International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.7, No.1 (2014), pp.379-392
http://dx.doi.org/10.14257/ijsip.2014.7.1.35

A New Framework for Direct Saliency Detection and Segmentation Based on Graph Methods

Lin Li[1,2], Yue Wu[1], Mao Ye[1]
[1]*University of Electronic Science and Technology of China, Chengdu 611731, China*
[2]*Sichuan TOP IT Vocational Institute*
lilin200909@gmail.com

Abstract

Saliency detection is an important research topic in computer vision. The traditional methods compute image saliency map, then salient segmentation is based on the corresponding saliency map. Unfortunately, overall performance of this method is poor due to the reason of losing some fine details and spatial information within image. This paper presents a new framework to overcome the drawback, named FDSRDS(Framework for Directly Salient Region Detection and Segmentation based on graph methods). Under FDSRSD, firstly, we get the foreground image by segmenting the original image via our extended grabcut algorithm. Mostly, the saliency region is within the foreground part. Secondly, we segment the foreground image into regions by means of graph based segmentation and nearest neighbor graph . Thirdly, we use relative weber's luminance rules to calculate every region's luminance. Finally, we get the maximum luminance region which is the saliency region. Under FDSRSD framework, algorithms we proposed capture fine details and spatial relationships in saliency computation. We demonstrate impressive results by evaluating our method with other five state-of-the-art methods on the publicly available data set.

Keywords: *Saliency Detection, Image Segmentation, Graphcut, Bounding Box*

1. Introduction

The saliency that can be an object, a person, a pixel, etc. is the state or quality which stands out relative to its neighbors. Saliency detection results has many applications in computer vision such as object recognition[1], content aware image editing[2], image segmentation[3,4] and retrieval [5], adaptive region-of-interest based image compression[6], image thumb-nailing [7,8], photo collages[9], *etc*.

There is no universal accepted concept about saliency. In [10], they define that a salient region is generally understood as a part of an image that stands out from its surrounding and thus captures the attention of a human observer. In [11], the saliency is approached in information theory framework with saliency based on self-information of each local image patch. In [12], a region is visually salient if it has unpredictable characteristics for different scales in some feature space. In [13], they define a new notion of saliency which is context-ware. The salient parts of the background dominate object context in image.

The relationship between salient region and non-salient region is just like the one between foreground and background in image. Rather *et al.*, [14] presents a good algorithm based on graph cuts which can separate the foreground and background in image.

ISSN: 2005-4254 IJSIP

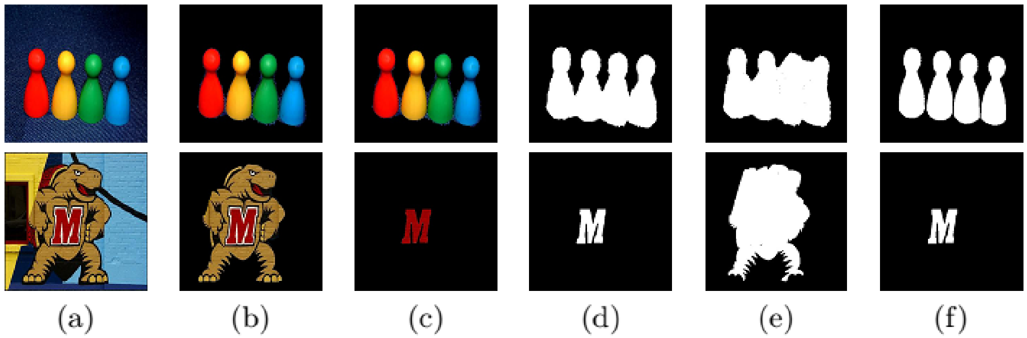


Figure 1. Examples. Panel(a) is the original image. Panel (b) shows intermediate salient segmentation results by our extended grabcut algorithm(see Algorithm 1). Panel (c) shows our final color segmentation results. Panel (d) shows our final binary segmentation. Panel (e) shows [17]'s binary segmentation results. Panel (f) shows the groundtruth segmentation results. We can seen that our FDSRDS framework captures more spatial relationships in row 1 and detail informations in row 2 while comparing with Panel (e)[17] and Panel (f) the groundtruth

The focus of this paper is the direct detection and segmentation of visual salient regions in image. More specifically, a novel salient region oriented solution framework is introduced(see Algorithm.1). Our method catches fine details in image(see Figure1). The novelties and our main contributions include: 1 A new framework named FDSRDS(Framework for Directly Salient Region Detection and Segmentation based on graph methods) is proposed for saliency detection and segmentation; 2 A salient oriented grabcut algorithm is introduced(see Algorithm.2). We all known that grabcut is an interactive algorithm. We extended the algorithm into a non-interactive environment for automatic image segmentation. On average, our extended algorithm works extremely well; 3 We also present an iterated graph segmentation[15] based on nearest neighbor graph[16]. Our extended algorithm can control the coarseness of the superpixels. At the same time, the nearest neighbor graph can catch more spatial relationships and fine details than grid graph method(see Figure4); 4 Finally, we introduce a new region-based luminance contrast method for region luminance comparison.

The rest of this paper is organized as follows: Section 2 gives a brief overview of the related works. For section 3, we introduced our FDSRDS framework. Then the experiments in section 4 is followed. Finally we give a thorough discussion and conclusion in section 5.

2. Related Work

Salient region detection has been widely studied in the community of computer vision. Most of these methods are using salient map for salient region detection.

Traditional methods consist of three steps: first, low level features are extracted; second, for each feature, a salient map is computed; then, salient maps for each feature combined and normalized. According to intrinsic principle difference, these methods can broadly classified as biological based, purely computation, or a combination[18]. The first category of these methods is based on biological vision principles. This includes the visual system presented by Itti et al.[19], which is inspired by the behavior and the neuronal architecture of the early primate visual system. The second category are purely computational methods, such as Achanta et al.[20] estimating saliency center-surround feature distances, Gao and Vasconcelos[21] maximizing the mutual information between the feature distributions of center and surround regions in an image, Cheng *et al.*, [17] evaluating global contrast and spatial coherence. The third category is partly based on biological models and partly on computational ones. For example, Harel *et al.*, [22] create feature maps using Itti's method but perform their normalization using a graph based approach.

Own to the low resolution limitation of saliency maps[18], performance by directly using saliency map for image segmentation are poor. Although Achanta *et al.*, [20] get good resolution saliency map, but still cannot catch global optimization and spatial relation details. Cheng *et al.*, [17] present saliency segmentation method based on grabcut and saliency map. However, this state-of-art method loses the spatial information and fine details.

Liu *et al.*, [23] propose an approach which is different from traditional methods. They use a supervised learning CRF framework to learn weights for linear features. On the other hand, they exploit regional and global features for salient object detection. Paria *et al.*, [10] present a another learning framework based on the superpixel, as opposed to individual image pixel. Features are chosen by color, texture, etc. However, All these methods spend huge amount of efforts to design features that are relevant to salient object detection.

Our FDSRDS framework is quite different from the approaches above mentioned. In our approach we exploit a novel idea based on global maximization. At the same time, more fine detail and spatial relationships can be preserved base on the nearest neighbor graph [16] segmentation.

3. The FDSRDS Framework

This section we describe the FDSRDS framework. Our overall framework algorithm is described as follows(see Algorithm1).

Algorithm 1: FDSRDS framework

Data: The original color image

Input: The final salient color and binary segmentation image.

1 Do extended grabcut segmentation. Output intermediate color segmentation image(only with foreground part).

2 Do graph segmentation based on intermediate color segmentation image. Group image into regions.

3 Do regions contrast luminance computation. Select maximum luminance region.

4 Output final segmentation results based on maximum luminance region and intermediate color segmentation image.

Step 1:The extended grabcut segmentation can get the foreground image which mostly contains the salient region with good fine details. However, for some images, we get some extra regions which are not belong to salient region. So we need eliminate those regions.

Step 2:We use graph segmentation[15] based on nearest neighbor graph[16] to separate the intermediate segmentation image into regions(usually 2 to 4 regions for most images).

Step 3:We compute every region's relative contrast luminance, and then select the maximum luminance region.

Step 4:According to the maximum luminance region, we can get final color and binary segmentation images.

3.1. Direct Segmentation by Extended Grabcut

Under our FDSRDS framework we have the assumption that the background in the image is more than 50% and less than 95% in image. While if the percentage of background is out of this range, we need to adjust the background and foreground. For most images, our extended grabcut algorithm for salient segmentation works well(see Figure 2).

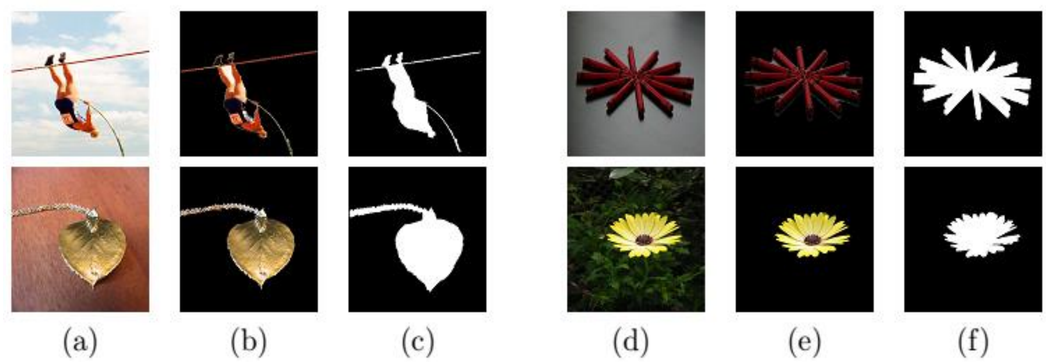


Figure 2. Good results by our extended grabcut algorithm(see Algorithm 2). Panels (a),(d) are the original images. Panels (b),(e) show intermediate salient segmentation results by our extended grabcut algorithm. Panels (c),(f) show the ground truth segmentation results. Apparently our intermediate salient segmentation results are extremely excellent for most images.

The algorithm is described as follows(see Algorithm 2).

Algorithm 2: Our extended grabcut algorithm for salient segmentation

Data: The original image

Input: The intermediate salient segmentation image

Do initialization

k:0, the loop times for maximum loop times.

bgpercent, the background pixel count percentage of the total image pixel count.

Rect: initialized with points (1,1,image width -2,image height -2), the foreground image region.

variationCount:100, after each grabcut, compute the variation pixel count of foreground pixel in image.

Do grabcut action once with initialization rectangle option.

While ((k < 50) ~and~ (bgpercent < 0.5 ~or~ bgpercent > 0.95)) ~or~ ((k < 4)~and~(variationCount < 10)) **do**

```
        Compute new bgpercent.
        if  (bgpercent < 0.5~or~bgpercent > 0.95) then
            Adjust rectangle size.
            Do grabcut action with new rectangle size.
        else
            Do grabcut action with previous rectangle size.
        end
        Compute new variationCount.
        Increase k.
    end
```

Step 1: There are three basic inputs for grabcut algorithm: The foreground, background, and the unknown part of the image that can be either foreground or background. This is normally done by selecting the a rectangle around the object of interest and mark the region inside that rectangle as unknown. Pixel outside this rectangle will then be marked as known background.

Step 2: Under the initialization step, we use rectangle whose coordinate starts at point (1,1), with width is image width subtracting 2 and height is image height subtracting 2. The algorithm creates an initial image segmentation, where the unknown pixels are placed in the foreground class and all known background pixels are classified as background.

Step 3: The foreground and background are modeled as Gaussian Mixture Models (GMMs) using the k-means clustering algorithm. A graph is built and used to find a new classification of foreground and background pixels.

Step 4: For quick convergence, there are three controlling mechanism. (1)We introduce a threshold value for *variationCount* which is background pixel variation between two sequence grabcut action. The convergence condition is that variation in every two grabcut actions is less than the threshold 10. (2)According to our massively test on the 1000 image data set, the *variationCount* variable is unstable for few exceptional images, so we set a maximum loop value 50. (3)If *bgpercent* is more than 95 percentage, we increase the rectangle area(foreground) gradually. If *bgpercent* is less than 50 percentage, we decrease the rectangle area(foreground) stepwise.

According to our experiments, our extended grabcut algorithm for salient segmentation among many images is with good results(see Figure2).

3.2. Grouping Intermediate Image into Regions by Graph Segmentation Based on Nearest Neighbor Graph

However, there still some images, the extended grabcut algorithm we proposed cannot segment the salient region with good result(see Figure3)

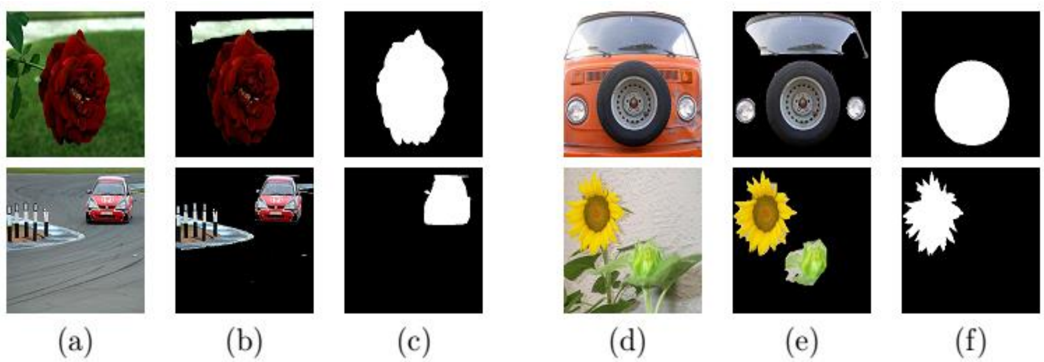


Figure 3. Bad results by our extended grabcut algorithm(see Algorithm 2}). Panels (a),(d) are the original images. Panels (b),(e) show intermediate salient segmentation results by our extended grabcut algorithm. Panels (c),(f) show the groundtruth segmentation results. There are some extra regions which don't belong to the salient region and should be eliminated.

So we need to eliminate those extra regions. First of all, we use the graph segmentation algorithm[15] to group the intermediate image into regions. Then we select salient region from these regions. There are two kinds of graph building methods for graph segmentation, one is grid graph and the other nearest neighbor graph. The nearest neighbor graph can catch the fine details and spatial neighbor information, so we use this method for graph building in our experiment(see Figure 3).

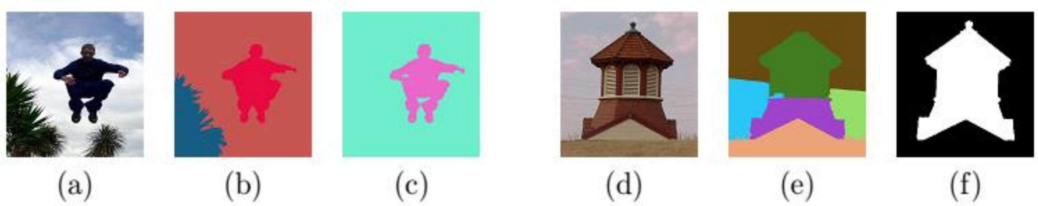


Figure 4. Comparisons of the segmentation results by graph segmentation[15] based on grid graph and nearest neighbor graph. Panels (a),(d) are the original images. Panels (b),(e) show the segmentation results by graph segmentation based on grid graph. Panels (c),(f) show the segmentation results by graph segmentation based on nearest neighbor graph. It can be seen that method based on nearest neighbor graph captures more fine details and spatial relationships.

The nearest neighbor relation defined as a set of points in a metric space has found many usages in computational geometry and clustering analysis. The nearest neighbor graph of *V*, denoted by NNG(*V*), is the directed graph <*V*,*E*> where $E = e(v) \mid v \in V$.

We can generalized the NNG(*V*) to *k*-NNG(*V*),the k-nearest-neighbor graph of *V* by introducing *k* edges from a vertex to its k nearest neighbors. In our experiment we use the algorithm described in [16]. High-dimensional nearest neighbor problems arise naturally when complex objects are represented by vectors of *d* numeric features. For our experiments we map each pixel to the feature point (*x*,*y*,*r*,*g*,*b*), where point (*x*,*y*) is the location of the pixel in the image and (*r*,*g*,*b*) is the color value (red, green, blue) of the pixel. The weight $w(v_i, v_j)$ of an edge is the distance between the two corresponding points in feature space.

Because of image variation, in order to get good experiment performance, we extend the basic graph segmentation to iterated graph segmentation. Our extended algorithm is described as follows(see Algorithm 3).

Algorithm 3: Our extended graph segmentation based on nearest neighbor graph

Data: The intermediate image

Input: Mostly 2 to 4 region markers of the intermediate image.

Do initialization

regNumbs: 10, the total region count by segmentation.

region_min_size: 800, every region minimum pixel count.

distant_size: 800, the distance for merging neighbor pixel within this constant.

While (regNumbs > 4~or~regNumbs < 2) **do**

Do graph segmentation based on intermediate segmentation results. Group image into regions.

Compute *regNumbs* (segmentation region numbers).

if (*regNumbs* > 4) **then**

Increase *region_min_size*, *distant_size*.

else

Decrease region_min_size, distant_size.

end

end

Step 1: For initialization, we set *regNumbs* to 10 for loop requirement. *region_min_size* is set to 800. This parameter is used to merge small regions into a large one. *distant_size* is set to 800. This one is used to merge neighbor pixels into one region within this constant.

Step 2: We do graph segmentation based on intermediate segmentation results and compute new *regNumbs*.

Step 3: If *regNumbs* is great than 4, we gradually increase *region_min_size*, *distant_size*.

Step 4: If *regNumbs* is less than 2, we gradually decrease *region_min_size*, *distant_size*.

Among the initialization stage, 800 for *region_min_size* and 800 for *distant_size* to most images, we can get good performance. The region count for 2 to 4 is suitable balance for capturing the fine details and getting a integrating salient region(see Figure5)

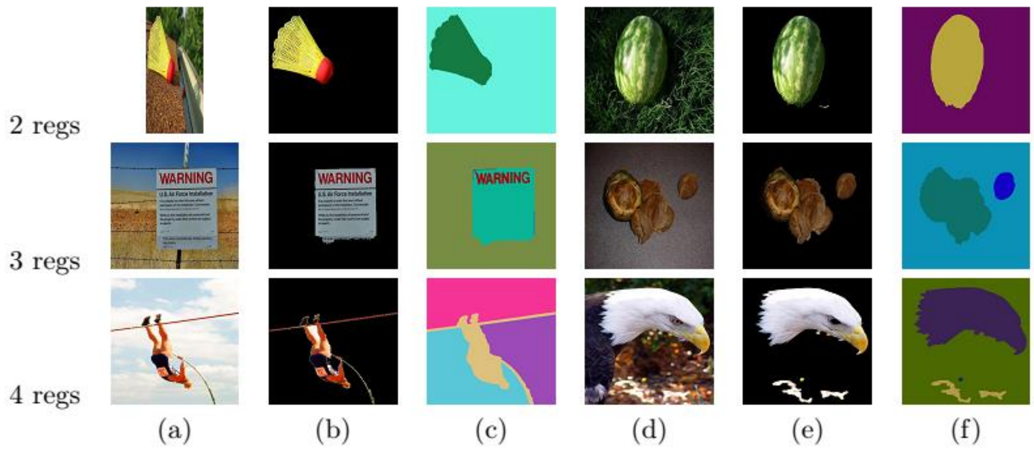


Figure 5. Grouping the intermediate image into 2 to 4 regions by our iterated graph segmentation based on nearest neighbor graph. Panels (a),(d) are the original images. Panels (b),(e) show intermediate salient segmentation results by our extended grabcut algorithm(see Algorithm 2). Panels (c),(f) show final color segmentation results by our iterated graph segmentation based on nearest neighbor graph for the intermediate image. Row 1 shows the grouping the intermediate image into 2 regions. Row 2 shows the grouping the intermediate image into 3 regions. Row 3 shows the grouping the intermediate image into 4 regions.

3.3. Eliminating Surplus Regions by Region Luminance Contrast Construction by Weber's Luminance Rules

In order to decide which part belongs to the salient region, we introduce a contrast region luminance based on weber's rules(see equation 1). We compute each region's contrast luminance, then select the maximum region.

In order to select the salient region, firstly we need compute each region's luminance. Then we select the region of maximum luminance. In our experiments we compute the luminance based on weber's luminance rules. we use web contrast due to its ability of distinguishing the background and foreground.

Weber contrast, one of the oldest luminance contrast statistics is also often used for these patterns (small, sharp-edged graphic objects like symbols and text characters on larger uniform backgrounds):

$$C_w = \frac{L_s - L_b}{L_b} \tag{1}$$

Where C_w is weber contrast coefficient, L_s is the luminance of the symbol and L_b is the luminance of the immediately adjacent background. On a display, the relationship between a pixel value and luminance is computed by first mapping the pixel to intensity, then we compute luminance from a weighted sum of the R(red), G(green) and B(blue) intensities.

3.3.1. Normal Region Luminance

The first step for our experiment, we compute each region's normal region luminance. we use the formulation:

$$L_{reg} = \left(\sum_{y,x \in C} 0.2126 * R(y,x) + 0.7152 * G(y,x) + 0.0722 * B(y,x)\right) / PixelCount \qquad (2)$$

where L_{reg} is the image region's luminance, $R(y,x)$ is the red value of the image at point (y,x), $G(y,x)$ is the green value of the image at point (y,x), $B(y,x)$ is the blue value of the image at point (y,x), C is the pixel coordinate sets in the image, *PixelCount* is pixel total count of the image region.

3.3.2. Spatial Normal Region Luminance Contrast

The second step, we need consider the interaction between regions. Here we introduce a concept of spatial normal region luminance contrast which consider the interaction between regions. We use this formulation:

$$L_i = \sum_{i \neq j}^{N} PixelCount(j) * | L_{reg(i)} - L_{reg(j)} | \qquad (3)$$

where L_i is spatial normal region luminance contrast of region i. N is the total region number. $PixelCount(j)$ is the pixel count of region j. $L_{reg(i)}$ is the region i's normal region luminance(see equation 2), and |.| denotes the absolute value.

Finally, we compare the each region's spatial normal region luminance contrast and then select the region of maximum spatial normal region luminance contrast for the salient region.

4. Experimental Comparisons

4.1. Experimental Setup

We have evaluated our FDSRSD framework on the publicly available database provided by Achanta *et al.*, [20]. We compared the proposed FDSRSD framework with 5 state-of-the-art methods: RC(Region based Contrast)[17], HC(Histogram based Contrast)[17], SR(Spectral Residual)[24], FT(Frequency Tuned)[18], LC(Luminance Contrast)[25].

We implemented our solution framework in C++ based on Opencv 2.3. In our comparison, we suppose that the fixed threshold value is 128. If the gray level value is more than 128, then it's a salient pixel. So, we convert other five state-of-the-art methods' saliency map into binary image(see Figure 6)

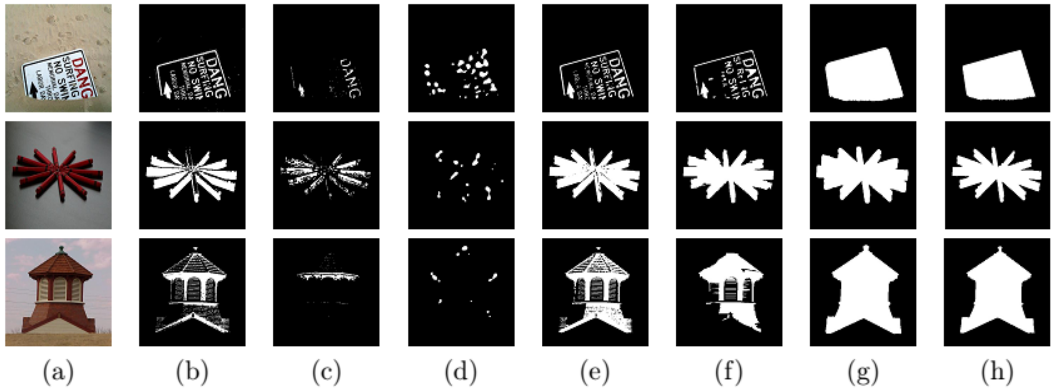


Figure 6. The converted binary images. Panel(a) is the original images. Panel (b) shows LC[25] binary images. Panel (c) FT[18] binary images. Panel (d) shows SR[24] binary images. Panel (e) shows HC[17] binary images. Panel (f) shows RC[17] binary images. Panel (g) shows our binary images. Panel (h) shows ground truth binary images.

Table 1. our method(FDSRDS) compared with RC[17],HC[17], FT[18],LC[25],SR[24]. The AUC of Methods

Method	SR	LC	FT	HC	RC	FDSRSD
AUC	0.511	0.641	0.560	0.825	0.834	0.875

According to test results, our proposed FDSRSD framework shows high precision, recall, F_β values and large AUC value.

4.2. Results

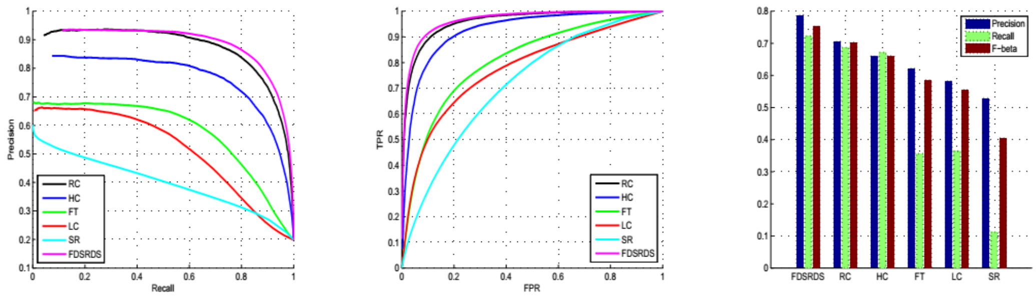
In our experiment, we compare the precision-recall curve (see Figure7 (a)), ROC (Receiver Operating Characteristic) graph(see Figure7 (b)), F-Measure(see Figure7 (c)) and AUC (Area Under Curve)(see Table 1).

Average values of precision, recall and F-Measure(see equation 4) are obtained in the previous experiment.

$$F_\beta = \frac{(1+\beta^2)Precision \times Recall}{\beta^2 \times Precision + Recall} \tag{4}$$

We use $\beta^2 = 0.3$ in our work to weight precision and recall.

According to test results, our proposed FDSRSD framework shows high precision, recall, F_β values and large AUC value.



(a) (b) (c)

Figure 7. our method(FDSRDS) compared with RC[17],HC[17], FT[18], LC[25], SR[24]. The AUC of Methods

5. Discussions and Conclusions

5.1. Results

5.1.1. Our Extended Grabcut Algorithm

Grabcut algorithm largely depend on the initialization location and rectangle size. Because outside of the rectangle is recognized as background part, the part within the rectangle is recognized as unknown test area(may be foreground or background). our proposed extended grabcut algorithm(see Algorithm 2) shows good performance on the 1000 image set.

5.1.2. Graph Segmentation Based on Nearest Neighbor Graph

Because of variation of image, the segmentation issue to find a segmentation that is neither too coarse nor too fine is an NP-hard problem[15].

According to our experiment results, segmenting image into three or four parts is fine for the overall performance, so we use iterated segmentation at each step adjusting the merging weight and minimum size of region to decrease or increase the total segmentation regions.

On the stage of graph segmentation, it's vital for parameters selection of $region\_min\_size$, $distant\_size$ and increase or decrease step size. So, there is no optimal parameter selection for each images(see Figure8). In our experiments, the step size for $region\_min\_size$ is 50 and $distant\_size$ is 50 can reach an overall performance.

In Figure 8 row 1, it reaches the optimal results with parameters of $region\_min\_size = 600$, $distant\_size = 600$. However, in Figure8 row 2 it reaches the optimal results with parameters of $region\_min\_size = 800$, $distant\_size = 800$.

According to our test, the overall performance can get with parameters of $region\_min\_size = 800$, $distant\_size = 800$.

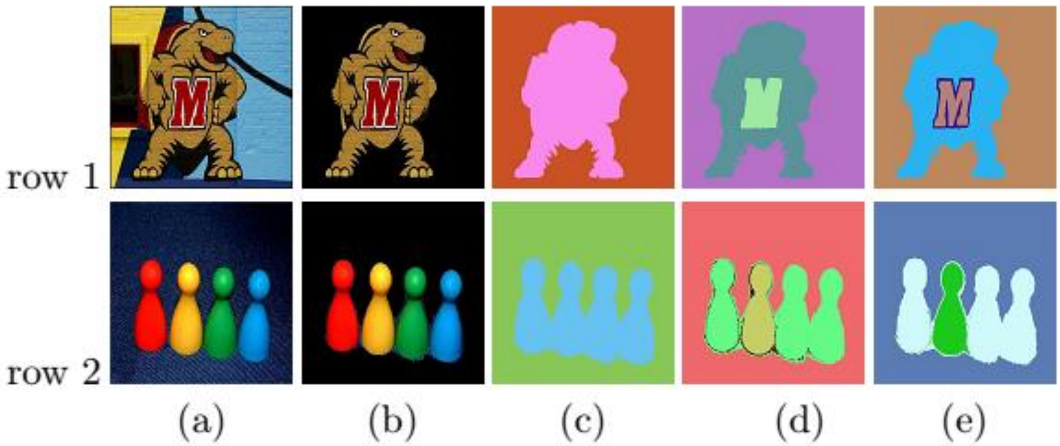


Figure 8. Grouping intermediate image into regions by different parameters of $region\_min\_size$, $distant\_size$. Panel (a) is the original images. Panel (b) shows intermediate salient segmentation results by our extended grabcut algorithm. Panel (c) shows the segmentation result by $region\_min\_size = 800$, $distant\_size = 800$. Panel (d) shows the segmentation result by $region\_min\_size = 700$, $distant\_size = 700$. Panel (e) shows the segmentation result by $region\_min\_size = 600$, $distant\_size = 600$

5.2. Conclusions and Future Works

We presented a novel framework for direct salient region detection and segmentation. Our method can catch spatial relations and global optimization of the image salient region. We evaluated our methods using the publicly available data set and got a good performance when comparing with other five state-of-the-art methods.

In future work, we want to find more accurate algorithm to define a region's luminance. We think a region's luminance contrast largely depend on its size and neighbor region's luminance. However to find it's different neighbor region is difficult in current algorithm. Another direction is to improve our framework's computation time for its complex iterating process.

Acknowledgements

This work was supported in part by 973 National Basic Research Program of China (2010CB732501) and Fundamental Research Funds for the Central University.

References

[1] U. Rutishauser, D. Walther, C. Koch, and P. Perona. Is bottom-up attention useful for object recognition. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, **(2004)** June 27-July 2; Washington DC, USA

[2] Y. S. Wang, C. L. Tai and T. Y. Lee. Optimized scale-and-stretch for image resizing. Proceedings of ACM SIGGRAPH Asia 08, **(2008)** August 11-15; California, USA

[3] B. C. Ko and J. Y. Nam Object-of-interest image segmentation based on human attention and semantic region clustering. Journal of the Optical Society of America A. 23, 2462 **(2006)**

[4] J. Han, K. N. Ngan, M. Li, and H. Zhang. Unsupervised extraction of visual attention objects in color images. IEEE Trans. Circuits Syst. Video Techn. 16, 141 **(2006)**

[5] T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S. M. Hu. Sketch2photo: internet image montage. ACM Transactions on Graphics (TOG). 28 **(2009)**
[6] C. Christopoulos, A. Skodras, and T. Ebrahimi, The jpeg2000 still image coding system: an overview. IEEE Transactions on Consumer Electronics. 46, 1103 **(2000)**
[7] L. Marchesotti, C. Cifarelli, and G. Csurka. A framework for visual saliency detection with applications to image thumbnailing. IEEE 12th International Conference on Computer Vision, **(2009)** September 29-October 2; Barcelona, Spain
[8] B. Suh, H. Ling, B. B. Bederson, and D. W. Jacobs, Automatic thumbnail cropping and its effectiveness. Proceedings of the 16th annual ACM symposium on User interface software and technology, **(2003)** November 2-5; Vancouver, Canada
[9] C. Rother, L. Bordeaux, Y. Hamadi, and A. Blake. Autocollage. Proceedings of ACM SIGGRAPH 06, (2006) July 30-August 3; Massachusetts, USA
[10] P. Mehrani and O. Veksler. Saliency segmentation based on learning and graph cut refinement. Procedings of the British Machine Vision Conference 2010, **(2010)** August 31-September 3 Aberystwyth, Ceredigion
[11] N. Bruce and J. Tsotsos, Saliency based on information maximization. Advances in neural information processing systems. 18, 155 **(2006)**
[12] T. Kadir and M. Brady, Saliency, scale and image description. International Journal of Computer Vision. 45, 83 **(2001)**
[13] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 34, 1915 **(2012)**
[14] C. Rother, V. Kolmogorov, and A. Blake, "grabcut": interactive foreground extraction using iterated graph cuts. Computer. 23, 309, **(2004)**
[15] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. International Journal of Computer Vision. 59, 167 **(2004)**
[16] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. Journal of the ACM. 45, 891 **(1998)**
[17] M. M. Cheng, G. X. Zhang, N. J. Mitra, X. Huang, and S. M. Hu. Global contrast based salient region detection. 2011 IEEE Conference on Computer Vision and Pattern Recognition, **(2011)**. June 20-25; Colorado Springs, CO, USA.
[18] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, Frequency-tuned salient region detection. 2009 IEEE Conference on Computer Vision and Pattern Recognition, **(2009)** June 20-25; Florida, USA.
[19] L. Itti, C. Koch, and E. Niebur, A model of saliency-based visual attention for rapid scene analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence. 20, 1254 **(1998)**.
[20] R. Achanta, F. Estrada, P. Wils, and S. Süsstrunk. Salient region detection and segmentation. Computer Vision Systems. 6th International Conference on Computer Vision Systems, **(2008)** May 12-15; Santorini, Greece.
[21] D. Gao and N. Vasconcelos. Bottom-up saliency is a discriminant process. IEEE 11th International Conference on Computer Vision, **(2007)** October 14-20; Rio de Janeiro, Brazil
[22] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. America. 19, 545 **(2007)**.
[23] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H. Y. Shum. Learning to detect a salient object. IEEE Transactions on Pattern Analysis and Machine Intelligence. 33, 353 **(2011)**.
[24] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. 2007 IEEE Conference on Computer Vision and Pattern Recognition, **(2007)** June 17-22; Minnesota, USA.
[25] Y. Zhai and M. Shah. Visual attention detection in video sequences using spatiotemporal cues. Proceedings of the 14th ACM International Conference on Multimedia, **(2006)** October 23-27; Santa Barbara, CA, USA

Authors

Lin Li, Ph.D. candidate at the School of Computer Science and Engineering, University of Electronic Science and Technology of China. He is an associate professor, system analyzer of computer and software technology of P.R. China and student member of China Computer Federation. His research interest covers machine learning and its application in computer image processing and vision. Corresponding author of this paper).

Yue Wu, Professor and doctor supervisor at the School of Computer Science and Engineering, University of Electronic Science and Technology of China. His interest covers computer network and data mining.

Mao Ye, is professor and doctor supervisor at the School of Computer Science and Engineering. His interest covers data mining, computer vision, intelligence information processing, etc.