Convex Grouping Combining Boundary and Region Information

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

Convexity is an important geometric property of many natural and man-made structures. Prior research has shown that it is imperative to many perceptual-organization and image-understanding tasks. This paper presents a new grouping method for detecting convex structures from noisy images in a globally optimal fashion. Particularly, this method combines both region and boundary information: the detected structural boundary is closed and well aligned with detected edges while the enclosed region has good intensity homogeneity. We introduce a ratio-form cost function for measuring the structural desirability, which avoids a possible bias to detect small structures. A new fragment-pruning algorithm is developed to achieve the structural convexity. The proposed method can also be extended to detect open boundaries, which correspond to the structures that are partially cropped by the image perimeter, and incorporate a human-computer interaction for detecting a convex boundary around a specified point. We test the proposed method on a set of real images and compare it with the Jacobs' convex-grouping method.

1. Introduction

This paper is concerned with *perceptual organization*, or *grouping*, which aims at detecting desirable structures from noisy images in a bottom-up way. As one of the most challenging problems in computer vision, grouping suffers from the difficulties of (a) how to define the *desirability* of a grouping, and (b) how to find an effective algorithm to accomplish the grouping with a maximum desirability. For different applications, the grouping desirability needs to be defined to incorporate different factors. For example, in general-purpose image understanding, the grouping desirability can usually be described by some psychological rules, such as Gestalt laws of closure, continuity, parallelism, symmetry, and so on. In specific applications, such as face detection, specific knowledge, such as the shape of the face, needs to be incorporated into the definition of the desirability to achieve a more reliable and useful grouping. Considering different factors in the definition of the desirability, a large number of grouping methods have been developed in the past decades [20, 1, 11, 18, 9, 6, 2, 23, 15, 22, 7, 19].

In this paper, our major goal is to find a way to integrate the structural *convexity* into the general-purpose grouping, i.e., to detect convex structures from real images. Several reasons make the convexity an important factor in grouping. First, in many applications the desirable structures may be *a priori* known to be convex, such as detecting the surface of some furniture, the silhouette of some buildings, and many kinds of fruits. In these cases, we have to explicitly enforce the convexity as a constraint in the grouping. Second, even if the desired structure is not fully convex, it may consist of several convex components and a grouping with a convexity constraint may produce more useful and accurate results. For example, Borra and Sarkar [3] compare several grouping methods in a recognition task and find that the grouping with the convexity constraint [11] leads to the best recognition performance, although the underlying structures are not convex. Third, convexity itself has been identified as a very important property in measuring the desirability of the general-purpose grouping. For example, Liu, Jacobs, and Basri [14] conduct several psychophysical experiments with observations that the convexity may dominate other Gestalt laws, such as continuity and closure, in many cases. Jacobs [11] shows that convexity is nonaccidental in real images and can be used to distinguish the perceptually salient structures from the image noise. Given its important role, convexity has been considered in grouping by many researchers [13, 10, 11, 16, 8].

One state-of-the-art convex-grouping method was developed by Jacobs [11] in the form of *edge grouping*. In this method, a set of disconnected line segments are first detected from an image by edge detection and line fitting. This method then identifies a subset of these line segments and connects them sequentially into a closed boundary in the form of a convex polygon. The grouping desirability, or the *saliency* of the resultant convex boundary in this case, is measured by the boundary proximity, i.e., the proportion of the detected line segments along the perimeter of the resultant boundary. Clearly, the larger this proportion, the better the resultant boundary is aligned with the detected line segments. In Jacobs' method, a sequential-search algorithm is developed to detect all the convex closed boundaries with the saliency larger than a given threshold. While this search algorithm has an exponential time complexity in the worst case, it has a polynomial-time complexity on average. Other re-

cent edge-grouping methods include [23, 15, 22]. However, these methods do not consider the convexity constraint. Note that these methods, as well as Jacobs' method, only consider the boundary geometry, such as the Gestalt laws of proximity, closure and continuity, but not the information within the region enclosed by the resultant boundary.

Prior research also shows that both boundary and region information are important to achieve a successful grouping. A grouping method only considering boundary geometry is usually sensitive to image noise while a grouping method only considering region information usually has difficulty in incorporating useful boundary geometries. In the past decades, many approaches have been developed to integrate boundary and region information into grouping, such as the *ratio-region* method [5], Jermyn and Ishikawa's method [12], and the *normalized-cut* method [21]. Recently, active-contour and level-set methods were extended to detect smooth boundaries that enclose regions with homogeneous image features [4]. However, as far as we know, none of these methods considers the factor of convexity and it is also not trivial to extend any of these methods to detect only convex structures.

In this paper, we present a new *edge-grouping* method to detect salient convex structures from real images by integrating boundary and region information. In general, the proposed method follows the usual edge-grouping process, i.e., the resultant structural boundary is formed by identifying and connecting a subset of the line segments detected from the input image. However, in measuring the structural saliency, we incorporate not only the boundary proximity defined in Jacobs' method [11], but also the pixel-intensity homogeneity within the enclosed region. To locate the convex boundary that maximizes this integrated saliency measure, we propose a new algorithm consisting of a novel fragment-pruning algorithm and the ratio-contour algorithm developed in [22]. The proposed algorithm is of polynomial time complexity in the worst case and guarantees the global optimality in terms of the defined saliency measure.

The remainder of this paper is organized as follows. In Section 2, we formulate the convex-grouping problem by considering both boundary and region information. In Section 3, we present the fragment-pruning algorithm and use it to solve the problem formulated in Section 2. Section 4 presents a set of experimental results on real images. These results are compared to the convex-grouping results from Jacobs' method. Section 5 further discusses two extensions to the proposed method to detect open-boundary structures and incorporate a simple human-computer interaction. Section 6 briefly concludes the paper.

2. Problem Formulation

As in other edge-grouping methods [11], we first detect a set of line segments from the image by edge detection and line fitting. We refer to these line segments as *detected fragments*. From these detected fragments, a valid closed boundary is defined as a polygon that is constructed by sequentially connecting a subset of those fragments, as shown in Fig. 1(c). In order to form this boundary, a new set of straight-line segments (dashed lines in Fig. 1(c)) needs to be constructed to fill the gap between the neighboring detected fragments. We refer to these dashed line segments as *gap-filling fragments*. We can see that, under this formulation, a boundary is a polygon that traverses a subset of detected and gap-filling fragments alternately. In the proposed method, we construct gap-filling fragments between each possible pair of detected fragments and then search for an optimal closed boundary.

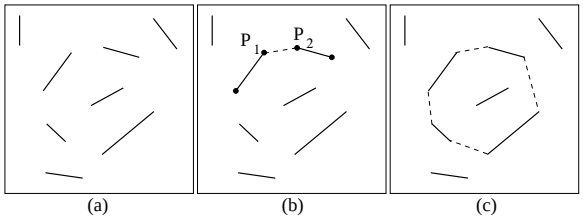

Figure 1: An illustration of the edge grouping for a closed boundary. (a) Detected fragments; (b) A gap-filling fragment (dashed line) connecting two detected-fragment endpoints P_1 and P_2. (c) A closed boundary that traverses detected and gap-filling fragments alternately.

The key problem is then to define a saliency measure for each valid closed boundary. As mentioned in Section 1, we have two important considerations in defining this saliency measure. First, the resultant boundary must be convex, which will be incorporated as a hard constraint in this measure. Second, we are going to integrate both boundary and region information into grouping, i.e., the intensity homogeneity within the region enclosed by the resultant boundary will be considered as well as the boundary geometry of the resultant boundary. With these two considerations, we define the cost (negatively related to the *saliency*) of the resultant closed boundary B by

$$\phi(B) = \frac{\int_B \sigma(t)dt + \lambda \cdot \iint_R |\nabla(I(u,v))|dudv}{\iint_R dudv} \qquad (1)$$

subject to a constraint that B must be convex. Here R is the region enclosed by the convex closed boundary B. $\sigma(t)$ is an arc-length parameterized function along the boundary B and it takes the value 1 for gap-filling fragments and takes the value 0 for detected fragments. The term $\int_B \sigma(t)dt$ measures the total gap-length along the boundary B and reflects the favor of good proximity, which has been incorporated in many prior edge-grouping methods. The term $\iint_R |\nabla(I(u,v))|dudv$ measures the total intensity variation inside the enclosed region and reflects the favor of good intensity homogeneity in region R. The normalization over $\iint_R dudv$, the area of the enclosed region R, can avoid a bias to produce overly small structures resulting from image noise. The parameter $\lambda > 0$ is used to balance the boundary proximity and region homogeneity.

In the next section, we develop an algorithm to find the optimal boundary that minimizes the cost function $\phi(B)$.

3. Algorithm

The problem formulated in Section 2 can be reduced to the same problem with the additional constraint that the boundary must traverse a given detected fragment. More specifically, this new problem can be written as

Problem 1: From the detected fragments $\Gamma_i, i = 1, 2, \ldots, n$ and the constructed gap-filling fragments, find a closed boundary B_k that minimizes the cost $\phi(B)$, subject to (a) B_k is convex, and (b) B_k traverses the detected fragment Γ_k.

This way, the optimal boundary for the problem formulated in Section 2 is the one in $\{B_1, B_2, \ldots, B_n\}$ with the minimum cost $\phi(B)$. Furthermore, a convex boundary traversing fragment Γ_k must be located completely in one of the two semiplanes partitioned by the straight line along Γ_k. Therefore, we can solve Problem 1 in two semiplanes as shown in Fig. 2(a) and among the two resultant boundaries, pick the one with smaller cost $\phi(B)$. To solve problem 1 in one semiplane, we can use a *fragment-pruning* algorithm, the basic idea of which is to prune all the detected and gap-filling fragments that are impossible to be located in B_k. After the fragment pruning, Problem 1 can be solved for the remaining fragments without considering the convexity constraint.

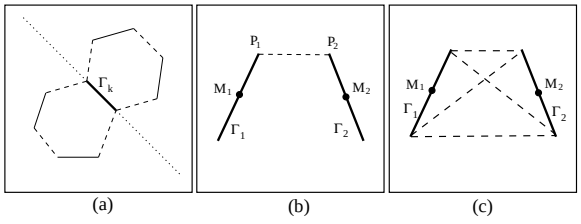

Figure 2: An illustration of the arc representation and construction. (a) Convex boundary traversing Γ_k must be located in one of the two semiplanes partitioned by Γ_k. (b) For the gap-filling fragment P_1P_2, an associated arc is constructed as the polyline $M_1P_1P_2M_2$, where M_1 and M_2 are the midpoints of Γ_1 and Γ_2. (c) Between two detected fragments Γ_1 and Γ_2, there may exist four gap-filling fragments (dashed lines), and therefore, we have four associated arcs for them.

To better describe this fragment-pruning algorithm, we first introduce the concept of *arc* to model the local connections between detected and gap-filling fragments. As shown in Fig. 2(b), each gap-filling fragment connects two detected-fragment endpoints $P_1 \in \Gamma_1$ and $P_2 \in \Gamma_2$. Let the midpoint of a detected fragment Γ_i be M_i. We define the triple-segment polyline $M_1P_1P_2M_2$ to be an *arc* associated to the gap-filling fragment P_1P_2, i.e., there is a one-to-one mapping between the gap-filling fragments and the arcs. Clearly, a closed boundary can also be described by a sequential connection of arcs associated to the gap-filling fragments along this boundary.

We carry out the following steps for fragment pruning. First, we prune all the fragments that are not located in the considered semiplane partitioned by Γ_k. As illustrated in Fig. 3(a), if we are considering solving Problem 1 in the top-right semi-plane, we can prune the detected fragments Γ_1 and Γ_3, and gap-filling fragments P_1P_2 and P_5P_6. Second, in the considered semiplane, we further prune all the gap-filling fragments for which the associated arcs are not convex, because a convex boundary can not contain any nonconvex segment. As shown in Fig. 3(b), gap-filling fragment P_3P_4 will be pruned. Finally, we prune all the gap-filling fragments for which the associated arcs are not convex *with respect to (w.r.t) M_k*, the midpoint of the given detected fragment Γ_k. An arc is convex *w.r.t* a point if they can be located along the same convex closed boundary. As shown in Fig. 3(c), the arc convexity w.r.t to a point can be decided by checking the convexity of the polygon (shaded in Fig. 3(c)) formed by connecting the arc and the point. Clearly, in Fig. 3(c), gap-filling fragment P_3P_4 is not convex w.r.t M_k and will be pruned.

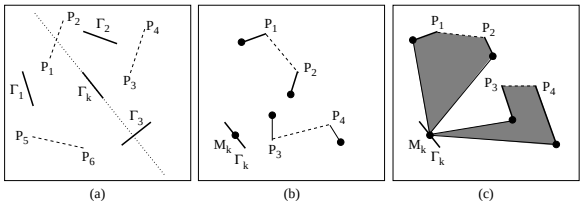

Figure 3: An illustration of the fragment pruning in solving Problem 1. Γ_k is the given detected fragment which the resultant boundary is required to traverse. (a) Prune all fragments not in the considered semiplane. (b) Prune all gap-filling fragments for which the associated arcs are not convex. (c) Prune all gap-filling fragments for which the associated arcs are not convex w.r.t M_k, the midpoint of Γ_k.

Now let's assume we have an algorithm to find the closed boundary B with the minimum cost $\phi(B)$ **without** the two constraints in Problem 1. By applying this algorithm to the remaining fragments after pruning, we can find a boundary that automatically satisfies the constraints in Problem 1. This can be proved by considering the cases shown in Fig. 4. First, the achieved boundary must be located in the considered semiplane partitioned by Γ_k, because all the fragments not in this semiplane have been pruned. Second, the resultant boundary must traverse Γ_k. Otherwise, as shown in Fig. 4(a), for any closed boundary that does not traverse Γ_k, we can always find from it a gap-filling fragment, say P_1P_2, for which the associated arc is not convex w.r.t M_k. This will not happen since all such gap-filling fragments have been pruned. Third, the resultant boundary must be convex. Otherwise, along this boundary we can always find an arc which is either nonconvex itself (see the arc associated to P_1P_2 in Fig. 4(b)) or not convex w.r.t M_k (see the arc associated to P_3P_4 in Fig. 4(c)). This will not happen since we have pruned all such gap-filling fragments. Also note that the resultant boundary

must be simple, i.e., it does not intersect itself. This is obvious since, for a self-crossing closed boundary, its inner loop is also a closed boundary that does not traverse Γ_k and therefore, the resultant boundary must contain an arc not convex w.r.t M_k, but all such arcs have been pruned.

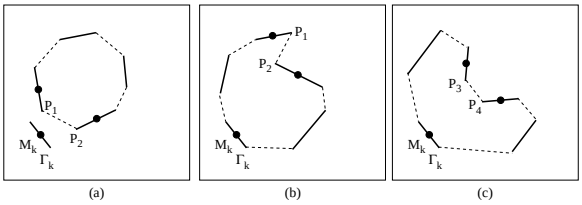

Figure 4: Solving Problem 1 on the remaining fragments without the two constraints can not produce (a) a boundary not traversing Γ_k and (b-c) nonconvex boundaries.

Another problem is the global optimality of the boundary detected from the remaining fragments in terms of Problem 1. In other words, we wonder whether the proposed fragment-pruning algorithm may reduce the search space by pruning some fragments along the desired global optimal boundary. This in fact will not happen, since all the pruned fragments are the ones that are impossible to be along a valid convex boundary that traverses Γ_k. Therefore, our only remaining problem is how to find an optimal boundary minimizing the cost $\phi(B)$ without considering the two constraints in Problem 1. Let's show that this problem can be solved by a graph-theoretic algorithm called *Ratio Contour*, which was developed by Wang et al. in [22].

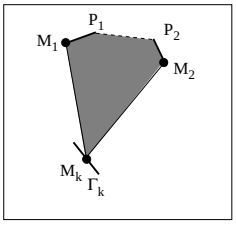

Figure 5: An illustration of the region $R(e)$ (shaded region) used for defining weight functions of a dashed edge.

We first construct an undirected graph $G = (V, E)$ with V as the vertex set and E as the edge set. Each fragment endpoint is represented by a vertex and each fragment is represented by an edge that connects two corresponding vertices. There are two kinds of edges, *solid* and *dashed* ones. The solid edges represent detected fragments and dashed edges represent gap-filling fragments. This way, a closed boundary is represented by an *alternate* cycle in this graph. An alternate cycle indicates a cycle that traverses solid and dashed edges alternately. Now let's define two weight functions, w_1 and w_2 for each edge. For a solid edge e, we simply set $w_1(e) = 0$ and $w_2(e) = 0$. For a dashed edge e, we get back and check its corresponding gap-filling fragment and associated arc. As shown in Fig. 5, let P_1P_2 be the gap-filling fragment corresponding to e and $M_1P_1P_2M_2$ be the arc associated to P_1P_2. We define

$$w_1(e) = |P_1P_2| + \lambda \cdot \iint_{R(e)} |\nabla I(u, v)| du dv \qquad (2)$$

and

$$w_2(e) = \iint_{R(e)} du dv, \qquad (3)$$

where $|P_1P_2|$ is the length of gap-filling fragment P_1P_2 and $R(e)$ is the convex region constructed by connecting the arc and the point M_k, as shown by the shaded region in Fig. 5. Note that a valid closed boundary B is uniquely represented by an alternate cycle C in this graph. It is easy to see that the cost $\phi(B) = CR(C)$, where the cycle ratio $CR(C)$ is defined as

$$CR(C) = \frac{\sum_{e \in C} w_1(e)}{\sum_{e \in C} w_2(e)}.$$

This way, the problem of finding an optimal boundary B from a set of fragments is reduced to a problem of finding an optimal alternate cycle $C \in G$ that minimizes this cycle ratio. This minimum-ratio-cycle problem can be solved in polynomial time using the ratio-contour algorithm [22].

The following summarizes the proposed convex-grouping algorithm.

```
// Initialization
Construct detected fragments {Γ_i}_{i=1}^n
Construct gap-filling fragments
//Main loop
Loop over Γ_k, k = 1, 2, ... n
     On semiplane 1
          Prune fragments
          Construct weighted graph G
          Find B_k1 with ratio contour
     On semiplane 2
          Prune fragments
          Construct weighted graph G
          Find B_k2 with ratio contour
     Take B_k = B_k1 if φ(B_k1) < φ(B_k2)
     Take B_k = B_k2 otherwise
Output B = {B_m | φ(B_m) ≤ φ(B_i), ∀i}
End
```

4. Experiments

In this section, we test the proposed convex-grouping method on some real images by detecting the most salient convex structures in terms of the cost function (1). Particularly, we adopt the Canny detector in the Matlab software for edge detection and Kovesi's software (`http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/`) for line fitting. The edge-detection, line-fitting, and ratio-contour algorithms all have some free

parameters. The selection of suitable values for these free parameters is usually dependent on the image size. Therefore, we normalize all of our test images to be of a size as close as possible to 160×120 or 120×160, for landscape or portrait images respectively, but maintaining the aspect ratio. This way, we fix all the free parameters. In these experiments, we leave the parameters for the Canny detector at their default values, and for Kovesi's line fitting, we set the minimum length of the processed edges to be 10 pixels and the allowed deviation between an edge and its fitted line to be 2 pixels. For the ratio-contour algorithm, we use the package developed in [22] and set the parameter $\lambda = 0.01$ for all experiments.

It is well known that the Canny edge detector usually performs poorly at corners and junctions. Therefore, the location of some detected-fragment endpoints may be affected by strong noise. As a result, constructing gap-filling fragments by simply connecting two detected-fragment endpoints may introduce some problems. As shown in Fig. 6(a), the gap-filling fragment connecting P_1 and P_2 will be pruned since it is associated to a nonconvex arc $M_1P_1P_2M_2$. However, this may be caused by noise on the endpoint P_2. To address this problem, we adopt the same strategy suggested in Jacobs' method [11], which allows the endpoint P_2 to slide along Γ_2 freely in constructing the gap-filling fragment P_1P_2. As shown in Fig. 6(b), the extension of the fragment Γ_1 intersects Γ_2. This intersection point is set to be the new endpoint P_2 of Γ_2. The newly constructed gap-filling fragment P_1P_2, shown by a dashed line in Fig. 6(b), is associated to a convex arc and will not be pruned.

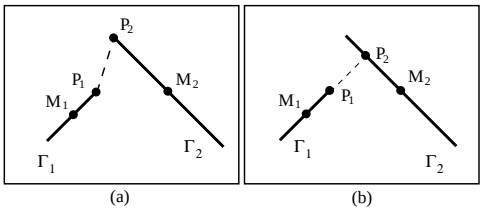

Figure 6: An illustration of constructing a gap-filling fragment. (a) Constructing a gap-filling fragment by connecting the two fixed endpoints directly. (b) Constructing a gap-filling fragment by allowing one endpoint to slide along its fragment.

If we construct gap-filling fragments between all possible pairs of fragment endpoints, we will have $O(n^2)$ number of gap-filling fragments from n detected fragments. In practice, we reduce the number of constructed gap-filling fragments by not considering the ones between two fragments endpoints that are far from each other. Specifically, we set a distance threshold d_T, if the distance between two endpoints is larger than d_T, we do not construct a gap-filling fragment between them. In our experiments, we set $d_T = \frac{\min\{W,H\}}{5}$, where W and H are the width and height of the image.

Figure 7 shows the convex-grouping results on 16

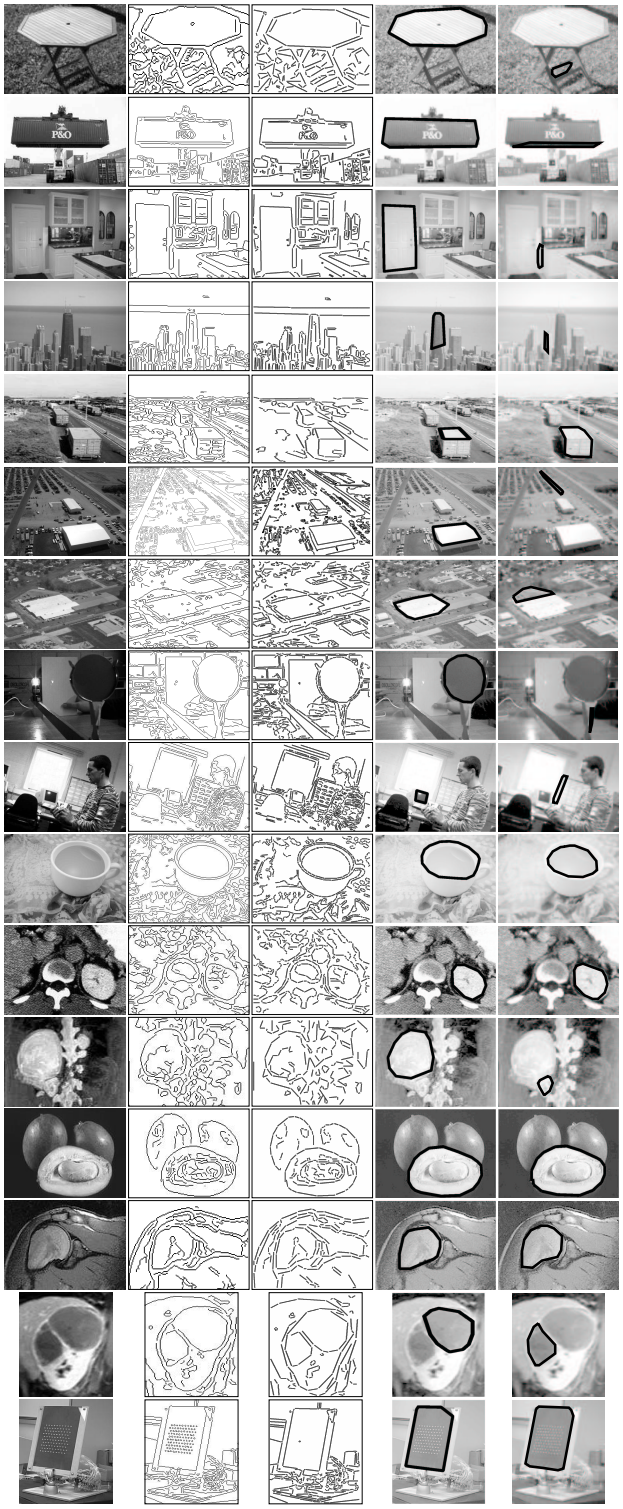

Figure 7: Convex-grouping results on 16 real images. First column: original images; Second column: Canny edge detection results; Third column: line-fitting results; Fourth column: the detected most salient boundaries using the proposed convex-grouping method; Fifth column: the detected most salient boundaries using Jacobs' method.

real images. The most salient convex boundaries extracted using the proposed method and Jacobs' method are shown in the fourth and fifth columns respectively. The code of Jacobs' method was downloaded from `http://www.cs.umd.edu/~djacobs/` and we set the parameter $k = 0.7$ (see [11]). We can see that by considering the region information, including both the intensity homogeneity and region area in the cost (1), the proposed method can extract structures with homogenous regions and large areas, as witnessed in the first, second, third, fourth, sixth, seventh, eighth, 12th, and 15th images in Fig. 7. Furthermore, although the convex-grouping problem is formulated as extracting a convex polygon in this paper, some smooth convex boundaries can still be quite accurately extracted in the form of a polygon approximation, as shown in the eighth, 10th, 11th, 12th, 13th, 14th, and 15th images in Fig. 7. One may notice that Jacobs' convex -grouping method sometimes produces nonconvex boundaries, such as the ones in the fifth and 14th images. This comes from a special processing in Jacobs' method which allows certain error in determining the colinearity between two detected fragments [11].

It is not difficult to extract multiple convex structures from an image by repeatedly applying the proposed convex-grouping method. We can run the proposed method once to extract the optimal boundary. After that, we can remove all the fragments along the detected boundary and then repeat the method to detect the second optimal boundary. Figure 8 shows the multiple structures detected on several real images.

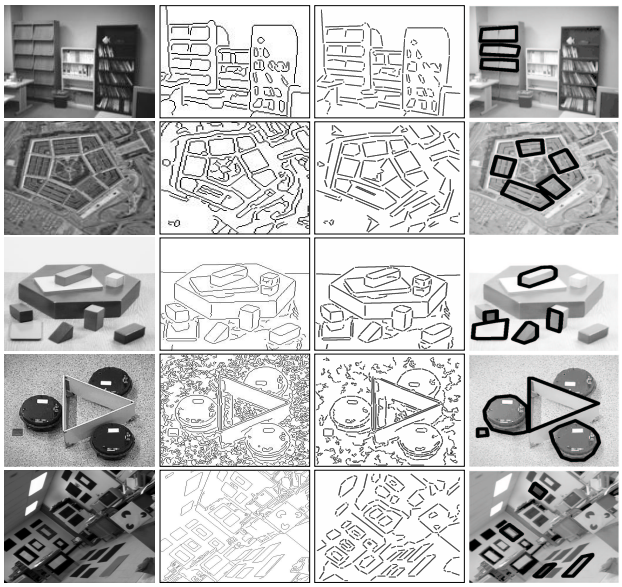

Figure 8: Sample results on detecting multiple convex structures by repeating the proposed convex-grouping method.

5. Extensions

In this section, we consider two important extensions of the proposed convex-grouping method. The first one enables the method to deal with open boundaries, i.e. boundaries of the structures that are cropped by the image perimeter and only partially located inside the image. The second extension is to incorporate a simple human-computer interaction to detect the salient convex structures around a user specified point.

5.1. Open-boundary Detection

The above convex-grouping method is developed based on the assumption that the desired structure is completely located inside the image perimeter and the structural boundary is a closed polygon, as shown in Fig. 9(a). However, in many real images, the most salient structure may only be partially present in the image and the structural boundary is cropped by the image perimeter. This results in an open boundary, as shown in Fig. 9(b) and (c). Open-boundary detection is a more difficult problem in grouping when incorporating the region information. The major reason is that we cannot distinguish which region is the one enclosed by an open boundary, i.e., region R in cost (1) is not well defined. One typical way to address this problem is to reformulate the cost function by seeking a balance between two resulting segments in terms of some region information, such as the area, the intensity homogeneity, or the total pixel affinity. Unfortunately, such a reformulation of the grouping cost usually results in a NP-hard problem [17, 18, 21].

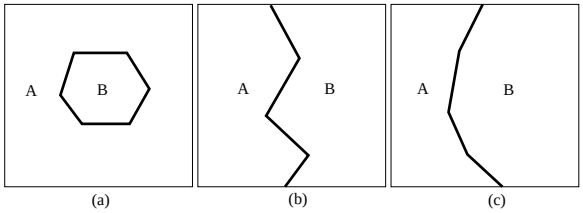

Figure 9: An illustration of determining the enclosed region R in the cost function (1) in various cases. (a) Enclosed region R is well defined given a closed boundary. In this case this region is B. (b) Enclosed region R is ill defined given an open boundary: Is the enclosed region A or B? (c) Enclosed region R is well defined given an open boundary in convex grouping. In this case, this region is B.

However, we have no such problems in the proposed convex-grouping method, because the structural convexity can be used to determine which one is the region enclosed by an open boundary. An example is shown in Fig. 9(c), where the region B is convex, so it is unambiguously the region R in the cost function (1). Therefore, we do not need to change the problem formulation to deal with the open-boundary detection. We only need to adapt the construction of some gap-filling fragments and the definition of edge weights for these

fragments so that they only count the boundary and region information inside the image perimeter.

First, we check all the fragment endpoints that are near the image perimeter. As shown in Fig. 10, even though the endpoints P_1 and P_2 are far from each other based on their spatial distance, we still construct a gap-filling fragment connecting them if they are both near the image perimeter. Such a gap-filling fragment may represent the cropped structural boundary and we call them *external* (gap-filling) fragments. Note that the external fragment between P_1 and P_2 is not the simple straight line segment P_1P_2. Instead, it is a convex polyline connecting P_1, Q_1, Q_2, and P_2 sequentially, where Q_1 and Q_2 are intersection points between the extended fragments and the image perimeter. Between Q_1 and Q_2, this external fragment is aligned with the image perimeter. Therefore, if Q_1 and Q_2 are located on the same side of the image perimeter, as shown in Fig. 10(a), they are connected by a straight line segment. If Q_1 and Q_2 are located on two different sides of the image perimeter, two or three straight line segments may be used to connect them, as shown in Fig. 10(b). All the external fragments are treated the same as other gap-filling fragments in the fragment pruning.

Second, like other gap-filling fragments, each external fragment is also represented by a unique dashed edge in the graph. However, its two weight functions are redefined to only consider the boundary and region information inside the image perimeter. Specifically, for the dashed edge e corresponding to the external fragment shown in Fig. 10(b), we adapt the definition of the first edge weight as

$$w_1(e) = |P_1Q_1| + |Q_2P_2| + \lambda \cdot \iint_{R(e)} |\nabla I(u,v)| du dv \quad (4)$$

because for this external fragment, only line segments P_1Q_1 and P_2Q_2 can be included in an open boundary. For both weight functions, $R(e)$ represents the convex region enclosed by the arc and the given point M_k. An example is shown in Fig. 10(c), where the shaded region is $R(e)$ for the external fragment $M_1P_1Q_1Q_2P_2M_2$ shown in Fig. 10(b).

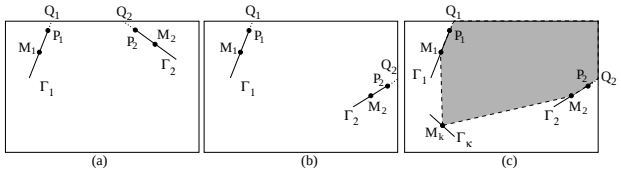

Figure 10: An illustration of constructing external gap-filling fragments. (a-b) Sample external fragments constructed between P_1 and P_2. (c) Shaded region is the $R(e)$ in the edge-weight definitions for the external fragment shown in (b).

By introducing the external fragments, we can apply the proposed convex-grouping method to detect a closed boundary as usual. If the resulting boundary contains an external fragment. We know that it traverses portions of the image perimeter and we in fact obtain an open boundary. This way, the detection of open and closed boundaries is unified in the proposed method. In Fig. 11, we show several sample images where the detected most salient structural boundaries are open. In these experiments, a fragment endpoint P_1 is considered to be near the image perimeter if $|P_1Q_1| < \frac{d_T}{2}$, where d_T is the same threshold defined in Section 4.

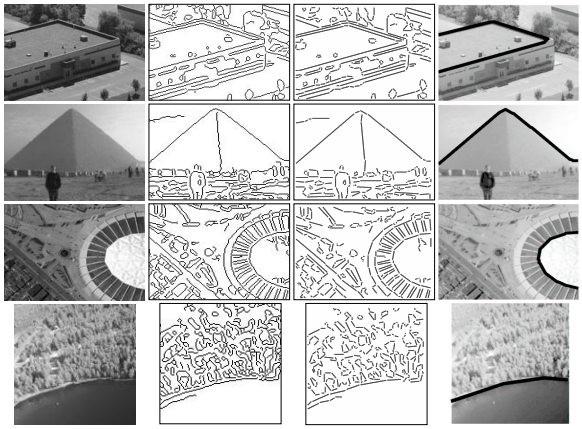

Figure 11: Sample results of detecting open boundaries using the proposed convex-grouping method. From left to right, each row shows the original image, the Canny detection result, the line-fitting result, and the detected open boundary, respectively.

5.2. Human-Computer Interaction

Adding some simple human-computer interactions may allow a user to more flexibly control the convex-grouping process. In this section, we consider the following simple human-computer interaction: The user specifies a point in the image and then the computer finds the most salient convex boundary around this specified point. In fact, this just adds a new constraint to the problem formulated in Section 2: the detected boundary should enclose a given point M. We address this problem by adding another step of fragment pruning: We prune all the gap-filling fragments for which the associated arcs are not convex w.r.t the given point M. This additional fragment-pruning step is executed before any other fragment pruning. After this, we run the standard arc-pruning and ratio-contour algorithms introduced in Section 3. Similar to the proof given in Section 3, we can easily show that the resultant boundary must be around M and is also the globally optimal one for the problem formulated in Section 2 with this additional interaction constraint. Figure 12 shows some sample convex-grouping results with some manually specified points. Note that, for the third image (from the left), the last row shows a result where the human-computer interaction leads to an open boundary.

6. Conclusions

This paper presented a new convex-grouping method for detecting salient convex structures from an image. In this

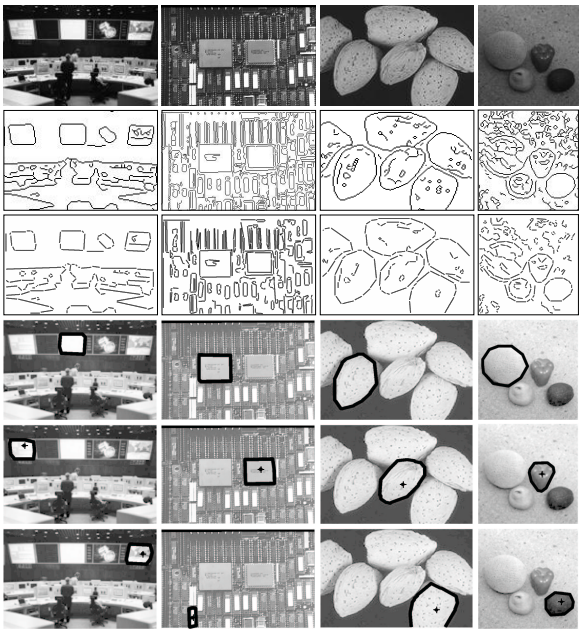

Figure 12: Sample convex-grouping results with the human-computer interaction introduced in Section 5.2. Each column, from top to bottom, shows the original image, the Canny detection result, the line-fitting result, the grouping results without human-computer interaction, and the results with two different specified points (marked by cross).

method, we consider both boundary and region information, including closure, convexity, proximity, and the intensity homogeneity of the enclosed region. This method can detect the globally optimal boundaries in terms of the defined cost function in polynomial time. We demonstrate the performance of this convex-grouping method on a set of real images and the results were compared to the convex-grouping method developed by Jacobs. This method can be extended to (a) detect open boundaries where the structure of interest is not fully located inside the image perimeter, and (b) incorporate a simple human-computer interaction for finding a convex boundary around a specified point.

Acknowledgements We would like to thank Dr. David Jacobs for important suggestions. This work was funded, in part, by NSF-EIA-0312861.

References

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

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

[3] S. Borra and S. Sarkar. A framework for performance characterization of intermediate-level grouping modules. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(11):1306–1312, 1997.

[4] T. Chan and L. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

[5] I. Cox, S. B. Rao, and Y. Zhong. Ratio regions: A technique for image segmentation. In *International Conference on Pattern Recognition*, pages 557–564, 1996.

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

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

[8] F. J. Estrada and A. D. Jepson. Controlling the search for convex groups. Technical Report CSRG-482, University of Toronto, Department of Computer Science, Jan. 2004.

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

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

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

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

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

[14] Z. Liu, D. Jacobs, and R. Basri. The role of convexity in perceptual completion: beyond good continuation. *Vision Research*, 39:4244–4257, 1999.

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

[16] H. Pao, D. Geiger, and N. Rubin. Measuring convexity for figure/ground separation. In *International Conference on Computer Vision*, pages 948–955, 1999.

[17] S. Rao. Faster algorithms for finding small edge cuts in planar graphs. In *24th Annual ACM Symposium on the Theory of Computing*, pages 225–237. ACM, 1992.

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

[19] E. Saund. Finding perceptually closed paths in sketches and drawings. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(4):475–491, 2003.

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

[21] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[22] S. Wang, T. Kubota, J. Siskind, and J. Wang. Salient closed boundary extraction with ratio contour. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2005.

[23] 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.