

Guided lazy snapping for long thin object selection

Nam H. Tran¹, Dongsun Seo^{1,*}, and Henry T.H.Tu²

¹ Department of Electronics, Myongji University, Yongin-si, Gyeonggi-do, Korea

² Department of Computer Science, FPT University, Hanoi, Vietnam

Abstract. We show a novel way to select long thin objects in an image by enhancing the output of the existing foreground/background image segmentation methods. Most superpixel-based methods fail to select the long thin details, such as legs and whiskers, and extended curves from the main objects. We observe, however, the output without long thin details, can be used as the guided information to obtain the connected components. Based on this observation, our Guided Lazy Snapping method overcomes the limitation of the Lazy Snapping methods (or other alternatives superpixel-based segmentation method) to select long thin objects. The results show that connected components in the image can be selected without having a lot of user interactions (mouse clicks) on each extended parts of the object.

1 Introduction

Image segmentation [1-10] is one of the most challenging tasks of image processing field, which has various novel proposed approaches in its long history in computer vision. The ultimate goal is to remove undesired regions and extracts the regions-of-interest to reuse or to transform.

Graph cut technique [1] had become the heart of many current state-of-art image segmentation techniques [2-4]. The image segmentation problem is modelled as binary labelling problem such that each pixel is assigned a unique label which denotes object and background classes. Based on the samples given by a user to the foreground and background set, the graph cut minimizes an energy function consisting of the data and prior knowledge as constraints. In general, the data constraint restricts the desired solution to be close to the observed data and the prior constraint confines the desired solution to have a form agreeable to the prior knowledge.

However, the graph cut approach has a shortcoming to a segment with thin elongated structure due to two basic reasons: the trade-off between the data constraint and prior constraint, and the bias towards shorter boundaries (also known as “shrinking bias”, or the twig problem). To illustrate the problems, the segmentation process and some results by the graph cut are shown in figure 1. First, the user provides some samples for both the object and background by drawing some strokes (a). If the influence of prior constraint is small (b),

* Corresponding author: sdsphoto@mju.ac.kr

the extracted object contains some grass regions belong to the background. Otherwise (c), the thin elongated details such as the legs of the insect are not included.

Inspired by Boykov and Jolly, Li et al. proposed Lazy Snapping [2] – an interactive image cutout system with simple user actions to extract the true object boundary efficiently. Furthermore, the framework applies the graph cut solution based on pre-computed over-segmentation generated by the Watershed algorithm to increase performance significantly without sacrificing pixel-accurate quality.

The idea of using over-segmentation as pre-processing refers to as superpixel segmentation. The concept of “superpixel” was first time introduced by Ren et al. [11] in 2003. It aims to naturally represent an image to superpixel map with many desired properties that mention in [12]. Unfortunately, superpixel segmentation algorithms [13-17] tend to group pixels that are small into a bigger superpixel, such as the detailed parts of the object. For this reason, the natural object boundaries are not conserved. In case of figure 2, the superpixel maps of the input image are generated by some well-known superpixel algorithms: Linear Spectral Clustering (LSC) [12], Simple Linear Interactive Clustering (SLIC) [13], Superpixels extracted via energy-driven sampling (SEEDS) [14], and SEEDS Revised using Mean pixel (reSEEDS) [15]. We can observe that the horn of the deer or the rostrum of the bird are not segmented well. This shows the trade-off between quality and efficiency of the superpixel-segmentation-based methods, and it indicates that pixel quality in thin elongated object is crucial.

In this paper, we propose a unique algorithm to overcome challenging shrinking bias problem. We retrieve the lost information with a novel post-processing step based on the uncompleted graph cut result. In a little more detail, we assume that a part of the object has been segmented using Augmented Lazy Snapping [9] as in figure 1(c). A structure guidance map is built by applying Guided Filter [16] to the partly cutout result as shown in figure 1(d). Finally, the user can cutout the main object by dragging the progress bar with the instant feedbacks in our framework. Experimental results show that our approach can preserve thin elongated structure objects more accurately than conventional methods. It is worth noting that our method also efficiently extracts the background pixels inside the object, as shown in figure 1(e) and restrict the influence of blending effect to long thin image segmentation.

2 Related works

Shrinking bias of graph cut approach is firstly addressed by Kolmogorov and Boykov [5]. The concept of flux is attempted to improve segmentation of thin elongated objects by integrating into the graph cut solution. The method produces better results but only on the grey-level images due to the limitation in definition of the flux. Later, Vicente [7] introduces the combination of the graph cut and Dijkstra algorithm in the name of DijkstraGC. Based on some additional inputs, the Dijkstra algorithm finds the missing paths to the main object with the controllable width of the structure. However, this approach requires highly precise user inputs at the end point of long thin parts and heavily recall max flow computations. Another work which tries to overcome shrinking problem was proposed by Lempitsky [8]. The tightness prior is introduced to prevent the result from excessive shrinking. Unfortunately, the local energy minimization sometimes produces poor results because of getting stuck in poor local minima. Candemir [10] integrates the statistical significance measure to build a new graph structure and redefine the relationship between data and smoothness. The segmented quality of results is slightly improved between traditional graph structure and the proposed graph structure. Other interesting approach is introduced by Dong et al. [17] by applying Random Walker algorithms on the graph including some new auxiliary nodes with label prior. However, the similarity

between the object and background may lead to incorrect segmentation of some foreground regions.

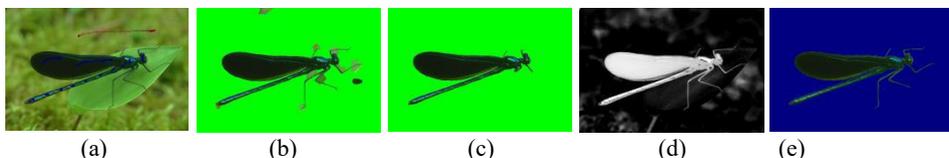


Fig. 1. Thin elongated image segmentation for a user input (a) using graph cuts with the small (b) and large (c) influence of prior constraint, post-processed structure guidance map (d), and our final result (e).

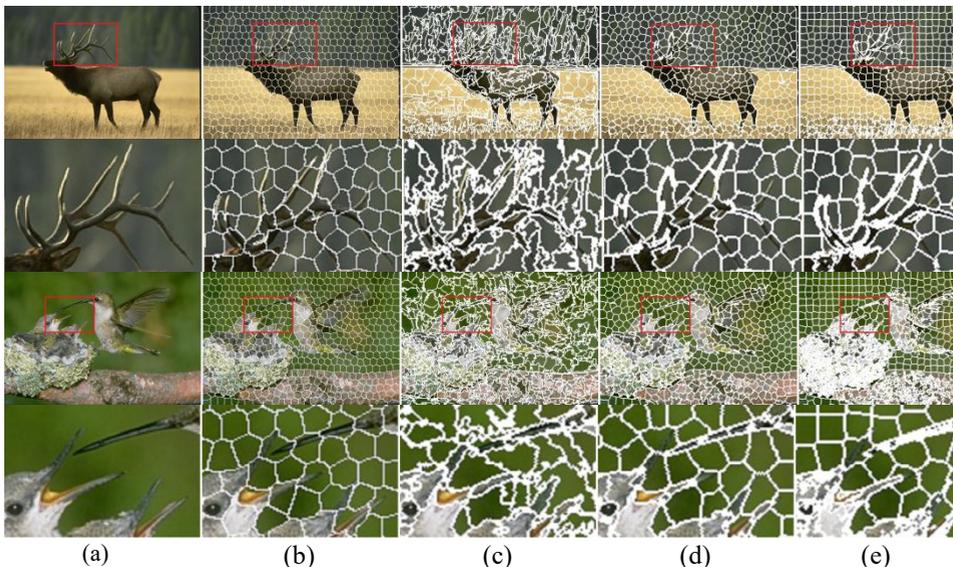


Fig. 2. Superpixel segmentation of the original images (a) using SLIC (b), SEEDS (c), reSEEDS [15](d), and LSC (e). All the superpixel images are generated by 600 superpixels with different regularity and compactness. The corresponding marked parts show the under-segmented regions, which obviously affect the graph cut results.

3 Structure-transferring filter as post-processing in augmented snapping framework

We first review the Augmented Lazy Snapping method and the influence of superpixel size to long thin structure image. We then investigate the main reasons lead to poor results of the current framework in solving the twig problem. Finally, we consider Guided filter as a post-processing step to recover the structure of details which is cutout in the graph cut process. We believe that the structure-transferring ability of the guided filter is outstanding to solve the existing problems. Based on the structure guidance map, we can easily extract the desired object with binary thresholding method. In addition to the significant improvement of long thin structure, the effect of the proposed step on the overall segmentation processing time is negligible.

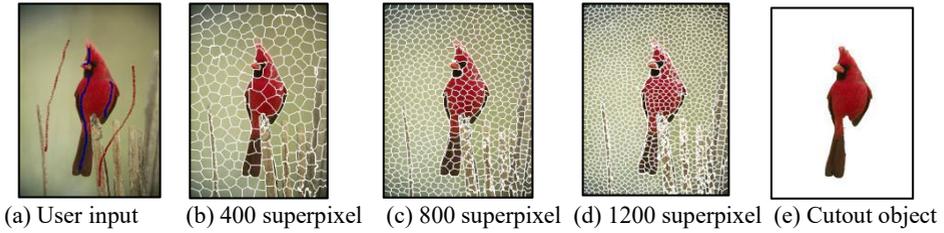


Fig. 3. Augmented Lazy Snapping framework consisting an object marking step (a) and a pre-segmentation step based on reSEEDS. In (a), the (blue) strokes are samples indicating the object, and the another (red) line indicates the background. The pre-segmentation step is flexible to generate 400 superpixels (b), 800 superpixels (c), and 1200 superpixels (d), which is not provided in the traditional Watershed algorithm. Finally, the object is extracted in (e).

3.1 Reviewing augmented lazy snapping

In the Augmented Lazy Snapping (ALS) paper [9], the authors studied the middle layer of 4-step pipeline that introduced by Lazy Snapping [2] to cut out the object from the background. In the work, by optimizing the pre-segmentation step, we could improve significantly the performance of framework with little loss in pixel quality. The reason is the ability to control over the size where the number of generated superpixel is the key to determine the Graph Cut performance. The experimental results show that the reSEEDS is the optimal algorithm in the selected methods to replace the traditional Watershed algorithm in the middle layer.

Figure 3 shows the more details about the ALS framework. The main goal of our work is to investigate the effect of the middle layer to the framework, so the last step in the Lazy Snapping's pipeline is not mentioned. As it can be seen in figure 3(b, c, and d), reSEEDS is well designed by balancing two important factors of the visual superpixel representation: the regularity and the compactness. To be more details, the compactness corresponds to the form of superpixel and the regularity refers to the positioning and the size of superpixel. In general, the compactness needs to be trade-off against the regularity, which presents the precise of object boundaries. The balance is the main reason makes reSEEDS outperforms other selected algorithms in term of quality to be chosen in the ALS. Nevertheless, the ALS framework still has some problems to select the thin elongated structure object.

3.2 Thin elongated object segmentation problems

As we mentioned before, there is a trade-off between segmentation quality and time in superpixel segmentation based on the graph cut. The ability to control over the size of generated superpixels in Augmented Lazy Snapping can easily overcome the under-segmentation problem by reducing the size of superpixel as small as its structure fits into long thin parts of the object. However, even the superpixel size is small enough to fit into the region-of-interest, the framework still produces poor segmentation results. It can be seen in figure 4, the legs and a part of the antennas of the insect are cut off in the segmented results while the body is well-segmented. Therefore, the superpixel size in the pre-segmentation step does not cause the poor object quality of the framework.

We discovered that there are three main reasons causing the poor results of the framework. First and foremost, the shirking bias problem in Graph Cut foundation. The prior constraint presents the energy summation over the boundary of the extract regions. At the thin elongated boundaries of the object, the graph cut algorithm may cut the boundary along shorter paths because a short expensive boundary may cost less than a very long cheap one. Equally important, it is very hard for the user to provide precise samples in

the thin elongated regions of the object. The lack of important information and weak data constraint lead to uncompleted global energy minimization solutions. Last but not least, the third problem is blending effect, which also known as **bokeh** [18] – the blur produced in the out-of-focus parts of an image produced by a lens. Bokeh is often mostly visible around small, thin and elongated regions, especially in the natural images. An example of blending effect is introduced in figure 5. The presence of ambiguous or low contrast regions in the image always makes the segmentation task more and more challenging.

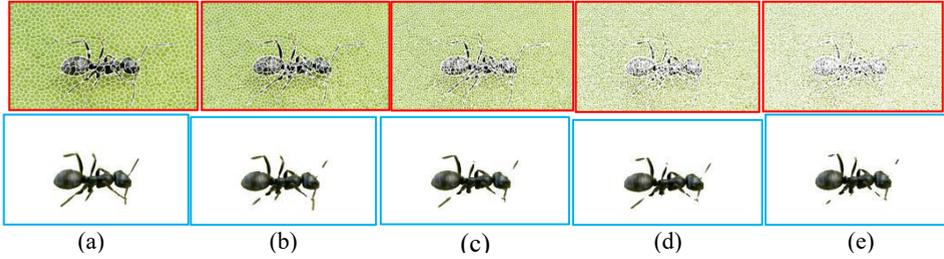


Fig. 4. Pre-segmentation results with 1000 (a), 2000 (b), 4000 (c), 8000 (d) and 16000 (e) superpixel map (first row) and the following segmented results (second row) of the Augmented Lazy Snapping framework based on input image *Insect_1*. Despite of the small size of generated superpixel, the segmented results still lose the long thin details of the object.

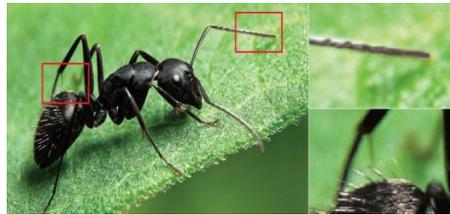


Fig. 5. Example of blending effect occurs in the thin elongated regions of the object that are marked by red rectangle.

Introduced by He et al. [16] in 2013, the guided filter can transfer the structures of the original image to the filtering output based on segmentation methods. To show the structure-transferring ability of the method, the authors presented *guided feathering* application, which is very useful to recover the hair, even though the filtering input is very rough. In this paper, we will concentrate on how the structure-transferring filtering can improve the thin elongated Augmented Lazy Snapping results. We refer readers to [16] for the detail formulation of the guided filter and how to construct it.

To recover the structure from the guidance input to the output, the guided filter requires a binary mask to appear an alpha mate near the object boundaries. The binary mask can be generated from the result of ALS framework and is used as the filter input. Moreover, it has a role as a navigator for the guided filter to locate which parts need to be focused (the object) from the rest (the background). Consider figure 6, there are two cases (a) and (b) that the binary masks are created and applied through the filter.

We can observe in the structure guidance map of case (b) that the hair of Ostrich bird is recovered clearly if the neighboring regions of long thin structure exist in the navigation binary mask. Otherwise, the filtering output is blurred and not so clear as shown in the case (a). To provide better illustration, we mark some thin elongated regions in the guided filtering output (a) and (b) and zoom in the detail parts as figure 6(c). It indicates that the quality of navigation binary image is crucial to retrieving the lost information. Based on our experiments, we believe that the guided filter can offset the lost thin elongated details due to the reasons we mentioned in the previous section.

4 Guided lazy snapping

In this section, we introduce the details about our segmentation technique. The framework consists of two steps: a quick object marking step and a simple thin elongated recovering step.

4.1 Object marking

In the object marking step, the user draws a few strokes on the image to specify the object. As shown in figure 7(a), the blue lines and the red lines are samples for the foreground and background. Based on the displayed segmentation result, the user decides if more samples are needed to be provided.

The goal of object marking step is to provide a quick cut that covers all the not thin elongated structure regions of the object. Besides, this step takes the majority of overall processing time of the framework. So that the pre-segmentation step in figure 7(b) plays an important role as improving the performance of the framework. As we discussed before, the size and the number of superpixels do not affect the segmented result in the object marking step. In order to optimize the performance of the framework, in our experiments, we divide input images into a usable number of superpixels. Therefore, the graph cut shows extremely fast response without sacrificing object quality in the regions do not belong to the thin elongated regions. Finally, a part of the object is segmented as the figure 7(c).

4.2 Thin elongated recovering

Based on the graph cut result, the binary image mask is automatically generated as the first component of the thin elongated recovering step as shown in figure 7(d). We apply Guided Filter to recover the structure of object with the guidance input image and the filtering input - binary mask. The filtering output is shown as the figure 7(e). Based on the filtering output, the user can drag the progress bar to select threshold value. The framework produces instant feedback by applying the simple Binary Thresholding, which is formulated as,

$$object(x,y) = \begin{cases} src(x,y), & \text{if } filter(x,y) \geq thresh \\ 0, & \text{if } filter(x,y) < thresh \end{cases} \quad (1)$$

to the filtering output and displays the final thin elongated object as we observed in figure 7(f). We believe that this is simple user interface for segmenting challenging objects.

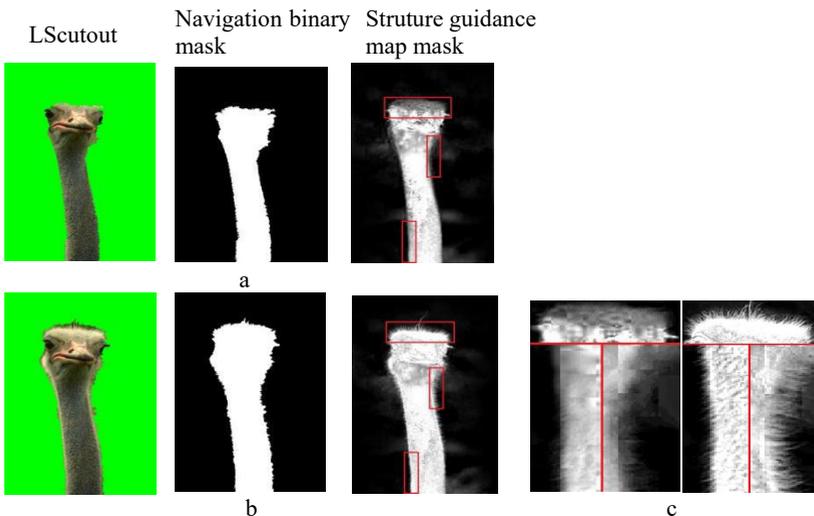


Fig. 6. Structure guidance map that is guided by the different navigation binary masks. From the ALS cutout result, the navigation binary mask is generated and filtered. The long thin structure is completely recovered if the neighbouring regions existed in the binary mask and vice versa. (a) low-quality-navigation map, (b) high-quality-navigation map, (c) some parts of the image marked by red boxes.

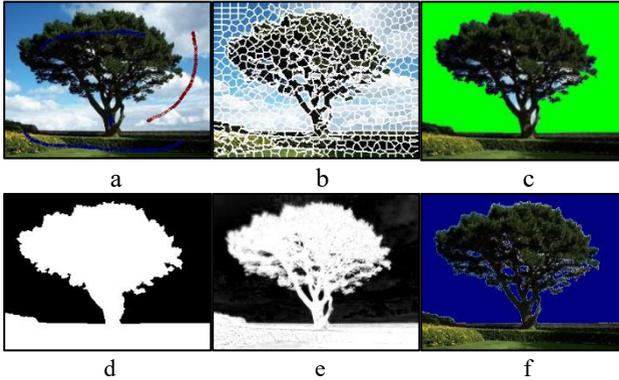


Fig. 7. Thin elongated image segmentation using graph cut based on user input Tree_1 (a). The pre-segmentation step (b) improve performance of the framework. A part of object with background is extracted (c) by the ALS framework. The binary image (d) is generated from cutout object. The post-processed recovered guidance map (e) and our final cutout result (f).

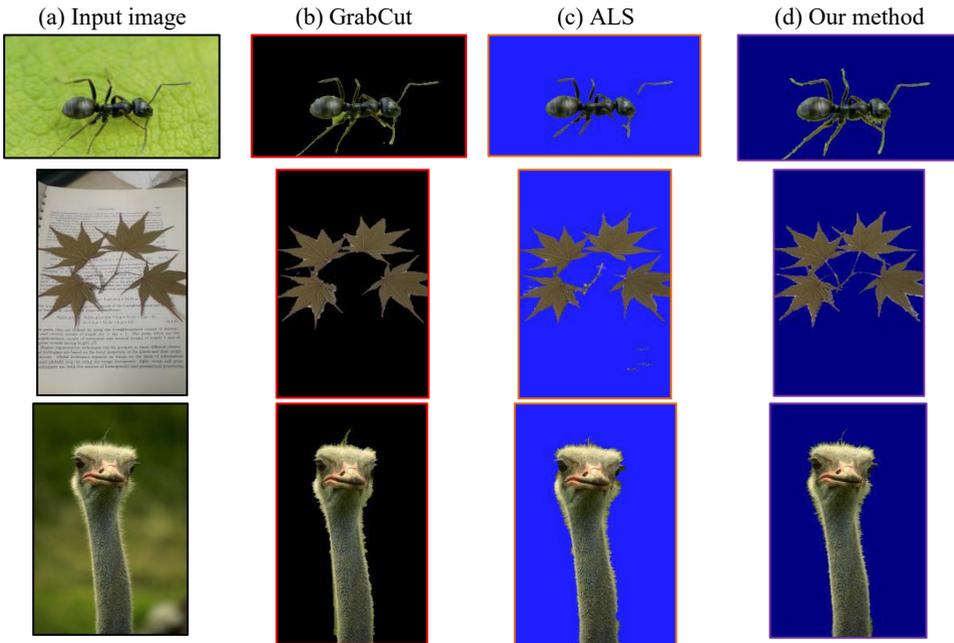


Fig. 8. Thin elongated image segmentation for a user input (a) using GrabCut [3] (b), Augmented Lazy Snapping [9] (c), and our method (d). The input images are the combination of thin elongated cases of Berkeley Segmentation dataset [19] and our dataset. The image names from top to bottom: Insect_1, Leaf, and Ostrich.

5 Experimental results and discussion

5.1 Thin elongated segmentation results

We compare our method to two interactive image segmentation frameworks including GrabCut [3] and Augmented Lazy Snapping (ALS) [9]. The implementations of the selected frameworks are based on the publicly available code. The experiments are performed on the combination of our own dataset and Berkeley Segmentation dataset [19].

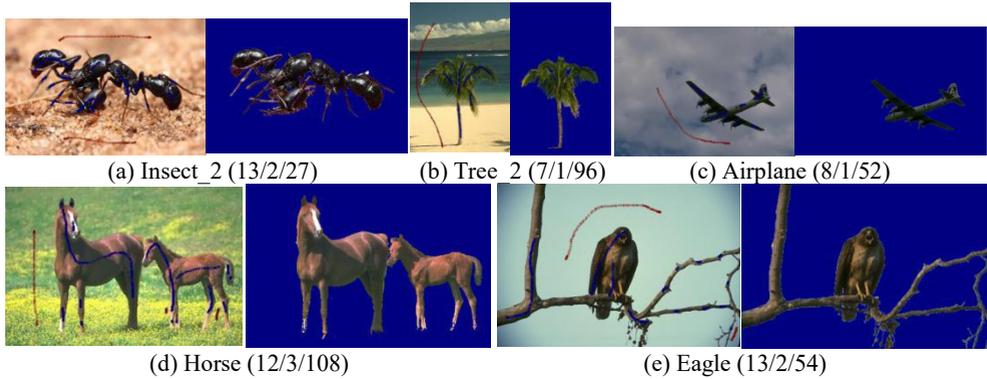


Fig. 9. More experiment results with the images of Berkeley Segmentation dataset [19] combined with our own dataset. The number in the brackets denotes the number of foreground markers, the number of background markers in the object marking step and the threshold value in the thin elongated recovering step. Each pair of images shows the user input and the final result, respectively.

Table 1. Guided Lazy Snapping: Comparison of processing time with and without recovering step.

Image	Image Dimension	Number of superpixels	Nodes ratio	Edges ratio	Lag without recovering step	Lag with recovering step
Tree 1	(300, 225)	400	168.8	118.6	0.596	0.640
Horse	(481, 321)	600	257.3	202.2	0.894	0.969
Tree 2	(321, 481)	600	257.3	202.7	0.904	1.004
Insect_1	(600, 338)	800	253.5	203.3	1.155	1.282
Insect_2	(900, 600)	1000	540.0	453.8	1.724	2.023
Leaf	(747, 1328)	1200	826.7	833.6	2.834	3.344

- The nodes/edges ratios are the number of nodes/edges divided by the number of superpixels/connection between superpixels in the pre-segmentation step.
- The feedback lag is the delay from when the user releases the mouse to when the object is extracted/recovered.

Figure 8 shows some cases in our experiment to select thin elongated structure objects. Due to the shirking bias problem, the GrabCut and ALS methods cuts off the legs of the insect, the twig of the leaf or a part of the Ostrich bird hair. In the results of GrabCut, the long/thin parts are less lost compare to the ALS’s result. However, the extracted objects contain pixels belong to background. Besides, the ALS method provides corrected labelling, which mean the cutout results have less background pixels compare to GrabCut. On the other hand, it can be observed that our method preserves thin elongated structure better comparing to the others.

Figure 9 shows more results of our method with user input and the specific threshold value in each case. We have tested our approach for 40 images with the thin structures which may have only 1-2 pixels wide as the legs of the insect or the hair of the bird; or have fewer pixels as the branches of the tree. Despite of the fact that we have the shrink bias problem on the Graph Cut foundation or the blending effect in the input data, our method still faithfully retrieves the thin elongated structures.

5.2 Overall segmentation time

In our experiment, the time in second, is computed on a desktop PC Inter® Core™ i5-7500 3.40GHz, 8.00 GB memory. To quantitatively evaluate the overall processing time of the segmentation of the proposed approach, we applied it to different dimensional images in two cases.

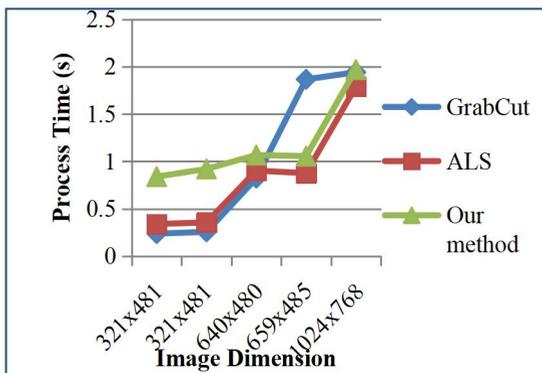


Fig. 10. Comparison of average long thin segmentation times of GrabCut, ALS and our method.

In the first case, to prove that the thin elongated recovering step affects negligibly on overall segmentation processing time, we compared the lag with and without the step in the experiments. Table 1 shows that the processing time to extract object in one image increase linearly as the number of superpixels increases. Also, the difference with and without the second step of the framework is clearly observed. The reason is that the guided filter naturally has an $O(N)$ time non-approximate algorithm, and independent of the setting parameters.

In the second case, we compare the processing time of the interactive image segmentation frameworks. As it can be seen from figure 10, there is not much different in the experiment of the processing time between GrabCut, ALS and our method, when the image size increases, ALS and our method will outperform the GrabCut because the efficiency of pre-segmentation step.

6 Conclusion and future work

In this paper, we have combined guided filter with Augmented Lazy Snapping to overcome to select thin elongated structure object. Unlike from the recent trend that integrates different measure as flux or statistical significance into the graph cut methods, we introduced Guided Lazy Snapping that uses the guided filter as post-processing to retrieve the lost information due to shrinking bias and other problems of conventional methods for long thin image segmentation. The experiments show that our method is easy to use, produces better quality than existing thin elongated image segmentation framework. The binary thresholding at the structure recovering step is perhaps the shortcoming in the framework. We believe that with the important recovered information in the filtering output, the thin elongated objects may automatically be segmented. We intend to explore it in the future.

Acknowledgments

This work was supported by the IITP funded by the Ministry of Science, ICT, and Future Planning (#2014-3-00501).

References

1. Yuri Y. Boykov, M. P. Jolly: 'Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images'. Proc. Int. Conf. Computer Vision (ICCV), Vancouver, Canada, July 2001, pp. 105-112

2. Y. Li, J. Sum, C. K. Tang et al.: 'Lazy Snapping'. Proc. Int. Conf. ACM SIGGRAPH, California, USA, August 2004, pp. 303-309
3. C. Rother, V. Kolmogorov, A. Blake: 'GrabCut-Interactive Foreground Extraction using Iterated Graph Cuts'. Proc. Int. Conf. ACM SIGGRAPH, California, USA, August 2004, pp. 304-314
4. A. Agarwala, M. Dontcheva, M. Agrawala et al.: 'Interactive digital photomontage'. Proc. Int. Conf. ACM SIGGRAPH, California, USA, August 2004, pp. 294-302
5. V. Kolmogorov, and Y. Boykov: 'What metrics can be approximated by geo-cuts, or global optimization of length/area and flux'. Proc. Int. Conf. Computer Vision (ICCV), Beijing, China, October 2005, pp. 564-571
6. J. Jia, J. Sun, C. K. Tang et al.: 'Drag-and-Drop Pasting'. Proc. Int. Conf. ACM SIGGRAPH, Massachusetts, USA, July 2006, pp. 631-637
7. S. Vincente, V. Kolmogorov, and C. Rother: 'Graph cut based image segmentation with connectivity priors'. Proc. Int. Conf. Computer Vision and Pattern Recognition, Anchorage, USA, June 2008, pp. 1-8
8. V. Lempitsky, P. Kohli, C. Rother et al.: 'Image segmentation with a bounding box prior'. Proc. IEEE Int. Conf. Computer Vision, Kyoto, Japan, September 2009, pp. 277-284
9. H. N. Tran, D. Seo, and T.H. Tu: 'Augmented Lazy Snapping: how does pre-segmentation help Lazy Snapping framework?'. To appear in Proc. Int. Conf. Graphic and Image Processing (ICGIP), Qingdao, China, October 2017.
10. S. Candemir, and Y. S. Akgul: 'Statistical significance based Graph Cut segmentation for shrinking bias'. Proc. Int. Conf. Image Analysis and Recognition, Burnaby, Canada, June 2011, pp. 304-313
11. X. Jen, J. Malik: 'Learning a classification model for segmentation'. Proc. Int. Conf. Computer Vision (ICCV), Nice, France, October 2003, pp. 10-17
12. Z. Li, J. Chen: 'Superpixel Segmentation using Linear Spectral Clustering'. Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR), Boston, MA, October 2015, pp. 1356-1363
13. R. Achanta, A. Shaji, K. Smith et al.: 'SLIC Superpixels Compared to State-of-the-Art Superpixel Methods'. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012, 10, vol. **34**, no. 11, pp. 2274-2282
14. M. van den Bergh, X. Boix, G. Roig, et al.: 'SEEDS: Superpixels extracted via energy-driver sampling'. Proc. European Conf. Computer Vision (ECCV), Florence, Italy, October 2012, pp. 13-26
15. Stutz, A. Hermans, B. Leibe: 'Superpixel Segmentation using Depth Information'. Bachelor thesis, RWTH Aachen University, Aachen, Germany, 2014
16. K. He, J. Sun, and X. Tang: 'Guided Image Filtering'. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013, vol. **35**, no. 6, pp. 1397-1409
17. X. Dong, J. Shen, L. Shao et al.: 'Sub-Markov Random Walk for Image Segmentation'. *IEEE Transactions on Image Processing*, 2016, vol. **25**, no. 2, pp. 516-527
18. 'Bokeh', <https://en.wikipedia.org/wiki/Bokeh>, access 14 November 2017
19. D. Martin, C. Fowlkes, D. Tal et al.: 'A database of Human Segmented Natural Images and its Application to evaluating Segmentation Algorithms and measuring ecological statistics'. Proc. 8th IEEE Int. Conf. Computer Vision (ICCV), Vancouver, Canada, July 2001, pp. 416-423