Saliency Detection via Graph-Based Manifold Ranking

Chuan Yang[1], Lihe Zhang[1], Huchuan Lu[1], Xiang Ruan[2], and Ming-Hsuan Yang[3]
[1]Dalian University of Technology [2]OMRON Corporation [3]University of California at Merced

Abstract

Most existing bottom-up methods measure the foreground saliency of a pixel or region based on its contrast within a local context or the entire image, whereas a few methods focus on segmenting out background regions and thereby salient objects. Instead of considering the contrast between the salient objects and their surrounding regions, we consider both foreground and background cues in a different way. We rank the similarity of the image elements (pixels or regions) with foreground cues or background cues via graph-based manifold ranking. The saliency of the image elements is defined based on their relevances to the given seeds or queries. We represent the image as a close-loop graph with superpixels as nodes. These nodes are ranked based on the similarity to background and foreground queries, based on affinity matrices. Saliency detection is carried out in a two-stage scheme to extract background regions and foreground salient objects efficiently. Experimental results on two large benchmark databases demonstrate the proposed method performs well when against the state-of-the-art methods in terms of accuracy and speed. We also create a more difficult benchmark database containing 5,172 images to test the proposed saliency model and make this database publicly available with this paper for further studies in the saliency field.

1. Introduction

The task of saliency detection is to identify the most important and informative part of a scene. It has been applied to numerous vision problems including image segmentation [11], object recognition [28], image compression [16], content based image retrieval [8], to name a few. Saliency methods in general can be categorized as either bottom-up or top-down approaches. Bottom-up methods [1, 2, 6, 7, 9–12, 14, 15, 17, 21, 24, 25, 27, 32, 33, 37] are data-driven and pre-attentive, while top-down methods [23, 36] are task-driven that entails supervised learning with class labels. We note that saliency models have been developed for eye fixation prediction [6, 14, 15, 17, 19, 25, 33] and salient object detection [1, 2, 7, 9, 23, 24, 32]. The former focuses on identifying a few human fixation locations on natural images, which is important for understanding human attention. The latter is to accurately detect where the salient object should be, which is useful for many high-level vision tasks. In this paper, we focus on the bottom-up salient object detection tasks.

Salient object detection algorithms usually generate bounding boxes [7, 10], binary foreground and background segmentation [12, 23, 24, 32], or saliency maps which indicate the saliency likelihood of each pixel. Liu et al. [23] propose a binary saliency estimation model by training a conditional random field to combine a set of novel features. Wang et al. [32] analyze multiple cues in a unified energy minimization framework and use a graph-based saliency model [14] to detect salient objects. In [24] Lu et al. develop a hierarchical graph model and utilize concavity context to compute weights between nodes, from which the graph is bi-partitioned for salient object detection. On the other hand, Achanta et al. [1] compute the saliency likelihood of each pixel based on its color contrast to the entire image. Cheng et al. [9] consider the global region contrast with respect to the entire image and spatial relationships across the regions to extract saliency map. In [11] Goferman et al. propose a context-aware saliency algorithm to detect the image regions that represent the scene based on four principles of human visual attention. The contrast of the center and surround distribution of features is computed based on the Kullback-Leibler divergence for salient object detection [21]. Xie et al. [35] propose a novel model for bottom-up saliency within the Bayesian framework by exploiting low and mid level cues. Sun et al. [30] improve the Xie's model by introducing boundary and soft-segmentation. Recently, Perazzi et al. [27] show that the complete contrast and saliency estimation can be formulated in a unified way using high-dimensional Gaussian filters. In this work, we generate a full-resolution saliency map for each input image.

Most above-mentioned methods measure saliency by measuring local center-surround contrast and rarity of features over the entire image. In contrast, Gopalakrishnan et al. [12] formulate the object detection problem as a binary segmentation or labelling task on a graph. The most salient

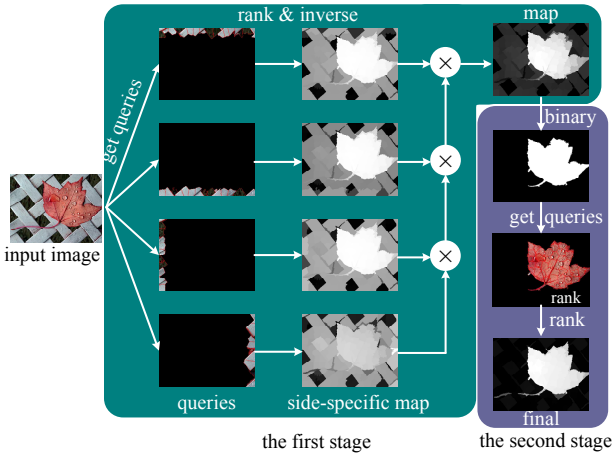


Figure 1. Diagram of our proposed model.

seed and several background seeds are identified by the behavior of random walks on a complete graph and a k-regular graph. Then, a semi-supervised learning technique is used to infer the binary labels of the unlabelled nodes. Recently, a method that exploits background priors is proposed for saliency detection [34]. The main observation is that the distance between a pair of background regions is shorter than that of a region from the salient object and a region from the background. The node labelling task (either salient object or background) is formulated as an energy minimization problem based on this criteria.

We observe that background often presents local or global appearance connectivity with each of four image boundaries and foreground presents appearance coherence and consistency. In this work, we exploit these cues to compute pixel saliency based on the ranking of superpixels. For each image, we construct a close-loop graph where each node is a superpixel. We model saliency detection as a manifold ranking problem and propose a two-stage scheme for graph labelling. Figure 1 shows the main steps of the proposed algorithm. In the first stage, we exploit the boundary prior [13, 22] by using the nodes on each side of image as labelled background queries. From each labelled result, we compute the saliency of nodes based on their relevances (i.e, ranking) to those queries as background labels. The four labelled maps are then integrated to generate a saliency map. In the second stage, we apply binary segmentation on the resulted saliency map from the first stage, and take the labelled foreground nodes as salient queries. The saliency of each node is computed based on its relevance to foreground queries for the final map.

To fully capture intrinsic graph structure information and incorporate local grouping cues in graph labelling, we use manifold ranking techniques to learn a ranking function, which is essential to learn an optimal affinity matrix [20]. Different from [12], the proposed saliency detection algorithm with manifold ranking requires only seeds from one class, which are initialized with either the boundary priors or foreground cues. The boundary priors are proposed inspired on the recent works of human fixations on images [31], which shows that humans tend to gaze at the center of images. These priors have also been used in image segmentation and related problems [13, 22, 34]. In contrast, the semi-supervised method [12] requires both background and salient seeds, and generates a binary segmentation. Furthermore, it is difficult to determine the number and locations of salient seeds as they are generated by random walks, especially for the scenes with different salient objects. This is a known problem with graph labelling where the results are sensitive to the selected seeds. In this work, all the background and foreground seeds can be easily generated via background priors and ranking background queries (or seeds). As our model incorporates local grouping cues extracted from the entire image, the proposed algorithm generates well-defined boundaries of salient objects and uniformly highlights the whole salient regions. Experimental results using large benchmark data sets show that the proposed algorithm performs efficiently and favorably against the state-of-the-art saliency detection methods.

2. Graph-Based Manifold Ranking

The graph-based ranking problem is described as follows: given a node as a query, the remaining nodes are ranked based on their relevances to the given query. The goal is to learn a ranking function, which defines the relevance between unlabelled nodes and queries.

2.1. Manifold Ranking

In [39], a ranking method that exploits the intrinsic manifold structure of data (such as image) for graph labelling is proposed. Given a dataset $X = \{x_1, \ldots, x_l, x_{l+1}, \ldots, x_n\} \in \mathbb{R}^{m \times n}$, some data points are labelled queries and the rest need to be ranked according to their relevances to the queries. Let $\mathbf{f}$: $X \rightarrow \mathbb{R}^n$ denote a ranking function which assigns a ranking value f_i to each point x_i, and $\mathbf{f}$ can be viewed as a vector $\mathbf{f} = [f_1, \ldots, f_n]^T$. Let $\mathbf{y} = [y_1, y_2, \ldots, y_n]^T$ denote an indication vector, in which $y_i = 1$ if x_i is a query, and $y_i = 0$ otherwise. Next, we define a graph $G = (V, E)$ on the dataset, where the nodes V are the dataset X and the edges E are weighted by an affinity matrix $\mathbf{W} = [w_{ij}]_{n \times n}$. Given G, the degree matrix is $\mathbf{D} = \text{diag}\{d_{11}, \ldots, d_{nn}\}$, where $d_{ii} = \sum_j w_{ij}$. Similar to the PageRank and spectral clustering algorithms [5, 26], the optimal ranking of queries are computed by solving the following optimization problem:

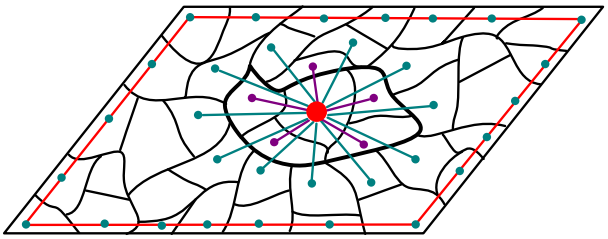


Figure 2. Our graph model. The red line along the four sides indicates that all the boundary nodes are connected with each other.

$$\mathbf{f}^* = \arg\min_{f} \frac{1}{2}\left(\sum_{i,j=1}^{n} w_{ij}\left\|\frac{f_i}{\sqrt{d_{ii}}} - \frac{f_j}{\sqrt{d_{jj}}}\right\|^2 + \mu\sum_{i=1}^{n}\|f_i - y_i\|^2\right), \tag{1}$$

where the parameter μ controls the balance of the smoothness constraint (the first term) and the fitting constraint (the second term). That is, a good ranking function should not change too much between nearby points (smoothness constraint) and should not differ too much from the initial query assignment (fitting constraint). The minimum solution is computed by setting the derivative of the above function to be zero. The resulted ranking function can be written as:

$$\mathbf{f}^* = (\mathbf{I} - \alpha\mathbf{S})^{-1}\mathbf{y}, \tag{2}$$

where $\mathbf{I}$ is an identity matrix, $\alpha = 1/(1+\mu)$ and $\mathbf{S}$ is the normalized Laplacian matrix, $\mathbf{S} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$.

The ranking algorithm [39] is derived from the work on semi-supervised learning for classification [38]. Essentially, manifold ranking can be viewed as an one-class classification problem [29], where only positive examples or negative examples are required. We can get another ranking function by using the unormalized Laplacian matrix in Eq. 2:

$$\mathbf{f}^* = (\mathbf{D} - \alpha\mathbf{W})^{-1}\mathbf{y}. \tag{3}$$

We compare the saliency results using Eq. 2 and Eq. 3 in the experiments, and the latter achieves better performance (See Figure 8). Hence, we adopt Eq. 3 in this work.

2.2. Saliency Measure

Given an input image represented as a graph and some salient query nodes, the saliency of each node is defined as its ranking score computed by Eq. 3 which is rewritten as $\mathbf{f}^* = \mathbf{A}\mathbf{y}$ to facilitate analysis. The matrix $\mathbf{A}$ can be regarded as a learnt optimal affinity matrix which is equal to $(\mathbf{D} - \alpha\mathbf{W})^{-1}$. The ranking score $\mathbf{f}^*(i)$ of the i-th node is the inner product of the i-th row of $\mathbf{A}$ and $\mathbf{y}$. Because $\mathbf{y}$ is a binary indicator vector, $\mathbf{f}^*(i)$ can also be viewed as the sum of the relevances of the i-th node to all the queries.

In the conventional ranking problems, the queries are manually labelled with the ground-truth. However, as queries for saliency detection are selected by the proposed algorithm, some of them may be incorrect. Thus, we need to compute a degree of confidence (i.e., the saliency value) for each query, which is defined as its ranking score ranked by the other queries (except itself). To this end, we set the diagonal elements of $\mathbf{A}$ to 0 when computing the ranking score by Eq. 3. We note that this seemingly insignificant process has great effects on the final results. If we compute the saliency of each query without setting the diagonal elements of $\mathbf{A}$ to 0, its ranking value in $\mathbf{f}^*$ will contain the relevance of this query to itself, which is meaningless and often abnormally large so as to severely weaken the contributions of the other queries to the ranking score. Lastly, we measure the saliency of nodes using the normalized ranking score $\bar{\mathbf{f}}^*$ when salient queries are given, and using $\mathbf{1} - \bar{\mathbf{f}}^*$ when background queries are given.

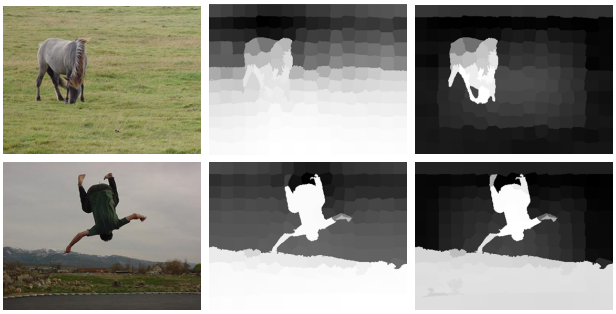


Figure 3. Graph labelling results using the top boundary prior. Left: input images. Center: Results without enforcing the geodesic distance constraints. Right: Results with geodesic distance constraints.

3. Graph Construction

We construct a single layer graph $G = (V, E)$ as shown in Figure 2, where V is a set of nodes and E is a set of undirected edges. In this work, each node is a superpixel generated by the SLIC algorithm [3]. As neighboring nodes are likely to share similar appearance and saliency values, we use a k-regular graph to exploit the spatial relationship. First, each node is not only connected to those nodes neighboring it, but also connected to the nodes sharing common boundaries with its neighboring node (See Figure 2). By extending the scope of node connection with the same degree of k, we effectively utilize local smoothness cues. Second, we enforce that the nodes on the four sides of image are connected, i.e., any pair of boundary nodes are considered to be adjacent. Thus, we denote the graph as the close-loop graph. This close-loop constraint significantly improves the performance of the proposed method as it tends to reduce the geodesic distance of similar superpixels, thereby improving the ranking results. Figure 3 shows some examples where the ranking results with and without these constraints. We note that these constraints work well when the

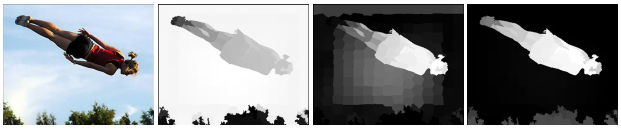


Figure 4. Saliency maps using different queries. From left to right: input image, result of using all the boundary nodes together as queries, result of integrating four maps from each side, result of ranking with foreground queries.

salient objects appear near the image boundaries or some of the background regions are not the same.

With the constraints on edges, it is clear that the constructed graph is a sparsely connected. That is, most elements of the affinity matrix $\mathbf{W}$ are zero. In this work, the weight between two nodes is defined by

$$w_{ij} = e^{-\frac{\|\mathbf{c}_i - \mathbf{c}_j\|}{\sigma^2}} \qquad i, j \in V, \tag{4}$$

where $\mathbf{c}_i$ and $\mathbf{c}_j$ denote the mean of the superpixels corresponding to two nodes in the CIE LAB color space, and σ is a constant that controls the strength of the weight. The weights are computed based on the distance in the color space as it has been shown to be effective in saliency detection [2, 4].

By ranking the nodes on the constructed graph, the inverse matrix $(\mathbf{D} - \alpha\mathbf{W})^{-1}$ in Eq. 3 can be regarded as a complete affinity matrix, i.e., there exists a nonzero relevance value between any pair of nodes on the graph. This matrix naturally captures spatial relationship information. That is, the relevance between nodes is increased when their spatial distance is decreased, which is an important cue for saliency detection [9].

4. Two-Stage Saliency Detection

In this section, we detail the proposed two-stage scheme for bottom-up saliency detection using ranking with background and foreground queries.

4.1. Ranking with Background Queries

Based on the attention theories of early works for visual saliency [17], we use the nodes on the image boundary as background seeds, i.e., the labelled data (query samples) to rank the relevances of all the other regions. Specifically, we construct four saliency maps using boundary priors and then integrate them for the final map, which is referred as the separation/combination (*SC*) approach.

Taking top image boundary as an example, we use the nodes on this side as the queries and other nodes as the unlabelled data. Thus, the indicator vector $\mathbf{y}$ is given, and all the nodes are ranked based on Eq. 3 in $\mathbf{f}^*$, which is a N-dimensional vector (N is the total number of nodes of the graph). Each element in this vector indicates the relevance of a node to the background queries, and its complement is

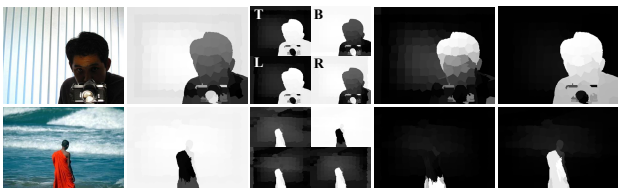


Figure 5. Examples in which the salient objects appear at the image boundary. From left to right: input images, saliency maps using all the boundary nodes together as queries, four side-specific maps, integration of four saliency maps, the final saliency map after the second stage.

the saliency measure. We normalize this vector to the range between 0 and 1, and the saliency map using the top boundary prior, S_t can be written as:

$$S_t(i) = 1 - \overline{\mathbf{f}}^*(i) \qquad i = 1, 2, \ldots, N, \tag{5}$$

where i indexes a superpixel node on graph, and $\overline{\mathbf{f}}^*$ denotes the normalized vector.

Similarly, we compute the other three maps S_b, S_l and S_r, using the bottom, left and right image boundary as queries. We note that the saliency maps are computed with different indicator vector $\mathbf{y}$ while the weight matrix $\mathbf{W}$ and the degree matrix $\mathbf{D}$ are fixed. That is, we need to compute the inverse of the matrix $(\mathbf{D} - \alpha\mathbf{W})$ only once for each image. Since the number of superpixels is small, the matrix inverse in Eq. 3 can be computed efficiently. Thus, the overall computational load for the four maps is low. The four saliency maps are integrated by the following process:

$$S_{bq}(i) = S_t(i) \times S_b(i) \times S_l(i) \times S_r(i). \tag{6}$$

There are two reasons for using the *SC* approach to generate saliency maps. First, the superpixels on different sides are often disimilar which should have large distance. If we simultaneously use all the boundary superpixels as queries (i.e., indicating these suprerpixels are similar), the labelled results are usually less optimal as these nodes are not compactable (See Figure 4). Note that the geodesic distance that we use in Section 3 can be considered as weakly labelled as only a few superpixels are involved (i.e., only the superpixels with low color distance from the sides are considered as similar) whereas the case with all superpixels can be considered as strongly labelled (i.e., all the nodes from the sides are considered as similar). Second, it reduces the effects of imprecise queries, i.e., the ground-truth salient nodes are inadvertently selected as background queries. As shown in the second column of Figure 5, the saliency maps generated using all the boundary nodes are poor. Due to the imprecise labelling results, the pixels with the salient objects have low saliency values. However, as objects are often compact "things" (such as a people or a car) as opposed to incompact

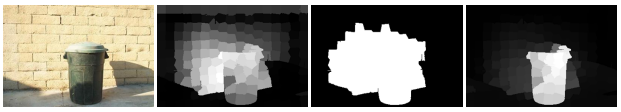


Figure 6. The example in which imprecise salient queries are selected in the second stage. From left to right: input image, saliency map of the first stage, binary segmentation, the final saliency map.

"stuff" (such as grass or sky) and therefore they rarely occupy three or all sides of image, the proposed *SC* approach ensures at least two saliency maps are effective (third column of Figure 5). By integration of four saliency maps, some salient parts of object can be identified (although the whole object is not uniformly highlighted), which provides sufficient cues for the second stage detection process.

While most regions of the salient objects are highlighted in the first stage, some background nodes may not be adequately suppressed (See Figure 4 and Figure 5). To alleviate this problem and improve the results especially when objects appear near the image boundaries, the saliency maps are further improved via ranking with foreground queries.

4.2. Ranking with Foreground Queries

The saliency map of the first stage is binary segmented (i.e., salient foreground and background) using an adaptive threshold, which facilitates selecting the nodes of the foreground salient objects as queries. We expect that the selected queries cover the salient object regions as much as possible (i.e., with high recall). Thus, the threshold is set as the mean saliency over the entire saliency map.

Once the salient queries are given, an indicator vector $\mathbf{y}$ is formed to compute the ranking vector $\mathbf{f}^*$ using Eq. 3. As is carried out in the first stage, the ranking vector $\mathbf{f}^*$ is normalized between the range of 0 and 1 to form the final saliency map by

$$S_{fq}(i) = \overline{\mathbf{f}}^*(i) \qquad i = 1, 2, \ldots, N, \tag{7}$$

where i indexes superpixel node on graph, and $\overline{\mathbf{f}}^*$ denotes the normalized vector.

We note that there are cases where nodes may be incorrectly selected as foreground queries in this stage. Despite some imprecise labelling, salient objects can be well detected by the proposed algorithm as shown in Figure 6. This can be explained as follows. The salient object regions are usually relatively compact (in terms of spatial distribution) and homogeneous in appearance (in terms of feature distribution), while background regions are the opposite. In other words, the intra-object relevance (i.e., two nodes of the salient objects) is statistically much larger than that of object-background and intra-background relevance, which can be inferred from the affinity matrix $\mathbf{A}$. To show this phenomenon, we compute the average intra-object, intra-background and object-background

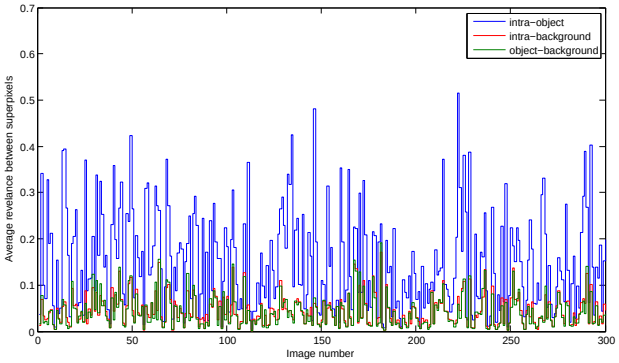


Figure 7. Analysis of the learned relevances between nodes in the affinity matrix $\mathbf{A}$.

relevance values in $\mathbf{A}$ for each of the 300 images sampled from a dataset with ground truth labels [2], which is shown in Figure 7. Therefore, the sum of the relevance values of object nodes to the ground-truth salient queries is considerably larger than that of background nodes to all the queries. That is, background saliency can be suppressed effectively (fourth column of Figure 6). Similarly, in spite of the saliency maps after the first stage of Figure 5 are not precise, salient object can be well detected by the saliency maps after the foreground queries in the second stage. The main steps of the proposed salient object detection algorithm are summarized in Algorithm 1.

Algorithm 1 Bottom-up Saliency based on Manifold Ranking

Input: An image and required parameters

1: Segment the input image into superpixels, construct a graph G with superpixels as nodes, and compute its degree matrix $\mathbf{D}$ and weight matrix $\mathbf{W}$ by Eq. 4.

2: Compute $(\mathbf{D} - \alpha\mathbf{W})^{-1}$ and set its diagonal elements to 0.

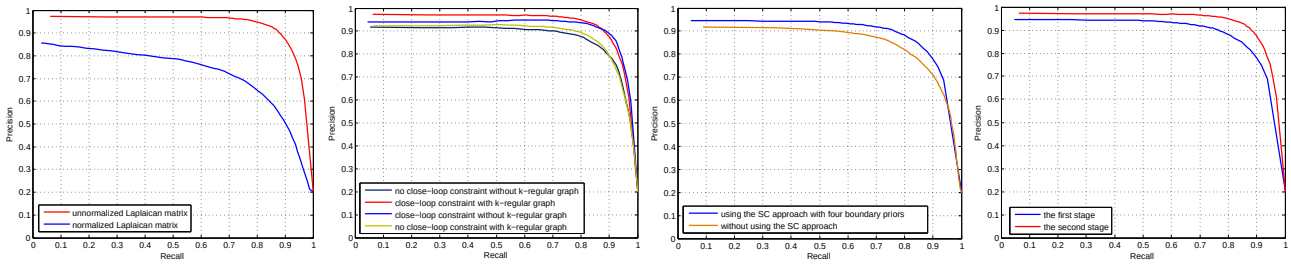
3: Form indicator vectors $\mathbf{y}$ with nodes on each side of image as queries, and compute their corresponding side-specific maps by Eq. 3 and Eq. 5. Then, compute the saliency map S_{bq} by Eq. 6.

4: Bi-segment S_{bq} to form salient foreground queries and an indicator vector $\mathbf{y}$. Compute the saliency map S_{fq} by Eq. 3 and Eq. 7.

Output: a saliency map S_{fq} representing the saliency value of each superpixel.

5. Experimental Results

We evaluate the proposed method on three datasets. The first one is the MSRA dataset [23] which contains 5,000 images with the ground truth of salient region marked by bounding boxes. The second one is the MSRA-1000 dataset, a subset of the MSRA dataset, which contains 1,000 images provided by [2] with accurate human-labelled masks for salient objects. The last one is the proposed DUT-OMRON dataset, which contains 5,172 carefully labeled images by five users. The source images, ground



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

Figure 8. Precision-recall curves on the MSRA-1000 dataset with different design options of the proposed algorithm. From left to right: ranking with normalized and unnormalized Laplacian matrices, graph construction, the SC approach, results generated by each stage.

truth labels and detailed description of this dataset can be found at http://ice.dlut.edu.cn/lu/DUT-OMRON/Homepage.htm. We compare our method with fourteen state-of-the-art saliency detection algorithms: the IT [17], GB [14], MZ [25], SR [15], AC [1], Gof [11], FT [2], LC [37], RC [9], SVO [7], SF [27], CB [18], GS_SP [34] and XIE [35] methods.

Experimental Setup: We set the number of superpixel nodes $N = 200$ in all the experiments. There are two parameters in the proposed algorithm: the edge weight σ in Eq. 4, and the balance weight α in Eq. 3. The parameter σ controls the strength of weight between a pair of nodes and the parameter α balances the smooth and fitting constraints in the regularization function of manifold ranking algorithm. These two parameters are empirically chosen, $\sigma^2 = 0.1$ and $\alpha = 0.99$, for all the experiments.

Evaluation Metrics: We evaluate all methods by precision, recall and F-measure. The precision value corresponds to the ratio of salient pixels correctly assigned to all the pixels of extracted regions, while the recall value is defined as the percentage of detected salient pixels in relation to the ground-truth number. Similar as prior works, the precision-recall curves are obtained by binarizing the saliency map using thresholds in the range of 0 and 255. The F-measure is the overall performance measurement computed by the weighted harmonic of precision and recall:

$$F_\beta = \frac{(1+\beta^2)Precision \times Recall}{\beta^2 Precision + Recall}, \qquad (8)$$

where we set $\beta^2 = 0.3$ to emphasize the precision [2].

5.1. MSRA-1000

We first examine the design options of the proposed algorithm in details. The ranking results using the normalized (Eq. 2) and unnormalized (Eq. 3) Laplacian matrices for ranking are analyzed. Figure 8 (a) shows that the ranking results with the unnormalized Laplacian matrix are better, and used in all the experiments. Next, we demonstrate the merits of the proposed graph construction scheme. We compute four precision-recall curves for four cases of node connection on the graph: close-loop constraint without extending the scope of node with k-regular graph, without close-loop constraint and with k-regular graph, without both close-loop constraint and k-regular graph and close-loop constraint with k-regular graph. Figure 8 (b) shows that the use of close-loop constraint and k-regular graph performs best. The effect of the SC approach in the first stage is also evaluated. Figure 8 (c) shows that our approach using the integration of saliency maps generated from different boundary priors performs better in the first stage. We further compare the performance for each stage of the proposed algorithm. Figure 8 (d) demonstrates that the second stage using the foreground queries further improve the performance of the first stage with background queries.

We evaluate the performance of the proposed method against fourteen state-of-the-art bottom-up saliency detection methods. Figure 9 shows the precision-recall curves of all methods. We note that the proposed methods outperforms the SVO [7], Gof [11], CB [18], and RC [9] which are top-performance methods for saliency detection in a recent benchmark study [4]. In addition, the proposed methods significantly outperforms the GS_SP [34] method which is also based on boundary priors. We also compute the precision, recall and F-measure with an adaptive threshold proposed in [2], defined as twice the mean saliency of the image. The rightmost plot of Figure 9 shows that the proposed algorithm achieves the highest precision and F-measure values. Overall, the results using three metrics demonstrate that the proposed algorithm outperforms the state-of-the-art methods. Figure 10 shows a few saliency maps of the evaluated methods. We note that the proposed algorithm uniformly highlights the salient regions and preserves finer object boundaries than the other methods.

5.2. MSRA

We further evaluate the proposed algorithm on the MSRA dataset in which the images are annotated with nine bounding boxes by different users. To compute precision and recall values, we first fit a rectangle to the binary saliency map and then use the output bounding box for

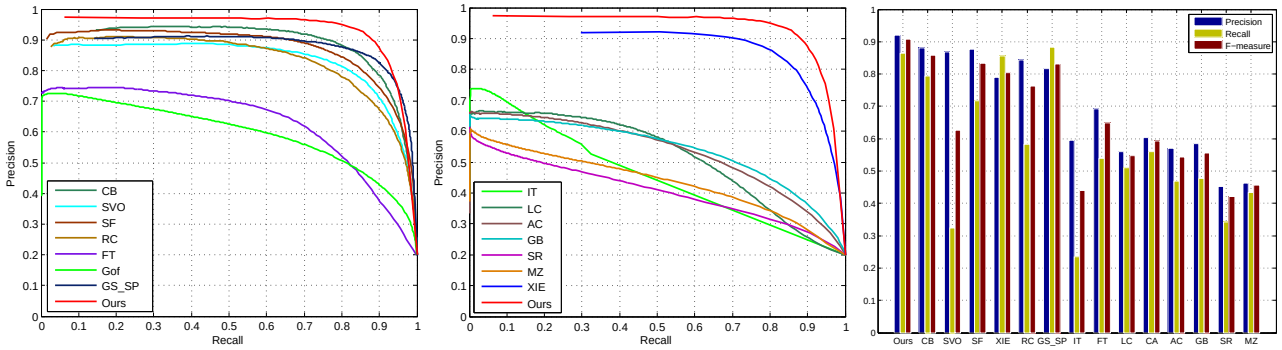


Figure 9. Left, middle: precision-recall curves of different methods. Right: precision, recall and F-measure using an adaptive threshold. All results are computed on the MSRA-1000 dataset. The proposed method performs well in all these metrics.

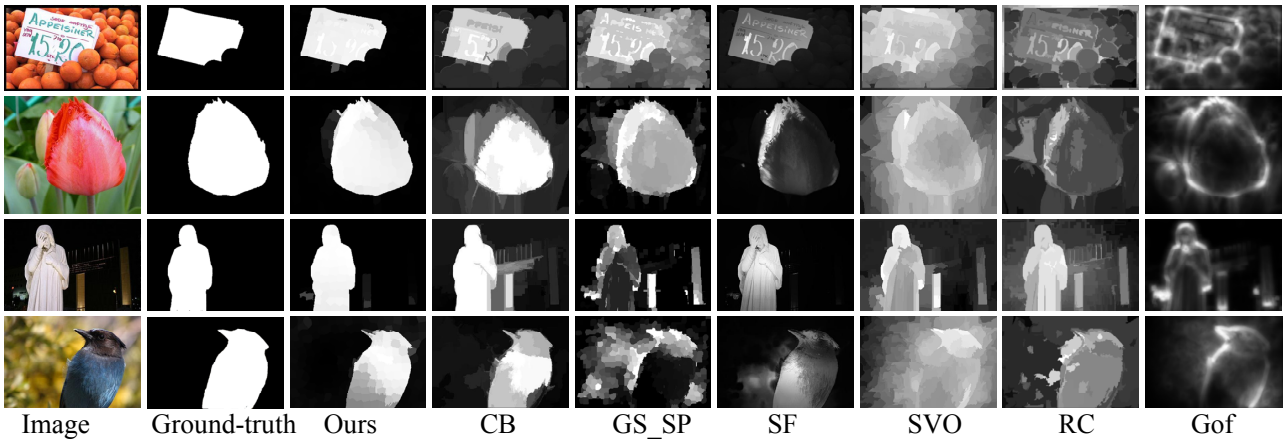


Figure 10. Saliency detection results of different methods. The proposed algorithm consistently generates saliency maps close to the ground truth.

Method	Ours	CB [18]	Gof [11]	SVO [7]
Time(s)	0.256	2.146	38.896	79.861

Table 1. Comparison of average run time (seconds per image).

the evaluation. Similar to the experiments on the MSRA-1000 database, we also binarize saliency maps using the threshold of twice the mean saliency to compute precision, recall and F-measure bars. Figure 11 shows the proposed model performs better than the other methods on this large dataset. We note that the Gof [11] and FT [2] methods have extremely large recall values, since their methods tend to select large attention regions, but at the expense of low precision.

5.3. DUT-OMRON

We test the proposed model on the DUT-OMRON dataset in which images are annotated with bounding boxes by five users. Similar to the experiments on the MSRA database, we also compute a rectangle of the binary saliency map and then evaluate our model by the fixed thresholding and the adaptive thresholding ways. Figure 12 shows that the proposed dataset is more challenging (all the models performs more poorly), and thus provides more room for improvement of the future work.

5.4. Run Time

The average run time of currently top-performance methods using matlab implementation on the MSRA-1000 database are presented in Table 1 based on a machine with Intel Dual Core i3-2120 3.3 GHz CPU and 2GB RAM. Our run time is much faster than that of the other saliency models. Specifically, the superpixel generation by SLIC algorithm [3] spends 0.165 s (about 64%), and the actual saliency computation spends 0.091 s. The MATLAB implementation of the proposed algorithm is available at http://ice.dlut.edu.cn/lu/publications.html, or http://faculty.ucmerced.edu/mhyang/pubs.html.

6. Conclusion

We propose a bottom-up method to detect salient regions in images through manifold ranking on a graph, which incorporates local grouping cues and boundary priors. We adopt a two-stage approach with the background and fore-

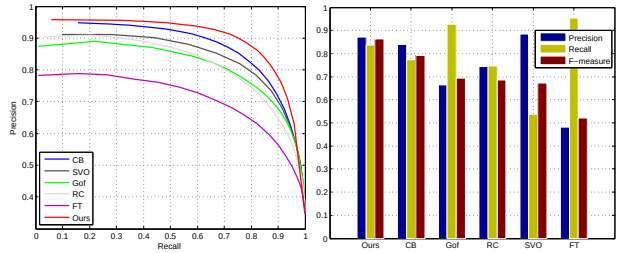


Figure 11. Left: precision-recall curves of different methods. Right: precision, recall and F-measure for adaptive threshold. All results are computed on the MSRA dataset.

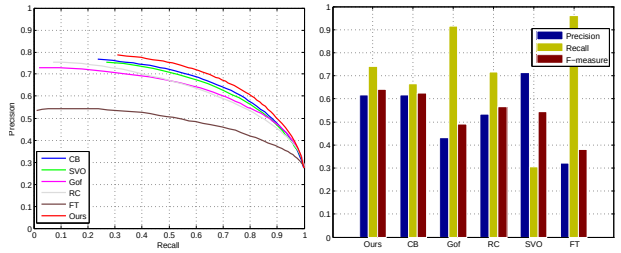


Figure 12. Left: precision-recall curves of different methods. Right: precision, recall and F-measure for adaptive threshold. All results are computed on the DUT-OMRON dataset.

ground queries for ranking to generate the saliency maps. We evaluate the proposed algorithm on large datasets and demonstrate promising results with comparisons to fourteen state-of-the-art methods. Furthermore, the proposed algorithm is computationally efficient. Our future work will focus on integration of multiple features with applications to other vision problems.

Acknowledgements

C. Yang and L. Zhang are supported by the Fundamental Research Funds for the Central Universities (DUT12JS05). H. Lu is supported by the Natural Science Foundation of China #61071209 and #61272372. M.-H. Yang is supported in part by the NSF CAREER Grant #1149783 and NSF IIS Grant #1152576.

References

[1] R. Achanta, F. Estrada, P. Wils, and S. Susstrunk. Salient region detection and segmentation. In *ICVS*, 2008. 1, 6

[2] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. In *CVPR*, 2009. 1, 4, 5, 6, 7

[3] R. Achanta, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels. Technical report, EPFL, Tech.Rep. 149300, 2010. 3

[4] A. Borji, D. Sihite, and L. Itti. Salient object detection: A benchmark. In *ECCV*, 2012. 4, 6

[5] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1):107–117, 1998. 2

[6] N. Bruce and J. Tsotsos. Saliency based on information maximization. In *NIPS*, 2005. 1

[7] K.-Y. Chang, T.-L. Liu, H.-T. Chen, and S.-H. Lai. Fusing generic objectness and visual saliency for salient object detection. In *ICCV*, 2011. 1, 6, 7

[8] T. Chen, M. Cheng, P. Tan, A. Shamir, and S. Hu. Sketch2photo: Internet image montage. *ACM Trans. on Graphics*, 2009. 1

[9] M. M. Cheng, G. X. Zhang, N. J. Mitra, X. Huang, and S. M. Hu. Global contrast based salient region detection. In *CVPR*, 2011. 1, 4, 6

[10] J. Feng, Y. Wei, L. Tao, C. Zhang, and J. Sun. Salient object detection by composition. In *ICCV*, 2011. 1

[11] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. In *CVPR*, 2010. 1, 6, 7

[12] V. Gopalakrishnan, Y. Hu, and D. Rajan. Random walks on graphs for salient object detection in images. *IEEE TIP*, 2010. 1, 2

[13] L. Grady, M. Jolly, and A. Seitz. Segmenation from a box. In *ICCV*, 2011. 2

[14] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *NIPS*, 2006. 1, 6

[15] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *CVPR*, 2007. 1, 6

[16] L. Itti. Automatic foveation for video compression using a neurobiological model of visual attention. *IEEE TIP*, 2004. 1

[17] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE PAMI*, 1998. 1, 4, 6

[18] H. Jiang, J. Wang, Z. Yuan, T. Liu, N. Zheng, and S. Li. Automatic salient object segmentation based on contex and shape prior. In *BMVC*, 2011. 6, 7

[19] T. Judd, K. Ehinger, F. Durand, and A. Torralba. Learning to predict where humans look. In *ICCV*, 2009. 1

[20] T. H. Kim, K. M. Lee, and S. U. Lee. Learning full pairwise affinities for spectral segmentation. In *CVPR*, 2010. 2

[21] D. Klein and S. Frintrop. Center-surround divergence of feature statistics for salient object detection. In *ICCV*, 2011. 1

[22] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. In *ICCV*, 2009. 2

[23] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H. Shum. Learning to detect a salient object. *IEEE PAMI*, 2011. 1, 5

[24] Y. Lu, W. Zhang, H. Lu, and X. Y. Xue. Salient object detection using concavity context. In *ICCV*, 2011. 1

[25] Y. Ma and H. Zhang. Contrast-based image attention analysis by using fuzzy growing. *ACM Multimedia*, 2003. 1, 6

[26] A. Ng, M. Jordan, Y. Weiss, et al. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856, 2002. 2

[27] F. Perazzi, P. Krahenbuhl, Y. Pritch, and A. Hornung. Saliency filters: Contrast based filtering for salient region detection. In *CVPR*, 2012. 1, 6

[28] U. Rutishauser, D. Walther, C. Koch, and P. Perona. Is bottom-up attention useful for object recognition? In *CVPR*, 2004. 1

[29] B. Scholkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 2001. 3

[30] J. Sun, H. C. Lu, and S. F. Li. Saliency detection based on integration of boundary and soft-segmentation. In *ICIP*, 2012. 1

[31] B. Tatler. The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions. *Journal of Vision*, 2007. 2

[32] L. Wang, J. Xue, N. Zheng, and G. Hua. Automatic salient object extraction with contextual cue. In *ICCV*, 2011. 1

[33] W. Wang, Y. Wang, Q. Huang, and W. Gao. Measuring visual saliency by site entropy rate. In *CVPR*, 2010. 1

[34] Y. C. Wei, F. Wen, W. J. Zhu, and J. Sun. Geodesic saliency using background priors. In *ECCV*, 2012. 2, 6

[35] Y. L. Xie, H. C. Lu, and M. H. Yang. Bayesian saliency via low and mid level cues. *IEEE TIP*, 2013. 1, 6

[36] J. Yang and M. Yang. Top-down visual saliency via joint crf and dictionary learning. In *CVPR*, 2012. 1

[37] Y. Zhai and M. Shah. Visual attention detection in video sequences using spatiotemporal cues. *ACM Multimedia*, 2006. 1, 6

[38] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Scholkopf. Learning with local and global consistency. In *NIPS*, 2003. 3

[39] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Scholkopf. Ranking on data manifolds. In *NIPS*, 2004. 2, 3