4-2005

Salient Closed Boundary Extraction with Ratio Contour

Song Wang
University of South Carolina - Columbia, songwang@cse.sc.edu

Toshiro Kubota

Jeffrey Mark Siskind

Jun Wang

Publication Info

IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 27, Issue 4, 2005, pages 546-561.

Salient Closed Boundary Extraction with Ratio Contour

Song Wang, *Member*, *IEEE*, Toshiro Kubota, *Member*, *IEEE*,
Jeffrey Mark Siskind, *Member*, *IEEE*, and Jun Wang

Abstract—We present ratio contour, a novel graph-based method for extracting salient closed boundaries from noisy images. This method operates on a set of boundary fragments that are produced by edge detection. Boundary extraction identifies a subset of these fragments and connects them sequentially to form a closed boundary with the largest saliency. We encode the Gestalt laws of proximity and continuity in a novel boundary-saliency measure based on the relative gap length and average curvature when connecting fragments to form a closed boundary. This new measure attempts to remove a possible bias toward short boundaries. We present a polynomial-time algorithm for finding the most-salient closed boundary. We also present supplementary preprocessing steps that facilitate the application of ratio contour to real images. We compare ratio contour to two closely related methods for extracting closed boundaries: Elder and Zucker's method based on the shortest-path algorithm and Williams and Thornber's method based on spectral analysis and a strongly-connected-components algorithm. This comparison involves both theoretic analysis and experimental evaluation on both synthesized data and real images.

Index Terms—Image segmentation, perceptual organization, boundary detection, edge detection, graph models.

1 Introduction

In this paper, we present and analyze a novel method for extracting perceptually salient closed boundaries in images. This problem can be divided into two parts: formulating appropriate criteria for salience that reflect human perceptual judgment and, subsequently, finding boundaries that meet those criteria. Cognitive scientists have articulated Gestalt laws, such as *closure*, *proximity*, and *continuity*, that attempt to characterize boundary salience. In this paper, we present one way of formalizing these laws in terms of a precise mathematical objective function to be optimized. We then present a novel graph-theoretic algorithm, called *ratio contour*, for finding a global optimum to this objective function in polynomial time. Our methods always yield closed boundaries and our saliency measure is largely insensitive to boundary length.

The earliest attempts at finding salient boundaries were based on edge detection (e.g. [45], [41], [55], [7], [43], [26], [24], [6], [32]). However, such methods usually produce boundary fragments that are not connected into closed boundaries, especially in the presence of noise and occlusion. Furthermore, most edge detectors do not incorporate a salience metric that reflects the Gestalt laws of proximity and continuity. Edge-linking methods, such as [23], attempt to address these problems by connecting boundary fragments into closed boundaries using local-search techniques. However, these methods do not guarantee an optimal solution to an independently specified salience measure.

A broader class of local optimization techniques have been applied to the general problem of boundary extraction. These include active-contour (e.g., [30], [3], [9], [63], [8], [67]) and shape-deformation (e.g., [57], [11], [28], [47], [36], [22]) methods. Both types of method usually force the boundary to be smooth and closed by iteratively updating an initial boundary to produce a series of boundaries that better meet the saliency criteria. However, the result of such an iterative method can depend on the initial boundary and there is no guarantee that a globally optimal boundary will be found. Other approaches for extracting boundaries include the Ising model [46], the Cartoon model [20], the Theater-Wing model [39], the Spectrogram model [35], and the Region-Competition model [68], where the saliency measure is explicitly formulated as a Bayesian variational problem. However, it is usually difficult to find optimal solutions for these variational problems. In fact, many of these problems are NP-hard. Practical approaches to solving these problems often use local-search techniques that require some form of initialization.

To avoid dependence on initialization, graph-theoretic approaches were introduced to guarantee production of boundaries that globally optimize a given saliency measure. These methods typically construct a graph where the vertices represent pixels or small regions and the weighted edges represent affinity between these pixels or regions. In this context, finding a boundary is reduced to the problem of partitioning the graph in a way that optimizes some cost function. Different graph-theoretic methods employ different algorithms to optimize different cost functions. These include Minimum Cut [66], Ratio Regions [13], Normalized Cut [54], Average Cut [49], the methods of Jermyn and Ishikawa [29], Ratio Cut [61], and many more recent methods (e.g., [40], [19], [51], [59], [52], [56], [5], [18]). Some of these methods attempt to solve NP-hard problems using approximate algorithms. Because many of those methods construct graphs where vertices correspond to pixels or small regions, it is difficult to incorporate many Gestalt laws,

- *S. Wang, T. Kubota, and J. Wang are with the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208. E-mail: {songwang, kubota, wang286}@cse.sc.edu.*
- *J.M. Siskind is with the School of Electrical and Computer Engineering, 465 Northwestern Ave., Room 330, Purdue University, West Lafayette, IN 47906. E-mail: qobi@purdue.edu.*

Manuscript received 16 Mar. 2004; revised 30 Sept. 2004; accepted 1 Oct. 2004; published online 10 Feb. 2005.
Recommended for acceptance by R. Basri.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0132-0304.

such as smoothness, into the formulation of the saliency measure.

Cost functions that measure the Gestalt properties of closure, proximity, and continuity can be more conveniently formulated in terms of preextracted boundary fragments than image pixels. These boundary fragments, or simply *fragments*,[1] can be obtained with an edge detector. In this context, boundaries are extracted by identifying and connecting a subset of these fragments into a salient boundary. The Gestalt laws correspond to enforcing specified properties of a boundary. *Closure* requires that the boundary be a cycle. *Proximity* requires the gap between two neighboring fragments to be small. *Continuity* requires the resulting boundary to be smooth. Methods have been proposed for connecting fragments in a way that attempts to satisfy these properties (e.g., [53], [16], [37], [2], [27], [48], [21], [64], [4], [65], [37], [25], [4], [27], [44]). These methods typically:

1. Define a *prominence*[2] measure for each fragment and/or gap between two neighboring fragments. This measure is usually based on the smoothness and length of these fragments and/or gaps to encode the local continuity and proximity properties.
2. Define a *saliency* measure for valid boundaries in terms of fragment and gap prominence defined above. For example, saliency of a boundary can be defined as the sum of the prominences of its component fragments and gaps.
3. Develop an optimization algorithm for finding the boundary that maximizes the selected saliency measure. Such algorithms may employ graph-theoretic techniques in an attempt to achieve global optimality [16], [37].

Prior methods differ in their choice of prominence and saliency measures and approaches to optimization.

A psychological study has shown that boundary closure plays a critical role in human perception, as shown by Kovacs and Julesz [33, p. 7496]:

> *We found an unexpected advantage of circular arrangements: Δ_c (maximum spacing for closed contours) was extended by a factor of 1.8 relative to Δ_o (maximum spacing for open ones).*

In other words, with similar maximum gap (spacing) between fragments, humans may perceive closed curves, but not open ones. The maximum allowed gap in perceiving a closed curve is 1.8 times the maximum allowed gap in perceiving an open curve. Since closure appears to be important in human vision, researchers deem it prudent to incorporate closure into computer vision.

As a global property, boundary closure is usually more difficult to enforce with prominence measures than local properties such as proximity and continuity. In fact, many of the above methods cannot guarantee production of closed boundaries. One way to enforce closure is to perform constrained optimization that only considers closed boundaries. In the context of graph-based optimization algorithms, this constraint corresponds to finding *cycles* in a graph. For example, Elder and Zucker [16] define boundary saliency as the product of fragment/gap prominence along the boundary and then use the shortest-path algorithm [14] to find the optimal cycle. Similarly, Williams and Thornber [65] define boundary saliency as the geometric mean of fragment/gap prominence along the boundary and use spectral-analysis techniques to enhance the fragment/gap prominence measure with closure information.[3] With this measure, Mahamud et al. [37] attempt to enforce closure using a strongly-connected-components algorithm [12]. Other related methods that attempt to find closed boundaries include [27], [4], [42].

This paper presents a new graph-based method, called *ratio contour*, for detecting closed salient boundaries, where the globally most-salient boundary is found in polynomial time. Like the methods described above, ratio contour finds boundaries by identifying and connecting a set of preextracted fragments. Further, ratio contour guarantees closure by constrained graph-based optimization. We encode the Gestalt laws of proximity and continuity in a novel boundary-saliency measure based on relative gap length and average curvature. Formulating saliency in terms of *relative* gap length and *average* curvature removes the bias toward short boundaries that is present in saliency measures based on *total* gap length and curvature.

Our paper contains two major parts. In the first part, we present the algorithmic details of ratio contour along with the supplementary processing steps needed to use this method to extract salient boundaries from images. More specifically, we present methods for 1) preprocessing the edge-detector output to produce a set of topologically unconnected fragments, 2) smoothing the detected fragments to remove noise, and 3) reliably estimating the curved gap-filling segment between two detected fragments. The latter two use a spline-based curve smoothing algorithm for noise removal and robust gap filling. In the second part, we analyze the differences between ratio contour and the two most-related prior methods: Elder and Zucker (EZ) [16] and Williams and Thornber (WT) [65], [37], and compare their performance on synthesized and real images in a unified framework.

The remainder of this paper is organized as follows: Section 2 formulates the saliency measure used by ratio contour and converts the problem of finding the most-salient boundary with this measure into the problem of finding an optimal cycle in a graph. Section 3 presents a polynomial-time algorithm for finding the desired optimal cycles. Section 4 discusses the preprocessing steps 1) through 3) described above to derive the fragment/gap prominence measure, in terms of graph edge weights, from noisy edge-detector output. Section 5 illustrates the boundaries detected by ratio contour on real images. Section 6 analyzes the similarities and differences between ratio contour and the methods of EZ and WT. Section 7 compares the performance of ratio contour with that of EZ and WT with experiments on real and synthetic images. Section 8 concludes with a summary of our method.

2 Problem Formulation

We refer to the process of identifying a subset of fragments produced by preprocessing and connecting the fragments in that subset to form a closed boundary as *boundary extraction*.

1. Prior literature often uses the term "edge" instead of "fragment." We do not use this terminology to avoid confusion with the notion of edge in its graph-theoretic sense.

2. Note that we have prominence measures for both fragments and gaps. Prior literature often uses the term "affinity" between a pair of fragments to refer to what we call gap prominence. We refrain from using the term affinity as it is not appropriate for individual fragments and we wish to use unified terminology for both fragments and gaps.

3. In [37], such enhanced fragment/gap prominence is referred to as "edge saliency" and "link saliency."

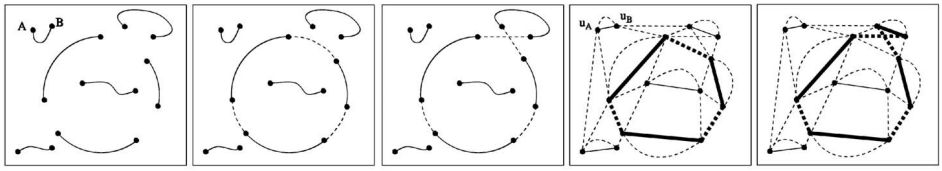


Fig. 1. Extracting a salient boundary from a set of fragments. (a) Real fragments. (b) A nondegenerate non-self-crossing closed boundary derived by connecting a subset of real and virtual fragments. (c) A nondegenerate self-crossing closed boundary. (d) The solid-dashed graph G constructed from (a) together with a simple alternate cycle (shown in thick lines) corresponding to the boundary shown in (b). (e) A simple alternate cycle (shown in thick lines) corresponding to the boundary shown in (c).

As shown in Fig. 1a, the input to boundary extraction consists of a set of noncrossing fragments, each of which is a continuous open curve segment with two endpoints. We further assume that a boundary produced by boundary extraction always consists of *indivisible* fragments, i.e., a boundary cannot contain only part of a fragment. The goal of boundary extraction is to find the closed boundary with maximum perceptual saliency, as shown in Fig. 1b. To form a closed boundary from disconnected fragments, we construct another set of fragments to fill the gaps between the initial fragments. To distinguish these two kinds of fragments, we refer to the initial fragments, the solid curves in Fig. 1b, as *real* fragments and the gap-filling fragments, the dashed curves in Fig. 1b, as *virtual* fragments.

Virtual fragments are derived from adjacent real fragments by a process of gap completion. Any gap-completion method can be used (e.g., [38], [58], [64], [51]). For the experiments in this paper, we use the particular gap-completion method that we present in Section 4. We associate a *prominence* or *fragment cost* with each fragment to describe how likely or unlikely that fragment is to be included in the most-salient boundary. Such fragment prominence or cost usually incorporates only local information derived from a given fragment. Many formulations of fragment prominence or cost have been proposed (e.g., [16], [64], [60]). Such formulations often attempt to codify the Gestalt laws of proximity and continuity: 1) real fragments are more prominent than virtual fragments, 2) short virtual fragments are more prominent than long virtual fragments, and 3) smooth fragments are more prominent than nonsmooth fragments.

A closed boundary can then be constructed by connecting a subset of real and virtual fragments, sequentially and alternately, as shown in Fig. 1b. We are only interested in *nondegenerate* boundaries where no fragment is traversed more than once when forming a closed boundary. For the remainder of the paper, we use the term "boundary," without the qualifier "degenerate," to refer to nondegenerate boundaries. Note that the nondegeneracy of a boundary does not require it to be non-self-crossing[4] in our problem formulation because a virtual fragment may cross another real or virtual fragment along an extracted boundary. An example of this is shown in Fig. 1c, where the boundary is nondegenerate but self crossing. The ratio-contour method presented in this paper extracts only nondegenerate boundaries, which, however, may be self-crossing. Associated with each closed boundary, we can define a *boundary saliency* $S(\cdot)$ or a *boundary cost* $\Gamma(\cdot)$ as a function of the prominence or cost of the fragments that form that boundary. Finally, we develop an optimization algorithm to search for a closed boundary that maximizes this boundary saliency or, alternatively, minimizes the boundary cost. In essence, various boundary-extraction methods differ mainly in their definition of boundary saliency/cost and/or their optimization algorithm.

We encode the Gestalt laws of proximity and continuity into the following boundary-cost definition:

$$\Gamma_{rc}(B) \triangleq \frac{W(B)}{L(B)} = \frac{\int_B [\sigma(t) + \lambda \cdot \kappa^2(t)]dt}{\int_B dt}, \tag{1}$$

where $\mathbf{v}(t)$, $0 \le t \le L(B)$, is the arc-length parameterized form of the boundary B, $\sigma(t) = 1$ if the point $\mathbf{v}(t)$ is in a virtual fragment, $\sigma(t) = 0$ if it is in a real fragment, and $\kappa(t)$ is the curvature of the boundary at $\mathbf{v}(t)$. The most-salient boundary B is then the one with the minimum cost $\Gamma_{rc}(B)$. The cost $\Gamma_{rc}(B)$ normalizes $W(B)$, a weighted sum of the total gap-length and curvature along the boundary B, by $L(B)$, the length of the boundary B, in an attempt to remove a possible bias toward short boundaries in $W(B)$. The parameter $\lambda > 0$ controls the relative contribution of proximity and continuity to this cost.

While Γ_{rc} avoids a bias toward short boundaries, it does not eliminate all biases dependent on boundary length. The choice of most-salient boundary using Γ_{rc} can depend on image scale, given a fixed λ. An example is shown in Fig. 2, which contains two circular boundaries, A and B. Along A, the virtual fragments count for half of the boundary length, while, along B, the virtual fragments count for one-third of the boundary length. Consider the case where the radius of these two circular boundaries are $r_A = 4$ and $r_B = 2$ and $\lambda = 2$. It is easy to see that

$$\Gamma_{rc}(A) = \frac{1}{2} + \frac{\lambda}{r_A^2} = \frac{1}{2} + \frac{1}{8} < \Gamma_{rc}(B) = \frac{1}{3} + \frac{\lambda}{r_B^2} = \frac{1}{3} + \frac{1}{2},$$

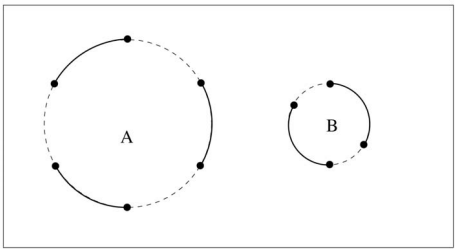


Fig. 2. The most-salient boundary that optimizes Γ_{rc} can depend on image scale and the parameter λ.

4. Prior literature often refers to non-self-crossing boundaries as "simple" boundaries. We do not use this terminology to avoid confusion with the notion of simple cycles in a graph.

i.e., A is more salient than B. If we enlarge the image by a factor of 2, we have $r_A = 8$ and $r_B = 4$. With the same $\lambda = 2$, we have

$$\Gamma_{rc}(A) = \frac{1}{2} + \frac{\lambda}{r_A^2} = \frac{1}{2} + \frac{1}{32} > \Gamma_{rc}(B) = \frac{1}{3} + \frac{\lambda}{r_B^2} = \frac{1}{3} + \frac{1}{8},$$

i.e., B is more salient than A.

The parameter $\lambda > 0$ specifies the relative contribution of proximity and continuity to saliency. With smaller λ, proximity dominates. With larger λ, continuity or smoothness dominates. Therefore, different values for λ may lead to different most-salient boundaries. To see this, consider again the two circular boundaries shown in Fig. 2 with the radii $r_A = 4$ and $r_B = 2$. We have previously shown that with $\lambda = 2$, A is more salient than B. However, with $\lambda = 0.5$, we have

$$\Gamma_{rc}(A) = \frac{1}{2} + \frac{\lambda}{r_A^2} = \frac{1}{2} + \frac{1}{32} > \Gamma_{rc}(B) = \frac{1}{3} + \frac{\lambda}{r_B^2} = \frac{1}{3} + \frac{1}{8},$$

i.e., B is more salient than A. Finding an optimal λ for an input image is one of our ongoing research topics. In this paper, we choose a fixed λ for all experiments.

We now formulate the problem of boundary extraction into an optimization problem in an *undirected* graph $G = (V, E)$. We construct a unique vertex for each fragment endpoint. Two kinds of edges, *solid* and *dashed*, are then constructed between vertices to represent real and virtual fragments, respectively. An example of such a graph is shown in Fig. 1d. This graph is constructed from the real fragments in Fig. 1a and consists of seven solid edges and 25 dashed edges. To illustrate this construction clearly, the graph in Fig. 1d is embedded so that each vertex u_A has the same spatial coordinate as its corresponding fragment endpoint A in Fig. 1a. G must have an even number of vertices since each real fragment has two endpoints. Furthermore, no two solid edges can be incident on the same vertex. We call such a graph an (undirected) *solid-dashed* (SD) graph. In this graph, an *alternate* cycle is defined as a cycle that alternately traverses solid and dashed edges and a *simple* cycle as a cycle that does not traverse a vertex more than once. For the remainder of the paper, we use the term "cycle," without the qualifier "nonsimple," to refer to simple cycles. Examples of an SD graph and alternate cycles are illustrated in Figs. 1d and 1e. It is easy to see that nondegenerate closed boundaries correspond exactly to simple alternate cycles in G. Thus, we can constrain our optimization of saliency to consider only nondegenerate closed boundaries by using graph algorithms that find simple alternate cycles.

To formulate the boundary cost (1) in terms of G, we associate a weight $w(e)$ and a length $l(e)$ with each edge e in G. For convenience, we define $B(e)$ as the original (real or virtual) fragment corresponding to an edge e. Based on this, we define

$$w(e) \triangleq W(B(e)) = \int_{B(e)} [\sigma(t) + \lambda \cdot \kappa^2(t)]dt, \tag{2}$$

which is the unnormalized cost of $B(e)$, and

$$l(e) \triangleq L(B(e)) = \int_{B(e)} dt, \tag{3}$$

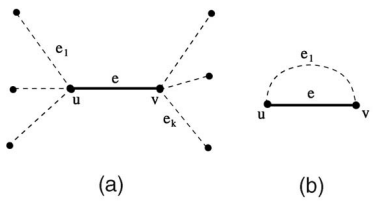


Fig. 3. Reassigning the weight and length of a solid edge to its adjacent dashed edges. (a) The cases where solid and dashed edges share only one vertex. (b) A case where solid and dashed edges share two vertices.

which is the curve length of $B(e)$. The most-salient closed boundary B with minimum cost $\Gamma_{rc}(B)$ corresponds to an alternate cycle C that minimizes the *cycle ratio*

$$\Gamma_{rc}(C) = \frac{\sum_{e \in C} w(e)}{\sum_{e \in C} l(e)}. \tag{4}$$

In the next section, we present a polynomial-time algorithm for finding such an optimal alternate cycle.

3 The Ratio-Contour Algorithm

For simplicity, we denote an alternate cycle with minimum cycle ratio as a *Minimum Ratio Alternate* (MRA) cycle. In this section, we introduce a polynomial-time algorithm for finding an MRA cycle in an SD graph G. This algorithm consists of three reductions: 1) Reduce the problem of finding an MRA cycle in a general SD graph G to the problem of finding an MRA cycle in a special SD graph with the same structure as G, except that all the solid edges have zero weight and length. This reduction is achieved by merging the weight and length of solid edges into the weight and length of their adjacent dashed edges, without changing the structure of the graph. 2) Reduce the problem of finding an MRA cycle in an SD graph G to the problem of finding a *Negative total Weight Alternate* (NWA) cycle in the same graph G. 3) Reduce the problem of finding an NWA cycle in an SD graph G with zero solid-edge weights and lengths to the problem of finding a *Minimum-Weight Perfect Matching* (MWPM) in the same graph G. One can find an MWPM in polynomial-time [15]. This allows us to find an MRA cycle in polynomial time as well.

3.1 Reduction 1: Setting the Weight and Length of Solid Edges to Zero

From (2) and (3), the weight and length of any edge in the constructed SD graph G are usually nonzero. In this section, we show how to transform the edge weights and lengths in G so that all solid edges have zero weight and length without changing the MRA cycle. As illustrated in Fig. 3, no two solid edges can be adjacent. Therefore, each solid edge $e = (u, v)$ can only be adjacent to a set of dashed edges, say $\{e_1, e_2, \ldots, e_K\}$, in G. We can decompose the weight $w(e)$ and length $l(e)$ of the solid edge e arbitrarily into sums of two nonnegative terms: $w(e) = w_u(e) + w_v(e)$ and $l(e) = l_u(e) + l_v(e)$. We can then reassign the weight and length of the solid edge to its adjacent dashed edges by the following transformation:

$$w(e_k) \leftarrow w(e_k) + \begin{cases} w_u(e) & \text{if } e_k \text{ only shares vertex } u \text{ with } e \\ w_v(e) & \text{if } e_k \text{ only shares vertex } v \text{ with } e \\ w(e) & \text{if } e_k \text{ shares both vertices } u \text{ and } v \text{ with } e \end{cases}$$

$$l(e_k) \leftarrow l(e_k) + \begin{cases} l_u(e) & \text{if } e_k \text{ only shares vertex } u \text{ with } e \\ l_v(e) & \text{if } e_k \text{ only shares vertex } v \text{ with } e \\ l(e) & \text{if } e_k \text{ shares both vertices } u \text{ and } v \text{ with } e \end{cases}$$

for $k = 1, 2, \ldots, K$ sequentially. We can then reset the weight and length of the solid edge e to zero. For example, we can set $w_u(e) = w_v(e) = \frac{1}{2}w(e)$ and $l_u(e) = l_v(e) = \frac{1}{2}l(e)$ to divide the weight and length of the solid edge e into two equal components and then reassign these components to the adjacent dashed edges. This reassignment process is performed iteratively over all solid edges. Since solid and dashed edges are traversed alternately in an alternate cycle, this process will not change the total weight and length of any alternate cycle in G. Therefore, this process does not change the MRA cycle in G.

3.2 Reduction 2: Detecting Negative-Weight Alternate Cycles

In this section, we reduce the problem of finding an MRA cycle in an SD graph G to the problem of finding an alternate cycle with negative total edge weight in the same graph G by searching for an appropriate transformation of the edge weights in G that preserves the MRA cycle but where the MRA cycle has a cycle ratio of zero. The cost of finding the appropriate parameter b^* for this transformation is incorporated into the complexity analysis of the algorithm. This reduction is possible because an MRA cycle in G is invariant to the following linear transformation of the edge weights:

$$w'(e) = w(e) - b \cdot l(e), \forall e \in E. \tag{5}$$

This holds because, for any two alternate cycles C_1 and C_2 in G, $\Gamma_{rc}(C_1) \leq \Gamma_{rc}(C_2)$ implies

$$\begin{aligned} \Gamma'_{rc}(C_1) &= \frac{\sum_{e\in C_1} w'(e)}{\sum_{e\in C_1} l(e)} = \frac{\sum_{e\in C_1}[w(e) - b \cdot l(e)]}{\sum_{e\in C_1} l(e)} \\ &= \Gamma_{rc}(C_1) - b \leq \Gamma_{rc}(C_2) - b = \frac{\sum_{e\in C_2} w'(e)}{\sum_{e\in C_2} l(e)} \\ &= \Gamma'_{rc}(C_2), \end{aligned}$$

where $\Gamma'_{rc}(\cdot)$ denotes the cycle ratio after the edge-weight transformation.

Since edge lengths are nonnegative, there exists an optimal $b = b^*$ so that, after the above edge-weight transformation, the resulting MRA cycles have a cycle ratio of zero. In this case, MRA cycles are the same as the cycles with zero total edge weight. Now, suppose that we have an NWA cycle-detection algorithm that can determine whether there is an NWA cycle in G and, if there is, can extract such a cycle, both in polynomial time. Then, we can use this NWA cycle-detection algorithm repeatedly to determine b^* using the following sequential-search algorithm, as shown in Fig. 4.

This sequential-search algorithm is adapted from Ahuja et al. [1, pp. 496-497], where it is used for finding a minimum ratio cycle in a directed graph. The correctness of this algorithm comes from the fact that an alternate cycle C always has zero total weight when all the edge weights are

Sequential-Search Algorithm:

1. Initialize $b = \max_{e\in E} \frac{w(e)}{l(e)} + 1$. We know that $b^* < b$.
2. Transform the edge weights using (5) and then use the NWA cycle-detection algorithm to detect an NWA cycle C. For the initial b, there must exist such an NWA cycle because the current $b > b^*$. If no NWA cycle is detected in a later iteration, return the alternate cycle C detected in the previous iteration as an MRA cycle in G.
3. Calculate the cycle ratio $\Gamma_{rc}(C)$ using the *original* edge weights without applying the edge-weight transformation (5). Update b to $\Gamma_{rc}(C)$ and go to Step 2.

Fig. 4. Sequential-search algorithm.

transformed by $w'(e) = w(e) - \Gamma_{rc}(C) \cdot l(e)$, after Step 3. To avoid confusion, we specify that the cycle ratio $\Gamma_{rc}(C)$ is always calculated using the original edge weights in G without applying the edge-weight transformation (5). Furthermore, we know that, with the current edge-weight transformation $w'(e) = w(e) - \Gamma_{rc}(C) \cdot l(e)$, there is no alternate cycle with negative total weight after termination. This implies that after termination $b^* = \Gamma_{rc}(C)$ and the current C is a desired MRA cycle.

It can be shown that this search process terminates in a (pseudo)polynomial number of iterations if the edge weights and lengths are all integers. Let $w_{max} = \max_{e\in E} w(e)$ and $l_{max} = \max_{e\in E} l(e)$. The cycle ratio of any alternate cycle C has the form

$$\Gamma_{rc}(C) = \frac{\sum_{e\in C} w(e)}{\sum_{e\in C} l(e)}.$$

When all edge weights and lengths are integral, the numerator, $\sum_{e\in C} w(e)$, can take on at most $w_{max}|E|$ different values. Similarly, the denominator, $\sum_{e\in C} l(e)$, can take on at most $l_{max}|E|$ different values. Therefore, the cycle ratio $\Gamma_{rc}(C)$ of any alternate cycle C in G can take on at most $w_{max}l_{max}|E|^2$ different values. Notice that, in the above sequential-search algorithm, the estimated $b = \Gamma_{rc}(C)$ will strictly decrease after each iteration. Therefore, this algorithm terminates in at most $w_{max}l_{max}|E|^2$ iterations.

Now, let us consider the combination of the first reduction described in Section 3.1 and this second reduction. After the first reduction, the SD graph G has a special property: All solid edges have zero weight and length. The edge-weight transformation (5) preserves this property. Therefore, we need only develop an NWA cycle-detection algorithm for SD graphs where all the solid edges have zero weight and length.

3.3 Reduction 3: Finding Minimum Weight Perfect Matchings

The problem of detecting an NWA cycle in an SD graph where all the solid edges have zero weight and length can be reduced to the problem of finding an MWPM in the same graph. A perfect matching in G denotes a subgraph of G that contains all the vertices in G, but where each vertex only has one incident edge. Fig. 5a shows an example of perfect matching where the seven thick edges, together with their vertices, form a perfect matching. The MWPM is the perfect matching with minimum total edge weight. In an SD graph, all the solid edges form a trivial perfect matching, which, in our case, has zero total weight given the first reduction from Section 3.1. Therefore, the MWPM in our SD graph will have nonpositive total weight.

We can derive a set of cycles from an MWPM P with the following two-step algorithm:

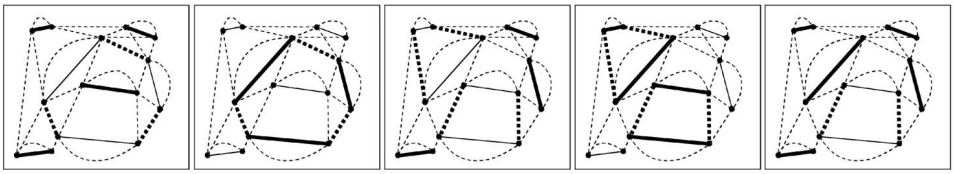


Fig. 5. Reduction 3 in the ratio-contour algorithm. (a) An MWPM (thick lines) in the G from Fig. 1. (b) One cycle (thick lines) derived from the MWPM in (a). (c) Another MWPM P in G. (d) Two cycles P'' derived from the MWPM P in (c). (e) The new perfect matching Q constructed from P in (c) when the upper-left cycle C^+ in P'' in (d) has positive total edge weight. This will contradict the assumption in (c) that P is an MWPM.

1. Remove from P all the solid edges and their incident vertices. We denote the resulting graph as P'.
2. Add the edge e to P' if (a) e is a solid edge in G and (b) e is not in the MWPM P. We denote the resulting graph as P''.

The graph P'' produced by the above algorithm consists of a set of disjoint alternate cycles because each vertex in P'' has two incident edges: one solid and one dashed. Two examples of this reduction are shown in Fig. 5. Since all the solid edges have zero weight and length and we only remove and add solid edges in the above algorithm, the total weight of the MWPM P is the same as the total weight of the derived alternate-cycle set P''. When P'' contains a single alternate cycle, as shown in Figs. 5a and 5b, the NWA cycle-detection problem is reduced to the problem of finding an MWPM and checking whether it has negative total weight.

However, P'' may contain multiple alternate cycles, as shown in Figs. 5c and 5d. It is easy to see that at least one of these alternate cycles must have nonpositive total weight, for otherwise, the sum of the total weights of all of the alternate cycles would be positive. This is sufficient for our reduction. We can show an even stronger property; however, each alternate cycle in P'' must have nonpositive total weight. This can be shown by contradiction. Assume one cycle, C^+, in the cycle set P'', has a positive total weight. Then, we can construct a new perfect matching Q from the MWPM P by: 1) removing from P all the dashed edges in C^+ and 2) adding into P all solid edges in C^+. The construction of Q from P is illustrated in Figs. 5c, 5d, and 5e. As all the solid edges have zero weight and length, Q must have less total edge weight than P. This contradicts the assumption that P is an MWPM. Therefore, even when P'' contains multiple alternate cycles, the problem of finding an NWA cycle in G can still be reduced to the problem of finding an MWPM in G simply by choosing from P'' the cycle with minimum cycle ratio $\Gamma_{rc}(C)$ (based on the original edge weights) for the sequential-search algorithm from Section 3.2. This cycle brings b closer to the desired b^* than any other cycle in P''. Although the earlier analysis provides an upper-bound of $w_{max}l_{max}|E|^2$ on the number of iterations required in the sequential-search algorithm from Section 3.2, none of the experiments reported in the paper requires more than eight iterations using this NWA cycle-detection algorithm.

4 Fragment Construction and Edge-Weight Function

This section discusses the supplementary processing steps needed to use the above ratio-contour algorithm to extract salient boundaries from real images: the construction of the real and virtual fragments and the calculation of edge weights and lengths. Specifically, we discuss the following issues: 1) preprocessing the edge-detector output to produce a set of topologically unconnected fragments, 2) smoothing the detected fragments to remove noise, and 3) reliably estimating the curved gap-filling segment between two detected fragments. The latter two use a spline-based curve-smoothing algorithm for noise removal and robust gap filling.

4.1 Preprocessing the Edge-Detector Output

We must construct fragments from the traces produced by an edge detector that meet the requirements of Section 2 and Fig. 1a. However, edge detectors may produce traces that contain *intersections*, as illustrated in Fig. 6a, *attachments*, as illustrated in Fig. 6c, and *closed curves*, as illustrated in Fig. 6e.

Intersections arise when more than two apparent fragments are incident on the same point, as shown in Fig. 6a. When using a Canny edge detector [7], it is trivial to identify the intersection points since all the traced pixels are ordered after nonmaximum suppression. Thus, we can identify intersection points as traced pixels with more than two neighbors. The traces are split at these intersection points to derive multiple real fragments. For example, the traces shown in Fig. 6a yield multiple fragments that correspond to the solid and dashed edges in Fig. 6b. In this example, the intersection point is split into three endpoints that correspond to three vertices u_1, u_2, and u_3 that are connected by dashed edges with zero weight and length.

Attachments arise when a long trace must be divided into more than one fragment so that the desired boundary can be constructed from indivisible fragments, as shown in Fig. 6c. In this example, we wish to divide the long trace $B(e_1) \cup B(e_2)$ into the fragments $B(e_1)$ and $B(e_2)$ to allow the extracted boundary to connect $B(e_1)$ to $B(e_3)$ and exclude $B(e_2)$. We accomplish this by dividing all traces at points of high curvature. For example, the traces in Fig. 6c yield the fragments that correspond to the solid and dashed edges in Fig. 6d. In this example, after splitting the long trace into two fragments at the point of high curvature, the two coincident endpoints of these fragments correspond to two vertices u_1 and u_2 that are connected by a dashed edge with zero weight and length. *Closed curves*, as shown in Fig. 6e, can be treated as a special case of attachments. We simply divide the trace at the highest point of curvature, as shown in Fig. 6f.

In dealing with both attachments and closed curves, we need to identify the points of high curvature along the trace. However, traces derived directly from edge detection may contain noise that prevents accurate estimation of the curvature (see Fig. 7a). In the next section, we present a spline-based curve-smoothing technique for noise removal.

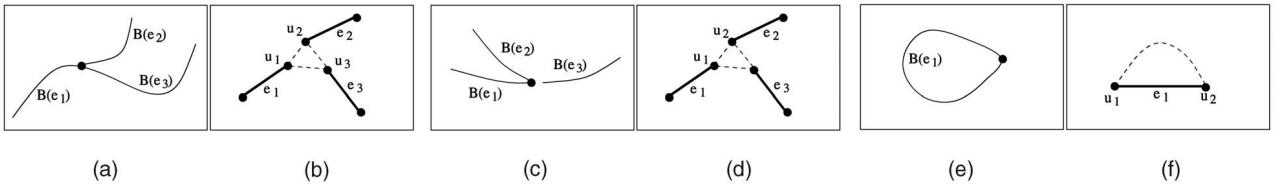


Fig. 6. Constructing fragments from traces. (a) and (b), (c) and (d), and (e) and (f) illustrate how to construct fragments from traces with intersections, attachments, and closed curves, respectively.

4.2 Smoothing the Detected Fragments

To accurately estimate the curvature along a trace, we need to remove noise and aliasing. To accomplish this, we employ a spline-based smoothing technique that represents a trace by a set of quadratic splines. Note that each trace is originally represented by a set of connected pixel points extracted by an edge detector. We refer to these pixel points as *control points* and associate a quadratic spline to each control point that interpolates that control point and its neighbor. The spline for the ith control point has the following parametric form:

$$\begin{pmatrix} x_i(t) \\ y_i(t) \end{pmatrix} = \begin{pmatrix} \bar{x}_i \\ \bar{y}_i \end{pmatrix} + \begin{pmatrix} A_i & B_i \\ C_i & D_i \end{pmatrix} \begin{pmatrix} t^2 \\ t \end{pmatrix},$$

where $\mathbf{p}_i(t) = (x_i(t), y_i(t))^T$ is the spatial coordinate of the ith spline parameterized with $t \in [-1, 1]$, $\bar{\mathbf{p}}_i = (\bar{x}_i, \bar{y}_i)^T$ is the spatial coordinate of the control point and A_i, B_i, C_i, and D_i are the coefficients of the quadratic. We smooth these splines by visiting each in turn and adjusting its attributes ($\bar{x}_i$, $\bar{y}_i$, A_i, B_i, C_i, and D_i) to improve a continuity measure. The process repeats iteratively until the measure reaches a given tolerance level. Thus, among all the representations whose continuity measure is under the tolerance level, the process returns the first one to be reached from the original representation by the iterative optimization process [34]. While we cannot prove that such a smoothed fragment remains aligned with the original unsmoothed fragment, in practice, we have never observed a case of significant misalignment. Furthermore, with the parametric form of the quadratic splines associated with each trace, the total length of the ith spline and the curvature along that spline can be computed by

$$l_i = \int_0^1 \sqrt{(2A_i t + B_i)^2 + (2C_i t + D_i)^2}\, dt,$$

$$\kappa_i(t) = \frac{2(B_i C_i - A_i D_i)}{\left((2A_i t + B_i)^2 + (2C_i t + D_i)^2\right)^{3/2}}.$$

This can be used to calculate the edge weights and lengths in the constructed SD graph.

To derive the optimal attributes for these quadratic splines, we form a prior energy term that penalizes deviation away from C^1 continuities. We use the following energy function:

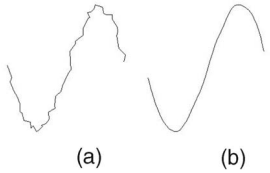


Fig. 7. Trace smoothing. (a) A noisy trace. (b) The same trace after smoothing.

$$E_p = \sum_{i=1}^{N-1} \Big\{ \|\mathbf{p}_i(0) - \mathbf{p}_{i+1}(-1)\|^2 + \|\mathbf{p}_i(1) - p_{i+1}(0)\|^2 + \|\dot{\mathbf{p}}_i(0) - \dot{\mathbf{p}}_{i+1}(-1)\|^2 + \|\dot{\mathbf{p}}_i(1) - \dot{\mathbf{p}}_{i+1}(0)\|^2 \Big\},$$

where $\|\cdot\|^2$ is the L_2 norm and $\dot{\mathbf{p}}(t)$ is the tangent vector at t. The smoothing process iteratively visits each spline sequentially to update its attributes [34]. This can be performed using a block Gauss-Seidel method. Throughout the experiments presented in this paper, we iterate the Gauss-Seidel process 50 times to smooth each trace. An example of trace smoothing using this method is shown in Fig. 7.

4.3 Gap Filling

The final supplementary processing step is to derive virtual fragments from the real fragments. One approach is to estimate a virtual fragment that interpolates each pair of real-fragment endpoints in a way that satisfies specified continuity and perceptual criteria. This approach is often called *curve completion* [21], [50], [31]. For example, Figs. 8a and 8b show a closed boundary and the same boundary with three gaps, respectively. In the ideal case, we wish curve completion to derive three virtual fragments to fill these gaps, as shown in Fig. 8c, and produce a boundary that resembles the original one in Fig. 8a.

However, curve completion does not work properly with noisy or aliased real fragments. For example, noise at the fragment endpoints shown in Fig. 8d may result in virtual fragments that deviate significantly from the original curve, as shown in Fig. 8e, because the noisy tangent values at fragment endpoints dominate this curve-completion process. Real fragments derived by preprocessing edge-detector traces with the methods from Sections 4.1 and 4.2 still may suffer from such endpoint noise and aliasing. Therefore, curve completion must use more information than is available at fragment endpoints.

In this paper, we estimate a virtual fragment from the general shape of its adjacent real fragments because fragment shape information is less susceptible to noise and aliasing. This way, the estimation of virtual fragments and the removal of the real-fragment endpoint noise and aliasing are combined into a single task. Specifically, each real fragment is divided into two real subfragments at its midpoint. Each real subfragment is then combined into its adjacent virtual fragment to form a *combined fragment*. As illustrated in Fig. 8f, the closed boundary consists of three combined fragments $\overrightarrow{AB}$, $\overrightarrow{BC}$, and $\overrightarrow{CA}$, where $\overrightarrow{AB}$ represents the curve segment traversing from A to B in the clockwise direction. We can see that each combined fragment is made up of a virtual fragment and two of their adjacent real subfragments. Instead of estimating the virtual fragments from the endpoints, we estimate the whole combined fragment that fits the included real subfragments while filling the gap smoothly. We not only

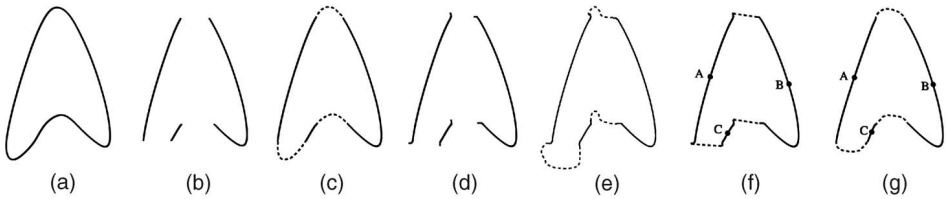


Fig. 8. Curve completion and combined fragments. (a) A smooth closed boundary. (b) The boundary in (a) with three gaps. (c) Curve completion with endpoint interpolation. (d) Endpoints with noise. (e) Curve completion using noisy endpoints. (f) Estimating initial virtual fragments by connecting corresponding noisy endpoints with straight lines. (g) Smoothing all three combined fragments.

allow the resulting combined fragment to deviate from the original real fragments, but also allow different combined fragments that are derived from the same real fragment to deviate differently from that real fragment.

We use a two-step algorithm to obtain the desired combined fragment. In the first step, we estimate all the virtual fragments by connecting the corresponding endpoints with straight lines, as shown in Fig. 8f. Combining the obtained virtual fragments and real fragments, we construct an initial representation of the combined fragments by dividing real fragments at their midpoints, as also shown in Fig. 8f. In the second step, we further smooth each combined fragment using the spline-based algorithm from Section 4.2. As illustrated in Fig. 8g, this process can alleviate endpoint noise during curve completion. In this smoothing process, we need to keep adjacent combined fragments smoothly connected to achieve boundary smoothness. For example, after smoothing, the combined fragment $\overrightarrow{AB}$ should still be smoothly connected to the combined fragment $\overrightarrow{BC}$ at point B. This problem can be addressed by fixing the coordinates and tangents of the connection points (points A, B, and C in Fig. 8f). This guarantees a smooth transition from $\overrightarrow{AB}$ to $\overrightarrow{BC}$ when forming a boundary. The coordinates and tangents of those connection points are calculated from the real fragments after the preprocessing step from Section 4.1.

Finally, we need to derive the edge weights and lengths from these smoothed combined fragments. The main challenge comes from the fact that real fragments may be smoothed in different ways when combined with different adjacent virtual fragments to yield different combined fragments.

The first reduction from Section 3.1 addresses this challenge. The weights and lengths of all solid edges can be set to zero after reassigning those weights and lengths to their adjacent dashed edges. Therefore, we can calculate the edge weights and lengths from the corresponding combined fragment, but only associate those weights and lengths with the relevant dashed edges (or virtual fragments). Assume e is a dashed edge whose corresponding virtual fragment $B(e)$ is contained in the smoothed combined fragment $B_c(e)$. In the constructed SD graph G, we can take the weight and length of this dashed edge e to be

$$w(e) = \int_{B_c(e)} [\sigma(t) + \lambda \cdot \kappa^2(t)]dt, \quad l(e) = \int_{B_c(e)} dt$$

and set the weight and length of all solid edges to be zero. Therefore, in our implementation of the ratio-contour method, the first reduction from Section 3.1 is implicitly achieved during fragment construction.

5 Experiments

We tested the ratio-contour method by extracting salient boundaries from real images in Figs. 9, 10, and 11. For fragment construction, we use the standard Canny edge detector in the Matlab (R13 with SP1) with its default threshold and smoothing settings. Our implementation of ratio contour uses the `Blossom4` implementation [10] of MWPM. Note that the number of dashed edges (or virtual fragments) may be very large, i.e., $O(|V|^2)$, if we fill all the possible gaps. Thus, to reduce the number of dashed edges, we only fill gaps whose endpoints are sufficiently close. More specifically, we fill at most a bounded number of shortest gaps incident on each real fragment endpoint. We use a bound of 20 in all experiments reported in this paper. Additionally, we set $\lambda = 50$ when computing edge weights. Finally, we discard all real fragments whose length is less than five pixels because it is difficult to derive accurate tangent and curvature information from very short noisy fragments. Fig. 9 illustrates the operation of the entire ratio-contour implementation on a sample image. We further tested ratio contour on 10 natural images and 10 medical images. The results are shown in the left four columns of Figs. 10 and 11, respectively. Note that all of the extracted boundaries are closed and nondegenerate.

We can apply ratio contour iteratively to extract multiple salient boundaries. For simplicity, we assume that no two salient boundaries share fragments. Under this assumption, we can use ratio contour to extract the most-salient boundary, remove from G the solid edges in the extracted boundary together with all adjacent dashed edges, and iterate this process to repeatedly extract subsequent salient boundaries. This process is illustrated in Fig. 12 and applied to real images in Fig. 13.

6 Related Work

We now compare ratio contour (RC) to the two most-related prior methods, namely, those of Elder and Zucker (EZ) [16] and Williams and Thornber (WT) [65], [37]. Among all prior work on boundary extraction, these two prior methods are most closely related to RC because all three extract closed boundaries. Note that, while [65] compares WT with five other methods, none of these other methods are guaranteed to produce closed boundaries. Thus, we do not compare RC with these five other methods. Further note that our comparison study differs from that in [65] in that our study evaluates whether the various methods can guarantee closure. We first give a brief introduction to EZ and WT.

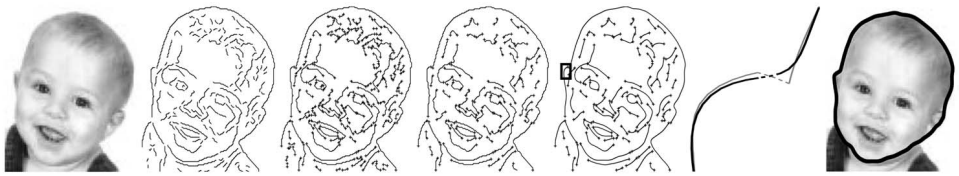


Fig. 9. The operation of the entire ratio-contour implementation on a sample image. (a) A sample image. (b) Output of edge detection. (c) The result after tracing and dividing intersection points in (b). Intersections and endpoints are indicated with crosses. (d) The result after pruning fragments with less than five pixels from (c). (e) Real fragments after all supplementary preprocessing. (f) An enlarged view of the area enclosed by the box in (e) that illustrates the smoothing of a combined fragment. The light and dark curves depict the combined fragment before and after smoothing. (g) The extracted salient boundary (the thick black curve) using ratio contour.

6.1 Elder & Zucker's Method (EZ) [16]

EZ formulates boundary extraction in a *directed* graph $\hat{G} = (\hat{V}, \hat{E})$, where real fragments are degenerately represented as straight line segments that correspond to pairs of vertices i and $\bar{i}$, one for each direction of the fragment. Virtual fragments are represented by pairs of directed edges between the vertices corresponding to directed real fragments. Fig. 14 illustrates the construction of this directed graph. Associated with each (directed) edge (i, j) is a weight $p_{ij} \in (0, 1)$ that represents the fragment prominence of the directed virtual fragment corresponding to that edge. EZ constructs these weights to encode the properties of proximity and continuity, as well as the neighboring-area's intensity distribution. Note that p_{ij} usually differs from p_{ji} while $p_{ij} = p_{\bar{j}\bar{i}}$. Also note that the number of vertices in $\hat{G}$ is the same as the number of vertices in G, the undirected graph used by RC, when both are constructed from the same set of real fragments.

EZ models a closed boundary as a directed cycle $\hat{C}$ in $\hat{G}$. The saliency of such a boundary is defined as:

$$S_{ez}(\hat{C}) = \prod_{(i,j)\in\hat{C}} p_{ij}. \tag{6}$$

EZ finds the cycle $\hat{C}^*$ that maximizes this boundary-saliency measure by minimizing the boundary cost

$$\Gamma_{ez}(\hat{C}) = -\sum_{(i,j)\in\hat{C}} \log p_{ij}.$$

If we redefine the edge weights in $\hat{G}$ as $\hat{w}_{ij} = -\log p_{ij}$, $\hat{C}^*$ is then a (directed) cycle with minimum total weight. Such a cycle can be found in polynomial time with a shortest-path algorithm.

6.2 Williams & Thornber's Method (WT) [65], [37]

WT formulates boundary extraction in the same (directed) graph $\hat{G}$ where each directed real fragment has infinitesimal length and is represented as a point and a direction. The fragment prominence p_{ij} in WT is estimated using stochastic-completion fields [64], which model the proximity and continuity between two real fragments. Real-fragment prominence is initially ignored. WT uses spectral analysis to enhance local fragment prominence to encode global boundary closure. This is done by updating the affinity matrix $\mathbf{P} = [p_{ij}]_{|V|\times|V|}$ to $\mathbf{C} = [c_{ij}]_{|V|\times|V|}$ $(0 \le c_{ij} \le 1)$ using an eigenvalue decomposition. To extract closed boundaries, WT uses a strongly-connected-components algorithm for finding a directed cycle $\hat{C} \subset \hat{G}$ that traverses the edges with large fragment prominence c_{ij} [37]. WT [65], [37] shows that, in certain cases, this process extracts the boundary that maximizes the boundary saliency

$$S_{wt}(\hat{C}) = \left(\prod_{(i,j)\in\hat{C}} p_{ij} \right)^{\frac{1}{|\hat{C}|}}. \tag{7}$$

6.3 Analysis

We analyze the similarities and differences between EZ, WT, and RC along four axes:

1. the structure of the graphs constructed,
2. the search spaces,
3. the boundary saliency measure, and
4. the optimality of the boundaries produced, for each method's respective search space and boundary saliency measure.

From this analysis, we produce a unified framework that enables comparison of the three methods using the same fragment prominence measure. This focuses the comparison on the different boundary saliency measures, search spaces, and optimization algorithms.

6.3.1 The Structure of the Graphs Constructed

EZ and WT use the same graph $\hat{G}$. There is a one-to-one correspondence between $\hat{G}$ and the graphs G used by RC. Real fragments are represented by solid edges in G and pairs of directed vertices in $\hat{G}$. Virtual fragments are represented by dashed edges in G and directed edges in $\hat{G}$. Both graphs describe precisely the real and virtual fragments together with their connection relations.

6.3.2 The Search Spaces

All three methods search a different space. RC searches the space $\mathcal{B}$ of boundaries that correspond to all the alternate (simple) cycles $C \subset G$. These are precisely the nondegenerate closed boundaries. EZ and WT search the space $\hat{\mathcal{B}}$ of boundaries that correspond to all the directed cycles $\hat{C} \subset \hat{G}$. Because $\hat{C}$ alternately traverses edges and vertices in $\hat{G}$, the resulting boundary also alternately traverses real and virtual fragments, as in RC. However, a directed simple cycle $\hat{C} \subset \hat{G}$ may traverse both the edge (i, j) and the edge $(\bar{j}, \bar{i})$ that represent the same virtual fragment. Therefore, $\hat{\mathcal{B}}$ contains degenerate closed boundaries, such as the one shown in Fig. 15c, that correspond to cycles such as the one shown in Fig. 15a. In G, such a degenerate boundary is represented by a nonsimple alternate cycle (i.e., one that traverses a vertex more than once) and is excluded from the search space $\mathcal{B}$. Therefore, $\mathcal{B} \subset \hat{\mathcal{B}}$ and $\hat{\mathcal{B}} \setminus \mathcal{B}$ constitute the set of all

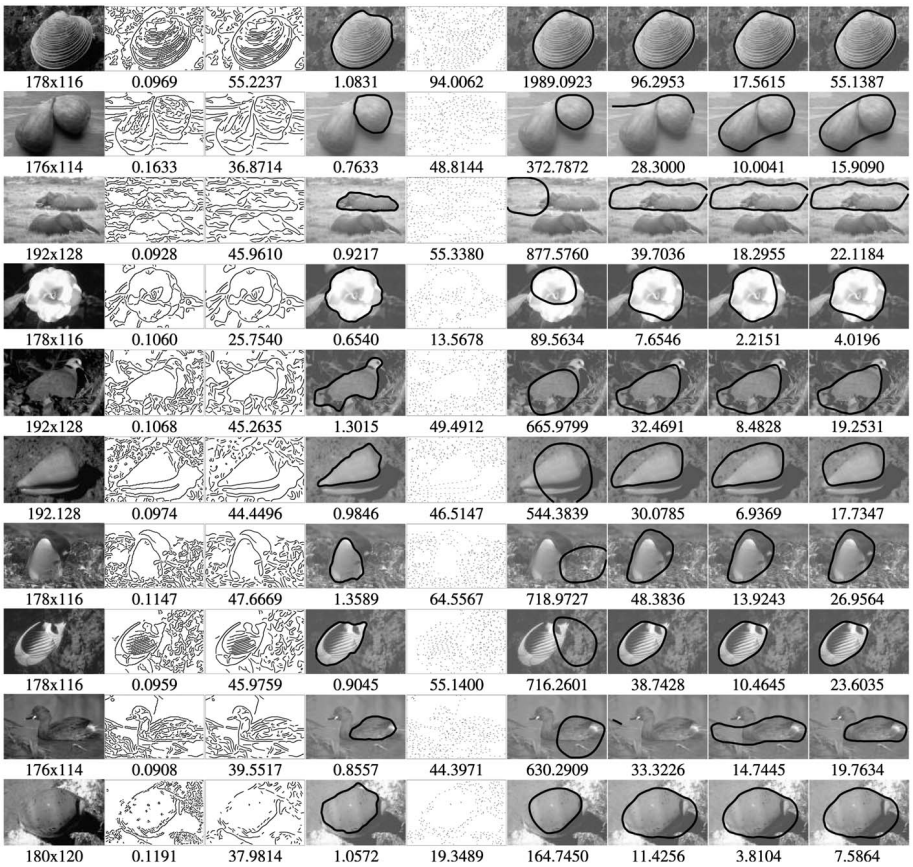


Fig. 10. Boundary extraction on 10 natural images. Column 1: original image. Column 2: output of Canny edge detection. Column 3: real fragments constructed by supplementary preprocessing. Column 4: the most-salient boundary extracted by RC on column 3. Column 5: real fragments constructed by the unified framework. Columns 6 through 9: the most-salient boundaries detected by EZ, WT, (P-)RC, and C-RC on column 5. Columns 2 through 4 correspond to experiments discussed in Section 5. Columns 5 through 9 correspond to experiments discussed in Section 7. Image size is indicated under the images in column 1 and processing time in CPU seconds is indicated under the images in the remaining columns. See Section 7.5 for details on the measurement of processing time. Note that some closed boundaries have been cropped by the image perimeter when producing this figure.

degenerate closed boundaries. While, in principle, EZ can produce degenerate boundaries because it searches $\hat{\mathcal{B}}$, it rarely produces degenerate boundaries, in practice, because, as analyzed later, EZ has a bias toward boundaries with fewer fragments.

WT employs the following local-search technique for finding strongly connected components in an attempt to produce nondegenerate closed boundaries: The technique selects an initial vertex k, expands the boundary by selecting the unvisited adjacent vertex j with maximum c_{kj}, and repeats this process on j until arriving back at k. This produces a cycle $\hat{C}_k$. This same process is used to produce a cycle $\hat{C}_{\bar{k}}$ starting from $\bar{k}$. A cycle $\hat{C}'_{\bar{k}} = \{\bar{i} | i \in \hat{C}_{\bar{k}}\}$ is then constructed from $\hat{C}_{\bar{k}}$. Note that $\hat{C}_k$, $\hat{C}_{\bar{k}}$, and $\hat{C}'_{\bar{k}}$ all correspond to closed boundaries because they are necessarily cycles. However, they may be degenerate. WT attempts to produce nondegenerate boundaries by taking $\hat{C}_k \cap \hat{C}'_{\bar{k}}$ to derive a boundary. However, $\hat{C}_k \cap \hat{C}'_{\bar{k}}$ might not be a cycle and, thus, might not correspond to a closed boundary. For example, consider the vertices k and $\bar{k}$ and the cycle $\hat{C}_k$ in Fig. 15a. Since $\hat{C}_k$ contains both k and $\bar{k}$, $\hat{C}_k$ and $\hat{C}_{\bar{k}}$ can be the same cycle. In this case, the subgraph $\hat{C}_k \cap \hat{C}'_{\bar{k}}$ corresponds to an open curve segment, as shown in Fig. 15d. Such situations arise in the experiments presented in the next section.

6.3.3 The Boundary Saliency Measure

All three methods adopt a different boundary saliency measure. As seen by (6), EZ formulates boundary saliency as the product of fragment prominence. As seen by (7), WT formulates boundary saliency as the geometric mean of fragment prominence. As pointed out in [65], EZ's boundary saliency decreases more quickly than WTs as the number of fragments along the boundary increases. Therefore, EZ favors boundary with fewer fragments that, in many cases, results in short boundaries.

To compare the boundary saliency measures of WT and RC, we can redefine the edge weight $\hat{w}_{ij} = -\log p_{ij}$ in $\hat{G}$ to represent fragment cost. Then, the cycle $\hat{C}^*$ that maximizes the boundary saliency (7) is the same as the one that minimizes the boundary cost

$$\Gamma_{wt}(\hat{C}) = \frac{\sum_{(i,j)\in\hat{C}} \hat{w}_{ij}}{|\hat{C}|}.$$

This has the same form as the RC boundary cost (4) when limited to the case where solid edges in G have zero weight and length and dashed edges have unit length. While WT and RC employ similar boundary saliency measures in this limited case, they can still produce different results, even in this limited case, because they search different spaces.

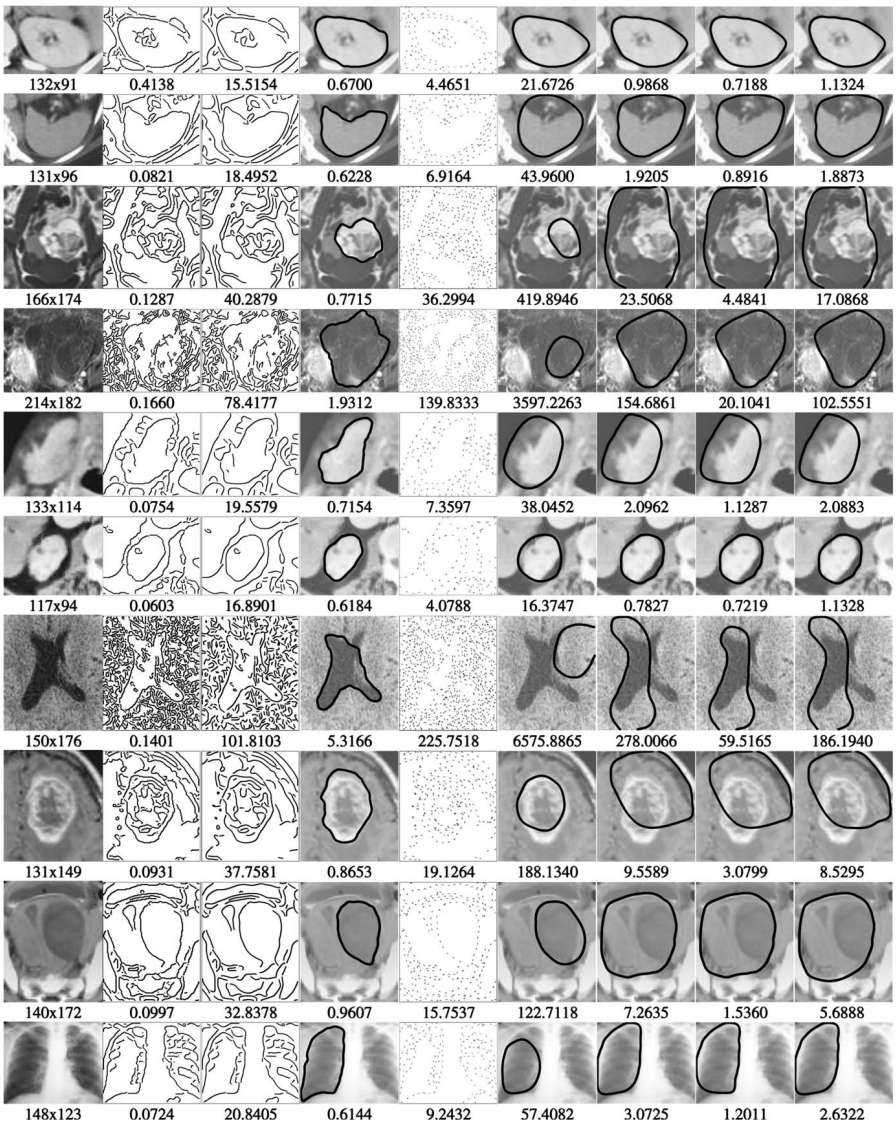


Fig. 11. Boundary extraction on 10 medical images. The columns depict the same information as in Fig. 10.

6.3.4 The Optimality of the Boundaries Produced

Elder and Zucker [16] show that EZ is guaranteed to find a boundary that optimizes the boundary saliency measure Γ_{ez} within the search space $\hat{\mathcal{B}}$ of closed boundaries whether degenerate or nondegenerate. This paper shows that RC is guaranteed to find a closed boundary that optimizes the boundary saliency measure Γ_{rc} within the search space $\mathcal{B}$ of nondegenerate closed boundaries. We know of no proof that WT is guaranteed to find a boundary that optimizes the boundary saliency measure Γ_{wt} within either $\hat{\mathcal{B}}$ or $\mathcal{B}$.

6.3.5 Unified Framework

The three methods use different kinds of real fragments. In EZ, real fragments are line segments. In WT, real fragments are line segments with infinitesimal length. In RC, real fragments are quadratic splines. To facilitate experimental comparison, we apply all three methods to the kind of real fragments used by WT, namely, line segments with infinitesimal length. The three methods use different approaches to gap filling for constructing virtual fragments. To facilitate experimental comparison, we apply all three methods to virtual fragments constructed using the stochastic-completion field method [64] of WT with the parameters $\gamma = 0.15$, $T = 0.004$, and $\tau = 5.0$, as suggested in [64], [37].

The three methods use different fragment prominence measures. In EZ and WT, edge weights p_{ij} represent fragment prominence, while, in RC, edge weights $w(\cdot)$ represent fragment cost. Furthermore, EZ and WT only associate prominence with virtual fragments, not real fragments, while RC associates costs with both. Finally, in WT, the number of virtual fragments along the boundary, i.e., $|\hat{C}|$, is used to measure the boundary length, while, in RC, boundary length is taken to be the total fragment length.

To facilitate experimental comparison, we apply all three methods to the same prominence measure constructed as follows: First, we take the weights p_{ij} used by EZ and WT to be related to the weights $w(\cdot)$ used by RC by the relation $w(i_2, j_1) = -\log p_{ij}$, given that the undirected dashed edge (i_2, j_1) in G and the directed edge (i, j) in $\hat{G}$ correspond to the same virtual fragment, as shown in Fig. 14. Second, we take the length of all dashed edges in G to be one and the weight and length of all solid edges in G to be zero. While this fragment prominence

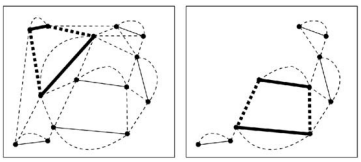


Fig. 12. Extracting multiple salient boundaries. (a) The original SD graph from Fig. 1e. (b) The new SD graph after extracting the most-salient boundary corresponding to the cycle (thick lines) shown in (a). Note that from (a) to (b), we remove not only all solid and dashed edges in the cycle corresponding to the most-salient boundary, but also all the dashed edges adjacent to the removed solid edges. A second iteration of ratio contour on this new SD graph yields the second most-salient boundary, shown in thick lines in (b).

measure is a special case of the ones used in EZ and RC, it is compatible with all three algorithms.

7 Comparison Study

Unless otherwise noted, all the experiments reported in this section use the unified framework of Section 6.3.

7.1 An Illustrative Example

We construct a simple synthetic example to show that WT can produce open curve segments. The real fragments in this examples are shown in Fig. 16a. Sixteen real fragments were constructed by sampling two circles of radius 100 and seven addition real fragments were constructed by sampling a straight line segment placed between these two circles. The exact information needed to reconstruct this example is given in Table 1. Figs. 16b, 16c, and 16d show the most-salient boundaries extracted using EZ, WT, and RC, respectively. This confirms that WT can produce open curve segments, as discussed in Section 6.3.

7.2 Synthetic Data

As in [65], we constructed a set of synthetic data to quantitatively evaluate the performance of EZ, WT, and RC. In this data, the input real fragments contain two superimposed patterns: a foreground pattern and a background pattern. The foreground pattern is selected from the nine objects shown in Fig. 17, where the object boundary is uniformly sampled to construct the real fragments for the foreground pattern. The first five fruit patterns shown in Fig. 17 were used in [65]. As in [65], the real fragments for the background pattern is derived from nine texture images: bark, fabric, sand, terrain, wood, brick, leaves, stone, and water, by uniformly sampling their edge-detector outputs. In this experiment, we sample each foreground-object boundary with seven different sampling rates to construct a set of synthetic data with varied signal to noise ratio (SNR). Here, we define the SNR as the ratio between the number of real fragments from the foreground pattern and the number of real fragments from the background pattern. This way, we have a total of $9 \times 9 \times 7 = 567$ synthetic data samples for our experiments by superimposing a foreground pattern with a background pattern.

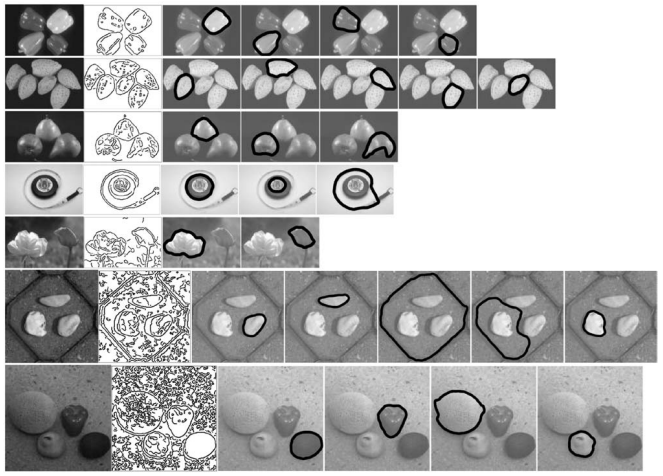


Fig. 13. Extracting multiple salient boundaries from seven real images with iterative ratio contour. Each row depicts an original image, the edge-detector output, and the sequence of boundaries extracted for that image. The images in the last two rows come from [37].

We applied both EZ, WT, and RC to extract the most-salient boundaries from all these 567 data samples. The accuracy is then evaluated by comparing the extracted salient boundaries and the underlying foreground object boundaries. We evaluate the accuracy using two measures: a region-based measure and a fragment-based measure. The region-based accuracy measure is defined as the relative region coincidence,

$$\frac{|R_1 \cap R_2|}{|R_1 \cup R_2|},$$

where R_1 is the region of the foreground object, R_2 is the region enclosed by the detected salient boundary, and $|R|$ denotes the area of R. The fragment-based accuracy measure is defined as the percentage of real fragments for the foreground pattern that are included in the detected salient boundary. Figs. 18a and 18c show the performance of EZ, WT, and RC on all 567 synthetic data samples using the above two measures. Note that the accuracy at each SNR is calculated by taking the average accuracy on $9 \times 9 = 81$ data samples with this SNR. In this experiment, WT produces open curve segments in 170 data samples out of the 567 data samples and its performance is degraded greatly by these 170 samples. Figs. 18b and 18d show the performance of EZ, WT, and RC on the remaining 397 data samples where all three methods produce closed boundaries. We see that WT has no better performance than RC, even if we only count these 397 data samples. Four examples of these experiments are shown in Fig. 19, where WT produces open curve segments in the second and third images.

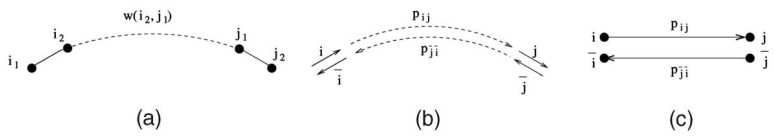


Fig. 14. The directed graph $\hat{G}$ used in EZ and WT. (a) Initial real/virtual fragments and the SD graph G, (b) directed real/virtual fragments, and (c) the resulting directed graph $\hat{G}$ analogous to (a). This figure is adapted from Fig. 3 from [65].

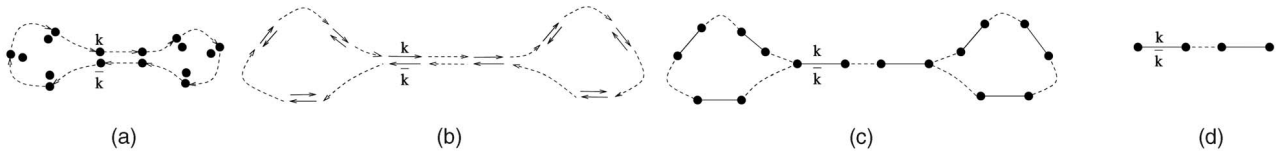


Fig. 15. Open curve segments produced by WT. (a) A directed cycle that traverses both k and $\bar{k}$ in $\hat{G}$, where vertices are always paired to represent the two possible directions of real fragments. (b) The same cycle represented by directed fragments. (c) The degenerate boundary corresponding to the cycle in (a) and (b). (d) An open curve segment produced by the strongly-connected-components algorithm of WT.

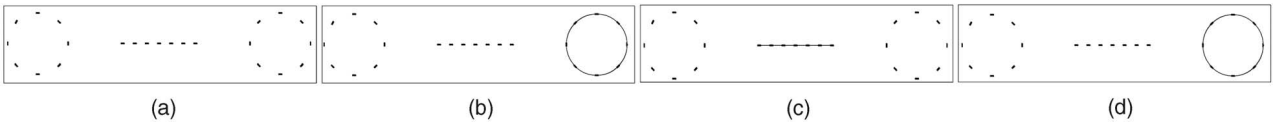


Fig. 16. A demonstration that WT can produce open curve segments. (a) The real fragments. (b)-(d) The boundaries extracted using EZ, WT, and RC, respectively.

7.3 Real Images

We evaluated the three methods on the real images shown in the first column of Figs. 10 and 11 using the unified framework of Section 6.3. Note that column 4 of these figures illustrates the output of RC using the full supplementary preprocessing method from Section 4, while column 8 illustrates the output of RC using the unified framework. Our unified framework constructed real fragments with infinitesimal length, shown in column 5, by sampling the traces produced by the Canny edge detector in column 2 at an interval of 8 pixels. We further smooth the traces with the technique from Section 4.2 before sampling to help achieve more reliable estimates of fragment direction when constructing the real fragments in column 5. Columns 6, 7, and 8 show the results of applying EZ, WT, and RC, respectively, on the real fragments from column 5. Note that EZ sometimes fails to extract the desired boundary, as witnessed by the fourth image in Fig. 10 and the third, fourth, eighth, and 10th images in Fig. 11, and instead yields a boundary with fewer fragments. Also note that WT sometimes extracts open curve segments, as witnessed by the second and ninth images in Fig. 10. Finally, note that RC with supplementary preprocessing (column 4) sometimes yields different results than RC with the unified framework (column 8). One advantage of the fragment cost used in (1) is that it has only one free parameter λ, while the fragment prominence based on stochastic-completion fields has three free parameters γ, T, and τ. Another advantage of the boundary cost (1) is that it is based on boundary length, which includes the lengths of both real and virtual fragments, rather than the number of virtual fragments in the boundary.

TABLE 1
The Data Used to Illustrate that WT Can Produce Open Curve Segments

Left circle (8 fragments)			Right circle (8 fragments)			Straight line (7 fragments)		
y	x	θ	y	x	θ	y	x	θ
85.3553	85.3553	2.3562	85.3553	485.3553	2.3562	50.0000	190.0000	0.0000
100.0000	50.0000	3.1416	100.0000	450.0000	3.1416	50.0000	210.0000	0.0000
85.3553	14.6447	3.9270	85.3553	414.6447	3.9270	50.0000	230.0000	0.0000
50.0000	0	4.7124	50.0000	400.0000	4.7124	50.0000	250.0000	0.0000
14.6447	14.6447	5.4978	14.6447	414.6447	5.4978	50.0000	270.0000	0.0000
0	50.0000	6.2832	0	450.0000	6.2832	50.0000	290.0000	0.0000
14.6447	85.3553	7.0686	14.6447	485.3553	7.0686	50.0000	310.0000	0.0000
50.0000	100.0000	7.8540	50.0000	500.0000	7.8540			

7.4 Should $\mathbf{P}$ Be Updated to $\mathbf{C}$ for RC?

As discussed in Section 6.2, WT attempts to limit the search to closed boundaries by first updating the local affinity matrix $\mathbf{P}$ (which measures prominence) to yield an enhanced matrix $\mathbf{C}$ that incorporates global closure information and then using a strongly-connected-components algorithm on $\mathbf{C}$ instead of $\mathbf{P}$. As shown in [37], running the strongly-connected-components algorithm on $\mathbf{C}$ produces significantly better results than running it on $\mathbf{P}$. A natural question to ask is whether RC also produces better results using $\mathbf{C}$ instead of $\mathbf{P}$ in the unified framework. To investigate this question, we conducted a series of experiments where we compare RC with $\mathbf{C}$ to RC with $\mathbf{P}$. We refer to the former as C-RC and to the latter as (P-)RC. In our unified framework, one can run RC with $\mathbf{C}$ by taking the edge weights to be $w(i_2, j_1) = -\log c_{ij}$.

We evaluate the performance of C-RC on the synthetic data from Section 7.2. The accuracy curves are also shown in Fig. 18, using both region-based and fragment-based measures. The results show that C-RC does not improve the performance of P-RC. We also compare P-RC with C-RC on the 20 real images in Figs. 10 and 11. The results of P-RC and C-RC are in columns 8 and 9, respectively. Note that there is no significant difference between P-RC and C-RC on these images, thus, our question is answered in the negative. While the strongly-connected-components algorithm of WT yields better results with $\mathbf{C}$ than with $\mathbf{P}$, the method of RC does not yield better results with $\mathbf{C}$ than with $\mathbf{P}$. The reason for this is that the update from $\mathbf{P}$ to $\mathbf{C}$ is redundant in RC because the purpose of this update is to encode closure in fragment prominence, but RC already limits the search space to include only closed boundaries. In fact, RC appears to return slightly better results with $\mathbf{P}$ than with $\mathbf{C}$ (see Fig. 18). This may be because the update from $\mathbf{P}$ to $\mathbf{C}$ encodes closure information for degenerate as well as nondegenerate boundaries (see Fig. 15). Therefore, using $\mathbf{C}$ for RC instead of $\mathbf{P}$ may misrepresent fragment prominence.

7.5 Algorithm Speed

7.5.1 Complexity Analysis

For EZ, the dominant step is the detection of cycles with minimum total weight. This is done with $O(|V|)$ applications of Dijkstra's shortest-path algorithm which has complexity $O(|V| \log |V|)$ when $|E| = O(|V|)$. Thus, the complexity of EZ

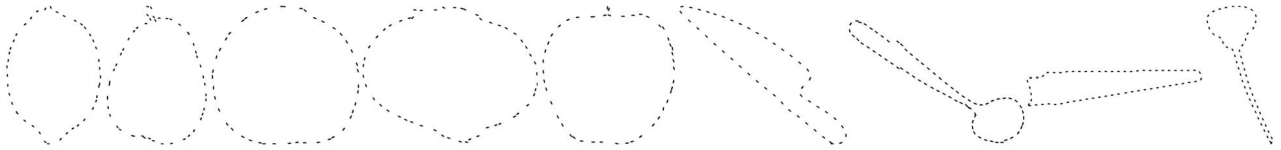


Fig. 17. The nine foreground patterns used for constructing synthetic data. From the left to the right are the patterns of a lemon, a pear, a peach, an onion, an apple, a knife, a spoon, a handsaw, and an oil can, respectively.

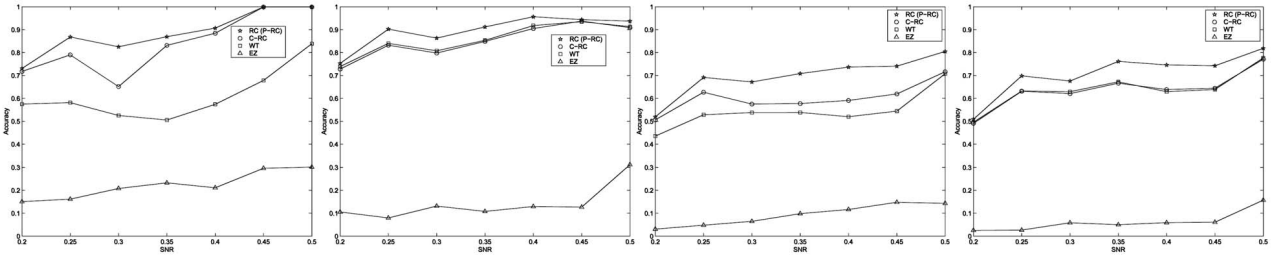


Fig. 18. Accuracy in extracting the foreground patterns out of background patterns. (a) Region-based accuracy computed on all 567 data samples. (b) Region-based accuracy computed on the 397 data samples where WT produces closed boundaries. (c) Fragment-based accuracy computed on all 567 data samples. (d) Fragment-based accuracy computed on the 397 data samples where WT produces closed boundaries. The curve for C-RC and the notation P-RC will be explained in Section 7.4.

is $O(|V|^2 \log |V|)$. For WT, the dominant step is to compute the principle eigenvector of $\mathbf{P}$, a reversal matrix whose size is $|\hat{V}| \times |\hat{V}|$. If $|\hat{E}| = O(|\hat{V}|)$, then $\mathbf{P}$ is sparse and has $O(|\hat{V}|)$ nonzero entries. Note that $|V| = |\hat{V}|$. The principle eigenvector of an arbitrary dense matrix of size $|V| \times |V|$ can be computed in $O(|V|^2)$ time. We do not know if the fact that $\mathbf{P}$ is sparse or is a reversal matrix can reduce this time complexity. For RC, the dominant step is the third reduction from Section 3, namely, MWPM. Gabow [17] shows that the time-complexity for finding an MWPM in an integer-weighted graph $G = (V, E)$ is $O(|V|^{\frac{3}{4}}|E| \log w_{max})$, where w_{max} is the maximum edge weight in the graph. In practice, since each vertex is only connected to a bounded number of vertices, $|E| = O(|V|)$. Thus, the complexity of RC is $O(|V|^{\frac{7}{4}})$.

7.5.2 Running Times

Figs. 10 and 11 indicate the running times, in CPU seconds, of various steps of the different methods. Column 2 indicates the time to perform edge detection. Column 3 indicates the time to perform supplementary preprocessing to construct the graph G. Column 4 indicates the running time of RC after constructing G. Column 5 indicates the time to construct $\hat{G}$. Columns 6 through 9 indicate the running times of EZ, WT, P-RC, and C-RC after constructing $\hat{G}$. All experiments were conducted on a Dell Precision Workstation equipped with a 3.06GHz Intel Xeon Processor with a 512K L2 Cache.

8 Conclusion

We have presented ratio contour, a novel method for extracting salient closed boundaries from noisy images. Starting with a set of boundary fragments that are detected in images using an edge detector, ratio contour is able to identify a subset of those fragments and connect them sequentially into a closed boundary with the largest saliency. Boundary saliency encodes the Gestalt laws of closure, proximity, and continuity. Our paper makes several contributions. First, it formulates a new boundary-saliency measure in terms of a ratio that is normalized relative to boundary length. This effectively avoids the possible bias toward short boundaries. Second, it presents a novel polynomial-time algorithm for finding the most-salient closed boundary with the saliency measure. Third, our method finds only closed nondegenerate boundaries by precisely constraining the search space to include only such desired boundaries. Fourth, it presents a novel collection of preprocessing methods to reduce the effects of noise on boundary detection. Finally, it constructs a unified framework for comparing ratio contour with two closely related previous methods, EZ and WT, and conducts such a comparison.

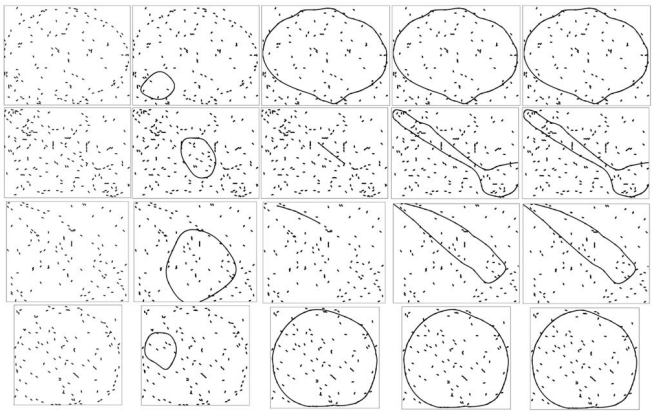


Fig. 19. Extracting salient boundaries on four synthetic data samples. From the leftmost column to the rightmost column are the real fragments and the boundaries extracted using EZ, WT, (P-)RC, and C-RC, respectively. The meaning of (P-)RC and C-RC will be explained in Section 7.4. From the top row to the bottom row are the superimposed patterns of onion and terrain, spoon and stone, knife and leaves, and peach and fabric, respectively.

Our comparison yields the following findings:

1. EZ shows a bias toward boundaries with fewer fragments, while WT and RC do not show such a bias.
2. WT may produce open curve segments, while RC is guaranteed to produce closed boundaries.
3. The space searched by EZ and WT includes degenerate boundaries, while the space searched by RC does not.
4. Adding the fragment-prominence update method used by WT to RC does not appreciably improve performance.
5. Experimental evaluation on natural and medical images shows that RC produces results that are as good as or better than EZ and WT.

We hope that the techniques underlying ratio contour will lead to future advances in boundary extraction.

ACKNOWLEDGMENTS

The authors would like to thank David Jacobs for important comments on an earlier draft of this paper. This work was funded, in part, by US National Science Foundation grants 0312861EIA and 0329156IIS. Some preliminary results from this paper were reported in [60] and [62].

REFERENCES

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

[2] T. Alter and R. Basri, "Extracting Salient Contours from Images: An Analysis of the Saliency Network," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 13-20, 1996.

[3] A.A. Amini, T.E. Weymouth, and R.C. Jain, "Using Dynamic Programming for Solving Variational Problems in Vision," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 12, no. 9, pp. 855-867, 1990.

[4] A. Amir and M. Lindenbaum, "A Generic Grouping Algorithm and Its Quantitative Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 2, pp. 168-185, Feb. 1998.

[5] A. Barbu and S.C. Zhu, "Graph Partition by Swendsen-Wang Cuts," *Proc. Int'l Conf. Computer Vision,* 2003.

[6] K. Bowyer, C. Kranenburg, and S. Dougherty, "Edge Detector Evaluation Using Empirical ROC Curves," *Computer Vision and Image Understanding,* vol. 84, no. 10, pp. 77-103, 2001.

[7] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 8, no. 6, pp. 679-698, 1986.

[8] V. Casselles, F. Catte, T. Coll, and F. Dibos, "A Geometric Model for Active Contours," *Numerische Mathematik,* vol. 66, no. 1, pp. 1-31, 1993.

[9] L.D. Cohen and I. Cohen, "A Finite Element Method Applied to New Active Contour Models and 3D Reconstruction from Cross Sections," *Proc. Int'l Conf. Computer Vision,* pp. 587-591, 1990.

[10] W. Cook and A. Rohe, "Computing Minimum-Weight Perfect Matchings," http://www.or.unibonn.de/home/rohe/matching.html, Aug. 1998.

[11] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham, "Active Shape Models—Their Training and Application," *Computer Vision and Image Understanding,* vol. 61, no. 1, pp. 38-59, 1995.

[12] T.H. Cormen, C.E. Leiserson, and R.L. Rivest, *Introduction to Algorithms.* Cambridge: MIT Press and New York: McGraw Hill, 1990.

[13] I. Cox, S.B. Rao, and Y. Zhong, "Ratio Regions: A Technique for Image Segmentation," *Proc. Int'l Conf. Pattern Recognition,* pp. 557-564, 1996.

[14] E. Dijkstra, "Note on Two Problems in Connection with Graphs," *Numerische Mathematik,* vol. 1, pp. 269-271, 1959.

[15] J. Edmonds, "Path, Trees, and Flowers," *Canadian J. Math.,* vol. 17, pp. 449-467, 1965.

[16] J. Elder and S. Zucker, "Computing Contour Closure," *Proc. European Conf. Computer Vision,* pp. 399-412, 1996.

[17] H.N. Gabow, "A Scaling Algorithm for Weighted Matching on General Graphs," *Proc. 26th Ann. Symp. Foundations of Computer Science,* pp. 90-100, 1985.

[18] M. Galun, E. Sharon, R. Basri, and A. Brandt, "Texture Segmentation by Multiscale Aggregation of Filter Responses and Shape Elements," *Proc. Int'l Conf. Computer Vision,* 2003.

[19] Y. Gdalyahu, D. Weinshall, and M. Werman, "Stochastic Image Segmentation by Typical Cuts," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 596-601, 1999.

[20] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 6, no. 6, pp. 721-741, 1984.

[21] G. Guy and G. Medioni, "Inferring Global Perceptual Contours from Local Features," *Int'l J. Computer Vision,* vol. 20, no. 1, pp. 113-133, 1996.

[22] X. Han, C. Xu, and J.L. Prince, "A Topology Preserving Deformable Model Using Level Set," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 765-770, 2001.

[23] P. Hart, N. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum-Cost Path," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 4, pp. 100-107, 1968.

[24] M.D. Heath, S. Sarkar, T. Sanocki, and K.W. Bowyer, "A Robust Visual Method for Assessing the Relative Performance of Edge-Detection Algorithms," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 12, pp. 1338-1359, Dec. 1997.

[25] L. Herault and R. Horaud, "Figure-Ground Discrimination: A Combinatorial Optimization Approach," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, pp. 899-914, 1993.

[26] L.A. Iverson and S.W. Zuker, "Logic/Linear Operators for Image Curves," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 10, pp. 982-996, Oct. 1995.

[27] D. Jacobs, "Robust and Efficient Detection of Convex Groups," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 1, pp. 23-37, Jan. 1996.

[28] A.K. Jain, Y. Zhong, and S. Lakshmanan, "Object Matching Using Deformable Templates," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 3, pp. 267-278, Mar. 1996.

[29] I.H. Jermyn and H. Ishikawa, "Globally Optimal Regions and Boundaries as Minimum Ratio Cycles," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 10, pp. 1075-1088, Oct. 2001.

[30] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," *Int'l J. Computer Vision,* vol. 1, no. 4, pp. 321-331, 1988.

[31] B.B. Kimia, I. Frankel, and A.-M. Popescu, "Euler Spiral for Shape Completion," *Int'l J. Computer Vision,* vol. 54, pp. 157-180, 2003.

[32] S. Konishi, A.L. Yuille, J.M. Coughlan, and S.C. Zhu, "Statistical Edge Detection: Learning and Evaluating Edge Cues," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 1, pp. 57-74, Jan. 2003.

[33] I. Kovacs and B. Julesz, "A Closed Curve is Much More than an Incomplete One: Effect of Closure in Figure-Ground Segmentation," *Proc. Nat'l Academy of Science USA,* vol. 90, pp. 7495-7497, 1993.

[34] T. Kubota, "Contextual and Non-Combinatorial Approach to Feature Extraction," *Proc. Int'l Workshop Energy Minimization Methods in Computer Vision and Pattern Recognition,* 2003.

[35] T.-S. Lee, D. Mumford, and A. Yuille, "Texture Segmentation by Minimizing Vector-Valued Energy Functionals: The Coupled-Membrane Model," *Proc. European Conf. Computer Vision,* pp. 165-183, 1992.

[36] M.E. Leventon, E.L. Grimson, and O. Faugeras, "Statistical Shape Influence in Geodesic Active Contours," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 316-323, 2000.

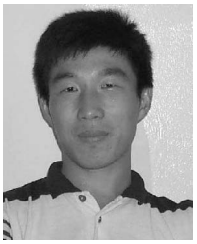
[37] S. Mahamud, L.R. Williams, K.K. Thornber, and K. Xu, "Segmentation of Multiple Salient Closed Contours from Real Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 4, pp. 433-444, Apr. 2003.

[38] D. Mumford, "Bayesian Rationale for the Variational Formulation," *Geometry-Driven Diffusion in Computer Vision,* B.M.H. Romeny, ed., pp. 135-143, Kluwer, 1994.

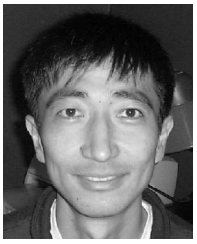
[39] M. Nitzberg, D. Mumford, and T. Shiota, "Filtering, Segmentation and Depth," *Lecture Notes in Computer Science,* vol. 662, New York: Springer-Verlag, 1993.

[40] P. Perona, "A Factorization Approach to Grouping," *Proc. European Conf. Computer Vision,* pp. 655-670, 1998.

[41] J.M. Prewitt, "Object Enhancement and Extraction," *Picture Processing and Psychopictorics,* B. Lipkin and A. Rosenfeld, eds., pp. 75-149, New York: Academic Press, 1970.
[42] S. Raman, S. Sarkar, and K. Boyer, "Hypothesizing Structures in Edge-Focused Cerebral Magnetic Resonance Images Using Graph-Theoretic Cycle Enumeration," *Computer Vision, Graphics, and Image Processing,* vol. 57, no. 1, pp. 81-98, 1993.
[43] K.R. Rao and J. Ben-Arie, "Optimal Edge Detection Using Expansion Matching and Restoration," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 16, no. 12, pp. 1169-1182, Dec. 1994.
[44] X. Ren and J. Malik, "A Probabilistic Multi-Scale Model for Contour Completion Based on Image Statistics," *Proc. European Conf. Computer Vision,* vol. 1, pp. 312-327, 2002.
[45] L.G. Roberts, "Machine Perception of Three-Dimensional Solids," *Optical and Electro-Optical Information Processing,* J.T. Tippett, ed., pp. 159-197, Cambridge, Mass.: Massachusetts Inst. of Technology Press, 1965.
[46] B.M.H. Romeny, *Geometry-Driven Diffusion in Computer Vision.* Dordrecht & Boston: Kluwer Academic Publishers, 1994.
[47] D. Rueckert and P. Burger, "Geometrically Deformable Templates for Shape-Based Segmentation and Tracking in Cardiac MR Image," *Proc. Int'l Workshop Energy Minimization Methods in Computer Vision and Pattern Recognition,* pp. 83-98, 1997.
[48] S. Sarkar and K. Boyer, "Quantitative Measures of Change Based on Feature Organization: Eigenvalues and Eigenvectors," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 478-483, 1996.
[49] S. Sarkar and P. Soundararajan, "Supervised Learning of Large Perceptual Organization: Graph Spectral Partitioning and Learning Automata," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 5, pp. 504-525, May 2000.
[50] E. Sharon, A. Brandt, and R. Basri, "Completion Energies and Scale," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 10, pp. 1117-1131, Oct. 2000.
[51] E. Sharon, A. Brandt, and R. Basri, "Fast Multiscale Image Segmentation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 70-77, 2000.
[52] E. Sharon, A. Brandt, and R. Basri, "Segmentation and Boundary Detection Using Multiscale Intensity Measurements," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 469-476, 2001.
[53] A. Shashua and S. Ullman, "Structural Saliency: The Detection of Globally Salient Structures Using a Locally Connected Network," *Proc. Int'l Conf. Computer Vision,* pp. 321-327, 1988.
[54] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 888-905, Aug. 2000.
[55] I.E. Sobel, "Camera Models and Machine Perception," PhD dissertation, Stanford Univ., Stanford, Calif., 1970.
[56] P. Soundararajan and S. Sarkar, "An In-Depth Study of Graph Partitioning Measures for Perceptual Organization," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 6, pp. 642-660, June 2003.
[57] L.H. Staib and J.S. Ducan, "Boundary Finding with Parametrically Deformable Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 11, pp. 1061-1075, Nov. 1992.
[58] K.K. Thornber and L.R. Williams, "Analytic Solution of Stochastic Completion Fields," *Biological Cybernetics,* vol. 75, pp. 141-151, 1996.
[59] O. Veksler, "Image Segmentation by Nested Cuts," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 339-344, 2000.
[60] S. Wang, T. Kubota, and J.M. Siskind, "Salient Boundary Detection Using Ratio Contour," *Proc. Neural Information Processing Systems Conf.,* pp. 1571-1578, 2003.
[61] S. Wang and J.M. Siskind, "Image Segmentation with Ratio Cut," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 6, pp. 675-690, June 2003.
[62] S. Wang, J. Wang, and T. Kubota, "From Fragments to Salient Closed Boundaries: An In-Depth Study," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. I, pp. 291-298, 2004.
[63] D.J. Williams and M. Shah, "A Fast Algorithm for Active Contours," *Proc. Int'l Conf. Computer Vision,* pp. 592-595, 1990.
[64] L. Williams and D. Jacobs, "Stochastic Completion Fields: A Neural Model of Illusory Contour Shape and Salience," *Neural Computation,* vol. 9, pp. 849-870, 1997.
[65] L. Williams and K.K. Thornber, "A Comparison Measures for Detecting Natural Shapes in Cluttered Background," *Int'l J. Computer Vision,* vol. 34, nos. 2/3, pp. 81-96, 2000.
[66] Z. Wu and R. Leahy, "An Optimal Graph Theoretic Approach to Data Clustering: Theory and Its Application to Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 11, pp. 1101-1113, Nov. 1993.
[67] C. Xu and J.L. Prince, "Snakes, Shapes, and Gradient Vector Flow," *IEEE Trans. Image Processing,* vol. 7, no. 3, pp. 359-369, 1998.
[68] S.C. Zhu and A. Yuille, "Region Competition: Unifying Snakes, Region Growing, and Bayes/MDL for Multi-Band Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 9, pp. 884-900, Sept. 1996.



Song Wang received the PhD degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign in 2002. From 1998 to 2002, he also worked as a research assistant in the Image Formation and Processing Group of the Beckman Institute, University of Illinois at Urbana-Champaign. Since August 2002, he has been an assistant professor in the Department of Computer Science and Engineering at the University of South Carolina. His research interests include computer vision, medical image processing, and machine learning. He is a member of the IEEE and the IEEE Computer Society.



Toshiro Kubota received the BS degree from Keio University, Japan, in 1988, and the MS and PhD degrees from the Georgia Institute of Technology, Atlanta, in 1989 and 1995, respectively. From 1996 until 2004, he was at the University of South Carolina, Columbia. He was an assistant professor from 2000 to 2004 and directed the Intelligent Systems Laboratory from 1999 to 2004. In 2004, he started a consulting business applying image processing and computer vision techniques to real-world problems. His research interests include computational vision, low-level vision, and neural networks. He is a member of the ACM and the IEEE.

Jeffrey Mark Siskind received the BA degree in computer science from the Technion, Israel Institute of Technology, in 1979, the SM degree in computer science from MIT in 1989, and the PhD degree in computer science from MIT in 1992. He did a postdoctoral fellowship at the University of Pennsylvania Institute for Research in Cognitive Science from 1992 to 1993. He was an assistant professor at the University of Toronto in the Department of Computer Science from 1993 to 1995, a senior lecturer in the Technion Department of Electrical Engineering in 1996, a visiting assistant professor at the University of Vermont Department of Computer Science and Electrical Engineering from 1996 to 1997, and a research scientist at NEC Research Institute, Inc. from 1997 to 2001. He joined the Purdue University School of Electrical and Computer Engineering in 2002, where he is currently an associate professor. His research interests include machine vision, artificial intelligence, cognitive science, computational linguistics, child language acquisition, and programming languages and compilers. He is a member of the IEEE and the IEEE Computer Society.

Jun Wang received the bachelor's degree from Shanghai JiaoTong University (SJTU) in July 1998 and the master's degree from the Department of Automation at Tsinghua University in 2003. He was a graduate student in the Department of Computer Science and Engineering, University of South Carolina, from August 2003 to May 2004. His research interest is computer vision.