MULTI-SCALE ANALYSIS OF COLOR AND TEXTURE FOR SALIENT OBJECT DETECTION

Ketan Tang, Oscar C. Au, Lu Fang, Zhiding Yu, Yuanfang Guo

Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology

ABSTRACT

In this paper we propose a multi-scale segment-based framework for salient object detection. In this framework texture and color features are used together to provide diverse information of salient object. Segmentation is performed on three different scales so that the object boundary can be accurately captured with high probability. Besides, we propose a novel adaptive feature combination mechanism to combine the saliency maps produced with different features, in which the combining weight of each saliency map is learned using online learning. Experiment results demonstrate that the proposed method significantly outperforms the state-of-the-art methods.

Index Terms— salient object detection, multi-scale analysis, online learning, color, texture

1. INTRODUCTION

Detecting salient object in images has recently received significant attention. The goal of salient object detection is to find the most informative and important region in images and then facilitate other image processing. It has a lot of applications, for example, video abstraction/summarization [1], adaptive image/video display on small devices [2], image/video compression, and object detection/recognition [3].

Most of salient object detection algorithms are based on pixels [4], or based on small rectangle blocks [5]. The drawbacks with these methods are 1) the computational complexity is too high; 2) the obtained salient region is not complete. To reduce computational complexity and find the object boundary more accurately, some segment-based methods are proposed, such as [6, 7]. The main problem of these methods is that, if the segmentation is not accurate, the result may be bad. Another problem with [6] is that they use only color information to compute saliency map. For most images it is enough, but for some grayscale images, there is no dominant color, thus only using color information is no longer sufficient. To solve this problem, Gopalakrishnan et al. [8] incorporate texture feature into the framework. They perform Fourier transform in a 8×8 block, and compute the orientation histogram according to the phase component of Fourier coefficients. The main problem of this method is, they have to segment images into small rectangle blocks since they use Fourier transform. In cases where the object is not rectangular, this method cannot find the object boundary accurately.

In this paper we propose a multi-scale framework for color and texture analysis based on segments. The contribution is threefold. Firstly, we propose a multi-scale framework. We understand that it is difficult to capture the object accurately through one segmentation, so we perform multiple segmentations on different scales and hopefully the object boundary can be accurately found by one of them. Secondly, we use Local Binary Pattern (LBP) ([9]) to extract the texture information. The goodness of LBP is that, it is robust and efficient. Also, it does not have to be computed in a rectangle, so it is easy to incorporate LBP in our segment-based framework. Thirdly, we propose a novel adaptive feature combination strategy using online learning. Compared to MRF in [10], online learning is much easier to implement, and the performance is comparable to MRF.

The paper is organized as follows. In Section 2 we systematically introduce the framework and the two operators. The combination strategy for multiple saliency maps follows in Section 3. In Section 4 we present our experiment results, including some detection figures and accuracy tables. In Section 5 we conclude our work and point our future research direction.

2. MULTI-SCALE ANALYSIS FOR COLOR AND TEXTURE BASED ON SEGMENTS

In this section a multi-scale framework for computing feature maps is introduced. The framework contains two operators as in [6], one is Center Surround Contrast operator (CSC), and the other one is Spatial Distribution Variation operator (SDV). For each of the two operators, users can input any feature and obtain a saliency map as output. In this paper we choose color histogram and LBP histogram as the features.

We first segment images using Efficient Graph Cuts (EGS) [1]. Then each image is represented by a set of segments. And then we do salient object detection on segment basis using the two operators. To accurately detect object

This work has been supported in part by the the Research Grants Council (RGC) of the Hong Kong Special Administrative Region, China. (GRF Project no. 610210)

with different scales, we downsample the original image to 1/2 and 1/4 of the original size, and redo the procedure. The saliency maps under the three scales are combined as the final output.

2.1. Center Surround Contrast

A salient object is usually quite different from its surrounding context. This difference can be expressed by center-surround contrast. Different from [10], we extract this feature on the basis of segments.

For a segment S, we first dilate it to get a larger region including S. Subtracting S from this larger region we get the surrounding region R. The dilation size is chosen as:

$$\text{dilation size} = \frac{\sqrt{2}-1}{2} \times \text{mean(height,width)} \qquad (1)$$

The reason we use (1) is that, when S is close to a square, the surrounding region R will have similar area as S. To do dilation, we use Matlab built-in function *imdilate*.

Suppose the feature vectors for S and R are f_C and f_S, we calculate the χ^2 distance of the two histograms, and this is the saliency value of segment S:

$$\chi^2(S,R) = \frac{1}{2}\sum_j \frac{(f_S(j)-f_R(j))^2}{f_S(j)+f_R(j)} \qquad (2)$$

Repeat this procedure for all segments and we get the Center Surround Contrast saliency map.

2.2. Spatial Distribution Variation

Color spatial distribution variation is firstly used in [10] and proved quite efficient. They use spatial variance to calculate the saliency of each color components. The idea is, the wider a color is distributed in the image, the less possible the salient object contains this color. In this paper we follow the idea of considering cluster saliency using spatial distribution variation, but we calculate saliency value on segment unit, and we use compactness and scatterness together as the measure of cluster importance. Moreover, our method incorporates both color and texture features, which makes our method more robust in the cases where there is no dominant color in the image.

2.2.1. GMM cluster learning

We represent all segment features (this feature can be color or texture) by a Gaussian Mixture Model (GMM)$\{\omega_c, \mu_c, \Sigma_c\}_{c=1}^{C}$, where ω_c, μ_c, Σ_c is the weight, the mean value and the variance matrix of the cth component. C is the number of clusters. In our experiment we set $C = 5$ for color feature, and $C = 7$ for LBP feature. Suppose segment S is represented by feature vector f_S, it is assigned to a component c with the probability:

$$p(c|f_S) = \frac{\omega_c N(f_S|\mu_c,\Sigma_c)}{\sum_c^C \omega_c N(f_S|\mu_c,\Sigma_c)}. \qquad (3)$$

By setting $p(c|\mathbf{x}) = p(c|f_S)$ for all pixels $\mathbf{x} \in S$ we get the posterior probability of pixel $\mathbf{x}$ belonging to cluster c.

2.2.2. Cluster Compactness and Scatterness

To evaluate the saliency of each cluster, we use compactness and scatterness jointly as a measurement. Compactness is to evaluate how compact the cluster is. Background clusters tend to have a larger spread compared to salient clusters. We use spatial variance to evaluate the compactness. The larger the spatial variance is, the less compact the cluster is.

$$V(c) = \frac{\sum_{\mathbf{x}} \|\mathbf{x}-\mu_c\|^2 \cdot p(c|x)}{\sum_{\mathbf{x}} p(c|\mathbf{x})} \qquad (4)$$

where μ_c is the spatial mean of the cth cluster. V is normalized to $[0,1]$.

Scatterness is to evaluated how scattered a cluster is. The observation is, if a cluster is mixed with other clusters, this cluster tends to be less salient. The scatterness is calculated as follows.

$$SCA(c) = \frac{\sum_{\mathbf{x}} p(c|\mathbf{x}) \sum_{\mathbf{y}\in N_{\mathbf{x}}} \|p(\mathbf{x})-p(\mathbf{y})\|}{\sum_{\mathbf{x}} p(c|\mathbf{x})} \qquad (5)$$

where $N_{\mathbf{x}}$ is the 8-point neighborhood of pixel $\mathbf{x}$, $p(\mathbf{x})$ is the posterior probability vector of $\mathbf{x}$. Again SCA is normalized to $[0,1]$.

The final saliency value Sal of pixel $\mathbf{x}$ is computed as

$$Sal_{\mathbf{x}} = \sum_c (1-V(c)) \cdot (1-SCA(c)) \cdot p(c|\mathbf{x}) \qquad (6)$$

2.3. Color map

For the two operators, the color features are different. For Center Surround Contrast, the color feature is a 768×1 histogram if current image is a color image, or a 256×1 histogram if it is a grayscale image. We calculate the histogram for each of the color channels, normalize them, and then concatenate them to get a long histogram. Feed this feature into CSC operator and we obtain the color-CSC saliency map.

For Spatial Distribution Variation, we treat a segment as a single color unit, which means all pixels in a segment have the same color. We do this in HSV color space because experiments show that HSV space is better than using RGB space. A segment is represented as the mean of the HSV values in the segment, i.e., each segment is represented as a 3×1 vector. Feed this feature into SDV operator and we obtain the color-SDV saliency map.

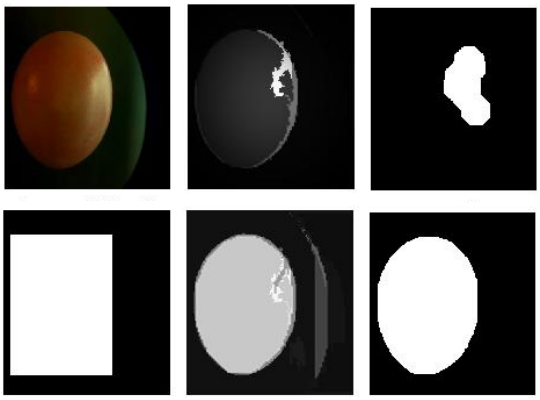

Fig. 1. Result example compared to SODS [6]. Left column from top to down: original image, ground truth bounding box; middle column from top to down: saliency map of SODS, proposed method; right column from top to down: binarized mask of SODS, proposed method

2.4. LBP texture map

LBP features are the same for the two operators. Firstly we calculate the $LBP^{riu2}_{8,1}$ coded image and $LBP^{riu2}_{16,2}$ coded image of the original image. For a segment S, we calculate the histogram of the two coded image, normalize them, and then concatenate them together as a the texture feature, which is a 28×1 vector. Feed this feature into the two operators and we obtain LBP-CSC saliency map and LBP-SDV saliency map.

3. SALIENCY MAP COMBINATION

After we obtain four saliency maps using color and LBP features, we use a globally and locally adaptive linear weighting method to combine them. Firstly, we compute the F-measure of each feature map on MSRA SOD database. Let the global weights be $Q = [q_1, q_2, \ldots, q_n]$ for n feature maps. We use an online learning method to update the global prior when processing each image in the learning set. The goodness of online learning is that it is efficient and easy to implement. According to the F-measure's, the global weights are updated as follows,

$$q_i^* = q_i + \lambda(F_i - q_i), i = 1, \ldots, n \tag{7}$$

where F_i is the normalized F-measure of feature map i for current image, q_i is the current weight for feature map i, and q_i^* is the new weight which will be used for the next image.

Then for each image, we compute the saliency index

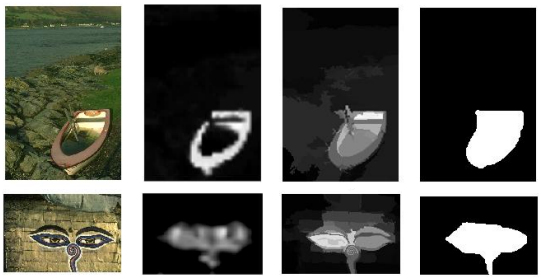

Fig. 2. Result examples on Berkeley segmentation database compared to [8]. From left to right: original image, saliency map of [8], saliency map of proposed method, binarized mask of proposed method

$SalIndex$ using following equations [8]:

$$SalMap_{Var} = \frac{\sum_{\mathbf{x}} \|\mathbf{x} - \bar{\mathbf{x}}\|^2 \cdot Sal_{\mathbf{x}}}{\sum_{\mathbf{x}} Sal_{\mathbf{x}}} \tag{8}$$

$$SalMap_{Con} = \frac{\sum_{\mathbf{p} \in ISal} \sum_{(x,y) \in N_{\mathbf{p}}} I_i(x,y)}{|ISal|} \tag{9}$$

$$SalIndex = \frac{SalMap_{Con}}{SalMap_{Var}} \tag{10}$$

where $\bar{\mathbf{x}}$ is the saliency value at location $\mathbf{x}$. $ISal$ is the set of salient pixels in the binarized saliency mask and $|ISal|$ is the cardinality. $I_i(x,y)$ is the indicator function denoting whether the pixel at location (x,y) is a salient pixel. $N_{\mathbf{p}}$ is the set of coordinates in the 8-point neighborhood of pixel $\mathbf{p}$.

The final weight w is computed as:

$$w_i = q_i \cdot SalIndex_i, i = 1, \ldots, n \tag{11}$$

The final saliency map is computed by linearly combining the n saliency maps using weight w, and then normalized to the interval $[0, 1]$.

4. EXPERIMENTS

To evaluate our method and compare it with existing methods, we conduct experiments on Microsoft Research Asia SOD database with 5000 images. Similar to [10], a ground truth bounding box (GTBB) is obtained by averaging the nine subjects' binary masks. We carry out the evaluation of the algorithms based on *Precision*, *Recall*, and *F-Measure*. *Precision* is calculated as ratio of the total saliency (sum of intensities in the saliency map) captured inside the GTBB to the total saliency computed for the image. *Recall* is calculated as the ratio of the total saliency captured inside the GTBB to the total saliency of the GTBB. *F-Measure* is the overall performance measurement and is computed as the weighted harmonic mean between the precision and recall values ([8]). It is defined as

$$\textit{F-Measure}_{\alpha} = \frac{(1+\alpha) \cdot \textit{Precision} \cdot \textit{Recall}}{\alpha \cdot \textit{Precision} + \textit{Recall}} \tag{12}$$

Table 1. Comparison of precision, recall and F-measure

	method of [8]	SODS [6]	Proposed
precision	0.68	0.80	0.80
recall	0.31	0.69	0.79
F-measure	0.53	0.77	0.80

where α is the parameter to decide the importance of precision over recall. To compare with [8] we also set $\alpha = 0.3$.

In our experiments, there are totally four saliency maps, i.e. Color-CSC, Color-SDV, LBP-CSC, LBP-SDV. To learn the global weights for each feature map, we do online learning on a subset of MSRA SOD database, which contains 500 images. The learned weights are $Q = [0.26, 0.27, 0.22, 0.25]$.

Table. 1 shows that our method is much better than [8] and [6], with 0.29 and 0.09 higher F-measure, respectively. As verified in Fig. 1, it is obvious that our method significantly outperforms [6], since their method fails to capture the salient object. In Fig. 2 the results are similar, but our saliency map are more accurate and complete. Fig. 3 verifies the effectiveness of multi-scale analysis. We can see that the first and the second scales fail to find the salient object accurately, but combining the three scales together we successfully capture the salient object.

Fig. 3. Multi-scale LBP-CSC saliency maps. Top row: CSC saliency maps on original size, 1/2 size, 1/4 size; Bottom row from left to right: Original image, Multi-scale CSC Saliency map, Binarized mask.

5. CONCLUSION

We propose a multi-scale framework for salient object detection based on segments. Using this framework salient object with different scales can be accurately detected. Besides, we propose a novel feature combination algorithm which incorporates both global prior and local adaptivity. We conduct experiments on Microsoft Research Asia SOD database, and our method has a significant gain over existing methods.

Besides of the good performance, our method is very flexible and open to new features. We propose two processors: CSC and SDV. One can choose any possible feature when using the two processors, such as Gabor or texton. Future work will be focused on more advanced feature integration technique and multi-scale fusion method.

6. REFERENCES

[1] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," *IJCV*, vol. 59, no. 2, pp. 167–181, Feb. 3 2004.

[2] L. Chen, X. Xie, X. Fan, W. Ma, H. Shang, and H. Zhou, "A visual attention mode for adapting images on small displays," *Technical report, Microsoft Research, Redmond, WA*, 2002.

[3] V. Navalpakkam and L. Itti, "An integrated model of top-down and bottom-up attention for optimizing detection speed," *CVPR*, pp. 2049–2056, 2006.

[4] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. Vol.20, no. 11, pp. 1254–1259, 1998.

[5] N. Bruce and J. Tsotsos, "Saliency based on information maximization," *NIPS*, pp. 155–162, 2005.

[6] Z. Liansheng, T. Ketan, Y. Nenghai, and Q. Yangchun, "Fast salient object detection based on segments," *ICMTMA*, vol. 1, pp. 469–472, 2009.

[7] Z. Wei, Q. M. J. Wu, W. Guanghui, and Y. Haibing, "An adaptive computational model for salient object detection," *Multimedia, IEEE Transactions on*, vol. 12, no. 4, pp. 300–316, 2010.

[8] V. Gopalakrishnan, H. Yiqun, and D. Rajan, "Salient region detection by modeling distributions of color and orientation," *Multimedia, IEEE Transactions on*, vol. 11, no. 5, pp. 892–905, 2009.

[9] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 971–987, 2002.

[10] T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H.-Y. Shum, "Learning to detect a salient object," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2007)*, pp. 1–8, June 17-22 2007.