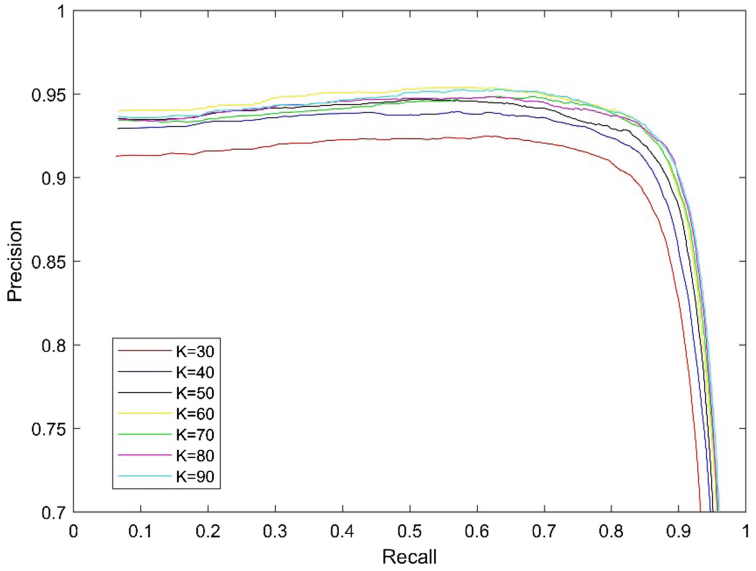



**Fig. 6.** Influence of K on the image.

### 3.2 Performance Comparison

We compare our method with 8 state-of-the-art algorithms, namely HC [3], MR [4], LC [8], DCLC [12], LPS [14], TC [15], DSG [16], and RCRR [17]. As shown in Fig. 7, our method acquires better subjective performance, and uniformly stress foreground salient object and suppress background even for complex natural images.

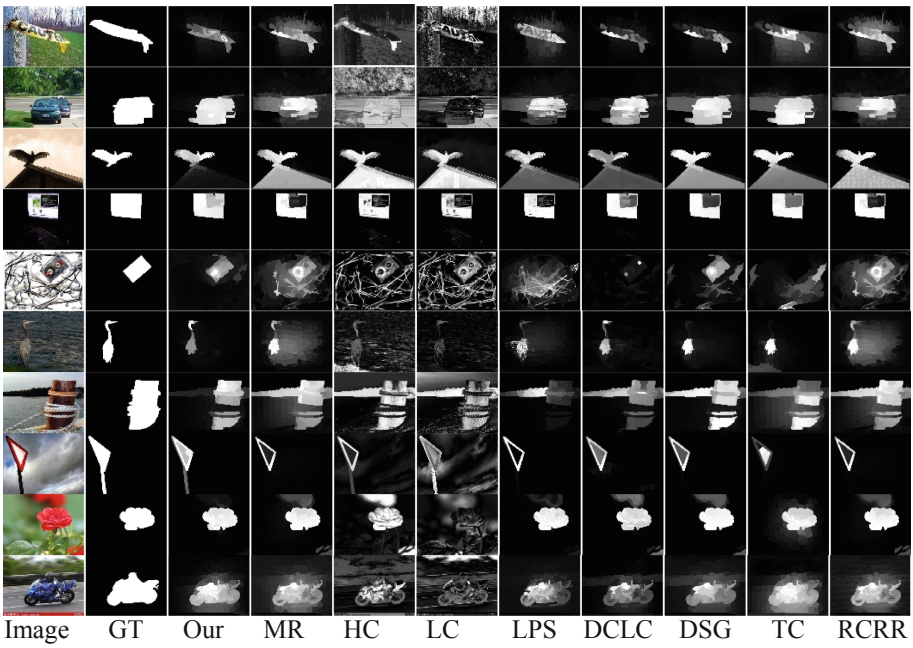



**Fig. 7.** Saliency detection results of different methods. The proposed algorithm consistently highlight foreground and suppress background.

We calculate P-R curve and F-measure on four databases. The result of F-measure is listed in Table 1. The P-R curves are shown in Fig. 8 and the precision, recall and F-measure indexes are shown in Fig. 9. Compared with other representative methods, the performance of our method is better in F-measure for CSSD, ECSSD and SOD databases. From the P-R curves, our algorithm performs also well, and it is competitive to DCLC, MR, and RCRR. Although the performance of the P-R curve does not surpass other algorithms by a large margin, our method obtains better subjective saliency map.

**Table 1.** F-measure results on ASD, CSSD, ECSSD and SOD databases.

| | Our | MR | HC | LC | LPS |
|---|---|---|---|---|---|
| ASD | 0.9115 | 0.9067 | 0.7264 | 0.5477 | 0.9009 |
| CSSD | **0.8377** | 0.8197 | 0.5196 | 0.4680 | 0.7922 |
| ECSSD | **0.7425** | 0.7355 | 0.4205 | 0.3793 | 0.6962 |
| SOD | **0.6395** | 0.6294 | 0.4157 | 0.4028 | 0.5868 |
| | DCLC | DSG | TC | RCRR | |
| ASD | **0.9121** | **0.9164** | 0.8600 | 0.9067 | |
| CSSD | 0.8275 | **0.8352** | 0.7183 | 0.8213 | |
| ECSSD | 0.7311 | **0.7445** | 0.6703 | 0.7390 | |
| SOD | 0.6169 | 0.6211 | 0.5785 | **0.6311** | |

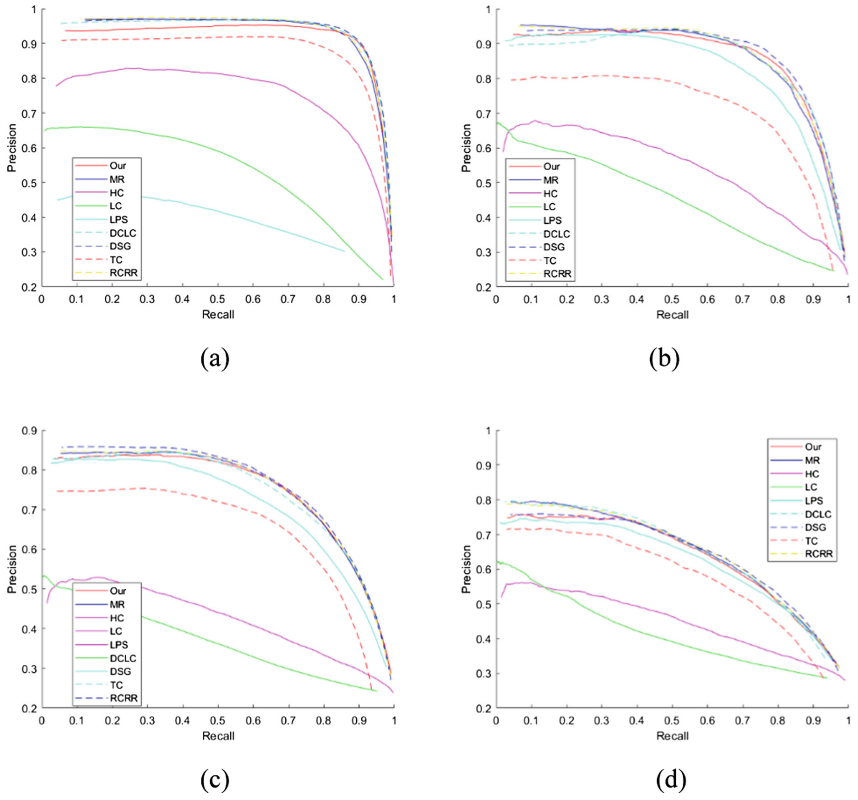



**Fig. 8.** Average precision-recall curves of the proposed method compared with 8 state-of-the-art methods. (a) the ASD database. (b) the CSSD database. (c) the ECSSD database. (d) the SOD database.

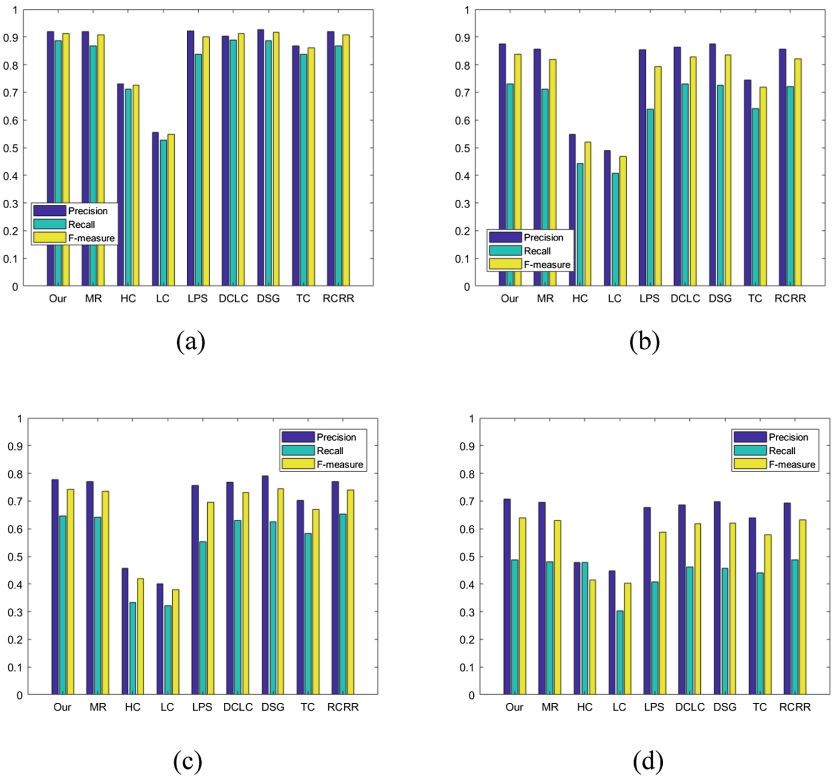



**Fig. 9.** F-measure of the proposed method compared with 8 state-of-the-art methods. (a) The ASD database. (b) The CSSD database. (c) The ECSSD database. (d) The SOD database.

### 3.3 Running Time

The running time is tested on a 64-bit PC with Intel Core i5-3337U CPU @ 1.80 GHz and 4 GB RAM. Average running time is calculated on ASD database. We compare five methods in recent years, and the results are shown in Table 2. Our method is slightly slower than MR and DSG, but it's faster than LPS, LC and RCRR. Considering the overall evaluation performances, our method acquires better trade-off between performance and complexity.

**Table 2.** Running time test results (seconds per image).

| Method | Our | MR | LPS | DSG | TC | RCRR |
|---|---|---|---|---|---|---|
| Time (s) | 0.834 | 0.667 | 1.287 | 0.630 | 1.664 | 1.531 |

## 4 Conclusion

We propose a bottom-up method to extract saliency region by calculating the relevance using manifold ranking with refined background and foreground labels. Our proposed half-two layers graph model alleviates the limitations in the prior graph models. In addition, we pick up the more precise labels using the cluster with k-means algorithm. The refined background and foreground labels can help to improve the performance of manifold ranking. By comparing with state-of-the-art saliency algorithms on four databases, it's confirmed that our method acquires better performance and can suppress background region and highlight foreground region accurately.

**Acknowledgements.** This work is supported by National Natural Science Foundation of China (NSFC) (61501260, 61471201, 61471203), Jiangsu Province Higher Education Institutions Natural Science Research Key Grant Project (13KJA510004), The peak of six talents in Jiangsu (RLD201402), and "1311 Talent Program" of NJUPT.

## References

1. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE TPAMI **20**(11), 1254–1259 (1998)
2. Gopalakrishnan, V., Hu, Y., Rajan, D.: Random walks on graphs for salient object detection in images. IEEE TIP **19**(12), 3232–3242 (2010)
3. Cheng, M.M., Zhang, G.X., Mitra, N.J., Huang, X., Hu, S.M.: Global contrast based salient region detection. In: CVPR, pp. 409–416 (2011)
4. Yang, C., Zhang, L., Lu, H., Ruan, X., Yang, M.H.: Saliency detection via graph-based manifold ranking. In: CVPR, pp. 3166–3173 (2013)
5. Gao, D., Vasconcelos, N.: Discriminant saliency for visual recognition from cluttered scenes. In: Advances in Neural Information Processing Systems, pp. 481–488 (2004)
6. Yang, J., Yang, M.H.: Top-down visual saliency via joint CRF and dictionary learning. In: CVPR, pp. 2296–2303 (2012)
7. Itti, L., Sihite, D.N., Borji, A.: Probabilistic learning of task-specific visual attention. In: CVPR, pp. 470–477 (2012)
8. Zhai, Y., Shah, M.: Visual attention detection in video sequences using spatiotemporal cues. In: ACM Multimedia, pp. 815–824 (2006)
9. Hou, X., Zhang, L.: Saliency detection: a spectral residual approach. In: CVPR, pp. 1–8 (2007)
10. Achanta, R., Hemami, S., Estrada, F., Susstrunk, S.: Frequency-tuned salient region detection. In: CVPR, pp. 1597–1604 (2009)
11. Perazzi, F., Krahenbuhl, P., Pritch, Y., Hornung, A.: Saliency filters: contrast based filtering for salient region detection. In: CVPR, pp. 733–740 (2012)
12. Zhou, L., Yang, Z., Yuan, Q., Zhou, Z., Hu, D.: Salient region detection via integrating diffusion-based compactness and local contrast. IEEE TIP **24**(11), 3308–3320 (2015)
13. Harel, J., Koch, C., Pietro, P.: Graph-based visual saliency. In: Advances in Neural Information Processing Systems, pp. 545–552 (2006)
14. Li, H., Lu, H., Lin, Z., Shen, X., Price, B.: Inner and inter label propagation: salient object detection in the wild. IEEE TIP **24**(10), 3176–3186 (2015)

15. Zhang, Q., Lin, J., Tao, Y., Li, W., Shi, Y.: Salient object detection via color and texture cues. Neurocomputing **243**, 35–48 (2017)
16. Zhou, L., Yang, Z., Zhou, Z., Hu, D.: Salient region detection using diffusion process on a 2-layer sparse graph. IEEE TIP **26**(12), 5882–5894 (2017)
17. Yuan, Y., Li, C., Kim, J., Cai, W., Feng, D.D.: Reversion correction and regularized random walk ranking for saliency detection. IEEE TIP **27**(3), 1311–1322 (2018)
18. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Susstrunk, S.: SLIC superpixels compared to state-of-the-art superpixel methods. IEEE TPAMI **34**(11), 2274–2282 (2012)
19. Zhou, D., Weston, J., Gretton, A., Bousquet, O., Scholkopf, B.: Ranking on data manifolds. In: Advances in Neural Information Processing Systems, pp. 169–176 (2014)
20. Movahedi, V., Elder, J.H.: Design and perceptual validation of performance measures for salient object segmentation. In: CVPRW, pp. 49–56 (2010)
21. Yan, Q., Xu, L., Shi, J., Jia, J.: Hierarchical saliency detection. In: CVPR, pp. 1155–1162 (2013)