Edge Adaptive Seeding for Superpixel Segmentation

Christian Wilms and Simone Frintrop

University of Hamburg

Abstract. Finding a suitable seeding resolution when using superpixel segmentation methods is usually challenging. Different parts of the image contain different levels of clutter, resulting in an either too dense or too coarse segmentation. Since both possible solutions cause problems with respect to subsequent processing, we propose an edge adaptive seeding for superpixel segmentation methods, generating more seeds in areas with more edges and vise versa. This follows the assumption that edges distinguish objects and thus are a good indicator of the level of clutter in an image region. We show in our evaluation on five datasets by using three popular superpixel segmentation methods that using edge adaptive seeding leads to improved results compared to other priors as well as to uniform seeding.

1 Introduction

Superpixels, defined by [19] as local and coherent sets of pixels capturing the relevant structures of an image, have become a popular pre-processing stage for different computer vision tasks over the last 15 years, like tracking [28, 31], recognition tasks [4, 27], or object proposal detection [15, 18, 25]. Integrating a number of neighboring pixels into one superpixel does not only decrease the number of basic entities of an image, but also the shape of these entities becomes arbitrary and can better fit to the image content as it is not defined by the layout of the imaging sensor. The decreased number of basic entities allows for more complex processing of the superpixels and possibly faster execution time.

An important aspect when using superpixels is the property that superpixels capture the relevant structure of an image. Missing boundaries of objects can lead to drastic degradation of the results of subsequent methods: objects that are segmented into the same superpixel can not be distinguished at later stages. Therefore, a good adherence to boundaries as well as minimizing the "leakage" of superpixels across boundaries is crucial for the success of the entire system. The size of the relevant structures varies not only between images but also within images. One area of an image might only feature few or no objects at all such as walls or skies, while other areas of the same image are highly cluttered with shelves or a crowd. The major difference between non-cluttered and cluttered regions is the number of edges as they separate objects from each other.

Most state-of-the-art superpixel segmentation algorithms [1, 5, 13, 32] are initially based on a uniform grid of seeds. A common problem with this approach is that all image regions are treated with the identical seed resolution while the level of clutter between image regions might vary substantially. One solution is to highly oversegment the image by generating a large amount of superpixels, which reduces the speed-up effect described above. The second solution does the opposite by only generating a relatively small number of superpixels that will reduce the complexity of subsequent analysis at the expense of missing small objects that are not segmented as individual superpixels. A third option is to use multiple superpixel resolutions [33], which again reduces the speed-up effect.

In this paper, we propose an edge adaptive seeding method to overcome this problem. The seeding can be combined with different segmentation algorithms. In contrast to [10, 12], which change the seeding of 3D point cloud segmentations based on saliency or colorfulness, we adapt the number of seeds in different image regions by measuring the edge density in an area around each pixel. We are to the best of our knowledge the first to apply adaptive seeding purely on 2D image data. As outlined in Fig. 1, a state-of-the-art edge detector [6, 7] is used to generate edge responses. The edge responses are smoothed and clustered using k-means clustering to generate areas in the image with different levels of clutter. The seeding resolution for each cluster is then adapted to the average edge density, leading to more dense segmentation in more cluttered areas and vise versa. Our results show that our edge adaptive seeding mechanism improves segmentation results in terms of boundary recall and undersegmentation error using multiple segmentation methods. Furthermore, we also released the source code of our implementation[1].

The rest of the paper is organized as follows. Section 2 discusses related work in segmentation and adaptive seeding. In Section 3, we describe the edge adaptive seeding approach. An evaluation of our method on five datasets is presented in Section 4. The paper closes with concluding remarks in Section 5.

2 Related Work

Since the definition of superpixels by [19], a variety of superpixel segmentation methods have been proposed [1, 5, 8, 13, 21, 32]. A state-of-the-art overview can be found in [24]. One of the most popular and successful methods in recent years [24] is SLIC [1]. The main principle of SLIC is to start from a regular grid of seeds and cluster the pixels in a combined LABXY-space that allows for a weighting of the influence of spatial and color difference. Second, the clustering is only done in a local region around each seed, significantly improving the runtime and restricting the area of each superpixel. Another popular method is SEEDS [5], which uses an energy function to generate homogeneous superpixels with regular boundaries. Other methods starting from a regular grid include [13, 32]. Despite the success of these methods on various datasets [24] and in numerous applications [4, 18, 27, 28, 31], none of them is able to adapt the number of seeds to the image content and segment different parts of the image with different resolutions. Other methods that are able to segment some parts of an image

[1] https://www.inf.uni-hamburg.de/easpxs

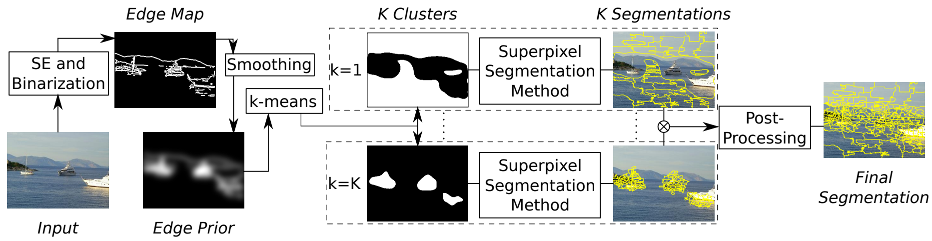

Fig. 1. Overview of the edge adaptive seeding mechanism. First, edges are generated using SE [6, 7] and the result is binarized. After smoothing, the result is used as a prior (edge density) to cluster the image into different parts. Each cluster is segmented with a segmentation method using a different number of superpixels depending on the average edge density. Finally, the results are combined and post-processed to form the final segmentation.

more densely than other parts like [8] and [26] produce inferior results compared to the above mentioned methods [24]. However, more recent approaches like SMURFS [14], based on iterative splitting and merging of superpixels, are able to keep up with the uniform seeding methods.

Three different adaptive seeding approaches exist in the literature all using 3D data. The first approach to apply adaptive seeding is DASP [29]. As a prior they use depth information and seed areas more densely if they are further away from the camera. This follows the principle that objects further away from the camera appear smaller on the image plane. [12] propose an approach that uses color information to adapt the seeding of a supervoxel clustering. Despite improving on the uniform seeding, this approach is generally not well-suited as neighboring objects do not necessarily have different colors.

Another approach to adapt the number of supervoxels to the image content is presented in [10]. As a prior for adapting the seeding resolution, a saliency map is used. The saliency system [9] highlights parts of the image that attract human visual attention and thereby give a prior for more dense segmentation. In contrast to [10], our method works on the 2D image plane and uses edge detection results as a prior. This seems more favorable since saliency highlights things in the image that stand out, which does not necessarily imply they should be segmented more densely. For example, in a scene with five blue balls and one red ball on the grass, the red ball is more salient. A more dense segmentation of the red ball in contrast to the blue balls however, is not useful. The positive effect of the edge based seeding with respect to the quality of the superpixels can be seen in our results.

3 Edge Adaptive Seeding

This section describes the edge adaptive seeding mechanism for superpixel segmentation methods (overview in Fig. 1). Given a desired number of superpixels and a number of clusters, it generates a prior about cluttered parts in the scene.

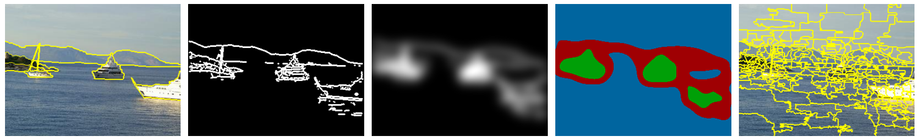

Fig. 2. Intermediate steps of the proposed method. From left to right: input image with ground truth, binarized edges (dilated for better visualization), smoothed prior, clustering for $K = 3$, and final segmentation.

This prior is used in a variation of the adaptive seeding approach of [10] to segment cluttered parts of the image more densely. As a superpixel segmentation method, any method that generates a predetermined number of superpixels can be used. First, the Structured Edges detector (SE) [6, 7] is applied to the input image, generating sharp edges with different strengths. To determine regions of different edge densities, binarization as well as smoothing is applied and the result is clustered using k-means. The seeding resolution of each cluster is determined using the average edge density within the cluster and a desired number of output superpixels. Thereby, a dense segmentation of regions with a high edge density is guaranteed, while regions with a low edge density are segmented more coarsely. After segmenting the image with a segmentation algorithm given the respective seeding resolutions of the clusters, the results of the different segmentations are cut and combined. Finally, we apply post-processing to eliminate disconnected and too small superpixels. The following subsections describe the steps of the approach in detail.

3.1 Edge Detection

To measure the edge density for each pixel in a given input image, we choose the detector SE [6, 7]. SE gives competitive results on relevant benchmarks like BSD [2] as well as NYUV2 [22] and shows good performance in many applications [3, 20, 34]. According to [6, 7], the detector can run at a rate of up to 30 Hz, while still achieving state-of-the-art results. SE detects edges in an image based on a random forest classifier, which makes it easily adaptable to different domains.

The result of SE is a sharp mask of edges with varying strengths. To transform this result into a prior representing the edge density, we first binarize the results given a lower bound τ for the strength of an edge to be detected. This binarization is necessary to become independent of the edge strength, which improved the results in our experiments and follows the assumption that stronger edges are found more easily anyway. Therefore, an overly dense segmentation is not necessary. The binarization is followed by a smoothing step with a Gaussian kernel to determine the edge density as the weighted average number of edge responses above τ within a certain area around each pixel. This leads to a prior that represents the edge density and highlights regions with many detected edges. The two steps are visualized in the second and third image of Fig. 2.

3.2 Adaptive Seeding

Given the prior from the processed edge detection result, a k-means clustering similar to [10] is done on the prior resulting in K clusters. The clusters are sorted in ascending order with respect to their average edge density. Each pixel of the image is assigned to one of the K clusters according to the edge density value of the prior. As we request superpixel algorithms used in this approach to generate a desired number of superpixels, we are in contrast to [10] able to determine the seeding resolution of each cluster.

For each of the K clusters, we apply a segmentation with a seeding resolution adapted to the average edge density e_k with $k = 1, \ldots, K$. Based on the assumptions of Section 1, a higher average edge density leads to a more dense seeding resolution and more superpixels. The exact number of superpixels of the k-th cluster n_k is defined based on the average edge density of a cluster, the desired number of superpixels in the final segmentation n, the number of superpixels of either the minimal or maximal resolution n_1 and n_K and a weight w_k determining the number of superpixels of the k-th cluster n_k in relation to n_1 and n_K. First, the number n of superpixels of the final segmentation generated by K different resolutions on K distinct clusters of the image is defined as

$$n = \sum_{k=1}^{K} a_k (n_1 + (n_K - n_1) \frac{w_k}{w_K}), \tag{1}$$

with a_k being the relative size of the k-th cluster. The weight factor w_k, which is normalized to the maximum of the weights, determines the number of superpixels relative to n_1 and n_K. w_k is chosen exponentially based on e_k as well as the minimum and maximum edge density e_1 and e_K. This weighting leads to a number of superpixels that is adaptive to the edge density in the clusters. The weight w_k is therefore defined as

$$w_k = \begin{cases} 1 - b^{\frac{e_k - e_1}{e_K - e_1}} & \text{if } b < 1 \\ b^{\frac{e_k - e_1}{e_K - e_1}} - 1 & \text{else} \end{cases} \tag{2}$$

with b being a parameter to adapt the weighting. As all variables of equation (1) are known with the exception n_1 and n_K, one of those has to be fixed as a parameter. While fixing n_K could lead to a negative number of superpixels for n_1, given an unfavorable choice of n, fixing n_1 and then determining n_K and with that all intermediate numbers of superpixels n_k is easily possible. Therefore, equation (1) can be transformed into

$$n_K = \frac{n_1 \sum_{k=1}^{K} a_k \left(1 - \frac{w_k}{w_K}\right) - n}{- \sum_{k=1}^{K} a_k \frac{w_k}{w_K}}, \tag{3}$$

giving the number of superpixels n_K for the densest cluster. The intermediate number of superpixels n_k for the respective clusters can now be calculated as

$$n_k = n_1 + (n_K - n_1) \frac{w_k}{w_K}. \tag{4}$$

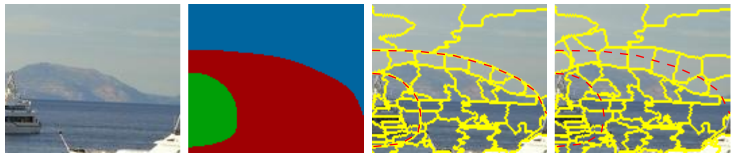

Fig. 3. Comparison of clipping the segmentations. To avoid artificial edges at cluster borders as in image 3 (red lines), coarser segmentations are overlayed with finer ones (image 4). Left: original image (part of Fig.1 and Fig.2), second image: clustering.

3.3 Superpixel Segmentation

After calculating the number of superpixels for all clusters, a superpixel segmentation algorithm that is able to generate a predetermined number of superpixels, can be applied to the whole image K times with the different number of superpixels n_k. Exemplary we use SLIC, SEEDS and SMURFS in Section 4. The parallelized application of the algorithm only on the relevant parts of the image for speed-up is also possible, if the algorithm supports masks or can be adapted in such way. This step results in K segmentations or partial segmentations of the image that need to be combined for the final segmentation.

While directly clipping the clusters in the respective segmentations and combining them would lead to a correct oversegmentation, it would introduce the continuous, artificial edges of the clusters into the segmentation result as marked red in the third image of Fig. 3 and visible in the results of [10]. This can be a drawback, e.g., if the continuation of edges is a cue for later merging steps. Therefore, we propose to sequentially combine the different segmentations starting with the coarsest. From the next finest segmentation, all superpixels that contain pixels from the respective cluster will be selected. The part of the image covered by those superpixels will be replaced in the combined result with this new, finer segmentation. This procedure continues until all segmentations are processed. As visible in the last image in Fig. 3, an edge adaptive segmentation without the artificial, continuous edges of the clusters is generated.

3.4 Post-processing

One problem introduced by the previously described combination technique is that superpixels might be cut into multiple components by overlaying a finer segmentation. Furthermore, due to the imperfect estimation of the number of superpixels a method produces, the overall number of superpixels might not fit the desired number. Therefore, we propose a two-step post-processing with first relabeling all unconnected superpixels and second merging small superpixels into one of their neighbors, similar to the post-processing done in [1]. To prevent merging only in the finest superpixel resolution, the size of a superpixel is normalized by the seeding resolution of the cluster the superpixel results from. This merging is done until the desired number of superpixels n is reached, resulting in a final edge adaptive segmentation as shown in the last image in Fig. 2.

4 Results and Evaluation

The evaluation is done based on the superpixel evaluation framework presented in [24]. Therefore, the datasets used are BSD [2], NYUV2 [22], SUNRGBD [23], SBD [11] and Fashionista [30]. They cover a wide variety of images containing different indoor and outdoor scenes with various levels of clutter. The number of images per dataset and splits are the same as in [24] resulting in around 200 images for training and 400 images for testing per dataset. Image sizes vary between 658×486 and 316×240 depending on the dataset. The metrics used for comparison are standard boundary recall (REC) [16] and undersegmentation error (UE) defined by [17] and recommended in [24]. For parameter optimization the same combination as in [24] is used ($\epsilon = (1 - \mathrm{REC}) + \mathrm{UE}$).

As superpixel segmentation methods we chose the widely used SLIC [1] and SEEDS [5] as both approaches satisfy our requirement of generating a specified number of superpixels. To compare our approach with an adaptive superpixel segmentation method, we also chose SMURFS [14]. We compare our method to SLIC, SEEDS and SMURFS with uniform seeding as well as with the saliency based seeding method of [10] adapted to the 2D-domain by using the saliency map instead of the edge prior. Furthermore, on the datasets NYUV2 and SUNRGBD a comparison to DASP is made as depth data is available. A comparison with [12] is not possible as their changes in seeding rely on the specifics of the used supervoxel segmentation method. To show that our general approach is beneficial, we also present the results of our system using the ground truth (GT) edges instead of the SE results as a perfect prior. This prior leads to results that are independent of the edge detection quality and therefore define an upper bound for our approach.

To make the results comparable, the parameters were optimized on the training sets for each dataset and each superpixel resolution independently as outlined in [24]. The parameters optimized were the parameters of the segmentations, n_1 and K in Eq. (1), b in Eq. (2) as well as τ and σ from pre-processing the edge detection result. n_1 was optimized in the range of $\frac{1}{10}n, \ldots, \frac{6}{10}n$, K in the range of $3, \ldots, 6$, b in the range of $0.75, 2, 5, 10$, τ in the range of $0.05, \ldots, 0.25$, and σ in the range of $5, \ldots, 30$. The models for the SE detector were also learned on the training datasets.

The results using SMURFS, described in the supplementary material, indicate that our method can also improve adaptive segmentations, given edge detection results that are superior to current state-of-the-art results. For DASP, the lower performances on NYUV2 and SUNRGBD, described in the supplementary material, confirm the findings of [24] that depth does not always lead to improved results.

4.1 Results using SLIC

In our first experimental set-up, we use SLIC as the segmentation method. Qualitative results are shown in Fig. 4. The results in row 1 and 2 clearly show the coarse segmentation in areas of the image covered with sky or sea, while areas

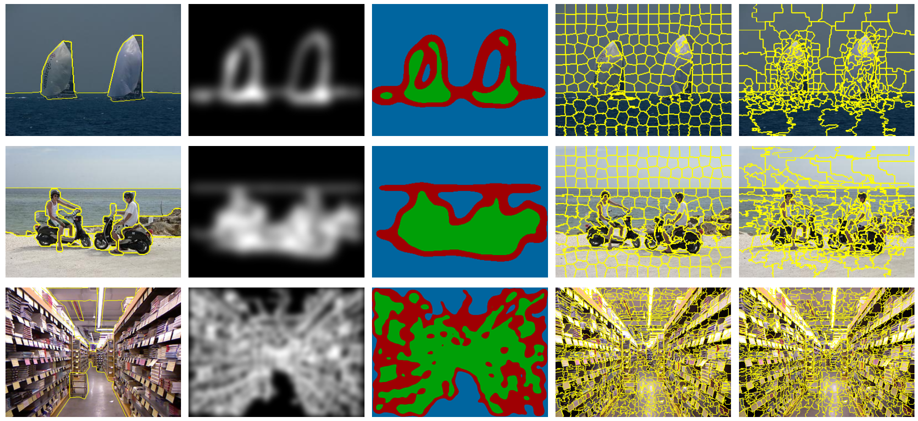

Fig. 4. Qualitative results of the edge adaptive seeding with SLIC on images from SBD (1st & 2nd row) and NYUV2 (3rd row). From left to right: input image with ground truth, edge density, clusters, uniform SLIC segmentation, result of proposed seeding with SLIC. For qualitative results using SEEDS and SMURFS see the supplementary material.

around objects are segmented more densely. The results of our experiments in terms of REC and UE over the test sets of BSD and SBD as a function of the number of superpixels are shown in Fig. 5. The results on the other datasets can be found in the supplementary material. It is clearly visible that our approach of using edge detection results is beneficial for the segmentation result. Our approach consistently outperforms SLIC in REC over both datasets and the rest of the datasets, as the results in the supplementary material show. For instance, on SBD with 250 superpixels the edge adaptive seeding improves the REC form around 77% to almost 82%. Therefore, the edge prior leads to a less densely segmented image in parts like backgrounds and a more densely segmented image in interesting parts with many objects. This is useful, as given the same number of superpixels the more complex parts can be analyzed in much more detail.

These findings can be confirmed if the GT edges of the images are used instead of the SE results. Those perfect priors improve the results even more, leading to 86% in REC on SBD using 250 superpixels (82% with SE based prior). The large improvement when using the GT prior compared to the SE prior on some images can be explained by many different entities of the same class in an image, like books in the library that generate many edge results. As the distinction between the individual entities is not always made in the GT data, the ability to segment individual entities much better is not reflected in positive performance (cf. Fig. 4, 3rd row). Similar effects arise on high textured images.

The relatively constant performance in terms of UE when using the edge detection prior across all datasets can be explained by GT segments in areas that are classified as background given the prior. In that case, the superpixels covering those background areas are much larger than before, leading to a large

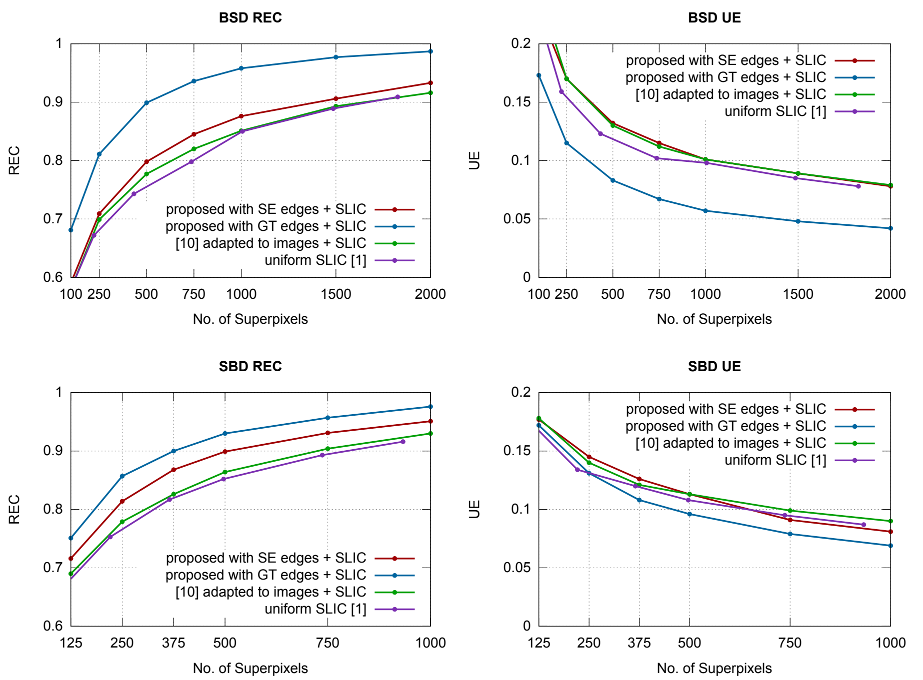

Fig. 5. Boundary recall (REC) and undersegmentation error (UE) on the BSD (top) and SBD (bottom) datasets using SLIC. For results on the other datasets see the supplementary material.

UE in those areas and neglecting the improvements in the finer segmented areas. This can also be validated when using the GT edges as input to the system. With those GT edges such cases are not possible, thus only the advantage of the finer segmentation of relevant parts remains.

We outperform [10] across all datasets and resolutions in REC and are on a similar level in terms of UE. The explanation of the lower performance of their approach is the different kind of data, as their approach was developed for supervoxel clustering of RGBD data. Using supervoxels and point clouds results in a different general seeding strategy, as the supervoxels can be located anywhere in space in contrast to the pixels of an image, that are fixed to a grid. As the clustering based on the saliency prior usually leads to thin components around objects, the seed resolution on the image plane can be too coarse for those thin components, which leads to components without a seed. These kinds of seeding artifacts are not possible with the supervoxel clustering used in [10].

4.2 Results using SEEDS

To show the generality of our approach, we set up a second experiment changing the segmentation method from SLIC to SEEDS. Fig. 6 shows exemplary results

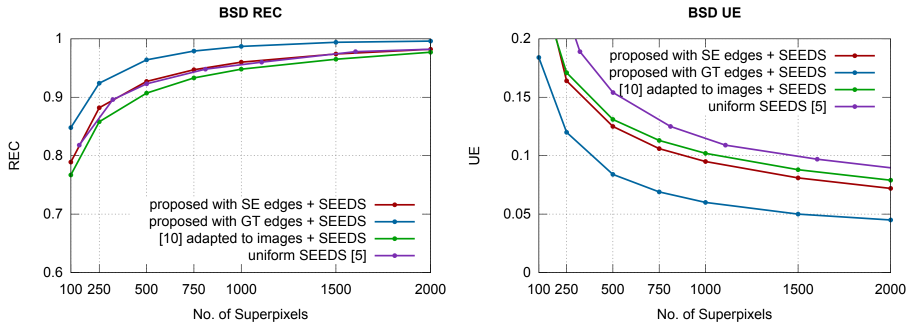

Fig. 6. Boundary recall (REC) and undersegmentation error (UE) on the BSD dataset using SEEDS. As for the results using SLIC, the usage of the edge based prior leads to improved results, here especially in terms of UE. For results on the other datasets see the supplementary material.

of our method in combination with SEEDS in terms of REC and UE on the BSD dataset. Despite minor improvement in REC using the SE based edge prior, the major improvement in UE still leads to an overall advantage using the proposed edge based seeding. Using the GT based prior again leads to even better results. Results on the other four datasets can be found in the supplementary material.

The difference in results between SLIC and SEEDS is mainly due to properties of the segmentations. SEEDS generates less equally sized superpixels that adapt better to the level of clutter. However, as some superpixels are larger, the edges missed by SEEDS lead to more leakage than for SLIC, where the size of the superpixels is more evenly distributed. Using the edge adaptive seeding balances this size variation by identifying more edges and forcing a more dense segmentation around them. This is supported by the results using the GT edges.

5 Conclusion

Finding one superpixel resolution that fits all the different parts of an image is impossible. However, segmenting as many objects correctly with as few superpixels as possible in images with different levels of clutter is important, since wrongly segmented objects or heavily oversegmented scenes hamper subsequent processing steps.

To tackle that problem, we have proposed an approach to adapt the seeding for superpixel segmentations based on the edge density. Edges are a good indicator for the level of clutter, as objects can be discriminated by an edge. Therefore, we segment parts of an image with many edges more densely and vise versa.

Our results show the improved segmentations using the edge adaptive seeding for different superpixel segmentation methods across five datasets in comparison to other adaptive seedings as well as the uniform seeding. In the future, we plan to use the edge detection result to further improve superpixel segmentations.

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 TPAMI 34(11), 2274–2282 (2012)
2. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE TPAMI 33(5), 898–916 (2011)
3. Arbelaez, P., Pont-Tuset, J., Barron, J.T., Marques, F., Malik, J.: Multiscale combinatorial grouping. In: CVPR (2014)
4. Baraldi, L., Paci, F., Serra, G., Benini, L., Cucchiara, R.: Gesture recognition in ego-centric videos using dense trajectories and hand segmentation. In: CVPR Workshops (2014)
5. Van den Bergh, M., Boix, X., Roig, G., de Capitani, B., Van Gool, L.: SEEDS: Superpixels extracted via energy-driven sampling. In: ECCV (2012)
6. Dollár, P., Zitnick, C.L.: Structured forests for fast edge detection. In: ICCV (2013)
7. Dollár, P., Zitnick, C.L.: Fast edge detection using structured forests. IEEE TPAMI 37(8), 1558–1570 (2015)
8. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. IJCV 59(2), 167–181 (2004)
9. Frintrop, S., Werner, T., Martín García, G.: Traditional saliency reloaded: A good old model in new shape. In: CVPR (2015)
10. Gao, G., Lauri, M., Zhang, J., Frintrop, S.: Saliency-guided adaptive seeding for supervoxel segmentation. In: accepted for IROS (2017)
11. Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions. In: ICCV (2009)
12. Kanezaki, A., Harada, T.: 3D selective search for obtaining object candidates. In: IROS (2015)
13. Levinshtein, A., Stere, A., Kutulakos, K., Fleet, D., Dickinson, S., Siddiqi, K.: Turbopixels: Fast superpixels using geometric flows. IEEE TPAMI 31(12), 2290–2297 (2009)
14. Luengo, I., Basham, M., French, A.: Smurfs: superpixels from multi-scale refinement of super-regions. In: BMVC (2016)
15. Manen, S., Guillaumin, M., Van Gool, L.: Prime object proposals with randomized prim's algorithm. In: ICCV (2013)
16. Martin, D., Fowlkes, C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE TPAMI 26(5), 530–549 (2004)
17. Neubert, P., Protzel, P.: Superpixel benchmark and comparison. In: Proc. of Forum Bildverarbeitung (2012)
18. Rantalankila, P., Kannala, J., Rahtu, E.: Generating object segmentation proposals using global and local search. In: CVPR (2014)
19. Ren, X., Malik, J.: Learning a classification model for segmentation. In: ICCV (2003)
20. Revaud, J., Weinzaepfel, P., Harchaoui, Z., Schmid, C.: Epicflow: Edge-preserving interpolation of correspondences for optical flow. In: CVPR (2015)
21. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE TPAMI 22(8), 888–905 (2000)
22. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: ECCV (2012)
23. Song, S., Lichtenberg, S.P., Xiao, J.: SUN RGB-D: A RGB-D scene understanding benchmark suite. In: CVPR (2015)

24. Stutz, D., Hermans, A., Leibe, B.: Superpixels: An evaluation of the state-of-the-art. CVIU (2017), in press
25. Uijlings, J.R.R., van de Sande, K.E.A., Gevers, T., Smeulders, A.W.M.: Selective search for object recognition. IJCV 104(2), 154–171 (2013)
26. Vincent, L., Soille, P.: Watersheds in digital spaces: An efficient algorithm based on immersion simulations. IEEE TPAMI 13(6), 583–598 (1991)
27. Wang, C., Liu, Z., Chan, S.C.: Superpixel-based hand gesture recognition with kinect depth camera. IEEE Transactions on Multimedia 17(1), 29–39 (2015)
28. Wang, S., Lu, H., Yang, F., Yang, M.H.: Superpixel tracking. In: ICCV (2011)
29. Weikersdorfer, D., Gossow, D., Beetz, M.: Depth-adaptive superpixels. In: ICPR (2012)
30. Yamaguchi, K., Kiapour, M.H., Ortiz, L.E., Berg, T.L.: Parsing clothing in fashion photographs. In: CVPR (2012)
31. Yang, F., Lu, H., Yang, M.H.: Robust superpixel tracking. IEEE TIP 23(4), 1639–1651 (2014)
32. Yao, J., Boben, M., Fidler, S., Urtasun, R.: Real-time coarse-to-fine topologically preserving segmentation. In: CVPR (2015)
33. Zhu, L., Klein, D.A., Frintrop, S., Cao, Z., Cremers, A.B.: A multisize superpixel approach for salient object detection based on multivariate normal distribution estimation. IEEE TIP 23(12), 5094–5107 (2014)
34. Zitnick, C.L., Dollár, P.: Edge boxes: Locating object proposals from edges. In: ECCV (2014)