Noname manuscript No.
(will be inserted by the editor)

Adaptive Saliency Cuts

Yuantian Wang · Tongwei Ren ✉ · Sheng-Hua Zhong · Yan Liu · Gangshan Wu

Received: date / Accepted: date

Abstract Saliency cuts aims to segment salient objects from a given saliency map. The existing saliency cuts methods are fixed to the input cues. It limits their performance when the input cues are changed. In this paper, we propose a novel saliency cuts method named adaptive saliency cuts, which takes advantage of all the input cues in a unified framework and adjusts its components adaptively. Given a saliency map, we first generate segmentation seeds with adaptive triple thresholding. Next, we extend GrabCut by combining different input cues, and use it to generate a rough-labeled map of salient objects. Finally, we refine the boundaries of the salient objects with adaptive initialized segmentation, and produce an accurate binary mask. To the best of our knowledge, this method is the first adaptive saliency cuts method for different input cues. We validated the proposed method on MSRA10K and NJU2000. The experimental results demonstrate that our method outperforms the state-of-the-art methods.

Yuantian Wang
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
E-mail: wangyt@smail.nju.edu.cn

Tongwei Ren
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
E-mail: rentw@nju.edu.cn

Sheng-Hua Zhong
College of Computer Science and Software Engineering, Shenzhen University, China
E-mail: csshzhong@szu.edu.cn

Yan Liu
Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China
E-mail: csyliu@comp.polyu.edu.hk

Gangshan Wu
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
E-mail: gswu@nju.edu.cn

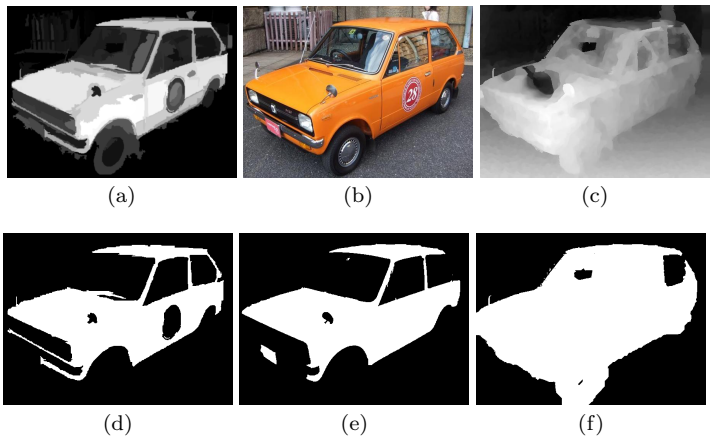

Fig. 1: An example of saliency cuts using different input cues. (a) Saliency map. (b) Color cue. (c) Depth cue. (d) Saliency cuts result only using saliency map. (e) Saliency cuts result using saliency map and color cue. (f) Saliency cuts result using saliency map, color cue and depth cue. The quality of saliency cuts results is ordered as (d)<(e)<(f).

Keywords Saliency cuts · segmentation seeds generation · rough-labeled map generation · object boundary refinement · adaptive GrabCut

1 Introduction

Large-scale multimedia data is generated every day in heterogeneous spaces, such as from social network [1] and surveillance [2]. It brings great challenges in semantically analyzing and understanding the data [3]. As the dominant modality of all the multimedia data, visual data plays an important role in recording and transferring information [4]. To visual data, segmenting the objects, especially the salient objects which may attract viewers' attention, from background is a fundamental of its semantic analysis and understanding [5].

Object segmentation has been widely studied in past decades, which aims to distinguish objects from background on pixel level in images and videos [6]. It is used in numerous applications, such as object recognition [7], detection [8], retrieval [9–11], action recognition [12, 13], and image annotation [14, 15]. As a special task in object segmentation [16–19], saliency cuts [20] aims to automatically segment salient objects from a given saliency map, which is generated by saliency detection algorithms [14, 21]. The existing saliency cuts methods can be roughly classified into two categories, according to the usage of saliency map. One category generates segmentation results from the saliency value or luminance of saliency map [22, 23]; while the other category extracts segmentation seeds from saliency map and feeds seeds to semi-supervised segmentation methods [20, 24].

An obvious limitation of current saliency cuts methods is that their performance cannot be adaptively improved when more input cues are supplemented. Figure 1 shows an example of saliency cuts using different input

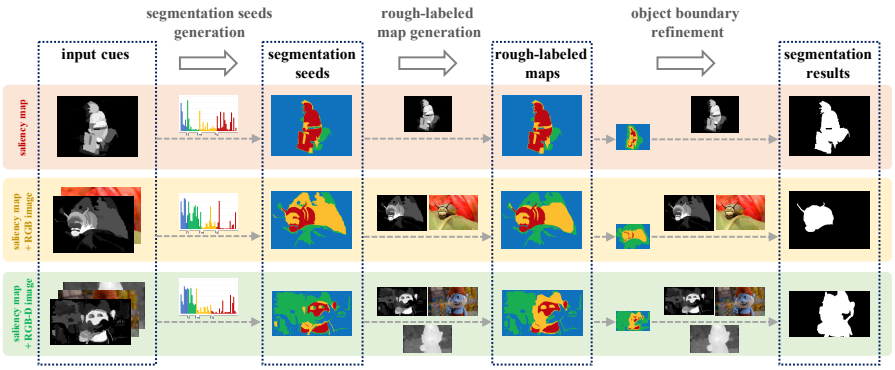

Fig. 2: An overview of our proposed method. For different input cues, namely saliency map only, saliency map and RGB image, and saliency map and RGB-D image, we first generate segmentation seeds using adaptive triple thresholding. Next, we feed these segmentation seeds to adaptive GrabCut to generate rough-labeled map. Finally, we refine the object boundaries with adaptive initialized segmentation to produce accurate segmentation results.

cues. Three saliency cuts results with different input cues, namely saliency map (Fig. 1 (a)), color cue (Fig. 1 (b)) and depth cue (Fig. 1 (c)), are shown in Fig. 1 (d) - (f). Specifically, the saliency cuts result in Fig. 1 (d) uses saliency map only; the one in Fig. 1 (e) uses saliency map and color cue; and the one in Fig. 1 (f) uses all the three cues. It shows that the saliency cuts results improve along with the supplement of input cues. Hence, it is important to make full use of all the input cues to improve the performance in saliency cuts.

Based on the above observation, we propose a novel saliency cuts method named *Adaptive Saliency Cuts* (ASC). Figure 2 shows an overview of the proposed method. We first use adaptive triple thresholding algorithm [25] to generate segmentation seeds from a given saliency map. Then, we feed the segmentation seeds together with different cues to adaptive GrabCut to generate a rough-labeled map of salient objects. Finally, we refine the boundaries of the salient objects and generate an accurate binary mask. Here, the "Adaptive" in the name of our proposed method has double meaning. First, our method can handle different input cues in a unified framework and take advantage of all the input cues. Second, our method can adjust all the components adaptively, namely segmentation seeds generation, rough-labeled map generation and object boundary refinement, to improve the performance of saliency cuts. We validated our method on two datasets, MSRA10K [24] and NJU2000 [26], which are the largest RGB image dataset and the largest RGB-D image dataset for salient object detection, respectively. The experimental results show that our method outperforms the state-of-the-art saliency cuts methods on different input cues.

Some preliminary results of our method were proposed in [25, 27], which presented the studies of saliency cuts on RGB images and RGB-D images, respectively. In this paper, we propose a unified framework for saliency cuts to make full use of different input cues, *i.e.*, only saliency map and

the combination of saliency map and RGB/RGB-D image. Inspired by [25] and [27], we construct the framework with several common components, namely segmentation seeds generation, rough-labeled map generation and object boundary refinement. Nevertheless, different to our previous works, we discuss the influences of different input cues and improve each component to handle the different input cues adaptively. For example, we extend the energy function of GrabCut in rough-labeled map generation to exploit all the input cues. Furthermore, we provide more comprehensive validation of the proposed method on two largest salient object detection datasets: MSRA10K and NJU2000.

Our contributions mainly include:

- We propose a novel saliency cuts method, which can adaptively handle different input cues with a unified framework and adjust all its components adaptively.
- We validate the proposed method on the largest salient object detection datasets for RGB images and RGB-D images. It shows that our method is superior to the state-of-the-art methods.

The rest of the paper is organized as follows. We briefly review the typical methods in object segmentation and saliency cuts in Section 2. Then, we introduce our method according to different input cues, namely saliency map only, saliency map and RGB image, and saliency map and RGB-D image in Section 3. The detailed validation and analysis of the experiments are presented in Section 4. Finally, we conclude our work in Section 5.

2 Related work

2.1 Object segmentation

The existing object segmentation methods can be summarized from different viewpoints. In this subsection, we roughly review object segmentation viewpoints in three aspects.

According to the requirement of user interaction, current object segmentation can be classified into two categories, namely automatic object segmentation and interactive object segmentation [28]. The former requires no user interaction in object segmentation, but it is usually difficult in localizing objects; the latter includes the user in the procedure of segmentation and analyzes the user intention from their interactions.

According to segmentation result representation, current object segmentation methods can also be classified into two categories, namely boundary based methods [29] and region based methods [30]. The former extracts an object by tracing its contour based on image properties; the latter models image content on regions, which considers both region statistics and inter-regional similarities.

Moreover, according to the dependence of training data, the existing object segmentation methods can be classified into three categories, namely

supervised methods, semi-supervised methods and unsupervised methods [20]. Supervised methods require prior knowledge from manually labeled training data [31]. Semi-supervised methods require human interactions to provide segmentation seeds [30]. Both supervised methods and semi-supervised methods are time and labor consuming, which limits their applications in practice. In contrast, unsupervised methods can generate segmentation without any training or manually-labeling process, which are more preferred in real applications [32].

Recently, some new advances appear in object segmentation. Co-segmentation focuses on extracting the same object from a set of images, which can analyze the representation of an object from its appearance in different situations and usually obtain better performance than those methods on a single image [33]. Specifically, object segmentation on stereo images can be treated as a special co-segmentation, in which the number of images is limited to two and object appearances on two views have high consistency and more strict constraints [34]. Another progress on object segmentation is combining multiple modalities, such as depth [35]. Multi-modal based methods can integrate the object representations on different modalities and improve the effectiveness and efficiency of object segmentation.

2.2 Saliency cuts

Saliency cuts methods utilize saliency map as the primary input cue, in which original images or videos are usually ignored or used for refinement. Otsu *et al.* [22] produce segmentation results using thresholds from gray-level histograms of saliency maps. Achanta *et al.* [23] segment salient objects from the saliency value and luminance of saliency map. Fu *et al.* [20] generate saliency cuts results via professional labels. Cheng *et al.* use a fixed threshold to binarize the saliency maps and produce results from iterative GrabCut calculation [24]. Banica *et al.* [36] segment video object via salient segment chain composition. The limitation of current saliency cuts methods is that they are fixed to the input cues and they cannot improve their performance adaptively when more input cues are supplemented.

3 Adaptive Saliency Cuts

Because our method deals with different input cues with a unified framework, we first present the details of the proposed method under the simplest case, *i.e.*, only using saliency map, and then present the additional or changed processing under more input cues. To distinguish the differences in input cues, we use ASC_S, ASC_{SC} and ASC_{SCD} to indicate our method in the cases using only saliency map, saliency map and RGB image, and saliency map and RGB-D image as the input, respectively.

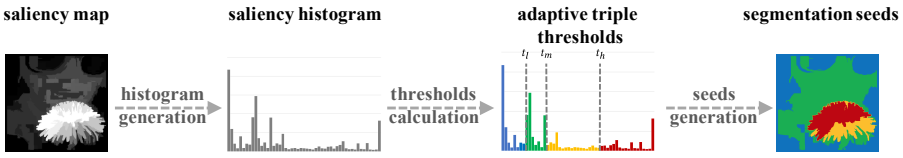

Fig. 3: Segmentation seeds generation via adaptive triple thresholding.

3.1 Only using saliency map

3.1.1 Segmentation seeds generation via adaptive triple thresholding

Inspired by [25], we generate segmentation seeds from saliency map using adaptive triple thresholding. As compared to Otsu multilevel thresholding method [37], adaptive triple thresholding method includes twice optimizations to obtain better performance.

Figure 3 shows an overview of segmentation seeds generation via adaptive triple thresholding. Given a saliency map, we first propose a histogram from saliency map, which is in the value range of $[0, 255]$ in our experiments. Then, we calculate a threshold t_m to divide saliency map M^s into two parts: $M^s = \Omega_b^s \cup \Omega_f^s$. Here, Ω_b^s and Ω_f^s denote background and foreground, which contain the pixels whose saliency values are in the value range of $[1, t_m]$ and $[t_m+1, H]$, respectively. Assume n is the number of pixels on M^s, and n_b and n_f are the numbers of pixels on Ω_b^s and Ω_f^s, respectively. t_m is calculated as follows:

$$t_m = \arg\max\{\omega_b\omega_f(\mu_b - \mu_f)^2\}, \tag{1}$$

where ω_b and ω_f are the weights of Ω_b^s and Ω_f^s, which equal n_b/n and n_f/n, respectively; μ_b and μ_f are the average saliency value of Ω_b^s and Ω_f^s, respectively.

We further calculate t_l and t_h to divide background Ω_b^s and foreground Ω_f^s into two sub-parts: $\Omega_b^s = \Omega_{cb}^s \cup \Omega_{pb}^s$ and $\Omega_f^s = \Omega_{pf}^s \cup \Omega_{cf}^s$. Here, Ω_{cb}^s, Ω_{pb}^s, Ω_{pf}^s and Ω_{cf}^s denote certain background, probable background, probable foreground and certain foreground, which contain the pixels whose saliency values are in the value range of $[1, t_l]$, $[t_l + 1, t_m]$, $[t_m + 1, t_h]$ and $[t_h + 1, H]$, respectively. Obviously, the intersection of each two in Ω_{cb}^s, Ω_{pb}^s, Ω_{pf}^s and Ω_{cf}^s is $\emptyset$. Assume n_{cb}, n_{pb}, n_{pf} and n_{cf} are the numbers of pixels on Ω_{cb}^s, Ω_{pb}^s, Ω_{pf}^s and Ω_{cf}^s, respectively. t_l and t_h are calculated as follows:

$$\{t_l, t_h\} = \arg\max\{\omega_{cb}\omega_{pb}(\mu_{cb} - \mu_{pb})^2 + \omega_{cf}\omega_{pf}(\mu_{cf} - \mu_{pf})^2\}, \tag{2}$$

where ω_{cb}, ω_{pb}, ω_{pf} and ω_{cf} are the weights of Ω_{cb}^s, Ω_{pb}^s, Ω_{pf}^s and Ω_{cf}^s, which equal n_{cb}/n, n_{pb}/n, n_{pf}/n and n_{cf}/n, respectively; μ_{cb}, μ_{pb}, μ_{pf} and μ_{cf} are the average saliency value of Ω_{cb}^s, Ω_{pb}^s, Ω_{pf}^s and Ω_{cf}^s, respectively.

3.1.2 Rough-labeled map generation via adaptive GrabCut

GrabCut is a pixel-level interactive object segmentation algorithm based on mini-cut optimization [30].

After we utilize the segmentation seeds generated in Section 3.1.1 as the predefined labels of GrabCut algorithm, the energy function $E(L, K, \theta, Z)$ is defined as follows:

$$E(L, K, \theta, Z) = U(l_i, k_i, \theta, z_i) + V(L, Z), \tag{3}$$

where L is the label set; K is the parameter set of GMM model on saliency map; θ is the gray histogram of foreground or background on saliency map; Z is the saliency value sets of saliency map; $U(l_i, k_i, \theta, z_i)$ is the data term; $V(L, Z)$ is the smooth term, which is calculated as follows:

$$V(L, Z) = \gamma \sum_{(p_m, p_n) \in C} [l_n \neq l_m] \exp -\lambda D(z_m, z_n)^2, \tag{4}$$

where constant γ equals 50 [38]; C is the set of pairs of neighboring pixels; $\lambda = (2\langle (z_m - z_n)^2 \rangle)^{-1}$ and $\langle \cdot \rangle$ in λ denotes expectation over an colorful image; $D(z_m, z_n)$ denotes the Euclidean distance between pixels p_m and p_n, which is defined as follows:

$$D(z_m, z_n) = ||z_m - z_n||, \tag{5}$$

where z_m and z_n are the saliency value of pixel p_m and p_n on saliency map.

Based on the above GrabCut algorithm, we generate a rough-labeled map M^{rl} after we feed the segmentation seeds M^s, which contains Ω^{rl}_{cb}, Ω^{rl}_{pb}, Ω^{rl}_{pf} and Ω^{rl}_{cf} with the corresponding definition to Ω^{s}_{cb}, Ω^{s}_{pb}, Ω^{s}_{pf} and Ω^{s}_{cf}, respectively.

3.1.3 Object boundary refinement via adaptive initialized segmentation

To obtain more accurate salient objects, we refine the object boundaries generated by M^{rl} via adaptive initialized segmentation [39].

Figure 4 shows an overview of object boundary refinement. In order to avoid containing background in the segmented salient objects, we first erode Ω^{rl}_{cf} as follows:

$$\Omega^{rl'}_{cf} = f_e(\Omega^{rl}_{cf}, \eta_1 R(\Omega^{rl}_{cf})), \tag{6}$$

where $R(\Omega^{rl}_{cf})$ is the radius of circumcircle of Ω^{rl}_{cf}, which is adaptive to Ω^{rl}_{cf} in the rough-labeled map M^{rl}; η is a parameter, which equals 0.2 in our experiments; $f_e(\Omega, R)$ is a function to erode Ω with a radius R.

Ω^{rl}_{pf} is also updated as follows:

$$\Omega^{rl'}_{pf} = (\Omega^{rl}_{cf} \backslash \Omega^{rl'}_{cf}) \cup \Omega^{rl}_{pf}. \tag{7}$$

Meanwhile, to improve the completeness of the segmented salient objects, we dilate the non-background region, *i.e.*, the union of Ω^{rl}_{cf}, Ω^{rl}_{pf} and Ω^{rl}_{pb}, and

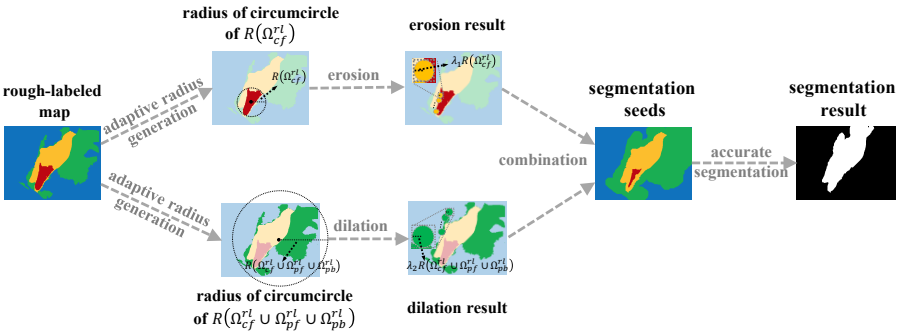

Fig. 4: Object boundary refinement via adaptive initialized segmentation.

refine probable background as the union of Ω_{pb}^{rl} and the newly covered region in dilation, which is defined as follows:

$$\Omega_{pb}^{rl'} = (f_d((\Omega_{cf}^{rl} \cup \Omega_{pf}^{rl} \cup \Omega_{pb}^{rl}), \xi R(\Omega_{cf}^{rl} \cup \Omega_{pf}^{rl} \cup \Omega_{pb}^{rl})) \backslash (\Omega_{cf}^{rl} \cup \Omega_{pf}^{rl} \cup \Omega_{pb}^{rl})) \cup \Omega_{pb}^{rl}, \quad (8)$$

where $R(\Omega_{cf}^{rl} \cup \Omega_{pf}^{rl} \cup \Omega_{pb}^{rl})$ is the radius of circumcircle of $\Omega_{cf}^{rl} \cup \Omega_{pf}^{rl} \cup \Omega_{pb}^{rl}$, which is adaptive to Ω_{cf}^{rl}, Ω_{pf}^{rl} and Ω_{pb}^{rl} in the rough-labeled map M^{rl}; ξ is a parameter, which equals 0.1 in our experiments; $f_d(\Omega, R)$ is a function to dilate Ω with a radius R.

$\Omega_{cb}^{rl'}$ is also updated as follows:

$$\Omega_{cb}^{rl'} = \Omega_{cb}^{rl} \backslash \Omega_{pb}^{rl'}. \quad (9)$$

We re-feed the segmentation seeds $M^{rl'}$ to the map generation algorithm in Section 3.1.2 to generate the accurate-labeled map M^{al}, which contains Ω_{cb}^{al}, Ω_{pb}^{al}, Ω_{pf}^{al} and Ω_{cf}^{al}, and produce the accurate binary mask by setting the values of pixels in Ω_{cf}^{al} and Ω_{pf}^{al} to 1 to denote object, and setting the values of pixels in Ω_{cb}^{al} and Ω_{pb}^{al} to 0 to denote background, respectively.

3.2 Using saliency map and RGB image

We utilize the same method to generate segmentation seeds from saliency map as defined in Section 3.1.1. When generating rough-labeled map, we extend the energy function of GrabCut by combining saliency map and color cue as follows:

$$E' = \alpha E(L, K^s, \theta^s, Z^s) + (1 - \alpha) E(L, K^c, \theta^c, Z^c), \quad (10)$$

where $E(L, K^s, \theta^s, Z^s)$ and $E(L, K^c, \theta^c, Z^c)$ are the energy functions on saliency map and color cue, respectively, which are same to the definition in Eq. (3); α is a parameter for combination, which equals 0.5 in our experiments. Based on the extended GrabCut algorithm, we generate the rough-labeled map M^{rl} using both saliency map and RGB image.

Finally, we refine the object boundaries generated by M^{rl} in the same way as defined in Section 3.1.3.

3.3 Using saliency map and RGB-D image

Similarly, we first generate segmentation seeds from saliency map in the same way as defined in Section 3.1.1. Then, we further extend GrabCut by incorporating depth cue.

Depth cue has natural advantages in salient object segmentation, due to the consistency of object appearance and dis-connectivity between foreground and background [40].

We extend the energy function of GrabCut as follows:

$$E'' = \alpha E(L, K^s, \theta^s, Z^s) + \beta E(L, K^c, \theta^c, Z^c) + (1-\alpha-\beta) E(L, K^d, \theta^d, Z^d), \quad (11)$$

where $E(L, K^s, \theta^s, Z^s)$, $E(L, K^c, \theta^c, Z^c)$ and $E(L, K^d, \theta^d, Z^d)$ are the energy functions of saliency map, color cue and depth cue, respectively, which are same to the definition in Eq. (3); α and β are parameters for combination, which equal 1/3 and 1/3 in our experiments, respectively.

Finally, we refine the distance $D(z_m, z_n)$ in Eq. (4). Referring to [35], we use Euclidean distance $D^s(z_m^s, z_n^s)$ and $D^c(z_m^c, z_n^c)$ on saliency map and color cue, and geodesic distance $D^d(z_m^d, z_n^d)$ on depth cue, respectively, because geodesic distance can better extract the spatial property of depth cue. We define the Euclidean distance $D^k(z_m^k, z_n^k), k \in \{s, c\}$ in Eq. (5), and define $D^d(z_m^d, z_n^d)$ as follows:

$$D^d(z_m^d, z_n^d) = \min\{\varphi_{m,n}\}, \quad (12)$$

where $\varphi_{m,n}$ denotes the distance of a path between pixel p_m and p_n, which is calculated as follows:

$$\varphi_{m,n} = \max_{i,j \in P_{m,n}} \{||z_i^d - z_j^d||\}, \quad (13)$$

where i and j are two neighbor pixels on path $P_{m,n}$; z_i^d and z_j^d are the depth value of i and j on depth cue. Based on the depth-aware GrabCut extension, we generate the rough-labeled map M^{rl} using both saliency map and RGB-D image.

Finally, we refine the object boundaries generated by M^{rl} in the same way as defined in Section 3.1.3.

4 Experiments

4.1 Dataset and experiment settings

We validated our method on two datasets: MSRA10K [24] and NJU2000 [26]. MSRA10K is the largest RGB image dataset for salient object detection, which contains 10,000 images and corresponding manually labelled masks in

ground truth. NJU2000 is the largest RGB-D image dataset for salient object detection, which contains 2,000 RGB-D images with manually segmented salient object in ground truth.

We utilize three common criteria for saliency cuts evaluation, namely *Precision*, *Recall* and F_β [41]. *Precision* and *Recall* are defined as follows:

$$Precision = \frac{1}{N_{img}} \sum_{i=1}^{N_{img}} \frac{|M_i \cap G_i|}{|M_i|}, \tag{14}$$

$$Recall = \frac{1}{N_{img}} \sum_{i=1}^{N_{img}} \frac{|M_i \cap G_i|}{|G_i|}, \tag{15}$$

where N_{img} is the number of images in a dataset, M_i is the binary mask of saliency cuts result of the ith image and G_i is the ground truth of the ith image.

F_β is defined with *Precision* and *Recall* as follows:

$$F_\beta = \frac{(1+\beta^2)Precision \times Recall}{\beta^2 Precision + Recall}, \tag{16}$$

where $\beta^2 = 0.3$ to emphasize precision following general practice in saliency cuts evaluation.

All the experiments were conducted on a computer with 2.9GHz Intel Core i5 CPU and 8GB memory. We applied the default settings of author suggestions for all the saliency cuts methods used in our experiments.

4.2 Component analysis

We first validated the effectiveness of three components of our method, namely adaptive triple thresholding segmentation seeds generation, salient object segmentation via input-adaptive GrabCut extension, and adaptive boundary refinement, on NJU2000. The input saliency maps are generated using RC [24].

We compare our method with three baselines. *Fixed* denotes the baseline with segmentation seeds generation using fixed thresholds which uniformly divide saliency value range (*i.e.*, (t_l, t_m, t_h) equals $(64, 128, 192)$), original GrabCut and no boundary refinement. *ASC-A* denotes the baseline using adaptive triple thresholding segmentation seeds generation to replace using fixed thresholds in Fixed baseline. *ASC-AD* denotes the baseline using salient object segmentation via input-adaptive GrabCut extension to replace using original GrabCut in ASC-A baseline. *ASC* denotes our method. Especially, when analyzing ASC using only saliency map as the input cue, ASC-A is the same as ASC-AD, because we make no extension on GrabCut algorithm in Section 3.1.

Figure 5 shows the precision, recall and F_β of three baselines and our method under different input cues. We can see that the recall and F_β grow

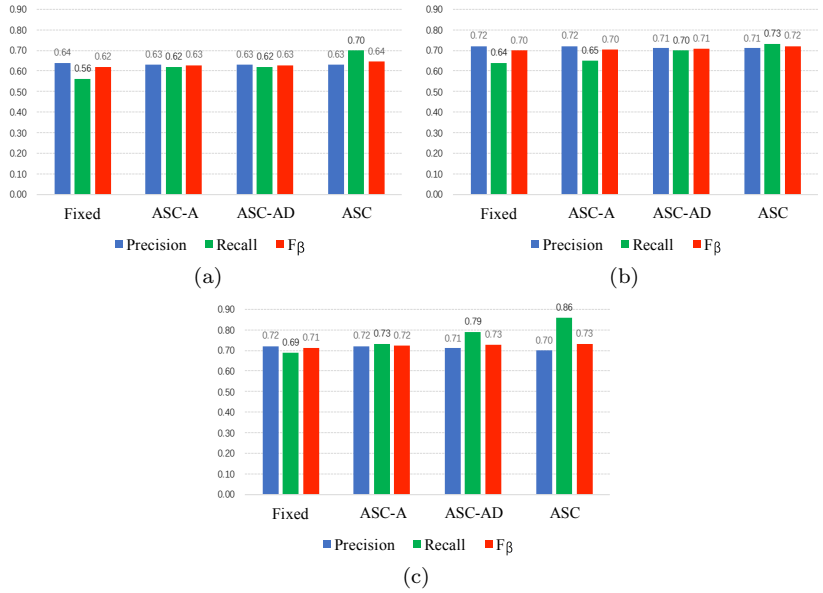

Fig. 5: Component analysis under different input cues. (a) Only using saliency map. (b) Using saliency map and RGB image. (c) Using saliency map and RGB-D image.

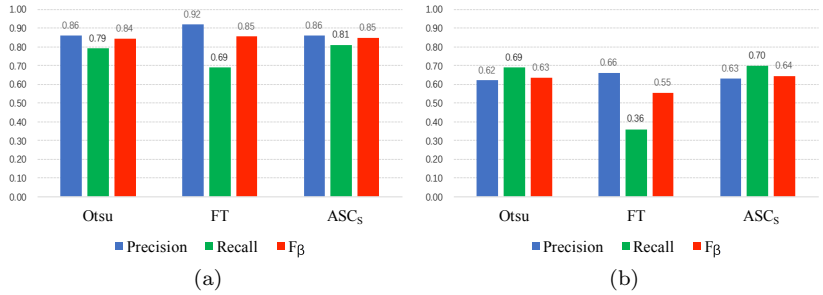

Fig. 6: Comparison of Otsu [22] , FT [23] and ASC_S using only saliency maps as the input cue. (a) MSRA10K. (b) NJU2000.

from baseline *Fixed* to *ASC* while precision keeps relatively consistent under different input cues. It indicates that each component in our method help to generate better saliency cuts results via improving the completeness of salient object segmentation while making little trade-off in accuracy.

4.3 Comparison with state-of-the-arts

We also compared ASC with the state-of-the-art saliency cuts methods using the same input value. The input saliency maps are generated using RC [24].

1) **Comparison using only saliency map as the input cue.** We compared ASC_S with two state-of-the-art saliency cuts methods using only

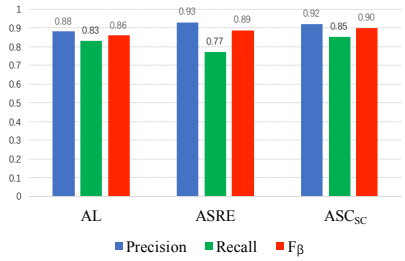

Fig. 7: Comparison of AL [20], ASRE [24] and ASC_{SC} using saliency maps and RGB images as the input cue on MSRA10K.

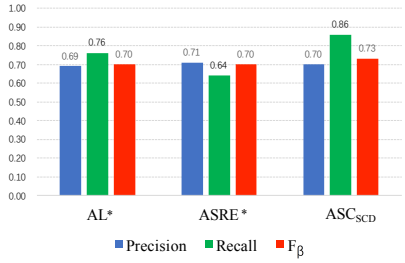

Fig. 8: Comparison of AL*, ASRE* and ASC_{SCD} using saliency maps and RGB-D images as the input cue on NJU2000.

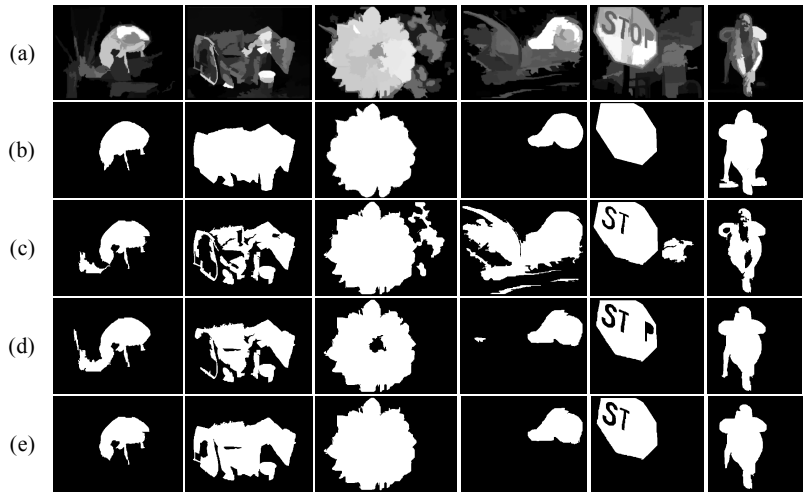

Fig. 9: Examples of saliency cuts results of different methods using only saliency map as the input cue. (a) Saliency map. (b) Ground truth. (c) Otsu. (d) FT. (e) ASC_S.

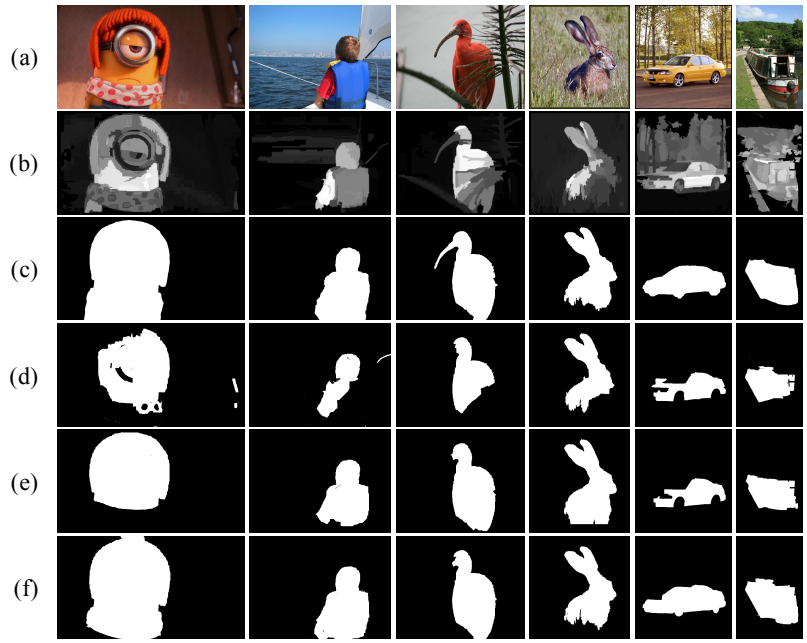

Fig. 10: Examples of saliency cuts results of different methods using saliency map and RGB image as the input cue. (a) Color cue. (b) Saliency map. (c) Ground truth. (d) AL. (e) ASRE. (f) ASC_{SC}.

saliency map as the input cue, namely Otsu [22] and FT [23], on two datasets: MSRA10K and NJU2000. Figure 6 shows the comparison results and Fig. 9 illustrates some examples of saliency cuts results of different methods using only saliency map as the input cue.

2) **Comparison using saliency map and RGB image as the input cue.** We compared ASC_{SC} with two state-of-the-art saliency cuts methods using saliency map and RGB image as the input cue, namely AL [20] and ASRE [24], on MSRA10K. Figure 7 shows the comparison results and Fig. 10 illustrates some examples of saliency cuts results of different methods using saliency map and RGB image as the input cue.

3) **Comparison using saliency map and RGB-D image as the input cue.** Because there is no specific saliency cuts method for RGB-D images, we simply extended AL and ASRE to AL* and ASRE* as the baselines by utilizing depth cue as the fourth dimension of RGB image. We compared ASC_{SCD} with the above two baselines using saliency map and RGB-D image as the input cue on NJU2000. Figure 8 shows the comparison results and Fig. 11 illustrates some examples of saliency cuts results of different methods using saliency map and RGB-D image as the input cue.

From Fig. 6 to 8, we can see that the methods achieving the highest precision usually have poor performance on recall. It means that they prefer omitting the uncertain parts in order to guarantee the accuracy

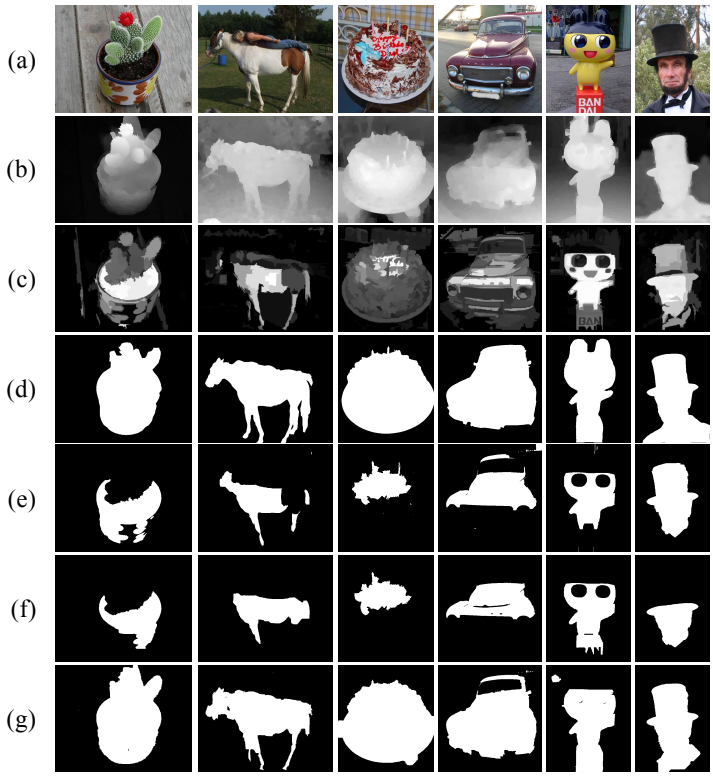

Fig. 11: Examples of saliency cuts results of different methods using saliency map and RGB-D image as the input cue. (a) Color cue. (b) Depth cue. (c) Saliency map. (d) Ground truth. (e) AL*. (f) ASRE*. (g)ASC_{SCD}.

of segmentation results. However, it leads to serious incompleteness of segmentation results. In contrast, our method has obvious promotion on recall with slight descent on precision. In this way, our method outperforms current methods on F_β value, because it obtains a better balance between precision and recall. From Fig. 9 to 11, we can see that our method produces the best segmentation results on various salient objects, such as flower, sign, animal and person.

4.4 Robustness analysis

We also compared our method with the state-of-the-art saliency cuts methods using five saliency map methods as the input cue, namely RC [24], RBD [42], MR [43], SD [44] and ACSD [26], to validate the robustness of ASC under different circumstance on NJU2000.

Table 1 shows the comparison results of different saliency cuts methods using five saliency map methods as the input cue. We can see that our method

Table 1: Comparison of ASC and six saliency cuts methods using five saliency map methods on NJU2000.

		RC [24]	RBD [42]	MR [43]	SD [44]	ACSD [26]
Otsu [22]	Precision	0.62	0.65	0.67	0.74	0.74
	Recall	0.69	0.64	**0.62**	0.32	0.74
	F_β	0.63	**0.65**	**0.66**	0.57	0.74
FT [23]	Precision	**0.66**	**0.70**	**0.71**	**0.82**	**0.80**
	Recall	0.36	0.39	0.40	0.31	0.40
	F_β	0.55	0.59	0.60	**0.59**	0.65
ASC_S	Precision	0.63	0.65	0.67	0.77	0.74
	Recall	**0.70**	**0.65**	0.61	**0.33**	**0.76**
	F_β	**0.64**	**0.65**	**0.66**	**0.59**	**0.75**
AL [20]	Precision	0.70	0.69	0.70	0.71	0.68
	Recall	**0.73**	0.70	0.65	0.39	0.66
	F_β	0.71	**0.69**	**0.69**	0.60	0.68
ASRE [24]	Precision	**0.73**	**0.70**	**0.71**	**0.78**	**0.81**
	Recall	0.61	0.63	0.59	0.43	0.69
	F_β	0.70	0.68	0.68	0.66	0.78
ASC_{SC}	Precision	0.71	0.68	0.69	**0.78**	0.77
	Recall	**0.73**	**0.71**	**0.70**	**0.46**	**0.84**
	F_β	**0.72**	**0.69**	**0.69**	**0.68**	**0.79**
AL*	Precision	0.69	**0.69**	0.69	0.68	0.66
	Recall	0.76	0.71	0.67	0.42	0.72
	F_β	0.70	0.69	0.68	0.60	0.67
ASRE*	Precision	**0.71**	**0.69**	**0.70**	0.77	**0.76**
	Recall	0.64	0.66	0.65	0.47	0.73
	F_β	0.70	0.68	0.69	0.67	**0.75**
ASC_{SCD}	Precision	0.70	0.68	0.69	**0.78**	0.71
	Recall	**0.86**	**0.77**	**0.77**	**0.51**	**0.90**
	F_β	**0.73**	**0.70**	**0.71**	**0.70**	**0.75**

outperforms other methods on F_β value with the input of all five saliency map methods, which indicates the effectiveness and robustness of our method.

5 Conclusion

In this paper, we propose an adaptive saliency cuts method which makes full use of different input cues with a unified framework, including the components of segmentation seeds generation via adaptive triple thresholding , rough-labeled map generation via adaptive GrabCut and object boundary refinement via adaptive initialized segmentation. The proposed method was validated on two largest datasets for salient object detection, namely MSRA10K and NJU2000. The experimental results show that our method is superior to the state-of-the-art saliency cuts methods under different input cues.

Acknowledgements This work is supported by National Science Foundation of China (61321491, 61202320), and Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

1. Zhang, H., Shang, X., Luan, H., Wang, M., Chua, T.S.: Learning from collective intelligence: Feature learning using social images and tags. Acm Transactions on Multimedia Computing Communications and Applications **13**(1), 1 (2016)
2. Bao, B.K., Liu, G., Xu, C., Yan, S.: Inductive robust principal component analysis. IEEE Transactions on Image Processing **21**(8), 3794–3800 (2012)
3. Zhang, Y., Hong, C., Wang, C.: An efficient real time rectangle speed limit sign recognition system pp. 34–38 (2010)
4. Guo, Y., Gu, X., Chen, Z., Chen, Q., Wang, C.: Denoising saliency map for region of interest extraction. In: International Conference on Advances in Visual Information Systems, pp. 205–215 (2007)
5. Gao, Z., Zhang, L.F., Chen, M.Y., Hauptmann, A., Zhang, H., Cai, A.N.: Enhanced and hierarchical structure algorithm for data imbalance problem in semantic extraction under massive video dataset. Multimedia Tools and Applications **68**(3), 641–657 (2014)
6. Carreira, J., Sminchisescu, C.: Cpmc: Automatic object segmentation using constrained parametric min-cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(7), 1312–1328 (2012)
7. Bao, B.K., Zhu, G., Shen, J., Yan, S.: Robust image analysis with sparse representation on quantized visual features. IEEE Transactions on Image Processing **22**(3), 860–871 (2013)
8. Guo, Y., Gu, X., Chen, Z., Chen, Q., Wang, C.: Adaptive video presentation for small display while maximize visual information. Lecture Notes in Computer Science **4781**, 322–332 (2007)
9. Yang, Y., Nie, F., Xu, D., Luo, J., Zhuang, Y., Pan, Y.: A multimedia retrieval framework based on semi-supervised ranking and relevance feedback. IEEE Transactions on Pattern Analysis & Machine Intelligence **34**(4), 723 (2012)
10. Yang, Y., Xu, D., Nie, F., Yan, S., Zhuang, Y.: Image clustering using local discriminant models and global integration. IEEE Transactions on Image Processing **19**(10), 2761–2773 (2010)
11. Zhang, H., Zha, Z.J., Yang, Y., Yan, S., Gao, Y., Chua, T.S.: Attribute-augmented semantic hierarchy:towards bridging semantic gap and intention gap in image retrieval. In: Acm International Conference on Multimedia, pp. 33–42 (2013)
12. Gao, Z., Li, S.H., Zhu, Y.J., Wang, C., Zhang, H.: Collaborative sparse representation leaning model for rgbd action recognition. Journal of Visual Communication and Image Representation **48**(C), 442–452 (2017)
13. Gao, Z., Zhang, H., Xu, G.P., Xue, Y.B., Hauptmann, A.G.: Multi-view discriminative and structured dictionary learning with group sparsity for human action recognition. Signal Processing **112**(C), 83–97 (2014)
14. Guo, J., Ren, T., Huang, L., Bei, J.: Saliency detection on sampled images for tag ranking. Multimedia Systems pp. 1–13 (2017)
15. Tang, J., Shu, X., Qi, G., Li, Z., Wang, M., Yan, S., Jain, R.: Tri-clustered tensor completion for social-aware image tag refinement. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(8), 1662–1674 (2017)
16. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis & Machine Intelligence **22**(8), 888–905 (2000)
17. Xu, N., Bansal, R., Ahuja, N.: Object segmentation using graph cuts based active contours. In: International Conference on Pattern Recognition, pp. II–46–53 vol. 2 (2007)
18. Song, H., Liu, Z., Du, H., Sun, G., Le, M.O., Ren, T.: Depth-aware salient object detection and segmentation via multiscale discriminative saliency fusion and bootstrap learning. IEEE Transactions on Image Processing **PP**(99), 1–1 (2017)
19. Ye, L., Liu, Z., Li, L., Shen, L., Bai, C., Wang, Y.: Salient object segmentation via effective integration of saliency and objectness. IEEE Transactions on Multimedia **PP**(99), 1–1 (2017)
20. 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)

21. Hou, X., Zhang, L.: Saliency detection: A spectral residual approach. In: International Conference on Pattern Recognition, pp. 1–8 (2007)
22. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Transactions on Systems Man & Cybernetics **9**(1), 62–66 (2007)
23. Achanta, R., Hemami, S., Estrada, F., Susstrunk, S.: Frequency-tuned salient region detection. In: International Conference on Pattern Recognition, pp. 1597–1604 (2009)
24. Cheng, M.M., Zhang, G.X., Mitra, N.J., Huang, X., Hu, S.M.: Global contrast based salient region detection. In: International Conference on Pattern Recognition, pp. 409–416 (2011)
25. Li, S., Ju, R., Ren, T., Wu, G.: Saliency cuts based on adaptive triple thresholding. In: IEEE International Conference on Image Processing, pp. 4609–4613 (2015)
26. Ju, R., Liu, Y., Ren, T., Ge, L., Wu, G.: Depth-aware salient object detection using anisotropic center-surround difference. Signal Processing Image Communication **38**(C), 115–126 (2015)
27. Wang, Y., Huang, L., Ren, T., Zhang, Y.: Saliency cuts on rgb-d images. In: International Conference on Internet Multimedia Computing and Service (2017)
28. Giroinieto, X., Martos, M., Mohedano, E., Ponttuset, J.: From global image annotation to interactive object segmentation. Multimedia Tools and Applications **70**(1), 475–493 (2014)
29. Kass, M., Witkin, A.P., Terzopoulos, D.: Snakes: Active contour models. International Journal of Computer Vision **1**(4), 321–331 (1988)
30. Rother, C., Kolmogorov, V., Blake, A.: "grabcut": Interactive foreground extraction using iterated graph cuts. ACM Transactions on Graphics **23**(3), 309–314 (2004)
31. Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. European Conference on Computer Vision Workshop on Statistical Learning in Computer Vision pp. 17–32 (2004)
32. Wang, C., Xue, Y., Zhang, H., Xu, G., Gao, Z.: Object segmentation of indoor scenes using perceptual organization on rgb-d images. In: International Conference on Wireless Communications and Signal Processing, pp. 1–5 (2016)
33. Rother, C., Minka, T., Blake, A., Kolmogorov, V.: Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 993–1000 (2006)
34. Ju, R., Ren, T., Wu, G.: Stereosnakes: contour based consistent object extraction for stereo images. In: IEEE International Conference on Computer Vision, pp. 1724–1732 (2015)
35. Ge, L., Ju, R., Ren, T., Wu, G.: Interactive rgb-d image segmentation using hierarchical graph cut and geodesic distance. In: Pacific-Rim Conference on Multimedia (2015)
36. Banica, D., Agape, A., Ion, A., Sminchisescu, C.: Video object segmentation by salient segment chain composition. In: International Conference on Computer Vision Workshop, pp. 283–290 (2013)
37. Huang, Y., Wang, S.: Multilevel thresholding methods for image segmentation with otsu based on qpso. In: International Conference on Image and Signal Processing, pp. 701–705 (2008)
38. Blake, A., Rother, C., Brown, M., Perez, P., Torr, P.: Interactive image segmentation using an adaptive gmmrf model. In: European Conference on Computer Vision, pp. 428–441 (2004)
39. Liu, J., Ren, T., Wang, Y., Zhong, S.H., Bei, J., Chen, S.: Object proposal on rgb-d images via elastic edge boxes. Neurocomputing **236**, 134–146 (2017)
40. Wang, Y., Huang, L., Ren, T., Zhong, S.H., Liu, Y., Wu, G.: Object proposal via depth connectivity constrained grouping. In: Pacific-Rim Conference on Multimedia (2017)
41. Borji, A., Cheng, M., Jiang, H., Li, J.: Salient object detection: A benchmark. IEEE Transactions on Image Processing **24**(12), 5706–5722 (2015)
42. Zhu, W., Liang, S., Wei, Y., Sun, J.: Saliency optimization from robust background detection. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2814–2821 (2014)
43. Yang, C., Zhang, L., Lu, H., Xiang, R., Yang, M.H.: Saliency detection via graph-based manifold ranking. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3166–3173 (2013)
44. Peng, H., Li, B., Xiong, W., Hu, W., Ji, R.: Rgbd salient object detection: A benchmark and algorithms. In: European Conference on Computer Vision, pp. 92–109 (2014)