Salient Object Cutout Using Google Images

Hongyuan Zhu, Jianfei Cai, Jianmin Zheng, Jianxin Wu, Nadia Thalmann
School of Computer Engineering
Nanyang Technological University, Singapore
Email: hzhu1@e.ntu.edu.sg, {asjfcai,asjmzheng,jxwu,nadiathalmann}@ntu.edu.sg

Abstract— **Given any image input by users, how to automatically cutout the object-of-interest is a challenging problem due to lack of information of the object-of-interest and the background. Saliency detection techniques are able to provide some rough information about object-of-interest since they highlight high-contrast or high attention regions or pixels. However, the generated saliency map is often noisy and directly applying it for segmentation often leads to erroneous results. Motivated by the recent progress on image co-segmentation and internet image retrieval techniques, in this paper, we propose to use the user input image for segmentation as a query image to Google Images and then employ the top returned Google images to build up the knowledge about the object-of-interest in the user input image. Particularly, we develop a lightweight algorithm to learn the knowledge of the object-of-interest in the retrieved images to enhance the saliency map of the input image. Then, the enhanced saliency map is used to initialize the graph-cut to extract the object-of-interest. Experiments with the Mcgill dataset and multiple challenge cases demonstrate the effectiveness of our method in terms of producing a clean cutout.**

I. Introduction

Object cutout, which attempts to segment an image into the object-of-interest and its corresponding background, is of great practical importance in image editing tasks, such as image montage orcolorization et al. This problem is challenging due to lack of information for the object-of-interest and the background. Fully supervised methods [1] learn the information from pre-labeled databases, which can only be used to detect the objects existing in the databases. Semi-supervised methods [2] which rely on a small amount of user input to obtain some prior information have achieved great success. Nevertheless, when the number of images is large, user input is time consuming and infeasible.

To reduce the workload of the user and develop fully automatic object cutout systems, saliency detection techniques [3], [4], [5], [6] which measure the pixels' distinctiveness have been used to replace the user input [7], [8] and have achieved certain degree of success. Cheng et al's work [6] is among the latest, which can detect regions highly contrast with its neighbors and achieves state-of-the-art detection results. Using the saliency detection followed by one of the existing segmentation algorithms such as [2] might be viable to segment a single image if the object-of-interest is prominent and the background is not too cluttered. This assumption, however, is not often met in real-world images. The segmentations on real-world images based on the saliency detection often lead to considerable noise and errors.

Recently, object co-segmentation of multiple images is introduced to avoid the dilemma of single image segmentation. The image co-segmentation problem was first introduced in [9], which deals with automatically segmenting a similar foreground object from two images with unrelated backgrounds. With the additional image, the knowledge about the object-of-interest is greatly enhanced, while the effect of background clutter is alleviated because the backgrounds in the two images are likely to be different. The co-segmentation problem was later being extended to scale invariance and multiple images and improved with multiple cues or a co-saliency prior [10]. Despite the novel idea, co-segmentation heavily relies on the assumption that the set of images have small variation in the object-of-interest but large variation for the background. For the application of segmenting one user input image, it is not easy to find such a set. In addition, to co-segment more than two images, current methods [9], [10] could take several minutes or hours.

Motivated by the idea of image co-segmentation and recent progress in internet image retrieval techniques, in this paper, we propose to automatically segment a user input image with the help of Google Images. Since 2011, Google begins to provide "Search Visually Similar Image" function, which can return many images that contain prominent objects which look similar to the object-of-interest in input image. Thus, we propose to use the user input image for segmentation as a query image to Google and then employ the top returned images to build up the knowledge about the object-of-interest in the user input image. To the best of our knowledge, such an idea has not been proposed in literature before. We would like to point out that our internet assisted image segmentation is different from the co-segmentation problem [9], [10] or the co-saliency detection problem [11], [10]. First, our target is to not to segment a set of images, but to cutout the object-of-interest from a single user input image, with the aid from a set of Google retrieved images. Thus, it is possible to develop lightweight and fast segmentation algorithms. Second, as the Google returned images might contain objects which have large variations to the object-of-interest in the user input image, even for the top ranked retrieved images, thus we can only deem that the object-of-interest of the input image and the top retrieved images share some appearance similarity, especially in local structure and color. Noted, some works have used extrinsic images in saliency detection tasks [12], [13], however they either require human annotated samples or the extrinsic images should contain objects that can be aligned with the object of interest in the user input image, which restrict the application of the methods.

Particularly, we propose to enhance the saliency map of the user input image by highlighting the regions that frequently appear in the salient areas of the top Google retrieved images and the regions that match the global color prior model trained from the common salient areas of the Google images. Extensive experiments on multiple challenging cases show that, with the help of Google Images, the proposed method is able to boost the object-of-interest's saliency and suppress the saliency of the background. Initialized with the enhanced saliency map, the existing iterative graph cut algorithm [2] is able to generate a much cleaner and accurate salient object.

II. Our Method

The proposed system consists of two major steps: 'saliency detection & salient region formation', and 'saliency map enhancement'.

A. Saliency Detection & Salient Region Formation

In this first step, we apply Cheng et al's saliency detection method [6] to each image (including top Google returned images and the user input image) to assign each pixel a 'Saliency' value. We choose Cheng et al's method because it offers state-of-the-art detection result. The Saliency values essentially highlight those salient pixels that are of high local contrast. However, salient pixels are often noisy and distributed everywhere, which does not provide sufficient regional information to lead to the object-of-interest.

Thus, we use the iterative graph cut scheme [2] to further process the initial saliency map to form the salient areas in each Google retrieved images $I_k, k = 1, 2, .., L$ (we only consider the top 10 returned images, i.e. $L = 10$). In particular, for each image, we first train the Gaussian Mixture Models (GMMs) for the graph cut. We treat the pixels with saliency values larger than the mean value as the foreground seeds and the pixels within a 10-pixel distance to the image boundary as the background seeds, which is according to the conventional photography composition rule that people usually do not place the object-of-interest at the image boundary. Then, we apply the graph cut algorithm that labels some pixels as foreground and some as background. After each iteration, the foreground GMM and background GMM are re-estimated with the currently labeled pixels. The iterative process repeats until convergence. With such a process, we can obtain rough and compact salient areas that will be used in the subsequent steps.

B. Saliency Map Enhancement

The purpose of the saliency map enhancement step is to use top returned Google images to enhance the saliency map of the input image. We consider that the 'object-of-interest' should have the following three characters: 1) *Distinct* - the object should be of interest and such a property is captured by the 'Saliency' value; 2) *Frequently appear in the salient areas of the top returned Google images*, and this can be described by the 'Object Frequency' property; 3) *Have some global similarity to most of the top retrieved images' salient parts*, which can be estimated by the 'Foreground Likelihood' property. Thus, we define our 'Internet aided Saliency Map' (ISM) as:

$$ISM = Saliency \times ObjectFrequency \times ForegroundLikelihood \tag{1}$$

where Saliency is directly obtained through the saliency detection in the first step and the calculation of the Object Frequency and Foreground Likelihood values are explained in the following subsections.

1) Object Frequency Estimation: An 'Object Frequency' value is computed for each pixel in the input image, which measures how frequently a pixel appears in salient parts of the Google retrieved images. This can be done by checking whether a similar pixel can be found in the salient area of each retrieved image. However, direct matching pixels in two images are computationally expensive and can lead to a lot of false detections. In our work, we measure the Object Frequency value of a pixel through measuring the Object Frequency of the region / superpixel a pixel belongs to because superpixel can reduce matching cost and increase matching accuracy [14].

To segment the input image and the salient areas of the retrieved images into superpixels, we choose the method proposed by Felzenszwalb and Huttenlocher[15] (with parameters $sigma = 0.5, K = 80, min = 50$). Other over-segmentation methods can also be used. The appearance of each superpixel is described by a codeword histogram which is generated by clustering the features consisting of dense HOG descriptors and local colors of all the pixels. In particular, we choose the HOG descriptor as it can capture the local gradient field around each pixel which reflects local patch structure information [16]. The cell, block and window sizes of the descriptor are all 3x3. To further enhance the feature description capability, we include the local color information. The color space we consider includes RGB, HSV, $L^*a^*b^*$. In this way, each pixel will be associated with a 18 dimension feature which consists of 9 HOG descriptors and 9 color features. Then, K-means is used to quantize the features into $N_c (N_c = 30)$ codewords. The codeword distribution in each superpixel can be described by a codeword histogram, which is normalized to the scale [0,1].

For every super-pixel i in the user input image I^0, we measure the appearance difference against each region j in the salient area of each searched image $I^k, k = 1, 2, ..., L$ by computing the chi-square distance of their codeword histograms h_i^0 and h_j^k:

$$d(h_i^0, h_j^k) = \chi^2(h_i^0, h_j^k) = \sum_{z=1}^{Z_f} \frac{(h_i^0(z) - h_j^k(z))^2}{h_i^0(z) + h_j^k(z)}$$

where Z_f is the number of codeword bins. The best match region j_k in retrieved image I^k is the one with minimum d value over all the valid regions in I^k. Then, we define the Object Frequency of super-pixel i in the input image I^0 as

$$ObjectFrequency(i) = \frac{1}{L} \sum_{k=1}^{L} \exp(-\frac{d(h_i^0, h_{j_k}^k)}{\sigma}) \tag{2}$$

where the distance d is transformed to similarity using a Gaussian kernel and σ is a scaling factor. In all the experiments,

σ is empirically fixed to 0.02. Finally, the Object Frequency value of a pixel is set to be the same as that for its belonging region.

2) Foreground Likelihood: The 'Foreground Likelihood' takes into account some global prior models to measure how probably a pixel belongs to the common salient areas in searched images. Considering that color Gaussian Mixture Models (GMMs) have been very successful in interactive image segmentation [2], we adopt color GMMs as our global prior models, which is built by following the method in [2]. First, to find the common parts in the salient areas of the retrieved images, we use a simple histogram filtering technique. In particular, for the salient areas extracted in the preprocessing step, we merge their histograms into a global histogram. Then we sort the codeword bins according to their numbers of samples, and threshold the global histogram at the bin where 80 percent of the samples exist. The remaining 20 percent pixels are discard, as they represent the pixels and regions which are much less likely to be common in the set of the images. Second, we use the filtered samples to train a foreground GMM. Meanwhile, a background GMM is also trained by using the pixel samples in the input image that are within a 10-pixel distance to the picture boundary, according to the conventional photography composition rule.

Let $\Pr(x|F)$ and $\Pr(x|F)$ denote the obtained foreground and background GMMs, respectively. For each pixel x, we compute a normalized probability $g(x)$ to indicates its probability of belonging to the common salient area, where $g(x)$ is defined as

$$g(x) = \frac{-\log\Pr(x|B)}{-\log\Pr(x|F) - \log\Pr(x|B)}.$$

To avoid instability, for a region i, we average the probabilities of all its pixels and use that as the "Foreground Likelihood" value for all its pixels, i.e.

$$ForegroundLikelihood(i) = \frac{1}{N(i)}\sum_{x\in i} g(x) \qquad (3)$$

where $N(i)$ is the number of pixels in region i.

After saliency map is enhanced with the foreground likelihood and object frequency prior, the iterative graph cut [2] is applied to cut out the object-of-interest from the input image.

III. Experiment Results

To demonstrate the boosting performance of using Google images in Cheng et al's region contrast method (RC) [6], we test our method over the Mcgill dataset [17] which contains hundreds of images which have sizable salient objects with cluttered backgrounds. We adopt the precision and recall metric [18] to compare the segmentation result with the binary human groundtruth, whereas a method which has a high precision and recall score is preferred. The F-measure which can reflect the average segmentation quality over the dataset is also listed in Fig. 1, a higher F-measure indicates better across dataset performance. Moreover, the score of our method which removes 'Foreground likelihood' component is listed for reference. Finally, we also compare some state-of-the-arts detection methods, SR[4], FT[3] and HC[6] and then measures their performance gain after using Google images. Fig. 1 demonstrates that using Google images can help various methods maintain a comparable recall score after boosting and meanwhile produce much higher precision score and better average across dataset segmentation performance, our method which uses Region Contrast (RC) with Google images achieves the highest precision and F-measure and a relative high recall value.

The visual result of some challenge cases are demonstrated in Fig. 2. In particular, the saliency maps of some state-of-the-art algorithms [4], [3], [5], [11], [6] are listed from column (c) to column (g) in Fig. 2. Most of these methods produce low-resolution saliency maps and can only highlight some salient contours instead of regions. Among these methods, Cheng et al's method [6] denoted as 'Region Contrast' produces relatively the best saliency map. However, directly applying the iterative graph cut to Cheng et al's saliency maps cannot produce clean cutouts, as shown in column (k), where there exist a lot of noise and errors. This is mainly because the quality of the saliency maps produced by Region Contrast is still limited.

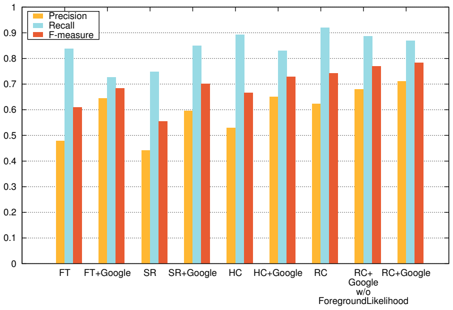

Fig. 1. Precision-recall bars for graph cut algorithm using different saliency maps as initialization, Our method (RC+Google) shows high precision, recall, and F-measure values over the Mcgill dataset

On the contrary, our method makes use of the images returned from Google (column (b) of Fig. 2) to enhance the saliency map of each input image. Particularly, we use the Region Contrast method to generate the initial map. We first show the effectiveness of the Object Frequency component by multiplying the initial map with the Object Frequency value computed in (2) and normalizing it to $[0, 1]$. The enhanced saliency map is shown in column (h) of Fig. 2, where we can see that the Object Frequency component is able to filter out less frequently appearing regions and highlight those regions that appear frequently. Second, we then evaluate the effectiveness of the Foreground Likelihood component by multiplying the initial map with the Foreground Likelihood value computed in (3) and normalizing it to $[0, 1]$. The corresponding enhanced saliency map is shown in column i of Fig. 2, where we can observe that those regions which have similar color to the common salient areas of the retrieved images are enhanced. Combining the Object Frequency and the Foreground Likelihood components, we achieve a nice balance of highlighting the foreground and filtering out the background in the saliency maps, as shown in column (j) of Fig. 2. Finally, by applying the iterative graph cut to the enhanced saliency maps in column (j), we are able to produce much cleaner

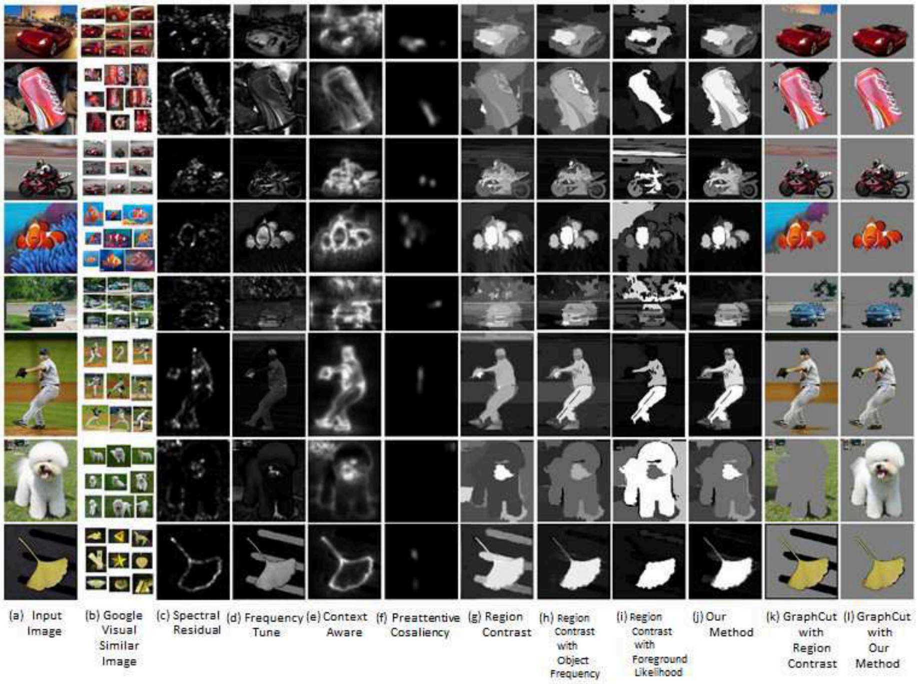
(a) Input Image
(b) Google Visual Similar Image
(c) Spectral Residual
(d) Frequency Tune
(e) Context Aware
(f) Preattentive Cosaliency
(g) Region Contrast
(h) Region Contrast with Object Frequency
(i) Region Contrast with Foreground Likelihood
(j) Our Method
(k) GraphCut with Region Contrast
(l) GraphCut with Our Method

Fig. 2. The summary of the experimental results. (a): User input images;(b): Top retrieved Google images; (c)-(g):Saliency detection results of various state-of-the-art algorithms (h) combining Region Contrast with Object Frequency; (i) Combining Region Contrast with Foreground Likelihood; (j) The enhanced saliency map produced by our proposed method; (k) The graph cut results initialized by (g); (l) The graph cut results initialized by (j).

cutouts in column (l). The average running time on a common PC to process a 400×300 image is around $10 \sim 15$ seconds (including image downloading time).

IV. Conclusions and Discussions

The major contributions of this paper are twofold: 1) the idea of using Google Images for automatic single-image salient object cutout; 2) a lightweight algorithm to learn the knowledge of the object-of-interest in the retrieved images to enhance the salient map of the input image. Experimental results show that with the help of Google Images, the proposed method is able to quickly boost the object-of-interest's saliency and suppress the saliency of the background, which eventually improve the iterative graph cut results significantly.

Although our method can help highlight the object-of-interest from ambiguous saliency map in many challenging cases, the performance of our method is bottle-necked by the retrieval quality of Google Images. Especially when most of the top returned images contain no object similar to the one in the input image or most of the top returned images are identical to the input image, our method fails in the sense that it will not outperform the corresponding single-image segmentation without the help of Google Images. We believe the performance of our proposed method will be improved with the development of image retrieval technology.

References

[1] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *CVPR*. IEEE, 2008.

[2] C. Rother, V. Kolmogorov, and A. Blake, ""grabcut": interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.

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

[4] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *CVPR*. IEEE, 2007.

[5] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," in *CVPR*. IEEE, 2010, pp. 2376–2383.

[6] M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu, "Global contrast based salient region detection," in *CVPR*, 2011, pp. 409–416.

[7] Y. Fu, J. Cheng, Z. Li, and H. Lu, "Saliency cuts: An automatic approach to object segmentation," in *ICPR*. IEEE, 2008, pp. 1–4.

[8] C. Jung, B. Kim, and C. Kim, "Automatic segmentation of salient objects using iterative reversible graph cut," in *ICME*. IEEE, 2010, pp. 590–595.

[9] C. Rother, T. P. Minka, A. Blake, and V. Kolmogorov, "Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs," in *CVPR*. IEEE, 2006, pp. 993–1000.

[10] K.-Y. Chang, T.-L. Liu, and S.-H. Lai, "From co-saliency to co-segmentation: An efficient and fully unsupervised energy minimization model," in *CVPR*, 2011, pp. 2129–2136.

[11] H.-T. Chen, "Preattentive co-saliency detection," in *ICIP*. IEEE, 2010, pp. 1117–1120.

[12] L. Marchesotti, C. Cifarelli, and G. Csurka, "A framework for visual saliency detection with applications to image thumbnailing," in *ICCV*. IEEE, 2009, pp. 2232–2239.

[13] M. Wang, J. Konrad, P. Ishwar, K. Jing, and H. A. Rowley, "Image saliency: From intrinsic to extrinsic context," in *CVPR*. IEEE, 2011, pp. 417–424.

[14] X. Ren and J. Malik, "Learning a classification model for segmentation," in *ICCV*. IEEE, 2003, pp. 10–17.

[15] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[16] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*. IEEE, 2005, pp. 886–893.

[17] J. Li, M. D. Levine, X. An, X. Xu, and H. He, "Visual saliency based on scale-space analysis in the frequency domain," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 99, no. PrePrints, pp. 1–1, 2012.

[18] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *in Proc. 8th Intl Conf. Computer Vision*, 2001, pp. 416–423.