Globally Optimal Grouping for Symmetric Boundaries

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

Abstract

Many natural and man-made structures have a boundary that shows certain level of bilateral symmetry, a property that has been used to solve many computer-vision tasks. In this paper, we present a new grouping method for detecting closed boundaries with symmetry. We first construct a new type of grouping token in the form of a symmetric trapezoid, with which we can flexibly incorporate various boundary and region information into a unified grouping cost function. Particularly, this grouping cost function integrates Gestalt laws of proximity, closure, and continuity, besides the desirable boundary symmetry. We then develop a graph algorithm to find the boundary that minimizes this grouping cost function in a globally optimal fashion. Finally, we test this method by some experiments on a set of natural and medical images.

1. Introduction

Many structures of interest encountered in computer-vision tasks show certain level of (bilateral) symmetry [4, 5]. For example, many man-made structures are constructed as surfaces of revolution or generalized cylinders, which result in some ribbons with a symmetry axis. This symmetry property has been shown to be an important cue in both boundary interpretation and matching, where the goal is to analyze and match some given boundaries, and grouping, where the goal is to extract desirable structural boundaries from real images [9]. As shown in Fig. 1, the goal of this paper is to develop a method to address the latter problem of grouping with symmetry, which, as pointed out in [9], is a particularly challenging problem.

Several reasons make the grouping with symmetry a challenging problem. First, unlike many other grouping cues, symmetry is not a simple local measure, and therefore we need to introduce some new grouping tokens to encode it, instead of the pixels or edges that are used in many previous grouping methods. Second, while symmetry is an important grouping cue, other cues, such as Gestalt laws of proximity, continuity, and closure, are also crucial to achieve a successful grouping [14]. This calls for a unified *grouping cost (function)* that can flexibly integrate these important grouping cues. Third, the grouping cost should be designed to avoid some undesirable explicit or implicit biases, such as a bias to produce shorter boundaries, which exist in many previous grouping methods. Finally, it is usually a challenging problem to develop an algorithm for finding the globally optimal grouping that minimizes the selected grouping cost.

In this paper, we propose a new grouping cost function that can integrate the cues of closure, proximity, continuity, and symmetry. This grouping cost function also combines both boundary and region information to avoid possible biases toward shorter boundaries. We then develop a graph algorithm to solve this grouping problem in a globally optimal fashion. In the proposed method, a set of line segments are first extracted from the input image. From these line segments, we construct new grouping tokens in the form of symmetric trapezoids and the grouping with symmetry is then formulated as identifying and connecting a sequence of the symmetric trapezoids into a closed boundary.

The related work includes the long-line research on edge grouping [2, 3, 7, 8, 10, 11, 12, 19, 21, 22, 23, 24, 25]. These methods aim to extract some salient boundaries from a set of detected line segments and their grouping costs usually combine some well known Gestalt laws, such as (a) *closure*, which requires the resulting boundary to be always closed, (b) *proximity*, which requires the gap length to be short in connecting line segments into a closed boundary, (c), *continuity*, which requires the resulting boundary to be as smooth as possible, and (d) *convexity*, which requires the resulting boundary to be convex. However, these edge-grouping methods do not consider the cue of symmetry in grouping.

The important role of symmetry has been studied in both human vision and computer vision. Particularly, prior research has shown that symmetry is non-accidental [18, 26] and therefore, can be used as a grouping cue to distinguish salient structures from noisy background. Symmetry analysis of a given object boundary is usually conducted by de-

Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)
0-7695-2597-0/06 $20.00 © 2006 IEEE

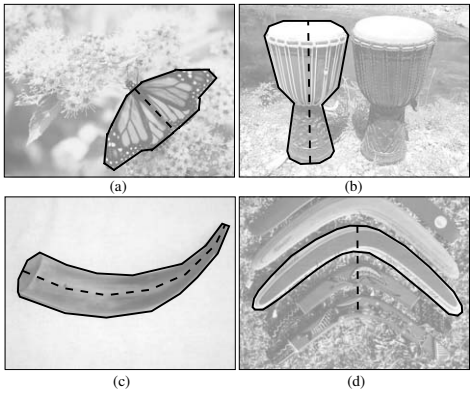

Figure 1. Some samples of structural boundaries that show symmetry. Structural boundaries are shown in solid while the symmetry axes are shown in dashed.

riving its *symmetry axis*, also named *median axis* in some prior literature. Symmetry axis information has been incorporated to facilitate boundary interpretation, matching, and recognition in many prior works [5, 6, 16, 27]. Note that the method developed in this paper aims to solve the inverse problem of these symmetry-analysis tasks: the structural boundaries are not available and our goal is to extract them by incorporating the symmetry cue.

Related to the proposed work is the grouping method developed by Mohan and Nevatia [20], where symmetry is considered along with closure and proximity. It applies both edge detection and corner detection to extract a set of line segments and corner points as the grouping tokens. The grouping cost is defined by a collinearity measure that actually reflects the proximity and continuity of the boundary. Symmetry is applied as a cue to pair the extracted curves by producing a set of *ribbons*. These ribbons are then grouped into some structures by some heuristic algorithms. This method neither introduces a unified grouping cost function combining various cues nor develops a globally-optimal grouping algorithm. As a result, it produces many small ribbons and may not handle the boundary occlusion very well. Another work that is closely related to our work is the grouping method developed by Liu, Geiger, and Yuille [17], which identifies the local symmetry-axis segments and then applies a shortest-path algorithm to connect some of them into a complete symmetry axis. The grouping cost function is defined as the sum of local costs along the symmetry axis. By manually selecting a starting pair of points that are symmetric to each other, this method produces an open boundary. Since this method does not consider region information or other kind of normalization of the cost function, it presents a bias towards smaller boundaries. Such a bias may prevent this method from accurately detecting the symmetry axis shown in Fig. 1(d).

The remainder of this paper is organized as follows. In Section 2, we present the problem formulation by introducing the general edge-grouping methodology, the new grouping token in the form of a symmetric trapezoid, and the new grouping cost function. In Section 3, we present the graph modelling of this problem and develop the graph algorithm to solve this grouping problem in a globally optimal fashion. In Section 4, we discuss some implementation details and report some experiment results on some real images. Section 5 gives a short conclusion.

2. Problem Formulation

2.1. Edge Grouping

Edge grouping is a widely studied grouping methodology, in which the grouping tokens are a set of line segments and the output is one or several perceptually-salient boundaries formed by connecting a subset of the line segments. Starting from an input image, a typical edge-grouping method operates as follows. First, a set of line segments, as shown in Fig. 2(b), is detected from the input image by edge detection and line fitting. Second, a new set of line segments, as shown by dashed lines in Fig. 2(c), are constructed to fill the gap between each pair of initial line segments. For convenience, we call the initial line segments resulting from edge detection the *detected segments* and the newly constructed ones the *gap-filling segments*. A boundary is then defined as a simple cycle that traverses a set of detected and gap-filling segments *alternately*, as shown in Fig. 2(d). Finally, a grouping cost function is defined for the boundaries and an algorithm is developed to find from all boundaries the one with minimum grouping cost, also as shown in Fig. 2(d). Usually, the grouping cost is defined to be a function of some local weights associated to each individual detected/gap-filling segment. As mentioned above, various grouping cues, such as proximity, closure, continuity, and convexity have been incorporated into edge grouping. However, it is difficult to incorporate symmetry as a cue directly into edge grouping because symmetry can not be encoded into an individual line segment. In the next section, we construct a new grouping token to address this problem.

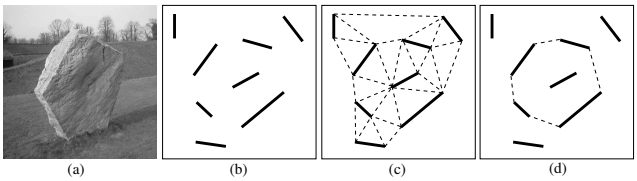

Figure 2. An illustration of the process of edge grouping.

2.2. Symmetric Trapezoids as Grouping Tokens

While it is difficult to encode symmetry into each individual segment, it can be encoded into a pair of segments.

For each pair of *detected* segments P_1P_2 and P_3P_4, as shown in Fig. 3, we identify their symmetric portions by following three steps:

1. As shown in Fig. 3(a), find the angle-bisector line l between P_1P_2 and P_3P_4. If $P_1P_2//P_3P_4$, then l is the line equidistant to both P_1P_2 and P_3P_4.

2. Find the projections of both segments P_1P_2 and P_3P_4 to l and denote them $P_1'P_2'$ and $P_3'P_4'$. The overlap of segments $P_1'P_2'$ and $P_3'P_4'$, as shown by $P_1'P_2'$ in Fig. 3(b), is denoted as the *axis segment* between P_1P_2 and P_3P_4.

3. Backproject this axis segment to segments P_1P_2 and P_3P_4, we achieve a *(symmetric) trapezoid* as shown in Fig. 3(c).

For each pair of detected line segments, we construct such a symmetric trapezoid (with its axis segment) as the new grouping token. Note that, for some pairs of line segments, their projections to l have no overlap. In this case, no symmetric trapezoid will be constructed for this pair of segments.

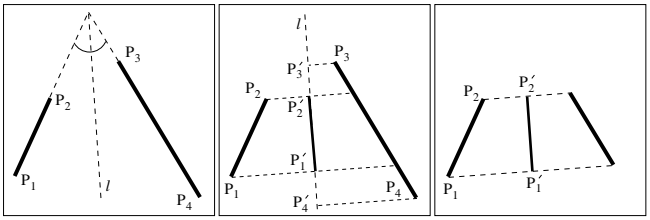

Figure 3. An illustration of constructing a symmetric trapezoid from a pair of detected line segments.

2.3. Trapezoid Grouping and Grouping Cost

Analogous to edge grouping, we need to connect a sequence of trapezoids by filling some gaps using *gap-filling quadrilaterals* to achieve a closed boundary. Two examples are shown in Figs. 4(a) and (b), where two gap-filling quadrilaterals $\mathcal{G}_1 = \{P_2P_3P_{10}P_{11}\}$ and $\mathcal{G}_2 = \{P_4P_5P_8P_9\}$ are constructed to connect three trapezoids $\mathcal{T}_1 = \{P_1P_2P_{11}P_{12}\}$, $\mathcal{T}_2 = \{P_3P_4P_9P_{10}\}$, and $\mathcal{T}_3 = \{P_5P_6P_7P_8\}$ into a closed polygonal boundary $P_1P_2 \dots P_{12}$. As in edge grouping, this closed boundary alternately traverses some solid lines, which correspond to the detected segments, and some dashed lines, which correspond to the gap-filling segments. The axis segments of $\mathcal{T}_1$, $\mathcal{T}_2$, and $\mathcal{T}_3$, shown as solid lines, are also connected by the axis segments of the gap-filling quadrilaterals $\mathcal{G}_1$ and $\mathcal{G}_2$, shown as dashed lines, to generate the polyline $Q_1Q_2 \dots Q_6$, which is the *axis* of the resulting boundary. Note that the gap-filling quadrilaterals may not be symmetric and its axis segment is constructed simply by connecting the endpoints of the axis segments of the neighboring symmetric trapezoids. As in edge grouping, we construct gap-filling quadrilaterals between each pair of trapezoids. Since the axis segment of a trapezoid has two enpoints, there are four different gap-filling quadrilaterals that can be constructed between a pair of trapezoids.

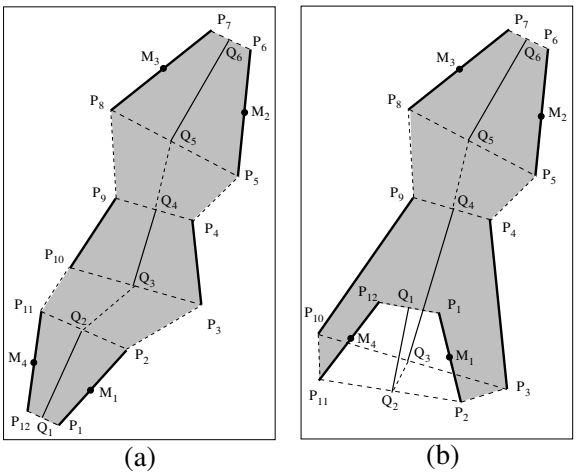

Figure 4. Two examples of grouping detected trapezoids into a closed boundary.

Since our goal is to detect symmetric boundaries, we are only interested in the boundaries with an axis, as shown in Fig. 4. In this paper, we define the grouping cost function for such a closed boundary $\mathcal{B}$ as

$$\phi(\mathcal{B}) = \frac{|\mathcal{B}_D| + \lambda_1 \cdot \int_{\mathcal{B}} \kappa^2(t)dt + \lambda_2 \cdot \rho(\texttt{axis}(\mathcal{B}))}{\texttt{area}(\mathcal{B})}, \quad (1)$$

where $|\mathcal{B}_D|$ is the total gap length along the boundary $\mathcal{B}$, *e.g.*, the length of the dashed segments along the polygon $P_1P_2 \dots P_{12}$ in Fig. 4. This term reflects the cue of proximity. $\kappa(t)$ is the curvature along the boundary. This term reflects the cue of continuity. $\rho(\texttt{axis}(\mathcal{B}))$ is a measure negatively related to the collinearity (straightness) of $\mathcal{B}$'s axis (*e.g.*, the polylines $Q_1Q_2 \dots Q_6$ in Fig. 4). This term reflects the cue of symmetry. $\texttt{area}(\mathcal{B})$ is the region area enclosed by the boundary $\mathcal{B}$. A normalization over this term sets a preference to produce larger structures, which improves the robustness against image noise. $\lambda_1, \lambda_2 > 0$ are two preset factors that balance the weights of these cues. Note that direct estimation of the curvature along the polygonal boundary makes no sense. In practice, we use Bezier curves to interpolate the polygonal boundary and then estimate its curvature, which well reflects the continuity (smoothness) of the boundary. This will be discussed in more detail in the next section.

3. Graph Modeling and Algorithm

In this section, we describe a graph algorithm to locate the boundary that minimizes the grouping cost in Eq. (1) in a globally optimal fashion.

3.1. Graph Construction I: Solid/Dashed Edges

We construct an undirected graph $G = (V, E)$ with vertex set V and edge set E to model the grouping tokens. Since each trapezoid or gap-filling quadrilateral has a unique axis segment, we can represent them by their axis segments. A straightforward idea is then to construct a vertex in G for each endpoint of axis segment, *e.g.*, $Q_i, i = 1, 2, \ldots, 6$ in Figs. 4(a) and (b), and construct an edge in G for each axis segment. However, this graph construction is insufficient to handle our problem since the grouping cost includes a normalization by the area of the enclosed region, i.e., $\texttt{area}(\mathcal{B})$ in Eq. (1). From Fig. 4(b), we can see this enclosed area may not be the simple summation of the areas of the traversed trapezoids and quadrilaterals. When the resulting boundary is nonconvex, an inclusion of a trapezoid (or a quadrilateral), such as $\mathcal{T}_1 = \{P_1P_2P_{11}P_{12}\}$ in Fig. 4(b), may decrease the total area of the resulting enclosed region.

Therefore, for each axis segment we are going to construct a pair of edges, one of them represents the case where the corresponding trapezoid or quadrilateral has a positive area while the other represents the case where the corresponding trapezoid or quadrilateral has a negative area. To distinguish these two cases, we write this pair of edges as e^+ and e^- for the cases of positive and negative areas respectively, and call them the *mirror* edges of each other. Accordingly, we actually construct two vertices for each axis-segment endpoint, as shown in Fig. 5. Furthermore, we distinguish the edges constructed for trapezoids and gap-filling quadrilaterals: the two edges corresponding to a trapezoid are set to be solid edges and the two edges corresponding to a gap-filling quadrilateral are set to be dashed edges. One problem in graph construction is then to determine the edge connection since each axis segment is represented by two edges. For example, in Fig. 6, trapezoid $\mathcal{T} = \{P_1P_2P_3P_6\}$ and gap-filling quadrilateral $\mathcal{G} = \{P_3P_4P_5P_6\}$ are adjacent, the solid edge pair constructed for $\mathcal{T}$ is e_T^+ and e_T^-, and the dashed edge pair constructed for $\mathcal{G}$ is e_G^+ and e_G^-, we need to decide whether e_T^+ is connected to e_G^+ or e_G^-.

In this paper, we use the following simple strategy to decide this edge-connection relations. We further set edge e^+ to represent its corresponding trapezoid or quadrilateral with a clockwise direction and e^- to represent its corresponding trapezoid or quadrilateral with a counterclockwise direction. Then the edge connection should make the line segments along the resulting boundary to be consistent, either clockwise or counterclockwise. As shown in Fig. 6, we need to consider three cases (shown for when $\mathcal{T}$ is clockwise, but it is similar for counter-clockwise case). (a) If the clockwise trapezoid should be connected to the clockwise quadrilateral, we connect their edges with same signs, as shown in Fig. 6(a). (b) If the clockwise trapezoid should be connected to the counterclockwise quadrilateral, we connect the edges of opposite signs, as shown in Fig. 6(b). (c) If the quadrilateral has a self-crossing, as shown in Fig. 6(c), it is nontrivial to determine whether its direction is clockwise or counterclockwise, since two resulting triangles, R_1 and R_2, always have opposite directions. In this case, we calculate the area of this quadrilateral as $|R_1 - R_2|$, the area difference between the two resulting triangles, set its direction to be the same as the direction of the larger triangle, and then follow the above two cases to decide the edge connection.

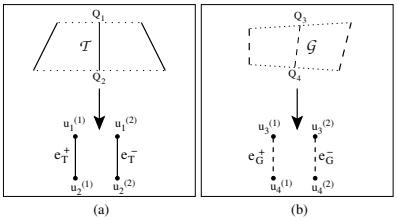

Figure 5. An illustration of the solid/dashed edges. (a) For a trapezoid $\mathcal{T}$, we construct a pair of solid edges e_T^+ and e_T^-. Each axis-segment endpoint Q_i corresponds to two vertices $u_i^{(1)}$ and $u_i^{(2)}$. (b) For a gap-filling quadrilateral $\mathcal{G}$, we proceed similarly as in (a), except that the two constructed edges are dashed.

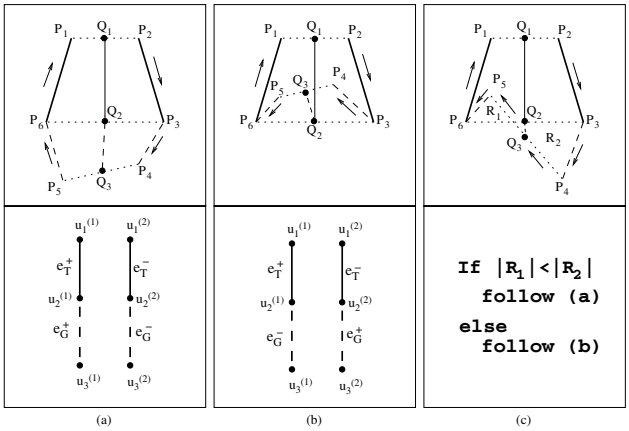

Figure 6. An illustration of the three cases of edge connection when a trapezoid $\mathcal{T}$ is connected to a quadrilateral $\mathcal{G}$. Each axis-segment endpoint Q_i corresponds to two vertices $u_i^{(1)}$ and $u_i^{(2)}$, $i = 1, 2, 3$.

For each edge e in this graph, we define two weight functions $w_1(e)$ and $w_2(e)$. We define $w_1(e) = 0$ if e is solid, and if e is dashed, *e.g.* corresponding to axis fragment Q_2Q_3 in Fig. 7, we define

$$w_1(e) = |P_3P_4| + |P_7P_8| + \lambda_1 \cdot (\kappa_1 + \kappa_2) + \lambda_2 \cdot \rho(e)$$

where $|P_3P_4| + |P_7P_8|$ is the gap length (along the

boundary) that results from the quadrilateral corresponding to e. $\kappa_1 = \int_{\mathcal{Z}(M_1,P_8,P_7,M_4)} \kappa^2(t)dt$ and $\kappa_2 = \int_{\mathcal{Z}(M_2,P_3,P_4,M_3)} \kappa^2(t)dt$, with $\mathcal{Z}(\cdot,\cdot,\cdot,\cdot)$ being the Bezier curve constructed from four control points and $M_i, i = 1,2,3,4$ being the midpoints of the respective side edges of the two neighboring trapezoids, as shown in Fig. 7. $\rho(e) = |\sin(\angle Q_1Q_2Q_3)| + |\sin(\angle Q_2Q_3Q_4)|$ is a measure negatively related to the collinearity (straightness) of the polyline $Q_1Q_2Q_3Q_4$, as shown in Fig. 7. We can see that the first edge weight $w_1(e)$ is always nonnegative and for a pair of mirror edges e^+ and e^-, we have $w_1(e^+) = w_1(e^-)$. The second edge weight, $w_2(e)$, is simply the signed area of the trapezoid or quadrilateral corresponding to e, i.e., $w_2(e^+) = -w_2(e^-) > 0$.

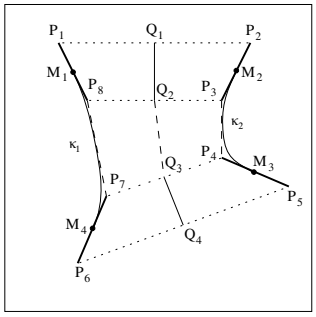

Figure 7. An illustration of defining the weights for the dashed edges corresponding to the quadrilateral $\mathcal{G} = \{P_3P_4P_7P_8\}$. This quadrilateral fills the gap between two trapezoids $\mathcal{T}_1 = \{P_1P_2P_3P_8\}$ and $\mathcal{T}_2 = \{P_4P_5P_6P_7\}$.

Note that each closed boundary $\mathcal{B}$ is uniquely defined by its axis, which is described by two *mirror* paths $\mathcal{P}^+$ and $\mathcal{P}^-$ in the constructed graph. Here $\mathcal{P}^-$ consists of all the mirror edges of the ones in $\mathcal{P}^+$. Both paths traverse solid and dashed edges alternately and start and end with a solid edge. Two examples are shown in Fig. 8. In fact, the total second weights along $\mathcal{P}^+$ and $\mathcal{P}^-$ exactly reflect the denominator of the grouping cost (1), i.e., the total area enclosed by $\mathcal{B}$, but with opposite signs. For convenience, we simply denote $\sum_{e\in\mathcal{P}^+} w_2(e) = -\sum_{e\in\mathcal{P}^-} w_2(e) = \texttt{area}(\mathcal{B}) > 0$. For the total first weights of the edges along these two paths, we have $\sum_{e\in\mathcal{P}^+} w_1(e) = \sum_{e\in\mathcal{P}^-} w_1(e)$, which reflects the numerator of the grouping cost (1) on boundary $\mathcal{B}$ except for the portions around the axis endpoints. For example, for the boundary $P_1P_2\ldots P_{12}$ shown in Fig. 4(a) (or (b)), the length and curvature of the gaps P_6P_7 and P_1P_{12} are missed when we use the total first weight along the path to describe the numerator of the grouping cost (1).

3.2. Graph Construction II: Auxiliary Edges

To include the gap length and curvature information around the axis endpoints, we construct a set of new dashed edges, named *auxiliary* edges, between the vertices corresponding to the axis endpoints. Particularly, let $u_1^{(1)}$ and $u_1^{(2)}$ be the vertex pair corresponding to an axis endpoint, *e.g.*, Q_1 in Fig. 4(a) (or (b)), and $u_6^{(1)}$ and $u_6^{(2)}$ be the vertex pair corresponding to the other axis endpoint, *e.g.*, Q_6 in Fig. 4(a) (or (b)). We simply construct four dashed edges $(u_1^{(i)}, u_6^{(j)}), i,j = 1,2$ as auxiliary edges (Fig. 8). For these four auxiliary edges, we set their second edge weights to be zero and the first edge weights to be the total gap length and curvature of the boundary portions around these two axis endpoints. For example, in Fig. 4(a) (or (b)), the first weight $w_1(e)$ of the auxiliary edges that reflects the connection between Q_1 and Q_6 is defined by

$$\begin{aligned} w_1(e) &= |P_1P_{12}| + \lambda_1 \int_{\mathcal{Z}(M_1,P_1,P_{12},M_4)} \kappa^2(t)dt \\ &+ |P_6P_7| + \lambda_1 \int_{\mathcal{Z}(M_2,P_6,P_7,M_3)} \kappa^2(t)dt, \end{aligned}$$

where $M_i, i = 1,2,3,4$ are the midpoints of the side edges of the two involved trapezoids, as shown in Fig. 4(a) (or (b)). Since the axis and axis endpoints of the optimal boundary are unknown before the grouping, we can treat each axis-segment endpoint as a possible axis endpoint and construct auxiliary edges between each possible pair of axis-segment endpoints. By including an auxiliary edge, the two mirror paths $\mathcal{P}^+$ and $\mathcal{P}^-$ discussed in the last section are expanded to two mirror cycles $\mathcal{C}^+$ and $\mathcal{C}^-$, which represent a unique boundary $\mathcal{B}$. Note that these two cycles traverse solid and dashed edges alternately and therefore, we call them *alternate* cycles. A simple analysis shows that the to-

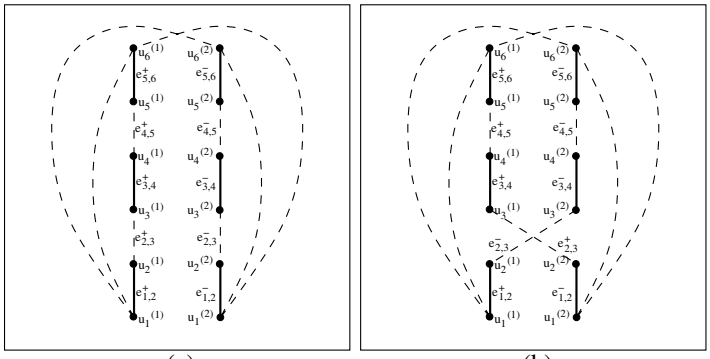

Figure 8. An illustration of the paths and cycles corresponding to the boundaries and their axes shown in Figs. 4(a) and (b), respectively. For the boundary in Fig. 4(a), the corresponding path $\mathcal{P}^+$ traverses $u_i^{(1)}, i = 1,2,\ldots,6$ sequentially and $\mathcal{P}^-$ traverses $u_i^{(2)}, i = 1,2,\ldots,6$ sequentially. For the boundary shown in Fig. 4(b), the corresponding path $\mathcal{P}^+$ traverses $u_1^{(2)}, u_2^{(2)}, u_3^{(1)}, u_4^{(1)}, u_5^{(1)}, u_6^{(1)}$ sequentially and $\mathcal{P}^-$ traverses $u_1^{(1)}, u_2^{(1)}, u_3^{(2)}, u_4^{(2)}, u_5^{(2)}, u_6^{(2)}$ sequentially. Vertices $u_i^{(1)}$ and $u_i^{(2)}$ correspond to the axis-segment endpoint Q_i, $i = 1,2,\ldots,6$ in Fig. 4. Auxiliary edges are shown by curved dashed lines.

tal first edge weights along both $\mathcal{C}^+$ or $\mathcal{C}^-$ equals to the numerator of the cost $\phi(\mathcal{B})$ in Eq. (1). This way, we have

$$\phi(\mathcal{B}) = \frac{\sum_{e\in\mathcal{C}^+} w_1(e)}{\sum_{e\in\mathcal{C}^+} w_2(e)} = -\frac{\sum_{e\in\mathcal{C}^-} w_1(e)}{\sum_{e\in\mathcal{C}^-} w_2(e)},$$

and locating the optimal boundary $\mathcal{B}$ that minimizes the cost (1) is reduced to finding an alternate cycle $\mathcal{C}$ in our constructed graph such that this cycle $\mathcal{C}$ minimizes the cycle ratio

$$\varphi(\mathcal{C}) = \frac{W_2(\mathcal{C})}{W_1(\mathcal{C})} = \frac{\sum_{e\in\mathcal{C}} w_2(e)}{\sum_{e\in\mathcal{C}} w_1(e)}. \quad (2)$$

The correctness of this reduction comes from the fact that any alternate cycle $\mathcal{C}$ has two mirror versions $\mathcal{C}^+$ and $\mathcal{C}^-$ with opposite signs on the total second weights $W_2(\mathcal{C})$. Therefore, the alternate cycle with the minimal cycle ratio (2) must be of a $\mathcal{C}^-$ version, i.e., it has negative cycle ratio $\varphi(\mathcal{C})$ and negative total second weights $W_2(\mathcal{C})$. This way, for the optimal boundary $\mathcal{B}$, we have $\phi(\mathcal{B}) = -\frac{1}{\varphi(\mathcal{C})}$ and therefore, the alternate cycle $\mathcal{C}$ with the minimum cycle ratio $\varphi(\mathcal{C})$ corresponds to the boundary $\mathcal{B}$ with the minimum cost $\phi(\mathcal{B})$.

One particular concern is that the resulting optimal alternate cycle $\mathcal{C}$ should not contain more than one auxiliary edge. Otherwise, the resulting boundary will contain multiple separate closed boundaries with unaligned axes. Fortunately, this case will not happen and the optimal alternate cycle $\mathcal{C}$ contains no more than one auxiliary edge. This can be proved by contradiction. Assume the resulting $\mathcal{C}$ contains k auxiliary edges. This means $\mathcal{C}$ contains k alternate paths $\mathcal{P}_1$, $\mathcal{P}_2$, and $\mathcal{P}_k$ after these k auxiliary edges are removed. From these k paths, we can construct k new alternate cycles $\mathcal{C}_1$, $\mathcal{C}_2$, and $\mathcal{C}_k$ by including the auxiliary edges between the two endpoints of each path. Given the cycle ratio defined in Eq. (2), it is not difficult to see that $\varphi(\mathcal{C}) = \frac{W_2(\mathcal{C})}{W_1(\mathcal{C})} = \frac{\sum_{i=1}^{k} W_2(\mathcal{C}_i)}{\sum_{i=1}^{k} W_1(\mathcal{C}_i)}$. This tells us that at least one cycle, say $\mathcal{C}_m$, out of $\mathcal{C}_i, i = 1, 2, \ldots, k$, has a smaller cycle ratio than $\mathcal{C}$, i.e., $\varphi(\mathcal{C}_m) = \frac{W_2(\mathcal{C}_m)}{W_1(\mathcal{C}_m)} \leq \varphi(\mathcal{C})$. This contradicts the assumption.

Finally, we only need to solve the graph problem of finding an alternate cycle with minimum cycle ratio in an undirected graph. This problem can be solved in polynomial time in a globally optimal fashion [1, 24].

4. Experiments

In our implementation, we use the ratio-contour package by [24] to solve the problem of finding the optimal alternate cycle that minimizes the cycle ratio $\varphi(\mathcal{C})$ defined in Eq. (2). Initial line segments are obtained by sequentially performing Canny edge detection and line fitting. Specifically, we use the Canny edge detector provided in the Matlab (7.1.0.183-R14 with SP3) image processing toolbox, using its default parameter values. For line fitting, we used the package by Kovesi [15], in which we set the minimum length of an edge to be processed to be 30 pixels, and the maximum deviation between an edge and its fitted line to be 2 pixels. In the grouping cost $\phi(\mathcal{B})$ in Eq. (1), we set $\lambda_1 = 0.5$ and $\lambda_2 = 0.5$ in all our experiments.

The next step is the construction of gap-filling quadrilaterals. Since each trapezoid axis segment has two endpoints, ideally we need to construct four possible gap-filling quadrilaterals between two trapezoids. However, two of them would introduce a self-crossing of the resulting boundary, and are therefore discarded. Given n detected line segments from the original image, we may have $O(n^2)$ trapezoids, and considering the construction of gap-filling quadrilaterals between each pair of trapezoids will generate $O(n^4)$ quadrilaterals. This is very computationally intensive. In our implementation, we apply the following three strategies to reduce the number of gap-filling quadrilaterals.

First, if the axis segment of a gap-filling quadrilateral is too long, say, longer than a given threshold, we are not going to include this quadrilateral since this indicates the case of a gap between two neighboring trapezoids to be too long, as shown in Fig. 9(a), and therefore, they are not likely to be directly connected to form a boundary. We set the this threshold to be 15 pixels in our experiments. Second, if a quadrilateral's two sides (in which its axis-segment endpoints lie) have substantially different length, we are not going to include this quadrilateral since any boundary traversing this quadrilateral will have poor continuity. An example is shown in Fig. 9(b), where, if $\frac{min(d_1,d_2)}{max(d_1,d_2)}$ is smaller than a given threshold, the quadrilateral $\mathcal{G}$ will not be constructed. This threshold is set to be 0.7 in the experiments. Finally, if collinearity of the polyline formed by the axis segments of a quadrilateral and its two neighboring trapezoids is poor, we are not going to include this quadrilateral. Particulary,as shown in Fig. 9(c), we require $|\sin(\angle Q_1Q_2Q_3)| \leq 0.5$ and $|\sin(\angle Q_2Q_3Q_4)| \leq 0.5$ to be a condition to include the quadrilateral $\mathcal{G}$.

In Section 3.2, we discuss the graph construction by adding auxiliary edges between the vertices corresponding to the pair of possible axis endpoints, which include the axis-segment endpoints of all the trapezoid. This way, given n^2 trapezoids, we may need to construct $O(n^4)$ auxiliary edges, which can be a very large number and introduce a huge computational load. In our implementation, we consider an axis-segment endpoint to be a possible axis endpoint only if the gap-filling segment it is located at is sufficiently short. For example, in Fig. 4(a), Q_1 is considered to be a possible axis endpoints if the length of the segment P_1P_{12} is smaller than a given threshold, which is set to be 10 pixels in our implementation. The auxiliary edges are only constructed between these identified possible axis end-

points.

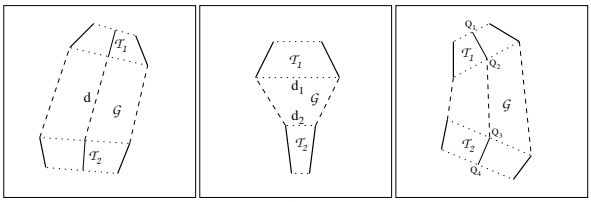

Figure 9. An illustration of the three strategies that are used to reduce the number of quadrilaterals.

Figures 10 and 11 show the optimal boundaries and their axes detected from 20 real images using the proposed method. Our test images include medical images, images of man-made structures, and images of nature scenes. This experiment clearly shows that the proposed method well integrates the cue of symmetry into the boundary grouping and in many images it can detect the symmetric structures (or symmetric components of these structures, such as Figs. 11(a-c,f,i)) from the noisy real images. Incorporating the region-area information also plays an important role in the proposed grouping. From the cost function in Eq. (1), it can be seen that, if we remove the region-normalization term, the optimal boundary may only bound a small-area region, such as the degenerate case of containing only one small detected trapezoid. Although we only consider region area information in the proposed method, other region information, such as the intensity homogeneity in the bounded region, can be incorporated into the cost function in a similar way [13]. We can also find examples where the detected boundary contains trapezoids with both positive and negative areas, such as Figs. 11(a) and (j). Figure 10(j) shows that the proposed method can handle images where the symmetric structure is partially occluded.

5. Conclusions

In this paper, we developed a new grouping method for detecting salient closed boundaries that show certain symmetry. In this method, we paired the detected line segments into a new type of grouping token called trapezoid, which encodes the symmetry. Based on these trapezoid tokens, we developed a new grouping cost that incorporates the cues of proximity, closure, continuity, and symmetry. Furthermore, we introduce a region-area term to normalize this grouping cost to avoid detecting overly small structures. We then constructed a graph model and developed a graph algorithm to find the optimal boundary with the minimum grouping cost in a globally optimal fashion. Experiments on a set of real images were conducted to demonstrate the performance of the proposed method.

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

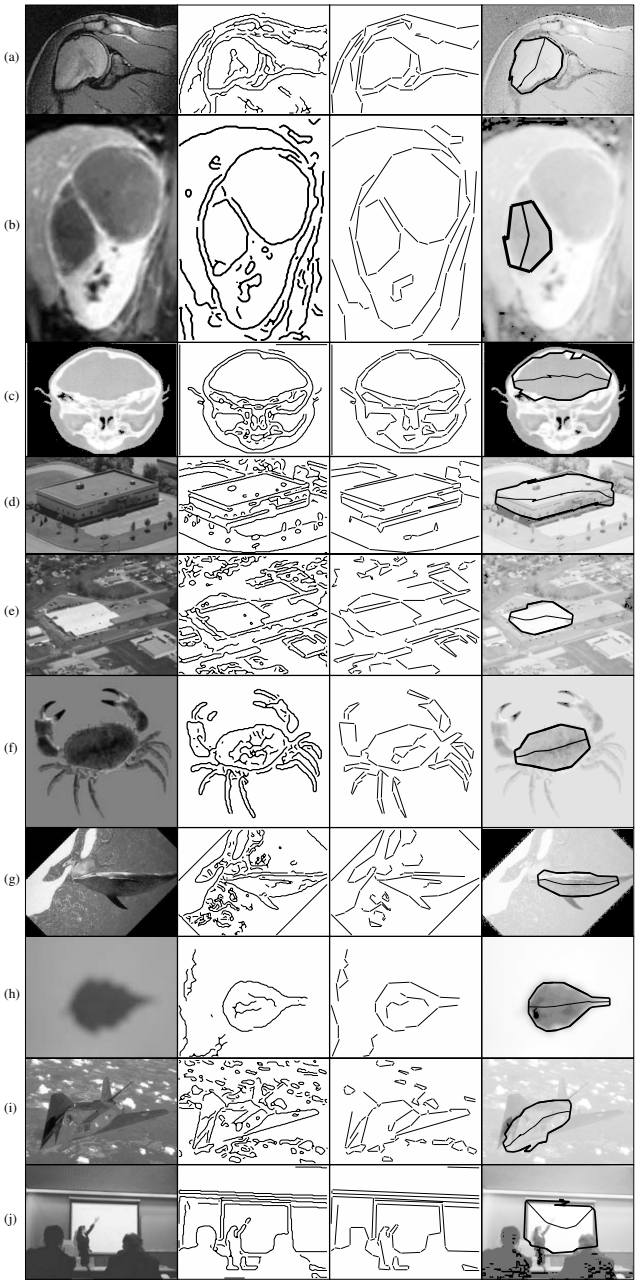

Figure 10. Boundaries and their axes detected on some real images. Column 1: the input images; Column 2: the edge-detection outputs; Column 3: the line-fitting outputs; Column 4: the detected closed boundaries with their symmetry axes.

References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, & Applications*. Prentice Hall, Englewood Cliffs, 1993.

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

[3] A. Amir and M. Lindenbaum. A generic grouping algorithm

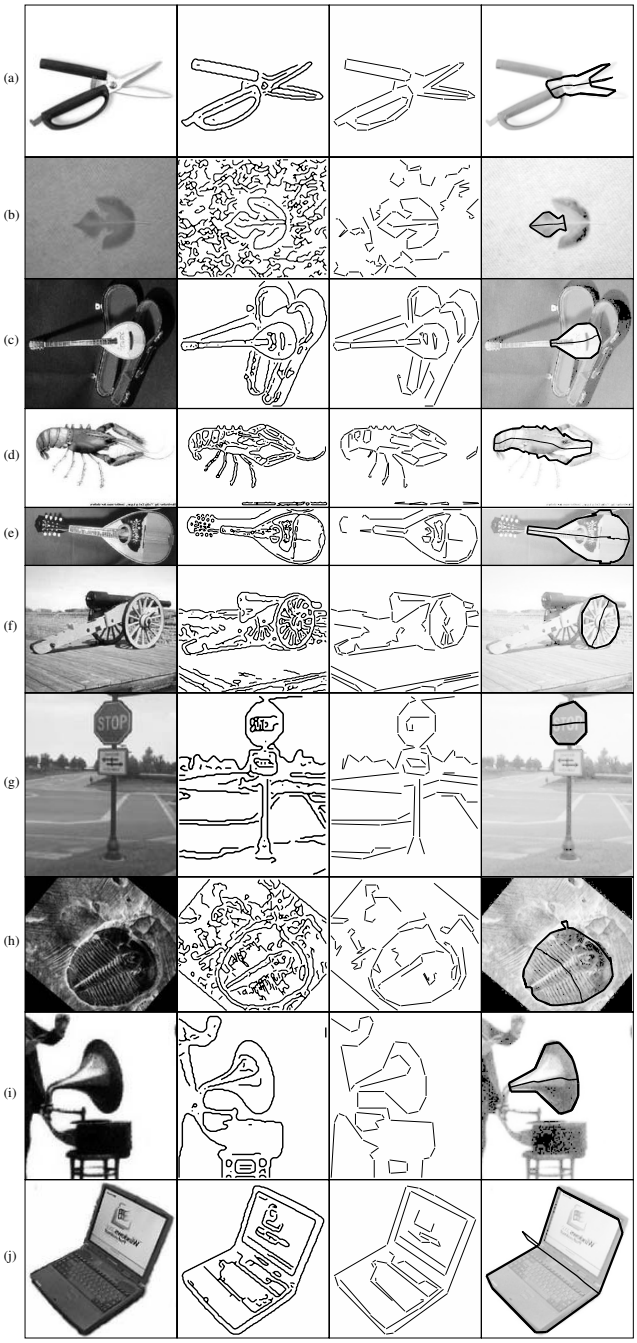

Figure 11. Boundaries and their axes detected on some real images (continued). The columns depict the same information as in Fig. 10.

and its quantitative analysis. *PAMI*, 20(2):168–185, 1998.

[4] H. Blum. Biological shape and visual science. *Journal of Theoretical Biology*, 38:205–287, 1973.

[5] H. Blum and R. N. Nagle. Shape description using weighted symmetric axis features. *Pattern Recognition*, 10:167–180, 1978.

[6] M. J. Brady and H. Asada. Smoothed local symmetries and their implementation. Technical Report AIM-757, Massachusetts Institute of Technology, February 1984.

[7] J. H. Elder, A. Krupnik, and L. A. Johnston. Contour grouping with prior models. *PAMI*, 25(6):661–674, 2003.

[8] J. H. Elder and S. W. Zucker. Computing contour closure. In *ECCV*, pages 399–412, 1996.

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

[10] G. Guy and G. Medioni. Inferring global perceptual contours from local features. *IJCV*, 20(1):113–133, 1996.

[11] D. Huttenlocher and P. Wayner. Finding convex edge groupings in an image. *IJCV*, 8(1):7–29, 1992.

[12] D. Jacobs. Robust and efficient detection of convex groups. *PAMI*, 18(1):23–27, 1996.

[13] I. H. Jermyn and H. Ishikawa. Globally optimal regions and boundaries as minimum ratio cycles. *PAMI*, 23(10):1075–1088, 2001.

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

[15] 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/.

[16] M. Leyton. *Symmetry, Causality, Mind*. MIT Press, Cambridge, 1992.

[17] T. Liu, D. Geiger, and A. L. Yuille. Segmenting by seeking the symmetry axis. In *CVPR*, pages 994–998, 1998.

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

[19] S. Mahamud, L. R. Williams, K. K. Thornber, and K. Xu. Segmentation of multiple salient closed contours from real images. *PAMI*, 25(4):433–444, 2003.

[20] R. Mohan. Perceptual organization for scene segmentation and description. *PAMI*, 14(6):616–635, 1992.

[21] S. Sarkar and K. Boyer. Quantitative measures of change based on feature organization: Eigenvalues and eigenvectors. In *CVPR*, pages 478–483, 1996.

[22] A. Shashua and S. Ullman. Structural saliency: The detection of globally salient structures using a locally connected network. In *ICCV*, pages 321–327, 1988.

[23] J. S. Stahl and S. Wang. Convex grouping combining boundary and region information. In *ICCV*, volume 2, pages 946–953, 2005.

[24] S. Wang, T. Kubota, J. Siskind, and J. Wang. Salient closed boundary extraction with ratio contour. *PAMI*, 27(4):546–561, 2005.

[25] L. Williams and K. K. Thornber. A comparison measures for detecting natural shapes in cluttered background. *IJCV*, 34(2/3):81–96, 2000.

[26] A. P. Witkin and J. M. Tenenbaum. On the role of structure in vision. In J. Beck, B. Hope, and A. Rosenfeld, editors, *Human and Machine Vision*, pages 481–543. Academic Press, New York, 1983.

[27] S. C. Zhu and A. Yuille. FORMS: a flexible object recognition and modelling system. In *ICCV*, pages 465–472, 1995.