04-26

15th IAPR International Conference on Machine Vision Applications (MVA)
Nagoya University, Nagoya, Japan, May 8-12, 2017.

Unsupervised Image Segmentation using Defocus Map and Superpixel Grouping

Chun-Kuei Lo
Department of Computer Science, NTHU,
Hsinchu, Taiwan
kinabcd@gmail.com

Long-Wen Chang
Department of Computer Science, NTHU,
Hsinchu, Taiwan
lchang@cs.nthu.edu.tw

Abstract

Image segmentation is an important and difficult issue in computer vision and image processing. It is categorized into two categories, supervised image segmentation and unsupervised image segmentation. The supervised method are not convenient since it needs the interactions of users. In this paper, we proposed an unsupervised method. It uses a defocus map, edge and color as similarity attributes of pixels or superpixels to generate an edge strength map. Then, we construct a minimum spanning tree with the superpixels and the edge map to divide the image to the foreground and background. In our experiment, our method doesn't need user interaction and the performance is better than previous superpixels grouping methods.

1. Introduction

Unsupervised Image segmentation has been addressed in different ways. Different superpixel grouping methods were applied by many researchers. In [7], edges are grouped using the boundary and region information. Levinshtein et al [4] used angles and shape of edges as similarity and group superpixels with maxflow [6]. Their methods have good performance, but it can't divide the image with complicated background well. In [10], pixels are grouped with a minimum spanning tree pyramid. It can merge the similar pixels as some regions using a threshold and pairwise region comparison, but it cannot find the foreground from the image well. In our method, we use a defocus map and edge and color of pixels or superpixels to group superpixels with a minimum spanning tree to solve segmentation problem.

2. Related Work

Recently, some researchers have exploited the degree of blur on the edge of a single color image to recover the defocus map from the image [1, 5]. We can consider the defocus map a depth information map. It is known that rays from a point of the object placed at the focus distance will be a single point on the sensor. Also, the image will appear sharp and rays from a point of object at distance will reach multiple sensor points and result in a blurred image. In other words, the distances of the blurred region and the sharp region are different. Figure 1 shows the blur estimation method. Figure 1(a) is a sharp region and Figure 1 (d) is a blurred one. After blurring them using a known Gaussian kernel, we can get Figure 1(b) and Figure 1(e). Then the ratio between the gradients of the raw region and their blur version is shown in Figure 1(g). The ratio of blur is

$$ratio\ of\ blur = \frac{Gradient\ of\ raw\ region}{Gradient\ of\ the\ blur\ version}.$$

The sharp region can be calculated with a higher blur ratio. According to the above, we can estimate the blur amount from the region and estimate the distance from the blur amount.

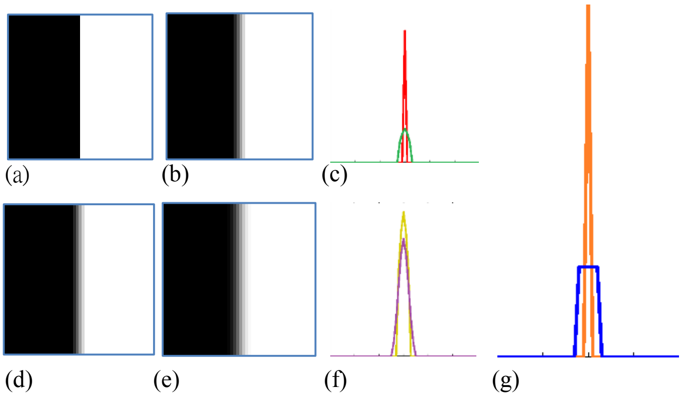


Figure 1. The blur estimation (a) A sharp region. (b) Reblur of (a). (c) Gradients of (a) and (b). (d) A blur region.(e) Re-blur of (d). (f) Gradients of (d) and (e).

Matting Laplacian [9] is an edge-aware interpolation method. It converts the blur estimation from the edges to full defocus map. The interpolation problem can be formulated as the minimum cost function:

$$arg\ min\, E(d) = d^T Ld + \lambda(d - \hat{d})^T D(d - \hat{d}),$$

where d and d ̂ are the vector form of the defocus map and the sparse defocus map. L is the matting Laplacian matrix and D is the diagonal matrix which indicates the pixels on edges in the image and λ is a scalar [1]. The optimal d can be solved by minimizing the equation above.

3. Proposed Method

Our method is divided into six parts shown as Figure 2. Giving an original image, we apply Canny edge detection to obtain the edge image. Second, we use a Defocus map estimation from the image [1] to estimate the distance from the camera to objects in the image. We divide the image into 300 superpixels with SLIC [2]. In the next step, we combine those three kinds of information as the edge strength map which can describe the probability of the edge between two neighboring superpixels. Then we construct a graph where each superpixel is a node and the edge in the graph is the edge between two superpixels in the image and the edge weight is the edge strength. We find the minimum spanning tree from the graph. Finally, we can find an edge with highest probability of be the edge on the original image and disconnect it to divide the minimum spanning tree to two trees of the foreground and background in the image.

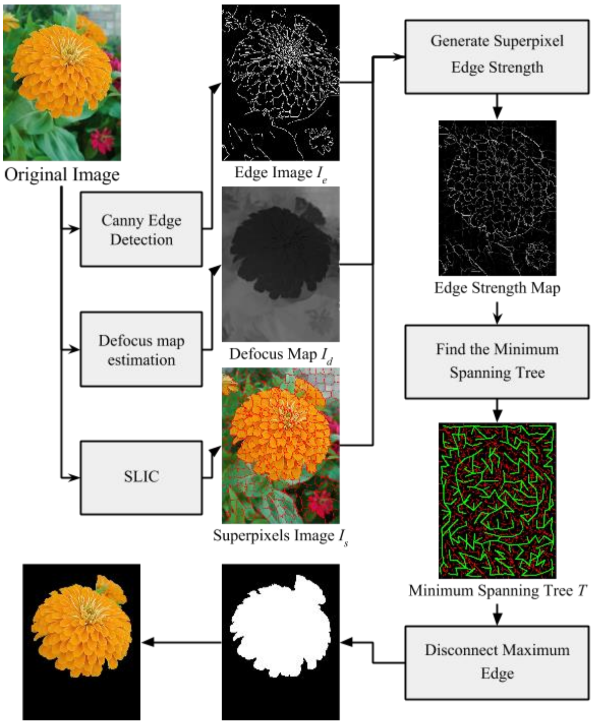


Figure 2. The flow diagram of the proposed segmentation method.

3.1 Edge Strength Map

Let n be the number of superpixels $S_1 \ldots S_n$. E_{ij} is an edge between neighboring superpixel S_i and superpixel S_j with 2 pixels width, where one pixel is in S_i and the other pixel is in S_j. For each edge E_{ij}, the edge strength $e(E_{ij})$ between S_i and S_j can be written as:

$$e(E_{ij}) = \alpha\, d_d(S_i, S_j) + \beta\, d_c(S_i, S_j) + \frac{\sum_{p \in E_{ij}} 1 - dis(p)}{|E_{ij}|},$$

where $1 \leq i, j \leq n$. $|E_{ij}|$ denotes the number of pixels in E_{ij}, and p is a pixels in the edge E_{ij}. d_d is the distance in depth from S_i to S_j on the defocus map I_d and d_c is the perceptual difference between S_i and S_j in in the original image. The defocus map I_d can be considered as a depth information map. d_d is the difference with mean intensity of S_i and S_j on the defocus map I_d and is written as

$$d_d(S_i, S_j) = \left| \frac{\sum_{p \in S_i} I_d(p)}{|S_i|} - \frac{\sum_{p \in S_j} I_d(p)}{|S_j|} \right|,$$

where $|S_i|$ and $|S_j|$ denote the number of pixels in S_i and S_j and p is a pixels in the superpixels. In other words, $d_d(S_i, S_j)$ is the distance in depth from S_i to S_j on the defocus map I_d. We convert the color space of the original image from RGB to CIELAB. CIELAB is a color-opponent space. It is designed to approximate human vision. L is lightness dimension and a, b are the color-opponent dimensions. We can calculate the perceptual difference between S_i and S_j in the image I as:

$$d_c(S_i, S_j) =$$

$$\sqrt[2]{d_L(S_i, S_j)^2 + d_a(S_i, S_j)^2 + d_b(S_i, S_j)^2},$$

where d_L, d_a and d_b are defined as:

$$d_L(S_i, S_j) = \left| \frac{\sum_{p \in S_i} L(p)}{|S_i|} - \frac{\sum_{p \in S_j} L(p)}{|S_j|} \right|,$$

$$d_a(S_i, S_j) = \left| \frac{\sum_{p \in S_i} a(p)}{|S_i|} - \frac{\sum_{p \in S_j} a(p)}{|S_j|} \right|,$$

$$d_b(S_i, S_j) = \left| \frac{\sum_{p \in S_i} b(p)}{|S_i|} - \frac{\sum_{p \in S_j} b(p)}{|S_j|} \right|.$$

Shown in Figure 3, E_{ij} is an edge between superpixel S_i and superpixel S_j. Let p be the pixel on E_{ij}. $dis(p)$ is the distance between the pixel that corresponding edge pixel of p and its nearest edge pixel(gray line) in the edge image I_e shown in Figure 3. We set the length of the diagonal line of the image as 1. Therefore, the value of $dis(p)$ *is* always between 0 and 1. Based on the defocus map I_d, the edge image I_e and the superpixel image I_s, we can compute the edge strength map of the superpixels image.

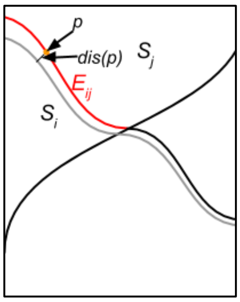


Figure 3. An edge between superpixels.

3.2 Minimum Spanning Tree

After getting the edge strength map, we can use it to generate a minimum spanning tree. Unlike the method [10] which used the minimum spanning tree pyramid to create many minimum spanning trees, we find only one minimum spanning tree containing all nodes. Let G be a graph with superpixels $S_1 \ldots S_n$ and the set of edges such that E_{ij} is an edge between two neighboring superpixels S_i and S_j. A spanning tree is an undirected graph with no cycles and includes all of the superpixels. The minimum spanning tree T is a spanning tree with total of edges weight less than or equal to the total edges weight of every other spanning tree in G. We set superpixels as nodes and edges between superpixels as branches of tree T and the total weight of the tree T is

$$w(T) = \sum_{E_{ij} \in T} e(E_{ij}).$$

The algorithm for finding the minimum spanning tree in G from the edge strength map is written as the followings:

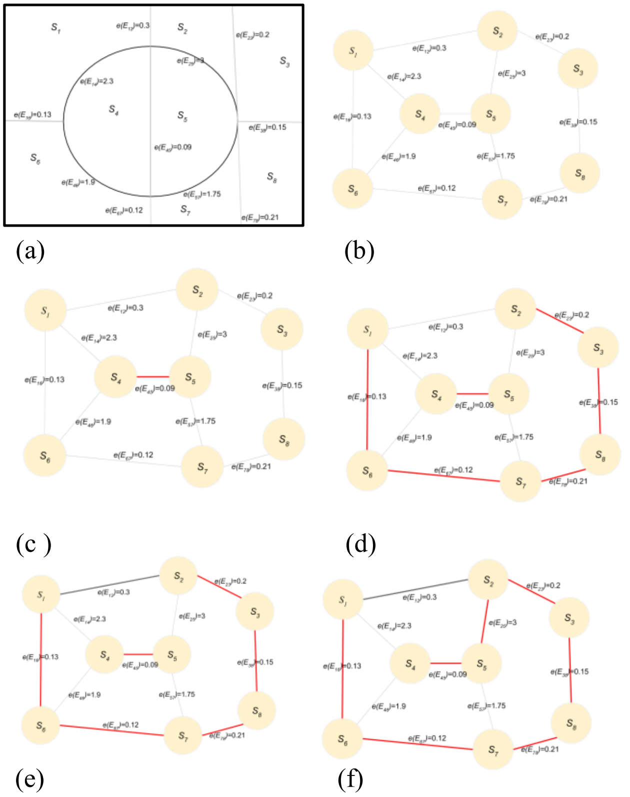
Step 1. Find the unmarked edge E_{ij} having the least weight $e(E_{ij})$ in G.

Step 2. Mark the edge E_{ij}.

Step 3. Connect superpixel S_i and superpixel S_j to make E_{ij} to branch of tree if S_i and S_j are not on same tree.

Step 4. Repeat Step 1, Step 2 and Step 3 until all superpixels are connected.

A simple example is shown in Figure 4. Figure 4(a) shows the superpixels image $S_1 \ldots S_8$. The object contains two superpixels S_4 and S_5. The backgroundcontains S_1, S_2, S_3, S_6, S_7, S_8. Figure 4(b) is the graph generated from the edge stregth map. Figure 4(c) shows an unmark edge with least weight (red one). Mark the edge and connect its two superpixels. Figure 4(d) shows the results by repeating 5 times of Step 1, Step 2 and Step 3. Figure 4(e) shows an unmark edge with least weight (black one). S_1 and S_2 are on the same tree. We mark the edge but don't connect them. Figure 4(f) shows the minimum spanning tree T that all superpixel are connected.



(a) (b)

(c) (d)

(e) (f)

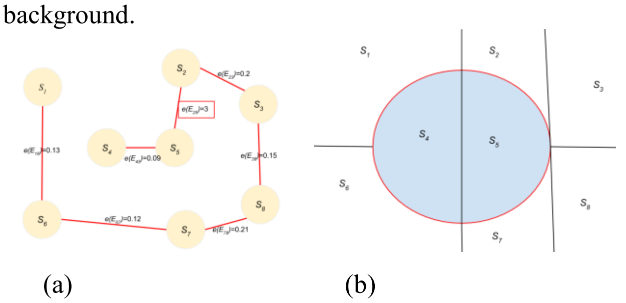
Figure 4. An example for finding minimum spanning tree from original image.

3.3 Disconnect the Edge with Maximum Strength

If we disconnect the edge E_{25}, we can divide the image into the foreground that contains S_4 and S_5 and the background that contains the remaining superpixels. The weight of the edge can describe the probability of being an edge in the original image. Higher weight means higher probability. After finding the minimum spanning tree, most of edges with higher weights were removed on the graph. At this time, the foreground and the background are connected with only one edge with the biggest edge strength. It is clear that the image segmentation is to find the edge having the maximum strength on the minimum spanning tree T in G and disconnect the edge with the biggest edge strength. Then, we can get the spanning tree T_1 that consists S_4, S_5 and the spanning tree T_2 that consists S_1, S_2, S_3, S_6, S_7, S_8. The corresponding superpixels on the original image can be classified into the foreground and background. The edge with the biggest edge strength can be written as

$$\arg\max e(E_{ij}), E_{ij} \in T.$$

Figure 5(a) shows an edge E_{25} having the maximum strength on the minimum spanning tree T. Figure 5(b) shows that two spanning trees: S_4, S_5 and others after disconnecting the edge. The corresponding superpixels on original image can be classified into the foreground and the background.



(a) (b)

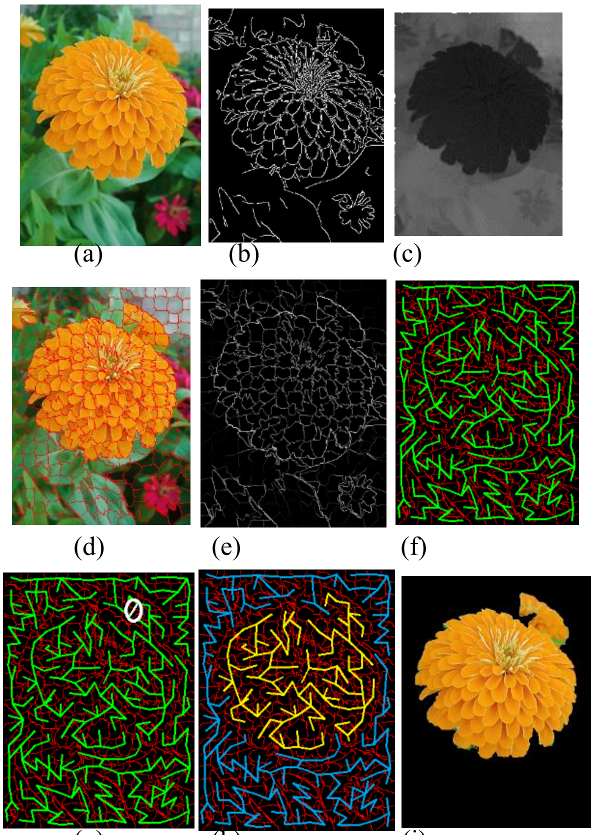
Figure 5. An example for disconnecting the edge with maximum strength on minimum spanning tree

4 Experimental Results

We divide the original image into 300 superpixels for our method and the method by Levinshtein et al [4]. In our method, we set α=3, β=1 and the lower and upper thresholds of canny edge detection are 100 and 200. We show some result of our method and the method by Levinshtein et al [4]. The test images come from MSRA-1000 which is published by Tie Liu et al [8] and used for salient objects with 1000 images, along with pixel ground-truth hand annotations. Figures 6 show the results for the test image. Giving an original image shown in Figure 6(a), we apply Canny edge detection method to obtain the edge image shown in Figure 6(b) and use Zhuo's method to estimate the defocus map shown in Figure 6(c) from the original image. We divide the image into superpixels shown in Figure 6(d) with SLIC. Then we combine those information as the edge strength map shown in Figure 6(e). We find the minimum spanning tree shown in Figure 6(f) from the graph which is generated with superpixels and edge strength map. Finally, we find the edge with maximum edge strength, the white line in Figure 6(g), from the minimum spanning tree and disconnect it. We can get the foreground spanning tree, the yellow tree in Figure 6(h), and the background spanning tree, the blue tree in Figure 4(h). Corresponding to the superpixel image, we label the superpixels in foreground spanning tree as the foreground and the result shown in Figure 6(i). Figures 7 and 8, show the other experiment results including the original images, the ground truths, the results of our proposed method, and the results of Levinshtein's method [4].

5 Conclusions

We proposed an unsupervised image segmentation method with a defocus map estimation method and a superpixels method. It is convenient because it does not need user interaction. Unlike previous methods, our method is easy to control the number of regions. It can divide the original image into the foreground and the background. The defocus map estimation method can't tell whether a blur edge is caused by defocus or blur texture of the original image. We correct it with color and edge information to improve our performance. Our method performs well on the MARS-1000 dataset. Also, our results are more precise than the other segmentation methods using superpixels grouping, especially for the images having good depth of field.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 6. Original image (b) Edge image (c) Defocus map (d)Superpixel image (e) Edge strength map (f) Minimum spanning tree (g) Found maximum strength edge (h) Result after maximum strength edge disconnecting (i) Result after image segmentation.

References

[1] S. Zhuo, T. Sim, "Defocus map estimation from a single image", Pattern Recognition, vol.44 no.9, pp.1852-1858, Sep. 2011.

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-The-art superpixel methods", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.34, no.11, pp.2274-2282, Nov. 2012.

[3] J. Canny, "A Computational Approach To Edge Detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.8, no.6, pp.679-698, Nov. 1986

[4]A. Levinshtein, C. Sminchisescu, and S. Dickinson, "Optimal Contour Closure by Superpixel Grouping", ECCV, p.480-493, Sep. 2010.

[5] C. Tong, C. Hou, and Z. Song, "Defocus map estimation from a single image via spectrum contrast", Optics Letters, v.3, n.10, p.1706-1708, May. 2013.

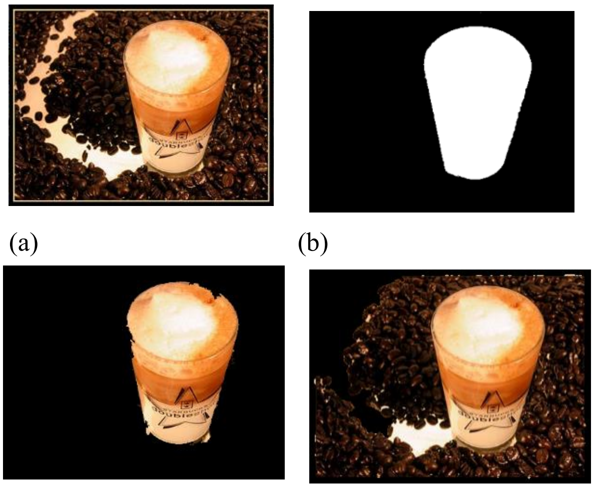
[6] V. Kolmogorov , Y. Boykov and C. Rother, "Applications of parametric maxflow in computer vision", ICCV, p.1-8, Oct. 2007.

[7] J. S. Stahl and S. Wang, "Edge grouping combining boundary and region information", IEEE Transactions on Image Processing, vol.16, no.10, pp.2590-2606, Oct. 2007.

[8] T. Liu, J. Sun, N. Zheng, X. Tang, and H. Y. Shum, "Learning to Detect A Salient Object", IEEE CVPR, p.4270072, Jun. 2007.

[9] A. Levin, D. Lischinski, Y. Weiss, "A closed-form solution to natural image matting", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.30, no.2, pp.228-242, 2008.

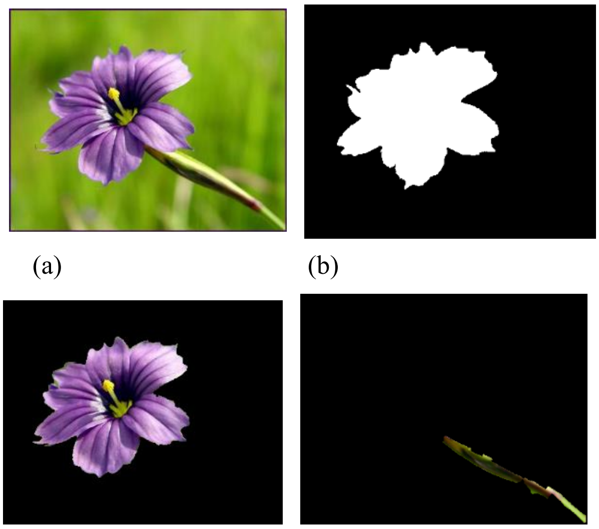
[10] Pedro F. Felzenszwalb and Daniel P. Huttenlocher, "Efficient Graph-Based Image Segmentation", International Journal on Computer Vision, 59(2):169-181, 2004.



(a) (b)

(c) (d)

Figure 7. (a) Original image (b) Ground truth (c) Proposed method (d) Levinshtein's method [4]



(a) (b)

(c) (d)

Figure 8 (a) Original image (b) Ground truth (c) Proposed method (d) Levinshtein's method [4]