AUTOMATIC OBJECT SEGMENTATION WITH SALIENT COLOR MODEL

Chieh-Chi Kao, Jui-Hsin Lai, and Shao-Yi Chien

Media IC and System Lab
Graduate Institute of Electronics Engineering and Department of Electrical Engineering
National Taiwan University
BL-421, 1, Sec. 4, Roosevelt Rd., Taipei 106, Taiwan
chiehchi.kao@gmail.com, juihsin.lai@gmail.com, sychien@cc.ee.ntu.edu.tw

ABSTRACT

Image segmentation is a well-developing topic in the image processing, and a number of previous works have been proposed and achieved high performance. However, most previous works needed user-assistance to provide the prior information of the target object in the segmentation. In this paper we propose an unsupervised scheme, combining the salient object detection and segmentation method, to segment the target object without any prior information from users. The experimental results show that the proposed salient color model derived with salient features can provide a prior information with high confidence to generate precise segmentation automatically. The proposed color model of salient objects can not only be applied with Min-Cut algorithm, but also extended to more segmentation algorithms, like matting or non-parametric model.

***Index Terms*—** Image Segmentation, Graph Cuts, Image Editing, Foreground Extraction, Salient Object Extraction

Fig. 1. Segmented results by our algorithm. The upper row is the original images, and the segmented results in the lower row are achieved without any prior information from users.

1. INTRODUCTION

The target of object segmentation in an image is to extract the semantically meaningful objects from an input image. It is an important pre-processing for a lot of applications, such as multimedia content analysis, retrieval, and video encoding.

Many object segmentation algorithms have been proposed for this purpose, such as GraphCut [1], GrabCut [2], Bayesian Matting [3], Gaussian Mixture Markov Random Field (GMMRF) [4] and Nonparametric Higher-Order Learning [5]. Most of them require user-assistant, which are called interactive segmentation algorithms. GraphCut and Nonparametric Higher-Order Learning need the user to mark certain pixels as “object” or “background;” GrabCut requires the user to drag a rectangle around the target object, dividing the whole picture into background and unknown regions; Bayesian Matting and GMMRF need the user to input a tri-map of the image. Although the above algorithms have high performance in segmentation, the results are still strongly affected by the quantity and quality of user inputs. Saliency Cuts [6] used spectral residue approach to segment salient object automatically. However, under their assumption, Saliency Cuts can not deal with salient object which is located at the right or left side of the image, like the goat in the second column of Fig. 1.

In order to reduce the burdens on users and to prevent the error resulted from the sensitivity to user inputs, in this paper, we propose an algorithm to segment the salient object automatically without any user-input information. Some segmented results without any user-assistance are shown in Fig. 1. The main concept is that in most cases, the foreground object is more attractive to users or has obvious color distribution difference from background scenes. Based on this observation, the user-assistance in the previous works can be removed with the assistance of saliency information.

For detecting a salient region, Itti et al. [7] and Liu et al. [8] put emphasis on bottom-up salient features and use these features to find a salient region. In order to detect the more detailed object boundary, we cannot only depend on these salient features. Therefore, the saliency map in our system is employed to replace the role of the user-input information in interactive segmentation algorithms. The main contributions

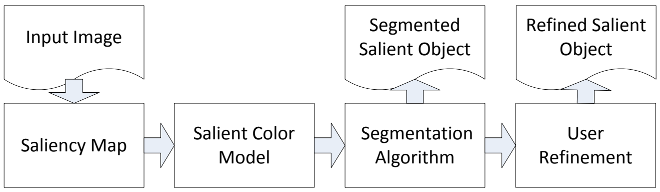

Fig. 2. The system flow of automatic salient object segmentation with salient color model construction.

of this work are shown as follows.

1. The salient features are used instead of user input for object segmentation. By this way, the burden on users can be reduced and the problem of sensitivity to user inputs can be solved.

2. A method to combine the salient features and object models is proposed to implement the object segmentation without user-assistance. This method is not only applicable to the Min-Cut algorithm but also suitable with other segmentation methods. In other words, the proposed method can remove the requirements of user-assistance in the previous object segmentation works.

3. A virtual set for foreground and background(F/B) color model is proposed, which is built from salient features. The salient color model can model the color distribution of the salient object and background more accurately.

The paper is organized as follows. First, an overview of our algorithm is described by a system flowchart in Section 2. In Section 3, we describe the details of our segmentation algorithm, which includes saliency map generation, salient color model construction, energy minimization by Min-Cut and user refinement. In Section 4, we show the segmented results and compare the performance with other competitive models. Finally, a short conclusion is given in Section 5.

2. SYSTEM FLOW

Fig. 2 shows the processing flow of the proposed automatic salient object segmentation. The first step is to calculate salient features for every pixel in the image. Since human pays more attention on humans than other objects in a picture, the saliency of each pixel contains not only bottom-up features but also top-down skin feature.

Given the saliency map, a salient object color model can be then constructed. According to the saliency information, each pixel is assigned to the F/B set. After the classification, every pixel in the F/B set gives different extent of influence to the salient object color model depending on their saliency. The pixels with higher saliency have more duplications of their RGB values in the salient object color model, and the color distribution of salient object can be modeled more precisely. After salient color model construction, the probability of a pixel belonging to the salient object or background can be calculated, which is included in energy function. The probability is calculated by K-means clustering.

Finally, the salient object segmentation is implemented by energy minimization, where Min-Cut algorithm is employed in this work. Nevertheless, there is no way to segment the salient object perfectly all the time, so we also offer a mechanism for user refinement to improve the segmentation result with minimum user interaction. The details of automatic segmentation and user refinement are described in Sec. 3.3.

3. SEGMENTATION ALGORITHM

This section shows the details of our segmentation technique and describes how the salient features are combined into a segmentation energy function which can be solved by min-cut algorithm.

3.1. Saliency Map

Liu et al. [8] propose a set of features. Among them, multi-scale contrast and color spatial-distribution feature in [8] are adopted as the bottom-up salient features. The multi-scale contrast feature is defined as follows:

$$f_c(x) = \sum_{l=1}^{L} \sum_{x' \in N(x)} ||I^l(x) - I^l(x')||^2, \qquad (1)$$

where $I^l(x)$ is the image of lth Gaussian image pyramid level in [7], L is the total number of Gaussian image pyramid level, and $N(x)$ includes the 4-neighbor pixels of pixel x. Another salient feature, color-spatial variance of every pixel, is defined as:

$$f_s(x) \propto \sum_{c} p(c|I) \cdot (1 - V(c)) \cdot (1 - T(c)), \qquad (2)$$

where c is the color of pixel x, $p(c|I)$ is the color appearance probability of color c, $V(c)$ is the spatial variance of color c, and $T(c)$ is the measurement of spatial distance from color c to the center position of the image. Therefore, $(1 - T(c))$ can be used to assign less importance to colors near image boundary, since the spatial variance of the color near the boundary may become smaller due to that the color may be cropped from the image.

Human pays more attention to people in a picture, so a top-down skin feature is also very helpful to find salient object. Therefore, we transform RGB to YCbCr space for skin detection, and define a binary skin feature $f_{sk}(x)$, which equals to 1 when the color is detected as skin; otherwise, it is set to 0. To prevent the false detection that a object with skin color is regarded as human skin, we assume that a real skin

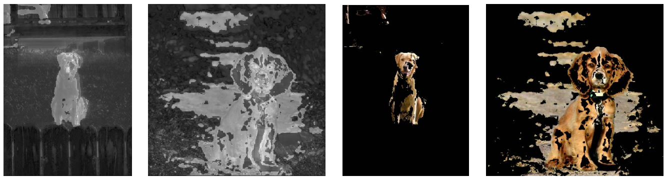

Fig. 3. The left part are two saliency maps of the fifth and sixth column picture in Fig.5. The right part are segmentation results by applying a threshold on the saliency map directly.

must has a skin color with small color-spatial variance. Therefore, an interaction term of skin feature and color-spatial variance is defined as $f_{sc}(x) = f_s(x)f_{sk}(x)$. After features are calculated, every feature is normalized to [0,1] for feature fusion. Finally, a saliency map is defined as:

$$F(x) = \sum_{k=1}^{K} \lambda_k f_k(x), \tag{3}$$

where λ_k is the weighting of the kth feature, and K is the number of features. With the saliency map, a segmentation by defining a threshold value can be achieved, as shown in Fig. 3, but the result is not satisfactory, a segmentation method with salient color model is proposed to achieve better results.

3.2. Salient Color Model

To generate the prior information of the target object, the salient features are used to detect the foreground object and background region in an image. The pixel having the higher saliency tends to be more possible as a point of the foreground object. Based on this assumption, pixels with the higher saliency should have more influence on salient color model. In order to model the color distribution of F/B, pixels in the image are divided into three sets: foreground, unknown, and background.

$$x \in \begin{cases} \chi_F, & \text{if } F(x) > \gamma_F, \\ \chi_B, & \text{if } F(x) < \gamma_B, \\ \chi_U, & \text{otherwise}, \end{cases} \tag{4}$$

where γ_F and γ_B are threshold saliency values of F/B set. In the F/B set, the element which has the higher/lower saliency should have more impact to the F/B color model. Therefore, every element in these two sets is duplicated in a virtual color model depending on their saliency: the higher/lower the saliency, the more the duplications. For the computational efficiency of our algorithm, we define the limit of total duplications in two virtual color sets are η_F and η_B. The number of duplications of each element in the salient color model is defined as follows:

$$N_F(x) = \begin{cases} \lfloor \frac{F(x)}{\sigma_F} \rfloor, & \text{if } x \in \chi_F, \\ 0, & \text{otherwise}, \end{cases} \tag{5}$$

$$N_B(x) = \begin{cases} \lfloor \frac{1-F(x)}{\sigma_B} \rfloor, & \text{if } x \in \chi_B, \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

where σ_F and σ_B are unit denominators, which are defined as:

$$\begin{aligned} \sigma_F &= \{y | \textstyle\sum_x \lfloor \frac{F(x)}{y} \rfloor = \eta_F\}, \\ \sigma_B &= \{y | \textstyle\sum_x \lfloor \frac{1-F(x)}{y} \rfloor = \eta_B\}. \end{aligned} \tag{7}$$

By finding proper unit denominators, the total number of elements in the salient color model can be controlled to improve the computational efficiency. With the information above, virtual color sets for foreground (θ_F) and background(θ_B) can be written as:

$$\begin{aligned} \theta_F &= (\{x_1, ..., x_N\}, \{(x_1, N_F(x_1)), ..., (x_N, N_F(x_N))\}), \\ \theta_B &= (\{x_1, ..., x_N\}, \{(x_1, N_B(x_1)), ..., (x_N, N_B(x_N))\}), \end{aligned} \tag{8}$$

where $\{x_1, ..., x_N\}$ are the pixels in the image. The virtual F/B color sets above are employed to derive the color distribution model for the foreground object and background ($\theta = \{\theta_F, \theta_B\}$) in the energy function in the following subsection.

3.3. Segmentation

In order to get a good salient object segmentation, an energy function is defined and a energy minimization framework is conducted. The energy function is defined as follows:

$$E(\alpha, \theta, X) = D(\alpha, \theta, X) + B(\alpha, X), \tag{9}$$

where α is the binary label of the whole image for foreground object, D is the data term which is derived from the color model (θ), and B is the boundary smoothness term.

The data term D gives high energy as penalty when the color of a pixel is close to the distribution of one color model, but the label of pixel is at the opposite side. The term D is defined as:

$$D(\alpha, \theta, X) = \sum_x H(\alpha, x, \theta), \tag{10}$$

where $H(\alpha, x, \theta)$ is the data term for pixel x:

$$H(\alpha, x, \theta) = (1-\alpha_x) \cdot Prob(x|\theta_F) + \alpha_x \cdot Prob(x|\theta_B), \tag{11}$$

where $Prob(x|\theta_F)$ and $Prob(x|\theta_B)$, calculated from F/B color models, are the probabilities of that pixel x belongs to foreground object and background scene, respectively. Here the K-means algorithm is employed to model the distribution, and a pixel has a higher foreground probability if it has a similar color distribution with the means of foreground color model. The equations of $Prob$ are written below:

$$\begin{aligned} Prob(x|\theta_F) &= 1/mindis(x, \theta_F), \\ Prob(x|\theta_B) &= 1/mindis(x, \theta_B), \end{aligned} \tag{12}$$

where $mindis(x, \theta_F)$ and $mindis(x, \theta_F)$ are the minimum distances to the means of every cluster in the color model:

$$\begin{aligned} mindis(x, \theta_F) &= min\{|x - t_{F_1}|_2, ..., |x - t_{F_C}|_2\}, \\ mindis(x, \theta_B) &= min\{|x - t_{B_1}|_2, ..., |x - t_{B_C}|_2\}, \end{aligned} \quad (13)$$

where t_F and t_B are the centers of clusters in the F/B color models, $|.|_2$ is the L2 distance of two points in the RGB color space, and C is the number of clusters.

To enhance labeling continuity, the boundary term $B(\alpha, X)$ is employed, which can ensure the label smoothness of neighboring pixels, and make the pixels with similar colors labeled in the same directory. B can be defined as follows:

$$B(\alpha, X) = \sum_x \sum_{x' \in N(x)} \zeta(\alpha_x, \alpha_{x'}) exp(-\beta |x - x'|^2), \quad (14)$$

where the function $\zeta(\alpha_x, \alpha_{x'}) = 1$ if $\alpha_x \neq \alpha_{x'}$, otherwise $\zeta(\alpha_x, \alpha_{x'}) = 0$. The coefficient β controls the impact of boundary term in the energy function.

After the energy function is fully defined, we use min-cut algorithm [9] to find the minimum energy. The segmentation labeling result $\hat{\alpha}$ can be written as:

$$\hat{\alpha} = \arg\min_{\alpha} E(\alpha, \theta, X). \quad (15)$$

Consequently, the segmentation is automatically done from saliency map without any user input, which means the proposed salient object segmentation algorithm is an unsupervised approach.

Sometimes the segmented results may not meet users' demand, and a refinement procedure is also proposed to let users add or remove the segmented objects just using one-click on the image. The click on the picture tells the information about what point the user wants to add or remove. The chosen point is regarded as a mean of one cluster in the salient color model, and the energy minimization is applied again with the new color model. Finally, we can get a refined segmentation result, and it will be further discussed in Sec. 4.3. The proposed one-click refinement procedure provides users an effortless way to do the refinement. Compared with GraphCut and GrabCut, which need users to input scribbles to refine the segmented result, the proposed algorithm greatly reduce the burden on users.

4. EXPERIMENT RESULTS AND DISCUSSION

In this section, we will discuss about the setting of parameters and show the segmentation results, where Microsoft GrabCut database is used as the test bed. Note that the segmentation results are automatically produced by our algorithm without any user-assistance.

4.1. Parameter Settings

There are two main groups of parameters, one is for saliency map and the other is for segmentation. Each parameter for saliency map gives different contribution, for example, larger λ_c puts more emphasis on the edges, but tends to be failed when the background scene is complex; larger λ_s emphasizes the part which has unique color, but tends to be failed when the color of target object is similar to the color of some background parts. Therefore, it is important to find a set of proper parameters. Based on our observations, the parameters for saliency map are chosen empirically, where $\lambda_c = 0.33$, $\lambda_s = 0.33$, $\lambda_{sk} = 0.1$ and $\lambda_{sc} = 0.24$.

The parameters for segmentation have huge influences on the segmentation quality and computation complexity. Although larger η_F and η_B can make the salient color model more precise, the computation complexity increases greatly. The values of γ_F and γ_B also affect the quality of the segmentation, where higher γ_F makes the color model concentrate on the most salient part of object but may overlook the other parts of the target object. Since there is a trade-off between segmentation quality and computation complexity, we tried to find an acceptable solution, and the parameters are set as follows: γ_F = the top 10% highest saliency, γ_B = the median of all saliency, $\eta_F = 20000$ and $\eta_B = 60000$. The experimental results below are generated by these parameters.

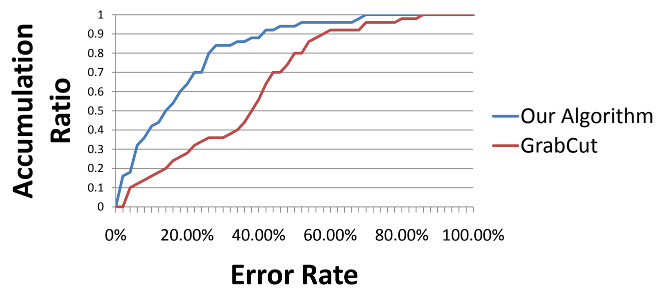

Fig. 4. The error rate of automatic segmentation algorithms, which are tested on GrabCut database. The curves show the accumulation of segmentations whose error rate (horizontal coordinates) under certain value.

4.2. Error Rate of Segmented Results

We evaluate the performance of our algorithm on 50 pictures in Microsoft GrabCut database. Many state-of-art methods are also tested on this database, but most of them use trimap as user input to implement segmentation, and the error rate is calculated by the portion of wrong labeling in the unknown/matting region. It is not reasonable to compare the error rate of user-assistant algorithms with ours, which does not have any prior information about the input image. Therefore, for fair comparison, we simulate GrabCut to be an automatic segmentation algorithm by providing a rectangle just close to the boundary of every image as the user-input information. By this way, the segmentations by GrabCut can be generated automatically and compared with our results, as shown in

Fig. 4. In Fig. 4, it illustrates that 42% of segmentations by our algorithm have the error rate lower than 10%, 66% have the error rate lower than 20%, and there are 8 segmentations (16%) of our algorithm having error rate under 2%, which means the proposed algorithm has high performances in certain cases. It also shows that the performance of the proposed automatic segmentation algorithm is better than that of GrabCut when no user-input information is available.

4.3. Subjective Segmentation Result Comparison

Fig. 5 shows the segmentation results which are compared with some state-of-the-art algorithms. Our algorithm provides a solution for automatic salient object segmentation with high quality. The first column in Fig.5 illustrates that even the background scene is complex, the proposed algorithm can still perform well since the color distribution of salient object is quite different from background. Furthermore, the fourth column in Fig.5 shows that although some parts of color of the dog's fur is very similar with the color of sand, since the other parts of the dog are definitely belong to salient object, it still can be segmented correctly through the effect of boundary term in the energy function.

Based on the results in Fig. 5, we can conclude that if a picture has an obvious salient object which has a color distribution different from the background scene, our salient object color model can model the color distribution of the object precisely. On the contrary, for example, the first row in Fig. 6 shows that when the color of the duck's neck is similar with the color of water, the segmented result is not desirable. With the one-click refinement procedure, the result can be improved as the duck with red boundary. If a picture has an ambiguous salient object, the constructed saliency map will not provide helpful information. Therefore, the proposed algorithm can handle the segmentation problem whose target object is a clear and definite one, but if the composition of the picture is complicated, user refinement may be needed to further enhance the quality.

5. CONCLUSION

In this work, we propose an automatic salient object segmentation method with salient color models generated with saliency map. The proposed concept of salient object color model can not only be used with Min-Cut algorithm, but also other segmentation algorithms. By this way, the problem of user-input sensitivity can be solved and the interactive time of users can be saved. This automatic segmentation concept can be employed in mobile applications, where users definitely do not like too much operations on the mobile devices with small touch panel. With this technique, mobile devices can find the salient objects in the frame automatically, which is beneficial for other applications like object query or object recognition through the internet.

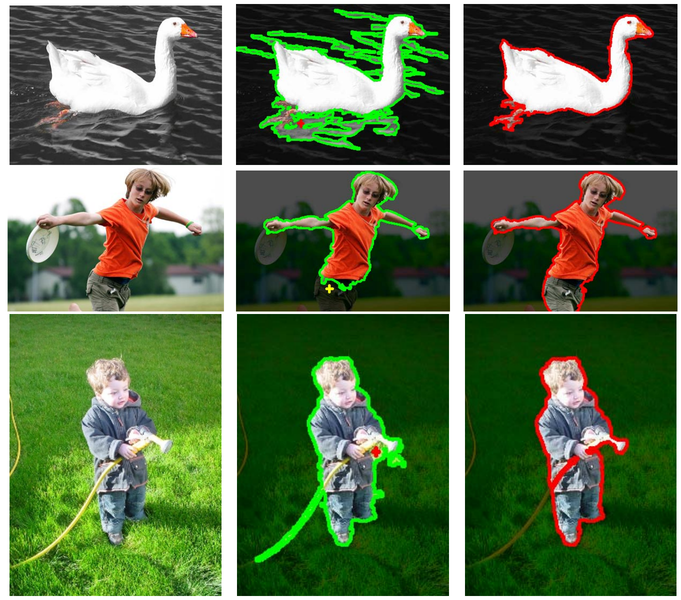

Fig. 6. User refinement by one click. The first column are original images. The second column are automatic segmented results. The center of red and yellow crosses in the second column are the one click points for user refinement; the pixel color at the center of red cross is added to background color model as one cluster center, and the yellow one is added to foreground color model. The last column are refined results.

The possible extension of this work is to combine more prior information into our framework to improve automatic segmentation results. There are still some other cues for detecting salient object on mobile devices, such as using disparity map to find the object which is closest to the user or a training-based database for a specific user, which records the color/texture information of the preferred salient objects of the user.

6. REFERENCES

[1] Y.Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images," in *Proceedings of IEEE International Conference on Computer Vision (ICCV 2001)*, 2001, vol. 1, pp. 105 –112 vol.1.

[2] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut: interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, pp. 309–314, August 2004.

[3] Y.-Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski, "A bayesian approach to digital matting," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2001)*. 2001, vol. 2, pp. 264–271, IEEE Computer Society.

Fig. 5. Segmentation result comparison. The first row is the original image and user inputs for other segmentation techniques. The blue rectangle is used for GrabCut; red and green lines are used for Nonparametric Higher-Order Learning model. The second row is the segmented results of GrabCut. The third row is the segmented results of Nonparametric Higher-Order Learning model. The last row is the automatic segmented results of our algorithm.

[4] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr, "Interactive image segmentation using an adaptive gmmrf model," in *Proceedings of European Conference on Computer Vision (ECCV 2004)*, 2004, pp. 428–441.

[5] T. H. Kim, K. M. Lee, and S. U. Lee, "Nonparametric higher-order learning for interactive segmentation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, 2010, pp. 3201 –3208.

[6] Yu Fu, Jian Cheng, Zhenglong Li, and Hanqing Lu, "Saliency cuts: An automatic approach to object segmentation," in *Proceedings of International Conference on Pattern Recognition (ICPR 2008)*, 2008, pp. 1 –4.

[7] 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*, vol. 20, no. 11, pp. 1254 –1259, 1998.

[8] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum, "Learning to detect a salient object," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 353 –367, 2011.

[9] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124 –1137, 2004.