The Visual Computer (2019) 35:385–398
https://doi.org/10.1007/s00371-018-1471-4

ORIGINAL ARTICLE

SSG: superpixel segmentation and GrabCut-based salient object segmentation

Xianen Zhou[1] · Yaonan Wang[1] · Qing Zhu[1] · Changyan Xiao[1] · Xiao Lu[2]

Published online: 22 January 2018

Abstract
Saliency detection is a popular topic for image processing recently. In this paper, we propose a simple, robust and fast salient object segmentation framework. Firstly, we develop a novel saliency map segmentation strategy, named SSG which consists of superpixel region growing, superpixel Density-Based Spatial Clustering of Applications with Noise (DBSCAN) clustering and iterated graph cuts (GrabCut), where DBSCAN makes similar background regions cluster as a whole, region growing groups similar regions together as much as possible, GrabCut segments salient objects accurately. Then, the proposed SSG is combined with saliency detection to abstract salient objects. Experimental results on three benchmark datasets demonstrate that the proposed method achieves the favorable performance than many recent state-of-the-art methods in terms of precision, recall, F-measure and execution time.

Keywords Salient object segmentation · Superpixel segmentation · GrabCut · Region growing · DBSCAN clustering

1 Introduction

The most visually noticeable foreground in the scene, known as salient objects, could be quickly, accurately identified by a human being. However, computationally identifying such salient regions is a challenging problem [9,25]. Applications to vision and graphics are numerous, especially in solving problems that require object-level [16]. Up to now, a great many of saliency detection methods have been proposed and achieved considerable progress. Borji et al. [7,8] introduced more than 65 visual attention modeling methods and 22 popular salient object datasets which could be used to evaluate the performance of saliency detection algorithms covering 256 publications from 1998 to 2014. Usually, from the perspective of information processing mechanisms, all saliency detection algorithms are divided into two classes: Bottom-up methods which are data-driven and top-down methods which are task-driven. Depending on the application of saliency detection, existing saliency estimation methods are categorized into fixation prediction and salient object detection approaches. Bottom-up and fixation prediction visual attention models are researched earlier than top-down and salient object detection methods. The development history of saliency models could be divided into two stages [7]. The first wave (1998–2007) mainly addressed fixation prediction while the second wave (2008-now) mainly solved the segmentation of the most salient objects.

Fixation prediction methods are created originally to predict visual points that observers look at free-view of static nature scenes and eye movement in dynamic scenes [19]. Itti et al. [20] firstly proposed a general computational framework and psychological theories of bottom-up and fixation prediction attention based on center-surround mechanisms. This saliency visual attention model abstracts colors, intensity and orientations of many scales images and obtains many scales saliency maps and adds these maps together to form a final enhanced saliency map. Later on, many models, which

✉ Qing Zhu
zhuqing_hnu@163.com

Xianen Zhou
zhouxianen@hnu.edu.cn

Yaonan Wang
yaonan@hnu.edu.cn

Changyan Xiao
c.xiao@hnu.edu.cn

Xiao Lu
xlu_hnu@163.com

[1] National Engineering Laboratory for Robot Visual Perception and Control Technology, Hunan University, Changsha 410082, China

[2] Hunan Normal University, Changsha 410082, China

are based on bottom-up features such as image entropy [23], color contrast [26], self-information [10], spectral residual [18] and so on, have been proposed to predict eye movement or to abstract regions of interesting. However, these fixation prediction models usually could not extract entire salient objects which limits their applications in computer vision-related tasks such as object detection, image segmentation and so on.

In contrast, salient object detection methods are usually able to segment the salient object as a whole. Therefore, in the last ten years, more and more researchers focus on salient object detection methods. Inspired by some earlier bottom-up for fixation prediction, a large number of computational models are developed for detecting saliency regions. These salient region detection methods are based on low-level features like color, texture and orientation. Among them, color contrast-based saliency detection is one of the most popular methods. Liu et al. [24] and Achanta et al. [1,2] firstly defined salient object detection as a segmentation problem. Since then, most methods of the salient object detection are usually compose of two steps: calculate saliency map and segment saliency map to extract salient objects. Cannon et al. [26] denoted a region contrast-based visual attention analysis method. Inspired by the work presented in [26], Zhai et al. [37] proposed an efficient algorithm, named Luminance-based Contrast (LC), for computing the pixel-level saliency maps by using the global color contrast between image pixels. To speed up, Zhai et al. reduced the number of colors by only using luminance. However, there is a disadvantage that the distinctiveness of color information is ignored. Chen et al. [11] proposed a Histogram-based Contrast (HC) saliency detection method. There are two differences between HC and LC. Firstly, HC uses full-color space instead of luminance only. Secondly, HC applies two methods to speed up, on the one hand, quantizes each color channel to have 12 different values. On the other hand, it ignores less frequently occurring colors. Meanwhile, Region-based Contrast (RC) saliency detection method is proposed by Chen et al. Firstly, RC segments the input image into regions by a graph-based image segmentation, then calculates color contrast at region-level and defines the saliency for each region as the weighted sum of the regions contrast to all other regions in the image. To gain binary salient mask, Chen et al. [11] proposed a segmentation approach, named saliencyCut which is an iteratively run GrabCut [30]. RC which combines superpixels with color contrast and saliencyCut could achieve high precision and recall, i.e., its precision and recall are 90% and 90% on the MSRA-1000 dataset, respectively. But it is not fast because that GrabCut algorithm usually needs to execute many times. Jiang et al. [21] presented an automatic salient object segmentation algorithm which integrates bottom-up salient stimuli and object-level shape prior that a salient object has a well-defined closed boundary. Recent years, the saliency methods combining bottom-up saliency map with high-level priors are very popular. Zhang et al. [38] combined an initial prior map based on the contrast and center bias with the boundary contrast and the smoothness prior. In [40], a simple and effective salient object detection exploring both patch-level and object-level cues is described. This method merges SLIC superpixel segmentation with affinity propagation clustering to obtain the compactness map. Wang and Jiang et al. [22,32] developed a principled extension, supervised feature integration, which learns a random forest regressor to discriminatively integrate the saliency features for saliency computation. This method consists of three parts including multi-level segmentation, saliency computation in each level and multi-level saliency fusion by using a linear combinatory.

We make two discoveries by analyzing the previous research: (1) Salient object segmentation mainly consists of two parts: saliency map computation and saliency object segmentation, and most researchers paid close attention to saliency detection. However, few researchers focus on how to segment the salient objects followed saliency detection process. (2) Foreground regions often locate in the center region of the image. Their sizes are smaller than those of the background regions. Their saliency values are usually large. Background regions are homogeneous and easily connect to each other and usually close to the boundary of the image. Their sizes are usually large. Their saliency values are usually low. Motivated by these discoveries, we propose a novel saliency map segmentation strategy called SSG which mainly consists of Simple Linear Iterative Clustering superpixel segmentation (SLIC) [3], feature extraction, superpixel Region Growing (RG) , superpixel DBSCAN clustering and GrabCut [30]. The proposed method is a new fusion method which is used to get salient objects in an image. We simultaneously applied four segmentation methods, i.e., SLIC, RG, DBSCAN and GrabCut. Because many segmentation techniques such as SLIC superpixels [3], mean-shift [12], graph-based [15] segmentation and so on could be useful for eliminating background noise and reducing computation by treating each segment as a processing unit. And a better grouping to cluster an object as a whole could be useful for salient object detection. In this paper, SLIC makes similar and adjacent pixels be classified as the same superpixel. DBSCAN makes similar background regions cluster as a whole. RG groups the similar regions together as much as possible. GrabCut abstracts salient objects accurately. The main contribution of the paper is that we propose SSG which could dramatically improve the recall and maintain high accuracy.

The rest of the paper is structured as follows. The proposed approach is presented in Sect. 2. Section 3 gives the experimental results and the comparison with other approaches. Finally, the conclusion is drawn in Sect. 4.

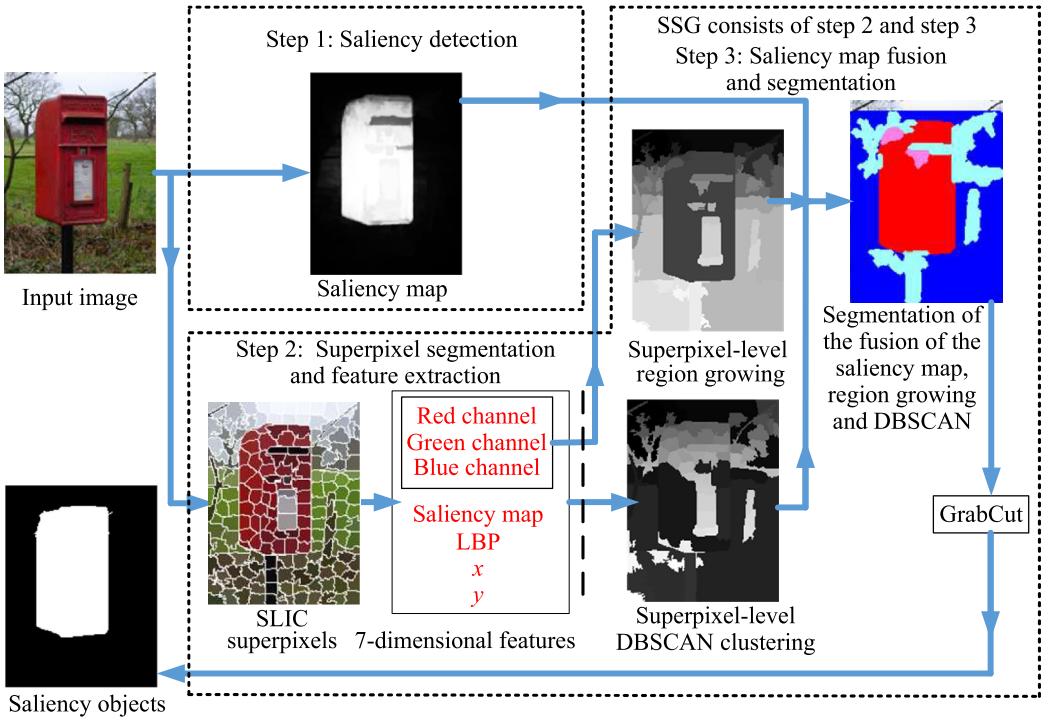

Fig. 1 The framework of our proposed salient object segmentation

2 Proposed method

GrabCut is a popular object segmentation method, but it needs to set trimap by manually. Fu et al. firstly developed an automatic implement object segmentation method which combines GrabCut with saliency region detection [17]. However, the accuracy of the method is not particularly high. To get higher accuracy, Chen et al. iteratively run GrabCut less than 4 iterations for segmenting histogram contrast and region contrast saliency map. Nonetheless, the execution time of Chen's method is longer than that of Fu's method since GrabCut implements many times. To overcome the problem, we propose a salient object segmentation called SSG consisting of two parts: superpixel segmentation and feature extraction, saliency map fusion and segmentation. The proposed SSG can be combined with any saliency detection for detecting salient objects. The block diagram of our complete scheme of salient object segmentation is briefly shown in Fig. 1. The procedures of the proposed framework are given as follows.

Step 1. Use the combination of the minimum barrier distance transform and the image boundary contrast saliency detection method to obtain the saliency map of the input image.

Step 2. Obtain superpixel regions and extract features of each superpixel. To abstract object-level information more effectively, we employ SLIC to generate a few category-independent superpixels. Taking superpixels as the minimum units, we calculate the center coordinates of x, y, the mean value of saliency map, the average value of Local Binary Pattern (LBP) [27] and the mean value of every color space channel.

Step 3. We propose superpixel region growing, superpixel DBSCAN clustering and combine both of them with GrabCut in order to segment saliency map. Firstly, we propose superpixel region growing to segment the output image of SLIC by using 3-Dimensional (3-D) color features. Secondly, we propose superpixel DBSCAN clustering to segment the output image of SLIC by employing 7-Dimensional (7-D). Then, The fusion of superpixel region growing, superpixel DBSCAN clustering and saliency map are classified into four classes. Finally, the segmentation result is fed into GrabCut to detect salient objects.

In the following, we present the three steps of our method including saliency detection, superpixel segmentation and feature extraction, and saliency map fusion and segmentation.

2.1 Saliency detection

To obtain saliency map, we apply Zhang's saliency detection algorithm [39] which combines the Minimum Barrier dis-

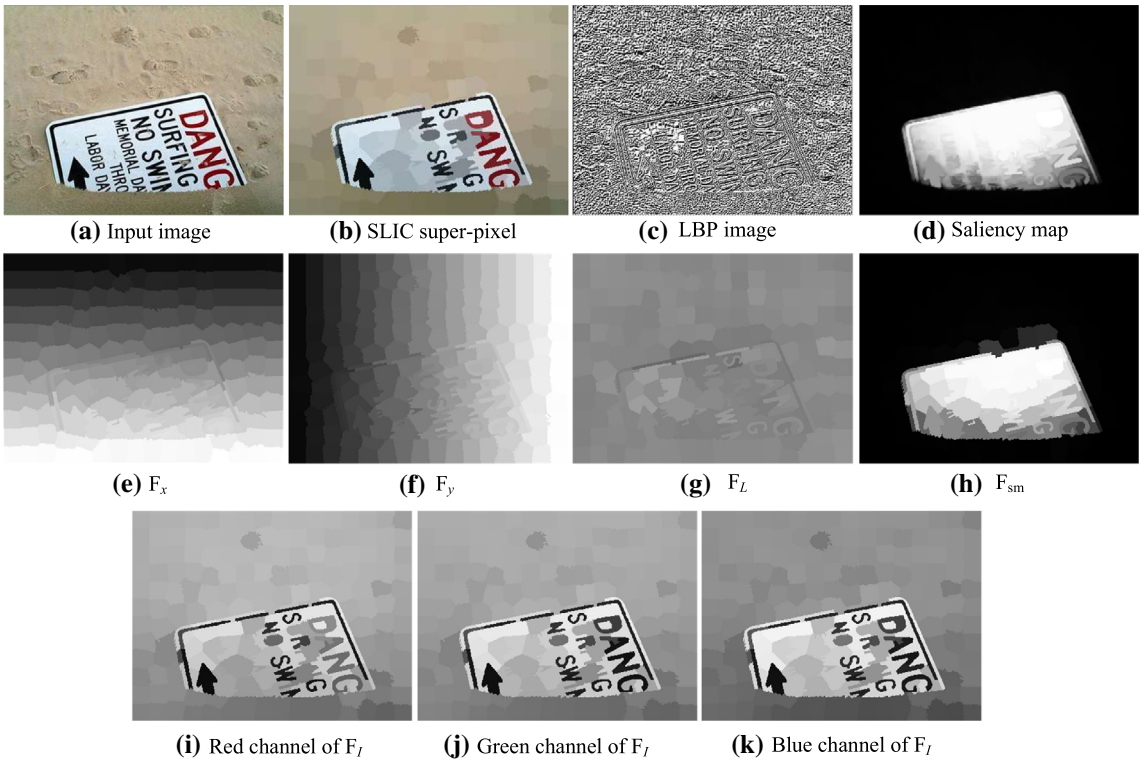

Fig. 2 Seven-dimensional features

tance transform Saliency map (MBS) with Image Boundary Contrast-based saliency map (IBC) in our proposed frame. Because Zhang's method [39] is simple and could achieve at 80 FPS even for the CPU-sequential implementation. Meanwhile, it could achieve high accuracy and recall. Note that the proposed SSG can be combined with any other saliency detection methods too.

2.2 Superpixel segmentation and feature extraction

To make similar regions cluster as a whole, we segment the input image by region growing and DBSCAN. However, the computational cost is high and the speed is slow, especially for DBSCAN, if image pixels are taken as the basic units of the input of region growing and DBSCAN clustering. To speed up, we firstly segment an input image into superpixels which are perceptually uniform regions, and use superpixels as the minimum units of the subsequent image processing. We choose SLIC superpixel segmentation algorithm [3,4] to segment input image. Because SLIC is simple, efficient and could achieve superior accuracy and boundary recall for object detection. And the GPU parallel implementation of the SLIC algorithm even achieves 250FPS [29]. Yang et al. [35] denoted that the number of superpixels is set equal to 200 that are suitable for detecting salient objects. Followed SLIC, taking superpixels as minimum units, we calculate the features of each superpixel as

$$F_x^j = \frac{1}{I_c * |R_S^j|} \sum_{i \in R_S^j} x_i \tag{1}$$

$$F_y^j = \frac{1}{I_r * |R_S^j|} \sum_{i \in R_S^j} y_i \tag{2}$$

$$F_{\text{sm}}^j = \frac{1}{255 * |R_S^j|} \sum_{i \in R_S^j} S_i \tag{3}$$

$$F_L^j = \frac{1}{255 * |R_S^j|} \sum_{i \in R_S^j} L_i \tag{4}$$

$$F_I^j = \frac{1}{255 * |R_S^j|} \sum_{i \in R_S^j} I_i \tag{5}$$

where R_S^j is the j-th superpixel and $|R_S^j|$ denotes the number of the j-th superpixel. I_c and I_r denote the width and height of the input image, respectively. F_x^j, F_y^j, F_{sm}^j, F_L^j and F_I^j are the center coordinates of x,y, the mean value of saliency map, the average value of Local Binary Pattern(LBP) [27] and the mean value of every color space channel of the j-th superpixel, respectively. x_i, y_i, S_i, L_i and I_i are the corresponding x, y coordinates, saliency map, LBP and the gray scales of color space, respectively. i, which denotes the i-th pixel, belongs to the j-th superpixel. If the input image is a color image, F_I^j has three dimensions since there are 3

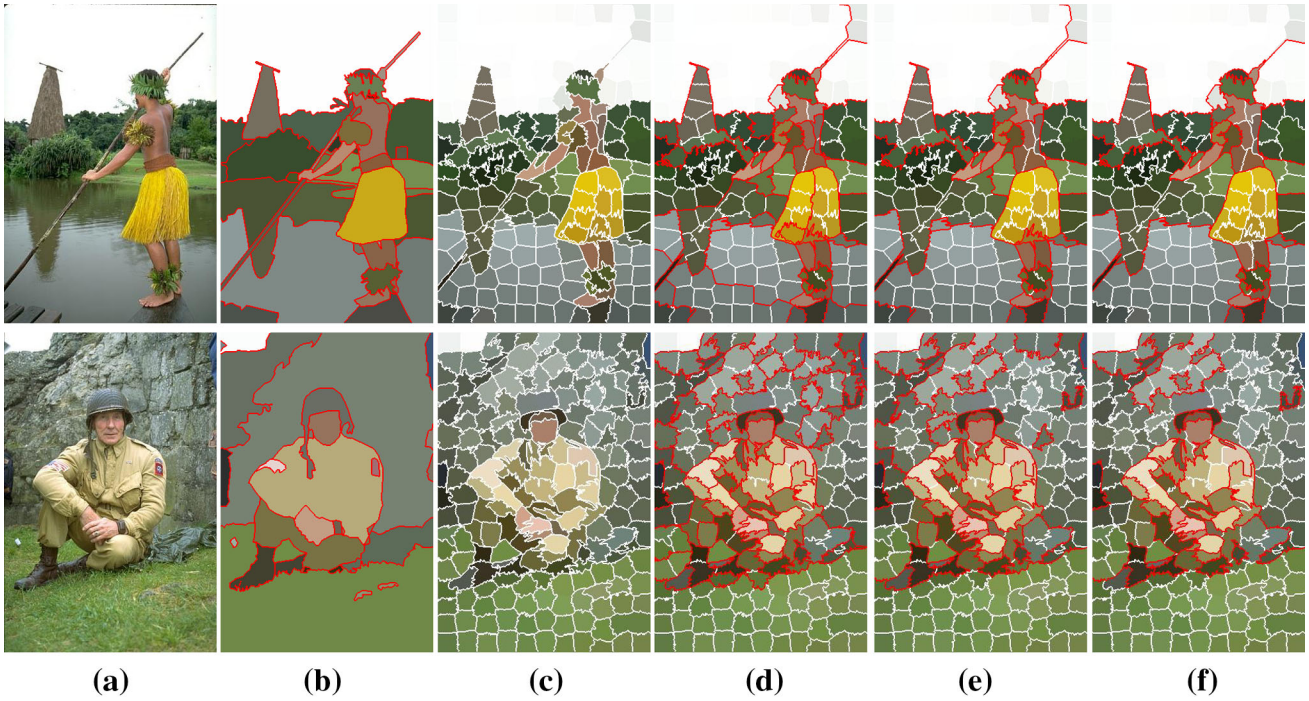

Fig. 3 Results of superpixel region growing when set different values of T_R. **a** Input images, **b** ground truth, **c** SLIC superpixels, **d** region growing segmentation results when $T_R = 500$. **e** Region growing segmentation results when $T_R = 1000$. **f** Region growing segmentation results when $T_R = 1500$. Where the red curves in **b** denote the ground truth segments' boundary. In Figure **c**, **d**, **e** and **f**, the white and red curves, respectively, denote SLIC superpixels' and region growing segments' boundary

color channels. Hence, each superpixel has seven features. For a color image, these 7-D features of superpixels could be exhibited as images, as shown in Fig. 2.

Any two superpixels belonging to the same salient object region or the same background region usually satisfy the following conditions: these superpixels are usually close to each other in space. And their saliency values, color and texture are similar. Motivated by these observations, we apply DBSCAN to segment superpixels by using 7-D features and employ region growing to segment superpixels by using 3-D color features.

2.3 Saliency map fusion and segmentation

To segment background regions accurately and to cluster similar superpixels, we propose superpixel DBSCAN and superpixel Region Growing (RG), respectively. Followed, we merge the results of RG and DBSCAN with saliency map in order to suppress the noise effectively and classify the fusion image into four states. Then, the result with four classes is fed into GrabCut, and salient objects could be got after that GrabCut is executed only one time, where we detailedly describe how to get the input of GrabCut, i.e., how to divide the input image into four states containing Obvious Background Pixel (OBP), Obvious Foreground (salient object) Pixel (OFP), Possible Background Pixel (PBP) and Possible Foreground Pixel (PFP). The third step of the proposed framework, i.e., saliency map fusion and segmentation, which contains three procedures, i.e., superpixel region growing, superpixel DBSCAN clustering and fusion and segmentation, is described in the following.

2.3.1 Superpixel region growing

Region growing [14] is a simple and fast image segmentation method based on pixel-level. It mainly involves two parts containing both the selection of an initial seed point and seed growing. Three key problems must be solved: (1) How to select initial seed point? (2) How to evaluate the similarity between the current class and the corresponding neighbors? (3) What are the stopping rules? In this paper, we develop a superpixel region growing method. Superpixel is taken as the minimal unit of region growing. The initial seed superpixel of a new class is picked orderly from the un-labeled superpixels. The distance measure of two adjacent superpixels R_i and R_j is computed as Eq. (6), where $\| * \|$ denotes the L2 norm. The growing process would stop if every superpixel has been assigned a category label.

$$d(R_i, R_j) = \| F_I^{(i)} - F_I^{(j)} \| \tag{6}$$

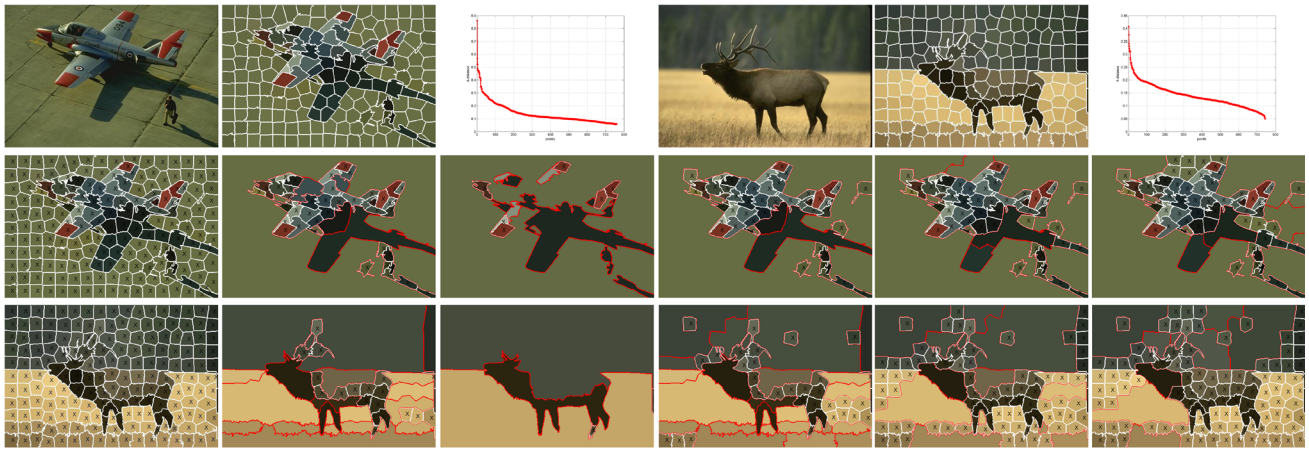

Fig. 4 Results of superpixel DBSCAN clustering when set different values of parameters and the sorted k-th distance graph (set $k = 4$). For the 1-th row figures: the 1-th and 4-th columns' figures are input images. The 2-th and 5-th columns' figures are SLIC superpixels images. The 3-th and 6-th are the sorted 4-distance graph. The 2-th and 3-th rows' figures are DBSCAN clustering results when set different values of Eps and minPts, and those superpixels classified as noise points are marked by $\times$: for 1-th column, minPts $= 3$ and Eps $= 0.05$; for 2-th column, minPts $= 3$ and Eps $= 0.15$; for 3-th column, minPts $= 3$ and Eps $= 0.25$; for 4-th column, Eps $= 0.13$ and minPts $= 3$; for 5-th column, Eps $= 0.13$ and minPts $= 4$; for 6-th column, Eps $= 0.13$ and minPts $= 5$. Where the boundaries of SLIC superpixels and DBSCAN segments are denoted by white curves and red curves, respectively

Superpixel region growing has only one threshold denoted by T_R for color similarity measurement. Larger T_R could make region growing achieve much better segmentation result, but if T_R is too larger, it may lead to over segmentation.

To further reveal the effects of T_R on the result of region growing, we take two images selected from BSD500 [6] as the testing example. The segmentation results when we set different values of T_R are shown in Fig. 3.

2.3.2 Superpixel DBSCAN clustering

To make the similar background regions group as a whole, we apply DBSCAN clustering [13] to group superpixels. The distance measure of any two superpixels p and q is calculated by

$$d(p, q) = \| F^p - F^q \| \tag{7}$$

where F is the 7-D feature of each superpixel.

DBSCAN requires two parameters: the maximum radius of the neighborhood from a core point called Eps-neighborhood radius and at least points within Eps-neighborhood radius. These two parameters are denoted by Eps and minPts, respectively. minPts is usually chosen at least 3, with minPts ≤ 2, the result is the same as of hierarchical clustering with the single link metric. Larger values are usually better for data sets with noise and will yield more significant clusters. Usually, the larger the data set, the larger the value of minPts should be selected. Eps is chosen by the sorted k-distance graph [13], and the desired parameter value is just the first point in the first valley of the sorted k-distance graph. In other words, good value of Eps is where this k-distance plot shows a strong bend. If Eps is much too small, a large part of the data will not be clustered. Whereas Eps is too large, the majority of objects will be clustered into the same cluster. More details of DBSCAN clustering algorithm could refer to [13].

To exhibit the impacts of parameters on the result of DBSCAN clustering, we set different values of Eps and minPts, DBSCAN clustering algorithm is tested on two images, the results and the corresponding sorted k-th distance graph are shown in Fig. 4.

By observing Fig. 4, we find that DBSCAN clustering algorithm can group a majority of background regions. And a great many of superpixels in the foreground are marked as noise. Because the density of these superpixels in background regions has higher consistency than that of foreground regions.

2.3.3 Fusion and segmentation

Assume that R_R^j denotes the j-th segments of region growing, the fusion of the saliency map and the results of region growing can be calculated by

$$S_R^j = \frac{1}{|R_R^j|} \sum_{i \in R_R^j} S_i \tag{8}$$

where $|R_R^j|$ is the number of pixels of the j-th region growing segment, S is saliency map.

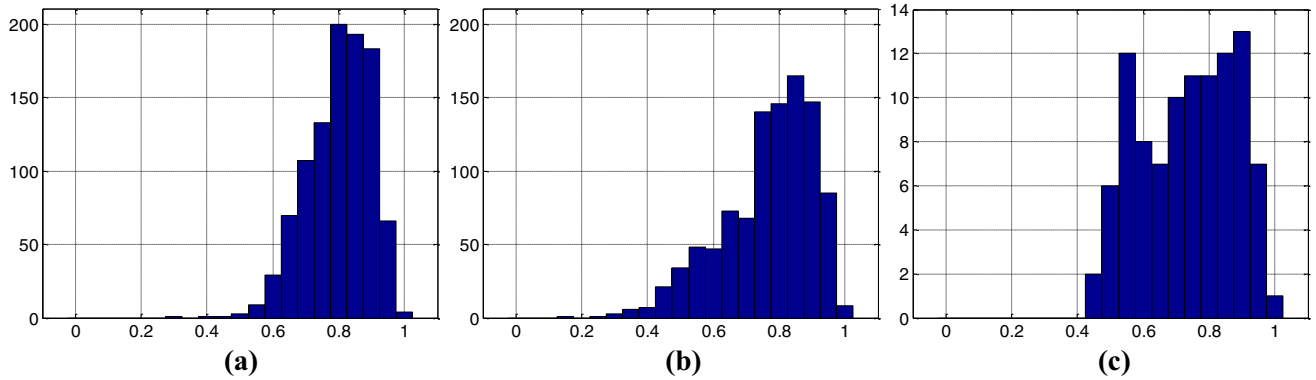

Fig. 5 Distributions of background regions' area, **a** MSRA-1000 dataset, **b** SEG1 dataset, **c** ECSSD dataset

To conveniently describe the process of salient object detection, we define a new state called Undetermined Pixel (UP) and all pixels of the input image are initialized as UP at the beginning. Salient objects usually have great saliency values. Contrary to salient objects, background regions often have low saliency values. Meanwhile, the sizes of salient objects are usually smaller than those of their surrounding background regions [36]. Therefore, the salient regions (foreground) and background regions of R_R^j could be distinguished depending to their saliency values as

$$R_{\mathrm{SRG}}^j = \begin{cases} \mathrm{OBP}, & S_R^j \le T_{\mathrm{SL}} \\ \mathrm{OFP}, & S_R^j \ge T_{\mathrm{SH}} \\ \mathrm{UP}, & T_{\mathrm{SL}} < S_R^j < T_{\mathrm{SH}} \end{cases} \tag{9}$$

where T_{SL} and T_{SH} are two segmentation thresholds which are calculated by

$$\frac{\sum_{i=0}^{T_{\mathrm{SL}}} S_A^i}{\sum_{i=0}^{255} S_A^i} = T_H \tag{10}$$

$$\frac{1 - \sum_{i=0}^{T_{\mathrm{SH}}} S_A^i}{\sum_{i=0}^{255} S_A^i} = T_L \tag{11}$$

where T_L, T_H are the ratio of salient objects' area to the total number of pixels of the image, the ratio background area to the total number of pixels of the image, respectively, i is the saliency level of saliency map, S_A^i denotes the area of the region whose saliency value is equal to i, and note that the saliency map is normalized to [0, 255] before implementing Eqs. (10) and (11). T_L and T_H are determined by the priors knowledge of background regions' sizes measured by Yildirim's method [36]. We use the MSRA-1000 [2], SEG1 [5] and (Extended Complex Scene Saliency Dataset) ECSSD [31] datasets to estimate the distribution of the sizes of the salient objects. Figure 5 shows the probability distributions of three datasets in terms of the background regions' areas.

We can see in Fig. 5 that all probability distributions resemble Gaussian distribution. And more than 90.80% images in these datasets, the ratio of the size of total background regions to the size of the total image ranges from 0.5 to 0.95. In other words, it is reasonable that T_L is set lower than 0.5 and T_H is set larger than 0.95.

After segmenting the input image according to Eq. (9), the input image is divided into three states including OBP, OFP and UP, each pixel belongs to only one of these states. And each region of the result of DBSCAN clustering consists of 3 segments marked OBG, OFG and UP. Assume that R_D^i denotes the i-th segments of DBSCAN. R_D^i consists of three regions denoted by F_{OBP}^i, F_{OFP}^i and F_{UP}^i. $|F_{\mathrm{OBP}}^i|$, $|F_{\mathrm{OFP}}^i|$ and $|F_{\mathrm{UP}}^i|$ denote the corresponding sizes. Each region of DBSCAN is segmented as Eq. (12).

$$R_{\mathrm{DR}}^i = \begin{cases} \mathrm{OBP}, \max(|F_{\mathrm{OBP}}^i|, |F_{\mathrm{OFP}}^i|, |F_{\mathrm{UP}}^i|) = |F_{\mathrm{OBP}}^i| & \text{and} \quad t \ge 50\% \\ \mathrm{OFP}, \max(|F_{\mathrm{OBP}}^i|, |F_{\mathrm{OFP}}^i|, |F_{\mathrm{UP}}^i|) = |F_{\mathrm{OFP}}^i| & \text{and} \quad t \ge 50\% \\ \mathrm{UP}, & \text{otherwise} \end{cases} \tag{12}$$

Where t is computed by

$$t = \frac{\max(|F_{\mathrm{OBP}}^i|, |F_{\mathrm{OFP}}^i|, |F_{\mathrm{UP}}^i|)}{|F_{\mathrm{OBP}}|^i + |F_{\mathrm{OFP}}^i| + |F_{\mathrm{UP}}^i|} \tag{13}$$

For the result segmented by Eq. (12), all those regions marked as UP are segmented again. Assume that R_{DR}^i = UP, and it consists of M SLIC superpixels, the m-th label of SLIC superpixel is SP^m. Superpixel is taken as the minimal unit, R_{DR}^i marked by UP could be segmented as

$$R_R^m = \begin{cases} \mathrm{PBP}, F_{\mathrm{sm}}^{\mathrm{SP}^m} \le T_{\mathrm{O}} & \text{and} \quad 1 \le m \le M \\ \mathrm{PFP}, F_{\mathrm{sm}}^{\mathrm{SP}^m} > T_{\mathrm{O}} & \text{and} \quad 1 \le m \le M \end{cases} \tag{14}$$

Where T_{O} is the Otsu threshold [28] and F_{sm} is the average saliency map obtained by Eq. (3).

Finally, the input image is divided into four states including OBP, OFP, PGP and PFP. Then, the segmentation result is fed into the GrabCut algorithm which executes only once.

GrabCut is an iterative image segmentation method based on graph cut [30]. The basic idea of GrabCut is given as follows: Firstly, GrabCut uses the labeled foreground and background pixels to build GMM model, and then employs the learned GMM to segment those un-labeled pixels. Therefore, it is necessary to label foreground and background regions to learn GMM parameters at the beginning. In this paper, we segment an image automatically by GrabCut consisting of both steps which are initialization and iterative minimization. During initialization, we use threshold segmentation method, as shown in Eqs. (8)–(14), instead of manually marking foreground and background regions in order to obtain the initial foreground and background regions automatically. The proposed method classifies the input image into four categories including OBP, OFP, PBP and PFP. Actually, only the OBP and OFP regions have impacts on the segmentation result. While PBP and PFP regions would be redefined by GrabCut algorithm. The final result of GrabCut consists of four parts containing the remained OBP, OFP and the redefined PBP, PFP. Where the remained OFP and the refined PFP regions constitute the detected object.

3 Experiments and results

In this section, we do various testing experiments to compare SSG with state-of-the-art saliency map segmentation methods and evaluate computational efficiency. The executable program of SSG used in these tests could be available.[1] Zhang's saliency detection method [39] employed in our salient object detection strategy could be available on their project website.[2]

3.1 Parameters, datasets and measures

Parameters: SSG algorithm has six parameters including the number of SLIC superpixel denoted by N_S, Eps, minPts, T_R, T_L, T_H, where Eps is determined by using the sorted k-distance graph [13] and it changes only in a small scope. So we only check the influence of the other five parameters on the final result of the salient object segmentation.

Datasets: Three benchmark datasets including MSRA-1000 [2], ECSSD [31] and SEG1 [5] are used for evaluation. The MSRA-1000 dataset includes 1000 images sampled from the first large image database for quantitative evaluation of visual attention algorithm [25], where the accurate object-contours are created by manual based on the corresponding bounding box-based ground truth database. The ECSSD dataset includes 1000 images, which are acquired from the internet and the corresponding ground truth masks are segmented by five people, with more complex scenes than many other saliency detection benchmark datasets [31]. The SEG1 dataset contains 100 images, and each image has only one saliency object [5]. And it contains a variety of images with objects that are different from their surroundings by either intensity, texture features or other low-level cues. To obtain ground truth segmentation, about 50 subjects manually segment images into two classes, foreground and background.

Measures: we apply three criteria including precision, recall and F-measure which are defined as

$$P = \frac{|S \cap G|}{|S|} \tag{15}$$

$$R = \frac{|S \cap G|}{|G|} \tag{16}$$

$$F_\beta = \frac{(1+\beta^2) \times P \times R}{\beta^2 \times P + R} \tag{17}$$

In the above three equations, P, R and F_β denote precision, recall and F-measure, respectively, S is salient object detection result which is a binary mask, G is the ground truth map, $|*|$ in Eqs. (15) and (16) denotes the sum area of masks, β^2 is set as 0.3 as suggested in previous work [2,11]. A good salient object segmentation algorithm can achieve large values of P, R and F_β.

Table 1 The results when N_S changes from 100 to 400, minPts = 3, $T_R = 714$, $T_L = 0.30$, $T_H = 0.95$ and Eps = 0.13

N_S	P (%)	R (%)	F_β (%)
100	89.45	91.03	89.81
150	90.33	91.32	90.56
200	90.44	90.95	90.56
250	90.62	91.04	90.72
300	90.71	90.56	90.74
350	90.65	90.85	90.67
400	90.90	90.63	90.84

3.2 Validation of individual modules

To show the impacts of parameters on performance, we provide the qualitative comparison with different values of 5 key parameters including N_S, minPts, T_R, T_L and T_H on the MSRA-1000 dataset, as reported in Tables 1, 2, 3, 4 and 5, respectively.

Results shown in Tables 1, 2, 3, 4 and 5 indicate that N_S, minPts, T_L and T_R give only few effects to precision, recall and F-measure. T_H is proportional to precision and recall,

[1] http://pan.baidu.com/s/1sl8YrXN, download code: 28uq.

[2] http://www.cs.bu.edu/groups/ivc/fastMBD/.

Table 2 The results when minPts changes from 3 to 9, $N_S = 400$, $T_R = 714$, $T_L = 0.30$, $T_H = 0.95$ and Eps = 0.13

minPts	$P(\%)$	$R(\%)$	$F_\beta(\%)$
3	90.90	90.63	90.84
4	90.75	90.67	90.73
5	90.74	90.76	90.75
6	90.70	90.88	90.75
7	90.74	91.00	90.80
8	90.65	91.04	90.74
9	90.89	90.89	90.89

Table 3 The results when T_R changes from 150 to 1050, $N_S = 400$, minPts = 3, $T_L = 0.05$, $T_H = 0.70$ and Eps = 0.13

T_R	$P(\%)$	$R(\%)$	$F_\beta(\%)$
150	91.05	91.20	91.08
300	90.97	91.06	90.99
450	91.06	91.16	91.09
600	91.03	90.91	91.00
750	90.93	90.54	90.84
900	90.96	90.25	90.80
1050	90.55	89.54	90.31

Table 4 The results when T_L changes from 0.05 to 0.40, $N_S = 400$, minPts = 3, $T_R = 714$, $T_H = 0.95$ and Eps = 0.13

T_L	$P(\%)$	$R(\%)$	$F_\beta(\%)$
0.05	89.94	91.61	90.32
0.15	90.41	91.28	90.61
0.20	90.58	91.05	90.69
0.25	90.68	90.87	90.72
0.30	90.90	90.63	90.84
0.35	90.94	90.50	90.84
0.40	91.03	90.40	90.88

Table 5 The results when T_H changes from 0.70 to 1.00, $N_S = 400$, minPts = 3, $T_R = 714$, $Tlow = 0.3$ and Eps = 0.13

T_H	$P(\%)$	$R(\%)$	$F_\beta(\%)$
0.70	62.45	77.97	65.45
0.75	68.36	82.00	71.09
0.80	74.58	85.83	76.91
0.85	81.10	88.69	82.73
0.90	87.23	90.73	88.01
0.95	90.90	90.63	90.84
1.00	47.98	63.50	50.85

but bigger is not always better, for instance, when it is equal to 1, the precision is 47.98% and the recall is 63.50%.

Besides, we empirically analyze the effects of each component of our proposed method and their combinations, i.e., we test the performances of different segmentation strategies which are similar to the proposed method on MSRA-1000 dataset, and the results are demonstrated as Table 6.

Table 6 The results of combining different segmentation strategies with MB saliency detection

methods' combinations	Precision (%)	Recall (%)	F-measure (%)
MB + GrabCut	91.46	84.18	89.67
MB + DBSCAN + GrabCut	92.29	87.07	91.03
MB + RG + GrabCut	89.95	88.01	89.49
MB + SSG	90.90	90.63	90.84

Where MB + SSG equates with MB + RG + DBSCAN + GrabCut. Observing in Table 6, we can discover that the precision, recall and F-measure of MB + DBSCAN + GrabCut have distinct improvement over those of MB + GrabCut. Comparing the result of MB + RG + GrabCut with that of MB+GrabCut, we can find that the recall raises to 88.01% from 84.18%. On the whole, MB + RG + DBSCAN + GrabCut can achieve higher recall which is 90.63% than any other combinations. Table 6 reveals that DBSCAN can help improve precision and recall. RG can help improve recall dramatically. When both of DBSCAN and RG are applied simultaneously, the recall is strongly promoted; meanwhile, the accuracy is maintained in a high level. The main reasons are given as follows: (1) DBSCAN clustering could make the background better separated from the total image. Because the background regions are homogeneous and easily connect to each other, the sizes of the background regions are usually large. (2) RG makes the similar regions cluster together. Hence, salient objects can be segmented much more accurately.

3.3 Qualitative comparison

We combine SSG with 7 state-of-the-art saliency detection methods including HC (Histogram Contrast) [11], RC (Region Contrast) [11], GS (Geodesic distance Superpixel) [33], MB (Minimum Barrier) [39], FASA (Fast Accuracy Size-Aware) [36], NCS (Normalized Cut-based Saliency) [16], HS (Hierarchical Saliency) [34]. We measure the performances of SSG and other saliency map segmentation methods, which are Achanta's method [2] that we named Adaptive Thresholding Segmentation (ATS), Achanta's method directly segments saliency map that we called Adaptive Thresholding Segmentation Direct (ATSD) and saliencyCut [11], on three commonly used datasets including MSRA-1000 [2], ECSSD [31] and SEG1 [5]. Each saliency detection approach is combined with four saliency map segmentation methods containing SSG, saliencyCut, ATS and ATSD. The MSRA-1000, ECSSD and SED1 datasets are fed into these methods. The corresponding results are, respectively, shown in Figs. 6, 7 and 8.

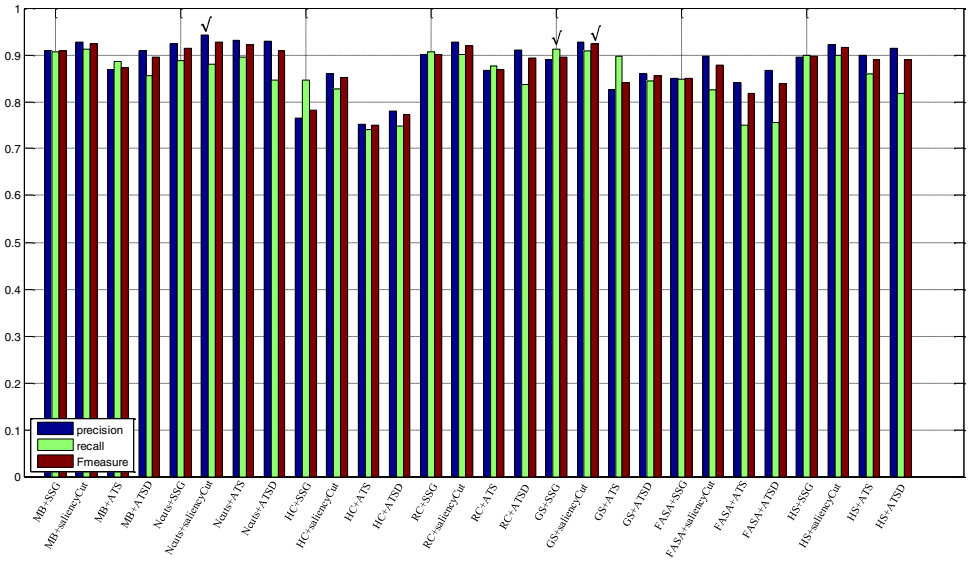

Fig. 6 Quantitative comparisons of saliency object detection on MSRA-1000. For SSG, set $N_S = 400$, minPts $= 3$, $T_R = 714$, $T_L = 0.30$, $T_H = 0.95$ and Eps $= 0.13$

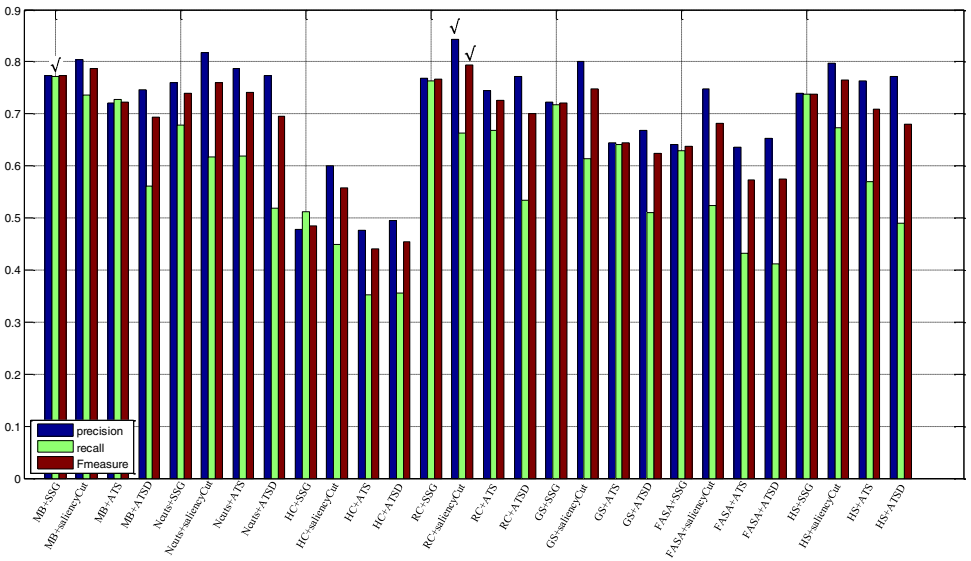

Fig. 7 Quantitative comparisons on ECSSD. For SSG, set $N_S = 400$, minPts $= 8$, $T_R = 500$, $T_L = 0.20$, $T_H = 0.97$ and Eps $= 0.10$

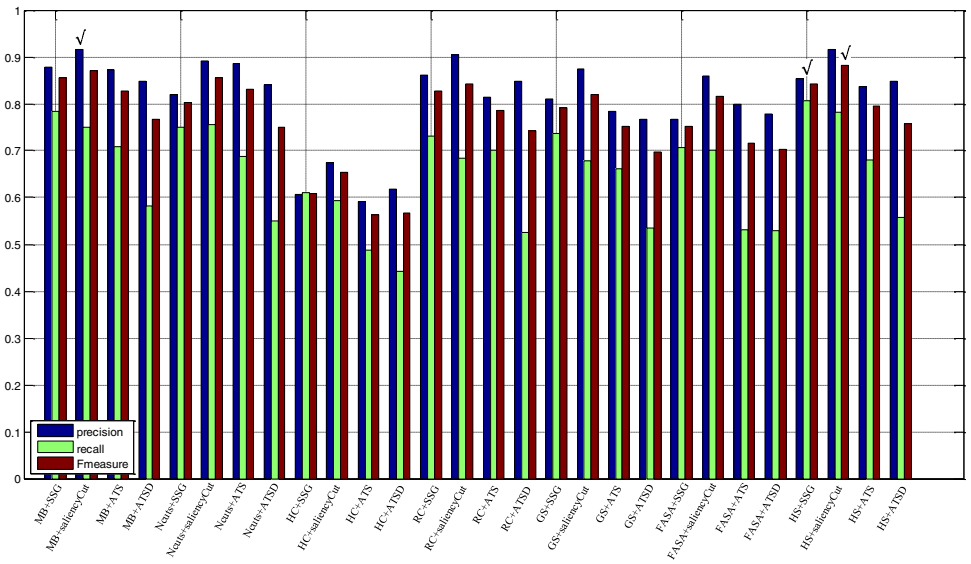

Fig. 8 Quantitative comparisons of saliency object detection on SED1. For SSG, set $N_S = 400$, minPts $= 3$, $T_R = 714$, $T_L = 0.30$, $T_H = 0.95$ and Eps $= 0.13$

Table 7 Parameters settings and results of MB + SSG and MB+saliencyCut on MSRA-1000

methods' combinations	T_L	T_H	Eps	minPts	T_R	P (%)	R (%)	F_β (%)
MB + GrabCut	0.40	0.97	0.10	9	400	91.56	91.30	91.50
	0.30	0.97	0.10	9	400	91.29	91.60	91.36
	0.30	0.97	0.10	9	200	91.27	**91.91**	91.42
	0.40	0.95	0.13	9	450	91.06	91.16	91.06
MB + saliencyCut	–	–	–	–	–	**92.80**	91.27	**92.44**

The bolded values denote the best results

Table 8 Parameters settings and results of MB + SSG and MB + saliencyCut on ECSSD

methods' combinations	T_L	T_H	Eps	minPts	T_R	P (%)	R (%)	F_β (%)
MB + GrabCut	0.40	0.97	0.10	9	400	78.62	74.77	77.77
	0.40	0.97	0.13	9	400	78.73	74.45	77.70
	0.40	0.97	0.10	3	200	78.92	74.78	77.92
	0.40	0.97	0.10	9	200	79.19	**75.19**	78.23
MB + saliencyCut	–	–	–	–	–	**80.48**	73.61	**78.79**

The bolded values denote the best results

Table 9 Parameters settings and results of MB + SSG and MB + saliencyCut on SED1

methods' combinations	T_L	T_H	Eps	minPts	T_R	P(%)	R(%)	F_β(%)
MB + GrabCut	0.40	0.97	0.13	8	400	90.55	76.89	86.98
	0.40	0.97	0.10	9	300	90.86	76.76	87.16
	0.40	0.97	0.10	12	200	91.41	**77.23**	**87.69**
	0.40	0.99	0.10	9	100	**91.79**	75.96	87.58
MB + saliencyCut	–	–	–	–	–	91.53	75.09	87.13

The bolded values denote the best results

The results shown in Figs. 6, 7 and 8 reveal that: on the MSRA-1000 dataset, for the same saliency detection method, no matter which segmentation algorithm is applied, the differences of precision, recall and F-measure between different segmentation strategies are small, the phenomenon contraries to that of the ECSSD and SED1 datasets, because the background regions of images of the MSRA-1000 dataset are much simpler than those of images of the ECSSD and SED1 datasets. For MSRA-1000, ECSSD and SED1 datasets, the highest recall methods are GS+SSG, MB+SSG, HS+SSG, respectively, and the corresponding values of recall are 91.27%, 77.21% and 80.70%; meanwhile, the corresponding values of precision and F-measure are greater than the vast majority of other methods. In other words, the proposed SSG can improve the recall dramatically. Meanwhile, it can maintain the high precision and F-measure.

Besides, observing Figs. 6, 7 and 8, we also find that the precision, recall and F-measure of MB + SSG are close to those of MB + saliencyCut. To further verify that whether the precision, recall and F-measure of the proposed method could exceed those of saliencyCut, we further fine-tune the parameters of the proposed method, and do experiments on MSRA-1000, ECSSD and SED1 datasets. For fair comparison, the input parameter $N_S = 400$, other input parameters settings of the proposed method and the corresponding results of MB + SSG and MB + saliencyCut are shown in Tables 7, 8 and 9, in which the bolded values indicate the best results.

The results shown in Tables 7, 8 and 9 give the facts that: the proposed method could achieve higher recall than that of saliencyCut. Especially, on the SED1 dataset, the precision, recall and F-measure of the proposed method are higher than those of saliencyCut. On the MSRA-1000 and ECSSD datasets, the recall of the proposed method is obviously higher than that of saliencyCut. And the corresponding precision and F-measure decrease only a little.

3.4 Computational efficiency

To compare the performance of our proposed method in terms of the average running time with the current most competitive methods, two group testing experiments are done. The average computation time, which does not include the time consumed by computing saliency map and is measured in milliseconds (ms), is acquired on an Intel Core i5-4210U, 1.7-2.4 GHz and 6 GB RAM.

On the one hand, MB saliency detection is combined with SSG, saliencyCut, ATS and ATSD segmentation methods and these combinations are tested on the MSRA-1000, ECSSD and SED1 datasets. The execution time is shown in Table 10.

Table 10 Comparison of the average running time using different segmentation methods on MSRA-1000, ECSSD and SED1 datasets

methods' combinations	MSRA-1000 (ms)	ECSSD (ms)	SED1 (ms)
$MB + SSG$	539.87	641.42	428.13
$MB + saliencyCut$	749.16	1010.80	669.08
$MB + ATS$	3476.28	4061.62	2408.74
$MB + ATSD$	4.44	4.82	7.85

Table 11 Comparison of the average running time using different saliency detection methods on MSRA-1000 dataset

methods' combinations	MSRA-1000 (ms)
SSG+MB	539.87
SSG+Ncuts	515.57
SSG+HC	510.32
SSG+RC	511.40
SSG+GS	508.23
SSG+FASA	472.38
SSG+HS	509.90
saliencyCut+MB	747.16
saliencyCut+Ncuts	579.13
saliencyCut+HC	823.28
saliencyCut+RC	705.60
saliencyCut+GS	668.40
saliencyCut+FASA	805.39
saliencyCut+HS	710.28

As can be seen, the execution time of SSG is much smaller than that of saliencyCut. Because GrabCut in SSG executes only one time. Meanwhile, SLIC, DBSCAN and RG algorithms have high efficacy. However, the GrabCut in saliencyCut is usually implemented 4 times. ATS is the slowest segmentation method since the speed of mean-shift segmentation is very slow.

On the other hand, SSG and saliencyCut segmentation approaches are combined with seven saliency detection methods, respectively. These combinations are tested on MSRA-1000 dataset. The execution time is reported in Table 11.

Table 11 shows that SSG is much faster than saliencyCut, no matter which saliency detection algorithm is employed to combine with SSG.

4 Conclusion

In this paper, we have proposed a new rapid and efficient salient object segmentation framework which mainly consists of the proposed SSG segmentation and saliency detection. To speed up the proposed method, SLIC superpixel segmentation is firstly used for the input image. To improve precision and recall, we combine region growing, DBSCAN clustering with GrabCut. The proposed approach has been validated on three public datasets. The experimental results revealed that our proposed method achieves good performance in terms of precision, recall and F-measure. Especially, SSG can improve recall dramatically. Meanwhile, it can maintain precision and F-measure in a high level, and it has high efficacy.

Acknowledgements This work was supported by the National Science Foundation of China (61573134, 61703155), the National Science and Technology Support Program (2015BAF13B00) and the Innovation Project of Postgraduate Student in Hunan Province, China (CX2017B108).

References

1. Achanta, R., Estrada, F., Wils, P., Süsstrunk, S.: Salient region detection and segmentation. Computer Vision Systems **5008**, 66–75 (2010)
2. Achanta, R., Hemami, S., Estrada, F., Susstrunk, S.: Frequency-tuned salient region detection. In: IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009. pp. 1597–1604 (2009)
3. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: Slic superpixels. Epfl (2010)
4. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Susstrunk, S.: Slic superpixels compared to state-of-the-art superpixel methods. IEEE Trans. Pattern Anal. Mach. Intell. **34**(11), 2274–2282 (2012)
5. Alpert, S., Galun, M., Basri, R., Brandt, A.: Image segmentation by probabilistic bottom-up aggregation and cue integration. In: IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR '07. pp. 1–8 (2007)
6. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **33**(5), 898–916 (2011)
7. Borji, A., Cheng, M.M., Jiang, H., Li, J.: Salient object detection: a survey. Eprint Arxiv **16**(7), 3118 (2014)
8. Borji, A., Itti, L.: State-of-the-art in visual attention modeling. IEEE Trans. Pattern Anal. Mach. Intell. **35**(1), 185–207 (2012)
9. Borji, A., Sihite, D.N., Itti, L.: Salient object detection: a benchmark. In: European Conference on Computer Vision, pp. 414–429 (2012)
10. Bruce, N.D.B., Tsotsos, J.K.: Saliency based on information maximization. In: International Conference on Neural Information Processing Systems, pp. 155–162 (2005)
11. Cheng, M.M., Zhang, G.X., Mitra, N.J., Huang, X., Hu, S.M.: Global contrast based salient region detection. In: Computer Vision and Pattern Recognition, pp. 409–416 (2011)
12. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. **24**(5), 603–619 (2002)
13. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise (1996)
14. Fan, J., Zeng, G., Body, M., Hacid, M.S.: Seeded region growing: an extensive and comparative study. Pattern Recognit. Lett. **26**(8), 1139–1156 (2005)
15. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. Int. J. Comput. Vis. **59**(2), 167–181 (2004)

16. Fu, K., Gong, C., Gu, I.Y., Yang, J.: Normalized cut-based saliency detection by adaptive multi-level region merging. IEEE Trans. Image Process. **24**(12), 5671–5683 (2015)
17. Fu, Y., Cheng, J., Li, Z., Lu, H.: Saliency cuts: an automatic approach to object segmentation. In: International Conference on Pattern Recognition, pp. 1–4 (2008)
18. Hou, X., Zhang, L.: Saliency detection: a spectral residual approach. In: IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR '07, pp. 1–8 (2007)
19. Huo, L., Jiao, L., Wang, S., Yang, S.: Object-level saliency detection with color attributes. Pattern Recognit. **49**, 162–173 (2016)
20. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. Pattern Anal. Mach. Intell. **20**(11), 1254–1259 (1998)
21. Jiang, H., Wang, J., Yuan, Z., Liu, T., Zheng, N., Li, S.: Automatic salient object segmentation based on context and shape prior. In: British Machine Vision Conference (2011)
22. Jiang, H., Wang, J., Yuan, Z., Wu, Y., Zheng, N., Li, S.: Salient object detection: A discriminative regional feature integration approach. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2083–2090 (2013). https://doi.org/10.1109/CVPR.2013.271
23. Kadir, T., Brady, M.: Saliency, scale and image description. Int. J. Comput. Vis. **45**(2), 83–105 (2001)
24. Liu, T., Sun, J., Zheng, N.N., Tang, X., Shum, H.Y.: Learning to detect a salient object. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007). https://doi.org/10.1109/CVPR.2007.383047
25. Liu, T., Yuan, Z., Sun, J., Wang, J., Zheng, N., Tang, X., Shum, H.Y.: Learning to detect a salient object. IEEE Trans. Pattern Anal. Mach. Intell. **33**(2), 353 (2011)
26. Ma, Y.F., Zhang, H.J.: Contrast-based image attention analysis by using fuzzy growing. In: Eleventh ACM International Conference on Multimedia, pp. 374–381 (2003)
27. Ojala, T., Pietikainen, M., Harwood, D.: Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In: Iapr International Conference on Pattern Recognition, 1994. Vol. 1—Conference A: Computer Vision and Image Processing, pp. 582–585 vol. 1 (2002)
28. Otsu, N., Ostu, N., Nobuyuki, O.: A threshold selection method from gray-level histograms. IEEE Transactions on Systems Man and Cybernetics **9**(1), 62–66 (1979)
29. Ren, C.Y., Prisacariu, V.A., Reid, I.D.: gslicr slic superpixels at over 250hz. Computer Science (2015)
30. Rother, C., Kolmogorov, V., Blake, A.: "grabcut": interactive foreground extraction using iterated graph cuts. In: ACM SIGGRAPH, pp. 309–314 (2004)
31. Shi, J., Yan, Q., Li, X., Jia, J.: Hierarchical image saliency detection on extended cssd. IEEE Trans. Pattern Anal. Mach. Intell. **38**(4), 717–729 (2016)
32. Wang, J., Jiang, H., Yuan, Z., Cheng, M.M., Hu, X., Zheng, N.: Salient object detection: a discriminative regional feature integration approach. Int. J. Comput. Vision **123**(2), 251–268 (2017). https://doi.org/10.1007/s11263-016-0977-3
33. Wei, Y., Wen, F., Zhu, W., Sun, J.: Geodesic saliency using background priors. In: European Conference on Computer Vision, pp. 29–42 (2012)
34. Yan, Q., Xu, L., Shi, J., Jia, J.: Hierarchical saliency detection. In: Computer Vision and Pattern Recognition, pp. 1155–1162 (2013)
35. Yang, C., Zhang, L., Lu, H., Ruan, X., Yang, M.H.: Saliency detection via graph-based manifold ranking. In: Computer Vision and Pattern Recognition, pp. 3166–3173 (2013)
36. Yildirim, G., Süsstrunk, S.: FASA: Fast, Accurate, and Size-Aware Salient Object Detection. Springer, Berlin (2015)
37. Zhai, Y., Shah, M.: Visual attention detection in video sequences using spatiotemporal cues. In: ACM International Conference on Multimedia, pp. 815–824 (2006)
38. Zhang, H., Xu, M., Zhuo, L., Havyarimana, V.: A novel optimization framework for salient object detection. Vis. Comput. **32**(1), 31–41 (2016)
39. Zhang, J., Sclaroff, S., Lin, Z., Shen, X., Price, B., Mech, R.: Minimum barrier salient object detection at 80 fps. In: IEEE International Conference on Computer Vision, pp. 1404–1412 (2016)
40. Zhang, Q., Lin, J., Li, W., Shi, Y., Cao, G.: Salient object detection via compactness and objectness cues. Vis. Comput. **1**, 1–17 (2017)

Xianen Zhou is pursuing the Ph.D. degree in Hunan University, Changsha, China. He received the M.S. degree in Circuits and Systems from East China institute of technology, in 2013. His research interests include real-time image processing and pattern recognition. Email: zhouxianen@hnu.edu.cn

Yaonan Wang is a Professor of Electrical and Information Engineering, Hunan University. He received Ph.D. degree in electrical engineering from Hunan University, in 1994. He was a Senior Humboldt Fellow in Germany from 1998 to 2000. His research interests include intelligent control, image processing and computer vision for industrial applications. Email: yaonan@hnu.edu.cn

Qing Zhu is an associate professor of Electrical and Information Engineering, Hunan University. She received the Ph.D. degree in electrical engineering from Hunan University, in 2008. Her research interests include voice and image processing, network and communication technology. Email: zhuqing_hnu@163.com

Changyan Xiao is a Professor of Electrical and Information Engineering, Hunan University. He received the Ph.D. degree in biomedical engineering from Shanghai Jiao Tong University, in 2005. He was a Visiting Postdoctoral Researcher with the Division of Image Processing, Leiden University Medical Center, Leiden, The Netherlands, from 2008 to 2009. His research interests include medical imaging and instruments. Email: c.xiao@hnu.edu.cn

Xiao Lu is a lecturer of Hunan Normal University. She received the Ph.D. degree in control science and engineering from Hunan University, in 2016. Her research interests include machine vision, pattern recognition and machine learning. xlu_hnu@163.com