MSGC: A NEW BOTTOM-UP MODEL FOR SALIENT OBJECT DETECTION

Zhi-Jie Wang[†,⋆], *Lizhuang Ma*[‡], *Xiao Lin*[⊣], *Xiabao Wu*[#]

[†]Sun Yat-Sen University [⋆]Guangdong Key Laboratory of Big Data Analysis & Processing
[‡]Shanghai Jiao Tong University [⊣]Shanghai Normal University [#] Shanghai Zhihuan Software Technology Co., Ltd

ABSTRACT

Saliency detection has been a hot topic in computer vision and image processing communities. Utilizing the global cues has been shown effective in saliency detection, whereas most of prior works mainly considered the *single-scale* segmentation when the global cues are employed. In this paper, we attempt to incorporate the *multi-scale global cues* (MSGC) for saliency detection. Achieving this proposal is interesting and also challenging (e.g., how to obtain appropriate foreground and background seeds; how to merge rough saliency results into the final saliency map efficiently). To alleviate various challenges, we present a solution that integrates three targeted techniques: (i) a self-adaptive approach for obtaining appropriate filter parameters; (ii) a cross-validation approach for selecting appropriate background and foreground seeds; and (iii) a weight-based approach for merging the rough saliency maps. Our solution is easy-to-understand and implement, but without loss of effectiveness. We have validated its competitiveness through widely used benchmark datasets.

Index Terms— Saliency detection; global cues; multi-scale segmentation

1. INTRODUCTION

Saliency detection is a classical problem in computer vision, and it has attracted much attention, due to its wide applications such as image segmentation, visual tracking, video prediction, face recognition [1, 2, 3]. In the existing literature there are two representative approaches: (i) the top-down approaches [4], which are task-driven; and (ii) the bottom-up approaches, which are data-driven [5, 6, 7, 8, 9]. These two types of approaches have their own advantages (e.g., bottom-up approaches can extract low-level features directly from the images, while top-down approaches can learn semantic content hidden in the images). In this paper, we restrict our attention to the *bottom-up approaches*. Some of prior works in this branch consider the local prior cues, i.e., utilizing either the prior cues from the background (e.g., [10]), or only the prior cues from the foreground (e.g., [11, 12]). Meanwhile, in this branch there are also many papers considering the *global prior cues*, i.e., utilizing the prior cues from both the background and foreground, see e.g., [13]. Utilizing the global prior cues has been shown more effective, especially for complicate images [14]. Although existing bottom-up approaches utilize the global prior cues, they mainly consider the single-scale segmentation. That is, they adopt only a single scale in terms of the number of superpixels, when they separate/segment an input image. Most single-scale segmentation based saliency detection algorithms are sensitive to the size of scale [15], since the sizes of objects (or targets) in images could be not the same, the single scale segmentation could not well fit in all images. See Fig. 1 for an illustration.

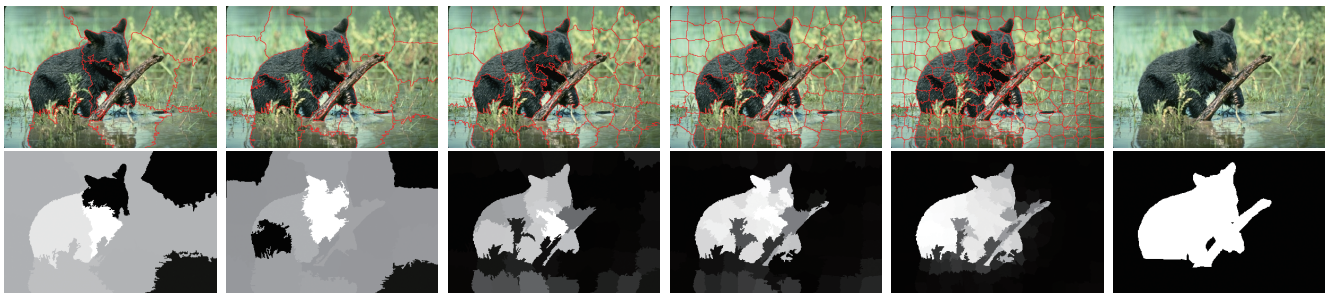
The multi-scale segmentation is complement to the single-scale segmentation. It adopts multiple scales, and so it allows users to obtain *more features* based on different scales [15, 16]. Yet, existing algorithms employing the multi-scale segmentation are either top-down approaches (e.g., [4]), or only use local prior cues (e.g., [15, 17]). To the best of our knowledge, in existing bottom-up approaches, few effort has been made on the *multi-scale segmentation* while considering the *global prior cues*. Motivated by these, this paper attempts to study the saliency detection problem, by incorporating *multi-scale global cues* (MSGC). To achieve this proposal, we present a solution, dubbed as the MSGC-based solution, that is composed of three phases (the details are covered in Section 2).

The main challenges to develop our solution are as follows. (1) Existing *multi-scale segmentation* based algorithms either fails to *flexibly* select the size of the scale [18], or fails to well process the texture and noise information [19]. A natural solution is to use the *bilateral filter*[1] [20] to smooth the image before segmenting the image. Yet, most of prior works apply the bilateral filter for only the case of single-scale segmentation (e.g., [21]). In our context, if one directly extends existing bilateral filter algorithms, it would be pretty tedious to adjust the "filter parameters" for different scales. Then,

∗ This work was supported by the NSFC (No. 61502220, 61472245, 61472453, 61572326, 61775139, U1401256, U1501252, U1611264, U1711261, and U1711262), and the STCSM Program (No. 16511101300). Email: wangzhij5@mail.sysu.edu.cn, ma-lz@cs.sjtu.edu.cn, lin6008@shnu.edu.cn, wuxiabao@hotmail.com.

[1]It uses two important components σ_d and σ_r to determine the final effect of the smoothed image. Specifically, σ_r is used to control smoothness: the larger the value is, the smoother the image shall be. σ_d is used to control the sharpness of the edges: the larger the value is, the more blurry the edges are.

978-1-5386-1737-3/18/$31.00



(a) (b) (c) (d) (e) (f)

Fig. 1. Examples of superpixels with different scales and their corresponding saliency maps: (a) t_1=10; (b) t_2=20; (c) t_3=50; (d) t_4=100; (e) t_5=200; (f) input image and ground-truth. Here t_i ($i \in [1, 2, 3, 4, 5]$) denotes the number of superpixels. Generally, the size of superpixels is larger, the number of superpixels in an image is less. In addition, different scales have different saliency effects.

how to assign the appropriate *filter parameters* for different scales? (2) It is extremely important to select the "appropriate" background and foreground seeds for generating the *rough* background-based (RBB) and foreground-based (RFB) saliency maps respectively [13, 14]. Existing algorithms either cannot well process complicate images [12, 13], or fail to control some threshold parameters easily [14, 22]. Then, how to obtain "appropriate" foreground and background seeds easily and effectively? (3) To merge the RBB and RFB saliency maps, an easily brought to mind method is to compute the average value of all *rough* saliency results. This method, however, could produce pretty poor results, since it ignores some important natures (e.g., different segmentation scales may have different effects). Another potential method is to compute the cross-product of the rough foreground and background maps. This approach also easily produces poor results, since the cross-product operation, to some extent, weakens the difference between the salient object region and background region. Then, how to merge all rough saliency maps such that the final saliency result is with the good quality?

To attack the challenges, our MSGC-based solution integrates three targeted techniques. (1) A self-adaptive approach for obtaining appropriate filter parameters. (2) A cross-validation approach for selecting appropriate background and foreground seeds. (3) A weight-based approach for merging the rough saliency maps. Viewed from a macro perspective, similar to many saliency detection methods in the literature, the proposed solution also partially inherits several nice proposals such as manifold ranking, utilizing foreground and background priors, whereas we advance existing results from various aspects. Our main contributions can be summarized as follows. First, we suggest the use of multi-scale segmentation when considering the global cues. Second, we propose a three-phase solution framework that integrates three targeted strategies addressing various challenges. Third, we verify the superiorities of our model through extensive experiments.

2. PROPOSED SALIENCY DETECTION MODEL

The general framework of our solution consists of three main phases: (1) *Segmenting the image based on multi-scales*. To achieve better segmentation effects, our solution employs a bilateral filter to smooth the image, before segmenting the image. In our context, the filter parameters are not easy-to-modulate, we develop a *self-adaptive* approach. The difficulty in developing it is to construct a relationship between the smoothness and filter parameters. (2) *Choosing prior cues (a.k.a., seeds) from background and foreground*. To select useful background/foreground seeds, we propose a new approach that employs an idea "*cross-validation*". Note that, these prior cues obtained can be immediately used to generate RBB and RFB saliency maps, using the *manifold ranking technique* [13]. (3) *Merging the rough saliency maps obtained before,* in order to get the final saliency result with a good quality. To achieve it, we suggest a *weight-based approach*, in which two novel concepts "scale weight" and "seed weight" are proposed.

2.1. Filter parameters selection

Observe that, preserving edges is urgently needed for subsequent segmentation operation; this allows us to set conservatively σ_d (cf. Footnote 1) to a relatively small value, so as to the "smoothed" image can still have enough sharpness of the edges. In our paper, σ_d is set to 0.2, unless stated otherwise. However, it remains unclear on how to set an appropriate value for σ_r. To attack the challenge, we propose a *self-adaptive approach* that can assist us to flexibly and automatically choose appropriate filter parameters for different scales. The rationale behind our approach is to construct an objective function by incorporating pixels' color differences and filter parameters together, and then to solve the function using a simple algorithm. The intuition for constructing the objective function is that, if the smoothness is insufficient, then the color differences of pixels (located in the same superpixel) is large, implying that the segmentation effect is not good.

Let T denote a *set* consisting of $|T|$ integers, and let each element (i.e., integer) $t_i \in T$ denote the number of superpixels when we separate the image. Assume, without loss of generality, that the image is to be segmented using a scale t_i. That

is, there would be t_i superpixels in the image. For a superpixel sp_j, let k be the number of pixels in it, $c_{i,j}$ (=$\{l, a, b\}^T$) be a pixel p_i's feature vector in terms of CIELab color space, and $\overline{c(j)}$ (=$\frac{1}{k}\sum c_{i,j}$) be the average feature vector of all $c_{i,j}$ in the corresponding superpixel sp_j.

For the jth superpixel, one can roughly measure the "local" smoothness in the superpixel region by computing the sum of all k pixels' color differences, i.e., $\sum_{i=1}^{k} \|c_{i,j} - \overline{c(j)}\|$. For all superpixels in the image, a "global" smoothness, denoted by S, can be measured as

$$S = \sum_{j=1}^{t_i} \sum_{i=1}^{k} \|c_{i,j} - \overline{c(j)}\| \tag{1}$$

Then, one can obtain the following (based on the intuition mentioned earlier):

$$F = \text{argmin}_{\sigma_r \in \mathbb{R}}(S + c\sigma_r) \tag{2}$$

where c is a constant real number, which is used to keep the two components (i.e., S and σ_r) in the same order of magnitude. In our paper, c is set to 100, unless stated otherwise. For the latter parts (i.e., $c\sigma_r$), one can essentially view them as the penalty factors, which are used to alleviate a too large global smoothness. It is easily verified that the equation above essentially characterizes an optimization problem. One can solve the above optimization problem trivially by executing a simple method below. To understand this method, it could be better to explain the following important observation: "S is inversely proportional to σ_r". This is because the larger σ_r is, the smoother the image shall be. In this case, the difference between $c_{i,j}$ and $\overline{c(j)}$ shall be smaller. Naturally, S shall be smaller (by Eq. 1). Specifically, our method first sets σ_r as a small value, and then increases it gradually, and it finally terminates the iteration when the value of F turns larger. This way, one can obtain the appropriate value for the filter parameter σ_r at scale t_i. (Clearly, given the "multi-scale" set T, one can apply the above strategy to determine the value of σ_r at any other scale $t_j \in T$.)

We may need to emphasize that, the difficulty in developing the self-adaptive approach could be not to solve the optimization problem described in Eq. 2. Essentially, the difficulty is to construct a relationship between the smoothness and filter parameters.

2.2. Background/foreground seeds selection

The rationale behind our cross-validation approach is utilizing a strategy to remove part of the "initial" background and foreground seeds, which are obtained based on existing techniques. Our strategy is based an observation — the differences between the background and foreground are usually larger than the differences between *internal regions* in background (or foreground). Particularly, the operation for removing part of initial background (resp., foreground) seeds *employs* the feature information from initial foreground seeds (instead, "refined" background seeds).

Specifically, for each image we first use the *objectness likelihood map* (OLM) technique [14] to get the "initial" foreground seeds, and conservatively use the boundary of the image as initial background seeds. Assume, without loss of generality, that m initial foreground seeds and n initial background seeds are obtained. Denote by is_b^j the jth initial background seed, and is_f^i the ith initial foreground seed.

For each initial background/foreground seed (e.g., is_b^j), we utilize two types of feature information: (i) color, i.e., $\{l, a, b\}$ in CIELab color space; and (ii) location, i.e., $\{x, y\}^T$ in Euclidean space. Denote by c_b^j (resp., c_f^i) the "color" feature vector of is_b^j (resp., is_f^i), and by l_b^j (resp., l_f^i) the "location" feature vector of is_b^j (resp., is_f^i). Let $D_{(is_b^j, is_f)}$ be the sum of the differences from is_b^j to each initial foreground seed. It can be computed as

$$D_{(is_b^j, is_f)} = \sum_{i=1}^{m} (\|c_b^j - c_f^i\| + \theta\|l_b^j - l_f^i\|) \tag{3}$$

where θ is a parameter used to adjust the weight of the location information. In our paper, it is set to 0.5, unless stated otherwise. For clarity, we dub the value of $D_{(is_b^j, is_f)}$ as the *credit score* of the background seed is_b^j. Naturally, we can get n credit scores (using the method above), since there are n initial background seeds. That is, we shall obtain a *set* with n real numbers: $\{D_{(is_b^1, is_f)}, D_{(is_b^2, is_f)}, ..., D_{(is_b^n, is_f)}\}$.

Next, we are ready to remove a part of initial background seeds whose *credit scores are small*. It is based on the following intuition — if an initial background seed is with a small credit score, it is usually more like to share the high similarity with the foreground. To this step, one could ask: how much initial background seeds should be removed? Specifically, we do as follows. We first sort the credit scores in ascending order. For clarity, we renumber the sorted credit scores as cs_1, cs_2, ..., cs_n, such that for any $k \in [1, n-1]$, $cs_k \leq cs_{k+1}$. Then, for each $k \in [1, n-1]$ we compute the value of $cs_{k+1} - cs_k$. This way, $n-1$ values are generated. Without loss of generality, assume that the maximum value among all the $n-1$ values is obtained when $k = \gamma$. We set the credit score cs_γ to be a "dividing line". Finally, for any initial background seed whose credit score is less than or equal to cs_γ, we remove it. This way, we obtain the "refined" background seeds.

On the other hand, for m initial foreground seeds, we can process them using the method similar to the above. A minor difference is that we employ the feature information from "refined" background seeds, instead of the one from "initial" background seeds. Notice that, although selecting seeds based on color and geometric distances was extensively used in the literature, the cross-validation strategy is novel. It is not covered in the domain of saliency detection, and is effective to improve the quality of saliency results, as demonstrated later.

2.3. RBB and RFB saliency maps fusion

This section suggests a *weight-based* approach for merging the RBB and RFB saliency maps. Our general idea is to first compute two weights (one is known as the *scale weight*, which reflects the effects of different segmentation scales; another is known as the *seed weight*, which reflects the similarity from a pixel p to foreground/background seeds), and then to merge the RBB and RFB saliency maps, by incorporating these two weights. The intuitions behind our idea are: (i) for a given image, different segmentation scales usually incur different effects, and if a scale has the better segmentation effect, it usually has the larger contribution to generate a good quality saliency map; and (ii) if a pixel is more similar to the foreground (resp., background) seeds, it usually has a higher probability to be similar to the corresponding pixel in the RFB (resp., RBB) saliency map.

Recall that we have obtained (i) $|T|$ images that have been segmented (Phase 1); and (ii) $|T|$ RBB and $|T|$ RFB saliency maps (Phase 2). Our approach shall take full use of these available information. Note that, in the following discussion, we focus on a single pixel. Other pixels in the final saliency map can be obtained one by one, using the same method introduced below.

Specifically, let $c(p) = \{l, a, b\}^T$, and $\sum_p$ be all pixels in a superpixel containing p. First, for each scale $t_i \in T$, we compute the pixel p's *scale weight*, denoted by $scw_i(p)$, as follows.

$$scw_i(p) = \left\| c(p) - \overline{c_i(p)} + \epsilon \right\|^{-1} \tag{4}$$

where $\overline{c_i(p)}$ is the average feature vector of $\sum_p \{l, a, b\}^T$, and ϵ is an arbitrary small constant, which is used to avoid the base to be zero. (As a remark, the computation above exploits the results obtained in Phase 1.) The equation above directly reflects the similarity, in terms of color, between a pixel p and the superpixel containing p. Essentially, it also reflects indirectly the segmentation effect at the scale t_i, viewed from another perspective. One can view $scw_i(p)$ as a bridge between the color similarity and the segmentation effect.

Assume, without loss of generality, that we have obtained m' (resp., n') refined foreground (resp., background) seeds in Phase 2. Denote by $\sum_p^{m'}$ (resp., $\sum_p^{n'}$) all the pixels in the m' (resp., n') refined foreground (resp., background) seeds. Next, for each scale $t_i \in T$, we compute the pixel p's *seed weight*, denoted by $sew_i(p)$, as follows.

$$sew_i(p) = \frac{\left\| c(p) - \overline{c_i^{m'}(p)} \right\|}{\left\| c(p) - \overline{c_i^{n'}(p)} \right\| + \left\| c(p) - \overline{c_i^{m'}(p)} \right\|} \tag{5}$$

where $\overline{c_i^{m'}(p)}$ (resp., $\overline{c_i^{n'}(p)}$) denotes the average feature vector of $\sum_p^{m'} \{l, a, b\}^T$ (resp., $\sum_p^{n'} \{l, a, b\}^T$).

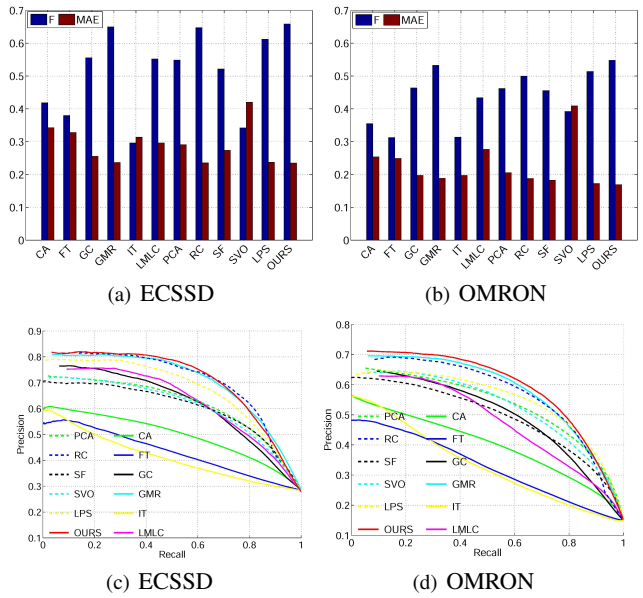
Finally, we obtain the final saliency result for pixel p by incorporating these two weights and the information in $|T|$ RBB and $|T|$ RFB saliency maps. (For short, we write $scw_i(p)$ and $sew_i(p)$ as ω_1 and ω_2, respectively.) Let $V(p)$ be pixel p's result in the final saliency map. It is computed as

$$V(p) = \frac{\sum_{i=1}^{|T|} \omega_1 \times [(1-\omega_2) \times S_f^i(p) + \omega_2 \times S_b^i(p)]}{\sum_{i=1}^{|T|} \omega_1} \tag{6}$$

where $S_b^i(p)$ (resp., $S_f^i(p)$) denotes the corresponding pixel's information value in the ith RBB (resp., RFB) saliency map.

3. EXPERIMENTS AND ANALYSIS

We evaluate our method using two representative datasets: (1) *ECSSD* [23], which includes a lot of images with complicated background; and (2) *OMRON* [13], which is a more challenging datasets containing 5168 images with complex backgrounds and texture structures, and the locations and sizes of objects/targets in images are diversified. We compare our algorithm against classic and some state-of-the art methods, including IT [9], FT [5], CA [8], SVO [6], RC [7], SF [24], PCA [25], LMLC [11], GC [26], GMR [13], and LPS [14].



(a) ECSSD (b) OMRON

(c) ECSSD (d) OMRON

Fig. 2. The results are obtained based on various methods over two representative datesets.

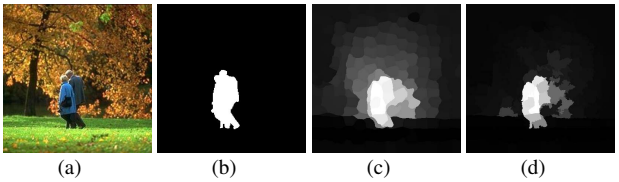
In our experiments, we use typical evaluation metrics. (1) *Precision and recall.* Denote by v_p and v_r the precision and recall values, respectively. Similar to [12, 22], we obtain P-R curve by binarizing the saliency map, using thresholds in [1,255]. (2) *F-measure.* It is computed as $F = \frac{(1+\eta^2) \times v_p \times v_r}{\eta^2 \times v_p + v_r}$, where η is used to control the ratio of precision and recall. (3) *Mean absolute error.* Denote by E_{ma} the mean absolute error (MAE). It is computed as $E_{ma} = \frac{1}{N_p} \sum_{i=1}^{N_p} |S(p_i) - GT(p_i)|$, where N_p denotes the number of all pixels in the image; $S(p_i)$ and $GT(p_i)$ denote the information of the ith pixel from the saliency map and from the ground truth, respectively. Following prior works [22, 12], the parameter η^2 is set to 0.3.

(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 3. Examples of saliency detection results: (a) *Input image*; (b) GC; (c) PCA; (d) LPS; (e) LMLC; (f) MR ; (g) *Ours*; (h) *Ground truth*. Our solution obtains the saliency maps close to the ground truth.

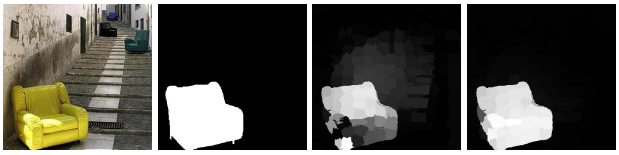
From Fig. 2(a) it can be seen that, our proposed method has the larger F-measure value than other methods, demonstrating that it performs well when the dataset contains images with complex background. On the other hand, for the more challenging dataset, one can see from Fig. 2(b) that, our proposed method also performs very well, compared with other methods. Again, from Figs. 2(a) and 2(b), it can been seen that, our MAE value is smaller than the ones of other methods, regardless of ECSSD or OMRON dataset. These results demonstrate the superiority of our proposed method, from another perspective. As mentioned earlier, the P-R curve is drawn by the value of the precision and recall; it can more directly reflect the performance of an algorithm [27, 12]. We can see from Figs. 2(c) and 2(d) that, for each of these two datasets the P-R curve of our algorithm (plotted as the red curve) dominates the ones of other methods. This further validates the effectiveness of our method.

Fig. 3 shows a few saliency maps generated by our solution and classic approaches. From this figure we can see that, for the scenarios where the images are with complex backgrounds and texture structures, our solution can accurately highlight the salient object, and can preserve well the edges of the object; see e.g., the first and second rows in Fig. 3. Furthermore, our solution performs better than other methods



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

Fig. 4. Effectiveness of self-adaptive technique. (a) Input image; (b) ground truth; (c) choose the value of σ_r randomly; (d) self-adaptive approach.



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

Fig. 5. Effectiveness of cross validation technique. (a) Input image; (b) ground truth; (c) without the cross-validation; (d) our result.

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

Fig. 6. Effectiveness of weight-based technique. (a) Input image; (b) fusion by computing the average value; (c) fusion by computing the cross-product; (d) our method.

when the salient object is near to the boundary of the image; see e.g., the last several rows in Fig. 3.

Besides, we also test the effectiveness of the proposed techniques, respectively. To evaluate the effectiveness of each technique, we replace it with the traditional approach, and then compare it with our solution. Fig. 4 shows the saliency results generated using a challenging input image. Compared with the baseline method (i.e., choosing the value of σ_r randomly), we can see that there are two major differences: (i) the saliency map generated by our method contains less background noises; and (ii) the saliency map generated by our method is much more close to the ground truth. These differences essentially reflect the effectiveness of the self-adaptive strategy. Fig. 5 shows the comparison result in terms of cross-validation technique. Compared with the baseline method, we can see that (i) the salient object's location has been obtained more exact; and (ii) the background noises have been well suppressed. These evidences demonstrate that the cross-validate scheme is effective.

Fig. 6 compares the result of three methods (i.e., fusion by computing the average value, by computing the cross-product, and by our method, respectively). We can see that the saliency map generated by the first baseline method has two major features: the foreground object contains a lot of background noises; meanwhile, the background is not clear. On the other hand, consider the saliency map generated by the second baseline method. The background in the saliency map is clear, while the salient object is not well highlighted. In contrast, compared with these two baseline methods, the saliency map generated by our method is clearer, regardless of the background or the salient object. Particularly, the salient object is well highlighted. These results validate the effectiveness of the weight-based strategy.

Our method exposes many advantages, while it also bears

a major limitation. That is, its running time is somewhat long, compared with other bottom-up methods. Specifically, by using a machine with Intel(R) Core(TM) i3 CPU @2.40 GHz and 3GB RAM, the average running time of each image on the ASD dataset [5] is 18.832 seconds, while other methods consume less time (e.g., GMR [13] uses 2.461 seconds, MS [12] uses 13.615 seconds, and PCA [25] uses 16.342 seconds). Note that, reducing the running time could be another independent work and somewhat challenging; in the future, we attempt to further optimize our solution and compare with more algorithms.

4. CONCLUSION

In this paper, we proposed a novel bottom-up saliency detection model. The central idea is to use multi-scale segmentation while considering both foreground and background priors. It is interesting and challenging to achieve this proposal. To this end, our solution integrates three targeted techniques tailored for different challenges. Experimental results benchmark datasets validated the effectiveness and competitiveness of our model. Also, we examined the major limitation, pointing out the future research direction.

5. REFERENCES

[1] J. Zhang and S. Sclaroff, "Exploiting surroundedness for saliency detection: A boolean map approach," *TPAMI*, vol. 38, no. 5, pp. 889–902, 2016.

[2] Z. Wu, L. Su, Q. Huang, B. Wu, J. Li, and G. Li, "Video saliency prediction with optimized optical flow and gravity center bias," in *ICME*, 2016, pp. 1–6.

[3] Y. Ren, M. Xu, R. Pan, and Z. Wang, "Learning gaussian mixture model for saliency detection on face images," in *ICME*, 2015, pp. 1–6.

[4] X. Li, F. Yang, H. Cheng, J. Chen, Y. Guo, and L. Chen, "Multi-scale cascade network for salient object detection," in *ACM Multimedia*, 2017, pp. 439–447.

[5] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *CVPR*, 2009, pp. 1597–1604.

[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*, 2011, pp. 914–921.

[7] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu, "Global contrast based salient region detection," *TPAMI*, vol. 37, no. 3, pp. 569–582, 2015.

[8] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," *TPAMI*, vol. 34, no. 10, pp. 1915–1926, 2012.

[9] L. Itti, C. Koch, E. Niebur *et al.*, "A model of saliency-based visual attention for rapid scene analysis," *TPAMI*, vol. 20, no. 11, pp. 1254–1259, 1998.

[10] B. Jiang, L. Zhang, H. Lu, C. Yang, and M.-H. Yang, "Saliency detection via absorbing markov chain," in *ICCV*, 2013, pp. 1665–1672.

[11] Y. Xie, H. Lu, and M.-H. Yang, "Bayesian saliency via low and mid level cues," *TIP*, vol. 22, no. 5, pp. 1689–1698, 2013.

[12] N. Tong, H. Lu, X. Ruan, and M.-H. Yang, "Salient object detection via bootstrap learning," in *CVPR*, 2015, pp. 1884–1892.

[13] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *CVPR*, 2013, pp. 3166–3173.

[14] H. Li, H. Lu, Z. Lin, X. Shen, and B. Price, "Inner and inter label propagation: salient object detection in the wild," *TIP*, vol. 24, no. 10, pp. 3176–3186, 2015.

[15] N. Tong, H. Lu, L. Zhang, and X. Ruan, "Saliency detection with multi-scale superpixels," *IEEE Signal Processing Letters*, vol. 21, no. 9, pp. 1035–1039, 2014.

[16] R. Huang, W. Feng, and J. Sun, "Saliency and co-saliency detection by low-rank multiscale fusion," in *ICME*, 2015, pp. 1–6.

[17] X. Hu, W. Yang, F. Zhou, and Q. Liao, "Saliency detection based on integration of central bias, reweighting and multi-scale for superpixels," in *ICASSP*, 2016, pp. 1946–1950.

[18] X. Liu, Q. Xu, J. Ma, H. Jin, and Y. Zhang, "Mslrr: A unified multiscale low-rank representation for image segmentation," *TIP*, vol. 23, no. 5, pp. 2159–2167, 2014.

[19] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *TPAMI*, vol. 34, no. 11, pp. 2274–2282, 2012.

[20] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *ICCV*, 1998, pp. 839–846.

[21] W. C. Wong and A. C. Chung, "Bayesian image segmentation using local iso-intensity structural orientation," *TIP*, vol. 14, no. 10, pp. 1512–1523, 2005.

[22] Y. Qin, H. Lu, Y. Xu, and H. Wang, "Saliency detection via cellular automata," in *CVPR*, 2015, pp. 110–119.

[23] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *CVPR*, 2013, pp. 1155–1162.

[24] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung, "Saliency filters: Contrast based filtering for salient region detection," in *CVPR*, 2012, pp. 733–740.

[25] R. Margolin, A. Tal, and L. Zelnik-Manor, "What makes a patch distinct?" in *CVPR*, 2013, pp. 1139–1146.

[26] M.-M. Cheng, J. Warrell, W.-Y. Lin, S. Zheng, V. Vineet, and N. Crook, "Efficient salient region detection with soft image abstraction," in *ICCV*, 2013, pp. 1529–1536.

[27] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," in *CVPR*, 2014, pp. 2814–2821.