Chapter 57
Image Segmentation Based on Edge Growth

Xiuyan Cao, Weili Ding, Shuo Hu and Liancheng Su

Abstract We propose a novel approach for solving the over-segmentation problem in image segmentation. Rather than focusing on clustering gray pixels in the image data, our approach aims at extracting salient regions based on edges information. We treat image segmentation as an edge linking problem and propose a novel process, *edge growth*, for segmenting the image. The edge growth process is starting from the breakpoints and then prolongs the edge chains based on its local structure until there is another breakpoint or edges. We have applied this approach to segmenting static images without tuning any parameters. The experimental result shows the effective of our proposed algorithm.

Keywords Image segmentation · Edge detection · Breakpoints · Edge Growth

57.1 Introduction

Image segmentation is an essential step for image analysis and understanding, which can be applied in several practical problems such as medical research, intelligent recognition, 3D reconstruction and image retrieval/compression. An ideal segmentation algorithm should cluster pixels into salient regions which corresponding to individual surfaces, objects or nature parts of objects.

X. Cao (✉) · W. Ding · S. Hu · L. Su
Institute of Electrical Engineering, Yanshan University, No 438 He Bei Road,
Qinhuangdao City, China
e-mail: xiuyan86@gmail.com

W. Ding
e-mail: weiye51@ysu.edu.cn

W. Lu et al. (eds.), *Proceedings of the 2012 International Conference on Information Technology and Software Engineering*, Lecture Notes in Electrical Engineering 212,
DOI: 10.1007/978-3-642-34531-9_57, © Springer-Verlag Berlin Heidelberg 2013

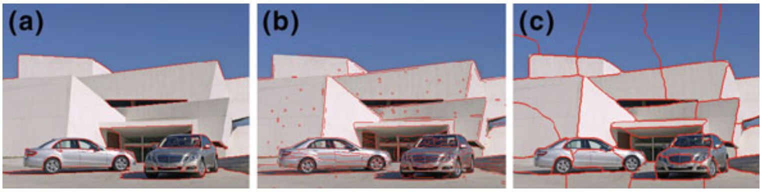


Fig. 57.1 Segmentation results with four classical algorithms. **a** wateshed. **b** Meanshift. **c** N-cuts

Image segmentation is an old and recurrent problem in computer vision. Various algorithms have been proposed to cope with this problem. Generally, The proposed classical algorithms including watershed [1, 2], mean shift [3–5], region-growing [6, 7], and superpixel [8, 9] are based on region, which start from the gray pixel, and cluster the pixels with similar value into different regions. As shown in Fig. 57.1, the Watershed algorithm is fast and easy to extend to n-dimensional images, but it may leads to under-segmentation in the absence of boundary cues (see Fig. 57.1b), and the result depend on the threshold; Mean shift is an effective clustering iterative algorithm, but it also creates under-segmentation and unmeaning regions because of the lacking compactness constraint; Super-pixel algorithms based on graph cut [9] and geometric flows are proposed in recent years, which can divide an image into many small regions. however most of the regions in Fig. 57.1d are not corresponding an independent part or object, and the long runtime is another disadvantage of these algorithms.

The other kind of algorithms is based on edges [10–13], which attain the segmentation goal by linking the edges into a series of closed boundary. For example, Chen and Sun [12] present a new edge-based interactive image segmentation method based the edge image which is generated with the edge detector; Takumi Uemura and Gou Koutaki [10] propose an algorithm by using their proposed boundary code method, which is more accurately than existing methods in images those have a narrow elongated object, shading and blurring. However, this kind of methods are not rich compared with the region based method, but it is more attractive because it can attain a more meaningful result and avoid the threshold problem.

In this paper, we propose an effective algorithm to divide the image into a series meaningful region by using the edge information. Our method is starting from the breakpoints which caused by the canny detector, and then the edge chains are grown at the breakpoints with the proposed criterions to achieve the goal of edge linking. Finally, two adjacent closed regions with similar color will be merged into one region.

Section 57.2 gives the global procedure of the proposed algorithm; Sect. 57.3 describes the key steps of the proposed algorithm; Sect. 57.4 comments on the experimental results in detail and Sect. 57.5 concludes the paper.

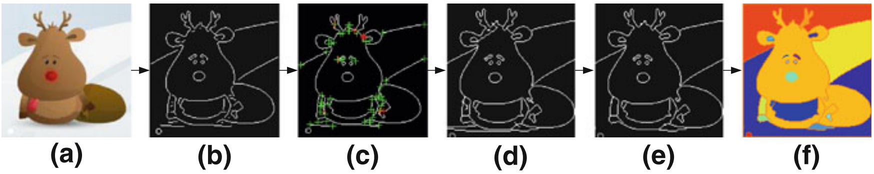


Fig. 57.2 The process of our proposed algorithm. **a** input image; **b** edge detection; **c** breakpoints detection; **d** edge growth; **e** region merging; **d** segmentation result

57.2 Segmentation Process

As shown in Fig. 57.2, the process of our algorithm mainly contains four steps:

1. **Edge Detection**. We choose canny detector to obtain the edges in the image because of its excellent linking ability for edges with low contrast.
2. **Breakpoints Detection**. As shown in Fig. 57.2c, there are a lot of breakpoints (the green crosses), so the boundary of the toy is not closed. In order to remove them, we extract breakpoints with the algorithm described in Sect. 57.3.1 and extract a set of edge chains which connected with the breakpoints.
3. **Edge Growth**. If the edge chain connected with the breakpoint is prolonged based on a certain criterion, all edges will be closed and the salient region will be segmented. Based on this idea, we propose our algorithm in Sect. 57.3.2, and the closed regions are labeled after edge growth (see Fig. 57.2d).
4. **Region Merging**. Because the unreasonable regions are still exist after edge growth (e.g. the regions in the head of the toy), we merge them based on the color similarity (See the detailed algorithm in Sect. 57.3.3), and the final labeled image is shown in Fig. 57.2f.

57.3 Key Steps in Segmentation

57.3.1 Breakpoint Detection

For a non-zero pixel (the gray pixel in Fig. 57.3) in the edge map and its 3-by-3 neighborhood whose label is shown in Fig. 57.3a, we define

$$f(x) = \sum_{j=1}^{8} (a_j - b_j), \tag{57.1}$$

Where $a = [x(1), x(2), x(3), x(6), x(9), x(8), x(7), x(4)]$, $b = [x(2), x(3), x(6), x(9), x(8), x(7), x(4), x(1)]$ and $x(i) = \{0, 1\}, (i = 1\ldots9)$ is the value of each pixel in the 3-by-3 neighborhood matrix. Based on the previous study [14], the non-zero pixel is a breakpoint (See Fig. 57.3b) when $f(\mathrm{x}) = 2$.

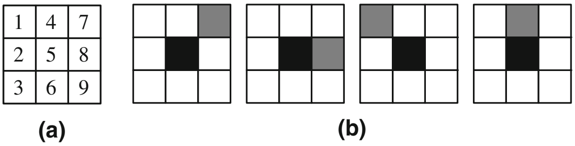


Fig. 57.3 8-neighborhood of a breakpoint. **a** The label; **b** Breakpoints

57.3.2 Edge Growth

Given a breakpoint P we consider its local structure in a sliding window $(M \times M)$. As shown in Fig. 57.4, the edge will be grown at the breakpoint based on two cases:

1. Other breakpoints such as $P_1, P_2, \ldots, P_n,$ (Fig. 57.4a) are existed in the window:

First, the total gradient of each route (four routes in Fig. 57.4a) from P to P_i will be computed by

$$G = \sum_{i=1}^{n-1} \left((Gr(x_{i+1}, y_{i+1}) - Gr(x_i, y_i))^2 + (Gr(x_i, y_{i+i}) - Gr(x_i, y_i))^2 \right)^{1/2} \tag{57.2}$$

Where G is the grayscale, (x, y) is the coordinates of the route, $Gr(x, y)$ is the gray value;

Then the route with the maximum value of G will be chosen as the final linking route (the black route in Fig. 57.4a), and the edge will be linked in the sliding window finally.

2. No other breakpoints are existed in the window (Fig. 57.4.b):

In this case, we map the edges connected with P (the gray pixels in Fig. 57.4b) to its symmetric positions(the black pixels in Fig. 57.4b) in the sliding window and update P. Then above step will be repeated until P meets the image boundary or other edges.

57.3.3 Region Merging

After edge growth, the edge map will be segmented into many closed regions. For an added edge with a certain length, we compute the average gray value of its two adjacent regions which labeled as i and j, and defined them as G_i and G_j. If their absolute difference is smaller than the predefined threshold T, this edge will be deleted in edge map B.

$$G_{i/j} = \frac{1}{m} \sum_{k=1}^{m} Gr(x_k, y_k) \tag{57.3}$$

Where Gr is the grayscale image, $Gr(x_k, y_k)$ is the kth pixel's gray value of the closed region i or j, and m is the total numbers of pixels.

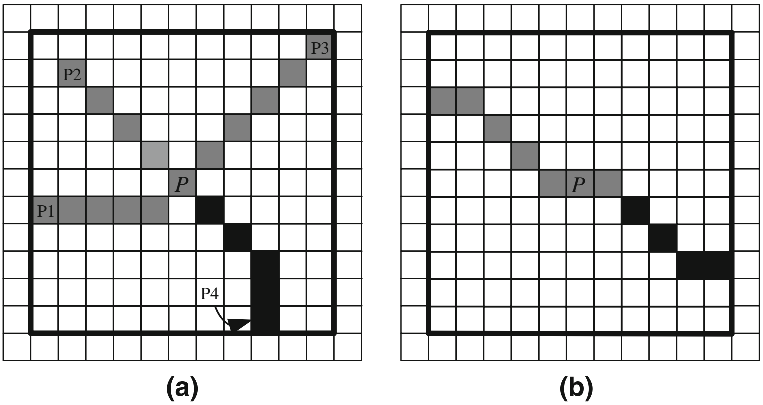
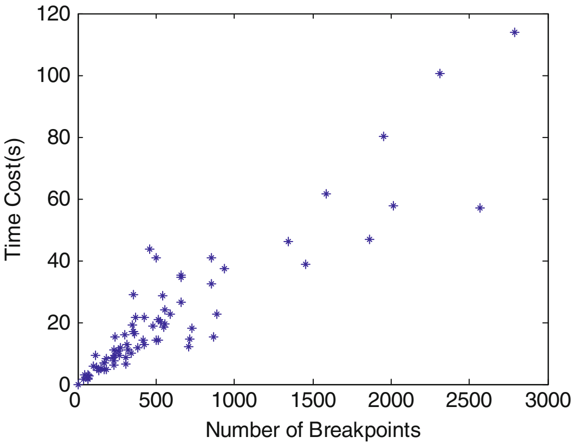


Fig. 57.4 Edge Growth method, the *black* boxes are the sliding window. **a** P_4 is the final point linked to P, the final linking route (*black*); **b** original edge (*gray*), the mapping part (*black*)

Fig. 57.5 Time cost (in seconds) requires by our algorithm for the processing of 72 images



57.4 Experiment

Figure 57.5 gives the computation time of all 72 images. In our experiments, all results are implemented on Intel Core 2 Duo CPU 2.8 GHz with Matlab 2008a. We can see that the computation time is approximately proportional to the number of breakpoints, and 91.54 % images require a time less than 40 s.

Figure 57.6 shows some qualitative results of our algorithm in a part of chosen nature images which contain highly geometrical contents. We can see that the results are close with the human vision system and nearly all regions are segmented completely, and the regions connected with low contrast edges are also extracted well. Specifically, the regions in the traffic sign image and the road

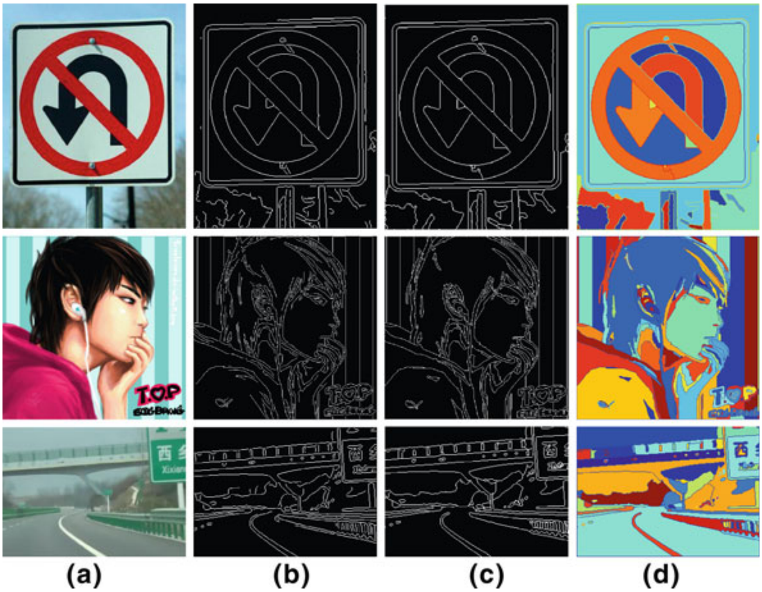


Fig. 57.6 Segmentation result. **a** input image; **b** edge map; **c** edge growth; **d** labeled segments

image are meaningful enough to the further image understanding such as shape analysis, object identification and 3D reconstruction.

Figure 57.7 shows a comparison of various classical algorithms and our proposed algorithm. As shown in the figure, the improved Watershed algorithm leads to under-segmentation because it depends much on the gray level information. For instance, the whole image was segmented into just two parts and some detailed regions such as the trees and the road sign were lost in the road image, and the cylinder is not segmented from the background in the second image; Mean shift algorithm cluster a lot of crushing and unmeaning regions(See the small regions in the human image) and need to set different parameters for different image; N-cuts algorithm divided an complete region into a lot of irregular regions, and the results around the edges with low contrast are not satisfied. Compared with the above algorithms, our algorithm gives a more encouraging result because the shadow of the person, the cylinder and the traffic signs are all segmented completely. In addition, the runtime of N-cuts is much longer than ours, as shown in Fig. 57.7.

57.5 Conclusion

In this paper, we proposed a novel method to segment image by using the edges information. Our approach avoids the problem in threshold setting and seed selection, and the segmentation regions are more practical significance. However,

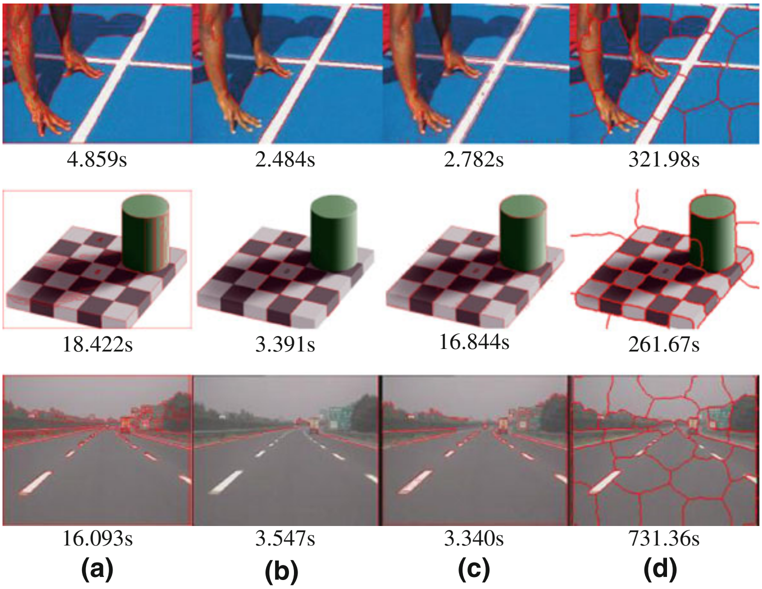


Fig. 57.7 A comparison of four image segment methods. **a** proposed algorithm; **b** Watershed; **c** Mean Shift; **d** N-cuts

there are still many regions are over-segmentation. The main reasons are: (1) noise interference. Canny operator takes noises as a lot of short edges, which will affect the segmentation result; (2) Errors during edge growth, Though most of edge chains can be linked or grown in a right way based on our algorithm, some edge chains whose shape is a curve will be prolonged in a wrong way. So our work was just beginning. It should be studied further to solve the above problems.

Acknowledgments The research was supported by NSFC (61005034, 60905046) and the natural science foundation of Hebei Province (F2012203185, F2012203182).

References

1. Vincent L, Soille P (1991) Watersheds in digital spaces: an efficient algorithm based on immersion simulations. IEEE Trans Pattern Anal Mach Intell 13(6):583–598
2. Beucher S, Lantue C (1979) Use of watersheds in contour detection. In: Proceedings Int'l Workshop Image Processing Real-Time Edge and Motion Detection/Estimation
3. Edoardo A, Roberto P, Orazio G (2007) Fuzzy C-Means segmentation on brain MR slices corrupted by RF-inhomogeneity. Lect Notes Comput Sci 4578:378–384
4. Zhou HY, Gerald S, Shi CM (2008) A mean shift based fuzzy c-means algorithm for image segmentation. Proceedings of the 30th annual international conference of the ieee engineering in medicine and biology society, Personalized Healthcare through Technology, pp 3091–3094
5. Comaniciu D, Meer P (2002) Mean shift: a robust approach toward feature space analysis. IEEE Trans Pattern Anal Mach Intell 24(5):603–619

6. Ren X, Malik J (2003) Learning a classification model for segmentation. In: Proceedings IEEE Int'l Conference Computer Vision, pp 10–17
7. Yu S, Shi J (2003) Multiclass spectral clustering. Proc IEEE Int'l Conf Computer Vision 1:313–319
8. Levinshtein A, Stere A, Kutulakos N (2009) Turbo pixels: fast superpixels using geometric flows. IEEE Int'l Conf Comput Vision 31:2290–2297
9. Shi J, Malik J (2000) Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell 22(8):888–905
10. Uemura T, Koutaki G, Uchimura K (2011) Image segmentation based on Edge detection using boundary code. ICCV 7:6073–6083
11. Lin CH, Chen CC (2010) Image segmentation based on edge detection and region growing for thinprep-cervical smear. Pattern Recognit Artif Intell 24(7):1061–1089
12. Chen Q, Sun QS, Xia DS (2010) A new edge-based interactive image segmentation method. The international society for optical engineering, international conference on image processing and pattern recognition in industrial engineering, p 7820
13. Lindeberg T, Li MX (1997) Segmentation and classification of edges using minimum description length approximation and complementary junction cues. Comput Vis Image Underst 67(1):88–98
14. Kovesi PD (2008) http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/index.html