

GraB: Visual Saliency via Novel Graph Model and Background Priors

Qiaosong Wang¹, Wen Zheng², and Robinson Piramuthu²

¹University of Delaware ²eBay Research

Abstract

We propose an unsupervised bottom-up saliency detection approach by exploiting novel graph structure and background priors. The input image is represented as an undirected graph with superpixels as nodes. Feature vectors are extracted from each node to cover regional color, contrast and texture information. A novel graph model is proposed to effectively capture local and global saliency cues. To obtain more accurate saliency estimations, we optimize the saliency map by using a robust background measure. Comprehensive evaluations on benchmark datasets indicate that our algorithm universally surpasses state-of-the-art unsupervised solutions and performs favorably against supervised approaches.

1. Introduction

Humans are able to rapidly identify the visually distinctive objects in a scene. This fundamental capability has long been studied in neuroscience and cognitive psychology. In the computer vision community, researchers focus on similar tasks to determine regions that attract attention from a human perception system. The selected regions contain finer details of interest and can be used for extraction of intermediate and higher level information. Therefore, a fast and robust saliency detection algorithm can benefit various other vision tasks.

The literature of saliency map estimation is vast. However, most existing approaches can be categorized into unsupervised (typically bottom-up) [10, 28, 32, 36, 19] and supervised (typically bottom-up, but more recent approaches are a combination of top-down and bottom-up) [17, 24, 15, 20] approaches.

While supervised approaches are able to automatically integrate multiple features and in general achieve better performance than unsupervised methods, it is still expensive to perform the training process, especially data collection. Also, compared to traditional special-purpose object detectors (e.g. pedestrian detection) where objects un-

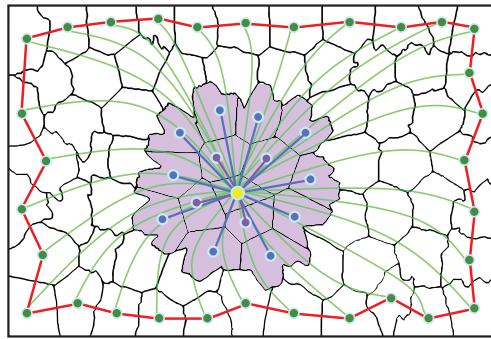


Figure 1: Our novel graph structure with superpixels as nodes. The purple and blue lines represent connections to first and second order neighbors, respectively. The green lines indicate that each node is connected to the boundary nodes on four sides of the image. The red lines show that the all boundary nodes are connected among themselves. See Sec. 3.1 for details.

der the same class share some consistency, the salient objects from two images are often found vastly different in terms of visual appearance, especially when the object can be anything. Furthermore, the process of generating pixel-wise ground truth annotations itself is expensive and labor-intensive, and sometimes may even be impossible considering the scale of today’s massive long-tailed visual repositories. This is typically the case in large e-commerce scenarios. A fast saliency technique can be an essential pre-processing step for background removal or object/product detection and recognition in large ecommerce applications.

In this paper, we propose an unsupervised bottom-up saliency estimation approach. Our method is based on the remarkable success of the spectral graph theory. We focus on the core elements of spectral clustering algorithms. Specifically, we introduce a new graph model which captures local/global contrast and effectively utilizes the boundary prior. Inspired by ISOMAP manifold learning [31], we introduce geodesic distance to calculate the weight matrix. This constraint maximally enforces the background connectivity prior. Furthermore, we exploit

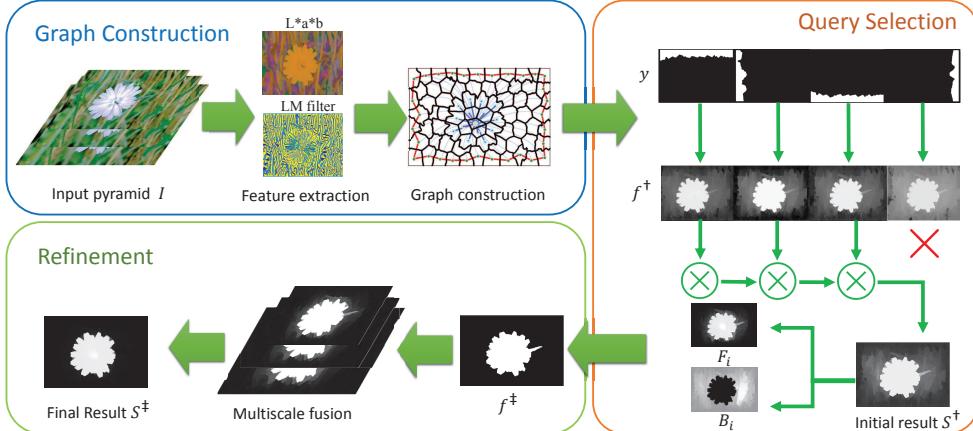


Figure 2: Pipeline of the proposed algorithm, divided into three parts: Graph Construction (Sec. 3), Query Selection (Sec. 4.1) and Refinement (Sec. 4.2). Nodes in the graph are superpixels. Weights are based on color and texture features. Groups of background seeds are selected for initial saliency based on their influence via the graph structure. Inconsistent groups of background seeds are eliminated. This estimated saliency map is refined by passing it again through the system. This process is repeated at multiple scales and results are fused.

boundary prior for selecting seeds to perform an initial background query. The resulting saliency map is further used to generate seeds to perform another query to obtain the final saliency map. As we will demonstrate empirically, the proposed method universally outperforms state-of-the-art unsupervised methods (e.g. GMR [36]) by a large margin, and in some cases even excels supervised methods (e.g. DRFI [14]). Our claim is that the proposed graph model provides more desirable characteristics for saliency detection and achieves unprecedented balance between computational complexity and accuracy.

2. Related Works

The core of our work is closely related to graph-based manifold ranking as in [36], geodesic distance as in [32], boundary prior sampling as in [19] and multi-scale fusion as in [35].

Supervised vs. Unsupervised Unsupervised methods [10, 28, 32, 36, 19] aim at separating salient objects by extracting cues from the input image only. To date, various low-level features have been shown to be effective for saliency detection, such as color contrast, edge density [28], backgroundness [32, 36], objectness [6, 15], focus [15], etc. By eliminating the requirement of training, unsupervised methods can be easily integrated into various applications. In contrast, supervised approaches [17, 24, 15, 20] acquire visual knowledge from ground truth annotations. Recent advances in deep learning show promising results on benchmark datasets [20]. However, it is expensive to collect the hand-labeled images and set up the learning framework.

Graph-based Models Graph-based approaches have gained great popularity due to the simplicity and efficiency of graph algorithms. Harel et al. [10] proposed the graph based visual saliency (GBVS), a graph-based saliency model with multiple features to extract saliency information. Chang et al. [6] present a computational framework by constructing a graphical model to fuse objectness and regional saliency. Yang et al. [36] rank the similarity of superpixels with foreground or background seeds via graph-based manifold ranking. This method is further improved by Li et al. to generate pixel-wise saliency maps via regularized random walks ranking [19].

Center vs. Background Prior Recently, more and more bottom-up methods prefer to use the image boundary as the background seeds. This *boundary prior* is more general than previously used *center prior*, which assumes that the saliency object tend to appear near the image center [17, 24]. Wei et al. [32] define the saliency of a region to be the length of its shortest path to the virtual background node. In [39], a robust background measure is proposed to characterize the spatial layout of an image region with respect to the boundary regions.

3. Graph Construction

Our approach is based on building an undirected weighted graph for superpixels. We first segment the input image I into n superpixels $\mathcal{S} = \{s_1, s_2, \dots, s_n\}$ via the Simple Linear Iterative Clustering (SLIC) [2] algorithm. For each superpixel s , we extract color and texture information to form a regional feature descriptor r . A metric is proposed to calculate the edge weight between two given

descriptors. Next, we construct a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ (see Fig. 1) where \mathcal{V} is a set of nodes corresponding to superpixels \mathcal{S} , and edges \mathcal{E} are constructed using the proposed graph model. \mathcal{E} is quantified by a weight matrix $W = [w_{ij}]_{n \times n}$ where the weights are calculated using distances between extracted feature descriptors. In Sec. 3.1, we describe our newly proposed graph model and in Sec. 3.2 we show how to extract regional features and calculate the weight matrix W .

3.1. Proposed Graph Model

Given a set of superpixels \mathcal{S} , we start by building a k -regular graph where each node is only connected to its immediate neighbors. We define the adjacency matrix of the initial graph \mathcal{G} to be $A = [a_{ij}]_{n \times n}$. If $a_{ij} = 1$, then the nodes s_i and s_j are adjacent, otherwise $a_{ij} = 0$. As \mathcal{G} is undirected we require $a_{ij} = a_{ji}$. $\mathcal{B} \in \mathcal{S}$ denotes a set of boundary nodes containing $|\mathcal{B}|$ superpixels on the four borders of the input image. For robust purposes, we only choose to use three borders, and the selection of borders is described in Sec. 4.1. We subsequently add edges to the initial graph \mathcal{G} to build a new graph model with the following rules: 1) Each node is connected to both its immediate neighbors and 2-hop neighbors; 2) We add edges to connect each node to boundary nodes on the four sides of the image. The weight for each edge is divided by the number of boundary nodes; 3) Any pair of nodes on the image boundary is considered to be connected. We denote the above three rules by R_1 , R_2 and R_3 , and the final edge set $\mathcal{E} = \{\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3\}$ can be obtained as:

$$\begin{aligned} R_1 : \mathcal{E}_1 &= \{(s_i, s_j) | s_i, s_j \in \mathcal{S}, a_{ij} = 1\} \\ &\cup \{(s_i, s_k) | s_k \in \mathcal{S}, a_{kj} = 1\}, \\ w_{ij} &= \text{weight}(r_i, r_j). \\ R_2 : \mathcal{E}_2 &= \{(s_i, s_j) | s_i \in \mathcal{S}, s_j \in \mathcal{B}\}, \\ w_{ij} &= \text{weight}(r_i, r_j)/|\mathcal{B}|. \\ R_3 : \mathcal{E}_3 &= \{(s_i, s_j) | s_i, s_j \in \mathcal{B}\}, \\ w_{ij} &= \text{weight}(r_i, r_j). \end{aligned} \quad (1)$$

The structure of our graph model is shown in Fig. 1. Since neighboring superpixels are more likely to be visually similar, R_1 enables us to effectively utilize local neighborhood relationships between the superpixel nodes. R_2 connects each node to all boundary nodes, enforcing the global contrast constraint. Since the number of boundary superpixels may be large, we average the edge weights, making the total contribution of boundary nodes equivalent to only one single superpixel. R_3 enforces the graph to be a closed-loop. Combined with R_2 which connects each superpixel to boundary nodes. R_3 further reduces the geodesic distance of two similar superpixels.



Figure 3: The effect of our graph model described in Sec. 3.1. From left to right: input image, result using the graph structure proposed by [36], result obtained using our graph model. Our model performs better since it encodes background consistency, global contrast and local contrast.



Figure 4: Examples where geodesic distance generate more accurate results. From left to right: input image, results without enforcing the geodesic distance constraints, results with geodesic constraints. Geodesic distance avoids missing parts due to color bleeding.

3.2. Feature Extraction

In this section, we detail the process of extracting feature descriptors from each superpixel. This process is crucial to the estimation of the final saliency map as the edge weights are calculated by comparing the feature descriptors of two nodes. A good feature descriptor should exhibit high contrast between salient and non-salient regions. In our work, we mainly adopt two kinds of features: color and texture. For color features, we consider mean color values and color histograms in the CIELAB [12] color space for each superpixel. For texture features, we use responses from the Leung-Malik (LM) filter bank [18]. Let v^{lab} , h^{lab} , h^{tex} be the mean L*a*b* color, L*a*b* histogram and max LM response histogram of superpixel s , we define the distance between two superpixels as:

$$\text{dist}(r_i, r_j) = \lambda_1 ||v_i^{lab} - v_j^{lab}|| + \lambda_2 \chi^2(h_i^{lab}, h_j^{lab}) + \lambda_3 \chi^2(h_i^{tex}, h_j^{tex}). \quad (2)$$

where $r = (v, h^{lab}, h^{tex})$ is the combined feature descriptor for superpixel s , λ_1, λ_2 and λ_3 are weighting parameters, $\chi^2(h_1, h_2) = \sum_{i=1}^K \frac{2(h_1(i) - h_2(i))^2}{h_1(i) + h_2(i)}$ is the chi-squared distance between histograms h_1 and h_2 with K being the number of bins. The edge weights can be obtained by the

Algorithm 1: Visual Saliency via Novel Graph Model and Background Priors

Data: Input image I and related parameters

1. Apply SLIC [2] and separate input image I into n superpixels $\mathcal{S} = \{s_1, s_2, \dots, s_n\}$, establish graph structure with Eq. (1).
2. Calculate W and D using Eq. (2) and Eq. (3).
3. Select three borders as query seeds as described in Sec. 4.1 and obtain query vector $y = [y_1, y_2, \dots, y_n]^T$.
4. Acquire initial saliency estimation S^\dagger using Eq. (7), Eq. (8) and Eq. (9).
5. Optimize S^\dagger using Eq. (10) and re-apply Eq. (7) to obtain the foreground estimation. Apply Eq. (4) and average results across different levels to obtain final saliency map S^\ddagger .

Result: A saliency map S^\ddagger with the same size as the input image

Gaussian similarity function:

$$weight(r_i, r_j) = \begin{cases} exp(-dist(r_i, r_j)/\sigma^2) & \text{if } a_{ij} = 1, \\ \min_{\rho_1=r_i, \rho_2=r_{i+1}, \dots, \rho_m=r_j} & \\ \sum_{\rho=1}^{m-1} weight(\rho_k, \rho_{k+1}) & \text{if } a_{ij} = 0. \end{cases} \quad (3)$$

where σ is a constant. In the above equation, the second condition considers the shortest path between nodes i, j . As can be seen from Eq.(2), our approach is completely based on intrinsic cues of the input image. Without any prior knowledge of size of the salient object, we adopt the L -layer Gaussian pyramid for robustness. The l -th-level pyramid I^l is obtained as:

$$I^l(x, y) = \sum_{s=-2}^2 \sum_{t=-2}^2 \omega(s, t) I^{l-1}(2x + s, 2y + t), l \geq 1. \quad (4)$$

where I^0 is the original image, $\omega(s, t)$ is a Gaussian weighting function (identical at all levels). The number of superpixels $n^l = |\mathcal{S}^l|$ for the l -th level pyramid I^l is set as:

$$n^l = \frac{n^{l-1}}{2^{2(l-1)}} \quad (5)$$

Next, we extract multiscale features r^l and build weight matrices W^l for each level. The final saliency estimation is conducted on each level independently and the output saliency map is combined using results from all levels (see Sec. 5.2 for details).

4. Background Priors

Given the weighted graph, we can take either foreground or background nodes as queries [36]. The resulting saliency map is calculated based on its relevance to the queries. Our algorithm is based on background priors, which consists of

two parts: the *boundary prior* and the *connectivity prior*. The first prior is based on the observation that the salient object seldom touches the image borders. Compared to the *center prior* [17, 24] which assumes that the salient object always stays at the center of an image, the *boundary prior* is more robust, which is validated on several public datasets [32]. In our work, we choose three out of four borders as background seeds to perform queries [19]. This is because the foreground object may completely occupy one border of an image, which is commonly seen in portrait photos. Therefore, eliminating one border which tends to have a very distinct appearance generates more accurate results. The second prior is based on the insight that background regions are usually large and homogeneous. Therefore, the superpixels in the background can be easily connected to each other. This prior is also applicable for images with a shallow depth of field, where the background region is out of focus. The rest of this section is organized as follows: Sec. 4.1 elaborates the detailed steps of the initial background query and Sec. 4.2 illustrates a refinement scheme based on the *connectivity prior*.

4.1. Query via the Boundary Prior

To provide more accurate saliency estimations, we first compare the four borders of the image and remove one with the most distinctive color distribution. We combine boundary superpixels together to form a single region, and use Eq. (2) to compute the distance of any two of the four regions $\{\mathcal{B}_{top}, \mathcal{B}_{bottom}, \mathcal{B}_{left}, \mathcal{B}_{right}\}$. The resulting 4×4 matrix is summed column-wise, and the maximum column corresponds to the boundary to be removed.

Once the query boundaries are obtained, we can label the corresponding superpixels to be background. More formally, we build a query vector $y = [y_1, y_2, \dots, y_n]^T$, where $y_i = 1$ if s_i belongs one of the four query boundaries, otherwise $y_i = 0$. Given the weight matrix $W = [w_{ij}]_{n \times n}$ computed in Sec. 3.2, we can obtain the degree matrix $D = diag(d_1, d_2, \dots, d_n)$, where $d_i = \sum_j w_{ij}$. Let f be the ranking function assigning rank values $f = [f_1, f_2, \dots, f_n]^T$ which could be obtained by solving the following minimization problem:

$$f^\dagger = \arg \min_f \frac{1}{2} \left(\sum_{i,j=1}^n w_{ij} \left\| \frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right\|^2 + \mu \sum_{i=1}^n \|f_i - y_i\|^2 \right). \quad (6)$$

where μ is a controlling parameter. The optimized solution is given in [38] as:

$$f^\dagger = (D - \frac{W}{\mu + 1})^{-1} y. \quad (7)$$

Three ranking results $f^\dagger(b)$ will be achieved after applying Eq. (7), where b corresponds one of the three borders. Since the ranking results show the background relevance of each

Table 1: Ablation study on adding different components to the baseline GMR [36] algorithm (Sec. 5.2). All results correspond to ECSSD. GF = guided filter [11], RTV = texture smoothing using relative total variation [34], EBR = erroneous boundary removal [19], RPCA = robust PCA [5], LAB = CIELAB color [12], HIST = L*a*b* histogram, LM = Leung-Malik filter bank [18], LBP = local binary patterns [26], AVE = simple averaging, HS = hierarchical saliency [35], GMR = graph based manifold ranking [36], BC = boundary connection, GEO = geodesic distance. Methods included in the final pipeline are marked in bold.

Evaluation	Preprocessing		Sampling		Features				Scaling		Graph		
	GF	RTV	EBR	RPCA	LAB	HIST	LM	LBP	AVE	HS	GMR	BC	GEO
Precision	0.712	0.716	0.725	0.755	0.731	0.725	0.718	0.614	0.727	0.734	0.731	0.771	0.743
Recall	0.729	0.712	0.723	0.646	0.575	0.631	0.682	0.577	0.710	0.716	0.569	0.626	0.618
F-Measure	0.713	0.716	0.725	0.745	0.715	0.716	0.715	0.610	0.725	0.733	0.714	0.756	0.730
Runtime (s)	0.136	2.237	0.011	4.782	0.025	0.031	0.094	0.047	0.002	0.129	0.258	0.327	0.538

node, we still need to calculate their complement values to obtain the foreground-based saliency:

$$S_i(b) = 1 - f_i^\dagger(b), i = 1, 2, \dots, n. \quad (8)$$

The results are then put into element-wise multiplication to calculate the saliency map:

$$S^\dagger = \prod_b S_i(b). \quad (9)$$

4.2. Refinement

In this section, we seek to optimize the result from the previous section. The optimized result will be used as foreground query by applying Eq. (7) again. The cost function is designed to assign 1 to salient region value and 0 to background region. The optimized result is then obtained by minimizing the following cost function [39]:

$$f^\ddagger = \arg \min_f \left(\sum_{i=1}^n F_i(f_i - 1)^2 + \sum_{i=1}^n B_i f_i^2 + \sum_{i,j} w_{ij} (f_i - f_j)^2 \right). \quad (10)$$

Where F_i and B_i are foreground and background probabilities, $F_i > \text{mean}(S_i)$ and $B_i < \text{mean}(S_i)$. The three terms are all squared errors and the optimal result is computed by least-square. The newly obtained f is a binary indicator vector and can be used as seed for foreground queries. By re-applying Eq. (7), we obtain the final saliency map $S^\ddagger = (D - \frac{W}{\mu + 1})^{-1} f^\ddagger$.

5. Experiments

5.1. Parameter Setup

We empirically set parameters in all experiments. λ_1 , λ_2 and λ_3 in Eq. (2) are set to 0.25, 0.45 and 0.3, respectively. In our experiment, we use a 3 level pyramid, hence $l = 3$ in Eq. (4). The constant σ in Eq. (3) and μ in Eq. (6) are empirically chosen and $\sigma^2 = 0.1$, $1/(\mu + 1) = 0.99$. Our method is implemented using Matlab on a machine with Intel Core i7-980X 3.3 GHz CPU and 16GB RAM.

5.2. Ablation Studies

We start by modifying the GMR framework proposed by [36]. We experiment different design options among five categories: preprocessing, sampling, features, scaling and graph structure. The individual components are added to the original GMR framework and quantitative evaluations are conducted on the entire ECSSD dataset (Fig. 5 and Fig. 8).

Preprocessing The input images are often composed of objects at various scales with diverse texture details. Therefore, it is important to remove detrimental or unwanted content. We choose two edge-preserving filters for testing: guided image filtering [11] and imaging smoothing via relative total variation [34]. The first method performs edge-preserving smoothing while the second method extracts important structure from texture based on inherent variation and relative total variation measures. Quantitative evaluations suggest that both methods are able to improve the saliency detection results with similar performance.

Sampling Our method estimates saliency by using boundary superpixels as queries. If the foreground object touches one or more boundaries of the image, then the query results may be problematic. Therefore, it is important to smartly choose boundary superpixels as seeds. We tested two schemes for sampling boundary superpixels: erroneous boundary removal and robust principle component analysis. The details of the first method is illustrated in Sec. 4.1. The second method is based on the recently proposed rank minimization model [5]. We randomly sample 25% of all superpixels on each border, and repeat this step n times. This results in $4n$ set of query seeds. For each set we apply Eq. (7) to estimate saliency values for all superpixels. We unroll each resulting image into a vector and stack them into a matrix P . The low rank matrix A can be recovered from the corrupted data matrix $P = A + E$ by solving the following convex optimization problem:

$$\min_{A,E} \|A\|_* + \lambda \|E\|_1. \quad (11)$$

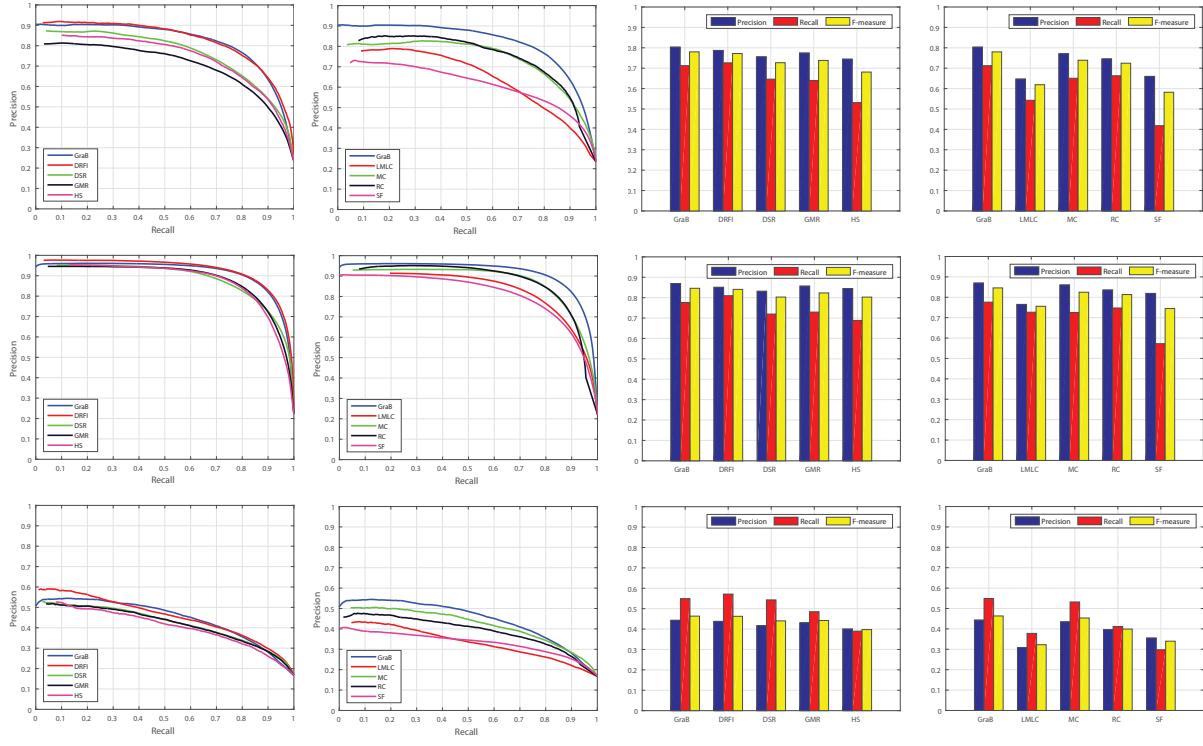


Figure 5: Quantitative PR-curve and F-measure evaluation of 9 approaches on 3 datasets. The rows from top to bottom correspond to ECSSD, THUS10K and JuddDB, respectively. Clearly, our approach excels all other unsupervised approaches and performs favorably against a powerful supervised approach (DRFI).

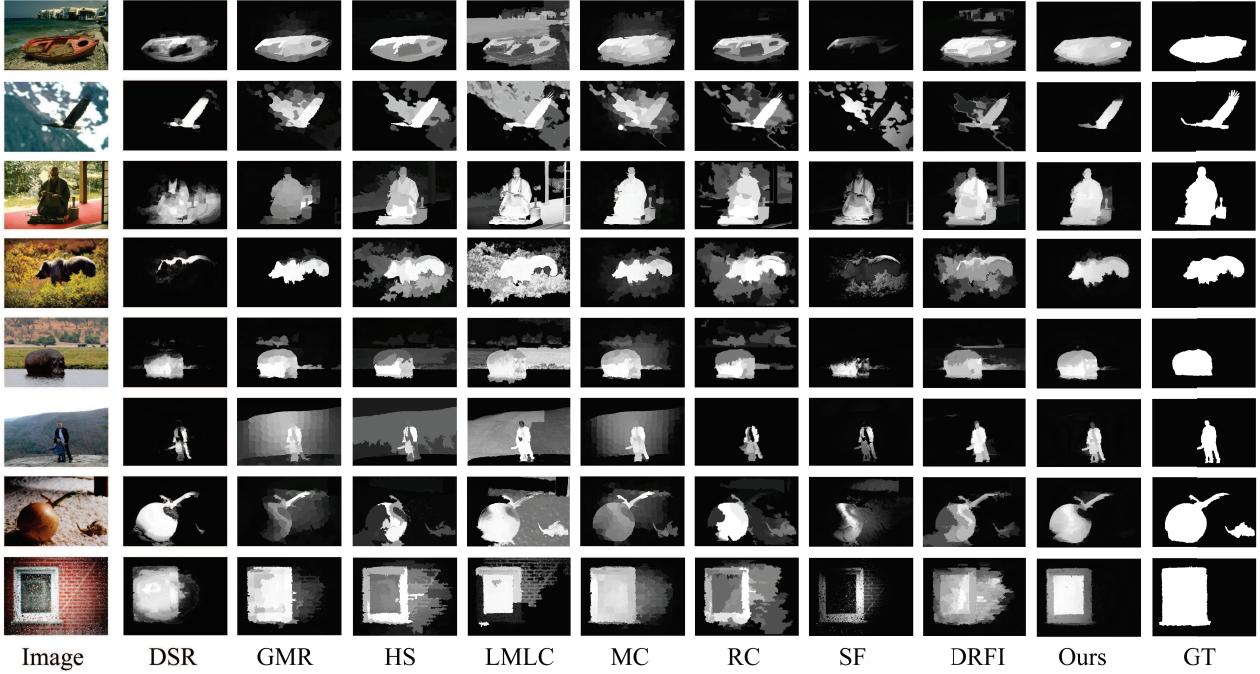


Figure 6: Qualitative evaluation. DRFI is one of the best supervised approaches. All other approaches shown here are unsupervised. Our model is closely related to GMR, but gives much better performance. See Sec. 5.3 for details.

where $\|\cdot\|_*$ denotes the nuclear norm, $\|\cdot\|_1$ denotes the sum of absolute values of matrix entries, λ is a positive weighting parameter and E is a sparse error matrix. In our experiment, we set $n = 5$ and perform the query 20 times for each image to get the initial saliency map S^\dagger . Evaluation on the complete ECSSD dataset shows that RPCA achieves better precision than erroneous boundary removal.

Features As stated in Sec. 3.2, we associate each superpixel with a feature vector to calculate the weight matrix W . A good feature descriptor should exhibit high contrast between salient and non-salient regions. In our experiment, we mainly test four different features: mean L*a*b* value [12], L*a*b* histogram, responses from the LM filter bank and local binary patterns (LBP) [26]. Among these features, the mean L*a*b* value is shown to be effective in [32, 36, 39]. According to Jiang et al. [14], the L*a*b* histogram is the most important regional feature in their feature integration framework. We are able to achieve satisfactory precision using the first two features. The LM filter response gives better overall recall. LBP feature seems to be not as effective as LM texture features in our case. Therefore, we linearly combine the first three features together to form the final feature vector.

Scaling In the saliency detection literature, hierarchical models are often adopted for robustness purpose [35, 14]. Our first experiment is to build an image pyramid, apply our algorithm to each layer and simply average all maps (Sec. 3.1). We subsequently test the approach proposed in [35]. This method differs from naive multi-layer fusion by selecting optimal weights for each region using hierarchical inference. Due to the proposed tree structure, the saliency inference can efficiently be conducted using belief propagation.

Graph Structure We use the model proposed by [36] as a baseline to test variations on the graph structure. The reference model enforces rule R_1 and R_3 in Sec. 1 and adopts Euclidean distance as the weighting metric. We conduct experiments on both graph structures (Sec. 3.1) and distance metrics (Sec. 3.2). Quantitative evaluations show a major performance improvement compared to other methodologies.

Combination We have presented 5 different strategies to facilitate more accurate saliency estimation. However, it is difficult to test all permutations and analyze the interactions between different methods. Therefore, how to optimally combine these methods still remains non-trivial. For example, the use of guided filter and multiscale averaging alone improves the recall scores. However, when combined together the performance drops slightly. Also, we choose not to use RPCA-based boundary sampling and belief-propagation based multi-layer fusion due to speed-accuracy tradeoffs. In our final model we choose not to perform any texture smoothing and employ the multiscale

averaging scheme due to its simplicity and efficacy. The color histogram based erroneous boundary removal scheme is used for generating the initial queries. The methods we choose to include in the final pipeline are marked in bold in Table 1. At the core of our algorithm is the newly proposed graph model and geodesic distance metric as they offer significant performance improvements.

5.3. Comparison with State-of-the-Art

Datasets In the experiments, we qualitatively and quantitatively compare the proposed approach with eight state-of-the-art approaches, including DRFI [14], DSR [22], GMR [36], HS[35], LMLC [33], MC [13], RC [7], SF [27]. It is important to note that besides DRFI, all other methods are unsupervised. The evaluation is conducted on three challenging datasets: ECSSD, THUS10K and JuddDB. The Extended Complex Scene Saliency Dataset (ECSSD) [35] contains 1000 semantically meaningful but structurally complex images from the BSD dataset [3], PASCAL VOC [9] and the Internet. The binary masks for the salient objects are produced by 5 subjects. THUS10K [8] contains 10000 images with pixel-level ground-truth labelings from the large dataset (20,000+ images) proposed by Liu et al. [24]. The JuddDB dataset [4] is created from the MIT saliency benchmark [16], mainly for checking generality of salient object detection models over real-world scenes with multiple objects and complex background. Additionally, we compared with all saliency object segmentation methods mentioned in [23] and [37] on the PASCAL-S dataset, including CPMC+GBVS [23], CPMC+PatchCut [37], GBVS+PatchCut [37], RC [7], SF [27], PCAS [25] and FT [1]. The PASCAL-S is proposed to avoid the dataset design bias, where the image selection process deliberately emphasizes the concept of saliency [23].

Evaluation We follow the canonical precision-recall curve and F-measure methodologies to evaluate the performance of our algorithm using the toolbox provided by [21]. The PR-curve and F-measure comparisons are shown in Fig. 5. Specifically, the PR curve is obtained by binarizing the saliency map using varying thresholds from 0 to 255, as mentioned in [1, 7, 27, 29]. F-measure is obtained using the metric proposed by [1]:

$$F_\beta = \frac{(1 + \beta^2)Precision \times Recall}{\beta^2 Precision + Recall} \quad (12)$$

Here, the precision and recall rates binarized using an adaptive threshold determined as two times the mean saliency of a given image. We set β to 0.3 to emphasize the precision[1, 36, 19].

As can be seen in Fig. 5, our method significantly outperforms all seven unsupervised methods by a large margin. Specifically, our method achieved an improvement of 6% in comparison with the baseline GMR model on the challenging ECSSD dataset. Also, our method is highly competitive

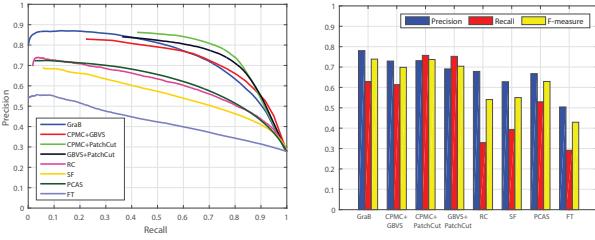


Figure 7: Quantitative PR-curve and F-measure evaluation of 7 methods on the PASCAL-S dataset. Note that our method achieves similar or better F-measure as more compute expensive methods.

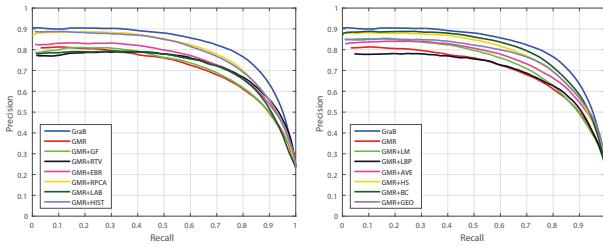


Figure 8: Quantitative PR-curve on different design options mentioned in Sec. 5.2. The baseline method (GMR) and final combined method (GraB) are added to both figures for comparison.

when compared to DRFI on all three datasets. It is worth noting that DRFI takes around 24 hours for training and 10 seconds for testing given a typical 400×300 image [14], whereas our method is fully unsupervised and only takes 800 milliseconds to process a similar image. Furthermore, DRFI takes 2500 images for training and extracts more than 20 different features, while our method is purely based on the input image and only uses 3 simple features. In other words, our method is much more efficient than DRFI yet still capable of maintaining competitive accuracy. The efficacy of our graph model is self-evident.

Quantitative evaluations on PASCAL-S [23] (Fig. 7) show that our method achieves higher precision, recall and F-measure scores compared to the state-of-the-art CPMC+GBVS algorithm presented in [23]. Also, our method performs favorably against the more recent PatchCut method [37] and clearly above all other saliency algorithms. Again, our method is training-free and performs much faster than CPMC+GBVS and PatchCut. (CPMC+GBVS takes around 30s to process a 400×300 image, according to our experiment; PatchCut takes around 10s for segmenting a 200×200 image, as reported by [37]. Both methods require extra training/example data).

Our evaluation does not include some of the latest deep-learning methods. The crux of this paper is to propose a novel heuristic model which is able to achieve

similar performance to supervised methods like DRFI or CPMC+GBVS without preparing expensive training data. This provides simplicity and easy-to-use generality in many practical applications where computing power is limited and ground truth annotations are very expensive or impossible to acquire.

Fig. 6 shows a few saliency maps for qualitative evaluation. We note that the proposed algorithm uniformly highlights the salient regions and preserves fine object boundaries than other methods.

6. Conclusion

We present a novel unsupervised saliency estimation method based on a novel graph model and background priors. Our graph model incorporates local and global contrast and naturally enforces the background connectivity constraint. The proposed feature distance metrics effectively and efficiently combines local color and texture cues to represent the intrinsic manifold structure. We further optimize the background seeds by exploiting a boundary query and refinement scheme, achieving state-of-the-art results. Our future work includes theoretical analysis on the proposed graph model and its potential towards building better clustering algorithms. Also, we would like to accelerate our algorithm via parallel computing, as large-scale spectral clustering has been trivially accomplished in high-performance graphics hardware [30].

7. Acknowledgement

We thank Jimei Yang and Professor Ming-Hsuan Yang from UC Merced for sharing the PatchCut data.

References

- [1] R. Achanta, S. Hemami, F. Estrada, and S. Sussstrunk. Frequency-tuned salient region detection. In *CVPR*, pages 1597–1604. IEEE, 2009.
- [2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Sussstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *PAMI*, 34(11):2274–2282, 2012.
- [3] P. Arbelaez, C. Fowlkes, and D. Martin. The berkeley segmentation dataset and benchmark. *see <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds>*, 2007.
- [4] A. Borji. What is a salient object? a dataset and a baseline model for salient object detection. *TIP*, 24(2):742–756, 2015.
- [5] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *JACM*, 58(3):11, 2011.
- [6] 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*, pages 914–921. IEEE, 2011.
- [7] M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S. Hu. Global contrast based salient region detection. *PAMI*, 37(3):569–582, 2015.

- [8] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu. Salient object detection and segmentation. *Image, 2(3):9*, 2011.
- [9] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV, 88(2):303–338*, 2010.
- [10] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *NIPS*, pages 545–552, 2006.
- [11] K. He, J. Sun, and X. Tang. Guided image filtering. *PAMI, 35(6):1397–1409*, 2013.
- [12] B. Hill, T. Roger, and F. W. Vorhagen. Comparative analysis of the quantization of color spaces on the basis of the cielab color-difference formula. *ACM TOG, 16(2):109–154*, 1997.
- [13] B. Jiang, L. Zhang, H. Lu, C. Yang, and M.-H. Yang. Saliency detection via absorbing markov chain. In *ICCV*, pages 1665–1672. IEEE, 2013.
- [14] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li. Salient object detection: A discriminative regional feature integration approach. In *CVPR*, 2013.
- [15] P. Jiang, H. Ling, J. Yu, and J. Peng. Salient region detection by ufo: Uniqueness, focusness and objectness. In *ICCV*, pages 1976–1983. IEEE, 2013.
- [16] T. Judd, F. Durand, and A. Torralba. A benchmark of computational models of saliency to predict human fixations. In *MIT Technical Report*, 2012.
- [17] T. Judd, K. Ehinger, F. Durand, and A. Torralba. Learning to predict where humans look. In *ICCV*, pages 2106–2113. IEEE, 2009.
- [18] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textures. *IJCV, 43(1):29–44*, 2001.
- [19] C. Li, Y. Yuan, W. Cai, Y. Xia, and D. D. Feng. Robust saliency detection via regularized random walks ranking. In *CVPR*. IEEE, 2015.
- [20] G. Li and Y. Yu. Visual saliency based on multiscale deep features. 2015.
- [21] X. Li, Y. Li, C. Shen, A. Dick, and A. Hengel. Contextual hypergraph modeling for salient object detection. In *ICCV*, pages 3328–3335, 2013.
- [22] X. Li, H. Lu, L. Zhang, X. Ruan, and M.-H. Yang. Saliency detection via dense and sparse reconstruction. In *ICCV*, pages 2976–2983. IEEE, 2013.
- [23] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille. The secrets of salient object segmentation. In *CVPR*, pages 280–287. IEEE, 2014.
- [24] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. *PAMI, 33(2):353–367*, 2011.
- [25] R. Margolin, A. Tal, and L. Zelnik-Manor. What makes a patch distinct? In *CVPR*, pages 1139–1146. IEEE, 2013.
- [26] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition, 29(1):51–59*, 1996.
- [27] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung. Saliency filters: Contrast based filtering for salient region detection. In *CVPR*, pages 733–740. IEEE, 2012.
- [28] P. L. Rosin. A simple method for detecting salient regions. *Pattern Recognition, 42(11):2363–2371*, 2009.
- [29] X. Shen and Y. Wu. A unified approach to salient object detection via low rank matrix recovery. In *CVPR*, pages 853–860. IEEE, 2012.
- [30] N. Sundaram and K. Keutzer. Long term video segmentation through pixel level spectral clustering on gpus. In *ICCV Workshops*, pages 475–482. IEEE, 2011.
- [31] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science, 290(5500):2319–2323*, 2000.
- [32] Y. Wei, F. Wen, W. Zhu, and J. Sun. Geodesic saliency using background priors. In *ECCV*, pages 29–42. Springer, 2012.
- [33] Y. Xie, H. Lu, and M.-H. Yang. Bayesian saliency via low and mid level cues. *TIP, 22(5):1689–1698*, 2013.
- [34] L. Xu, Q. Yan, Y. Xia, and J. Jia. Structure extraction from texture via relative total variation. *ACM TOG, 31(6):139*, 2012.
- [35] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. In *CVPR*, June 2013.
- [36] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, pages 3166–3173. IEEE, 2013.
- [37] J. Yang, B. Price, S. Cohen, Z. Lin, and M.-H. Yang. Patch-cut: Data-driven object segmentation via local shape transfer. In *CVPR*, pages 1770–1778. IEEE, 2015.
- [38] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf. Ranking on data manifolds. *NIPS, 16:169–176*, 2004.
- [39] W. Zhu, S. Liang, Y. Wei, and J. Sun. Saliency optimization from robust background detection. In *CVPR*, pages 2814–2821. IEEE, 2014.