Improvements on the Superpixel Hierarchy Algorithm with Applications to Image Segmentation and Saliency Detection

Marcos José Canejo and Carlos Alexandre Barros de Mello

October 1, 2020

Improvements on the Superpixel Hierarchy Algorithm with Applications to Image Segmentation and Saliency Detection

Marcos J.C.E. Azevedo and Carlos A.B. Mello

Centro de Informática, Universidade Federal de Pernambuco, Brazil
{mjcea,cabm}@cin.ufpe.br

Abstract. Superpixel techniques aim to divide an image into predefined number of regions or groups of pixels, to facilitate operations such as segmentation. However, finding the optimal number of regions for each image becomes a difficult task due to the large difference of features observed in images. However, with the help of edge and color information, we can target an ideal number of regions for each image. This work presents two modifications to the known Superpixel hierarchy algorithm. These changes aim to define the number of superpixels automatically through edge information with different orientations and the Hue channel of the HSV color model. The results are presented quantitatively and qualitatively for edge detection and saliency estimation problems. The experiments were conducted on the BSDS500 and ECSSD datasets.

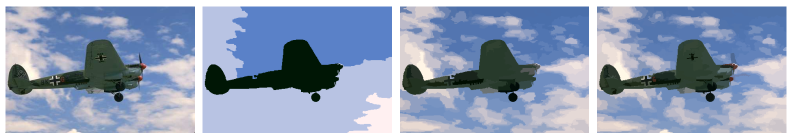
Keywords: superpixel · image segmentation · saliency map

1 Introduction

Superpixel techniques aim to group pixels under a unique label, creating a so called superpixel, which can be used for different types of applications [25, 8, 21]. Its concept was first introduced in [20] when they were used for segmentation based on Gestalt laws. After that, superpixels were more widely used through the well-known Simple Linear Iterative Clustering (SLIC) algorithm [1]. The input for SLIC is the k number of equally-sized superpixels in the final image. This number is essential for a good segmentation.

Image segmentation plays a key role in computer vision. Its main goal is to divide an image into small parts for further object detection and recognition. It is the first and fundamental step of several applications. The segmentation of an image can be done in many ways, without the use of superpixels (as in [18, 17, 16, 5]), or with them (as [27]).

The result of a superpixel processing can be seen as an intermediate representation of an image. In Fig. 1, we can see different ways of representing an image through different number of superpixels. It is easy to identify the relationship between this number and the image visualization. For example, for four superpixels (Fig. 1.b), the aircraft is represented in a very simple way with few details. For 256 superpixels (Fig. 1.d), some details are perceptible, just as for the intermediary value of 128 superpixels (Fig. 1.c). Despite the loss of details, the shape of the aircraft has been almost completely preserved in all cases. Thus, for example, in an application that aims to detect the position of the aircraft, the reduction of details would help to simplify the task.



(a) (b) (c) (d)

Fig. 1: (a) Original image and its segmented version for (b) 4, (c) 128 and (d) 256 superpixels.

Among several characteristics found in an image, we can mention color (which can be related to similarity information) and edge (which can be related to discontinuity information). Both color and edge can compose an important knowledge to guide the process of superpixel segmentation. The Superpixel Hierarchy algorithm (SH) [22] makes use of color and edge information to perform the segmentation through the region merging in which the pixels are weighted and, from their similarities, grouped together. The work also shows that edge information is very useful in the process of merging regions. However, just as SLIC, they do not define the number of superpixels in an image automatically; it must be set by the user according to the image.

This paper proposes modifications to the original version of SH algorithm to make its execution automatic for each image. The first modification is related to how the number of superpixels can be defined from the number of distinct tones present in the image. The second modification is by improving the edge information used.

In the following section, some related works are described. In Section 3, the proposed modifications are explained. Section 4 presents the experiments and quantitative and qualitative results which are described and compared. Section 5 concludes the paper with possible future works.

2 Related works

To evaluate the performance of our proposal, the superpixel images are submitted to different edge detection algorithms. Among them, Structured Forest Edges (SFE) [6] (which is used in [22] to feed the edge information), Holistically-Nested Edge Detection (HED) [23] (with its Deep Learning approach), and the well-known Globalized Probability of Boundary (gPb) [13]. For this reason, these algorithms are summarized in this section.

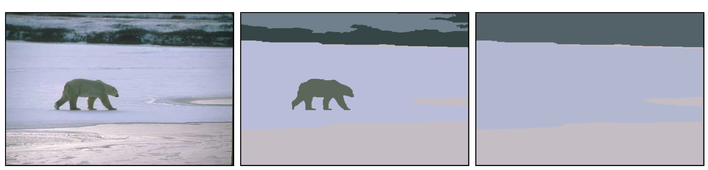
The Structured Forest Edge Detection (SFE) is used in the original version of SH due to its low processing time. The authors also evaluate the running time for superpixel generation (which is out of our scope). SFE detects edges using a previously trained Random Forests (RF) to label each pixel of the input image as edge or not in 16x16 masks.

The Holistically-Nested Edge Detection (HED) is part of a category of edge detection algorithms that make use of Deep Learning techniques, in particular Convolutional Neural Networks (CNN) [10]. Its architecture is exclusively formed by convolution layers to learn the hierarchical representation of the edges (size, shape, etc.).

Finally, the Globalized Probability of Boundary (gPb) edge detector is an extension of Pb [15], but with edges definition based on global information from eight different orientations. Local information of color, texture, and brightness, is also used. The great advantage of gPb over other algorithms is exactly in collecting information from various orientations, having success where several algorithms fail.

3 Proposed Modifications on Superpixel Hierarchy Algorithm

Superpixel Hierarchy [22] has two parameters that guide the pixels grouping process: the number of superpixels and the edge information. The number of superpixels represents the number of regions present in the final image. Fig. 2 exemplifies a result for five and three superpixels. As we can see, the segmented image for 3 superpixels does not have the region of the bear. Thus, it is easy to conclude that the number of superpixels has major impact on the final result.



(a) (b) (c)

Fig. 2: Example of (a) an original image, its SH segmented version for (b) 5 superpixels and (c) 3 superpixels, where the bear was suppressed.

The second parameter (edge information) has also a big impact on the final result. In the original work [22], the authors show the difference between the algorithm with and without edge information. Thus, by adding more information to the process, they achieved better results. However, still according to the authors, the use of different edge detection algorithms does not result in a big difference, although it exists in a small form.

The first objective of our method is to try to identify a suitable value for the number of superpixels for each image, automatically. Regardless how many different colors can be represented in an image in RGB color model, usually, just few distinct "tones" can be clustered. In this sense, "tone" is an informal way to define the hue which is the attribute of the pixel to be clustered. For example, the blue hue can be clustered in many different colors (as a dark blue or a light blue) [19]. Thus, the first step is to identify the number of distinct "tones" present in the image and define it as the number of superpixels to be used.

As a first improvement, the input image is converted into the HSV color model, which is inspired by the human visual system. As explained before, it is important to group pixels with similar hue, this is why the image is converted into HSV (in fact, just the hue channel is needed). The hue component is used for grouping. Any clustering algorithm could be used in order to group similar pixels. We propose the application of the Mean Shift (MS) [9] clustering algorithm to the hue channel. One of the reasons to choose MS is that there is no need for a training step in proper datasets, in contrast to supervised machine learning techniques. The MS presents as disadvantage its high computational cost, but the kernel bandwidth value is its unique parameter which has several automation proposals for different applications [28, 7]. We have decided to perform the tests for a single bandwidth value for all images; this value was chosen from empirical tests, observing the metrics used in the experiments. Thus, for this work, the bandwidth value is equal to 0.75. MS automatically returns the number of groups present in the hue channel of the input image; a value close to the number of colors observed by the human eye. This is considered as the number of superpixels. Fig. 3 shows examples of images with their number of superpixels defined using MS.

As mentioned previously, the second parameter of the method is the edge information that helps the SH algorithm during the fusion process between the different regions. In [22], the authors show that the use of this information is crucial to achieve good quality results, but different edge detection methods did not caused significant differences. Thus, as a second improvement, instead of providing only a single edge image to the algorithm, we propose the use of the gPb edge images created for each of the eight different orientations, as shown in Fig 4. For these gPb images, superpixel images are created. Thus, at the end of the process, we have eight superpixel images for the input image. The resulting SH images for each gPb orientations are shown in Fig. 5 for the bear image example; in this figure, we can see how the edge information impacts the generation of superpixel images. For the edge image in Fig. 4a, which has a limited quantity of edge information, the superpixel image created (Fig. 5a) shows no similarity to the original image.

Fig. 3: Examples of SH result with number of superpixels defined automatically by Mean Shift over the hue version of the images. Original images are presented in the left column and their respective superpixel versions are in the right column.

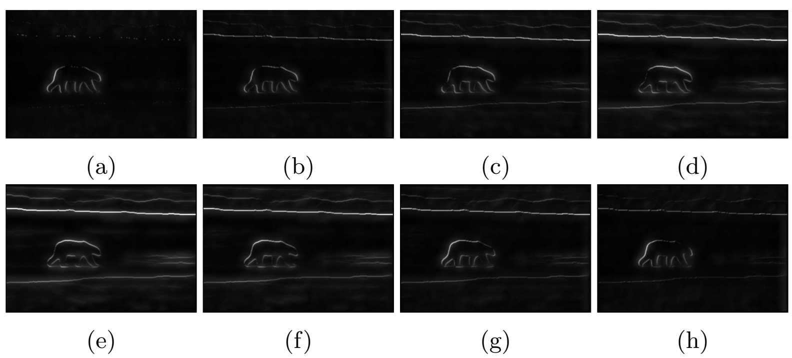


Fig. 4: gPb images for eight different orientations for the bear image of Fig. 2a.

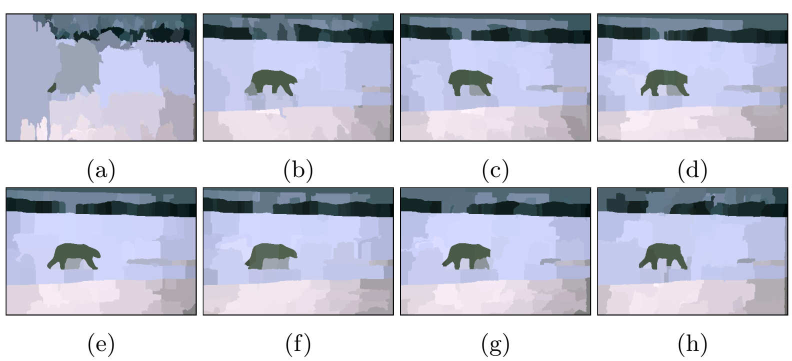


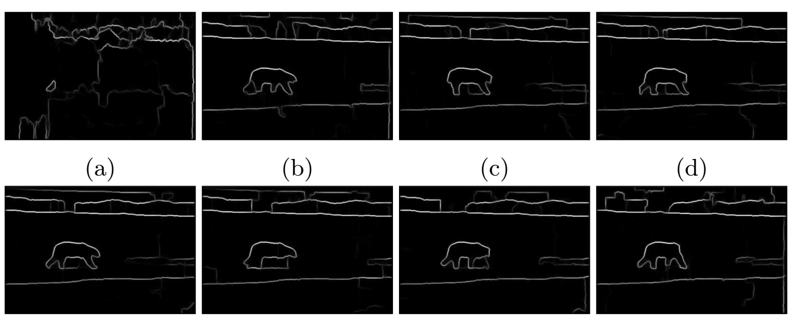
Fig. 5: Superpixel images created for the eight gPb edge images.

4 Experiments and Results

Unlike the original method, which has only one superpixel image as the final result, we have eight superpixel images for each input image. Therefore, the amount of edge information added to the method is much higher than the original method. Thus, to evaluate the performance of this addition together with the use of mean shift, we evaluated the result with a benchmark on edge detection problem. We also did a qualitative analysis for saliency detection application. Both experiments were performed on the 200 test images of the BSDS500 database [2]; saliency is also analyzed in the Extended Complex Scene Saliency Dataset (ECSSD) [24].

4.1 Edge Detection Experiments

The first step is to generate the intermediate edge images. For each superpixel image, an edge image is generated by Structured Forest Edge Detection (SFE), as shown in Fig. 6. Just as in [22], we have also used SFE due to its low running time. These intermediate edge images are grouped into a single image, as shown in Fig. 7, and submitted to the benchmark. This final image was created by a sequence of sum and normalization of the intermediate edge images. Following the methodology of [2], they were evaluated for a fixed threshold for all images (the Optimal Dataset Scale - ODS), the best threshold for each image (the Optimal Image Scale - OIS), and the average precision (AP) for 30 thresholds.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 6: Eight intermediate edge images generated by SFE for each edge orientation of gPb.

We have also investigated the number of superpixels defined by MS on our proposal and observed that, for 10 images (from the 200 images dataset), Mean Shift returned the value of 1; this happened in images with few different tones. Thus, we conducted two experiments to verify the impact of this issue. In the first, the results of the algorithm without any interference on the number of

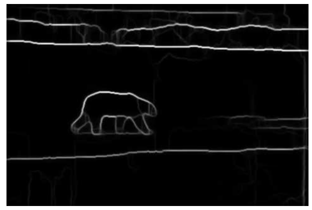


Fig. 7: Image resulting from the combination of the intermediate edge images.

superpixels defined by MS (called SHm) were analyzed. In the second, we defined a minimum value of five superpixels (called SHm5) - it is important to remember that this should be a low value due to the small number of different tones in the images. Fig. 8 illustrates this problem with two sample images and the results for the fixed minimum number of superpixels. The benchmark results for these experiments for the complete training dataset are provided in Table 1.

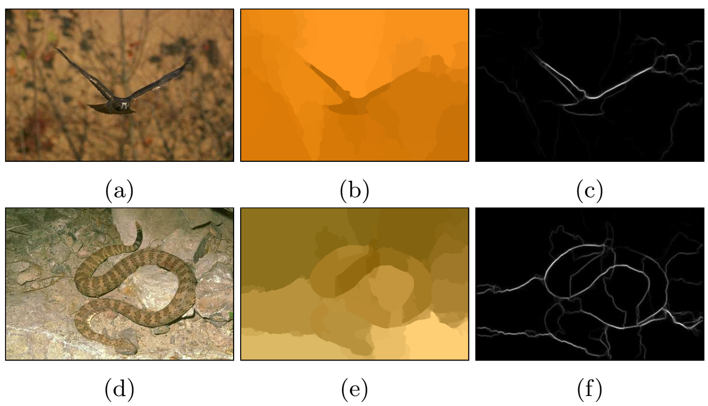


Fig. 8: Examples of images that MS had difficulty to identify the distinct number of colors: (a and d) original images, (b and e) their versions with number of superpixels fixed at 5 and (c and f) their edge images by SFE.

Observing the results on Table 1, we can notice that the performance of SFE, HED, and gPb algorithms are very close, as already shown in [22]. For the modified version of SH, we can see that the main difference is in the AP value to the version with the minimum number of superpixels fixed at five, achieving 0.63 whereas other methods obtained around 0.60. For ODS and OIS values, all methods provided very close results.

We have also made a comparison with the original version of SH, as shown in Table 2. In this table, it is shown the results for 100 to 500 superpixels as well as 1,000 and the total number of pixels in the images (called NP). This was done to compare the performance of our proposal with the version of the algorithm with

Table 1: Results of the proposed method compared to HED, SFE and gPb, considering the minimum number of superpixels.

	SFE	HED	GPB	SHm	SHm5
ODS	0.677	0.677	0.677	0.666	0.675
OIS	0.689	0.690	0.690	0.684	**0.694**
AP	0.605	0.605	0.605	0.615	**0.638**

no automatic definition of the number of superpixels. Because of the previous results, we are always using the fixed minimum number of 5 superpixels.

Table 2: Benchmark results from the original version of SH for different numbers of superpixel for the BSDS500 database.

	100	200	300	400	500	1000	NP
ODS	0.669	0.682	0.692	0.696	0.700	0.709	0.719
OIS	0.681	0.694	0.705	0.708	0.711	0.722	0.740
AP	0.584	0.610	0.626	0.634	0.636	0.643	0.624

Table 2 shows that SH achieves stable results around 500 superpixels. We can see that the difference for the values 500 and 1000 is not very high, despite the number of superpixels has doubled, especially when the threshold is the same for all images. About AP, SH improved the edge identification even when compared to the NP version, since the boundaries between the regions are redefined, leading to the removal of irrelevant edges. It is important to observe that superpixels make the images more simple; this can lead to a faster application of a segmentation algorithm with the image divided into a proper number of superpixels (as it was shown in Fig.2b). Comparing Tables 2 and 1, one can see that we have achieved results quantitatively very satisfactory automatically, without a brute force search for the best number of superpixels.

4.2 Saliency Detection

A saliency estimator aims to create a saliency map which is a gray level image, where dark tones mean not important areas, while light tones mean more important areas. For saliency detection, the eight superpixel images (as the ones in Fig. 5) are grouped into a single image in a similar way to what was done to combine the intermediate edge images. This grouping of the superpixel images is exemplified in Fig. 9. This image is then submitted to the *Minimum Barrier Salient Object Detection* (MBS+) [26], *Visual Saliency by Extended Quantum Cuts* (EQCUT) [4, 3], *Inner and Inter Label Propagation: Salient Object Detection in the Wild* (LPS) [11, 12] and *Saliency Optimization from Robust Background Detection* (RBD) [29] algorithms. Fig. 10 and Fig. 11 present the saliency maps applied to the original SH with 100, 300 and 500 predefined superpixels, the original image and SH after our proposal for the MBS+ algorthm.

In Figure 10e, we can see that areas around the owl have less light gray tones, when compared to the other results in the same figure. In this sense, the owl is more salient than the other elements of the scene as would be expected. The differences between the results in Fig. 11 is even more clear. Although it is a very challenging image, the two runners are better detected after our proposal (Fig. 11e), especially in relation to the amount of detected background in the other results.

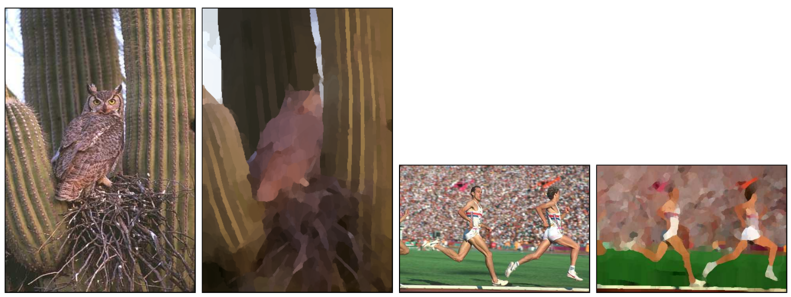


Fig. 9: (left column) Original images and (right column) the results after grouping the eight superpixel images.

(a) (b) (c) (d) (e)

Fig. 10: Saliency map generated by MBS+ for the superpixel images from SH with (a) 100, (b) 300 and (c) 500 superpixels; then (d) the map for the original image (number of superpixels equals to the number of pixels of the original image) and (e) our proposal. The original image is shown in Fig. 9 top-left.

We also conducted quantitative experiments on the Extended Complex Scene Saliency Dataset (ECSSD) [24] database (with 1,000 images) for the weighted-F_β [14] and Precision-Recall (PR) Curve metrics. As we can see in Table 3 and Fig. 12, both metrics show that our proposal achieved similar results for the

(a) (b) (c) (d) (e)

Fig. 11: Another example of saliency map generated by MBS+ for the superpixel images from SH with (a) 100, (b) 300 and (c) 500 superpixels; (d) the map for the original image (number of superpixels equals to the number of pixels of the original image) and (e) our proposal. The original image is shown in Fig. 9 bottom-left.

problem of saliency detection, improving the results for LPS, EQCUT and RBD algorithms. Similar to the BSDS500, the Mean Shift clustering algorithm also had problems in defining the number of colors for a small set of images.

Table 3: Weighted-F_β results for problem of saliency detection in the ECSSD database, analyzing the Sh algorithm with 100, 300 and 500 superpixels, the original image and after our proposal with minimum number of superpixels.

	100	300	500	NP	SHm5
MBS+	0.564	0.566	0.563	0.561	0.561
LPS	0.457	0.461	0.460	0.456	0.473
EQCUT	0.495	0.496	0.496	0.492	0.523
RBD	0.534	0.517	0.517	0.513	0.547

5 Conclusions and future work

Superpixel segmentation techniques are widely used in several areas. Generally, their result is a superpixel image containing regions regarding the number of superpixels previously defined. This is the proposal of Superpixel Hierarchy algorithm [22]. In our proposal, two modifications to SH are presented. The first is the automatic definition of the number of superpixels for an image based on the number of similar colors; the second is the addition of edge information for different orientations to improve the results.

To evaluate the performance of our modifications, the superpixel images were submitted to edge and saliency detection problems. The method achieved very satisfactory results overall and, in some cases, improvements for saliency maps. Our proposal presented good results in a completely automatic way, without the need to set manually the number of superpixels. The drawback is its running time, since no training is conducted.

For future work, other ways to automate the choice of superpixel numbers will be evaluated. The MS has a high execution time, so analyzing other cost-

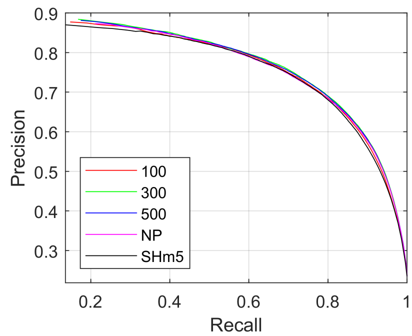


Fig. 12: Precision-recall curves of the tested methods.

effective proposals will also be considered. As the proposed method results in eight superpixel images for each input image, performing traditional area evaluation is not an easy task. Therefore, we will also perform an analysis in ways to group the images into a single image.

References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: Slic superpixels compared to state-of-the-art superpixel methods. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(11), 2274–2282 (2012)
2. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **33**(5), 898–916 (May 2011). https://doi.org/10.1109/TPAMI.2010.161, http://dx.doi.org/10.1109/TPAMI.2010.161
3. Aytekin, C., Kiranyaz, S., Gabbouj, M.: Automatic object segmentation by quantum cuts. In: 2014 22nd International Conference on Pattern Recognition. pp. 112–117 (2014)
4. Aytekin, C., Ozan, E.C., Kiranyaz, S., Gabbouj, M.: Visual saliency by extended quantum cuts. In: 2015 IEEE International Conference on Image Processing (ICIP). pp. 1692–1696 (2015)
5. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation (2017)
6. Dollár, P., Zitnick, C.L.: Structured forests for fast edge detection. In: 2013 IEEE International Conference on Computer Vision. pp. 1841–1848 (2013)
7. Fang, B., Zhang, X., Ma, Y., Han, Y.: Dti images segmentation based on adaptive bandwidth mean shift algorithm. In: 2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP). pp. 248–251 (2016)
8. Fang, F., Wang, T., Zeng, T., Zhang, G.: A superpixel-based variational model for image colorization. IEEE Transactions on Visualization and Computer Graphics pp. 1–1 (2019)
9. Fukunaga, K., Hostetler, L.: The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Transactions on Information Theory **21**(1), 32–40 (1975)

10. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016), http://www.deeplearningbook.org
11. Li, H., Lu, H., Lin, Z., Shen, X., Price, B.: Inner and inter label propagation: Salient object detection in the wild. IEEE Transactions on Image Processing **24**(10), 3176–3186 (2015)
12. Li, H., Lu, H., Lin, Z., Shen, X., Price, B.: Inner and inter label propagation: Salient object detection in the wild (2015)
13. Maire, M., Arbelaez, P., Fowlkes, C., Malik, J.: Using contours to detect and localize junctions in natural images. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8 (2008)
14. Margolin, R., Zelnik-Manor, L., Tal, A.: How to evaluate foreground maps. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2014)
15. Martin, D.R., Fowlkes, C.C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(5), 530–549 (2004)
16. Nameirakpam, D., Singh, K., Chanu, Y.: Image segmentation using k -means clustering algorithm and subtractive clustering algorithm. Procedia Computer Science **54**, 764–771 (12 2015). https://doi.org/10.1016/j.procs.2015.06.090
17. Nock, R., Nielsen, F.: Statistical region merging. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(11), 1452–1458 (2004)
18. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics **9**(1), 62–66 (1979)
19. Reinhard, E., Khan, E.A., Akyz, A.O., Johnson, G.M.: Color Imaging: Fundamentals and Applications. A. K. Peters, Ltd., USA (2008)
20. Ren, X., Malik, J.: Learning a classification model for segmentation. In: Proceedings Ninth IEEE International Conference on Computer Vision. pp. 10–17 vol.1 (2003)
21. Wang, Y., Ding, W., Zhang, B., Li, H., Liu, S.: Superpixel labeling priors and mrf for aerial video segmentation. IEEE Transactions on Circuits and Systems for Video Technology pp. 1–1 (2019)
22. Wei, X., Yang, Q., Gong, Y., Ahuja, N., Yang, M.: Superpixel hierarchy. IEEE Transactions on Image Processing **27**(10), 4838–4849 (2018)
23. Xie, S., Tu, Z.: Holistically-nested edge detection (2015)
24. Yan, Q., Xu, L., Shi, J., Jia, J.: Hierarchical saliency detection. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1155–1162 (2013)
25. Yang, H., Huang, C., Wang, F., Song, K., Yin, Z.: Robust semantic template matching using a superpixel region binary descriptor. IEEE Transactions on Image Processing **28**(6), 3061–3074 (2019)
26. Zhang, J., Sclaroff, S., Lin, Z., Shen, X., Price, B., Mech, R.: Minimum barrier salient object detection at 80 fps. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 1404–1412 (2015)
27. Zhang, Y., Li, X., Gao, X., Zhang, C.: A simple algorithm of superpixel segmentation with boundary constraint. IEEE Transactions on Circuits and Systems for Video Technology **27**(7), 1502–1514 (2017)
28. Zheng, Y., Zheng, P.: Hand tracking based on adaptive kernel bandwidth mean shift. In: 2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference. pp. 548–552 (2016)
29. Zhu, W., Liang, S., Wei, Y., Sun, J.: Saliency optimization from robust background detection. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 2814–2821 (2014)