Saliency Detection with Sparse Prototypes: An Approach Based on Multi-Dictionary Sparse Encoding

Jun Wang*, Zemin Wu, Chang Tian and Lei Hu

College of Communications Engineering, Army Engineering University of PLA, Nanjing, China.

Abstract. This paper proposes a bottom-up saliency detection algorithm based on multi-dictionary sparse recovery. Firstly, the SLIC algorithm is used to segment the image into superpixels in multilevel and atoms with a high background possibility are selected from the boundary superpixels to construct the multi-dictionary. Secondly, sparse recovery of the entire image is achieved using multi-dictionary to get sub-saliency maps from the perspective of sparse recovery errors. The final saliency map is generated in a weighted fusion manner. Experimental results on three public datasets demonstrate the effectiveness of our model.

1 Introduction

Visual saliency detection aims to identify the most distinctive regions in an image that draw human attention. This topic has received increasing interest in recent years because of its wide range of applications such as image segmentation, image retrieval, target recognition, image classification, and video compression. Although much significant progress has been made, salient object detection remains a challenging problem.

Sparse representation has been widely used in computer vision task in recent years. The typical sparse representation based saliency detection models have the following two types. The first type is the centre-surround model. Li et al. [1] and Han et al. [2] reconstruct a central image patches using its nearby local regions (i.e., using the nearby local regions as dictionary to recover the centre patch) and then defined its saliency degree using sparse reconstruction error. However, a given patch is usually very similar to those nearby. As a result, the internal region of a large object cannot be detected well. The second method is based on the hypothesis that image boundaries belong to the background in most cases. Li et al. [3] and Jia et al. [4] propose to construct a background dictionary using superpixels at image boundary and perform sparse recovery to get the saliency map, as shown in Figure 2(a). Recovery error is used to determine saliency degree. In general, their performance is better than that of centre-surround model. However, they are imprecise for using the entire boundary superpixels as dictionary atoms if the object appears on the boundary.

Different from pervious works, in this paper, we propose a bottom-up saliency detection algorithm with sparse prototypes. First, the SLIC algorithm [5] is used to segment the image into superpixels in multi-level and reliable atoms is selected from boundary super-pixels to

construct multi-dictionary. Second, sparse encoding of the entire image is achieved based on these dictionaries. The initial sub-saliency maps are generated according to the sparse recovery errors. The final saliency map is obtained through sub-saliency maps' fusion. Our algorithm flow is shown in Fig. 1 and we name it 'MDSR' in the following.

Fig. 1. The overall flow of our algorithm

2 Dictionary construction

To better capture intrinsic structural information and improve computational efficiency, an input image I is segmented into N super-pixels $\{p_i\}_{i=1}^N$ by the simple linear iterative clustering (SLIC) algorithm. The mean colour features c_i in CIE-Lab colour space and mean

* Corresponding author: wangjun_ice@126.com

normalized coordinates $\mathbf{x}_i$ of pixels are used to describe each superpixel p_i .

$$p_i = [\mathbf{c}_i; \mathbf{x}_i] \quad (1)$$

Therefore, the entire image can be represented as:

$$I = [p_1, p_2, \dots, p_N] \quad (2)$$

It has been shown that image boundaries are good visual cues for background models which can be exploited for saliency detection [3, 4]. On the other hand, salient object are likely to appear at the centre of a scene. But these assumptions may not always hold. So instead of directly extracted all the image boundary superpixels to construct dictionary, we firstly compute the background possibility of each boundary superpixel:

$$P_{bg}(p_i) = 1 - \exp\left(-\frac{BC^2(p_i)}{2\delta^2}\right) \quad (3)$$

$BC(p_i)$ stands for the boundary connectivity of superpixel p_i , which was proposed by Zhu et al. [6] to estimate the probability that an image patch belongs to background. It is defined by:

$$BC(p_i) = \frac{L(p_i)}{\sqrt{A(p_i)}} \quad (4)$$

$$L(p_i) = \sum_{j=1}^N \exp\left(-\frac{d_{geo}^2(p_i, p_j)}{2\delta_L^2}\right) \times \delta(p_j \in I^{bd}) \quad (5)$$

$$A(p_i) = \sum_{j=1}^N \exp\left(-\frac{d_{geo}^2(p_i, p_j)}{2\delta_L^2}\right) \quad (6)$$

Where $L(p_i)$ is the length along the image boundary I^{bd} . $\delta(\cdot)$ is 1 for superpixels on the image boundary and 0 otherwise. $A(p_i)$ represents the spanning area of p_i , and $d_{geo}^2(p_i, p_j)$ indicates the geodesic distance between the superpixel p_i and p_j in the CIE-Lab colour space.

Then we select the boundary superpixels with its background possibility value bigger than ξ as the atoms. ξ is defined as bellow:

$$\xi = \frac{1}{2K} \sum_{j=1}^K P_{bg}(p_j) \quad (7)$$

K denotes the number of all boundary superpixels. The dictionary $\mathbf{D}$ is constructed based on the selected atoms, as shown in Fig. 2(b).

Fig. 2. Comparison of dictionary construction: (a) traditional methods which directly use the entire boundary superpixels as the atoms of dictionary. (b) Our method that select the

superpixels with high background possibility as the atoms (the superpixels marked with small white square).

3 Saliency computing

3.1 Sparse encoding

We use the set of dictionary $\mathbf{D}$ as the bases for sparse representation, and encoded the superpixel p_i by:

$$\mathbf{a}_i = \arg \min_{\mathbf{a}_i} \|\mathbf{p}_i - \mathbf{D} \cdot \mathbf{a}_i\|_2^2 + \lambda \|\mathbf{a}_i\|_1 \quad (8)$$

The sparse recovery error is:

$$\varepsilon_i = \|\mathbf{p}_i - \mathbf{D} \cdot \mathbf{a}_i\|_2^2 \quad (9)$$

Therefore, the reconstruction error matrix $\mathbf{D}$ of the entire image based dictionary is:

$$\mathbf{E} = [\varepsilon_1, \varepsilon_2, \dots, \varepsilon_N] \quad (10)$$

According to theory of sparse representation, for the superpixels belonging to salient region (e.g., the person in Fig. 2), their features are quite different from the atoms in the dictionary, so the sparse recovery error is larger than that of superpixels belonging to the background (e.g., the lawn in Fig. 2). So, we could calculate the saliency value of every superpixel via sparse recovery error. To reduce the impact of some noise, we use a sigmoid function to optimize it.

$$\mathbf{S} = [s_1, s_2, \dots, s_N] \quad (11)$$

$$s_i = \frac{1}{1 + \exp\left(-b\left(\varepsilon_i - \bar{\varepsilon}\right)/\bar{\varepsilon}\right)} \quad (12)$$

s_i is the saliency value of superpixel p_i . $\bar{\varepsilon}$ is the mean of ε . b control the magnitude of stretch and is set to 10 empirically.

3.2 Saliency fusion

The scale of salient object is different, and the degree of its contact with the image boundary also varies under different circumstance. To handle this problem, we segment the image at M different scales. At every segment scale, we construct the dictionary respectively as shown in Fig. 1. We perform sparse recovery and optimize tasks based on every dictionary to get M sub-saliency maps. The final saliency map is calculated in a weighted fusion manner:

$$\mathbf{S}_{final} = \frac{1}{M} \sum_{i=1}^M \mathbf{S}_i \quad (13)$$

4 Experiment result

4.1 Experiment setting

We test each algorithm on three public datasets ASD [7], MSRA5000 [8], and SED2 [9]. ASD consists of 1,000 images. It is simple and widely used in practice. MSRA5000 consists of 5,000 challenging images characterized by complex backgrounds and objects at the boundary. SED2 has 100 images, where each image has

two objects. Most objects in SED2 are located at the boundary.

For each image, we obtained 10 saliency maps using the proposed model and other 9 state-of-the-art methods: GB [10], SR [11], PCA [12], DSR [3], LR [13], RPCA [14], BFS [15], RBD [6], and BSCA [16]. We use AUC value and MAE value to evaluate all algorithms. For each method, a binary map is obtained by segmenting each saliency map with a given threshold $T \in [0, 255]$ and then compared with the ground truth mask G to compute the true positive and false positive rates to plot the ROC curve and compute the area under ROC curve to get the AUC value. The MAE metric calculates the average absolute error between the saliency map and ground truth mask of test image ($W \times H$ resolution) in pixel:

$$MAE = \frac{1}{W \times H} \sum_{i=1}^W \sum_{j=1}^H |S(i, j) - G(i, j)| \quad (14)$$

When in experiment, we segment every test image at three scale (i.e., we set $M = 3$) for accuracy and efficiency. The segment parameters are [100 10], [200 20] and [300 30]. The parameter δ in equation (3) is set to 1. The sparse reconstruction parameter λ in equation (8) is set to 0.01.

4.2 Experiment results

As in shown in Table 1, our proposed algorithm outperforms other 9 methods lower MAE and higher AUC scores, which indicates the predicted saliency map is more similar to ground truth. Note that, the advantage of our algorithm is more obvious than others in SED2 datasets, demonstrating the effectiveness of MDSR in addressing the boundary saliency problems. A visual comparison of several saliency maps is give in Fig. 3. For the first example, the image background is complicated and there is an interference object at the boundary. For the other test images, the salient object all contacts the image boundaries. As can be seen clearly, compared with the saliency maps generated by other algorithms, our results can highlight the salient object more clearly and completely under different scenes, which demonstrates the robustness of our model.

Table 1: Comparison of AUC and MAE value on three public datasets. (The best performance are marked with red color)

	ASD		MSRA5000		SED2	
	AUC	MAE	AUC	MAE	AUC	MAE
GB	.8858	.2289	.8711	.2411	.8441	.2436
SR	.7651	.2299	.7553	.2422	.8193	.2250
RPCA	.8994	.2090	.8724	.2265	.8536	.2284
PCA	.9702	.1552	.9481	.1889	.9157	.2015
LR	.9584	.1885	.9238	.2215	.8869	.2374
BFS	.9702	.1285	.9481	.1666	.9157	.1669
DSR	.9792	.0809	.9530	.1204	.9172	.1388
BSCA	.9783	.0855	.9533	.1312	.8965	.1632
RBD	.9702	.0659	.9481	.1106	.9157	.1301
Ours	.9810	.0558	.9527	.1101	.9441	.1289

Fig. 3. Visual comparison between our results and others. (a) Input. (b) SR. (c) GB. (d) RPCA. (e) PCA. (f) LR. (g) DSR. (h) BFS. (i) RBD. (j) BSCA. (k) Ours. (l) Ground truth mask.

5. Conclusion

In this paper, we propose a bottom-up saliency detection algorithm via multi-dictionary sparse encoding. To handle the misjudgment problem when salient objects appear at the image boundary, we segment the image in multilevel. In each segment level, we compute the background possibility of superpixels located at image boundaries and then select the reliable ones as the atoms of dictionary. Based on the multi-dictionary, sparse encoding of the whole image is achieved and the saliency map s are generated according to the sparse recovery errors. Experimental results demonstrate the effectiveness of our algorithm.

It is necessary to explain that our method also has good expansibility. If we have a strict requirement for the accuracy of the result in practice, we can set a larger M . However, if we emphasis on the speediness, we can set a smaller M . In the further research, we will focus on the relationship between the speediness and accuracy of our algorithm.

References

1. Li Y, Zhou Y, Xu L, et al. Incremental sparse saliency detection. *IEEE International Conference on Image Processing*. IEEE Press, 2009: 3057-3060.
2. B. Han, H. Zhu, and Y.D. Ding, 2011. Bottom-up Saliency based on weighted sparse coding residual. In *Proceeding of ACM International conference on multimedia*, 2011: 1117-1120.
3. X. Li, H. Lu, Zhang, X. Ruan et al, 2013. Saliency detection via dense and sparse reconstruction. In *IEEE International Conference on Computer Vision*.
4. Jia. C, J. Qi, X. Li et al. (2016), "Saliency detection via a unified generative and discriminative model". *Neurocomputing*, Vol. 173 No.P2, pp. 406-417.
5. Achanta R, Shaji A, Smith K, et al. (2012), "Slic superpixels compared to state-of-the-art superpixel methods". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 34 No.12, pp. 2274–2281.
6. Zhu W, Liang S, Wei Y, et al, 2014. Saliency optimization from robust background detection. In

IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Columbus, USA.

7. Achanta R, Hemani S, Estrada F, et al, 2009. Frequency-tuned salient region detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Miami, USA.
8. Liu T, Sun J, Zheng N, et al. "Learning to detect a salient object". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2011, 33(2): 353-367.
9. Alpert S, Galun M, Brandt A, et al. "Image segmentation by probabilistic bottom-up aggregation and cue integration". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012, 34(2): 315-327.
10. Harel J, Koch C, and Perona P, 2006. Graph-based visual saliency. In *Proceedings of Neural Information Processing Systems*.
11. Hou X, Zhang L., 2007. Saliency detection: a spectral residual approach. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Portland, USA.
12. Margolin R, Tal A, and Zelnik-Manor L, 2013. What makes a patch distinct? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Portland, USA.
13. X. Shen, Y. Wu, 2012. "A Unified Approach to Salient Object Detection via Low Rank Matrix Recovery". *Computer Vision and Pattern Recognition*, Vol. 23 No. 10, pp. 853-860.
14. J. Yan, M Zhu, H Liu, et al. (2010), "Visual Saliency Detection via Sparse Pursuit". *IEEE Signal Processing Letters*, Vol. 17 No. 8, pp. 739-742.
15. J Wang, H Lu, N Tong, et al. (2016), "Saliency Detection via Background and Foreground Seed Selection". *Neurocomputing*, Vol. 152 No. C, pp. 359-268.
16. Qin Y, Lu H, Xu Y, et al, 2015. Saliency detection via cellular automata. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, USA, pp. 111-119.