Edge Grouping Combining Boundary and Region Information

Joachim S. Stahl and Song Wang

Abstract

This paper introduces a new edge grouping method that combines boundary and region information to detect perceptually salient structures in noisy images. Particularly, we define a grouping cost in a ratio form, where the numerator is a measure of boundary proximity of the resulting structure and the denominator is a measure of the area of the resulting structure. This area term in fact introduces a preference towards detecting larger-size structures and therefore, makes the resulting edge grouping more robust to image noise. To find the optimal grouping with the minimum grouping cost, we develop a special graph model with two different kinds of linking edges and then reduce the grouping problem to a problem of finding a special kind of cycle in this graph with a minimum cost in the ratio form. We finally show that such an optimal cycle-finding problem can be solved in polynomial time by a graph algorithm. We implement this edge grouping method, test it on both synthetic data and some real images, and compare its performance against the previous ratio-contour method that does not consider region information. Furthermore, we discuss several extensions of the proposed method, including the incorporation of the well-known grouping cues of continuity and intensity homogeneity, introducing a factor to balance the contributions from the boundary and region information, and the prevention of detecting self-intersecting structural boundaries.

Index Terms

Perceptual organization, edge grouping, edge linking, boundary detection, graph models.

I. INTRODUCTION

Grouping (or *perceptual organization*) is an important problem in computer vision and image processing that seeks to identify some perceptually salient structures in noisy images. It is usually achieved by first

The authors are with Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208. Email: stahlj@cse.sc.edu, songwang@cse.sc.edu. Tel: 803-777-2487. Fax: 803-777-3767. Corresponding author: Song Wang.

constructing a set of tokens from the input image and then grouping a subset of these tokens into some salient structures. The grouping process is usually designed to minimize a pre-defined *grouping cost (function)* that negatively measures the perceptual saliency of the resulting structure based on some psychological vision rules, such as Gestalt laws [1]. Grouping is an important step in mid-level computer vision, which can provide with useful input to many high-level computer-vision applications such as object recognition or content-based image retrieval.

The challenge in grouping comes from both the definition of the grouping cost and the development of an algorithm for finding the optimal grouping with the minimum grouping cost. In this paper, we develop a new grouping method within the *edge grouping* framework, where the grouping tokens are a set of line segments detected from the input image and the goal is to identify a subset of these line segments and grouping them into the complete boundaries of some perceptually salient structures. Being able to more conveniently encode the well known Gestalt laws [1], edge grouping has been studied for decades with a long line of available edge-grouping methods, e.g. [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14].

Most of previous edge grouping methods only consider the boundary information in their grouping cost. For example, in almost all the previous edge-grouping methods, boundary *proximity* is always considered to make the resulting boundary contain as short gaps as possible in connecting the line segments into a boundary. However, many constructed line segments in fact come from image noise (or image texture) and therefore, considering only boundary information usually makes the grouping very sensitive to the image noise. Some boundary properties, such as *continuity*, which requires the resulting boundary to be smooth, and *convexity*, which requires the resulting boundary to be convex, may partially solve this sensitivity problem by only detecting smooth and convex structures. However, the incorporation of these properties may limit the applicability of the grouping methods since many salient structures in real applications are not always smooth or convex.

To address this problem, in this paper we develop a new edge grouping method that combines boundary and region information. In its baseline form, it combines two boundary properties of proximity and *closure* and one region property of the enclosed region area. Specifically, the grouping cost for a resulting boundary is defined to be a ratio between the total gap length along the boundary and the area enclosed by the boundary. The closure is set as a hard constraint by requiring the detected boundary to be always closed. This way, we in fact introduce a preference to detect a larger-size structures and therefore, make this grouping method more robust to the image noise. From this baseline method, we also discuss the extensions of finding better balance between the boundary and region properties and incorporating other boundary and region properties that may be desired in some special applications.

To locate the closed boundary that minimizes this ratio-form grouping cost, we first develop a new graph model where line segments and in-between gaps are modelled by two different kinds of linking edges. Particularly, we introduce two edges to represent each line segment or in-between gap so that the boundary and region information can be encoded into two edge-weight functions, respectively. Based on this graph model, we reduce the edge grouping problem to a problem of finding a special kind of cycle with minimum ratio-form cost. We find that this cycle-finding problem can be solved in polynomial time by a globally optimal graph algorithm. We implement this edge grouping method and test it on both synthetic data and some real images, and compare its performance against ratio contour, another similar edge grouping method that considers boundary continuity instead of region information.

The remainder of this paper is organized as follows. Section II formulates the problem by introducing the new grouping cost that combines boundary and region information. Section III presents the details of the graph model and the graph algorithm that are used for solving the formulated edge-grouping problem. Section IV presents the experiment results on some synthetic data and real images, with a comparison against ratio-contour method. Section V discusses several extensions to the proposed method. Section VI briefly discusses some major related work. A brief conclusion is given in Section VII.

II. Problem Formulation

As illustrated in Fig. 1, it usually involves three steps when applying an edge grouping method to a real image to detect perceptually salient structural boundaries. The first step is to construct a set of line segments by running an edge detector, such as the Canny detector [15], on the input image to detect a set of edges. We then approximate the detected edges with a set of straight line segments, as shown in Fig. 1(b). These straight line segments are usually referred to as *detected (line) segments*. Since these detected segments are disconnected from each other, in order to construct a closed boundary we need to fill the gaps between them. So in the second step, we fill the gaps between the detected segments by connecting all the possible pairs of endpoints of different detected segments. These connections are referred to as *gap-filling (line) segments*, as denoted by dashed lines in Fig. 1(c)[1]. This way, a boundary is defined as a cycle that traverses a set of detected and gap-filling segments *alternately*. The third step is to develop an algorithm to find such a boundary that minimizes a selected grouping cost, as shown in Fig. 1(d).

[1] Note that not all possible gap-filling segments are shown in Fig. 1(c), in order to keep it readable.

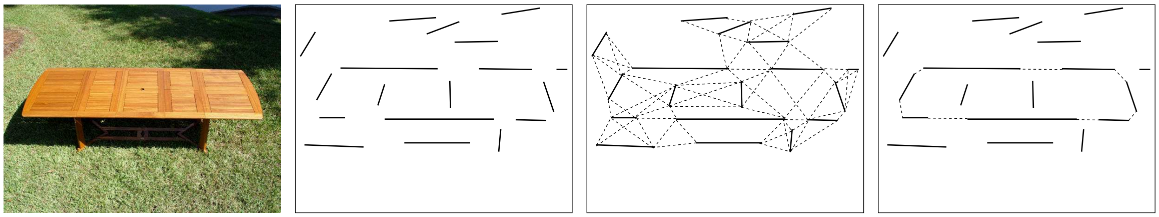


Fig. 1. The typical steps in applying an edge-grouping method to a real image to detect salient boundaries. (a) The input image, (b) the detected segments, (c) constructing gap-filling segments (dashed lines), and (d) the detected closed boundary that traverses some detected and gap-filling segments alternately.

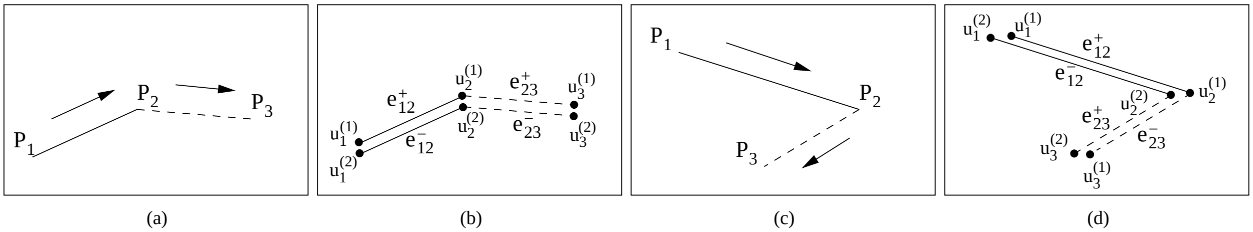
To combine the boundary and region information, in this paper, we introduce a ratio-form grouping cost for a closed boundary $\mathcal{B}$ that traverses some detected and gap-filling segments alternately as

$$\phi(\mathcal{B}) = \frac{|\mathcal{B}_G|}{\iint_{R(\mathcal{B})} dxdy}, \tag{1}$$

where the numerator $|\mathcal{B}_G|$ is the total length of the gap-filling segments along the boundary $\mathcal{B}$ and reflects the proximity of the boundary. $R(\mathcal{B})$ is the region enclosed by the boundary $\mathcal{B}$ and the denominator $\iint_{R(\mathcal{B})} dxdy$ is the region area, which sets a preference to detect larger structures. Such a preference makes the grouping more robust against image noise. In the following section, we develop a graph model and algorithm to address the formulated edge grouping problem by finding the boundary that minimizes this new grouping cost (1).

III. Graph Modelling and Algorithm

In this section, we develop a graph model and algorithm to address the problem formulated in the previous section. We begin by constructing a graph $G = (V, E)$, with a set of vertices $V = \{u_1, u_2, \ldots, u_n\}$ and a set of edges $E = \{e_1, e_2, \ldots, e_m\}$. Particularly, we construct a pair of edges e^+ and e^- for each line segment. We call the constructed pair of edges to be *solid* edges, if the corresponding line segment is a detected one, and *dashed* edges, if the corresponding line segment is a gap-filling one. This way, we actually construct two vertices, $u_i^{(1)}$ and $u_i^{(2)}$, for each line-segment endpoint. Figure 2 shows an example, where for the detected line segment P_1P_2 shown in Fig. 2(a), we construct two solid edges, e_{12}^+ and e_{12}^-, shown by solid lines in Fig. 2(b). For the gap-filling segment P_2P_3, in Fig. 2(a), we construct two dashed edges, e_{23}^+ and e_{23}^-, shown by dashed lines in Fig. 2(b), and for each line-segment endpoint P_i, $i = 1, 2, 3$, we construct two vertices, $u_i^{(1)}$ and $u_i^{(2)}$, $i = 1, 2, 3$. We will show that this construction of edges in pairs facilitates the quantization of the region area enclosed by the boundary $\mathcal{B}$.



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

Fig. 2. An illustration of the graph construction. Linking edges corresponding to two segments with (a,b) same direction, (c,d) opposite direction.

One problem in this graph construction is how to determine the edge connection relations, since each line segment is represented by a pair of edges. For example, in Figs. 2(a) or (c), the detected segment P_1P_2 is connected to the gap-filling segment P_2P_3 at P_2. In the constructed graph we need to decide whether we are going link e_{12}^{+} to e_{23}^{+} and e_{12}^{-} to e_{23}^{-}, or link e_{12}^{+} to e_{23}^{-} and e_{12}^{-} to e_{23}^{+}. In this paper, we solve this problem by associating each of the two edges in the graph, corresponding to the same line segment, with a different direction. Particularly, e^{+} indicates that the direction along the corresponding line segment is from the left endpoint to the right endpoint (LR), and e^{-} indicates that the direction along the corresponding line segment is from the right endpoint to the left endpoint (RL). For any line segment, the left endpoint is the one with the smaller x-coordinate and the right endpoint is the one with the larger x-coordinate. For example, for the line segment P_1P_2 in both Figs. 2(a) and (c), P_1 is the left endpoint and P_2 is the right endpoint. This way, we can uniquely determine the edge-connection relation by requiring consistency in direction between the two neighboring line segments. Figures 2(b) and (d) show the edge linking obtained from the line segments P_1P_2 and P_2P_3 shown in Figs. 2(a) and (c), respectively. If the x-coordinates of the two endpoints are equal, we can decide by the y-coordinates of the two endpoints in a similar fashion. Note that, the constructed graph is still an *undirected* one and we only use this direction information to define the edge-weight functions as discussed later.

In this constructed graph G, a closed boundary $\mathcal{B}$ that traverses some detected and gap-filling segments alternately, is in fact modeled by two cycles that traverse the corresponding solid and dashed edges alternately. An example is shown in Fig. 3, where the boundary $P_1P_2 \ldots P_6$ is modeled by the two cycles shown in Figs. 3(b) and (c). We can see that these two cycles are the "mirrors" of each other, i.e., for a pair of edges e^{+} and e^{-} constructed for the same line segment, if one of them is contained in one cycle , the other must be contained in the other cycle. For convenience, we call the graph G to be a *solid-dashed (SD)* graph, because no two solid edges are neighboring to each other, and a cycle that traverses solid and dashed edges alternately to be an *alternate* cycle. This way, the problem of finding the boundary $\mathcal{B}$ that minimizes the grouping cost $\phi(\mathcal{B})$ given in Eq. (1) can be reduced to the problem of finding an optimal alternate cycle $\mathcal{C}$ in the constructed SD graph G if we can quantify the grouping cost $\phi(\mathcal{B})$ by some edge weights in G.

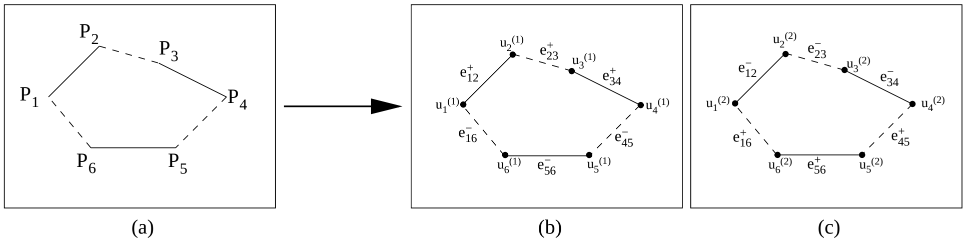


Fig. 3. (a) A boundary with three detected segments and three gap-filling segments. (b) & (c) Two "mirror" cycles in the graph G corresponding to the boundary shown in (a).

We define two edge-weight functions, the *first (edge) weight* $w_1(e)$ and the *second (edge) weight* $w_2(e)$, for each edge $e \in E$. Given any line segment P_1P_2, we set the first weight for the corresponding two edges to

$$w_1(e_{12}^+) = w_1(e_{12}^-) = \begin{cases} 0 & \text{if } P_1P_2 \text{ is a detected segment} \\ |P_1P_2| & \text{if } P_1P_2 \text{ is a gap-filling segment,} \end{cases}$$

where $|P_1P_2|$ is the length of the line segment P_1P_2. For both solid and dashed edges, their second weights are defined as the signed area associated to the corresponding line segment. As shown in Fig. 4(a), let the bottom-left pixel in the input image be the origin, the horizontal direction be the direction of the x-axis, and the vertical direction be direction of the y-axis. The area associated to a line segment P_1P_2 is defined as the area of the region bounded by this line segment and its projection in the x-axis. The sign of this area is defined to be positive for the edge corresponding to a line segment that bears a LR direction and negative otherwise. For the example shown in Fig. 4(a), we have $w_2(e_{12}^+) = -w_2(e_{12}^-) = \texttt{area}(P_1P_2P_2^xP_1^x) > 0$, where P_1^x and P_2^x are the projections of P_1 and P_2 onto the x-axis. This definition allows us to calculate the total area within a boundary by simply summing up the signed areas associated to each of its line segments. An example is shown in Fig. 4(b), where the area of the polygon $P_1...P_6$ is equal to the summation of positive areas associated to P_1P_2, P_2P_3, P_3P_4, and negative areas associated to P_4P_5, P_5P_6 and P_6P_1.

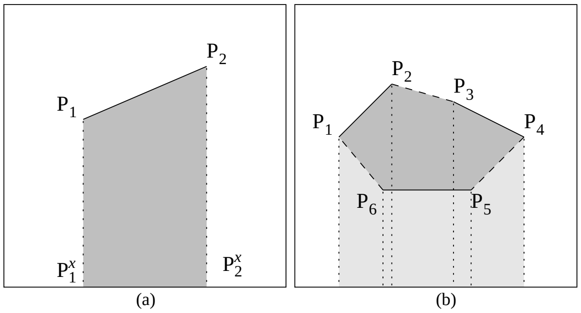


Fig. 4. An illustration of defining the second weight for an edge. (a) Area associated to a line segment. (b) The region area enclosed by a closed boundary is equal to the sum of the signed areas associated to the line segments along this boundary.

As discussed above, a closed boundary $\mathcal{B}$ corresponds to two alternate "mirror" cycles $\mathcal{C}^+$ and $\mathcal{C}^-$ in G, as shown in Figs. 3(b) and (c). Since the edges in $\mathcal{C}^+$ and $\mathcal{C}^-$ are constructed in pairs, we have

$$W_1(\mathcal{C}^+) = \sum_{e \in \mathcal{C}^+} w_1(e) = W_1(\mathcal{C}^-) = \sum_{e \in \mathcal{C}^-} w_1(e) > 0$$

$$\text{and} \quad W_2(\mathcal{C}^+) = \sum_{e \in \mathcal{C}^+} w_2(e) = -W_2(\mathcal{C}^-) = -\sum_{e \in \mathcal{C}^-} w_2(e).$$

Without loss of generality, let the cycle $\mathcal{C}^+$ be the one with the positive total second weight, i.e., $W_2(\mathcal{C}^+) = -W_2(\mathcal{C}^-) > 0$. It is easy to verify that $W_1(\mathcal{C}^+)$ is equal to the numerator of $\phi(\mathcal{B})$ and $W_2(\mathcal{C}^+)$ is equal to the total area of the enclosed region, i.e., the denominator of $\phi(\mathcal{B})$. Since for every cycle $\mathcal{C}^+$ there exists a "mirror" cycle $\mathcal{C}^-$ in G, it is easy to see that the cycle $\mathcal{C}$ that minimizes

$$\varphi(\mathcal{C}) = \frac{W_2(\mathcal{C})}{W_1(\mathcal{C})}, \tag{2}$$

is a $\mathcal{C}^-$ version (i.e., $W_2(\mathcal{C}) < 0$) that corresponds to the boundary $\mathcal{B}$ that minimizes $\phi(\mathcal{B})$, i.e. $\phi(\mathcal{B}) = -\frac{1}{\varphi(\mathcal{C})}$. This way, we only need to find an alternate $\mathcal{C} \in G$ that minimizes the cycle ratio $\varphi(\mathcal{C})$. This problem can be solved in polynomial time by the minimum-ratio-alternate-cycle algorithm presented in [13].

IV. Experiments

We implement the above graph model and algorithm in C++ and evaluate the proposed edge-grouping method on a set of synthetic data and real images[2]. The synthetic data was directly generated as a set of detected line segments. For the real images, we construct the detected segments by edge detection

[2] The software developed in this work can be downloaded from http://www.cse.sc.edu/∼songwang/document/RRC.tgz.

and line approximation. Particularly, we use the Canny edge detector from the Matlab image processing toolbox, and the line approximation package developed by Kovesi [16]. For the Matlab Canny edge detector, we leave the parameters at their default values, and for the line-approximation package, we set the minimum edge length to be processed to 30 pixels, and the maximum deviation between an edge and its fitted line segment to 2 pixels.

A. Experiments on Synthetic Data

To evaluate the proposed edge-grouping method quantitatively, we construct a set of synthetic data with known desirable salient structural boundaries, or *ground truth*. We measure the accuracy of an edge-grouping result by comparing with the ground truth, using Jaccard's similarity coefficient, $\frac{|R_D \cap R_G|}{|R_D \cup R_G|}$, where R_D and R_G are the region bounded by the optimal boundary detected by the proposed edge-grouping method and the one bounded by the ground-truth boundary, respectively, and $|R|$ is the area of R. Based on this accuracy measure, we also compare the performance of the proposed method against the ratio-contour method [13], which considers only boundary properties of proximity and continuity in its grouping cost

$$\phi_r(\mathcal{B}) = \frac{|\mathcal{B}_G| + \lambda \cdot \int_{\mathcal{B}} \kappa^2(t)dt}{\int_{\mathcal{B}} dt},$$

where $\kappa(t)$ is the curvature and t is the parameter of the arc-length parameterized boundary $\mathcal{B}$. This grouping cost is also of a ratio form, with a denominator of the boundary perimeter, which helps improve the grouping robustness against image noise by avoiding producing overly short boundaries. Note that in this ratio-contour method, the gap-filling tokens are not constructed as straight line segments, but instead approximated as Bezier curves, which connect the detected segments with continuous tangent directions and therefore, allow the measuring of curvature along the resulting boundary. In our experiments we set the parameter λ at its default value of 10 [13].

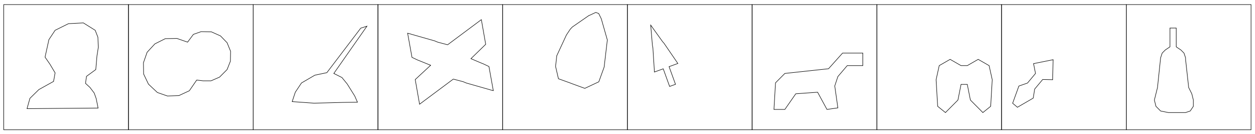


Fig. 5. 10 polygonal closed boundaries used for the construction of synthetic data.

For constructing a synthetic data sample, we pick one of the 10 polygonal boundaries shown in Fig. 5 as the ground truth, placed inside a square region of size 128×128. Then we remove a certain percentage of segments along this ground truth boundary at random locations to construct some gaps and the remaining segments then are included as detected line segments. The gap percentage along the ground truth boundary is chosen from the set $\{0\%, 5\%, 10\%, 20\%, 30\%, 40\%, 50\%\}$. We then construct a set of additional detected line segments to simulate the image noise. Specifically, these noise segments are placed at random locations (inside the 128×128 square region), in random directions, and with a length selected randomly between 3 and 7 pixels (all properties uniformly distributed). The number of added noise segments is chosen from the set $\{0, 10, 20, 40, 80\}$. An example is shown in Fig. 6, where the ground truth is chosen to be the $4th$ boundary in Fig. 5. Fig. 6(c) shows a constructed synthetic data sample by removing 30% of the ground-truth boundary's perimeter and then adding 40 noise segments. To remove p percent of the ground-truth boundary, we uniformly partition the boundary into 20 line segments, and then removing p percent of these line segments randomly. Figures 6(d) and (e) show the grouping results of running the ratio-contour method and the proposed method, respectively. These results represent a grouping accuracy of 0.51 and 0.99 respectively.

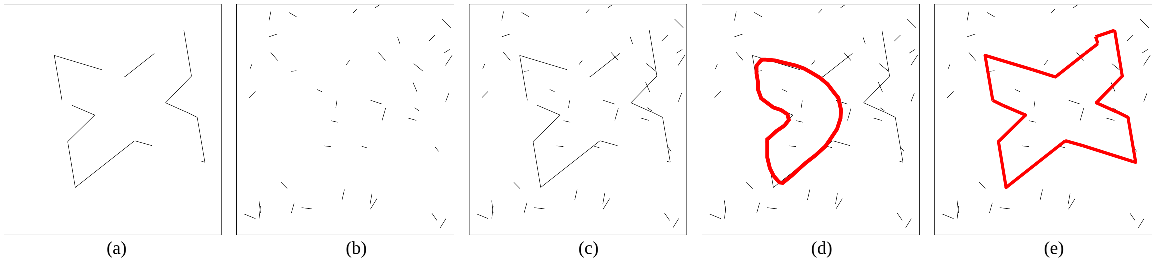
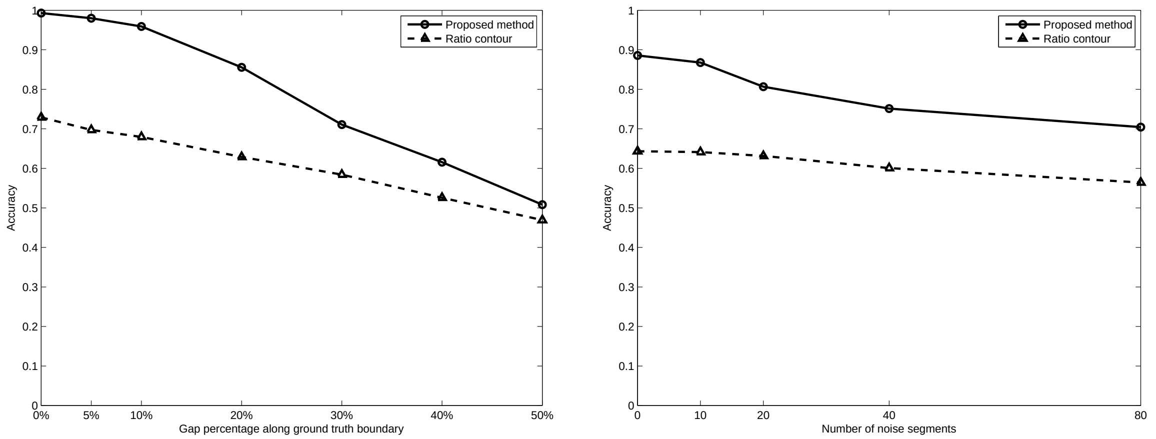


Fig. 6. An illustration of the synthetic data construction and the grouping results. (a) Detected segments constructed from the ground truth boundary. (b) Additional noise segments. (c) A constructed synthetic data sample by combining the segments shown in (a) and (b). (d) Optimal boundary detected from (c) by applying the ratio-contour method developed in [13]. (e) Optimal boundary detected from (c) by applying the proposed edge grouping method.

As mentioned above, we have 10 different choices of the ground-truth boundaries, 7 different choices of the gap percentage along the ground truth boundary, and 5 different choices of the number of additional noise segments. For each possible combination of these choices, we also run the random sampling for noise segments 10 times to achieve 10 different sets of noise segments. Therefore, in total we construct $10 \times 10 \times 7 \times 5 = 3500$ synthetic data samples. We run the edge grouping methods on each of them and evaluate the grouping accuracy by comparing the detected optimal boundary with the ground truth. Figure 7 shows the performance curves of the proposed edge-grouping method and the method developed in [13]. The performance in Fig. 7(a) is shown in terms of the gap percentage along the ground truth boundary and each point in this figure indicates the average accuracy over $10 \times 10 \times 5 = 500$ data samples with the same gap percentage along the ground-truth boundary. The performance in Fig. 7(b) is shown in terms of the number of the additional noise segments and each point in this figure indicates the average accuracy over $10 \times 10 \times 7 = 700$ data samples with the same number of additional noise segments. These curves clearly show that the inclusion of the region-area information in the proposed method tends to produce better accuracy than the ratio-contour method. The main reason is that the ratio-contour method explicitly incorporates the continuity property to improve its robustness against noise, but many boundaries in the real world are not necessarily smooth everywhere. For example, as shown in Fig. 6(d), the continuity preference may prevent the ratio-contour method to correctly detect the desirable ground-truth boundary completely.



(a)

(b)

Fig. 7. Performance of the proposed method and the ratio-contour method on the 3500 synthetic data samples.

B. Experiments on Real Images

We also test the proposed edge-grouping method on a set of real images selected from the Berkeley segmentation dataset [17]. All real images have a size of either 481×321 or 321×481. In order to reduce the number of the constructed gap-filling segments, which can run in the order of n^2 where n is the number of detected segments, we do not construct gap-filling segments that are highly unlikely to belong to the optimal salient boundary. Specifically, in our experiments, we do not construct a gap-filling segment between the two segment endpoints if the distance between these two endpoints is larger than 50 pixels.

The results on 20 sample images are shown in Figs. 8 and 9. We can see that, by considering the region information, the proposed method detects more desirable boundaries in most images, as shown in Figs. 8(a, c-h, j) and 9(b-j). There are also some cases where both methods produce similar results, as shown in Figs. 8(b), and cases where results are different yet both acceptable, as shown in Fig. 8(i). Note that, the incorporation of the region-area term in the proposed method does not mean that the proposed method always produces a boundary that encloses larger area than the one produced by the ratio-contour method. For example, Figure 9(h) shows a case where the proposed method detects a smaller structure than the ratio-contour method. In this case, the proximity may play a more dominating role than the region-area term, but may not overplay the continuity used in the ratio-contour method. Table I gives the number of detected segments and the CPU time taken by the proposed method for processing these 20 real images.

Image	Fig.8(a)	Fig.8(b)	Fig.8(c)	Fig.8(d)	Fig.8(e)	Fig.8(f)	Fig.8(g)	Fig.8(h)	Fig.8(i)	Fig.8(j)
# Detected segments	562	382	784	525	596	482	451	406	502	656
CPU time (s)	50.20	3.94	161.39	26.00	45.01	25.92	25.86	13.94	28.07	69.00

Image	Fig.9(a)	Fig.9(b)	Fig.9(c)	Fig.9(d)	Fig.9(e)	Fig.9(f)	Fig.9(g)	Fig.9(h)	Fig.9(i)	Fig.9(j)
# Detected segments	1208	679	594	654	390	413	557	235	390	457
CPU time (s)	858.48	98.47	44.18	65.92	9.62	30.97	120.19	3.63	40.24	68.10

TABLE I

THE NUMBER OF DETECTED LINE SEGMENTS AND THE CPU TIME (IN SECONDS) TAKEN BY THE PROPOSED METHOD IN THE EXPERIMENTS SHOWN IN FIGS. 8 AND 9.

C. Multiple Boundary Detection

So far we present the proposed method in the context of only detecting the boundary that minimizes the grouping cost (1). In fact, it is easy to extend the proposed method to detect multiple boundaries in an image by iterating the method. The basic principle has been used widely in many previous edge-grouping methods, including the ratio-contour method. Given an image, e.g. the one shown in Fig. 10, we first process it with the proposed method to detect the optimal boundary as introduced above. Then we remove from the graph G all the edges associated to line segments that belong to the detected optimal boundary, i.e. for each line segment P_iP_j in the boundary we remove both edges e_{ij}^{+} and e_{ij}^{-} from G. We then run the minimum-ratio-alternate-cycle algorithm again on G, to detect the second optimal boundary. This

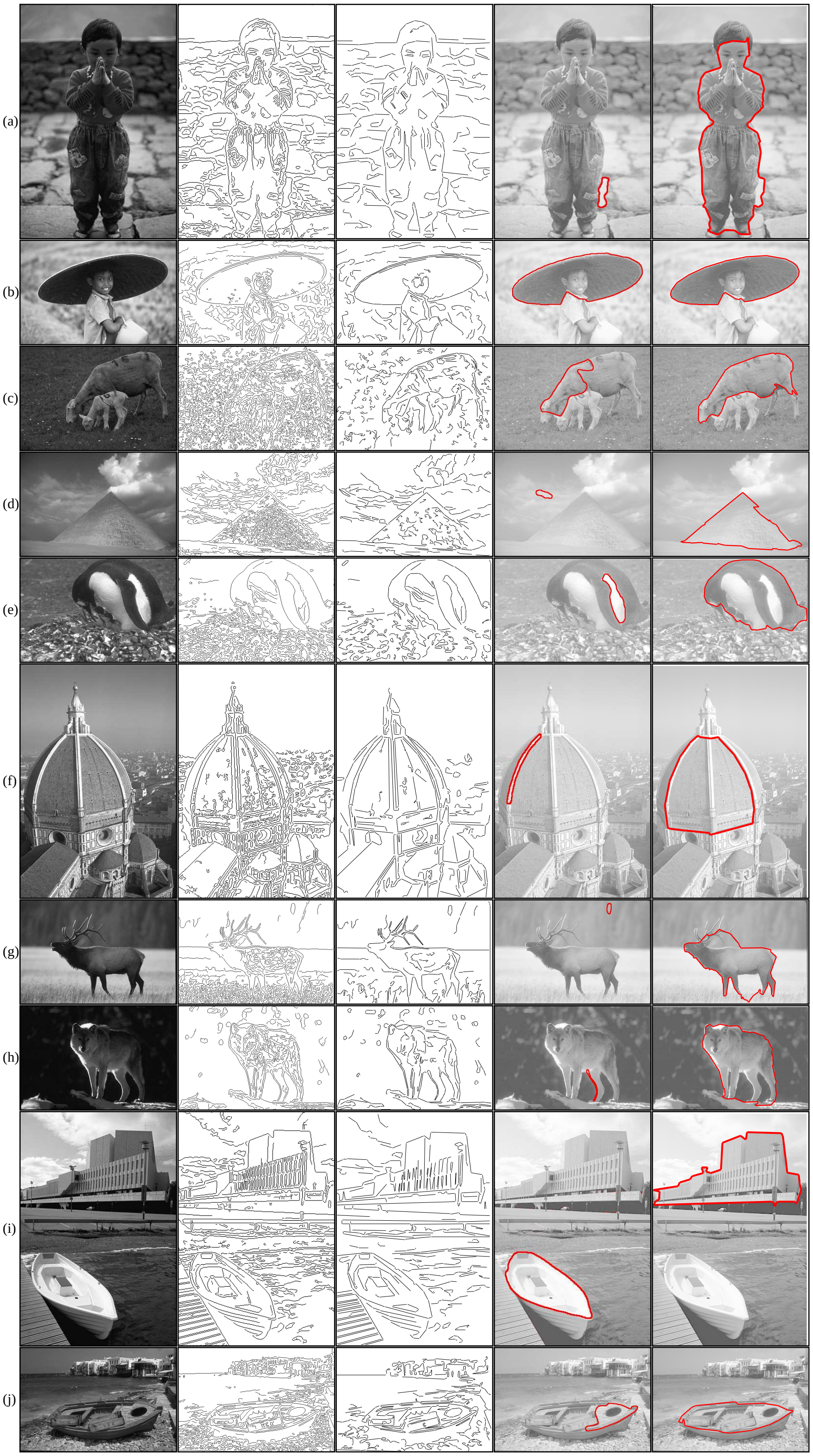


Fig. 8. Edge grouping results on 10 real images. From left to right, Column 1: the input image; Column 2: the Canny detection result; Column 3: the detected segments resulting from line approximation; Column 4: the optimal boundary detected by the ratio-contour method; Column 5: the optimal boundary detected by the proposed method.

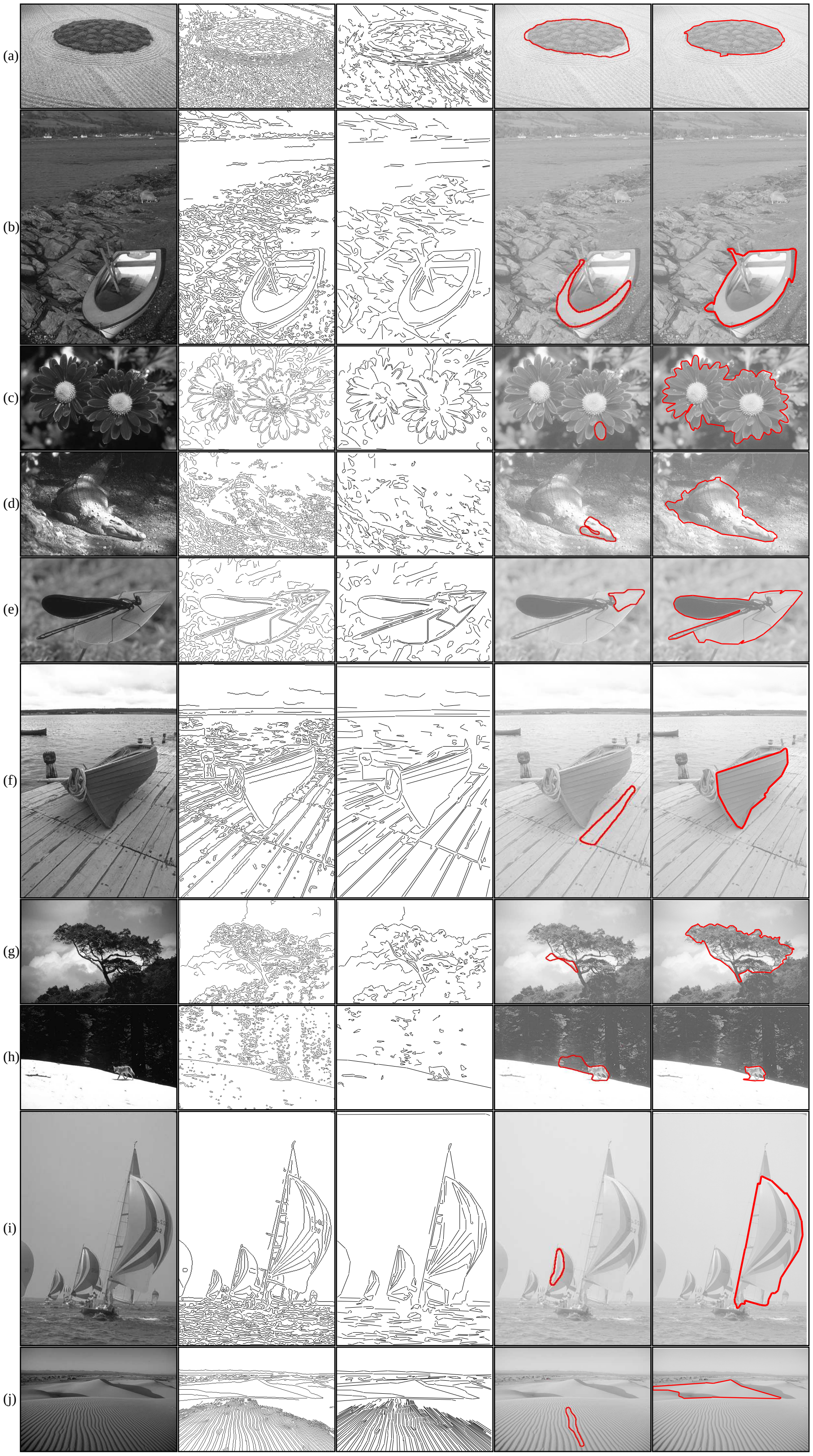


Fig. 9. Edge grouping results on another 10 real images. Each column depicts the same information as in Fig. 8.

process can be repeated k times to detect the kth optimal boundary. It is easy to show that the grouping cost for the detected boundaries increases monotonically in the iteration process. An example is shown in Fig. 10, where the multiple salient structures can be detected by iterating the proposed edge grouping method.

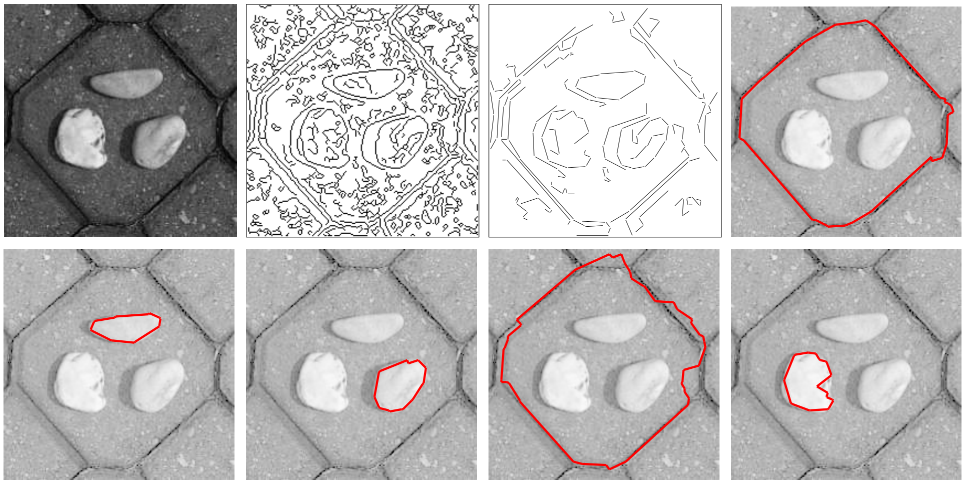


Fig. 10. An example of detecting multiple boundaries from a real image by repeating the proposed edge-grouping method. On the first row, from left to right, shows the input image, the Canny edge detection result, the line approximation result (detected segments), and the 1st optimal boundary. On the second row, from left to right shows the 2nd through 5th optimal boundaries obtained by iterating the proposed method.

V. Extensions

In this section we introduce four extensions to the proposed method. The first two can be useful to exploit possible prior knowledge about the desirable salient structures, by adding the properties of continuity and intensity homogeneity to the grouping cost. The third extension seeks to adjust the balance between proximity and region-area terms in the grouping cost, since we find that, in certain cases, the region-area term might have undesirable dominance in grouping. The fourth extension is to ensure that detected salient boundaries are always simple without containing any self intersections.

A. Adding Continuity

In the previous section, we show that, in general, the use of the region-area property in the proposed method leads to more favorable grouping than the use of continuity alone in the ratio-contour method. However, there are certain cases where the desirable salient structure in an image is *a priori* known to be smooth. In this section, we show that we can extend the proposed method to include continuity as part of its grouping cost. This way, this extended edge grouping method in fact combine both proximity, continuity and region-area properties.

In this extension, we first modify the grouping cost to

$$\phi_c(\mathcal{B}) = \frac{|\mathcal{B}_G| + \lambda \cdot \int_{\mathcal{B}} \kappa^2(t)dt}{\iint_{R(\mathcal{B})} dxdy},$$

where $\kappa^2(t)$ is the squared curvature along the arc-length parameterized boundary $\mathcal{B}$, and as in the ratio-contour method [13], λ is a regularization factor that balance the proximity and continuity. In our experiments, we consistently set λ to be 10. The additional curvature term in this extension makes the resulting edge grouping more biased to detect smoother boundaries.

However, it is difficult to directly measure the boundary curvature in our formulation since the boundary $\mathcal{B}$ is a polygon consisting of a set of straight line segments. We address this problem by interpolating the polygon by smooth cubic splines. Particularly, given a gap-filling segment P_2P_3 that connects detected segments P_1P_2 and P_3P_4, we measure its curvature κ over $\mathcal{Z}(M_1P_2P_3M_2)$, the Bezier curve with control points M_1, P_2, P_3 and M_2, as shown in Fig. 11. Here M_1 and M_2 are the midpoints of P_1P_2 and P_3P_4 respectively. We can then calculate the curvature along this Bezier curve and use it to evaluate the continuity of a boundary. In the graph modeling, we only need to modify the definition of the first edge weight to incorporate this curvature term. Specifically, for the dashed edges e_{23}^+ and e_{23}^- corresponding to the gap-filling segment P_2P_3 shown in Fig. 11, we define their first edge weight as

$$w_1(e_{23}^+) = w_1(e_{23}^-) = |P_2P_3| + \lambda \cdot \int_{\mathcal{Z}(M_1P_2P_3M_2)} \kappa^2(t)dt.$$

With this modification, the graph G remains essentially the same and we can still apply the same graph algorithm to detect the optimal boundary that minimizes this modified grouping cost.

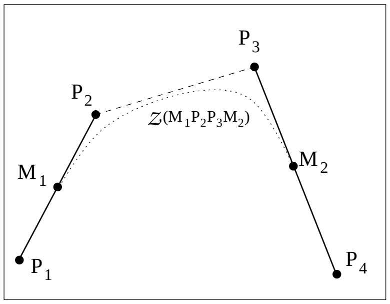


Fig. 11. An illustration of using Bezier curve to approximate the gap-filling and detected segments.

Note that this type of Bezier-curve interpolation may not reflect the boundary continuity accurately when the angles $\angle P_1P_2P_3$ and $\angle P_2P_3P_4$ become too small. However, when any one of these two angles becomes too small, this gap-filling segment P_2P_3 is not likely to be included in a smooth boundary. Therefore, in practice, we additionally impose the following constraint: we do not construct a gap-filling segment P_2P_3 between detected segments P_1P_2 and P_3P_4, if either of the angles $\angle P_1P_2P_3$ and $\angle P_2P_3P_4$ is smaller than a given threshold. In our experiments we set this threshold to be $\pi/2$. Figure 12 demonstrates several examples of applying this extended edge-grouping method. For comparison, we also include the grouping results from the proposed method without the extension of adding the continuity. These results show that this extension may produce more favorable grouping results when the desirable structural boundary is relatively smooth.

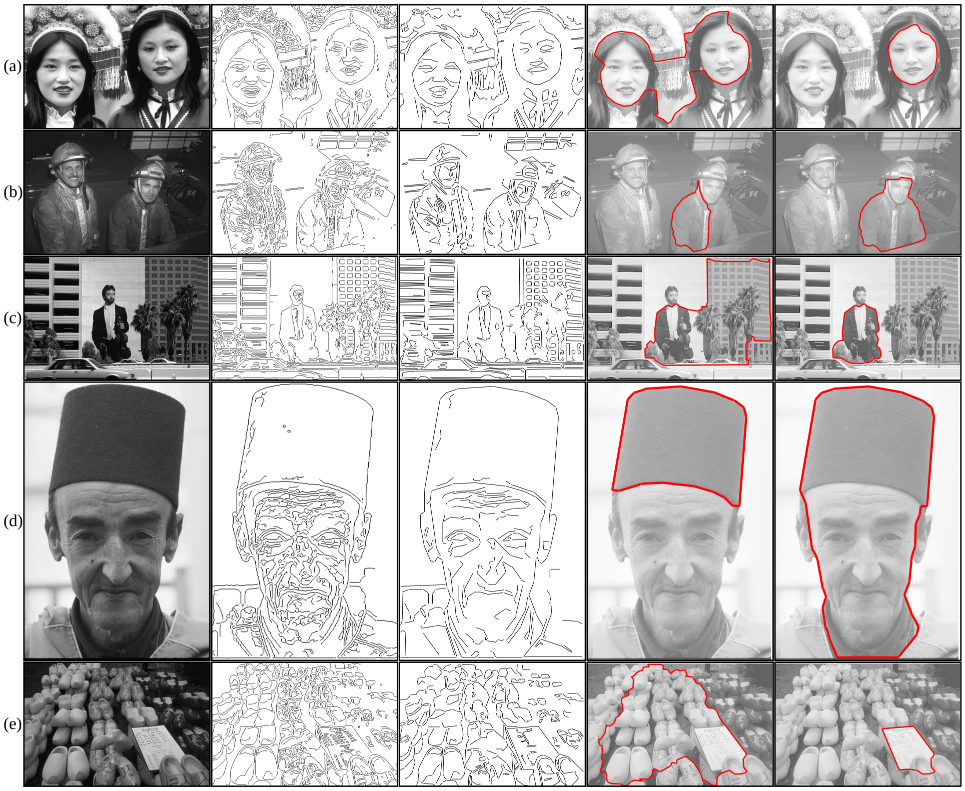


Fig. 12. Sample grouping results of the proposed method with and without the extension of adding the boundary continuity. Each row shows, from left to right, the input image, the Canny edge detection result, the line-approximation result, the optimal boundary detected by the proposed method without adding the continuity, and the optimal boundary detected by the proposed method extended with continuity.

B. Incorporating Intensity Homogeneity

In many real images, the desirable salient structure shows largely homogeneous intensity, while the intensity across the structural boundary changes abruptly. In this section, we extend the proposed method to incorporate such an intensity-homogeneity property to help detect the desirable boundaries.

To incorporate the intensity homogeneity, we modify the grouping cost (1) to

$$\phi_h(\mathcal{B}) = \frac{|\mathcal{B}_G|}{\iint_{R(\mathcal{B})} \sigma_\epsilon^T(x,y)dxdy}, \tag{3}$$

where $\sigma_\epsilon^T(x,y)$ is a function defined by

$$\sigma_\epsilon^T(x,y) = \begin{cases} 1 & \text{if } |I(x,y) - T| < \epsilon \\ 0 & \text{if } |I(x,y) - T| > \epsilon. \end{cases}$$

Here $I(x,y)$ is the image intensity of the pixel at (x,y). T is a specified pixel intensity for the desirable structure enclosed by the detected boundary. T can be either user specified or automatically selected by some histogram analysis. $\epsilon \geq 0$ is the expected pixel-intensity variation within the region enclosed by the detected boundary. In essence, this new grouping cost only counts the pixels with an intensity in $[T - \epsilon, T + \epsilon]$ in calculating the enclosed region area, and therefore favoring in detecting a boundary that encloses a region with intensity as close to T as possible. The smaller the value of ϵ, the more homogeneous we expect the region enclosed by the detected boundary.

With this new grouping cost, the graph G remains the same. The only difference is to slightly modify the definition of the second edge weight w_2 to count only the pixels with an intensity in $[T - \epsilon, T + \epsilon]$ in calculating the enclosed region area. Therefore, we can still apply the proposed graph algorithm to detect the optimal boundary that minimizes this new grouping cost. Figure 13 shows some experiment results on some real images by applying this extended edge-grouping method. We can see that the proposed extension of incorporating intensity homogeneity can improve the grouping results when we have some *a priori* knowledge on the intensity of desirable structure. For example, in Fig. 13(e), we set a smaller value for T and detect the bird while the original edge grouping method without this extension detects the chunk of the tree, which shows larger intensity.

Another interesting extension is to replace $\sigma_\epsilon^T(x,y)$ by $(1 - \sigma_\epsilon^T(x,y))$ in the grouping cost (3). This makes the edge grouping to detect a boundary that encloses a region with intensity not close to T. Some experiment results are shown in Fig. 14. We can see that this extension may also help improve the grouping results when it is *a prior* known that the desirable salient structure do not show certain intensity. Note that, the proposed edge grouping can only produce a single closed boundary in one iteration. Therefore,

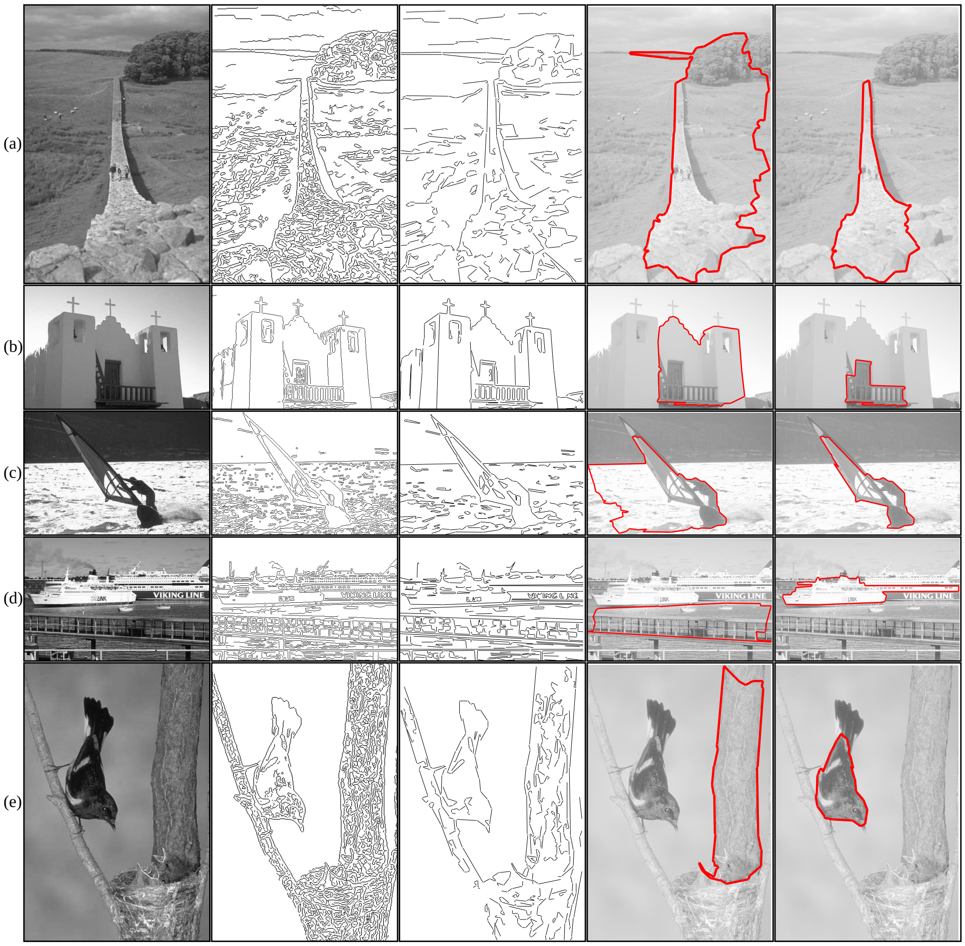


Fig. 13. Edge grouping results with the extension of incorporating intensity homogeneity. Each row shows, from left to right, the input image, the Canny edge-detection result, the line-approximation result, the optimal boundary detected by the proposed method without any extension, and the optimal boundary detected by the proposed method with the extension of adding intensity homogeneity. ϵ is set to 50 for all images. T is set to 165, 50, 70, 230, and 50 for images (a-e), respectively.

with two disjoint salient structures with same intensity, such as the two eyes shown in Fig. 14(d), we have no way to detect both of them in one iteration. Instead, we need to apply the multiple boundary detection strategy introduced in Section IV-C to detect them sequentially.

C. Proximity Exponentiation

The grouping cost (1) is simply a ratio between the total gap length along the boundary and enclosed-region area. While we have shown that this grouping cost usually leads to good grouping results in many images, there are cases where the region-area term dominates the grouping cost prompting the proposed method to detect an overly large region that does not align well with any salient structural boundaries.

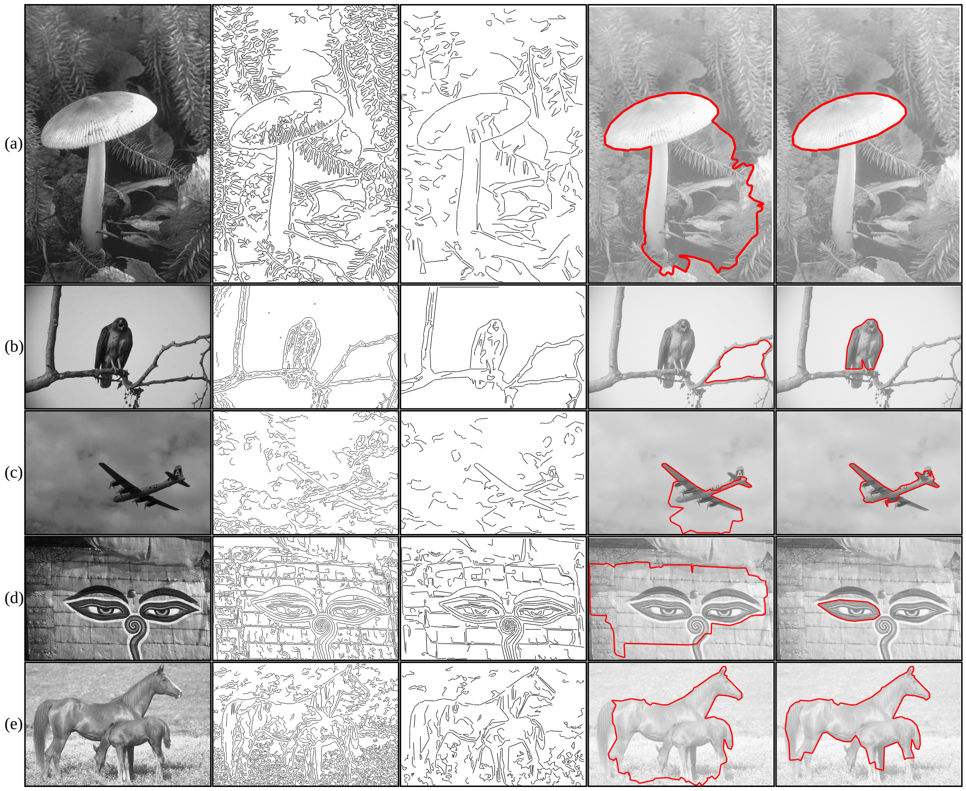


Fig. 14. Each row depicts the same information as in Fig. 13 except that the rightmost column shows the optimal boundary detected by the proposed method with the extension of replacing $\sigma_\epsilon^T(x, y)$ by $(1 - \sigma_\epsilon^T(x, y))$. ϵ is set to 50 for all images. T is set to 50, 200, 150, 150, and 220 for images (a-e), respectively.

This is mainly due to the fact that region area is quadratic with respect to the boundary perimeter and therefore the total gap length along the boundary. This problem has been noticed in previous pixel-grouping methods that seek to combine boundary and region information [18].

To address this problem, we introduce a new measurement of proximity by exponentiating the gap length to some power. This would reduce the order of magnitude difference between the proximity and region-area terms. While we could simply choose an exponent α and directly modify the grouping cost to

$$\phi(\mathcal{B}) = \frac{|\mathcal{B}_G|^\alpha}{\iint_{R(\mathcal{B})} dxdy},$$

it becomes difficult to encode such a grouping cost to the constructed graph model. Instead we propose to modify the grouping cost to

$$\phi(\mathcal{B}) = \frac{\sum_{\Gamma \in \mathcal{B}_G} |\Gamma|^\alpha}{\iint_{R(\mathcal{B})} dxdy},$$

where Γ is the gap-filling segments along $\mathcal{B}$. To encode this grouping cost to the graph G, we only need to modify the first edge weight of the dashed edges, e.g. e_{12}^+ and e_{12}^- corresponding to the gap-filling segment P_1P_2, to

$$w_1(e_{12}^+) = w_1(e_{12}^-) = |P_1P_2|^\alpha,$$

where $1 \leq \alpha \leq 2$. This does not change the graph structure of G and therefore, we can still apply the same graph algorithm to detect the optimal boundary that minimizes this new grouping cost.

Figure 15 shows an experiment where the different α's are used in the proposed edge grouping method with the extension of proximity exponentiation. We can clearly see that the increase of the proximity exponentiation factor α can reduce the dominance of the region area in the resulting edge grouping. A clear observation is that with the increase of α, we detect boundaries with smaller enclosed region areas. Figure 16 shows more experiment results in applying this extended method on some real images. The grouping results are compared to the proposed edge grouping method without this extension (or equivalently, with $\alpha = 1$). We can see that setting $\alpha = 1.5$ usually reduce the size of the detected structures, which may be desirable in some applications. This improvement is most noticeable on the image shown in Fig. 16(a), where the grouping result without the extension does not detect any line segments along the boundary of the bird, while the extension of proximity exponentiation allows the proposed method to detect the bird's boundary accurately. As with previous extensions, it is also possible to apply the strategy introduced in Section IV-C to iteratively detect multiple boundaries.

D. Detecting Simple Boundaries

Just like many previous edge-grouping methods, the proposed edge-grouping method has no guarantee to detect only *simple* boundaries without self intersections. The major reason lies in that the involved gap-filling segments may intersect with other gap-filling or detect segments. The boundaries with self intersections are not desirable since they do not represent the boundaries of any real structures. However, it should be noted first that, in using the proposed method, the nonsimple boundaries do not happen very frequently, since the presence of a self-intersection would produce a boundary that encloses multiple subregions with opposite-sign region areas. In this case, the total enclosed area is relatively small and therefore, such a boundary is not likely to be detected in using the proposed method. For example, the nonsimple boundary shown in Fig. 17 encloses two subregions with similar area but different area signs.

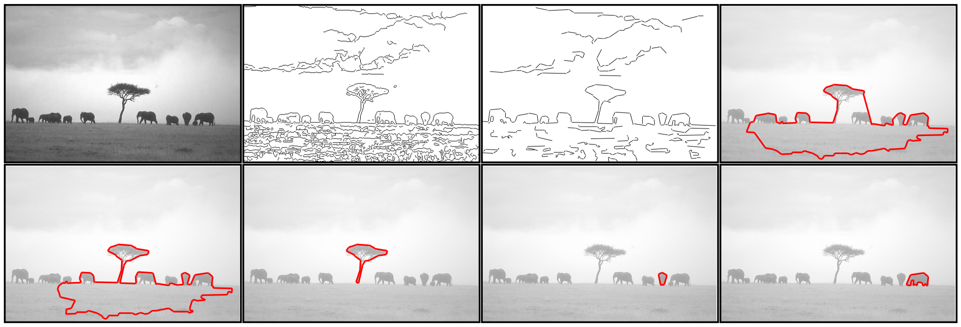


Fig. 15. Sample grouping results using the proposed grouping method with the extension of proximity exponentiation. The first row, from left to right, shows the input image, the Canny edge detection result, the line-approximation result, and the optimal boundary detected by the proposed method with proximity exponentiation factor $\alpha = 1$ (equivalent to the proposed method without proximity exponentiation). The second row, from left to right, shows the optimal boundaries detected by the proposed method with proximity exponentiation factor $\alpha = 1.1, 1.2, 1.5, 1.7$, respectively.

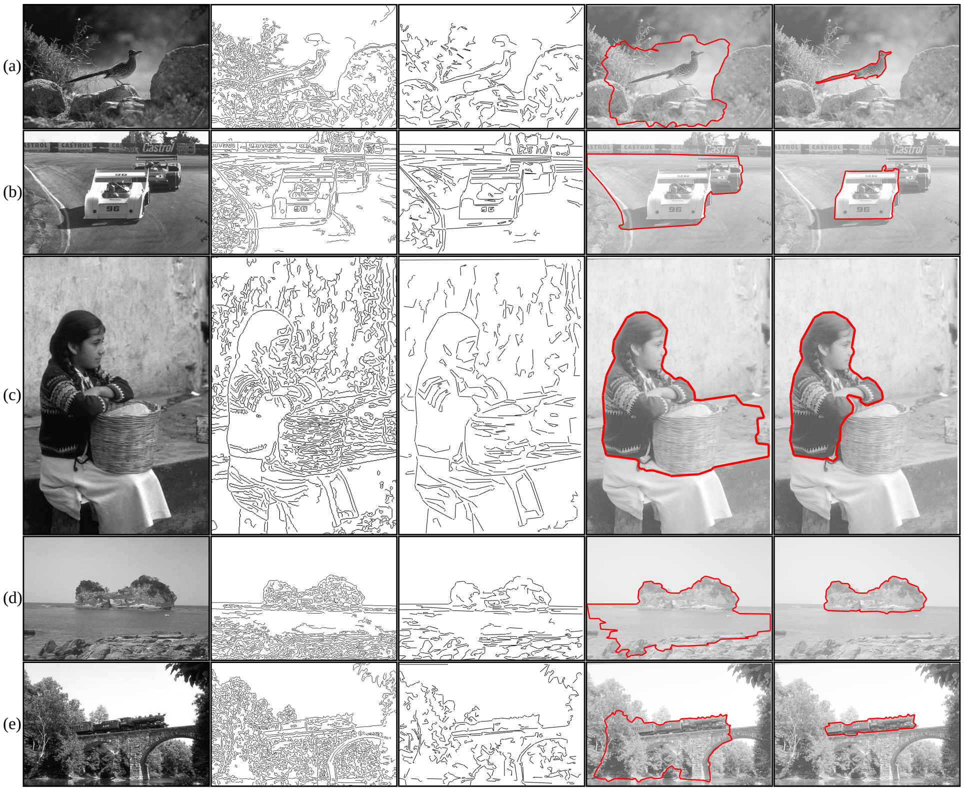
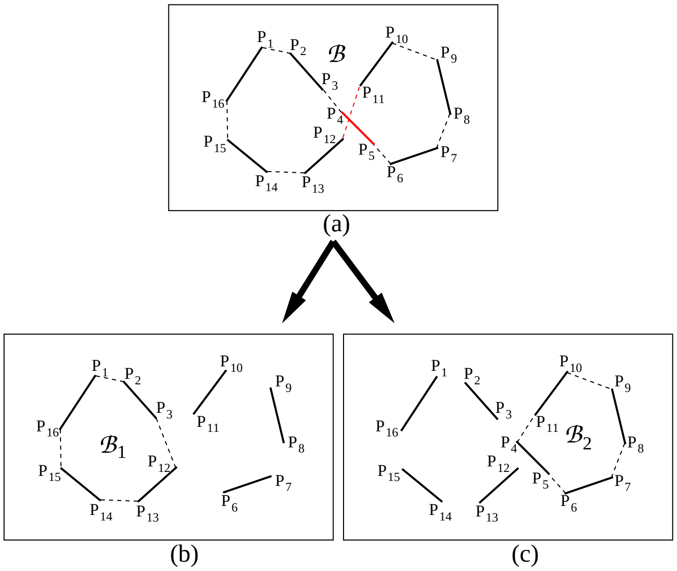


Fig. 16. More experiment results of applying the proposed method with proximity exponentiation on real images. Each row shows, from left to right, the input image, the Canny edge detection result, the line-approximation result, the optimal boundary detected when $\alpha = 1$, and optimal boundary when $\alpha = 1.5$.

The total area enclosed by this boundary is in fact close to zero. Anyway, we still need to solve this self-intersection problem when it happens. In this section, we present a strategy to force the proposed edge-grouping method to produce only *simple* boundaries.

The basic idea of this strategy is that, when a detected boundary contains a self intersection, we try to avoid this self intersection by not allowing the involved intersecting line segments to be included simultaneously in the detected boundary. For example, if the detected boundary $P_1 \dots P_{16}$ traverses two intersecting line segments $P_4 P_5$ and $P_{11} P_{12}$ as shown in Fig. 17(a), we consider two cases. Case 1: Remove segment $P_4 P_5$ from the input set of segments and repeat the proposed method to obtain a boundary $\mathcal{B}_1$ as shown in Fig. 17(b); Case 2: Remove segment $P_{11} P_{12}$ from the input set of segments and repeat the proposed method to obtain a boundary $\mathcal{B}_2$ as shown in Fig. 17(c). If both $\mathcal{B}_1$ and $\mathcal{B}_2$ are simple, the one with smaller grouping cost $\phi(\cdot)$ is then the desirable salient simple boundary. If any one of them is nonsimple, we may continue considering two more cases by further removing one more involved segment.



(a)

(b)

(c)

Fig. 17. An illustration of the strategy for detecting only simple boundaries.

We can see that the direct implementation of this strategy in fact generates a binary tree where the root node represents an edge grouping on all line segments, each nonroot node represents an edge grouping by removing some line segments, and each leaf node represents an edge grouping where the resulting boundary is simple. It is also easy to see that the grouping cost keeps increasing from a parent node to its children. In practice, we can use a *branch-and-bound* technique [19] to improve the algorithm efficiency:

we always process the node with the smallest grouping cost from all the available ones and for any node with a grouping cost larger than a known leaf node (a known detected simple boundary), we are not going to proceed to its children. It is well known that this brand-and-bound strategy may still result in an exponential complexity in the worst case. However, as mentioned above, the consideration of region-area information makes the proposed edge-grouping method biased to produce simple boundaries. Therefore, even when a nonsimple boundary is detected, we expect that the branch-and-bound tree would have small depth and the optimal simple boundary can be found in a very small number of iterations. In our about 4500 experiments on detecting the first optimal boundaries (on different real images, with/without various extensions), we only come across one case of detecting a nonsimple boundary, which is shown in Fig. 18. In this case, the optimal simple boundary shown in Fig. 18(d) is achieved after the blue segment shown in Fig. 18(c) is removed and the depth of the constructed branch-and-bound tree is two. However, when repeating the proposed method to detect multiple boundaries using the strategy introduced in Section IV-C, we may encounter more cases of self-intersected boundaries.

VI. Related Work

A. Edge Grouping

There has been a long line of research on edge grouping with many grouping costs and grouping algorithms developed in past decades. However, most of the available edge-grouping methods only consider the boundary information, such as the Gestalt laws of closure, proximity and continuity. For example, many edge-linking algorithms have been developed to connect the disjoint edges resulting from an edge detector into longer edges or complete boundaries [20], [21], [22], [23], [24]. A typical criteria used in edge linking is to connect the edges that are close to each other, which in fact indicates the preference of boundary proximity. However, it is well known that by only considering proximity, most edge-linking algorithms are very sensitive to image noise and it may be difficult for them to directly extract perceptually salient structures.

In [11], Shashua and Ullman use a local parallel network to model the line segments and define a grouping cost by combining the boundary proximity and continuity. An iterative update algorithm is developed to search for an optimal boundary that may not be closed. Alter and Basri [2] further conduct an extensive analysis of this parallel-network method and point out the problems when applying this method iteratively to detect the second most salient boundary and the problems due to discretization. Recent work on edge grouping includes Elder and Zucker [5], Williams and Thornberg [14], and the ratio-contour method [13] where boundary closure, continuity and proximity are combined in the grouping

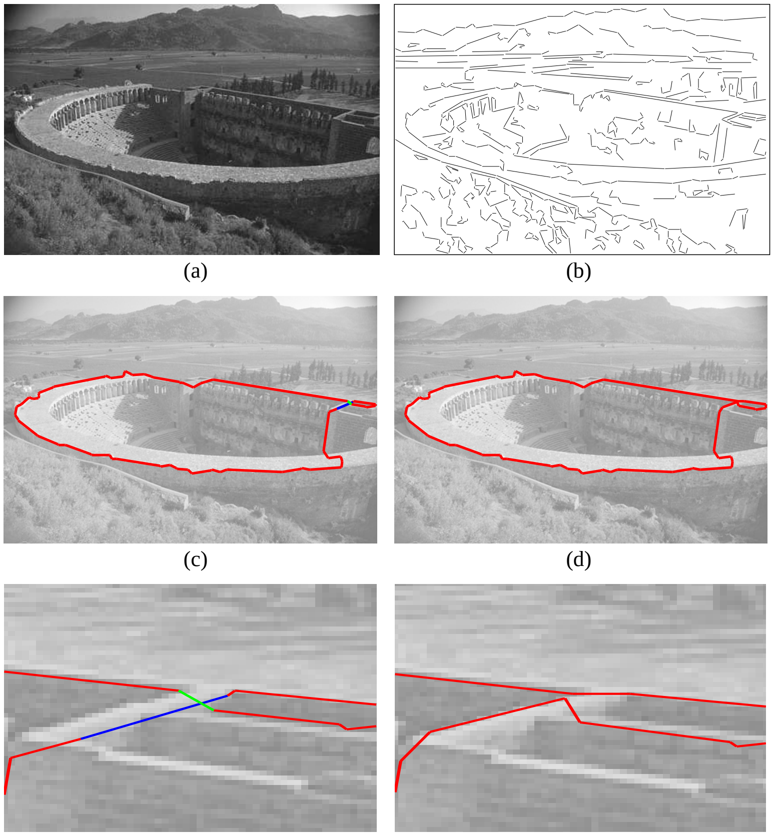


Fig. 18. An example of applying the branch-and-bound strategy to detect simple boundaries. (a) The input image. (b) The detected segments. (c) The edge-grouping result by applying the proposed method with the continuity extension (Section V-A), with $\lambda = 1$. The blue and green segments intersect each other. (d) The simple boundary detected by the branch-and-bound strategy. (e) & (f) The zoomed version of the subregions of (c) and (d) around the blue/green segments in (c), respectively.

cost. Among these three methods, the ratio-contour method has been shown to have a better performance by being more robust to image noise. Note that, our experiments in Section IV has shown that the method developed in this paper usually performs better than the ratio-contour method.

In [12], we develop a convex edge-grouping method that combines both boundary and region information. The grouping cost is also defined in a ratio form with a combined measure of proximity and region intensity homogeneity in the numerator and the region area as the denominator. However, the graph modelling and algorithm used in [12] can only detect convex boundaries. Without convex constraint, we believe it is an NP-hard problem to minimize the grouping cost in [12]. In [25], we pair up line segments into a new grouping token in order to detect symmetric boundaries by combining boundary and region information. However, the assumption of symmetry does not hold for all structures in the real world. Note that our previous work on ratio contour, convex grouping and symmetric grouping also use a ratio-form grouping cost and finally reduces the problem to the graph problem of finding the minimum ratio alternate cycle, which is similar to the one derived in this paper, as shown at the end of Section III. However, both the grouping cost and graph-modelling process developed in this paper are different from the ones developed in these previous work since the grouping tasks are different.

B. Combining Boundary and Region Information

Pixel grouping is another important class of grouping methods that has been widely investigated in past years. In a pixel-grouping method, each image pixel is treated as a token and the goal of grouping is usually to find a boundary that partitions the image into foreground and background regions or two subregions without labelling the foreground and background. Comparing edge grouping with pixel grouping, there are both advantages and disadvantages. For example, edge grouping simplifies the quantization of many important boundary properties, such as continuity, convexity, and symmetry, which may be difficult to consider in pixel grouping. In addition, the number of line segments detected in an image is usually much smaller than the number of pixels in an image. This may make the edge grouping to take less CPU time when conducting globally optimal grouping. However, edge grouping may fail when the line segments can not be well detected from the input image.

While combining boundary and region information has not been extensively explored in previous edge grouping methods, except for our recent work on convex and symmetric edge grouping [12], [25], it has been investigated in several pixel-grouping methods. In [18], Cox, Rao and Zhong present a ratio-region method to detect a closed boundary that partitions an image into foreground and background regions. The grouping cost is of a ratio form, with the numerator measuring the boundary property and the denominator measuring the foreground region area. It uses an optimization algorithm of repeating the minimum-cut max-flow algorithm to find the graph partitioning that minimizes the grouping cost. In [26], Jermyn and Ishikawa further extend the ratio-region method so that different boundary and region properties can be considered in the grouping cost. In this method, the problem is modelled in a directed graph and finally reduced to a graph problem of finding the minimum ratio cycle in the constructed directed graph. This graph problem shares some similarity with the one formulated in this paper, but differs in (a) its graph is *directed* and the graph in this paper is an *undirected solid-dashed* graph, and (b) it searches over all simple cycles while we only consider *alternate* cycles. In addition, some extensions, such as the proximity exponentiation introduced in Section V-C, cannot be applied in pixel grouping.

In [27], Shi and Malik develop a normalized-cut method that seeks to partition an image into two balanced regions, without the labelling of the foreground and background. Given the NP-completeness of this problem, spectral graph theory is applied to achieve an approximate solution. In [28], Sumengen and Manjunath present a graph-partitioning active contour approach to iteratively search for a locally optimal boundary that minimizes the grouping cost used in normalized cut. In [29], Chan and Vese define a grouping cost that requires the resulting boundary to be both smooth and enclose a region with homogenous intensity. In [30], Wang and Oliensis define a comprehensive grouping cost that measures the complexity of both foreground and background regions. These complicated grouping costs, however, usually lead to NP-hard or even non-polynomial-time problems and only local optimal solutions can be found by some gradient-descending approaches, such as active contours.

VII. Conclusions

In this paper, we presented a new edge grouping method that can detect perceptually salient structures from an image by combining the boundary and region information. In its baseline form, the boundary proximity and region area are combined into a ratio-form grouping cost function. We then develop a graph model to reduce this edge grouping problem to a graph problem that can be solved in polynomial time in a globally optimal fashion. We tested this edge-grouping method on a large set of synthetic data and some real images, both with comparisons to the ratio-contour method, which does not consider region information. We showed that the inclusion of region-area information makes the proposed method more robust against image noise and improves the performance in general. We also presented several extensions to the proposed method that might be useful in some applications. These extensions include the addition of boundary continuity and region intensity homogeneity into the grouping cost, the better balance between the boundary and region terms in the grouping cost, and the enforcement of detecting only simple boundaries. Some sample experiment results are shown to validate the use and the effectiveness of these extensions.

Acknowledgments

This work was funded, in part, by grant NSF-EIA-0312861.

References

[1] G. Kanizsa, *Organization in Vision*. New York: Praeger, 1979.

[2] T. Alter and R. Basri, “Extracting salient contours from images: An analysis of the saliency network,” in *International Journal of Computer Vision*, 1998, pp. 51–69.

[3] A. Amir and M. Lindenbaum, “A generic grouping algorithm and its quantitative analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 2, pp. 168–185, 1998.

[4] J. H. Elder, A. Krupnik, and L. A. Johnston, “Contour grouping with prior models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 6, pp. 661–674, 2003.

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

[6] G. Guy and G. Medioni, “Inferring global perceptual contours from local features,” *International Journal of Computer Vision*, vol. 20, no. 1, pp. 113–133, 1996.

[7] D. Huttenlocher and P. Wayner, “Finding convex edge groupings in an image,” *International Journal of Computer Vision*, vol. 8, no. 1, pp. 7–29, 1992.

[8] D. Jacobs, “Robust and efficient detection of convex groups,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 1, pp. 23–27, 1996.

[9] 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*, vol. 25, no. 4, pp. 433–444, 2003.

[10] 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*, 1996, pp. 478–483.

[11] 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*, 1988, pp. 321–327.

[12] J. S. Stahl and S. Wang, “Convex grouping combining boundary and region information,” in *IEEE International Conference on Computer Vision*, vol. 2, 2005, pp. 946–953.

[13] S. Wang, T. Kubota, J. Siskind, and J. Wang, “Salient closed boundary extraction with ratio contour,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 4, pp. 546–561, 2005.

[14] L. Williams and K. K. Thornber, “A comparison measures for detecting natural shapes in cluttered background,” *International Journal of Computer Vision*, vol. 34, no. 2/3, pp. 81–96, 2000.

[15] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[16] 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/.

[17] 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*, vol. 2, July 2001, pp. 416–423.

[18] I. Cox, S. B. Rao, and Y. Zhong, “Ratio regions: A technique for image segmentation,” in *International Conference on Pattern Recognition*, 1996, pp. 557–564.

[19] R. Horst and H. Tuy, *Global Optimization: Deterministic Approaches*, 3rd ed. Berlin: Springer-Verlag, 1996.

[20] A. Farag and E. Delp, “Edge linking by sequential search,” *Pattern Recognition*, vol. 28, no. 5, pp. 611–633, 1995.

[21] E. Saber, A. Tekalp, and G. Bozdagi, “Fusion of color and edge information for improved segmentation and edge linking,” *Image and Vision Computing*, vol. 15, no. 10, pp. 769–780, 1997.

[22] R. Gonzalez and R. Woods, *Digital Image Processing*. Upper Saddle River: Prentice Hall, 2002.

[23] O. Ghita and P. Whelan, “Computational approach for edge linking,” *Journal of Electronic Imaging*, vol. 11, no. 4, pp. 479–485, 2002.

[24] A. Sappa, “Unsupervised contour closure algorithm for range image edge-based segmentation,” *IEEE Transactions on Image Processing*, vol. 215, no. 2, pp. 377–384, 2006.

[25] J. S. Stahl and S. Wang, “Globally optimal grouping for symmetric boundaries,” in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2006, pp. 1030–1037.

[26] I. H. Jermyn and H. Ishikawa, “Globally optimal regions and boundaries as minimum ratio cycles,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 10, pp. 1075–1088, 2001.

[27] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[28] B. Sumengen and B. S. Manjunath, “Graph partitioning active contours (GPAC) for image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 509–521, 2006.

[29] T. F. Chan and L. A. Vese, “Active contours without edges,” *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.

[30] H. Wang and J. Oliensis, “Salient contour detection using a global contour discontinuity measurement,” in *IEEE Workshop on Perceptual Organization in Computer Vision*, 2006, p. 190.