

”RegionCut” - Interactive Multi-Label Segmentation Utilizing Cellular Automaton

Oliver Jakob Arndt, Björn Scheuermann, Bodo Rosenhahn
Institut für Informationsverarbeitung (TNT), Leibniz Universität Hannover, Germany
{arndt,scheuermann,rosenhahn}@tnt.uni-hannover.de

Abstract

This paper addresses the problem of interactive image segmentation. We propose an extension of the GrowCut framework which follows Cellular Automaton theory and is comparable to a label propagation algorithm. Therefore, user labels are propagated according to Cellular Automaton until convergency. A common problem of GrowCut is the time consuming user initialization which requires distributed seeds. Our main contribution focuses on determining such an initialization utilizing GMMs and spherical coordinates. Furthermore we propose a new weight function based on the mean image gradient. According to our evaluation, our extensions result in a simplified user interaction and in better results in terms of accuracy and running time. Our experiments show that our method can compete with state-of-the-art energy minimization frameworks.

1. Introduction

Interactive image segmentation is a popular field of research in computer vision. The aim of an interactive segmentation framework is to extract one or more foreground objects from the background. Applications for interactive image segmentation include photo editing and medical image analysis. Examples for popular interactive frameworks are Graph Cuts [2, 4, 3], GrabCut [12], active contours and interactive level-sets [9, 5, 6, 15], TV-Seg [16, 14], efficient belief propagation [7, 19] or Intelligent Scissors [10].

GrabCut [12] was introduced by Rother et al. in 2004. It is based on the works of Boykov et al. [2] by iteratively performing Graph Cuts. The user provides a rectangle around the desired object as an initial segmentation. This segmentation is used to approximate the appearance of the foreground and background respectively by Gaussian Mixture Models (GMMs). The negative log-likelihoods of the GMMs are assigned to each pixel and Graph Cut is used to update the segmentation. These steps are repeated iteratively.



Figure 1. Segmentation results using the proposed (user interactive) segmentation framework and a lasso initialization. The images come from the Berkeley Segmentation Database.

Similarly, Scheuermann et al. [15] proposed an interactive segmentation scheme based on level-sets and Dempster’s theory of evidence. Here, the user provides several strokes marking object and background regions. The histograms of these regions are included in a continuous energy function using Dempster’s rule of combination. The defined energy function is minimized iteratively by solving the corresponding Euler-Lagrange equation.

In [19], Wang et al. proposed the efficient label propagation (ELP) for interactive image segmentation. In an interactive manner, user defined labels are automatically propagated over the remaining unlabeled pixels. ELP is related to the efficient belief propagation proposed by Felzenszwalb and Huttenlocher [7].

It has been shown in [12] and [15] that energy minimization methods based on Graph Cuts or level-sets outperform other methods in terms of segmentation quality and required user effort. However, these energy minimizing methods are hard to implement and hard to control.

In [17] V. Vezhnevets and V. Konouchine suggested an interactive segmentation scheme based on cellular automata (CA), the so called ”GrowCut” algorithm. S. Ulam and J. von Neumann proposed the CA 1966 in [18]. It can be implemented very easily and has been used in the computer vision community for image denoising [11] and edge detection [8]. Important properties of this method are the possibility to perform multi-label segmentation tasks and the ability that the algorithm can be extended to higher dimensions. However, compared to other methods, the results of

GrowCut are not competitive.

In this paper we propose several extensions to GrowCut which have specific advantages compared to the original algorithm. We will show that our extensions make the GrowCut algorithm competitive to *state-of-the-art* segmentation schemes.

The **contributions** we propose in the paper are: (i) a novel pre-initialization step based on user labeling; (ii) a new weight function including global image statistics; (iii) measures for automatic parameter settings; (iv) for further evaluation we will provide C-Sources to the scientific community.

The first improvement brings regional properties into the edge based GrowCut algorithm. Therefore, region statistics are modeled based on the user input and similar pixels are assigned with high probability to the same region. The second improvement makes use of the mean color noise to refine the costs for neighboring pixels, taking into account the overall pixel gradients. The last improvement proposes different measurements based on regional image statistics to automatically set the included parameters.

2. GrowCut Segmentation

The main idea of GrowCut is that user defined seed pixels iteratively try to occupy their neighbors. Therefore each pixel has a defined strength. Initially the strength is set to one for each seed pixel and zero for all other. So, the seed pixels expand over the image until the edges of two different labels contact each other. Henceforward pixels try to occupy their neighbors.

At each iteration the strength of an attacking pixel is multiplied with a linear weighting function $g_{lin} \rightarrow [0, 1]$ to reduce the attacking pixels strength. The color difference of attacking pixel p and attacked pixel q is used to define g_{lin} . The idea is to highly reduce the strength of an attacking pixel if the color difference is high. It is defined by

$$g_{lin}(I_p, I_q) = 1 - \frac{\|I_p - I_q\|}{c_{max}} \geq 0, \quad (1)$$

where $I_{\{p,q\}}$ is the color vector of pixels p and q , and c_{max} is the maximum color difference.

Let p be the attacking and q the attacked pixel, Θ_p the strengths and x the color gradient between p and q , then p occupies q if the decreased strength $\Theta_p \cdot g_{lin}(x)$ is higher than Θ_q . In this case the label l_q will be set to l_p and the strength Θ_q will be set to $\Theta_p \cdot g_{lin}(x)$. Iteratively each pixel in I tries to occupy its neighbors until convergence.

Since the given competition rule is multi-label capable, GrowCut naturally supports multi-label segmentations. GrowCut is also attractive as a user interactive segmentation method because of the possibility to directly influence the algorithm while iterating. A mindful user who notices a label area flooding across an edge is able to decrease the

flooding pixel's strength to allow the surrounding pixels to recapture this area.

Experiments showed that the algorithm is very sensitive to small changes in the user given seeds. Thus, the algorithm can be caught in local minima which are not ideal segmentations. For an adequate segmentation many user seeds, distributed over the entire image, are needed. That means the user has to invest much time in placing the seeds. For an interactive segmentation scheme it is not acceptable to pursue a trial-and-error-procedure for the initialization. This is the motivation for our pre-initialization which determines distributed seeds automatically only with a few given strokes.

3. Contributions

GrowCut, motivated on cellular automaton, is an intuitive method to segment an image. It is simple to implement and able to handle multiple labels. However, experiments showed that the algorithm never acts as efficiently as shown in [17]. One point is that typically given user initializations are strokes but never hundreds of pixel-like seeds distributed over the image. In our experiments the segmentations using GrowCut never achieved the quality of energy minimizing segmentation algorithms. Based on the idea behind GrowCut, we will now present several improvements that makes it competitive to state-of-the-art segmentation schemes like the often used Graph Cut approaches.

3.1. Pre-Initialization

The first improvement focuses on the problem of determining an adequate initialization with distributed seeds. Those seeds can be stated as the expected initialization of GrowCut. Based on the given user seeds we learn GMMs to approximate the appearance of each region and compute the probability for each pixel belonging to a particular region. The probability differences are used to decide which label is likely for a given pixel.

Our idea is, that for a given threshold ϑ we automatically pre-initialize the non-labeled pixels. If the probability difference $P_i - P_j$ for a label i is greater than the threshold ϑ for all other labels j , the pixels initial label l_p is i .

Since there is no guarantee for a correct decision, it is necessary to ensure that the pre-initialized pixels can be occupied by regions coming from user seeds. That means the strength Θ_i for a pre-initialized pixel has to be lower than the user-initialized pixels' strength. Since the user-initialized pixels' strength is set to 1, we set the pre-initialized pixels' strength to 0.9.

Due to the differences of color distribution from image to image, the label probabilities can fluctuate. Additionally, the fluctuation is influenced by the significance of the initial strokes set by the user which are used as training data for region models. To prevent the pre-initialized area to be

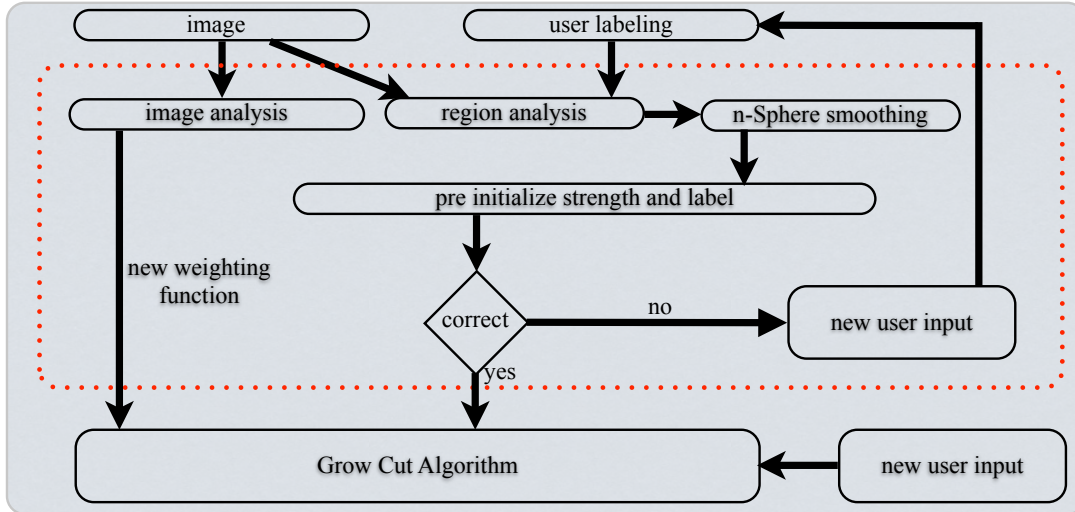


Figure 2. Work-flow of RegionCut. After initializing the algorithm with a user labeling, the regions are analyzed to compute the pre-initialization. The user can control the pre-initialization by adding new strokes or varying the parameter θ . If the pre-initialization is correct, the weighting function is computed and the iterative GrowCut algorithm starts.

influenced by the significance of user strokes we set up the following method.

We start the pre-initialization by setting the threshold ϑ to an initial value ϑ_0 in the interval $[0; 1]$. In images with clearly color divided regions the probabilities might be split into values around 0 and values around 1. This great gap between the probabilities causes the pre-initialization to fill nearly the whole image by choosing an initial value ϑ_0 which is too low. We choose ϑ_0 as 0.95. Now the threshold ϑ is decreased by ε for every iteration. A new iteration with a decreased threshold will be carried out while the initialized area A (in percent), of the area which was not user-initialized, is lower than a given value A_{max} . Experiments showed that a good choice is $A_{max} = 70\%$, meaning that 70% of the non-seed pixels are pre-initialized.

Compared with GrowCut much less time is needed to calculate the proposed pre-initialization. Consequently, the work-flow should stop after pre-initialization to let the user correct the strokes or the initialization area A_{max} . This method includes the possibility of controlling the level of pre-initializing automatically distributed seeds and expands the user convenience.

On the one hand the idea is to pre-initialize the area by region measures as much as possible. Thus, no iterating is needed for these pixels. On the other hand when the image is pre-segmented there has to be space between the regions for the GrowCut algorithm in order to find edges which cannot easily be found with color distribution analysis. Adding this method combines the advantages of region- and pixel-based segmentation algorithms.

3.2. Smoothing Seeds Using Spherical Coordinates

Due to automatically seeding the image, seeds may occur in wrong regions. To avoid this effect an erosion on the pre-initialized seeds could be performed, but this method also reduces the pre-initialized area. Instead of manipulating the seeds computed before, this section demonstrates how to smooth the probability data to improve the pre-initialization.

The first idea of smoothing the probabilities is a simple smoothing of each probability field. When smoothing the particular probability fields itself we ignore the dependency information between the labels probabilities. Another risk of this method might be a falsification of the tendency, when abstractly manipulating meaningless data. This may cause small regions of one label with high, but surrounded of pixels with low intensity, to be eliminated, irrespective of other labels probabilities, which might be equally insignificant. This situation can effect eliminating right detected small enclosed areas, useful when segmenting images with small structures, like leaves or hair. Instead, we propose a method that utilizes the dependency information effectively.

The probabilities assigned to a pixel describe the tendency of that pixel belonging to the different labels. This tendency can be easily imagined as a vector's direction, that has to be smoothed. Therefore, we look at a single pixel and interpret the collectivity of probabilities for each region as spanning an orthogonal n -dimensional space. Now the probabilities of each pixel are represented as an n -dimensional vector in cartesian coordinates, in which n is the number of used labels. Via this association we can do a transformation to n -dimensional spherical coordinates,

where the probability dependencies are directly mapped to the angles.

The transformation is given by:

$$\vartheta_k = \arctan \frac{\sqrt{\sum_{i=k+1}^n P_i^2}}{P_k}, \quad r = \sqrt{\sum_{i=1}^n P_i^2} \quad (2)$$

Processing a gaussian-smoothing on each dimension of the spherical coordinates respects the dependencies between the different probabilities.

To recover each pixel's probabilities, the smoothed vector field has to be retransformed, as shown in Equation 3. The enhanced probabilities are used for pre-initialization to compute distributed seeds.

$$P_k = r \cdot \nu(k) \cdot \prod_{i=1}^{k-1} \sin(\vartheta_i), \quad \text{with} \quad (3)$$

$$\nu(k) = \begin{cases} \cos(\vartheta_k) & , k < n \\ 1 & , k = n \end{cases}$$

The proposed smoothing, using spherical coordinates to represent the probabilities of each pixel, utilizes the dependency between the possible labels and significantly enhanced the proposed pre-initialization.

3.3. Weighting Function

Objects in an image rarely are homogeneous in color. According to the rising spatial distance to an initial seed, the pixels' strength decreases because color variations are summed up. This problem can be partly improved by using the proposed pre-initialization, because this method tries to distribute seeds all over the image which prevents long distances. Nevertheless, a long spatial distance between a non-seed pixel and a seed pixel can occur.

If there is a high gradient, the probability for this position of being an edge is high. Consequently, the attacking pixel should be weak in order not to break through the edge. The GrowCut algorithm uses a simple linear function for that, see Equation (1). In this section we present a weighting function that adapts to probability distributions in the image.

In order to avoid weakening pixels' strength in homogeneous regions due to long spatial distances but still let the region growing stop at high gradients, the mean image noise is used to influence the weighting function. Therefore we define:

$$g_{cos}(x, k) = \frac{1}{2} \cdot (1 + \cos(\pi \cdot x^k)) \quad (4)$$

This function keeps the pixels' weight for small color differences and decreases it for greater color differences by

using an adjusted cosine. The parameter k is used to slide the falling edge towards or away from the y -axis.

Analyzing the image data and extracting the mean gradient allows to calculate k automatically by setting a desired tolerance. The tolerance in this case denotes the weakening of the pixels' strengths while the regions and the distances to the seeds are growing. Setting a low tolerance at the mean gradient of the image will cause moving the g -functions' falling edge away from the y -axis. Thus, the strengths will decrease minimally by lower color differences. However, setting the tolerance too high will cause the regions breaking through high gradients. In this way the algorithm will not detect edges.

Let t be the tolerance. Based on the condition $g_{cos}(x_{mean}, k) \stackrel{!}{=} 1 - t$, we compute k for each image automatically. Thus, the weakening will be constant for every mean image gradient instead of the linear g -function whose weakening is not predictable. Experiments have shown that a good tolerance is 0.005.

To stretch the falling edge, we use a linear combination of $g_{lin}(x)$ and $g_{cos}(x, k)$ weighted with the parameter l as shown in equation 5. Due to our experiments we noticed that l does not depend on region measures. In all experiments the value l is fixed to 0.8.

$$g(x, l, k) = l \cdot g_{cos}(x, k) + (1 - l) \cdot g_{lin}(x) \quad (5)$$

3.4. RegionCut

In this section we will summarize the segmentation work-flow of GrowCut, which is shown in Figure 2. To stay focused on user convenience it is necessary to find a balance between user interactivity and automatic parameter calculation. In this case it is possible for the user to manipulate and correct the automatic calculated initialization.

The first point for the user to control the algorithm is when placing strokes. The second point to influence the work-flow can be integrated after the first pre-initialization. Here, the user can place additional strokes or change the area of pre-initialized pixels to refine the pre-initialization.

To extend the GrowCut algorithm we used different kinds of region analysis to pass an optimized initialization and information about the noise level within the image. From the user strokes we learn GMMs to approximate the appearance of each region. Based on these region models we assign the probabilities $P_{i,p}$ for each pixel p belonging to region i . These probabilities are smoothed using spherical coordinates (Section 3.2) and the smoothed probabilities are used for the proposed pre-initialization (Section 3.1). The mean gradient of the image is utilized to define the proposed cosine weighting function (Section 3.3).

All steps of the proposed RegionCut method are highly parallelizable. The evolution rule can be implemented as

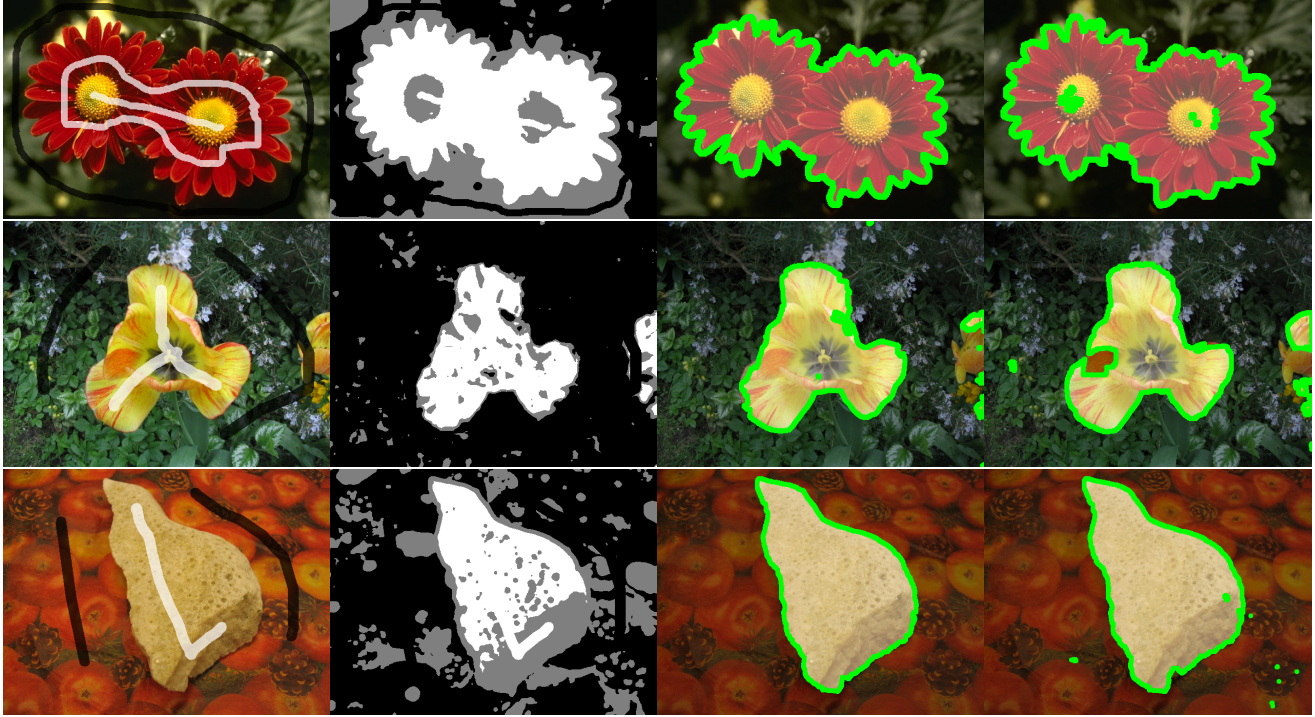


Figure 3. Example segmentations on the Microsoft segmentation benchmark. Columns: (i) original image and user strokes (black + white); (ii) proposed pre-initialization, white pixels are preinitialized as foreground and black pixels are background; (iii) segmentation result using RegionCut; (iv) segmentation result using Graph Cut. Comparing the segmentation results there are only minor differences.

a two-dimensional convolution so that the algorithm could assert itself in real-time-applications.

The proposed pre-initialization is also compatible to other segmentation methods using initial seeds.

4. Experiments

In this section we present the results of our contribution. For binary segmentation we compared RegionCut to other segmentation algorithms using the Microsoft segmentation benchmark^{1,2}. Since our algorithm uses user strokes we created our own stroke initializations. The error measurement used for binary segmentation is the same as proposed by Blake et al. in [1]. It is defined by

$$\varepsilon = \frac{\text{no. misclassified pixels}}{\text{no. pixels in unclassified region}}. \quad (6)$$

For the problem of multi-label segmentation we employ the IcgBench dataset, proposed by Santner et al. [13]³ and the

¹<http://research.microsoft.com/en-us/um/cambridge/projects/visionimagevideoediting/segmentation/grabcut.htm>

²<http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/>

³<http://gpu4vision.icg.tugraz.at/>

mean Dice evaluation score given by

$$\begin{aligned} \text{score} &= \frac{1}{N} \sum_{i=1}^N \text{dice}(E_i, GT_i) \\ &= \frac{1}{N} \sum_{i=1}^N \frac{2|E_i \cap GT_i|}{|E_i| + |GT_i|}, \end{aligned} \quad (7)$$

where $|\cdot|$ denotes the area of a segment E_i, GT_i the ground-truth labeling and N the number of segments.

Binary Segmentation: Table 1 shows the evaluation of the proposed RegionCut algorithm on the Microsoft segmentation benchmark in comparison with the original GrowCut algorithm and Graph Cut using a rough stroke initialization. We can observe that both improvements, the proposed pre-initialization and the new weighting function, enhance the GrowCut segmentation scheme. In combination the proposed RegionCut clearly outperforms the original GrowCut algorithm as well as the Graph Cut framework in terms of the mean error rate. Figures 3 and 6 visualize the proposed pre-initialization, the segmentation results based on a sparse initialization and compare it to Graph Cut. A comparison of the propagation between GrowCut and RegionCut is given in Figure 4.

We also evaluated the proposed RegionCut using a lasso initialization. Therefore we used the initializations provided

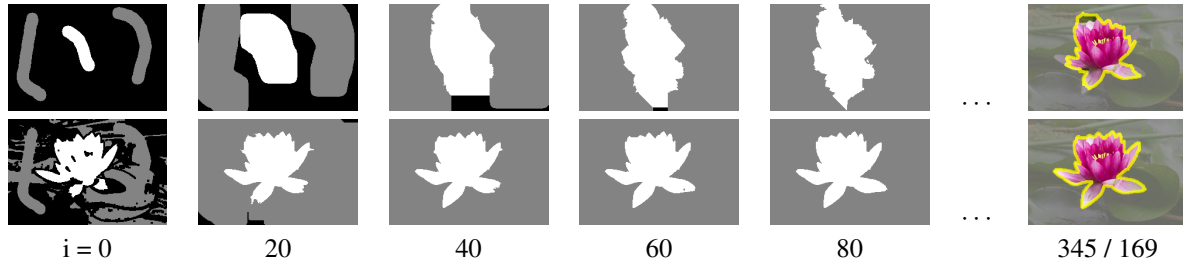


Figure 4. Comparison of the seed propagation. (Top) Original GrowCut; (Bottom) RegionCut using the proposed pre-initialization and the new weighting function; i states the number of iterations. Compared to GrowCut, RegionCut performed better and converged significantly faster.

Table 1. Mean error rates on the 50 images of the Microsoft segmentation benchmark for different combinations of our extensions using a few user strokes for initialization. Compared to GrowCut the proposed RegionCut performed significantly better and comparable to Graph Cut.

Segmentation method	Error rate
original GrowCut	11.59%
GrowCut with new g	10.96%
GrowCut with init (5 Gaussian)	7.19%
RegionCut	6.46%
Graph Cut ($\gamma = 50$, 5 Gaussian)	7.22%
Graph Cut ($\gamma = 20$, 5 Gaussian)	7.72%

Table 2. Error rates on the 50 images from the Microsoft segmentation benchmark using a lasso initialization. The results show that the proposed RegionCut outperforms the GMMRF approach [1].

Method	Tri-Map	Error rate
original GrowCut	lasso	5.57%
RegionCut	lasso	4.66%
GMMRF	lasso	7,9%

by the Microsoft segmentation benchmark. Using lasso initialization there is no need for region analysis. Our pre-initialization method aims to initialize as much as possible of the image except of the edges. The boundaries will then be converged by GrowCut oriented on the highest gradient. Lasso initializations pass exactly this data. Likewise, the course of the weighting function does not matter, because the algorithm does not have to grow over a long distance. However, compared to the original GrowCut the proposed extensions performed better and also outperformed the GMMRF [1] with discriminatively learned parameters. Results using the lasso initialization are given in Table 2.

Multi-Label Segmentation: The proposed RegionCut can also be applied to the problem of multi-label segmentation. To evaluate the segmentation quality we used the

IcgBench dataset, which contains 262 stroke initializations and corresponding ground-truth labels on 158 natural images. For evaluation we compared RegionCut to the original GrowCut and the interactive segmentation scheme based on random forests proposed by Santner et al. [13]. The mean dice score for the 262 stroke - ground-truth pairs of the IcgBench dataset using RegionCut was 0.886. Compared to GrowCut ($score = 0.820$) the score increased significantly. Furthermore, RegionCut is competitive to the interactive energy minimization scheme proposed by Santner et al. [13] using only RGB color features ($score = 0.887$). Example segmentation results and pre-initializations are given in Figures 5 and 7.

5. Conclusion

The proposed RegionCut combines the robustness of region information and the precision of gradient oriented segmentation methods. We propose a pre-initialization using region probabilities to create distributed seeds. This algorithm allows to find enclosed but non user initialized areas. In addition, smoothing probability data using spherical coordinates significantly enhances pre-initialization. Furthermore, we introduced an adaptive weighting function, based on the mean color noise, which allows to decouple image noise and the evolution progress.

The experimental results illustrate how powerful GrowCut can be if an initialization with distributed seeds is provided. However, the results also show the ineffectiveness of GrowCut only using strokes set by the user. In contrast RegionCut is able to compete with state-of-the-art segmentation methods without the need of time consuming user initializations.

An extended implementation of GrowCut will be provided on our web page. In order to verify our experiments it contains all proposed methods. Additionally, we will share the user strokes used to evaluate the Microsoft segmentation benchmark mentioned in Section 4.

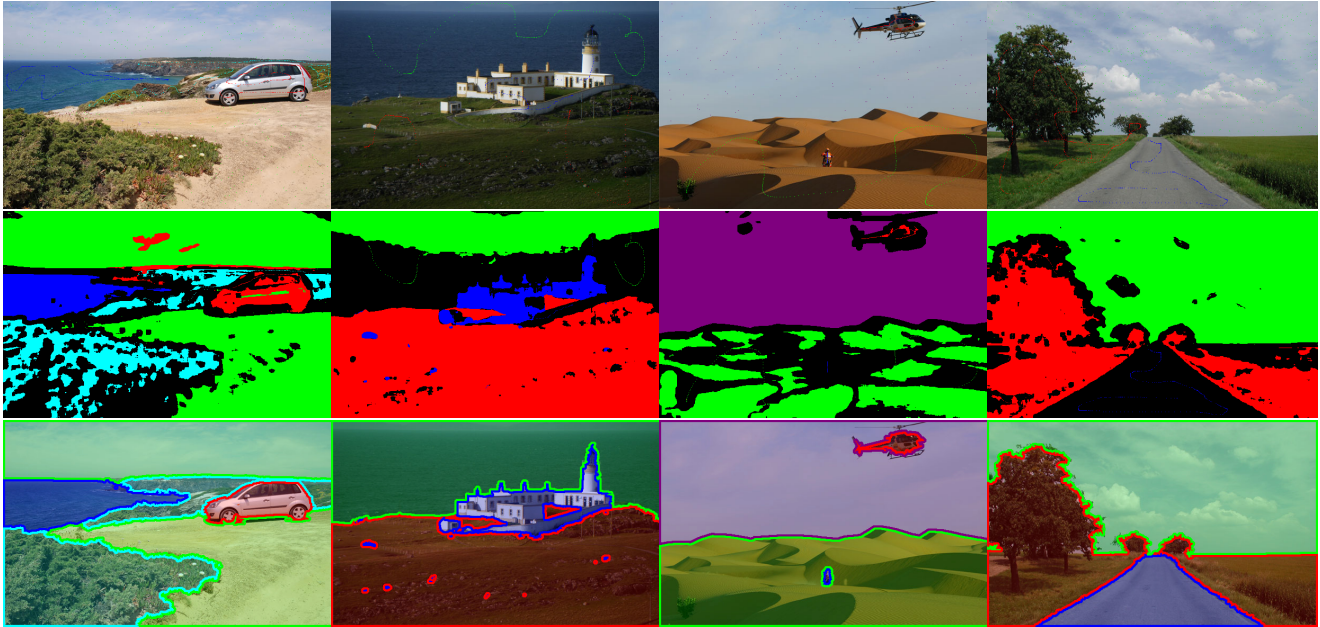


Figure 5. Example multi-label segmentations on the IcgBench benchmark dataset. Rows: (i) original image and seeds; (ii) proposed pre-initialization; (iii) segmentation result.

Acknowledgment

This research was conducted at the Institut für Informationsverarbeitung at Leibniz University Hannover, Germany. It was partially funded by Technicolor Research & Innovation Hannover, Germany.

References

- [1] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive gmmrf model. *ECCV*, pages 428–441, 2004. 5, 6
- [2] Y. Boykov and M. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *ICCV*, volume 1, pages 105–112, 2001. 1
- [3] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *TPAMI*, 26(9):1124–1137, 2004. 1
- [4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *TPAMI*, 23(11):1222–1239, 2002. 1
- [5] T. Chan and L. Vese. Active contours without edges. *IEEE TIP*, 10(2):266–277, 2001. 1
- [6] D. Cremers, O. Fluck, M. Rousson, and S. Aharon. A probabilistic level set formulation for interactive organ segmentation. In *Proc. of the SPIE Medical Imaging*, 2007. 1
- [7] P. Felzenszwalb and D. Huttenlocher. Efficient belief propagation for early vision. *IJCV*, 70(1):41–54, 2006. 1
- [8] G. Hernandez and H. Herrmann. Cellular automata for elementary image enhancement. *Graphical models and image processing*, 58(1):82–89, 1996. 1
- [9] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *IJCV*, 1(4):321–331, 1988. 1
- [10] E. Mortensen and W. Barrett. Interactive segmentation with intelligent scissors. *Graphical Models and Image Processing*, 60(5):349–384, 1998. 1
- [11] A. Popovici and D. Popovici. Cellular automata in image processing. In *Fifteenth International Symposium on Mathematical Theory of Networks and Systems*, 2002. 1
- [12] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *SIGGRAPH*, 23(3):309–314, 2004. 1
- [13] J. Santner, T. Pock, and H. Bischof. Interactive multi-label segmentation. *ACCV*, pages 397–410, 2010. 5, 6
- [14] J. Santner, M. Unger, T. Pock, C. Leistner, A. Saffari, and H. Bischof. Interactive texture segmentation using random forests and total variation. In *BMVC*, 2009. 1
- [15] B. Scheuermann and B. Rosenhahn. Interactive image segmentation using level sets and dempster-shafer theory of evidence. *SCIA*, pages 656–665, 2011. 1
- [16] M. Unger, T. Pock, W. Trobin, D. Cremers, and H. Bischof. Tvseg-interactive total variation based image segmentation. In *BMVC*, 2008. 1
- [17] V. Vezhnevets and V. Konouchine. Growcut: Interactive multi-label nd image segmentation by cellular automata. In *Proc. of Graphicon*, pages 150–156, 2005. 1, 2
- [18] J. von Neumann. Theory of self-reproducing automata. *University of Illinois Press. Ed. and Completed by A. Burks*, 1966. 1
- [19] F. Wang, X. Wang, and T. Li. Efficient label propagation for interactive image segmentation. In *ICMLA*, pages 136–141, 2007. 1

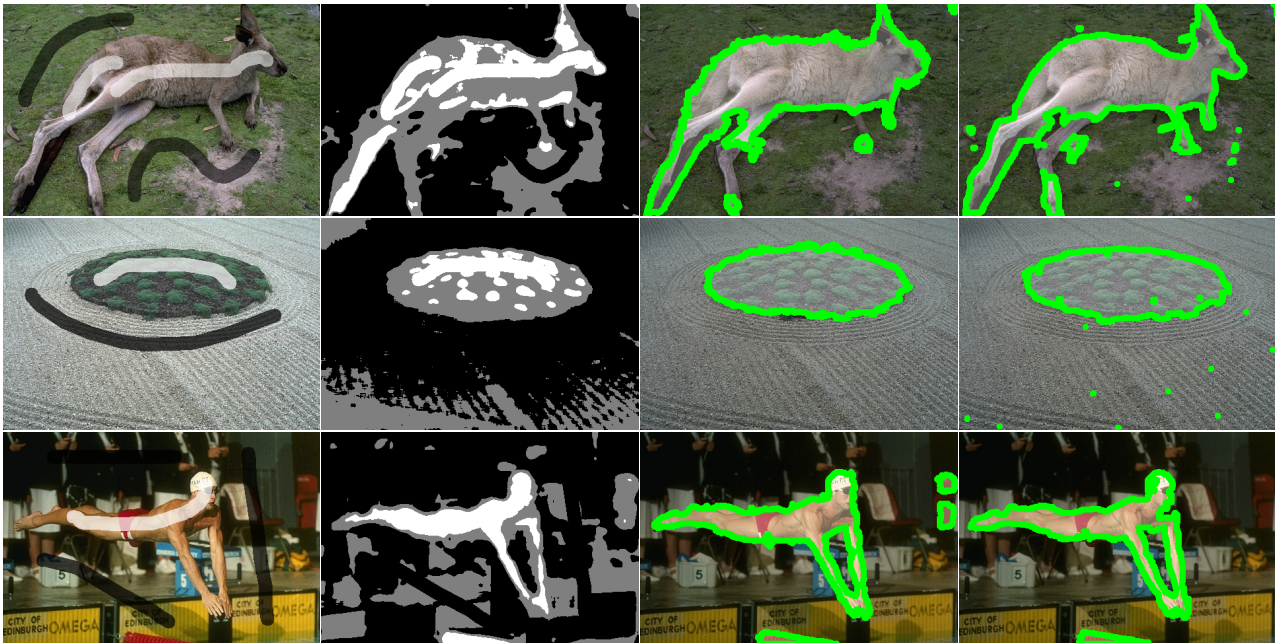


Figure 6. Further example segmentations on the Microsoft dataset. Columns: (i) original image and seeds; (ii) proposed pre-initialization, white pixels are preinitialized as foreground and black pixels are background; (iii) segmentation result using RegionCut; (iv) segmentation result using graph cut. Comparing the segmentation results there are only minor differences.

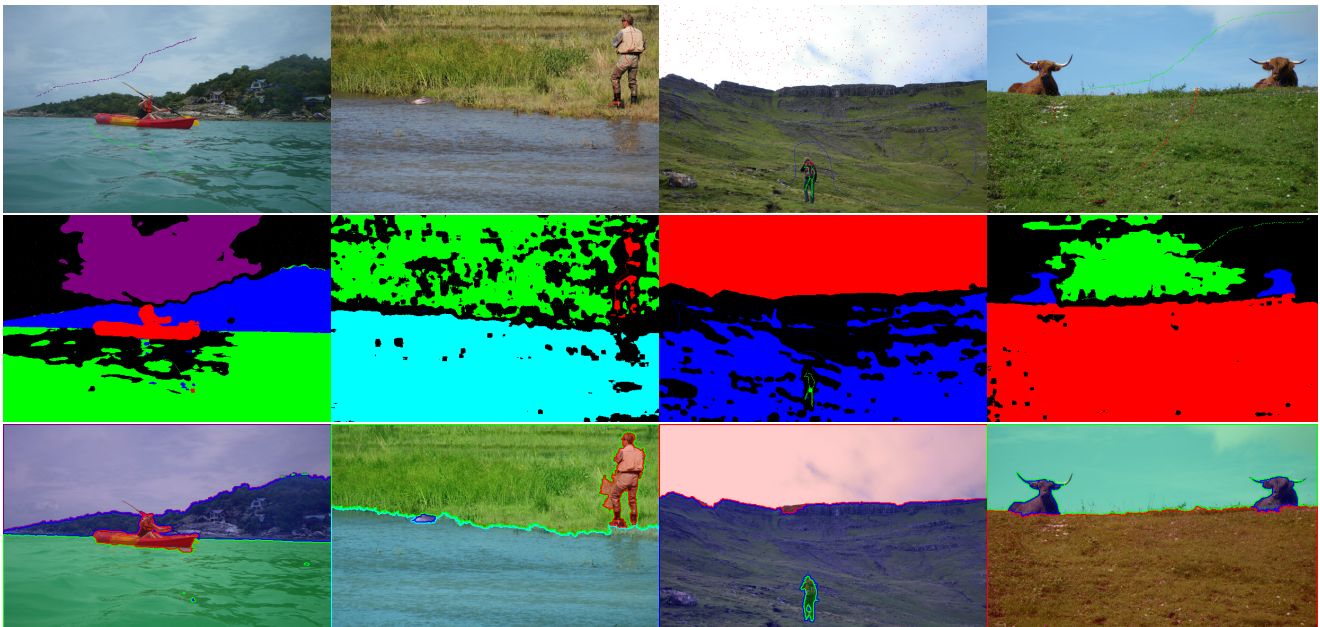


Figure 7. Additional multi-label segmentation results on the IcgBench benchmark dataset. Rows: (i) original image and seeds; (ii) proposed pre-initialization; (iii) segmentation result.