Saliency Detection via Multi-Scale Global Cues

Xiao Lin, Zhi-Jie Wang, Lizhuang Ma, and Xiabao Wu

***Abstract*—The saliency detection technologies are very useful to analyze and extract important information from given multimedia data, and have already been extensively used in many multimedia applications. Past studies have revealed that utilizing the global cues is effective in saliency detection. Nevertheless, 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 for saliency detection problem. Achieving this proposal is interesting and also challenging (e.g., how to obtain appropriate foreground and background seeds effectively? how to merge rough saliency results into the final saliency map efficiently?). To alleviate the challenges, we present a three-phase solution that integrates several targeted strategies: (i) a self-adaptive strategy for obtaining appropriate filter parameters; (ii) a cross-validation scheme 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. Extensive experimental results based on benchmark datasets demonstrate the feasibility and competitiveness of our proposed solution.**

***Index Terms*—Saliency region; global prior; image smoothing and segmentation**

I. Introduction

With the rapid increase in multimedia services, a huge amount of multimedia data is created everyday, e.g., speech, text, image, video, and graphics. Therefore, it is drawing much attention to analyze and extract important information from given multimedia data [1]–[3]. The saliency detection technologies, which exploit the most important areas for natural scenes, are very useful in practice, and have already been widely used in many multimedia applications such as coding, retrieval, tracking, adaptation, classification, segmentation, and streaming [4]–[10].

Generally speaking, saliency detection is to extract the target area, suppress the background noise, and represent the final saliency map in an appropriate way, such as the grey mode with normalized values ranging from 0 to 1 [11], [12]. In the existing literature there are two representative approaches: (i) the top-down approaches, which generate the final saliency map by utilizing the high-level prior knowledge [13]–[20]; and (ii) the bottom-up approaches, which are data-driven, and generate the final saliency map by directly simulating the underlying visual information [1]–[3], [8], [11], [12], [21]–[36]. These two types of approaches have their own advantages (e.g., bottom-up approaches are simple, easy-to-understand, and can extract low-level features directly from the images, while top-down approaches can learn semantic content hidden in the image) [11], [20], [23]. In this paper, we restrict our attention on the *bottom-up approaches*.

Some excellent works in this branch consider the local prior cues, i.e., utilizing either the prior cues from the background [37], or only the prior cues from the foreground [34], [38]. Here prior cues are referred as to prior knowledge such as contrast, spatial information, etc [25], [34], [35], [37], [38]. Meanwhile, there are also many classic papers considering the global prior cues, i.e., employing the prior cues from both the background and foreground [25], [35], [39], [40]. Utilizing the global prior cues has been shown more effective, especially for complicate images [25], [35]. Although existing works employ 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* [41], when they separate/segment an input image. Most single-scale segmentation based saliency detection algorithms are sensitive to the size of scale [42], as the sizes of objects (or targets) in images could be not the same. In other words, the single scale segmentation could not well fit in all images. See Fig. 1 for an illustration. Besides the sizes of objects in images, some other elements such as background, color and contrast in images might be also sensitive to the size of scale. See Fig. 2 for an illustration. All these phenomena imply that, using the single-scale segmentation could be not very appropriate for saliency detection.

The *multi-scale segmentation* [43] is "complement to" the single-scale segmentation. That is, it adopts *multiple scales* in terms of the number of superpixels (when they separate/segment an input image), and so it allows users to obtain more features based on different scales [11]. Existing approaches that employ multi-scale segmentation are either top-down approaches (e.g., [16], [44]–[46]), or only use local prior cues (e.g., [43], [47]). To the best of our knowledge, in existing bottom-up approaches, few attention has been made on the *multi-scale segmentation* when the *global prior cues* are used. (A more comprehensive review will be given in Section II.) Motivated by these, this paper attempts to study the saliency detection problem, by incorporating *multi-scale global cues*. Our main goal is to examine (i) the feasibility and challenges (to incorporate multi-scale global cues), and (ii) the final effect (if feasible) or the reasons (if infeasible).

To achieve the proposal above, we suggest a solution consisting of three main phases: (1) it first segments the image based on multiple scales; and then (2) extracts the prior cues from background and foreground to generate *rough*

X. Lin is with the Department of Computer Science, Shanghai Normal University, Shanghai, China (Lin6008@126.com).

Z.-J. Wang is with the Guangdong Key Laboratory of Big Data Analysis and Processing, School of Data and Computer Science, Sun Yat-Sen University, Guangzhou, China. (wangzhij5@mail.sysu.edu.cn).

L. Ma is with (i) the Department of Computer Science & Engineering, Shanghai Jiao Tong University, Shanghai, China, and (ii) the School of Computer Science and Software Engineering, East China Normal University, Shanghai, China (ma-lz@cs.sjtu.edu.cn).

X. Wu is with the Shanghai Zhihuan Software Technology Co., Ltd. (wuxiaobao@hotmail.com).

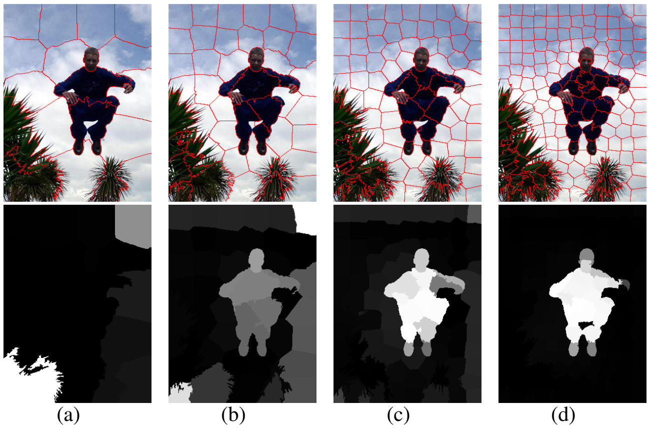


Fig. 1. Examples of superpixels with different scales and their corresponding salient maps: (a)t_1=20; (b) t_2=50; (c) t_3=100; (d) t_4=200, where t_i ($i \in [1, 2, 3, 4]$) denotes the number of superpixels. In general, the larger the size of the superpixels is, the fewer the number of superpixels is.

background-based and *foreground-based* (known as RBB and RFB) saliency maps respectively; and finally (3) merges the available RBB and RFB saliency maps, obtaining the final saliency result. To develop our solution above, the main challenges needing to be addressed are threefold (a more detailed analysis is covered in Section IV): (i) how to assign appropriate filter parameters for different scales? (ii) how to obtain appropriate foreground and background seeds easily and effectively? and (iii) how to merge all the rough saliency maps such that the final saliency result is with the good quality? To alleviate these challenges, three targeted strategies are presented in Sections VI $\sim$ VIII: (1) a self-adaptive strategy for obtaining appropriate filter parameters; (2) a cross-validation scheme for selecting appropriate background and foreground seeds; and (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, and objectness likeness map, whereas we advance existing results from various aspects. To summarize, our main contributions are as follows.

- We propose a novel bottom-up model for the saliency detection problem. To our knowledge, this is the first bottom-up model that suggests the use of multi-scale segmentation and global cues together.
- We develop a self-adaptive strategy that allows us to flexibly and automatically choose the appropriate smoothness for different scales.
- We develop a cross-validation scheme that can select the useful foreground/background seeds effectively.
- We develop a weight-based strategy to fusion rough saliency results. Our strategy takes full consideration into the effects of different scales and also the similarity from the pixel to foreground/background seeds.

We provide the rigorous theoretical analysis on the computational complexity of our algorithm (Section IX). Also, we examine the feasibility of our proposed solution through extensive experiments, and verify its superiorities by comparing against classic and state-of-the-art algorithms (Section X). In the next section, we review prior works most related to ours, followed by introducing some preliminaries that will be used in the remainder of the paper (Section III).

II. Related Work

As stated before, in this paper our focus is on the bottom-up approaches, and so we mainly review prior works in this branch. (Nevertheless, some representative works in other branches could be also mentioned, when they bear some similarities with our work.) In general, the bottom-up approaches can be classified into two categories according to whether the prior cues are used. In existing literature, some works did not employ the prior cues (e.g., [11], [12], [23], [48], [49]), while most of prior works (e.g., [25], [34], [35], [37], [38]) incorporated the prior cues. Similar to the latter, our proposed solution also employs the prior cues. Next, we go into more details about the previous works in this category.

▷ *Local prior cues based model.* For methods in this model, their common characteristic is to use only the local prior cues [34], [37], [38], [43], [47]. In other words, they utilized either the prior cues from the background [37], [50], or only the prior cues from the foreground [34], [38], [43]. For example, in [37] the Markov chain technique was employed to achieve better boundary prior cues (as the background seeds). In addition, Wang *et al.* [50] proposed a background-driven salient object detection (BD-SOD) method to more comprehensively exploit the background prior. On the other hand, the authors in [34] proposed using the convex hull prior cues (as the foreground seeds). Moreover, in [38] the center-prior (as the foreground seeds) was employed. Hu *et al.* [47] used the central bias and the reweighting of the salient regions in the convex hull to guide the prior map. Compared with these nice works, the major difference is that we use the global prior cues, instead of local prior cues. Among these works, the ones most similar to our could be [43], [47], in which they also used the multi-scale segmentation. Yet, these works are different from ours. For example, the method proposed in [43] used the average value-based method to merge the rough saliency results, while our work presents a more competitive approach; again, our solution considers the global, instead of the local prior cues (which was employed in these papers). Nevertheless, we would like to point out that, besides these works, there are some excellent *top-down approaches* such as convolutional neural network based solutions that also consider multi-scale segmentation [16], [44]–[46]. Note that, as stated earlier, the focus of this paper is on the bottom-up approaches, instead of top-down approaches. It is not hard to understand that their works are different from ours.

▷ *Global prior cues based model.* A common characteristic of the methods in this category is to use the global prior cues [25], [35], [39], [40]. That is, they choose the prior cues from both the background and foreground. For example, Li *et al* [25] put forward a unified approach to incorporate low-level features and the objectness measure for saliency detection via label propagation, where labels are extracted from the most certain background and object regions. Wu *et al.* employed the Bayesian framework to classify each pixel into salient

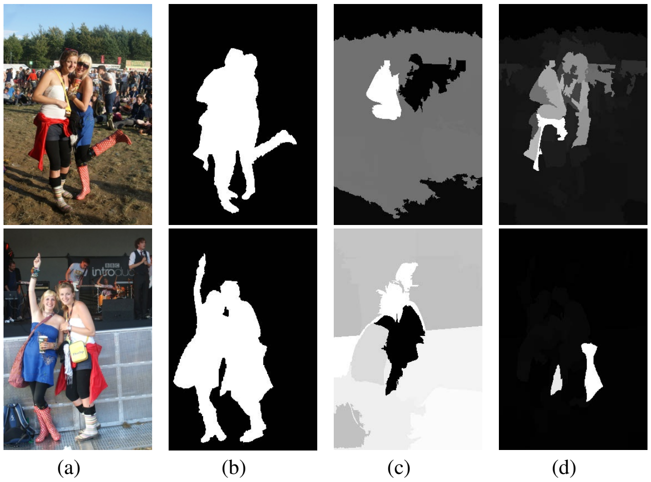


Fig. 2. Example for explaining other elements: (a) input images; (b) ground-truth; (c) saliency maps using $t_i = 10$; (d) saliency maps using $t_i = 100$. Although the sizes of objects in these two images are almost the same, their saliency maps are significantly different even if we use the same scale; see the 3rd (or 4th) column.

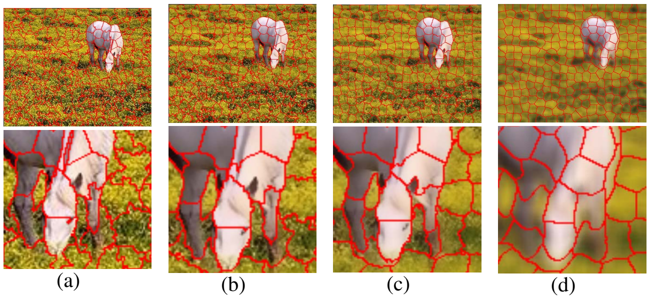


Fig. 3. Illustration of smoothing and segmentation. (a) segmentation without smoothing; (b) segmentation with a slight smoothing; (c) segmentation with an appropriate smoothing; (d) segmentation with an over-large smoothing. Note that, the bottom four images are the partial enlarged drawings of the top four images, respectively.

object or background object. Wang *et al.* [40] employed two complementary correspondence strategies: a global matching scheme based on scene-level analysis and a local matching scheme based on patch-level inference, and also introduced two refinement measures to further refine the saliency maps. In existing literature, methods in this category mainly involve the use of the single-scale segmentation. In other words, they separate the image according to a single scale. Compared with these distinguished works, the concern of our work is the use of the multi-scale segmentation. Among these works, the one most similar to ours is [35]. This is because both of two papers utilize the global prior cues and the manifold ranking technique. Yet, our method is different from that work in several points at least: (i) we select background and foreground seeds by the *cross-validation* strategy, which is a novel strategy and is not covered in the domain of saliency detection; (ii) we consider the multi-scale segmentation, instead of the single-scale segmentation that was used in their paper; and also (iii) selecting appropriate filter parameters and merging multiple rough saliency maps are not covered in that work. We remark that this article is a full version of the preliminary work [51]. In the article, we made the following extra contributions: (i) we give a more comprehensive review on previous works; (ii) we provide more insights into our algorithms; (iii) we give the rigorous theoretical analysis for our proposed algorithms; and (iv) we conduct more comprehensive experimental comparisons.

III. Preliminaries

For ease of understanding the rest of the paper, we here *review* some necessary concepts [25], [35], [52] , which will be used extensively later.

▷ *Manifold ranking.* It is a saliency calculation method originated from graph theory [35]. Its general steps are as follows. It first constructs an affinity matrix $\mathbb{W}$ by checking the relationship between the superpixel-level regions, and finds a series of seed regions Y, based on the prior knowledge (e.g., the boundary and/or center prior). Then, it computes a value f, between the seed region and other regions, using the following equation:

$$f = (D - \alpha\mathbb{W})^{-1} * Y \tag{1}$$

where $\mathbb{W} = \{w_{i,j}\}_{n\times n}$, $D = diag\{d_{11}, \dots, d_{nn}\}$, $d_{ii} = \sum_j w_{i,j}$. Finally, it obtains the saliency result by normalizing the values of f.

▷ *Bilateral filter.* Its basic principle is to consider both the *spatial relationship* and the *color similarity* between pixels, when one attempts to smooth an image while preserving edges (i.e., the contours of objects/targets in the image) [52]. The bilateral filter can be described formally as follows [52].

$$h(x) = k^{-1}(x)\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(\xi)c(\xi,x)s(f(\xi),f(x))d_\xi \tag{2}$$

where $k(x) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} c(\xi,x)s(f(\xi),f(x))d_\xi$ is used for normalization; x and ξ denote the neighborhood center and a nearby point, respectively; $c(\xi,x)$ and $s(f(\xi),f(x))$ are usually computed as

$$\begin{cases} c(\xi,x) = e^{-\frac{1}{2}(\frac{\|\xi,x\|}{\sigma_d})^2} & \text{(3a)} \\ s(f(\xi),f(x)) = e^{-\frac{1}{2}(\frac{\|f(\xi),f(x)\|}{\sigma_r})^2} & \text{(3b)} \end{cases}$$

where the parameters σ_d and σ_r are two important components, which 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; and σ_d is used to control the sharpness of the edges — the larger the value is, the more blurry the edges are.

▷ *Objectness likelihood map.* The objectness likelihood map (OLM) is used to describe the probability of the pixel belonging to the foreground [25]. In brief, its general steps are as follows. It first generates a set M of (sampling) image windows. For each image window im_i ($\in M$), we denote by w_i, h_i, and (x_i^c, y_i^c) its width, height and center, respectively. Then, for each image window im_i ($\in M$) it computes the *objectness score* [53], denoted by OS_i, based on low-level cues. The low-level cues, to some extent, reflect the likelihood of a given image window containing an object/target. Usually, three types of low-level cues "color-contrast, multi-scale saliency, and edge density" are used, in order to obtain the objectness

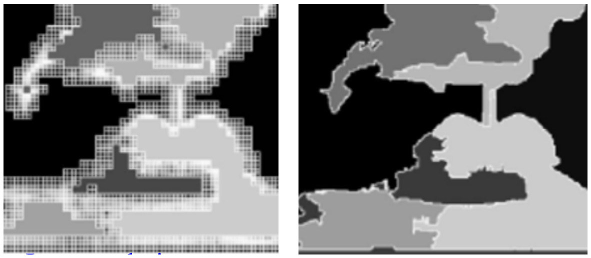


Fig. 4. Example of images segmented using M-RSST algorithm with different resolutions: (a) low resolution; (b) high resolution.

score of a given (sampling) image window. Next, it computes the pixel-level objectness, denoted by $O_{(p)}$, as follows.

$$O_{(p)} = \sum_{i=1}^{|M|} OS_i * exp[-(\frac{(x_p - x_i^c)^2}{2\delta_x^2} + \frac{(y_p - y_i^c)^2}{2\delta_y^2})], \quad (4)$$

where x_p and y_p denote the location of a pixel p; in addition, δ_x and δ_y are set to $0.25w_i$ and $0.25h_i$, respectively. Note that, the above equation can be essentially viewed as a multiplication of two components: (i) an overlapping score, i.e., the objectness score OS_i; and (ii) the Gaussian smoothing kernel of all sampling windows, i.e., $exp[-(\frac{(x_p-x_i^c)^2}{2\delta_x^2} + \frac{(y_p-y_i^c)^2}{2\delta_y^2})]$. Finally, it computes the superpixel-level objectness likelihood map as follows.

$$O_{(sp_i)} = \frac{1}{k} \sum_{p \in sp_i} O_{(p)}, \quad (5)$$

where k denotes the number of pixels in a superpixel. Here the superpixel-level objectness likelihood map corresponds to the concept of *region-level objectness map* in [25].

IV. A DEEP ANALYSIS ON THE CHALLENGES

To understand the difficulty of introducing the multi-scale global cues for saliency detection, this section examines the challenges in more detail.

▷ *Challenge 1.* Most of *multi-scale segmentation* based algorithms fail to *flexibly* select the size of the scale [54]–[56]. A recent proposal, the *simple linear iterative clustering* (SLIC) algorithm [41], can allow users to flexibly set the size of the scale. Observe that the *texture* and *noise* information could appear in most of images, which can incur the poor results if one directly uses the SLIC algorithm; see Fig. 3(a), for example. To alleviate this issue, a natural solution is to first smooth the image, instead of executing the SLIC algorithm at the beginning. There are many proposals for smoothing the image, and most of which easily lead to *loss of information* on the edges [43], [57]. The bilateral filter can efficiently address this limitation. That is, it can smooth the image while preserving the edges [58], [59]. So, it is pretty natural to use the bilateral filter to smooth the image.

However, there is a trouble needing to be alleviated. Specifically, most of prior works apply the bilateral filter for the case of single-scale segmentation (see e.g., [58], [59]). In those works, the "filter parameters" (i.e., σ_r and σ_d, recall Section III) are chosen manually. In our context, if one directly extends existing bilateral filter algorithms, it would be pretty tedious to adjust the parameters for different scales[1]. Then, the following natural question arises: How to assign the appropriate *filter parameters* for different scales?

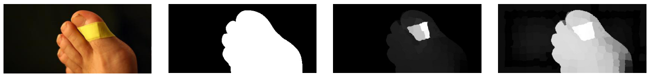


Fig. 5. Example of all boundary regions as the background seeds. (a) Input image; (b) ground truth; (c) all boundary regions as background seeds; (d) our method.

Remark. The SLIC algorithm used in our paper is different from the classic M-RSST algorithm [60], the former operates the image at the superpixel level with different scales; it can segment the image into superpixels with different sizes, as shown in the first row of Fig. 1. In contrast, the latter operates the image or frame (in video sequences) at the pixel level with different resolutions, and it is initially developed for motion and color segmentation (the segmentation effect is as shown in Fig. 4). Since the framework proposed in the paper needs to operate the image at the superpixel level with different scales, it is natural to use the SLIC algorithm.

▷ *Challenge 2.* To generate the final saliency map, most of recent methods first generate RBB and RFB saliency maps respectively, and then merge them using various techniques. The quality of RBB and RFB saliency maps are closely related to the effect of the final saliency map [25]. Particularly, *selecting* the appreciate background (resp., foreground) seeds is extremely important for constructing the RBB (resp., RFB) saliency map [25], [35].

On one hand, the boundary prior has been extensively used and shown to be effective in selecting the background seeds [25], [30], [36], [38]. However, if one chooses all boundary regions as the background seeds, it could lead to poor results as shown in Fig. 5(c); this is because objects in some images could appear in the boundary regions. This observation motivates recent works (see e.g., [35]) to filter out part of background seeds by contrasting the color similarity of each boundary region and other regions. Yet, as pointed out in [25], this line of methods could produce poor results when the background of the image is complicate.

On the other hand, the center-prior has been shown effective in selecting the foreground seeds [11], [38]. Yet, such method could not work well (c.f., Figs 6(b) and 6(f)) when the saliency object does not appear at the center of the image. The convex hull based methods can capture the location of object [34], [61], whereas some background regions could be mistaken for the foreground seeds (c.f., Figs 6(c) and 6(g)). Recently, the objectness likelihood map (OLM) technique has been also used in the construction of the RFB saliency map [25], whereas the threshold parameters (involved in this line of methods) cannot be determined easily. A "bad" threshold could incur

[1] Note that, different filter parameters shall incur different smoothness [43], [52], and for a certain scale an "appropriate" smoothness (e.g., Fig. 3(c)) is vital when segmenting the image [58], [59]. This implies that a "bad" filter parameter could incur the poor segmentation results as shown in Figs 3(b) and 3(d).

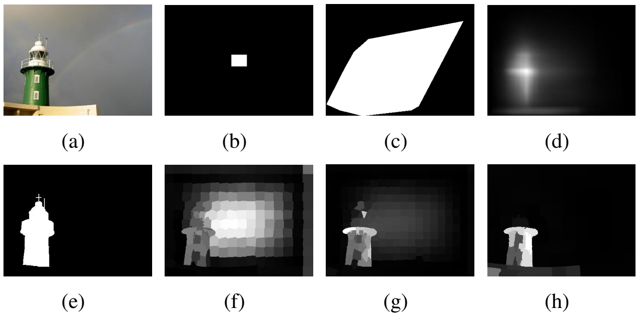


Fig. 6. Illustration of center prior, convex hull prior, and objectness likeness map: (a) Input image; (b) center prior; (c) convex hull prior; (d) objectness likelihood map (OLM); (e) ground truth; (f) saliency result via center prior; (g) saliency result via convex hull prior; (h) saliency result via OLM.

superabundant foreground seeds, getting a poor RFB saliency map; see Figs 6(d) and 6(h).

Essentially, the above issues further emphasize the importance of selecting the background (resp., foreground) seeds. Then, how to select "useful" background and foreground seeds easily and effectively?

▷ *Challenge 3.* Without loss of generality, assume that we have obtained $|T|$ RBB and $|T|$ RFB saliency maps. Denote by S_b^i (resp., S_f^i) the ith RBB (resp., RFB) saliency map. To obtain the final saliency map, an easily brought to mind method is computing the average value of all rough saliency results (for each pixel p). Let $S_b^i(p)$ (resp., $S_f^i(p)$) be a pixel from the ith RBB (resp., RFB) saliency map. Denote by $V(p)$ the pixel of the final saliency result. It is computed as

$$V(p) = \frac{\sum_{i=1}^{|T|} S_b^i(p) + \sum_{k=1}^{|T|} S_f^i(p)}{2 \times |T|}. \tag{6}$$

This natural method, however, could produce pretty poor results as shown in Fig. 7(c). This is because (i) it ignores the fact that different segmentation scales may have different effects even if one segments the same image; and (ii) it ignores the differences of information from the RBB and RFB saliency maps (c.f., Fig. 8). Note that, although some rough saliency map may has the poor effect (e.g., the RFB saliency map in Fig. 8(d)), a single rough saliency map with poor effect cannot make decisional impact on our final salient result; see Fig. 7(c) or Fig. 7(d) for an illustration.

Another potential method is the cross-product based method. That is,

$$V(p) = \frac{1}{|T|} \sum_{i=1}^{|T|} S_b^i(p) \times S_f^i(p) \tag{7}$$

This approach also easily produces poor results as shown in Fig. 7(d), since the pixels' feature values in RBB saliency maps are usually small, the cross-product operation weakens the difference between the saliency object region and background region.

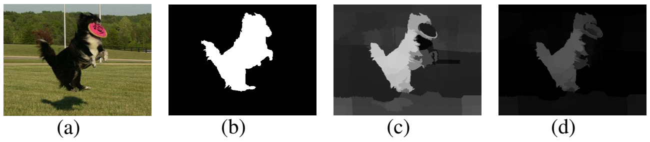


Fig. 7. Example of fusion: (a) Input image; (b) ground truth; (c) result by Eq. 6; (d) result by Eq. 7.

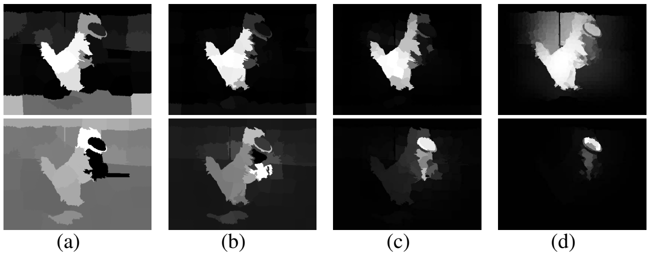


Fig. 8. RBB saliency maps (c.f., the first row) and RFB saliency maps (c.f., the second row) based on different scales: the scale t_i (from left to right) is 50, 100, 200, 500, respectively.

The above issues immediately highlight the need for developing a more competitive approach (to merge the rough saliency maps). That is, how to merge, in a better way, all rough saliency maps such that the final saliency map can have better quality?

V. A THREE-PHASE SOLUTION

This section describes our solution at a high level. Fig. 9 covers the general framework of our solution. It consists of three main phases: (1) segmenting the image based on multi-scales; (2) choosing the prior cues from background and foreground; and (3) refining the rough saliency results.

▷ *Phase 1.* The main goal in the first phase is to separate/segment the image into superpixels according to different scales. This way, multiple images shall be generated. As discussed in previous sections, the texture and noise information could appear in most of images. Our solution employs a bilateral filter to smooth the image, before segmenting the image. In the context of our concern, the filter parameters are not easy-to-modulate. To address this issue, we present a *self-adaptive* strategy to obtain appropriate filer parameters for different scales. The difficulty in developing this strategy is to construct a relationship between the smoothness and filter parameters. Our approach first constructs an objective function by incorporating pixels' color differences and filter parameters, and then to solve the function using a simple algorithm.

▷ *Phase 2.* The second phase mainly serves as selecting "useful" background/foreground seeds as the prior cues. These prior cues can be immediately used to generate RBB and RFB saliency maps, using the *manifold ranking technique* (c.f., Section III). To select useful background/foreground seeds, we propose a new scheme that employs an idea "cross-validation". Generally speaking, our *cross-validation* idea first generates "initial" background and foreground seeds via existing techniques; it then removes part of initial background seeds by exploiting the feature information from initial foreground seeds, getting the "refined" background seeds; and finally it removes part of initial foreground seeds by exploiting the feature information form refined background seeds. A central observation utilized by this phase is that, the differences between the background and foreground are usually larger than the differences between internal regions in background or foreground.

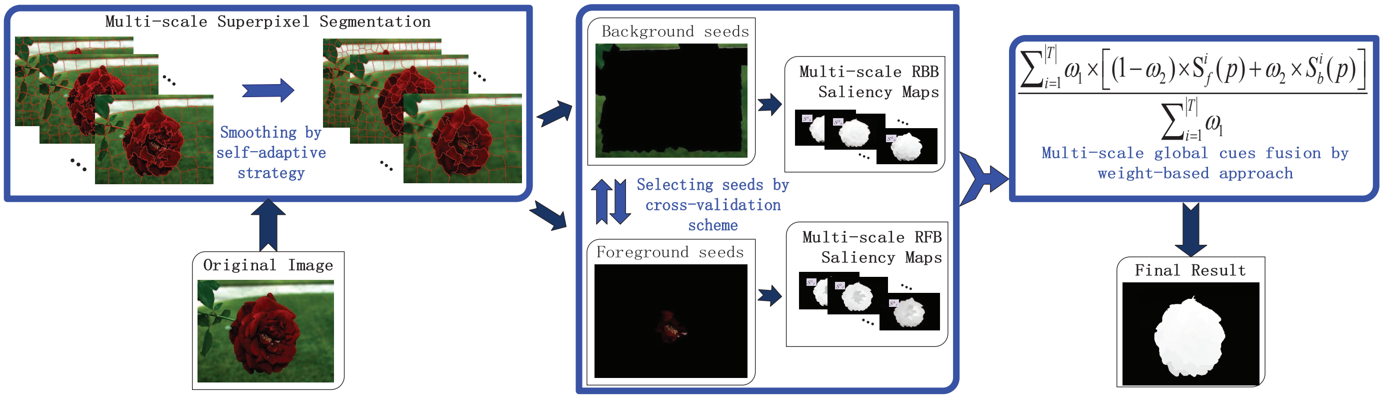

Multi-scale Superpixel Segmentation
Smoothing by self-adaptive strategy
Original Image
Background seeds
Multi-scale RBB Saliency Maps
Selecting seeds by cross-validation scheme
Foreground seeds
Multi-scale RFB Saliency Maps
$\frac{\sum_{i=1}^{|T|}\omega_1\times\left[(1-\omega_2)\times S_f^i(p)+\omega_2\times S_b^i(p)\right]}{\sum_{i=1}^{|T|}\omega_1}$
Multi-scale global cues fusion by weight-based approach
Final Result

Fig. 9. Overview of our solution. Three phases are involved in our solution: (i) segmenting the image based on multi-scales; (ii) choosing the prior cues from background and foreground; and (iii) refining the rough saliency results. In this figure, three boxes in blue corresponding to three phases of our solution, respectively; and the texts in blue corresponding to strategies/approaches that address challenges in these three phases.

▷ *Phase 3.* The third phase is to merge rough salient maps obtained before, in order to get the final salient 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. The main idea of our approach is that, for each single pixel p, we consider the differences among various segmentation scales, and also the similarity from p to the refined foreground/background seeds; we then utilize fully these information in the merging phase.

In what follows, we detail each key step of our solution (Sections VI∼VIII).

VI. Obtaining filter parameters

As mentioned earlier, the challenge in the first phase is on how to choose the appropriate filter parameters for different scales. Observe that, preserving edges is urgently needed for subsequent segmentation operation; this allows us to set conservatively the parameter σ_d to a relatively small value, so as to the "smoothed" image can still have enough sharpness of the edges. Naturally, the salient object shall be not totally removed; see Fig. 3(c) for an illustration. In our experiment, σ_d is empirically chosen and 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 mentioned above, we propose a *self-adaptive strategy* that can assist us to flexibly and automatically choose the appropriate filter parameters for different scales. The rationale behind our strategy is to construct an objective function by incorporating pixels' color differences and filter parameters, 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. The formal steps of our approach are introduced below.

Let T be 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

$$S=\sum_{j=1}^{t_i}\sum_{i=1}^{k}\|c_{i,j}-\overline{c(j)}\|. \quad (8)$$

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

$$F=\text{argmin}_{\sigma_r\in\mathbb{R}}(S+c\sigma_r), \quad (9)$$

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 experiment, c is set to 100, unless stated otherwise. For the latter parts (i.e., $c\sigma_r$) in the above equation, one can essentially view them as the penalty factors, which are used to alleviate a too large global smoothness (notice: an over-large smoothness shall incur great loss of information, which makes no contribution or even negative contribution to the subsequent operations; see Fig. 3(d), for example.

It is easily verified that the equation above essentially characterizes an optimization problem. One can solve the above optimization problem trivially by executing the following simple method[2], which can be viewed as a simple version of the *gradient descent algorithm* [62]. 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 Equation 8). Specifically, our method first sets σ_r as a small value, and then increases it gradually,

[2] Remark that, the approach executes the segmentation operation many times, since (i) our algorithm needs multiple times iterations, in order to find the appropriate value for the parameter σ_r; and (ii) it needs to compute S according to Equation 8.

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 is not to solve the optimization problem described in Equation 9. Essentially, the difficulty is to construct a relationship between the smoothness and filter parameters.

VII. Selecting background/foreground seeds

This section describes our cross-validation scheme for selecting the background/foreground seeds. The rationale behind our scheme is utilizing an observation, "the differences between the background and foreground are usually larger than the differences between *internal regions* in background or foreground", to remove part of the "initial" background and foreground seeds, which are obtained based on existing techniques. 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[3] we first use the *objectness likelihood map* technique mentioned in Section III to get the "initial" foreground seeds, and conservatively use the boundary of the image as initial background seeds[4]. Without loss of generality, assume 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

$$D_{(is_b^j, is_f)} = \sum_{i=1}^{m} (\|c_b^j - c_f^i\| + \theta \|l_b^j - l_f^i\|) \qquad (10)$$

where θ is a parameter used to adjust the weight of the location information. In our experiment, it is empirically chosen and set to 0.5, unless stated otherwise.

For ease of discussion, we dub the above value 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. This 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 score 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". Note that, selecting the dividing line in this way is essentially an implementation to the rationale mentioned at the beginning of this section. 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 also obtain their credit scores; and then sort the m credit scores; and finally remove some initial foreground seeds, using the method similar to the above. A minor difference is that we employ the feature information from the "refined" background seeds[5], instead of the one from the initial background seeds. The details are omitted, for saving space. (As mentioned in previous sections, once the refined background/foreground seeds are obtained, one can use the manifold ranking technique to get $|T|$ RBB and also $|T|$ RFB saliency maps.)

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 it is effective to improve the quality of saliency results, as demonstrated in Section X.

VIII. Refining rough saliency results

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 the saliency map with a good quality; 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

[3] Note that, we previouslly segmented the image into superpixels according to different scales. Now we have $|T|$ images.

[4] The reason we use the objectness map technique to obtain the initial foreground seeds is that, it allows us to capture the location of saliency object(s); this nature is important for applying our subsequent operations. In addition, the reason we use the naive method, "choosing the boundary of the image as the background seeds", is that, the background seeds chosen by existing approaches may have loss of information; it will make negative impact on our subsequent operations.

[5] The reason we use refined background seeds is that, these seeds have better quality than the initial background seeds, and so it is more likely to obtain the refined foreground seeds with good quality.

maps (Phase 2). Our approach shall take full use of these available information. Note that, in the following discussion, we focus on considering a single pixel. Other pixels in the final saliency map can be obtained one by one, using the same method introduced below. Specifically, our approach works as follows.

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.

$$scw_i(p) = \left\| c(p) - \overline{c_i(p)} + \epsilon \right\|^{-1} \tag{11}$$

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 (or connector) 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.

$$sew_i(p) = \frac{\left\| c(p) - \overline{c_i^{m'}(p)} \right\|}{\left\| c(p) - \overline{c_i^{n'}(p)} \right\| + \left\| c(p) - \overline{c_i^{m'}(p)} \right\|} \tag{12}$$

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 the 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 the pixel p's result in the final saliency map. It is computed as

$$V(p) = \frac{\sum_{i=1}^{|T|} \omega_1 \times [(1 - \omega_2) \times S_f^i(p) + \omega_2 \times S_b^i(p)]}{\sum_{i=1}^{|T|} \omega_1} \tag{13}$$

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. Remark that, in our approach the scale and seed weights are computed inspired by heuristics or intuitions mentioned earlier; and our approach is not optimal for each type of image, as same as all existing salient detection algorithms.

IX. Computational Complexity Analysis

In this section, we discuss the computational cost of our proposed method. In the following discussion, we use N to denote the number of pixels in the image, and use K to denote $\text{argmax}_{i \in [1, |T|]} t_i$. As stated before, our method consists of three phases. In phase 1, our method adopts $|T|$ scales, and it segments the image using SLIC algorithm [41]. As a result, the corresponding cost for segmentation is $O(|T| * N)$, since the computational complexity of the SLIC algorithm is $O(N)$, and independent of the number of superpixels. Next, our method chooses appropriate parameter for each scale and uses the bilateral filter to smooth the images. To get the appropriate parameter, the dominant step is to compute Eq. 8, which needs $O(N)$ cost. As for the smoothing operation, it takes $O(2{*}N)$, which can be understood from [52]. Thus, the overall cost for choosing appropriate parameters and for smoothing is $O(|T| * N)$, since we have $|T|$ scales. To summarize, the first phase takes $O(|T| * N + |T| * N)$ cost, i.e., $O(|T| * N)$.

As for phase 2, our method first generates initial foreground and background seeds via objectness likelihood map technique and boundary prior. In this process, the dominant step is to compute Eq. 4, which needs $O(|M|{*}N)$ cost, since computing objectness score OS_i takes $O(N)$ time [53]. Note that, here $|M| << N$, so the above result is approximately $O(N)$. Next, our method obtains refined foreground and background seeds via the cross-validation scheme. Obtaining refined background and foreground seeds takes $O(m{*}n{+}n \log n)$ and $O(m*n + m \log m)$ cost, respectively. Note that, here $m < t_i$, $n < t_i$, and $m{+}n < t_i$, since the foreground and background seeds are chosen from superpixels. Thus, obtaining refined foreground and background seeds takes $O(2*m*n + n \log n + m \log m)$ cost, i.e., $O(t_i^2)$. In addition, the manifold ranking technique takes $O(t_i^3)$. Overall, the second phase takes $O(|T| * (N + t_i^3 + t_i^2))$ cost, since we have $|T|$ scales. Note that, this result is approximately $O(|T| * (N + t_i^3))$.

In the 3rd phase, our method first computes scale and seed weights, respectively. Since the average feature vector has been computed in phase 1, Eq. 11 takes constant time. Similarly, since $m' << N$ and $n' << N$, Eq. 12 takes also constant time. Furthermore, since we have $|T|$ scales, the overall complexity for computing the above two scales is $O(|T|)$. After that, our method merges the RFB and RBB saliency maps via Eq. 13, which takes $O(|T|)$ time. To summarize, phase 3 takes $O(|T| * N)$ cost, since each image contains N pixels. Putting all together, the overall computational complexity of our method is $O(((|T| * N) + (|T| * N + |T| * t_i^3) + (|T| * N))$, i.e., $O(|T| * N + |T| * t_i^3)$. Note that, t_i is upper bounded by K, so the theoretical time complexity of our algorithm is $O(|T| * (N + K^3))$. As a remark, $|T| << N$ in practice.

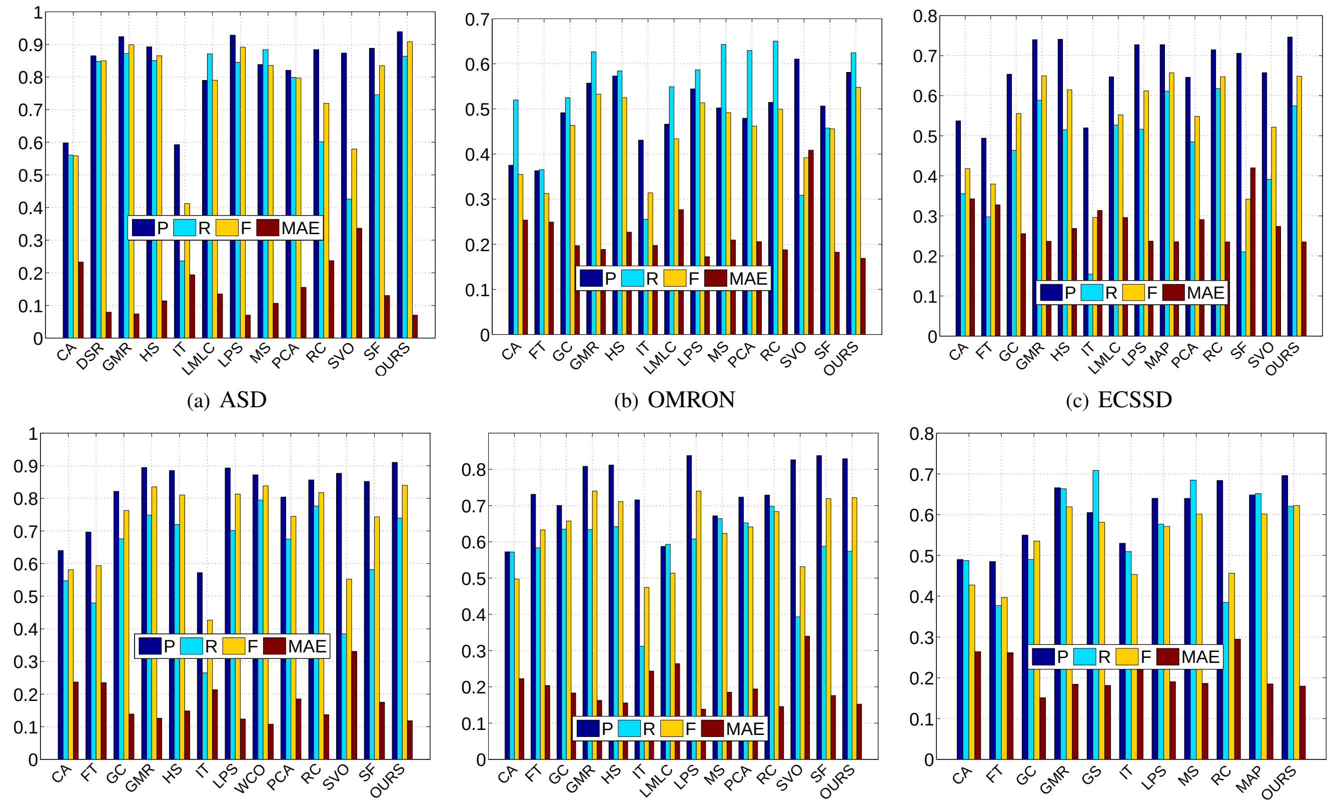
X. Experiments

In this section, we first describe the datasets, evaluation metrics, and existing methods used in our experiments (Section X-A), and then cover the experimental results (Sections X-B $\sim$ X-D).

A. Experimental settings

We evaluate our method using six datasets (available for public use):

- *ASD* [21]. This dataset is selected from MSRA-5000 dataset. It is widely used in prior works (see e.g., [30], [38]).
- *OMRON* [35]. It is one of most challenging datasets, in which 5168 images are included.



(a) ASD (b) OMRON (c) ECSSD

(d) THUS (e) SED2 (f) PASCAL

Fig. 10. The results are obtained based on various methods over different datasets (including ASD, OMRON, ECSSD, THUS, SED2, and PASCAL).

TABLE I
THE NUMERICAL RESULTS: *F-measure* (F) AND *mean absolute error* (MAE)

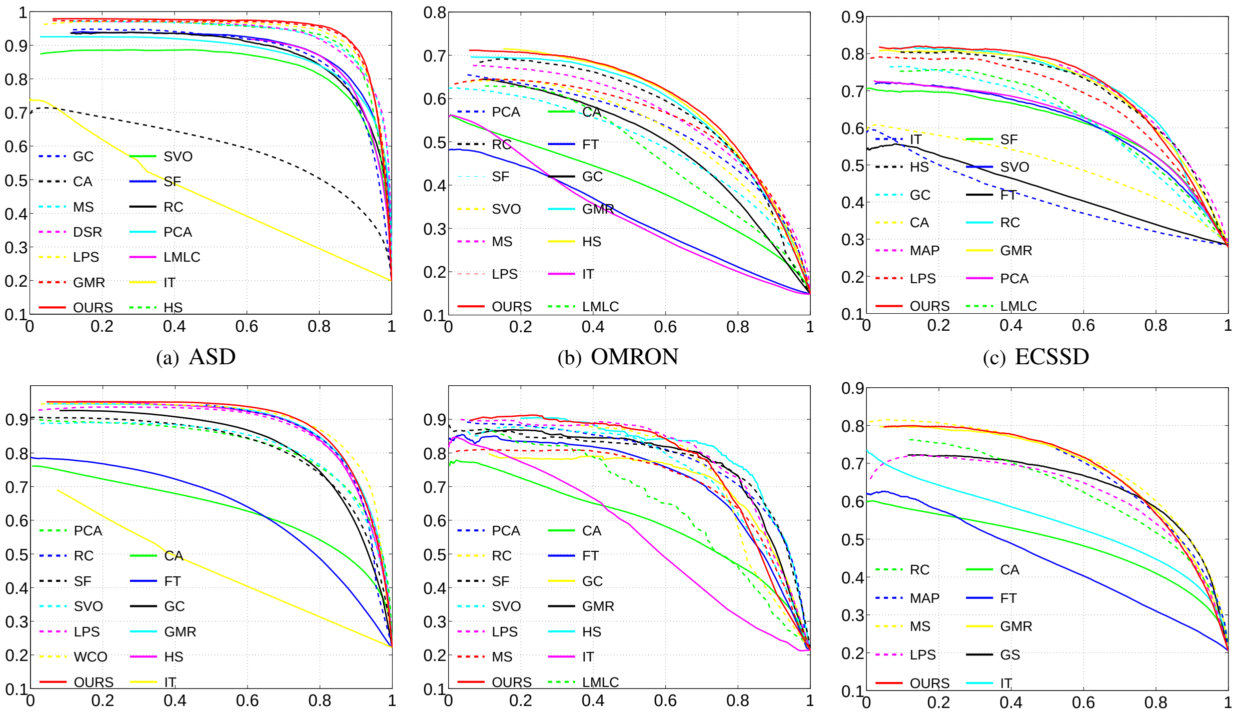
Metric	**Dataset**	**Method**													
		CA	**FT**	**GC**	**GMR**	**HS**	**IT**	**LMLC**	**LPS**	**MS**	**PCA**	**RC**	**SF**	**SVO**	**OURS**
F	ASD	0.558	0.640	0.820	0.898	0.865	0.412	0.790	0.891	0.835	0.796	0.719	0.834	0.579	**0.908**
	OMRON	0.355	0.313	0.464	0.533	0.525	0.313	0.434	0.510	0.492	0.462	0.499	0.456	0.392	**0.545**
	ECSSD	0.418	0.380	0.556	0.650	0.615	0.296	0.552	0.612	0.636	0.548	0.647	0.521	0.341	**0.658**
	THUS	0.581	0.593	0.762	0.835	0.810	0.426	0.760	0.812	0.151	0.744	0.817	0.743	0.552	**0.840**
	SED2	0.497	0.633	0.657	**0.739**	0.710	0.474	0.513	0.738	0.623	0.641	0.683	0.719	0.531	0.721
	PASCAL	0.427	0.397	0.535	0.619	0.595	0.453	0.470	0.571	0.601	0.548	0.456	0.506	0.400	**0.622**
MAE	ASD	0.232	0.205	0.100	0.074	0.114	0.194	0.135	0.071	0.106	0.155	0.237	0.130	0.336	**0.070**
	OMRON	0.253	0.249	0.197	0.188	0.227	0.197	0.276	0.172	0.209	0.205	0.187	0.182	0.408	**0.169**
	ECSSD	0.342	0.327	0.255	0.237	0.268	0.314	0.296	0.237	0.250	0.290	0.236	0.273	0.420	**0.235**
	THUS	0.237	0.234	0.139	0.125	0.148	0.213	0.164	0.123	0.794	0.185	0.137	0.175	0.331	**0.118**
	SED2	0.222	0.203	0.183	0.162	0.155	0.243	0.264	**0.138**	0.185	0.194	0.145	0.176	0.339	0.152
	PASCAL	0.264	0.261	0.215	0.184	0.223	0.259	0.289	0.190	0.186	0.209	0.294	0.216	0.368	**0.179**

- *ECSSD* [11]. It includes a lot of images with complicated background. Some of which are selected from the Berkeley-300 dataset.
- *THUS* [23]. It contains 10000 images, labelled with pixel-wise ground truth masks.
- *SED2* [63]. It contains 100 images in which multiple objects/targets appear, and the objects are with different sizes.
- *PASCAL* [64]. It contains 1500 images with pixel-wise ground truth masks.

We compare our algorithm against the classic and/or some state-of-the-art methods, including IT [12], FT [21], CA [24], SVO [22], RC [23], SF [29], GS [33], PCA [28], LMLC [34], HS [11], GC [65], GMR [35], DSR [27], MS [43], WCO [36], LPS [25], and MAP [31]. In the paper, we use 20 scales ranging from 10 to 390, the incremental factor is 20 between two adjacent scales. These diversified scales can allow us to capture saliency information for various images (e.g., different sizes in terms of salient object, different color distributions or contrasts in images, etc). Same to [35], to evaluate the methods, we use results provided by authors or run their implementations based on the available codes or softwares.

In our experiments, we use several main evaluation metrics below.

- *Precision and recall.* The precision value, denoted by v_p, refers to the ratio of salient pixels correctly assigned to all the pixels of extracted regions. The recall value, denoted by v_r, refers to the percentage of detected salient pixels



(a) ASD (b) OMRON (c) ECSSD

(d) THUS (e) SED2 (f) PASCAL

Fig. 11. The P-R curve diagram (notice: the vertical axis reflects the precession while the horizontal axis reflects the recall). The results are obtained based on various methods over various datesets including ASD, OMRON, ECSSD, THUS, SED2, and PASCAL.

with regard to the ground-truth number. Following prior works [30], [36], [38], we obtain P-R (precision-recall) curve by binarizing the saliency map, using thresholds in the range of 0 and 255.

- *F-measure.* It is for evaluating the quality of saliency map. It is computed as

$$F = \frac{(1+\eta^2) \times v_p \times v_r}{\eta^2 \times v_p + v_r}, \tag{14}$$

where η is used to control the ratio of precision and recall. Following prior works [30], [38], η^2 is set to 0.3.

- *Mean absolute error.* It is the mean difference between the saliency map and the ground truth. Denote by E_{ma} the mean absolute error (MAE). It is computed as

$$E_{ma} = \frac{1}{N_p} \sum_{i=1}^{N_p} |S(p_i) - GT(p_i)|, \tag{15}$$

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.

In our experiments, we mainly report F-measure, mean absolute error, precision, recall and also the *precision-recall* (P-R) curve.

B. Comparing with existing methods

This section compares our proposed method with existing methods. We first analyse the quantitative results, and then compare the qualitative results of different saliency detection methods.

1) Quantitative evaluation: Fig. 10 plots the experimental results of various methods (including ours) over six datasets, in terms of F-measure (F) and mean absolute error (MAE) respectively. (Table I summarizes the numerical results, for ease of comparison in detail.) In addition, the precision-recall (P-R) curves for various methods over these datasets are covered in Fig. 11.

▷ *F-measure.* On one hand, for the datasets in which images are with a complex background (e.g., ASD, ECSSD and PASCAL datasets), we can see from Figs 10(a), 10(c), and 10(f) that, our proposed method has the larger F-measure value[6] than other methods, demonstrating that our method performs well against the competitors. On the other hand, for the more challenging datasets in which not only the images are with complex backgrounds and texture structures, but also the locations and sizes of objects/targets are diversified (e.g., OMRON and THUS datasets), we can see from Figs 10(b) and 10(d) that, our proposed method has also the larger F-measure values, compared with other methods. In summary, these results show us that our proposed method could be more competitive when the images are with complex background, or more challenging scenarios. We have to point out that, for the SED2 dataset (in which the images usually contain multiple saliency objects), our proposed method is superior than most of competitors, whereas it cannot beat all the competitors (e.g., GMR and LPS methods). Nevertheless, one can find

[6] Note that, the F-measure takes both the precision and recall into account, it is a comprehensive measurement on the effectiveness of the algorithms [30], [36], [38]. Usually, the larger the F-measure value is, the better the algorithm is.

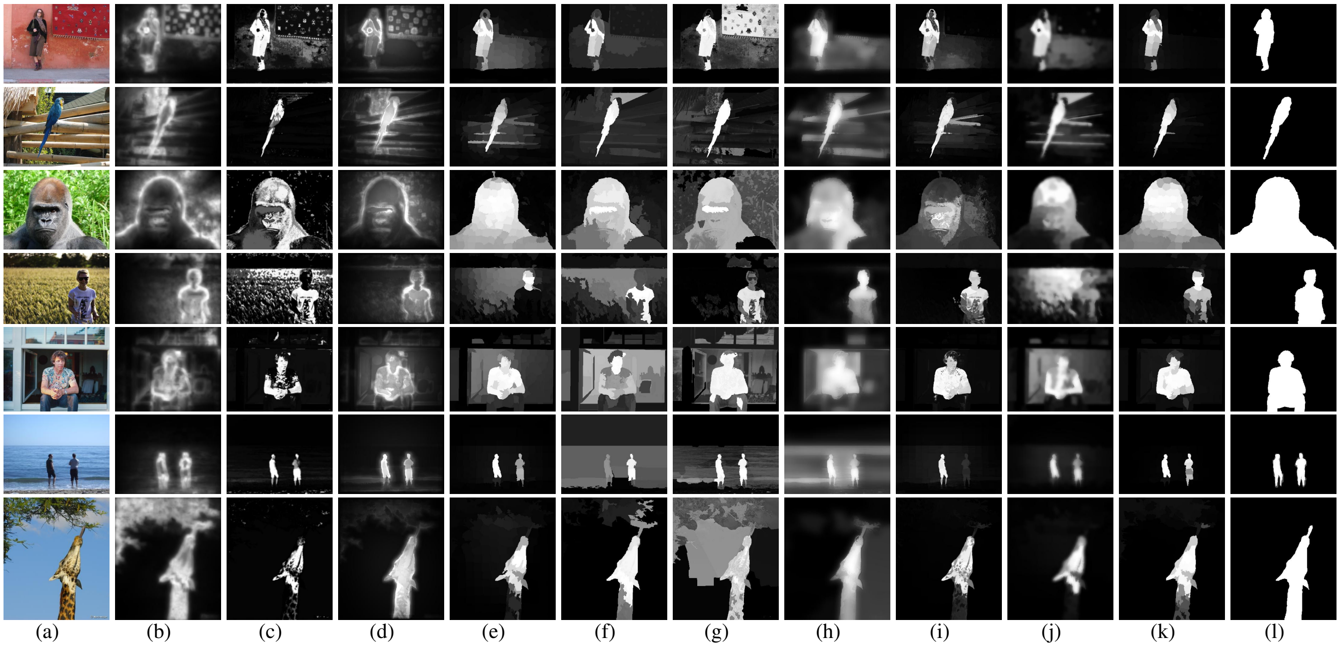


Fig. 12. Examples of saliency detection results: (a) *Input image*; (b) CA; (c) GC; (d) PCA; (e) MR; (f) HS; (g) LMLC; (h) MS; (i) LPS; (j) MAP; (k) *OURS*; (l) *Ground truth*. Our solution obtains the saliency maps close to the ground truth.

from Table I that the gap between our method (with 0.721 F-measure value) and GMR or LPS (with 0.739 F-measure value) is not very large. This phenomenon also happens in the following experiments.

▷ *Mean absolute error.* As described in Section X-A, the MAE is the average difference between the saliency map and the ground truth map. Usually, the smaller the value is, the better the algorithm is. From Figs 10(a), 10(c), 10(f), 10(b) and 10(d), one can see that, for the datasets in which the images are with complex background (e.g., ASD, ECSSD and PASCAL) or the images are more challenging (e.g., OMRON and THUS), our proposed method always has smaller values than other methods. These results further demonstrate that our proposed method could be more competitive for the datasets in which the images are with complex background, or the images are more challenging. Also, for the SED2 dataset, our proposed method has smaller values than most of competitors. Yet, the MAE values of our proposed method are larger than the ones of LPS and RC methods (see Fig. 10(e) or Table I). This fact demonstrates that there are still some improvement spaces for us to further enhance the proposed method. Note that, it is challenging to further narrow the gap (or even to obtain better performance) on this types of datasets; we leave this interesting problem as our future work.

▷ *Precision-recall curve.* 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 [36], [38]. We can see from Fig. 11 that, for most of datasets our algorithm (plotted as the red curve) shows the strong robustness and performs well. Particularly, the precision of our algorithm is pretty good. Specifically, on the ASD and THUS datasets (c.f., Figs 10(a) and 10(d)), our proposed solution achieves more than 90% precision value; and it has the highest precision value, compared with other methods. For other datasets, our proposed solution cannot achieve the precision value up to 90%, whereas it is still pretty good — the precision value of our proposed solution ranks in the top three.

2) Qualitative evaluation: Fig. 12 shows a few saliency maps generated by our solution and state-of-the-art approaches. From this figure, we can see that, for the complicate scenarios where the images are with complex backgrounds and texture structures, our solution can accurately highlight the saliency object/target, and can preserve well the edges of the object (see e.g., the first and second rows in Fig. 12). Furthermore, our solution performs well even if the saliency object is near to the boundary of the image (see e.g., the third, fourth, fifth and seventh rows in Fig. 12). As noted earlier, when the image are with multiple objects/targets, our solution also achieves better results than most of methods, see e.g., the sixth row in Fig. 12). All of the facts presented in Section X-B demonstrate that the overall performance of our solution is pretty good, implying that it is a competitive and attractive solution.

C. Effectiveness of the proposed strategies

In this section, we examine the effectiveness of the strategies/techniques employed in our solution. Recall that three major strategies are developed: (i) a self-adaptive strategy for obtaining the appropriate filter parameters; (ii) a cross-validation scheme for selecting background and foreground seeds; and (iii) a weight-based approach to merge the rough saliency results. To evaluate the effectiveness of each technique, we replace it with the traditional approach, and then compare it with our solution.

1) Self-adapative strategy: For ease of validating the effectiveness of this technique, we compare our proposed solution with the following baseline method: for different scales we

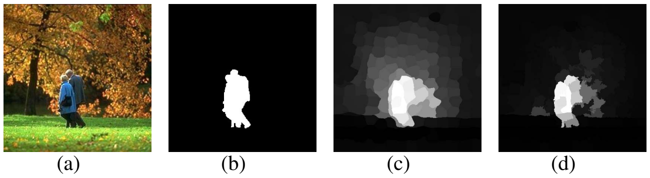


Fig. 13. Effectiveness of self-adaptive technique. (a) Input image; (b) ground truth; (c) choose the value of σ_r randomly; (d) self-adaptive approach.

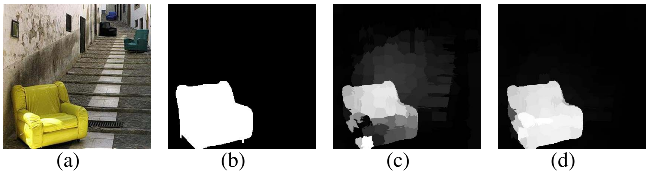


Fig. 14. Effectiveness of cross validation technique. (a) Input image; (b) ground truth; (c) without the cross-validation; (d) our result.

randomly choose a value from $(0, 100]$[7], and assign it to σ_r; the rest of steps are the same as our proposed method. Fig. 13 shows the saliency results generated using a challenging input image. Compared with the baseline method, we can see from this figure that there are two major differences: (i) the saliency map generated by our proposed method contains less background noises; and (ii) the saliency map generated by our proposed method is much more close to the ground truth. These differences essentially reflect the effectiveness of the self-adaptive strategy.

2) Cross-validation scheme: We compare our proposed solution with the following baseline method: it uses our self-adaptive scheme to choose filter parameters for different scales, and then directly uses existing approaches (mentioned in Section VII) to choose background and foreground seeds, while it does not employ the cross-validation scheme; the rest of steps are the same as our proposed method. Fig. 14 shows the comparison result. Compared with the baseline method, we can see from this figure that (i) the saliency object's location has been obtained more exact; and (ii) the background noises have been well suppressed. These evidences demonstrate us that the cross-validate scheme is effective.

3) Weight-based approach: We compare our proposed solution with the following baseline methods: (1) we compute the average value of all rough saliency results; and (2) we compute the cross-product for getting the final saliency maps, recall Section IV. Note that, similar to our previous settings, the other steps are same to our proposed method. Fig. 15 compares the result of these three methods. 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 saliency object is not well highlighted. In contrast, compared with these two baseline methods, the saliency map generated by our proposed method is clearer, regardless of the background or the saliency object. Particularly, the saliency object is well highlighted. These results validate the effectiveness of the weight-based approach.

D. Limitation

Our method exposes many advantages, while it also bears a major limitation. That is, its running time is somewhat long. This section reports the average execution time of our algorithm and some representative algorithms. In our test, we employ the widely used ASD [21] dataset. We run our algorithm (and also the competitors) on this dataset to get the overall execution time T_a, and then obtain the average execution time on each image by computing $\frac{T_a}{N}$, where $N = 1000$ refers to the number of images in ASD dataset. All experiments are conducted on a machine with Intel(R) Core(TM) i3 CPU @2.40 GHz and 3GB RAM.

Table II shows the results, in which the notation M denotes Matlab, and C denotes C/C++. It can be seen that the execution time of our algorithm is 18.832 seconds, while other methods consume less time (e.g., GMR uses 2.461 seconds, MS uses 13.615 seconds, and PCA uses 16.342 seconds). It could be mainly due to that Phases 1 and 2 in our solution are time-consuming, since it self-adaptively chooses the filter parameters for different scales and also chooses the foreground and background seeds with a cross-validation scheme. Note that, although the execution time is not the most important evaluation metric for saliency detection, it is still interesting to reduce the execution time. This could be another independent and challenging work; in the future, we attempt to further optimize our solution and compare with more algorithms.

E. Further Discussion

Furthermore, we would like to point out that, although this paper focuses on bottom-up approaches (which are mostly based on heuristics; sometimes they are called the *hand-crafted features based methods*), it is interesting to give a brief discussion on the deep learning based methods. For completeness, we compare our method with several deep learning based methods, including ELD [66], RFCN [67], DCL [68], UCF [69]. Table III shows the F-measure and MAE values on the PASCAL dataset. From this table, one can easily see that this line of methods obtain higher F-measure values and lower MAE values, which are better than our results.

Nevertheless, it is worth noting that, most of deep learning based methods take a long time to train their models (e.g., DCL [68] consumes 25 *hours* on an NVIDIA Titan Black

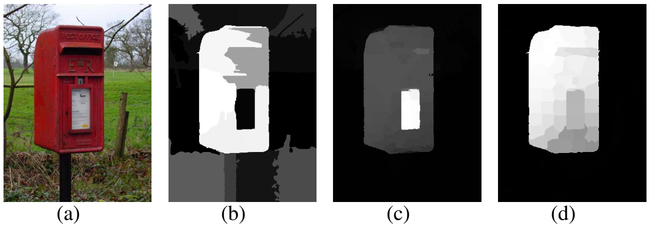


Fig. 15. Effectiveness of weight-based technique. (a) Input image; (b) fusion by computing the average value; (c) fusion by computing the cross-product; (d) our method.

[7] Theoretically speaking, the upper bound can be $+\infty$, we here empirically use 100 as the upper bound for the baseline method, this is because a value larger than 100 shall incur very blur effect, which is basically not benefit for the segmentation operations.

TABLE II
AVERAGE EXECUTION TIME

Method	IT	HS	GMR	MS	PCA	OURS
Time	1.152	2.135	2.461	13.615	16.342	18.832
Code	M	C	M	M	M+C	M

GPU and a 3.4GHz Intel processor, UCF [69] consumes 23 *hours* on an NVIDIA Titan X GPU and an i7-4790 CPU). On the contrast, our method mainly relies on traditional heuristics (just like the methods in [25], [34], [35], [37]–[40], [43], [47]) and thus does not include this time-consuming training process. We argue that both deep learning-based methods and hand-crafted features based methods have their own advantages. In the resource-constraint scenarios (e.g., time-constraint and/or hardware-constraint), hand-crafted features based methods could be more attractive; in the other case, deep learning-based methods could be more attractive. In this regard, our point of view is essentially consistent with that in [25], [31], [43], [70].

TABLE III
EXTRA RESULTS

Method	ELD	RFCN	DCL	UCF	Ours
F	0.771	0.832	0.810	0.818	0.622
MAE	0.121	0.118	0.115	0.116	0.179

XI. Conclusion

In this paper, we proposed a novel bottom-up salient detection mode that is based on multi-scale global cues. The central idea is to use multi-scale segmentation while considering both foreground and background priors. We analyzed and pointed out the challenges to achieve our proposal, and presented a three-phase solution. Our solution alleviated the challenges by developing targeted strategies. Empirical results based on various datasets validated the feasibility and effectiveness of our proposed solution. Also, we pointed out the main limitation of our solution, revealing the future research direction.

Acknowledgement

This work was partially supported by the National Key R&D Program of China (2018YFC0810204, 2018YFB1004400), and the NSFC (61472245, 61472453, 61502220, 61572326, 61775139, 61872242, U1401256, U1501252, U1611264, U1711261, and U1711262), and Shanghai Science and Technology Innovation Action Plan Project (16111107502, 17511107203), and the program for tackling key problems in Henan Science and Technology (172102310636).

References

[1] Y. Fang, W. Lin, B. Lee, C. T. Lau, Z. Chen, and C. Lin, "Bottom-up saliency detection model based on human visual sensitivity and amplitude spectrum," *TMM*, vol. 14, no. 1, pp. 187–198, 2012.

[2] N. Imamoglu, W. Lin, and Y. Fang, "A saliency detection model using low-level features based on wavelet transform," *TMM*, vol. 15, no. 1, pp. 96–105, 2013.

[3] J. Lei, B. Wang, Y. Fang, W. Lin, P. L. Callet, N. Ling, and C. Hou, "A universal framework for salient object detection," *TMM*, vol. 18, no. 9, pp. 1783–1795, 2016.

[4] C. Ma, Z. Miao, X.-P. Zhang, and M. Li, "A saliency prior context model for real-time object tracking," *TMM*, vol. 19, no. 11, pp. 2415–2424, 2017.

[5] C. Siagian and L. Itti, "Rapid biologically-inspired scene classification using features shared with visual attention," *TPAMI*, vol. 29, no. 2, pp. 300–312, 2007.

[6] C. Privitera and L. W. Stark, *Algorithms for Defining Visual Regions-of-interest*. Electronics Research Laboratory, College of Engineering, University of California, 1998.

[7] D. Gao, S. Han, and N. Vasconcelos, "Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition," *TPAMI*, vol. 31, no. 6, pp. 989–1005, 2009.

[8] V. Gopalakrishnan, Y. Hu, and D. Rajan, "Salient region detection by modeling distributions of color and orientation," *TMM*, vol. 11, no. 5, pp. 892–905, 2009.

[9] E. Rahtu, J. Kannala, M. Salo, and J. Heikkilä, "Segmenting salient objects from images and videos," in *ECCV*, 2010, pp. 366–379.

[10] S. Jeong and J. Sim, "Saliency detection for 3d surface geometry using semi-regular meshes," *TMM*, vol. 19, no. 12, pp. 2692–2705, 2017.

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

[12] 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.

[13] J. Kuen, Z. Wang, and G. Wang, "Recurrent attentional networks for saliency detection," in *CVPR*, 2016, pp. 3668–3677.

[14] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?" in *CVPR*, 2010, pp. 73–80.

[15] 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, pp. 2083–2090.

[16] G. Li and Y. Yu, "Visual saliency detection based on multiscale deep cnn features," *TIP*, vol. 25, no. 11, pp. 5012–5024, 2016.

[17] J. Yang and M.-H. Yang, "Top-down visual saliency via joint crf and dictionary learning," in *CVPR*, 2012, pp. 2296–2303.

[18] S. He and R. W. H. Lau, "Exemplar-driven top-down saliency detection via deep association," in *CVPR*, 2016, pp. 5723–5732.

[19] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell, "Sun: A bayesian framework for saliency using natural statistics," *Journal of vision*, vol. 8, no. 7, pp. 32–32, 2008.

[20] N. Liu and J. Han, "Dhsnet: Deep hierarchical saliency network for salient object detection," in *CVPR*, 2016, pp. 678–686.

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

[22] 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.

[23] 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.

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

[25] 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.

[26] K. Fu, I. Y. Gu, and J. Yang, "Saliency detection by fully learning a continuous conditional random field," *TMM*, vol. 19, no. 7, pp. 1531–1544, 2017.

[27] X. Li, H. Lu, L. Zhang, X. Ruan, and M.-H. Yang, "Saliency detection via dense and sparse reconstruction," in *IJCV*, 2013, pp. 2976–2983.

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

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

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

[31] J. Sun, H. Lu, and X. Liu, "Saliency region detection based on markov absorption probabilities," *TIP*, vol. 24, no. 5, pp. 1639–1649, 2015.

[32] H. Dou, D. Ming, Z. Yang, Z. Pan, Y. Li, and J. Tian, "Object-based visual saliency via laplacian regularized kernel regression," *TMM*, vol. 19, no. 8, pp. 1718–1729, 2017.

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

[34] 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.

[35] 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.

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

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

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

[39] W. Zhang, Q. M. J. Wu, G. Wang, and H. B. Yin, “An adaptive computational model for salient object detection,” *TMM*, vol. 12, no. 4, pp. 300–316, 2010.

[40] W. Wang, J. Shen, L. Shao, and F. Porikli, “Correspondence driven saliency transfer,” *TIP*, vol. 25, no. 11, pp. 5025–5034, 2016.

[41] 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.

[42] J. Qi, S. Dong, F. Huang, and H. Lu, “Saliency detection via joint modeling global shape and local consistency,” *Neurocomputing*, vol. 222, pp. 81–90, 2017.

[43] N. Tong, H. Lu, L. Zhang, and X. Ruan, “Saliency detection with multiscale superpixels,” *SPL*, vol. 21, no. 9, pp. 1035–1039, 2014.

[44] Y. Tang and X. Wu, “Salient object detection with chained multi-scale fully convolutional network,” in *ACM Multimedia*, 2017, pp. 618–626.

[45] 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.

[46] G. Li and Y. Yu, “Visual saliency based on multiscale deep features,” in *CVPR*, 2015, pp. 5455–5463.

[47] 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.

[48] S. Lu and J.-H. Lim, “Saliency modeling from image histograms,” in *ECCV*, 2012, pp. 321–332.

[49] S. Lu, C. Tan, and J.-H. Lim, “Robust and efficient saliency modeling from image co-occurrence histograms,” *TPAMI*, vol. 36, no. 1, pp. 195–201, 2014.

[50] Z. Wang, D. Xiang, S. Hou, and F. Wu, “Background-driven salient object detection,” *TMM*, vol. 19, no. 4, pp. 750–762, 2017.

[51] Z.-J. Wang, L. Ma, X. Lin, and X. Wu, “Msgc: A new bottom-up model for salient object detection,” in *ICME*, 2018.

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

[64] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *IJCV*, vol. 88, no. 2, pp. 303–338, 2010.

[53] B. Alexe, T. Deselaers, and V. Ferrari, “Measuring the objectness of image windows,” *TPAMI*, vol. 34, no. 11, pp. 2189–2202, 2012.

[54] L. Vincent and P. Soille, “Watersheds in digital spaces: an efficient algorithm based on immersion simulations,” *TPAMI*, vol. 13, no. 6, pp. 583–598, 1991.

[55] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *TPAMI*, vol. 24, no. 5, pp. 603–619, 2002.

[56] P. F. Felzenszwalb and D. P. Huttenlocher, “Efficient graph-based image segmentation,” *IJCV*, vol. 59, no. 2, pp. 167–181, 2004.

[57] D. Comaniciu and P. Meer, “Mean shift analysis and applications,” in *ICCV*, vol. 2, 1999, pp. 1197–1203.

[58] 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.

[59] T.-H. Lin, D.-L. Way, Z.-C. Shih, W.-K. Tai, and C.-C. Chang, “An efficient structure-aware bilateral texture filtering for image smoothing,” *Comput. Graph. Forum*, vol. 35, no. 7, pp. 57–66, 2016.

[60] A. D. Doulamis, N. D. Doulamis, and S. D. Kollias, “A fuzzy video content representation for video summarization and content-based retrieval,” *Signal Processing*, vol. 80, no. 6, pp. 1049–1067, 2000.

[61] C. Gong, D. Tao, W. Liu, S. J. Maybank, M. Fang, K. Fu, and J. Yang, “Saliency propagation from simple to difficult,” in *CVPR*, 2015, pp. 2531–2539.

[62] X. Yi, D. Park, Y. Chen, and C. Caramanis, “Fast algorithms for robust pca via gradient descent,” in *NIPS*, 2016, pp. 4152–4160.

[63] S. Alpert, M. Galun, A. Brandt, and R. Basri, “Image segmentation by probabilistic bottom-up aggregation and cue integration,” *TPAMI*, vol. 34, no. 2, pp. 315–327, 2012.

[65] 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.

[66] G. Lee, Y. Tai, and J. Kim, “Deep saliency with encoded low level distance map and high level features,” in *CVPR*, 2016, pp. 660–668.

[67] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan, “Saliency detection with recurrent fully convolutional networks,” in *ECCV*, 2016, pp. 825–841.

[68] G. Li and Y. Yu, “Deep contrast learning for salient object detection,” in *CVPR*, 2016, pp. 478–487.

[69] P. Zhang, D. Wang, H. Lu, H. Wang, and B. Yin, “Learning uncertain convolutional features for accurate saliency detection,” in *ICCV*, 2017, pp. 212–221.

[70] L. Zhang, J. Ai, B. Jiang, H. Lu, and X. Li, “Saliency detection via absorbing markov chain with learnt transition probability,” *IEEE Transactions on Image Processing*, vol. 27, no. 2, pp. 987–998, 2018.