Salient object detection via spectral clustering

Xiaoling Hu, Wenming Yang*, Xingjun Wang, Qingmin Liao
Shenzhen Key Lab. of Information Sci&Tech/Shenzhen Engineering Lab. of IS&DRM
Department of Electronic Engineering/Graduate School at Shenzhen, Tsinghua University, China
Email: hxl14@mails.tsinghua.edu.cn, yangelwm@163.com, liaoqm@sz.tsinghua.edu.cn

***Abstract*—Detection of salient object is useful in many computer tasks. In this paper, we propose a novel spectral clustering (SC) based method to detect salient object. Image can be expressed as a graph which is composed by nodes and edges, where nodes are superpixels generated by segmenting algorithms and edge strengths are proportional to superpixels similarity. We jointly consider color distribution and spatial distance to measure the similarity between superpixels. Then SC algorithm is applied to cluster the nodes(superpixels) into two classes, one class for foreground and the other for background. It is reasonable that salient regions are far different from the background, so we can take salient object detection as a two-class clustering problem. Extensive experiments on a public database demonstrate that our model is not only easy to implement but also outperforms lots of recently proposed methods.**

***Index Terms*—Salient detection, superpixel, graph-based, spectral clustering (SC)**

I. Introduction

OUR human vision system can quickly pick out interesting parts from a complicated scene. This ability also enables animals to efficiently find prey and avoid predators, which is important for them to survive. Due to this, a lot of researchers have done a lot to learn the mechanism of attention. Salient detection aims to find parts of an image that attract the most attention, and it can be applied to many research areas in image processing, and the research of salient detection has efficiently promoted the progress of other fields.

Generally, salient object detection models can be categorized as two classes: bottom-up and top-down. The former methods [1], [2] are fast, driven by data, while the latter methods [3] are slow, driven by goal and may use some prior knowledge.

Over the past decade, most researchers pay attention to bottom-up methods. Bottom-up saliency methods mainly measure saliency by measuring contrast (local or global) or rarity of features over the entire image, and rely on prior knowledge. Different saliency models use different prior knowledge. In [1], Itti *et al.* focus on color and orientation, and use center-surround contrast to find salient regions. Bruce *et al.* [4] compute saliency by self-information measurement based on local contrast. On the contrary, Cheng *et al.* [2] use global contrast based on region contrast to detect visual saliency. In this way, we can usually detect the whole salient object. Perazzi *et al.* [5] modify Cheng *et al.*$'s$ idea and propose a linear-time computation method. Fourier spectrum is another novel method to detect visual saliency [6]. Wei *et al.* [7] use geodesic distance to detect salient objects. Though these methods can detect some interesting points in the image, but they can't find the exactly salient objects. Recently, graph-based methods have come to be increasingly used to detect salient objects [8]. They map an image into a graph $G(V, E)$, in which V stands for vertices (nodes) and E stands for undirected edges. The nodes are superpixels generated by segmenting algorithms, and the edges stand for the relationships between nodes. Gopalakrishnan *et al.* [9] apply random walks in graph, and they assume that it is related to image saliency. Yang *et al.* [8] formulate saliency object detection as a graph-based ranking problem and propose a graph labelling solution for it. But how to construct a graph that can characterize the original image well still remains a problem.

A new graph-based model based on spectral clustering (SC) to detect salient objects is proposed in this paper. It is observed that the foreground would group together, and differs from the background greatly. We take it for granted that clustering all the pixels of an image into two classes would be able to segment salient objects. So the new model is proposed as follows. First, SLIC method is one of the most popular algorithms that can be applied to divide an image into regions as nodes of constructed graph. Then, color feature and spatial distance are combined to measure the similarity between nodes, with the strength between nodes proportional to the similarity. Last, we use SC algorithm to cluster the nodes into two classes. One class is for foreground, and the other for background.

There are some contributions in our paper:

1.A novel model for salient object detection is proposed in this paper. Through segmentation and distance measurement, image is represented as an adjacent graph in our model. This new model is easy to implement and preforms well in our test dataset;

2.Spectral clustering is applied in our model. It is an efficient algorithm in graph theory and has rarely been used in salient object detection.

In our paper, section 1 introduces the paper. The details of our idea are represented in the next section. Section 3 display the experimental results. And we make the concludes in the last section.

II. Graph-Based Spectral Clustering Model

First, image can be expressed as a graph, and effective metrics can be found to measure the similarity between nodes. Second, salient regions will always group together and are far

*Corresponding author: yangelwm@163.com

978-1-4673-9194-8/16/$31.00 ©2016 IEEE

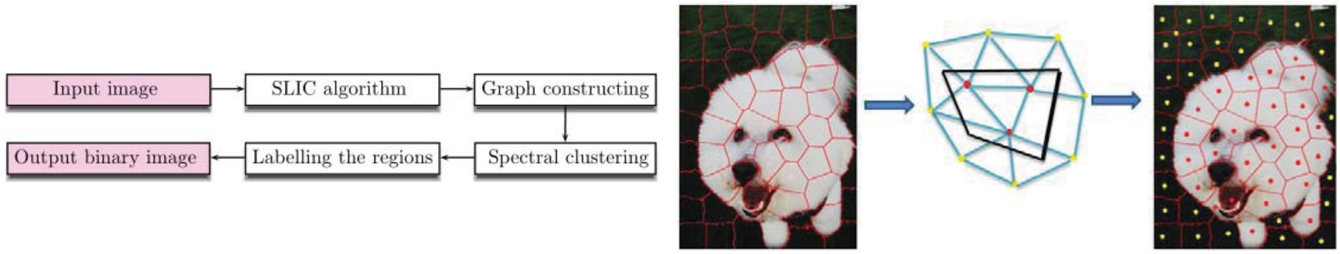

Fig. 1. Salient object detection via spectral clustering based on graph. Left figure is the flowchart of the proposed model, and the right is the schematic diagram.

different from the background. So clustering algorithms can be applied to obtain the salient object.

The main steps of the proposed method is represented in Figure 1. First, we use SLIC algorithm to segment an image into superpixels. Taking the superpixels as the nodes in a graph and an adjacent graph is constructed. Then spectral clustering is applied to divide the graph into subgraphs.

The goals of our model is to construct a graph which can characterize the original image well and cluster the nodes efficiently.

A. Graph Construction

As mentioned above, image can be expressed as a graph. Specially, the edge strength between region i and j is represented as w_{ij}, computed based on the similarity between the two nodes. Instead of constructing a k-regular graph in [8], an adjacent graph which is of higher efficiency is constructed in our model. On the graph, the neighboring nodes are connected. In addition, we let the nodes which stands for the image boundary are connected.

As mentioned above, we construct a sparse graph, which means most edge strengths between nodes are zero.

B. Distance Measurement

In our method, one of our goals is to construct a suitable graph which can characterize the original image well. The key to construct a good graph is to choose an efficient metric to measure the distances between different superpixels.

Color and spatial distances are combined in our model to measure the similarity between two superpixels. Two features are extracted from a pixel, with color represented in the CIELAB color space $[l\ a\ b]$, and the pixel's spatial position $[x\ y]$.

Each superpixel is color consistent after segmenting, so it is reasonable to compute the mean $[l\ a\ b]$ vector of all the pixels belonging to a superpixel, and then use the mean vector to capture the color feature of the superpixel. The mean vector of superpixel i and j is $[l_i\ a_i\ b_i]$ and $[l_j\ a_j\ b_j]$, respectively. So the distance in color field between superpixel i and j is computed as

$$d_c = \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2}, \tag{1}$$

In the same way, the mean vector $[x_i\ y_i]$ is used to represent the position of superpixel i. And the distance of spatial field between superpixel i and j is computed as

$$d_s = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}, \tag{2}$$

Then color and spatial distances are normalized to $[0, 1]$. Also, the two terms (color and spatial distance) may not contribute equally, so we introduce a variable λ to balance the two terms. At last, the distance between superpixel i and j d_{ij} is written as

$$d_{ij} = \sqrt{(\frac{d_c}{N_c})^2 + \lambda(\frac{d_s}{N_s})^2}, \tag{3}$$

Thus, the strength between two regions (superpixels) is calculated as

$$w_{ij} = e^{-\frac{||d_{ij}||}{\sigma^2}} \quad i, j \in V, \tag{4}$$

where σ is used to control the weight. The results of our model show that the metric is easy and of high efficiency.

C. Spectral Clustering

The other goal of our method is to cluster the nodes efficiently. Spectral clustering is applied in our model.

Spectral clustering is an algorithm based on graph aiming to divide undirected graph into two or more subgraphs. We expect the nodes in the same subgraph are similar, while the nodes in different subgraphs are different.

The basic idea of spectral clustering is to do eigen decomposition of the similarity matrix (Laplace matrix), and then cluster the eigenvectors.

One of the most popular spectral clustering algorithms which is used widely in image processing is **Normalized cuts**.

Normalized cuts:A graph $G = (V, E)$ can be divided into two parts, part A, B. A and B satisfy: $A \cup B = V, A \cap B = \emptyset$. In graph theory, it is calculated as:

$$cut(A, B) = \sum_{\mathrm{u} \in A, v \in B} w(u, v). \tag{5}$$

The optimal solution of dichotomy for a graph is to minimize *cut* value. To avoid the circumstances that one part will be small set, Shi *et al.* propose a new measurement of disassociation, and it is named *normalized cut (Ncut):*

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}, \tag{6}$$

in which $assoc(A, V) = \sum_{u \in A, v \in V} w(u, v)$ is the sum of the connections between node A and all other nodes, and *assoc(B,V)* is similarly defined.

Similarity matrix (Laplace matrix) W can be obtained from our analysis above, and the normalized Laplacian matrix is calculated as

$$L = I - D^{-1/2} W D^{-1/2}, \tag{7}$$

D is the diagonal matrix, where

$$D_{ii} = \sum_j W_{ij}. \tag{8}$$

We use *normalized cut* algorithm to divide all points in the graph into two sets (A, B). It is expected that the salient nodes would group together and we could find the salient object efficiently.

The main steps of the proposed algorithm are summarized in Algorithm 1 as follows.

Algorithm 1 Graph-based salient object detection via spectral clustering.

Input: An input natural image, I;
Output: A binary image, $I^{'}$;
1: For a natural image I, divide it into superpixels $V_i(i = 1, 2, \ldots, M)$;
2: Represent the image as a graph $G(V, E)$, where V is the set of nodes and E is the set of edges;
3: Compute the strength between nodes;
4: Apply spectral clustering algorithm to the generated graph;
5: Cluster the nodes of the graph into two classes, one for foreground and the other for background.

III. Experiments

We cluster all the pixels in the image into two clusters in our model. After dividing the image graph into two subgraphs, it remains a problem for us to decide which one is the foreground. In salient object detection, *central bias* is a commonly used premise, indicating that our eyes will focus on the center of the image. So the salient regions will be always in the center of an image. We calculate the pixel numbers on the image boundary. The cluster which has more pixels on the image border is the background, and the other cluster is the salient object. The binary image is used to compare with the ground truth image to obtain the precision and recall. And then we can calculate the average precision and recall from all the test images.

A. Experimental setup

In the experiment, we set the number of superpixels N=200. There are two other parameters in our method: λ in Eq.3 to balance color distance and spatial distance, and the σ in Eq.4 to control the edge weight. In our experiment, $\lambda = 0.001$ and $\sigma^2 = 0.125$.

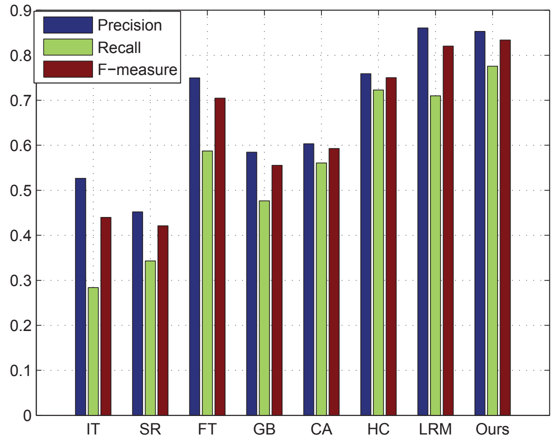

Fig. 2. Average precision, recall and F-measure on the 1000-image dataset. F-measure of the proposed method achieves the best.

B. Evaluation Metrics

Taking salient object detection as a two-class clustering problem, we can get a binary image using our model.

We evaluate our method by precision, recall and F-measure. The precision is defined as: the ratio of salient pixels correctly detected to the number of pixels in the extracted regions. And the recall is defined as: the ratio of salient pixels correctly detected to the ground truth. Then, the F-measure is computed as:

$$F_\beta = \frac{(1 + \beta^2) \times Precision \times Recall}{\beta^2 \times Precision + Recall} \tag{9}$$

where we set $\beta^2 = 0.3$, similar to [10].

When comparing with other methods, we follow [10], [2], [5] to use a threshold T_α to get the binary saliency maps obtained by other methods before calculate F_β. The threshold is set as follows:

$$T_\alpha = \frac{2}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} S(x, y) \tag{10}$$

C. Quantitative Comparison

In this paper, the proposed method is compared with other recently proposed methods in the public dataset. The compared methods are: (FT [10], HC [2]), graph-based (GB [11]), spectrum-based (SR [6]), high-level priors based (CA [12]), and the one with low rank matrix recovery (LRM [13]), most of which were proposed recently.

We test our method on the 1000 images, and the average precision, recall and F-measure are shown in Figure 2.

And a few visual results of the proposed method are represented in Figure 3 shows. Overall, the results shows that the proposed method works well compared with other methods.

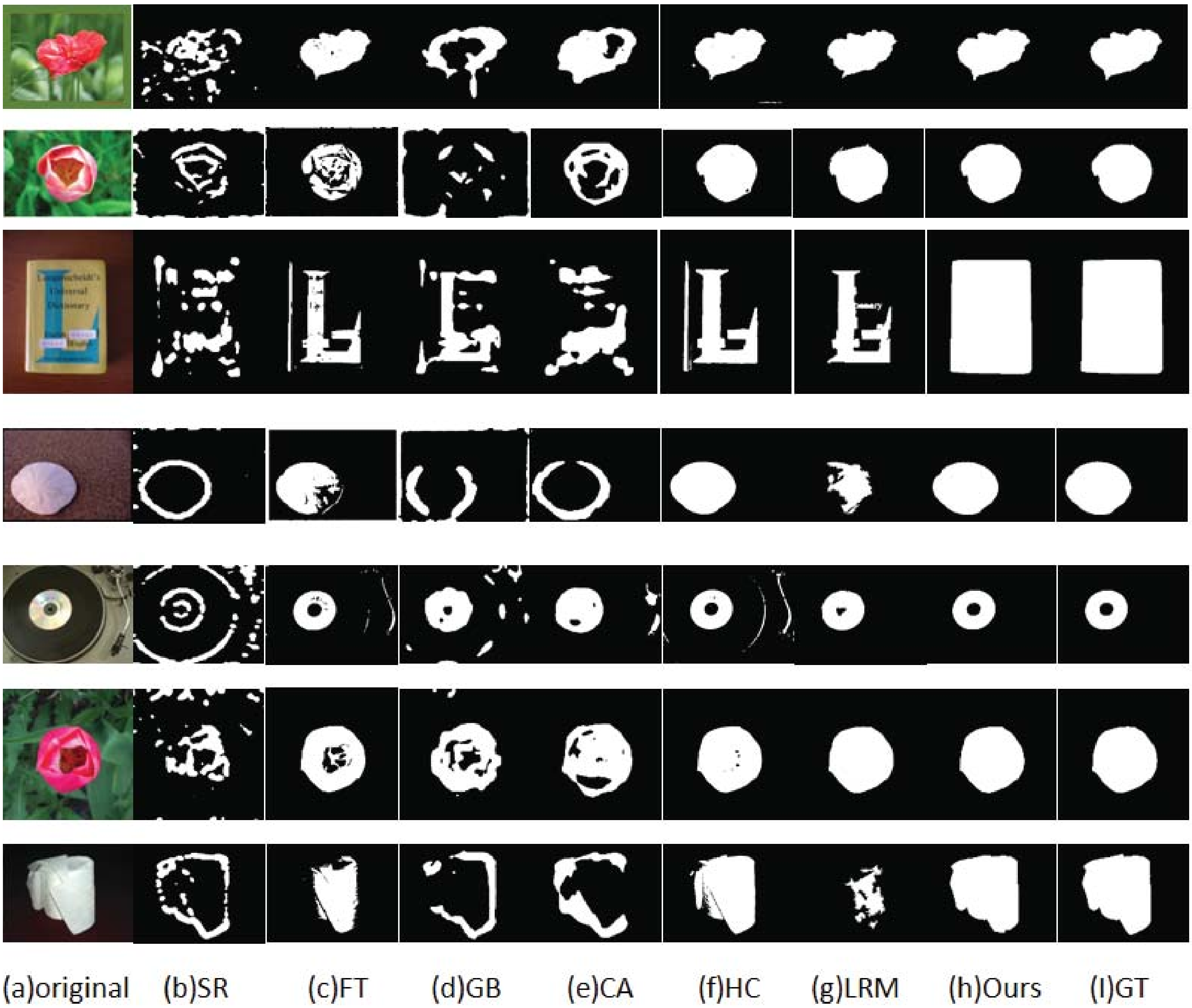

Fig. 3. Saliency detection results of different methods. (a)original images, results produced using (b)SR, (c)FT, (d)GB, (e)CA, (f)HC, (g)LRM, (h)our method and (i)ground truth.

TABLE I
AVERAGE TIME TAKEN TO COMPUTE BINARY SALIENT REGIONS IN THE DATABASE. CODE: M=MATLAB, C=C/C++.

Method	SR	FT	GB	CA	LRM	Ours
Time(s)	0.075	0.095	1.027	66.281	14.233	0.907
Code	M	M	M+C	M+C	M+C	M

D. Efficiency

We use our method and the others to compute binary salient regions for the test images. Our method is implemented by Matlab, while the other algorithms are implemented in Matlab or C/C++, and we use the authors′ implementations. Table I compares the implementation time of each method. These methods are tested on a machine with Intel Quad Core i5-2500 3.30GHz CPU and 4.00GB RAM. The running time shows that our salient object detection model is efficient.

IV. CONCLUSION

A novel model for saliency object detection is proposed in this paper. In this model, an image is segmented into several superpixels, and is represented as a graph. Spectral clustering algorithm is applied to cluster the nodes into two clusters. One is for foreground, and the other for background. We evaluated the proposed method on large a dataset and extensive experiments on a large database show that the proposed model is not only easy to implement but also outperforms state-of-the-art methods.

V. ACKNOWLEDGMENT

This work was supported by NSFC under Grant No.61471216 and Special Foundation for the Development of Strategic Emerging Industries of Shenzhen under Grant No.ZDSYS20140509172959974.

References

[1] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 11, pp. 1254–1259, 1998.

[2] M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S. Hu, "Global contrast based salient region detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 37, no. 3, pp. 569–582, 2015.

[3] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum, "Learning to detect a salient object," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 2, pp. 353–367, 2011.

[4] N. Bruce and J. Tsotsos, "Saliency based on information maximization," in *Advances in neural information processing systems*, 2005, pp. 155–162.

[5] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung, "Saliency filters: Contrast based filtering for salient region detection," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 733–740.

[6] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.

[7] Y. Wei, F. Wen, W. Zhu, and J. Sun, "Geodesic saliency using background priors," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 29–42.

[8] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 3166–3173.

[9] V. Gopalakrishnan, Y. Hu, and D. Rajan, "Random walks on graphs to model saliency in images," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1698–1705.

[10] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Computer vision and pattern recognition, 2009. cvpr 2009. ieee conference on*. IEEE, 2009, pp. 1597–1604.

[11] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Advances in neural information processing systems*, 2006, pp. 545–552.

[12] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 10, pp. 1915–1926, 2012.

[13] X. Shen and Y. Wu, "A unified approach to salient object detection via low rank matrix recovery," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 853–860.