Robust Edge Grouping with Region Information

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

Abstract

This paper introduces a new edge grouping method that combines boundary and region information to detect salient objects in noisy images. Previous work on edge grouping has shown that methods that are based on boundary information alone are biased to detect smaller boundaries, and are more sensitive to noise. Some previous methods have introduced additional cues to guide the grouping, such as continuity or convexity, to improve the boundary detection, but these cues usually restrict the results that can be obtained in general applications. We introduce region information in the form of region area instead, which produces more favorable results in general, and is more robust against noise when compared to other cues. We test this method on a set of synthetic data to obtain a quantitative comparison against a previous method. We then present the results on a set of real images.

1 Introduction

This paper presents a new method on edge grouping, an area of computer vision that seeks to identify objects in noisy images by obtaining a set of edge fragments and grouping a subset of these to correspond with the boundary of a salient structure in the image [1]. This problem is an important step in mid-level computer vision, that can benefit high-level tasks such as object recognition or image retrieval from databases. There has been a long line of previous research on this subject, including [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14].

These grouping methods usually attain their goal by first obtaining a set of edge fragments from the original image, usually as line segments by using an edge detection and line fitting algorithm, and then group a subset of these line segments. To prioritize one subset over others, a function is defined that assigns a cost to a given subset. The method then finds the subset among all possible that minimizes this cost function. This approach makes edge grouping very appealing since it allows for an integration of perception rules or cues, such as the ones proposed by the psychology Gestalt laws of human perception [15]. These include cues such as *proximity*, which requires grouped line segments to be close to each other, *closure*, which requires the obtained boundary to be always closed, and *continuity*, which requires the boundary to be smooth. These can be used as criteria to formulate the cost function and usually involve properties of the desired boundary.

In this paper we present a new edge grouping method that combines boundary and region information in its cost function. Namely it uses the cues of proximity, closure and region area. Proximity has traditionally been considered by many researchers in edge grouping. Region information has been proposed in pixel grouping methods such as [16], which can be adapted to a variety of region cues, such as region area or intensity homogeneity. These cues can avoid a bias toward detecting small boundaries, and the use of these cues in edge grouping methods would be desirable. Edge grouping methods are usually faster than pixel grouping methods and less sensitive to noise since they deal with line segments, which are a smaller number than the number of pixels in the image. However, the use of region information as a cue has not been widespread among edge grouping methods. Previous edge grouping methods that have included region area usually are limited by some other geometric constraint, as in [12], which formulated a simple way of measuring region area but can only detect convex shapes. The method presented in this paper does not have such a constraint and thus can be applied to more general cases. A graph representation of the problem and algorithm is presented that solves the minimization of the cost function in a globally optimal fashion. We then compare it against a previous method and present the results on experiments on both synthetic data and real images.

The remainder of this paper is organized as follows. In Section 2 we formulate the problem. In Section 3 we present the construction of the graph that models the formulated problem, and the graph algorithm to solve it. In Section 4 we present the experiment results on synthetic data and real images, with a comparison against a previous method. In Section 5 we give a brief conclusion.

2 Problem Formulation

Given an input image, a typical edge grouping method performs the following three steps, illustrated on Fig. 1. First, an edge detector such as the Canny detector is applied to the image, and the resulting edges are then approximated by straight line segments, as shown in Fig. 1(b). These are usually referred to as *detected (line) segments*. Second, since the resulting detected segments are disconnected, to form a boundary the gaps between them need to be filled. Every pair of endpoints from different detected segments is then connected with new line segments, shown as dashed lines in Fig. 1(c) [1]. These are usually referred to as *gap-filling (line) segments*. A boundary is then defined as a cycle of alternate detected and gap-filling segments. Third, an algorithm is developed to find the boundary among all possible that minimizes the selected cost function. This boundary is expected to align well with the boundary of a salient structure in the real image, as shown in Fig. 1(d). The cost function encodes the desired properties of the boundary, such as proximity or continuity.

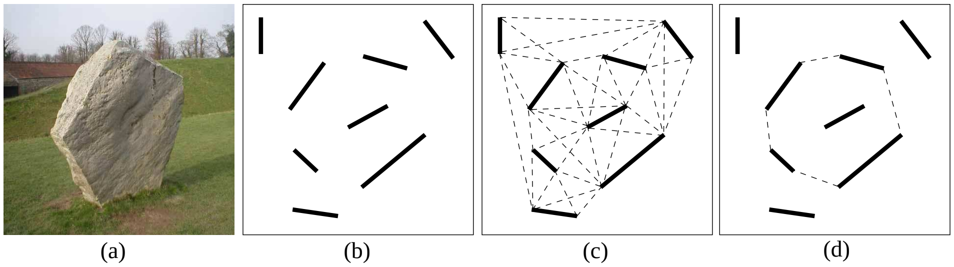


Figure 1: The typical stages of an edge grouping method

We will follow this strategy for the proposed method. Closure will be enforced as a hard constraint in the optimization algorithm, and we introduce the following cost function for a boundary $\mathcal{B}$,

$$\phi(\mathcal{B}) = \frac{|\mathcal{B}_G|}{\texttt{area}(\mathcal{B})}, \tag{1}$$

where $|\mathcal{B}_G|$ is the sum of all the lengths of the gap-filling segments along the boundary $\mathcal{B}$, which represents the proximity cue. The normalization term $\texttt{area}(\mathcal{B})$ is the region area enclosed by the boundary $\mathcal{B}$, which will set a preference for larger structures and more robustness against noise. The problem is then formulated as finding the boundary that minimizes this cost function.

3 Graph Modeling and Algorithm

To solve the problem formulated in the previous section, we will map it to a graph and apply a known graph algorithm to solve it. We begin by constructing a graph $G = (V, E)$, with a set of

[1] Not all possible gap-filling segments are shown, in order to keep the figure readable.

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, we construct two dashed edges, e_{23}^+ and e_{23}^-, shown by dashed lines in Figs. 2(a) and (b), and for each line-segment endpoint P_i, $i = 1, 2, 3$, we construct two vertices, $u_i^{(1)}$, $u_i^{(2)}$, $i = 1, 2, 3$. We will show that this construction of edges in pairs facilitates the quantization of $\mathtt{area}(\mathcal{B})$, the region area enclosed by $\mathcal{B}$.

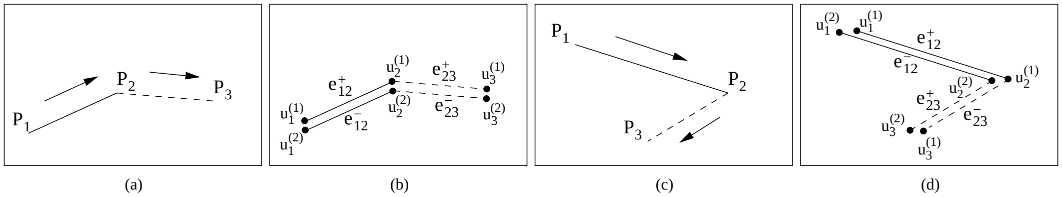


Figure 2: An illustration of the graph construction. Linking edges corresponding to two segments with (a,b) same direction, (c,d) opposite direction.

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 Fig. 2(a) and Fig. 2(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 two possibilities in linking the edges obtained from the line segments P_1P_2 and P_2P_3 shown in Figs. 2(a) and (c). If the x-coordinates of the endpoints are equal, we decide by the y-coordinates in a similar fashion.

In this constructed graph, a closed boundary $\mathcal{B}$, which traverses some detected and gap-filling segments alternately, is in fact modeled by two cycles in G 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.

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

$$w_1(e_{12}^+) = w_1(e_{12}^-) = \begin{cases} 0 & \text{if } P_1P_2 \text{ is solid} \\ |P_1P_2| & \text{if } P_1P_2 \text{ is dashed,} \end{cases}$$

where $|P_1P_2|$ is the length of the line segment. 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

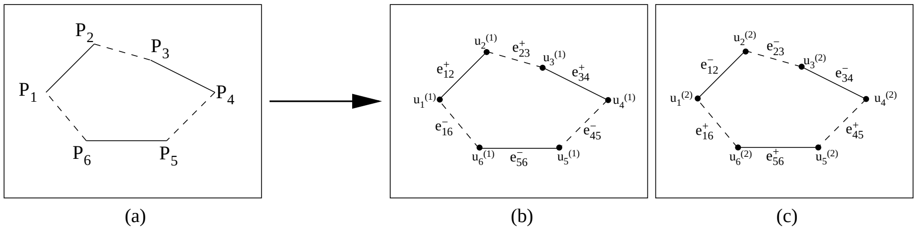


Figure 3: A boundary $P_1P_2 \ldots P_6$ corresponds to two "mirror" cycles in the graph G.

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 of 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.

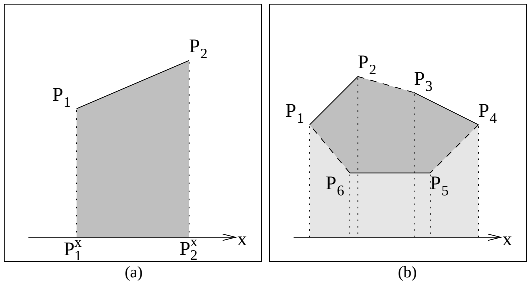


Figure 4: An illustration of defining the second weight for an edge. (a) Area associated to a line segment. (b) The region area enclosed by a closed boundary is equal to the sum of the signed areas associated to the line segments along this boundary.

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 $W_1(\mathcal{C}^+) = \sum_{e\in\mathcal{C}^+} w_1(e) = W_1(\mathcal{C}^-) = \sum_{e\in\mathcal{C}^-} w_1(e) > 0$ and $W_2(\mathcal{C}^+) = \sum_{e\in\mathcal{C}^+} w_2(e) = -W_2(\mathcal{C}^-) = -\sum_{e\in\mathcal{C}^-} w_2(e)$. 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 the $\phi(\mathcal{B})$ 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 the $\mathcal{C}^-$ version 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 ratio-contour algorithm[13].

4 Experiments

The implementation of the proposed method was written in C++, and evaluated on a set of synthetic data and real images [2]. The synthetic data was generated as a set of line segments, and for the real

[2] A weblink to the software developed in this work will be placed here. It is now omitted for double-blind review.

images we obtained the initial set of detected segments from edge detection and line approximation. We used the Canny edge detector from the Matlab image processing toolbox, and the line approximation package by Kovesi [17]. For the Canny edge detector we left the parameters at their default values, and for line approximation we set the minimum edge length to be processed to 30 pixels, and the maximum deviation between an edge and its fitted lines to 2 pixels.

4.1 Experiments on Synthetic Data

To evaluate the proposed method quantitatively we constructed a set of synthetic data and measured its accuracy 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 detected optimal boundary and the ground truth, respectively, and $|R|$ is the area of R. The method is compared against the method proposed by Wang et al. in [13], which is an edge grouping method that uses proximity and continuity as its cues, normalized by the boundary perimeter length, in its cost function

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

where $\kappa(t)$ is the curvature at position t. In our experiments we left the free parameter λ at its default value of 10.

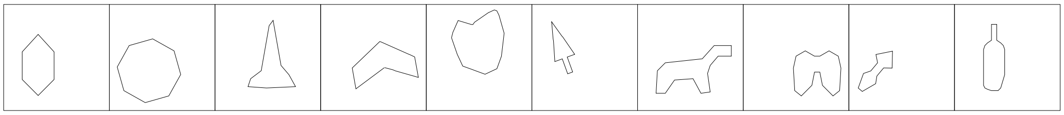


Figure 5: 10 objects used for the construction of synthetic data.

For constructing our synthetic data we used the 10 objects shown in Fig.5 as ground truths, placed inside square regions of size 128×128. Noise was added in the form of line segments that were placed at random locations, in random directions, with a length selected randomly between 3 and 7 pixels (all properties uniformly distributed). The number of noise line segments added is picked from the set $\{0, 10, 20, 40, 80\}$. Figure 6(a) shows the $4th$ object from Fig. 5 with 40 noise segments added. Additionally, from each object we removed a certain percentage of its boundary at random locations. This gap percentage was chosen from the set $\{0\%, 5\%, 10\%, 20\%, 30\%, 40\%, 50\%\}$. Figure 6(b) shows the same as (a) but with 30% of the object's boundary removed. Figures 6(c) and (d) show the result of running the proposed method and the Wang et al. method on Fig. 6(b), respectively. These results represent an accuracy of 0.96 and 0.56 respectively.

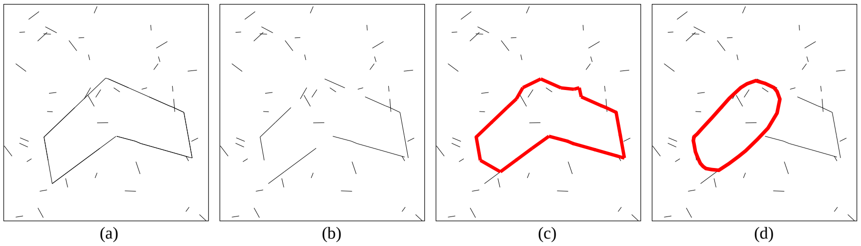
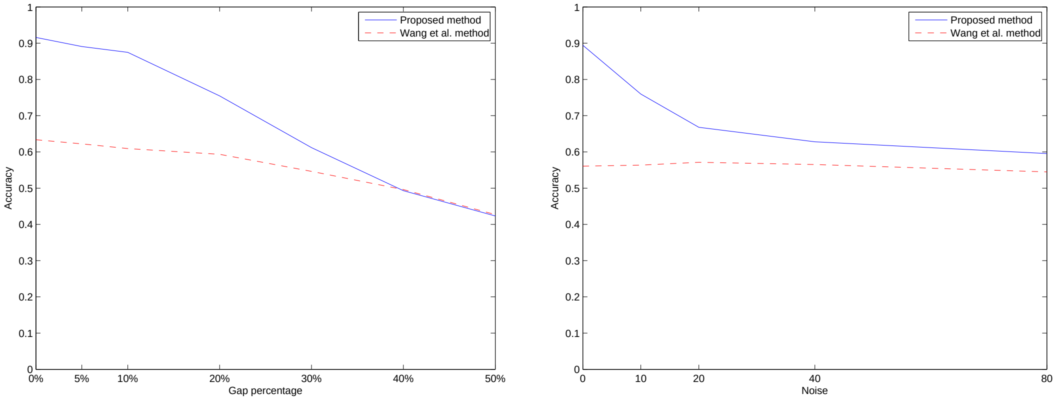


Figure 6: Sample constructed synthetic data and results.

We constructed 10 sets of synthetic data, where each set consists of an instance of all possible combinations of noise segments and gap percentages, for a total of $10 \times 10 \times 7 \times 5 = 3500$ synthetic samples. Figure 7 shows the results obtained by running the proposed method and the Wang et al. method on these synthetic samples. These graphs show that the inclusion of the region area in the cost function is in general more robust against noise than the use of continuity, as well as an improvement in overall performance.



(a)

(b)

Figure 7: Statistical results obtained from the synthetic experiments.

4.2 Experiments on Real Images

We tested the proposed method on real images obtained from the Berkeley segmentation dataset [18]. The results for 10 images are shown in Figs. 8, where each column shows, from left to right, the original image, Canny detection result, line approximation result, the optimal boundary obtained by the Wang et al. method [13], and the optimal boundary obtained by the proposed method.

These results show the better performance of the proposed method for most images, as seen in Figs. 8(a-h,j). There are some cases where both methods produce different yet both acceptable results, as in Fig. 8(i). Figure 8(f) shows a case where the proposed method detects a smaller structure than the Wang et al. method, despite the use of region area. In this case the proximity plays a more important role, but in the Wang et al. method the use of continuity may prevent it from detecting the animal correctly.

4.3 Multiple Boundary Detection

So far we presented the proposed method in the context of obtaining the boundary that optimizes the cost function (1). However, it is easy to extend this method to obtain multiple boundaries in an image by iterating the method. Given an image, e.g. the one shown in Fig. 9, we first process it with the proposed method to obtain the optimal boundary. Then, we remove from the graph G all the edges associated to line segments that belong to the optimal boundary, i.e. for each line segment P_iP_j in the boundary we remove both e_{ij}^+ and e_{ij}^-. We then run the graph algorithm again on G, to obtain the second optimal boundary. This process, can be repeated n times to obtain the nth optimal boundary. An example is presented in Fig. 9, where on its first row, from left to right, it shows the original image, the Canny edge detection result, the line approximation result and the 1st optimal boundary. On the second row, it shows the 2nd through 5th optimal boundaries obtained by iterating the proposed method.

5 Conclusions

In this paper we presented a new edge grouping method that integrates the cues of proximity and region area in its cost function, to detect salient closed boundaries. We tested the method on a large set of synthetic data and also presented results on real images, both with comparisons to a previous method. We showed that the inclusion of region area makes the proposed method more robust against noise and improves the performance in general.

Acknowledgments

(Acknowledgments omitted for blind review).

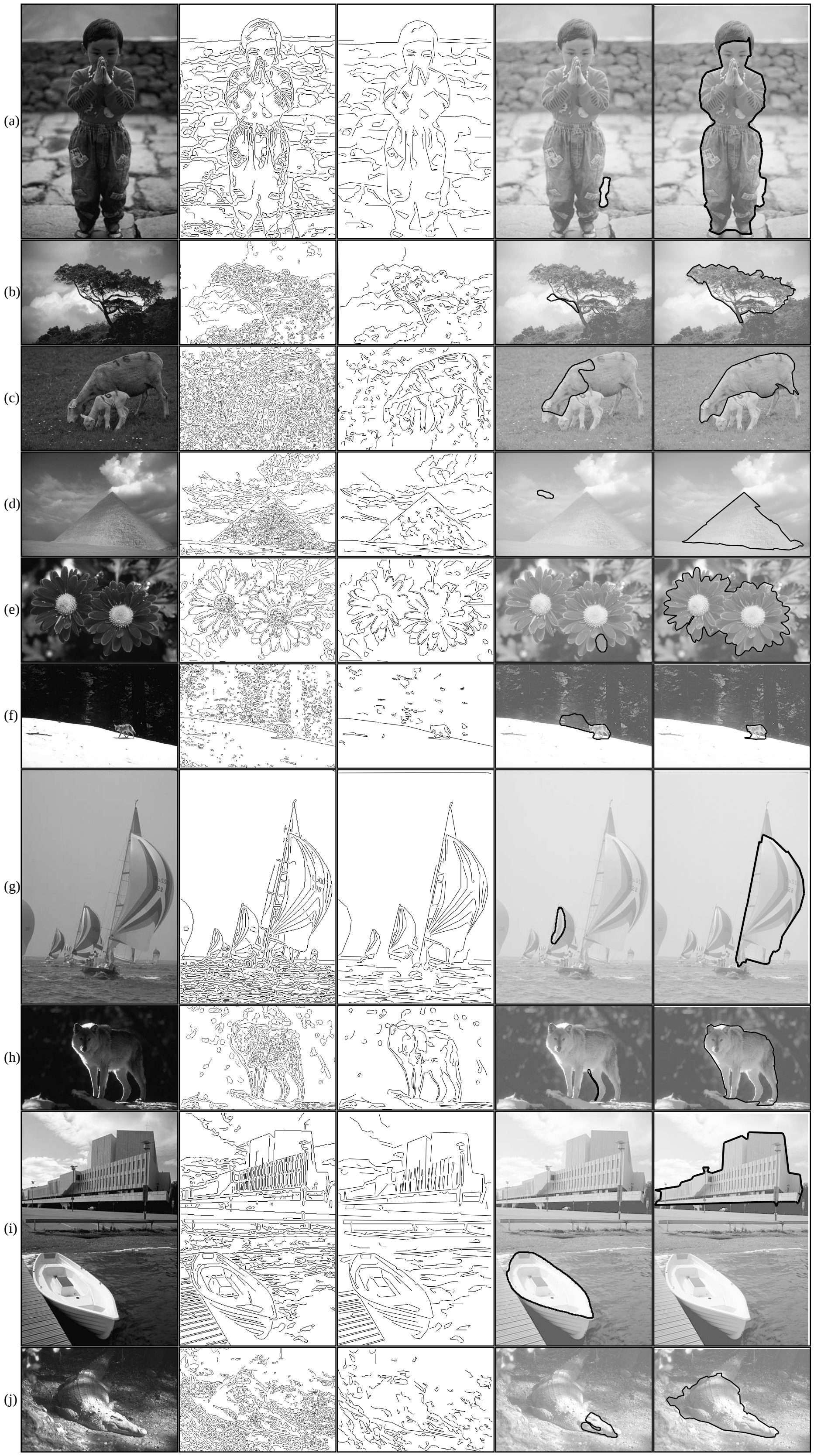


Figure 8: Results on 10 real images.

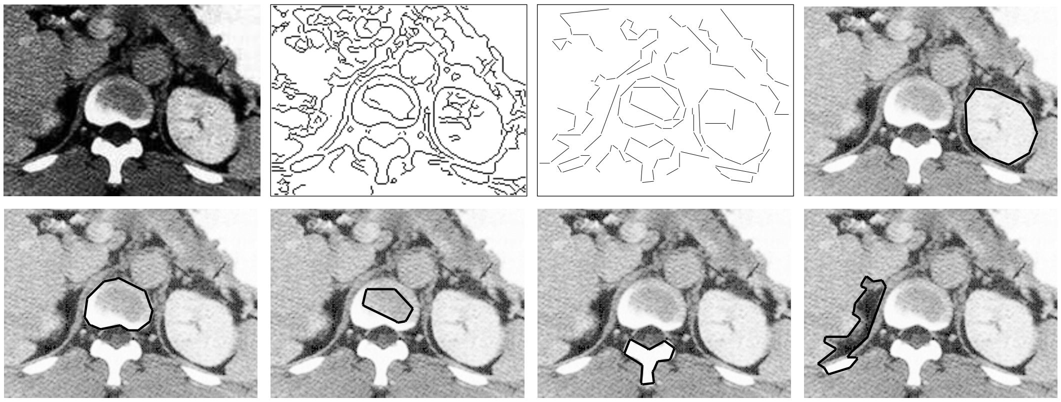


Figure 9: Example of obtaining multiple boundaries from a real image.

References

[1] D. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Upper Saddle River, NJ: Prentice Hall, 2003.

[2] 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.

[3] 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.

[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] 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.

[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*, pages 478–483, 1996.

[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*, pages 321–327, 1988.

[12] 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.

[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*, 27(4):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*, 34(2/3):81–96, 2000.

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

[16] 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.

[17] 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/.

[18] 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 416–425, 2001.