Open Boundary Capable Edge Grouping with Feature Maps

Joachim S. Stahl, Kenton Oliver and Song Wang
Department of Computer Science and Engineering
University of South Carolina, Columbia, SC 29208
{stahlj|oliverwk|songwang}@engr.sc.edu

Abstract

Edge grouping methods aim at detecting the complete boundaries of salient structures in noisy images. In this paper, we develop a new edge grouping method that exhibits several useful properties. First, it combines both boundary and region information by defining a unified grouping cost. The region information of the desirable structures is included as a binary feature map that is of the same size as the input image. Second, it finds the globally optimal solution of this grouping cost. We extend a prior graph-based edge grouping algorithm to achieve this goal. Third, it can detect both closed boundaries, where the structure of interest lies completely within the image perimeter, and open boundaries, where the structure of interest is cropped by the image perimeter. Given this capability for detecting both open and closed boundaries, the proposed method can be extended to segment an image into disjoint regions in a hierarchical way. Experimental results on real images are reported, with a comparison against a prior edge grouping method that can only detect closed boundaries.

1. Introduction

Edge grouping methods are mid-level vision methods that can help identify salient structures in noisy images, and are very important to solve many high-level vision problems such as object recognition or content-based image retrieval [11, 19]. Edge grouping methods work by first obtaining a set of edge segments from an image (usually using edge detection and line approximation), and then grouping and connecting a subset of these edge segments to form the boundary of a salient structure. For example, an edge grouping method may first detect a set of edge segments (Fig. 1(b)) from the image shown in Fig. 1(a), and then group these edge segments to produce a boundary shown in Fig. 1(c).

While edge grouping has been widely investigated for decades [20, 7, 5, 6, 1, 2, 4, 8, 18, 12, 21, 23, 24], most available methods still lack the necessary robustness when applied to noisy real images. Three of the reasons that may degrade the grouping robustness are:

1. Most available edge grouping methods only consider boundary information, such as proximity, closure, and continuity, and do not consider any region information, such as the intensity homogeneity and appearance of the desirable salient structures. In fact, it is usually difficult to detect the salient structures by using only boundary information. For example, it is difficult even for human vision to identify salient structures from edge segments shown in Fig. 1(b).

2. Many available edge grouping methods employ grouping algorithms that can only find locally optimal solutions in terms of the selected grouping cost. The groupings resulting from locally optimal solutions are usually more sensitive to image noise.

3. Many available edge grouping methods can not detect complete structure boundaries. A complete boundary may be *closed* or *open*. As shown in Fig. 1(c) and (d), a closed boundary indicates that the structure of interest lies completely inside the image perimeter and an open boundary indicates that the structure of interest is located partially inside the image perimeter. Note that, an open boundary is different from an incomplete boundary fragment (Fig. 1(e)). As shown in Fig. 1(d), just like a closed boundary, an open boundary always partitions an image into two disjoint regions. Many available edge grouping methods can only detect an incomplete boundary fragment that cannot bipartition the image. Several recent edge grouping methods impose the constraint of boundary closure to detect complete boundaries. However, it is usually nontrivial to extend them to also detect open boundaries.

Detecting a complete boundary always partitions an image into two disjoint regions. We can then repeat the same edge grouping method on each of the resulting regions to further partition them and construct an image hierarchy. Being able to detect open boundaries is important not only for

Figure 1. An illustration of boundary completeness. (a) Input image, (b) detected edge segments, (c) a closed boundary. (d) an open boundary, (e) an incomplete boundary fragment, (f) An open boundary (in blue) is detected when repeating edge grouping for constructing image hierarchy.

locating salient structures partially inside the image perimeter, but also for constructing the image hierarchy by repetitive application of an edge grouping method. The major reason is that structures detected in different rounds may partially overlap or occlude each other. An example is shown in Fig. 1(f), a closed boundary (in red) and an open boundary (in blue) are extracted from the first and second rounds respectively. The second boundary (in blue) is open (cropped by the red boundary detected in the previous round) because it describes a partially occluded structure.

In this paper, we present a new edge grouping method that addresses the above three problems simultaneously. First, we incorporate region information by using a binary feature map of the same size as the input image. By showing whether a pixel is part of the desirable structure or not, this feature map can be constructed by any region-based analysis method and may contain much noise. We define a unified grouping cost that combines the boundary information of proximity and the region information of this feature map. Second, we show that the globally optimal solution that minimizes the proposed grouping cost can be found by an available graph algorithm. Third, by carefully specifying the feature values in the feature map and dividing the image perimeter into a set of edge segments, we show that the proposed method can detect both closed and open boundaries.

Closely related to the proposed method is the work developed by Stahl and Wang [22], where the boundary information of proximity and the region information of area and intensity are combined into a ratio-form grouping cost. By applying the ratio-contour algorithm [23], the method introduced in [22] can detect salient structures with a preference for larger areas and certain intensities. The method developed in this paper adopts a similar ratio-form grouping cost function and also uses the ratio-contour algorithm for finding the optimal solution. However, different from the method in [22], we use the binary feature map to more accurately represent the desirable region information and more importantly, we develop a new way to choose the feature values in the feature map, which makes the developed method capable of detecting open boundaries as well as closed boundaries. Note that, the ratio-form grouping cost for combining boundary and region information was first developed by Jermyn and Ishikawa [9] for grouping pixels

into a closed boundary. We believe the proposed method of defining a binary feature map with specific feature values can also be applied to this pixel-grouping method for better detecting open boundaries.

The remainder of this paper is organized as follows: Section 2 describes the edge grouping and the cost function used in this paper. Section 3 describes how to extend the edge grouping method to produce open boundaries. Section 4 presents the method of dividing the image perimeter into a set of edge segments. Section 5 describes a simple method for constructing feature maps to support edge grouping on color images. Section 6 introduces an approach to repeat the proposed method for constructing an image hierarchy. Section 7 reports experiment results on a set of real images. Section 8 concludes this paper.

2. Edge Grouping for Closed Boundaries

Typically, edge grouping for detecting *closed* boundaries takes three steps. First, from an input image we detect a set of edges, e.g. using a Canny edge detector [3] and based on these edges, we construct a set of parametric edge segments using line or curve fitting. In this paper, we simply construct edge segments as straight line segments, as shown in Fig. 2(b). We refer to these straight line segments as *detected segments*. Note that, a detected segment may come from either the boundaries of the salient structures or image noise/texture. Second, since the detected segments are disconnected from each other, we construct an additional set of straight line segments to connect them, and this will allow us to construct closed boundaries. We refer to these new straight line segments as *gap-filling segments*, and construct them between every pair of detected segments. A closed boundary is then defined as a cycle of alternating detected and gap-filling segments, as shown in Fig. 2(d). A grouping cost is defined that describes the saliency of each valid closed boundary. Third, we develop a grouping algorithm to identify from all valid *closed* boundaries the one with the minimum grouping cost, also as shown in Fig. 2(d).

In this paper, we consider the boundary information of proximity, i.e., the smaller the total gap length, the more salient the resulting boundary. To improve the grouping robustness, we further introduce the region information in the form of a binary feature map. As shown in Fig. 2(c), a bi-
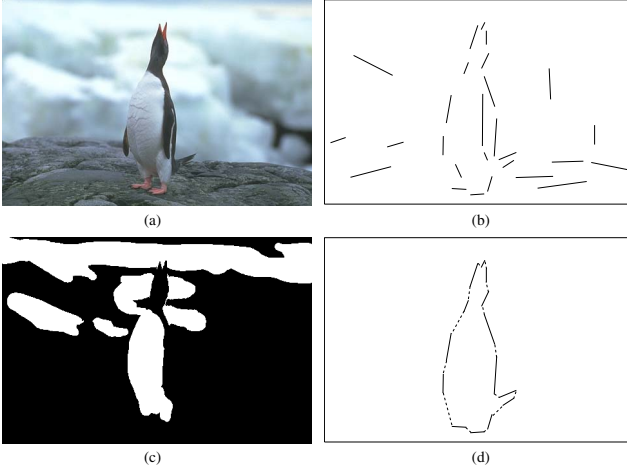
Figure 2. An illustration of edge grouping with a binary feature map. (a) Input image, (b) detected segments, (c) binary feature map, and (d) the detected *closed* boundary that traverses detected (solid) and gap-filling (dashed) segments alternately.

nary feature map $M(x, y)$ is of the same size as the input image $I(x, y)$ and reflects whether pixel (x, y) has the desired property or not. It can be constructed from the input image $I(x, y)$ using an image-analysis method and/or any available *a priori* knowledge of the appearance of the desirable salient structures. Therefore, just like the construction of detected segments, the feature map also contains noise and errors. For example, the feature map $M(x, y)$ shown in Fig. 2(c) is constructed by thresholding the pixel intensity of the input image, since the desirable structure, i.e., the penguin, in general has a larger intensity than the background. We can set $M(x, y) = \alpha$ (white) to indicate that pixel (x, y) belongs to the penguin and $M(x, y) = \beta$ (black) otherwise. However, in Fig. 2(c), part of the background is labeled with a feature value of α because of being white, while part of the penguin is labeled with a feature value of β because of being black. The fundamental goal of this work is to integrate the detected segments and this feature map, both of which contain noise and errors, to more robustly detect the desirable salient structure.

By following [22], we can define a unified grouping cost for a closed boundary $\mathcal{B}$ as

$$\phi(\mathcal{B}) = \frac{|\mathcal{B}_G|}{\iint_{R(\mathcal{B})} M(x, y) dx dy}, \qquad (1)$$

where $|\mathcal{B}_G|$ is the total length of all the gap-filling segments along the boundary $\mathcal{B}$. This accounts for the Gestalt law of proximity, where a smaller total gap length $|\mathcal{B}_G|$ represents better proximity. $R(\mathcal{B})$ is the region enclosed by the boundary $\mathcal{B}$ and $\iint_{R(\mathcal{B})} M(x, y) dx dy$ is the sum of the feature values of the pixels, taken from a binary feature map M, inside the region enclosed by $\mathcal{B}$. Since we want this

grouping cost to produce a boundary that encloses as many pixels with a feature value of α as possible, by following [22], we can choose the feature values to be $\alpha = 1$ and $\beta = 0$ for the feature map $M(x, y)$ and it has been shown that the ratio-contour algorithm [23] can be used to find the global optima of this grouping cost.

3. Extension to Detect Open Boundaries

The above grouping cost and feature values are well suited for closed boundary detection, however these values make it difficult to detect open boundaries. The typical way of handling open boundary detection in edge grouping is to divide the image perimeter into a set of detected/gap-filling segments. This way, we can directly apply the same edge grouping method for detecting only closed boundaries. If the resulting optimal closed boundary contains one or more detected/gap-filling segments constructed from the image perimeter, it is in fact an open boundary, as shown in Figs. 3(b) and (d), where the resulting closed boundaries (red detected segments) contain part of the perimeter and in fact are open boundaries. To make it unbiased between detecting closed and open boundaries, we usually need to assume zero or small gaps along the image perimeter. In this case, the above edge grouping method based on grouping cost (1) and the feature values of $\alpha = 1$ and $\beta = 0$ would lead to a degenerate case of detecting the image perimeter as the optimal boundary $\mathcal{B}$, because $|\mathcal{B}_G|$ is zero or small and $\iint_{R(\mathcal{B})} M(x, y) dx dy$ is maximum.
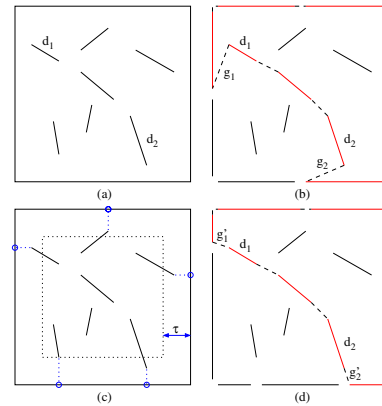

Figure 3. An illustration of open boundary detection and perimeter-segment construction, (a) Detected segments and image perimeter. (b) Constructing perimeter segments by a uniform division and the possible resulting open boundary. Dashed lines are gap-filling segments. The gap-filling segments along the image perimeter are in fact very short, but they are exaggerated here for clarity. (c) Projecting segment endpoints to the image perimeter. (d) Constructing perimeter segments by dividing the image perimeter at the projection points obtained in (c).

To address this problem, we choose $\alpha > 0$ and $\beta < 0$ such that $\iint_{R(\mathcal{B})} M(x, y) dx dy = 0$, when $\mathcal{B}$ is the image

perimeter. Without loss of generality, we choose $\alpha = 1$ and

$$\beta = -\frac{\sum_{(x,y):M(x,y)>0} 1}{\sum_{(x,y):M(x,y)<0} 1}. \tag{2}$$

Since the feature value β is negative, $\iint_{R(\mathcal{B})} M(x,y)dxdy$ may also take a negative value for a boundary $\mathcal{B}$. Therefore, we also update the grouping cost to

$$\phi(\mathcal{B}) = \frac{|\mathcal{B}_G|}{\left| \iint_{R(\mathcal{B})} M(x,y)dxdy \right|}, \tag{3}$$

where both the numerator and denominator are non-negative. It is easy to verify that the same ratio-contour algorithm [22] can be used to find the globally optimal boundary $\mathcal{B}$ that minimizes this new grouping cost with the newly chosen feature value of β.

There is another advantage of choosing a negative value for β as given in Eq. (2) in the grouping cost (3). The detected region $R(\mathcal{B})$ tends to not contain many pixels with different feature values because they usually lead to smaller $\iint_{R(\mathcal{B})} M(x,y)dxdy$. Therefore, this new edge grouping method can detect tightly aligned structure boundaries by excluding the background pixels. This advantage holds whether a closed or open boundary is detected. In Section 7, we are going show this advantage through experiments on real images.

There is also a problem with this new method: it may detect a boundary that encloses a region only or mainly with the feature value β. With nonzero values α and β, the grouping cost (3) may treat them equally and has no preference to detect a boundary that encloses mainly pixels with feature value α. However, in practice, this is not a serious problem because (a) we always detect a complete boundary that bipartitions the image. Detecting the background fully equals to detecting the foreground; (b) as discussed later, we can repeat the proposed method to iteratively partition the image. Therefore, even if only part of the background is detected in the first round, we may detect the desirable salient structure in a later round.

4. Dividing Image Perimeter into Segments

As discussed above, by dividing the image perimeter into a sequence of detected/gap-filling segments, the proposed method supports the detection of both closed and open boundaries. Note that, by defining a negative β as in Eq. (2), the length of gap-filling segments along the image perimeter can be set to be zero or very small. This way, the problem is reduced to dividing the image perimeter into a set of detected segments, which we refer to as *perimeter* segments. One strategy is to uniformly divide the image perimeter to construct equal-length perimeter segments, as

shown in Fig. 3(b). A problem of this strategy is the choice of the perimeter-segment length. If it is too small, it may result in too many perimeter segments. Since we need to construct gap-filling segments between each perimeter segment and other detected segments inside the image perimeter, more perimeter segments results in many more gap-filling segments and increases the algorithm's computational time. Also having too many perimeter segments can be a problem even if we allow only a small gap between adjacent segments. A large amount of perimeter segments in this case can make summation of such small gaps add up to a significant total gap that can affect the total cost of the boundary. This would bias the method to avoid using the perimeter segments except in small quantities. If the perimeter segment-length is too large, it may affect the correct detection of an open boundary. As shown in Fig. 3(b), two long gaps g_1 and g_2 are filled to construct the open boundary shown in red. In fact, the gaps from the detected segments d_1 and d_2 to the image perimeter are not that large. Also, having long perimeter segments can bias the method to produce open boundaries because the total gap becomes very small.

In this paper, we develop a new strategy to divide the image perimeter for constructing perimeter segments. As shown in Fig. 3(c), we project the endpoints of each detected segment inside the image perimeter to the image perimeter (all four sides). These projection points are chosen as the division locations. Between each pair of neighboring projection points, we construct a perimeter segment, and at each projection point, we construct a zero-length or very small length gap-filling segment to connect two neighboring perimeter segments. No gap-filling segment is constructed between non-neighboring perimeter segments. But gap-filling segments will be constructed between non-perimeter detected segments and perimeter segments. In practice, a long gap is not likely to be included in an optimal boundary. We can filter out many projection points for which the projection length is larger than a threshold τ, i.e., we only project all segment endpoints outside of the dotted box shown in Fig. 3(c) to the image perimeter for constructing the perimeter segments. For all the experiments in this paper we set $\tau = 50$ pixels. Using this construction the new gaps, g_1' and g_2', for the detected segments d_1 and d_2 are not that large, as shown in Fig. 3(d). By following the construction presented in this section we avoid using an arbitrary uniform division of the image perimeter, which could lead to the problems mentioned earlier. Instead the decision to divide the perimeter solely depends on the detected segments.

5. Feature Map Construction

In this section we introduce a simple method to construct the feature map by assuming that the desirable salient

structure shows different intensity from the background. It should be noted that any method for generating a binary feature map is applicable, we chose to demonstrate this one for its simplicity, and ease of implementation. Particularly, this method consists of two steps: First, we apply a region merging method to the input image to divide the image into a small number of disjoint regions, say $R_1, R_2, \ldots, R_K$, as shown in Fig. 4(b). We denote the average intensity in R_k to be $\bar{I}_k$, $k = 1, 2, \ldots, K$. We group these K regions into two clusters according to their average intensities: all R_k that satisfy $\bar{I}_k > \bar{I}$ constitute one cluster and the remaining constitute the other cluster, where

$$\bar{I} = \frac{\sum_{k=1}^{K} \bar{I}_k}{K}.$$

We build the binary feature map based on these two clusters, as shown in Fig. 4. In this paper, we use the method developed by Nock and Nielsen [16, 17] for the above region merging.
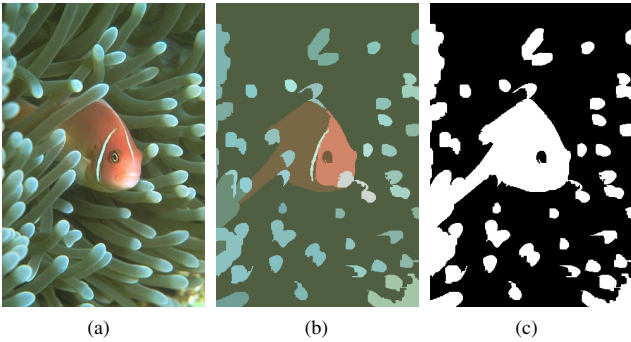

Figure 4. An illustration of constructing the binary feature map: (a) input image, (b) region merging result, (c) the resulting binary feature map.

Certainly, for different applications, we may need to choose different image features and use different methods to construct the binary feature map. For example, if we know the desirable structure contains a certain texture, we may apply a texture-analysis step to divide the image into regions with and without such texture and then construct the binary feature map based on these regions.

6. Multiple Region Segmentation

In this section we introduce an extension to repeat the proposed method for segmenting an image into multiple regions. In this extension, we first run the proposed method as described earlier to obtain an optimal boundary $\mathcal{B}_0$. This boundary, whether open or closed, always partitions the image into two disjoint regions R_1 and R_2. We can repeat the proposed method on each of these two regions to further partition them. This process can be repeated to construct an image hierarchy.

If $\mathcal{B}_0$ is an open boundary, as shown in Fig. 5(a), both regions R_1 and R_2 are bounded by a single perimeter. In this case, this proposed method can be applied to one of them, say R_1, after the following steps:

1. Prune all detected/gap-filling segments that are not fully located in R_1.

2. Construct a binary feature map M_1 by using the same algorithm as described in Section 5 on the sub-image in R_1, as shown in Fig. 5(b) by the black and white regions (the gray region with value of zero is not part of M_1). Note that, M_1 is of the same size and shape as R_1.

3. Construct the perimeter segments by dividing the perimeter of R_1 using the same algorithm described in Section 4. As illustrated in Fig. 5(c), all the segment endpoints located outside the dotted contour will be projected to the perimeter of R_1 to construct perimeter segments.

If $\mathcal{B}_0$ is a closed boundary, as shown in Fig. 5(e), one obtained region, R_2, has a single perimeter and the other, R_1, has two perimeters. Further segmentation of R_2 can be achieved by following the above steps. However, to segment region R_1, we need to make an extension in the step of perimeter-segment construction. As shown in Fig. 5(g), all the segment endpoints inside the inner dotted contour will be projected to $\mathcal{B}_0$, the inner perimeter of R_1, to construct perimeter segments on $\mathcal{B}_0$. All the segment endpoints outside the outer dotted contour will be projected to the outer perimeter of R_1 for constructing perimeter segments.

7. Experiments

We conduct experiments on a set of real images [15], with a comparison to a previous edge grouping method [22], which does not consider a feature map. We expect this comparison to justify the usefulness of the region-based feature map in edge grouping and that the proposed method supports both closed and open boundary detection.

To obtain the initial set of edges from the input image, we use the detector developed by Martin, Fowlkes and Malik [13, 14], leaving the parameters at their default values. From the obtained edges we construct the detected segments using a line-approximation algorithm, implemented by Kovesi [10]. We set the minimum length of an edge to 20 pixels and the maximum deviation from a line to 2 pixels in line approximation. We construct the binary feature maps from color images as described in Section 5. In all our experiments we set the gap between consecutive perimeter segments to 0.1. We run the comparison edge grouping
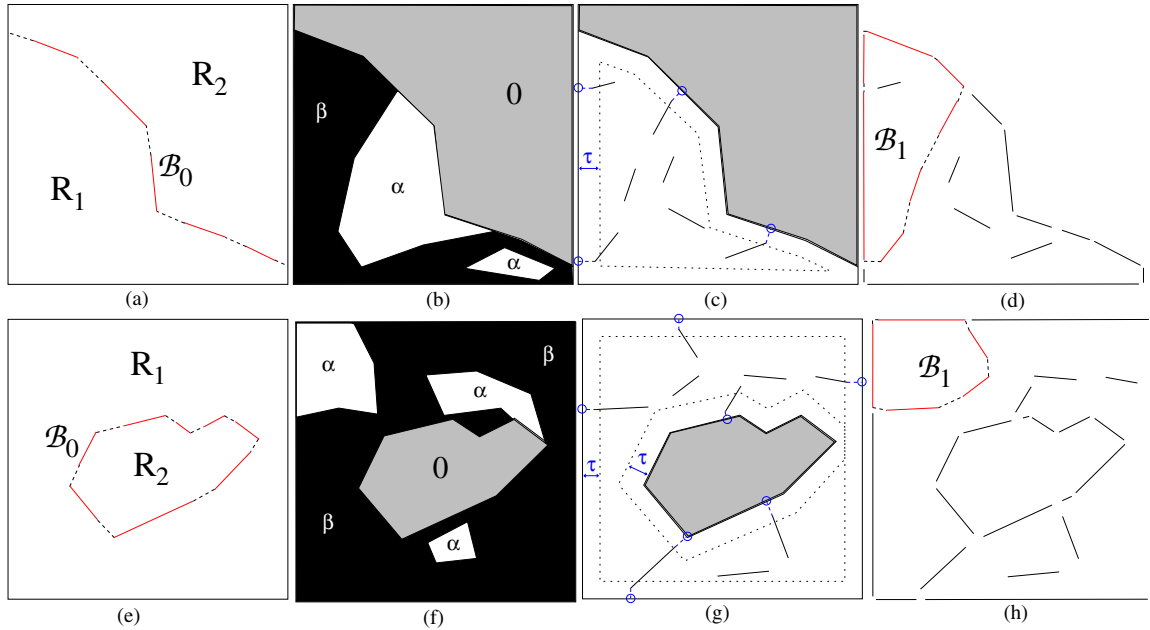
Figure 5. An illustration of multiple region segmentation. (a) Detecting an open boundary $\mathcal{B}_0$. (b) Binary feature map M_1 for region R_1 shown in (a). (c) Constructing perimeter segments for R_1 with a single perimeter. (d) An open boundary detected from R_1 shown in (a). (e) Detecting a closed boundary $\mathcal{B}_0$. (f) Binary feature map M_1 for region R_1 shown in (e). (g) Constructing perimeter segments for R_1 with two perimeters. (h) An open boundary detected from R_1 shown in (e).

algorithm [22] on the same detected segments as used for testing the proposed method.

Figure 6 shows the results obtained by using the proposed method and the comparison method on 10 real images. We can see that, by incorporating a region-based feature map, the proposed method can detect the structure boundaries in a more robust way. For example, in Fig. 6(e), the proposed method detects the profile of a horse by excluding all undesirable background, because of its consideration of the feature map. In the images shown in Figs. 6(a,d,f,g,i), the proposed method detects an open boundary, which is more accurately aligned with the salient structures in these images. Figure 6(j) presents a result where the structure detected may be considered part of the background, which can occur as discussed in section 3. There were no significant differences between the methods in terms of CPU time.

We also conducted experiments to repeat the proposed method for multiple region segmentation, as discussed in Section 6. Figure 7 shows the multiple region segmentation results on four real images. The optimal boundary detected from the original image is shown in red, which partitions the image into two regions. The optimal boundary detected from one of these two regions is shown in blue. Note that, both the boundary detected in the original image or the one detected in a region, can be either closed or open. For example, for the image shown in Fig. 7(a), we first detect an open boundary and then detect a closed boundary. For the

image shown in Fig. 7(b) and (c), we first detect a closed boundary and then detect an open boundary. These results show the potential of the proposed method to segment an image into multiple regions due to its capability for producing both closed and open boundaries.

8. Conclusion

In this paper we introduced a new edge grouping method that combines boundary and region information for salient boundary detection. In particular, region information is incorporated as a binary feature map that roughly divides the structure of interest from the background. The proposed method can then detect salient structures with the desired region features. By choosing both positive and negative feature values and constructing perimeter segments, the proposed method supports both closed and open boundary detection. This way, we can repeat the proposed method to segment the image into multiple regions in a hierarchical way. Experiments were conducted on a set of real images, with a comparison to a previous method that does not consider region information in the form of a binary feature map. We found that the proposed method performs more favorably than this comparison method.

Acknowledgments

	Input	Detected Lines	Feature Map	Proposed Method	Previous Method
(a)					
(b)					
(c)					
(d)					
(e)					
(f)					
(g)					
(h)					
(i)					
(j)					

Figure 6. Results on 10 real images. From left to right, each column shows the original image, the segments produced by the edge detection and line approximation, the binary feature map, the result obtained by the proposed method, and the result obtained by a comparison method [22].
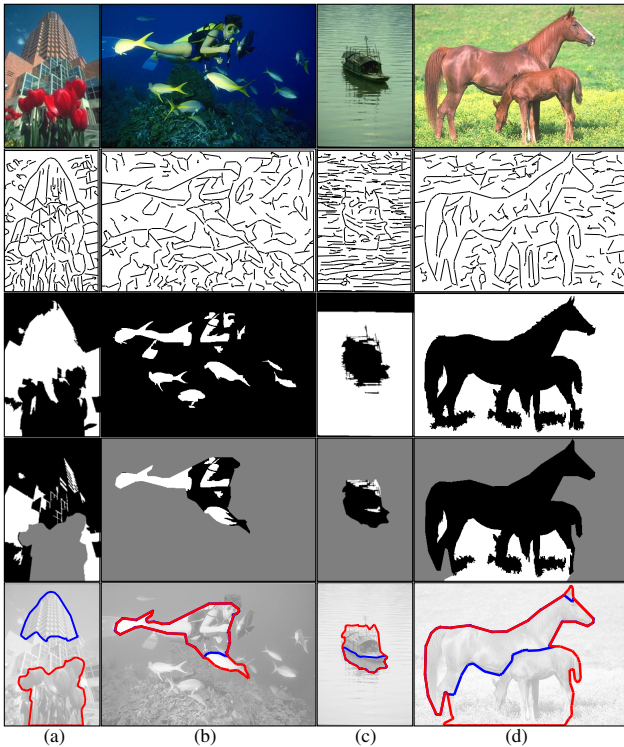
Figure 7. Multiple region segmentation on 4 real images. From top to bottom are the original image, detected line segments, feature map for the original image, feature map for one region, and boundaries detected from the original image (shown in red) and the region (shown in blue).

References

[1] T. Alter and R. Basri. Extracting salient contours from images: An analysis of the saliency network. In *International Journal of Computer Vision*, pages 51–69, 1998.

[2] A. Amir and M. Lindenbaum. A generic grouping algorithm and its quantitative analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2):168–185, 1998.

[3] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[4] J. H. Elder, A. Krupnik, and L. A. Johnston. Contour grouping with prior models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):661–674, 2003.

[5] J. H. Elder and S. W. Zucker. Computing contour closure. In *European Conference on Computer Vision*, pages 399–412, 1996.

[6] G. Guy and G. Medioni. Inferring global perceptual contours from local features. *International Journal of Computer Vision*, 20(1):113–133, 1996.

[7] D. Huttenlocher and P. Wayner. Finding convex edge groupings in an image. *International Journal of Computer Vision*, 8(1):7–29, 1992.

[8] D. Jacobs. Robust and efficient detection of convex groups. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(1):23–27, 1996.

[9] I. H. Jermyn and H. Ishikawa. Globally optimal regions and boundaries as minimum ratio cycles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10):1075–1088, 2001.

[10] P. D. Kovesi. Matlab functions for computer vision and image analysis. School of Computer Science & Software Engineering, The University of Western Australia. http://www.csse.uwa.edu.au/∼pk/research/matlabfns/.

[11] D. G. Lowe. *Perceptual Organization and Visual Recognition*. Boston: Kluwer Academic Publishers, 1985.

[12] S. Mahamud, L. R. Williams, K. K. Thornber, and K. Xu. Segmentation of multiple salient closed contours from real images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(4):433–444, 2003.

[13] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using brightness and texture. In *Neural Information Processing Systems Conference*, pages 1255–1262, December 2002.

[14] D. Martin, C. Fowlkes, and J. Malik. 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.

[15] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *IEEE International Conference on Computer Vision*, volume 2, pages 416–423, July 2001.

[16] F. Nielsen and R. Nock. On region merging: The statistical soundness of fast sorting, with applications. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 19–26, 2003.

[17] R. Nock and F. Nielsen. Statistical region merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1452–1458, 2004.

[18] S. Sarkar and K. Boyer. Quantitative measures of change based on feature organization: Eigenvalues and eigenvectors. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 478–483, 1996.

[19] S. Sarkar and K. L. Boyer. *Computing Perceptual Organization in Computer Vision*. Singapor: World Scientific, 1994.

[20] A. Shashua and S. Ullman. Structural saliency: The detection of globally salient structures using a locally connected network. In *IEEE International Conference on Computer Vision*, pages 321–327, 1988.

[21] J. S. Stahl and S. Wang. Convex grouping combining boundary and region information. In *IEEE International Conference on Computer Vision*, volume 2, pages 946–953, 2005.

[22] J. S. Stahl and S. Wang. Edge grouping combining boundary and region information. *IEEE Transactions on Image Processing*, 16(10):2590–2606, 2007.

[23] S. Wang, T. Kubota, J. Siskind, and J. Wang. Salient closed boundary extraction with ratio contour. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):546–561, 2005.

[24] L. Williams and K. K. Thornber. A comparison measures for detecting natural shapes in cluttered background. *International Journal of Computer Vision*, 34(2/3):81–96, 1999.