Untangling Cycles for Contour Grouping

Qihui Zhu, Gang Song and Jianbo Shi
University of Pennsylvania
3330 Walnut Street, Philadelphia, PA 19104
{qihuizhu,songgang}@seas.upenn.edu, jshi@cis.upenn.edu

Abstract

We introduce a novel topological formulation for contour grouping. Our grouping criterion, called untangling cycles, exploits the inherent topological 1D structure of salient contours to extract them from the otherwise 2D image clutter. To define a measure for topological classification robust to clutter and broken edges, we use a graph formulation instead of the standard computational topology. The key insight is that a pronounced 1D contour should have a clear ordering of edgels, to which all graph edges adhere, and no long range entanglements persist. Finding the contour grouping by optimizing these topological criteria is challenging. We introduce a novel concept of circular embedding to encode this combinatorial task. Our solution leads to computing the dominant complex eigenvectors/eigenvalues of the random walk matrix of the contour grouping graph. We demonstrate major improvements over state-of-the-art approaches on challenging real images.

1. Introduction

Objects with salient contours tend to stand out from an image – they are nice to look at. Aside from their esthetics, salient contours help invoke our object shape memory, and speed up our visual perception [8]. In computer vision, good bottom-up salient contour detection can be extremely useful for object recognition. It provides global shape information, and simplifies object recognition by alignment.

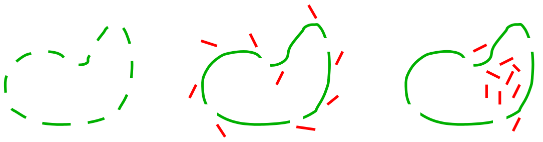

(a) Gaps (b) Distractions (c) 2D clutter

Figure 1. Challenges for contour grouping. (a) Contours have gaps to bridge. (b) Sporadic distractions mislead contour tracing. (c) 2D clutter confuses grouping when topology is not considered.

Contour grouping methods often start with edge detection, followed by linking edgels to optimize a saliency measure. Finding salient contours is easy when the image is clean, and contours are well separated. Gestalt factors of grouping, such as proximity and curvilinear continuity, define local likelihood of grouping two nearby edgel. Locally greedy search, such as shortest path, guided by the grouping measure can compute an optimal contour grouping efficiently. However, existing contour grouping algorithms are extremely unstable. They fail on natural images where image clutters are mixed with gaps on contours. Fundamentally it is difficult to distinguish gaps with clutter locally (see Fig. 1). A common mistake is finding too many false contours in a cluttered textured region.

We study contour grouping from a novel perspective of topology. We ask a harder question: does the image contain any 1D curve-like structure, and if so, can we show that it is *topologically 1D*? *By topologically 1D, we mean a set of edgels that have a well defined ordering, and the connections between them strictly follow that ordering*. By looking at the topology, we explicitly exclude 2D clutter or region-like structure from our contour search.

A key notion we introduce for this topological curve finding task is *entanglement*. Intuitively, a set of edges are entangled if we can not put them in an ordering without breaking many locally strongly linked edge pairs. We provide a graph formulation with a topological curve grouping score evaluating both separation from the background and disentanglement within the curve. Computationally, finding such curves requires *simultaneously* determining a subset of edgels and their ordering in the graph. We translate it to a

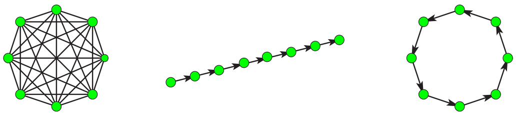
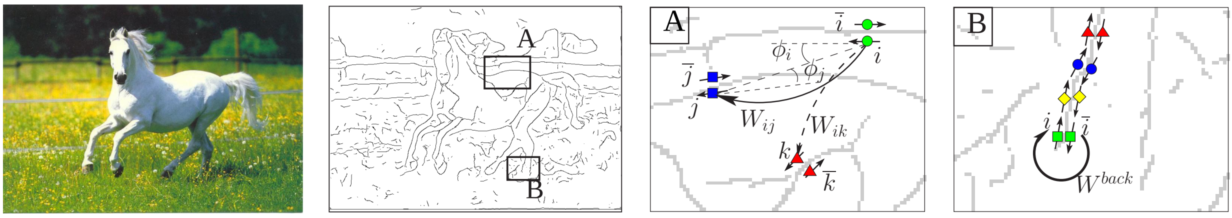

(a) Clique (b) Chain (c) Cycle

Figure 2. Distinction between 1D vs 2D topology. (a) 2D topology assumes a clique model. (b)(c) 1D topology assumes a chain or cycle model. A ring has 1D topology but 2D geometry.



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

Figure 3. Directed graph for contour grouping. (a) Horse image. (b) Edge map extracted from Pb [10]. (c) Zoom-in view of graph connection in window A. Each edge node is duplicated in two opposite orientations. Oriented nodes are connected according to elastic energy and their orientation consistency. Here $W_{ij} \gg W_{ik}$. Salient contours form 1D topological chain or cycle in this graph. (d) In window B, adding W^{back} to duplicated nodes $i, \bar{i}$ turns a topological chain into a cycle.

circular embedding in the *complex* domain, where disentanglement can be easily encoded and checked. We seek the desired circular embedding by computing complex eigenvectors of the graph weight matrix.

In Section 2, we review the related work. In Section 3, we define a directed graph for contour grouping using three untangling cycle criteria. The novel circular embedding is introduced in Section 4. We present the computational solution in Section 5, and experimental results in Section 6.

2. Background

The use of graph formulation for contour grouping has a long history, and we have drawn ideas from many of them [9, 17, 11, 2, 1, 15, 19]. The most related work is [9] by Mahamud *et al*. who use a similar directed graph for salient contour detection. However, they compute the top *real* eigenvectors of the *un-normalized* graph weight matrix. As we will show, the relevant topological information is encoded in the *complex* eigenvectors/eigenvalues of the *normalized* random walk matrix. This is an important distinction because the real eigenvectors contain no topological information of the graph. The works of [4, 7, 18] also seek salient contours. In contrast, we seek closed topological cycles which can include open contours, and are more robust to clutter. We are also motivated by the work of [6] which showed classical pairwise grouping is insufficient for contour detection. However, their solution of using shortest path is very sensitive to clutter. Our approach computes not only the parametrization, but also the segmentation simultaneously. Searching for subgraphs with the specified topology is a much harder combinatorial problem.

3. Untangling Cycle Formulation

In this section, we formulate the topological requirement of 1D structures as *Untangling Cycle Cut Score*. The cut score is defined on a *directed* contour grouping graph. We start by introducing the construction of the graph.

3.1. Directed graph and contour grouping

For contour grouping, we first threshold the output of edge detector (*e.g.* Pb [10]) to obtain a discrete set of edgels. We define a directed graph $\mathbf{G} = (\mathbf{V}, \mathbf{E}, \mathbf{W})$ as follows.

Graph nodes $\mathbf{V}$ correspond to all edgels. Since the edge orientation is ambiguous up to π, we duplicate every edgel into two copies i and $\bar{i}$ with opposite directions θ, $\theta + \pi$.

Graph edges $\mathbf{E}$ include all the pairs of edgels within some distance r_e: $\mathbf{E} = \{(i, j) : \|(x_i, y_i) - (x_j, y_j)\| \leq r_e\}$. Since every edgel is directed, we connect each edgel i only to the neighbors in the its direction.

Graph weights $\mathbf{W}$ measure *directed* collinearity using the elastic energy between neighboring edgels, which describes how much bending is needed to complete a curve between i and j:

$$W_{ij} = e^{-(1-\cos(|\phi_i|+|\phi_j|))/\sigma^2} \quad \text{if } i \longrightarrow j \tag{1}$$

Here $i \longrightarrow j$ means that j is in forward direction of i. $W_{ij} > 0$ implies that $W_{ji} = 0$. ϕ_i and ϕ_j denote the turning angles of i and j w.r.t. the line connecting them (see Fig. 3(c)).

In this graph, an ideal closed contour forms two directed cycles, one for each duplicated direction. Similarly, an ideal curve leads to two chains. On the other hand, random clutter produces fragmented clusters in the graph. Our task is to detect such topological differences, and extract 1D topological structures only.

To simplify the topological classification task and reduce the search to only cyclic structures, we transform two duplicated chains into a cycle by adding a small amount of connection W^{back} between the duplicated nodes i and $\bar{i}$. For open contours, W^{back} connects the termination points back to the opposite direction to create a cycle (see Fig. 3(d)).

Image clutter presents a challenge by creating leakages from a contour to the background. This is a classical problem in 2D segmentation as well. To prevent leakages, we borrow the concept from random walk interpretation of Normalized Cut [12]. We define the random walk matrix:

$$P = D^{-1}W \tag{2}$$

where D is diagonal with $D_{ii} = \sum_i W_{ij}$. This amounts to normalizing connection from each node by its total outward connections. Such normalization has two good side-effects: it boosts W^{back} connection at termination points of a chain, making the returning links there as strong as the interior of the contours; it also enhances connections for jagged salient contours which do not fit our curvilinear model.

3.2. Criteria for 1D topological grouping

Graph topology highlights the key difference between salient 1D curves and 2D clusters. The ideal model of a 2D cluster is a graph *clique*. In contrast, the ideal model for a 1D curve is a graph *cycle* or *chain* – it requires that the intra-group connections must be strictly ordered (see Fig. 2).

Ordering plays an important role in distinguishing 1D topological grouping. We define **entanglement** as *connection of nodes violating a given ordering*. Any 1D topological structure can be put into a specific ordering, such that each graph node connects to exactly one successor and is connected to exactly one predecessor (see Fig. 2 (b)(c)). In 2D topological structures, it is impossible to find a good ordering without entanglement (see Fig. 2 (a)). Entanglement is a tell-tail sign of 2D topological structure.

It is important to generalize the notion of strictly topological 1D to a coarser level. In real images, most image curves have missing edges, *i.e.* gaps. In order to bridge gaps without including clutter, each node needs to connect multiple neighboring nodes. These neighbors will contain *multiple* (k) nodes in the forward direction of ordering. As a result, its underlying graph topology is no longer strictly 1D. We need to relax the topologically 1D to a coarser level k – allowing up to k forward connections for each node (see Fig. 4). One can think that k defines a "thickness" factor on the 1D topology. As the number k increases, the topological structure gradually changes from 1D to 2D. When k equals the length of the contour, the group becomes 2D.

Given the directed graph $G = (V, E, W)$, we seek a group of vertices $S \subseteq V$ and an ordering on it such that they maximize the following score:

Untangling Cycle Cut Score (Max over $S, \mathcal{O}, k$)

$$C_u(S, \mathcal{O}, k) = \frac{1 - E_{cut}(S) - I_{cut}(S, \mathcal{O}, k)}{T(k)} \qquad (3)$$

S: Subset of graph nodes V, *i.e.* $S \subseteq V$.
$\mathcal{O}$: Cycle ordering on S.
k: Cycle thickness.

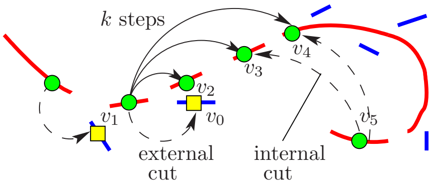


Figure 4. Criteria of untangling cycles for contour grouping. Even salient contour (red) might have gaps and distracting clutter (blue). To bridge gaps, we allow $k > 1$ steps of forward connections: v_1v_2, v_1v_3, v_1v_4. We define external cut as links from foreground to background nodes: v_1v_0. 1D contours should have perfect ordering. We define internal cut as out-of-order connections: v_5v_3, v_5v_4. We seek 1D topological structures minimizing all three.

External cut (E_{cut})

First, we need to measure how strongly S is separated from its surrounding background. We define a cut on the random walk matrix P that separates S from V:

$$E_{cut}(S) = \frac{1}{|S|} \sum_{i \in S, j \in (V-S)} P_{ij} \qquad (4)$$

We call it *external cut*, reflecting that we are cutting off external background nodes from vertex set V. This cost is closely related to $\frac{cut(S,V-S)}{Vol(S)}$, which is a "1-sided" Normalized Cut. This cut criterion is resistant to accidental leakages from background clutter to foreground. In contrast to the standard Normalized Cut cost [16], our contour grouping does not care about the cut from background clutter to foreground; hence it is "1-sided".

Internal cut (I_{cut})

A key distinguishing factor of a 1D structure is that it has a clear node ordering. It requires minimal entanglement between nodes far away in the ordering. We define the node ordering as a one-to-one mapping:

$$\mathcal{O} : S \mapsto S = \{1, 2, ..., |S|\} \qquad (5)$$

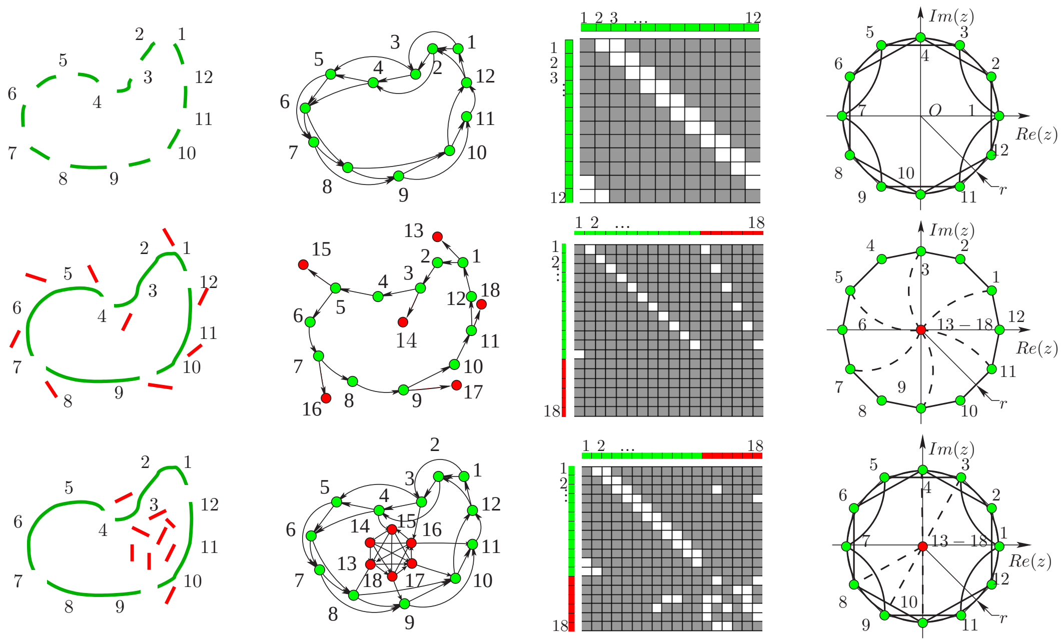
where $\mathcal{O}$ introduces a permutation of the nodes in S.

The "thickness" factor k measures *maximal step size* defining how much each link can violate the ordering $\mathcal{O}$. Edge (i, j) is *forward* if $0 < \mathcal{O}(j) - \mathcal{O}(i) \leq k$; *backward* if $-|S|/2 \leq \mathcal{O}(j) - \mathcal{O}(i) \leq 0$; *fast forward* otherwise. A perfect 1D cycle requires all the links to be forward (see Fig. 4) up to k steps ahead. No backward and fast forward links should exist. Backward and fast forward links are *entanglement* since they make the group tangle into a 2D structure. Untangling 1D cycles amounts to reducing such links.

Given a subset S, $\mathcal{O}$ and k, we define *internal cut* as the total entangled random walk transition probability:

$$I_{cut}(S, \mathcal{O}, k) = \frac{1}{|S|} \sum_{(\mathcal{O}(i) \geq \mathcal{O}(j)) \vee (\mathcal{O}(j) > \mathcal{O}(i)+k)} P_{ij} \qquad (6)$$

Here $\mathcal{O}(i) \geq \mathcal{O}(j)$ counts for backward links and $\mathcal{O}(j) > \mathcal{O}(i) + k$ for fast forward links. For simplicity, we assume that S is circular, i.e. the successor of $|S|$ wraps back to 1.



(a) Image (b) Graph (c) Weight matrix (d) Circular embedding

Figure 5. Circular embedding in complex space for finding 1D topological cycles. Three canonical cases are shown: a perfect cycle (green) shown in row 1, a cycle with sporadic distracting edges (red) in row 2, and with 2D clutter (red) in row 3. Column (a) Canonical image cases. Column (b) Directed graph constructed from edgels. Column (c) Random walk transition matrix P (white for strong links). Column (d) The optimal circular embedding. Distracting edges and 2D clutter are embedded into the origin.

Tube size (T)

The maximal step size k is a crucial factor involved with internal cut. In the ideal case of 1D cycle, we only allow connection with $k = 1$ step forward. As stated before, we need to measure 1D topology at a coarser scale to resist clutter and tolerate gaps. Therefore we want k to be as small as possible while keeping the internal and external cut low.

A physical analogy is very useful for understanding our task. Imagine we are asked to pull out string-like (1D) and ball-like (2D) interconnected particles through a tube. As long as the tube is narrow, we have to pull things out little by little, and we must untangle the strings to prevent jamming up in the tube. In contrast, it is impossible to pull out ball-like structures through the narrow tube.

We define tube size to measure how much entanglement is allowed in topological 1D structures as:

$$T(k) = k/|S| \tag{7}$$

Note that tube size $T(k)$ is independent of cycle length. Intuitively, the tube size describes how 'thick' the cycle is: the thinner the cycle is, the easier to pull it out through the tube. $T(k)$ reaches minimum of $1/|S|$ when $k = 1$.

Finally, we combine minimization of all the above three criteria into maximization of score (3).

One way to visualize the three criteria is to observe the structures of matrix P (Fig. 5(c)). Selecting S amounts to choosing a sub-block of P. External cut removes all the links outside the sub-block. After permutation $\mathcal{O}$, internal cut removes all the links outside the sub-band of P's diagonals. k is exactly the width of this sub-band. Therefore, Eq. (3) boils down to finding a sub-block of P, a permutation and a bandwidth k, such that the fewest links are left outside the sub-band. Note that standard graph cut algorithms (*e.g.* [16]) only consider external cut, but do not take internal cut and cycle thickness into account.

4. Circular embedding

Optimizing Eq. (3) essentially performs segmentation and parametrization on the graph *simultaneously*. We only cut out a subset of nodes with a good parametrization, *i.e.* ordering. This is a hard combinatorial task. Our strategy is to embed the graph into a circular space, such that the three criteria in (3) can be encoded and checked effectively.

Definition Circular embedding is a mapping from the vertex set V of the original graph to a circle plus the origin:

$$\mathcal{O}_{circ} : V \mapsto (r, \theta) : \mathcal{O}_{circ}(i) = x_i = (r_i, \theta_i) \tag{8}$$

Here r_i is the circle radius which can only take a positive fixed value r_0 or 0. θ_i is the angle associated with each node. Circular embedding can easily encode both the *cut* and the *ordering* of graph nodes. $S = \{v_i : r_i = r_0\}$ specifies the nodes being cut out, as in Eq. (4). Angle θ_i specifies the ordering. We simplify the embedding by restricting $\theta_i = 2\pi i/|S|$ (see Fig. 5), *i.e.* x_i is distributed uniformly on the circle. It is important to force x_i to spread out in the circular embedding. If x_i's all map to the same point, no order information can be obtained.

Average jumping angle In order to express tube size, we define the *average jumping angle* of the links as:

$$\Delta\theta = \overline{\theta_j - \theta_i} \tag{9}$$

Note that the average only counts (i, j) where there is an edge (i, j) in the original contour grouping graph. Since angle θ encodes the ordering, $\Delta\theta$ describes how far one node is expected to jump through the links.

We seek a circular embedding such that 1D topological structure is mapped to the circle while background is mapped to the origin. The optimal circular embedding maximizes the following score:

Circular Embedding Score (Max over $r, \theta, \Delta\theta$)

$$C_e(r, \theta, \Delta\theta) = \sum_{\substack{\theta_i < \theta_j \leq \theta_i + 2\Delta\theta \\ r_i > 0,\ r_j > 0}} P_{ij}/|S| \cdot \frac{1}{\Delta\theta} \tag{10}$$

r: Circle indicator with $r_i \in \{r_0, 0\}$.
θ: Angles on the circle specifying an ordering.
$\Delta\theta$: Average jumping angle.

The equivalence of the circular embedding and untangling cycles is established by the following lemma:

Lemma 1. *Circular Embedding Score (Eq. (10)) is equivalent to Untangling Cycle Cut Score (Eq. (3)) assuming that the angle difference $\theta_j - \theta_i$ is distributed uniformly.*

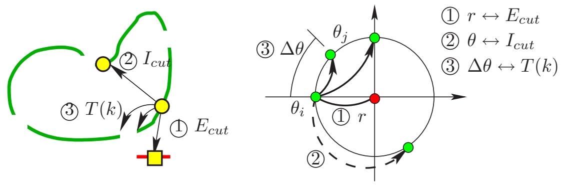
The derivation of Lemma 1 includes interpretation of the three criteria in the embedding space, shown as follows.

1) *External cut* requires that there are minimal links from the circle to the origin. Because $S = \{v_i : r_i = r_0\}$ specifies foreground nodes and $V - S = \{v_i : r_i = 0\}$ specifies background nodes, all links involved in E_{cut} are those from the circle to the origin (see Fig. 6).

2) *Internal cut* requires angles spanned by links on the circle to be small. Edges in the original graph is mapped to chords on the circle. The angle spanned by the chord is $\theta_i - \theta_j = \frac{2\pi}{|S|}(i - j)$. Therefore, links involved in I_{cut} are those with either negative angle (backward links) or large positive angle (fast forward links).

3) *Tube size* is given by the average jumping angle $\Delta\theta$. Recall that k gives the upper bound determining which links are forward. In circular embedding, it means the angle difference of forward links does not exceed $k \cdot \frac{2\pi}{|S|}$. If the jumping angles are distributed uniformly within $k \cdot \frac{2\pi}{|S|}$, then

$$\Delta\theta = (2\pi/|S|) \cdot (k/2) = \pi \cdot k/S = \pi \cdot T(k) \tag{11}$$


③ $T(k)$
② I_{cut}
① E_{cut}
③ $\Delta\theta$
θ_j
θ_i
① r
②
① $r \leftrightarrow E_{cut}$
② $\theta \leftrightarrow I_{cut}$
③ $\Delta\theta \leftrightarrow T(k)$

(a) Untangling cycle criteria (b) Circular embedding

Figure 6. Interpretation of the three untangling cycle criteria E_{cut}, I_{cut} and $T(k)$ in circular embedding.

This seems to be a crude approximation to $\Delta\theta$ at the first glance. However, in contour grouping, it is reasonable because we set up the graph with multiple forward links for each node and the weights for these links tend to be similar.

Now we can rewrite the score function (3) in circular embedding, expressed by (r, θ) and the average jumping angle $\Delta\theta$. Because P_{ij} is row normalized (Eq. (2)), $\sum_i P_{ij}/|S| = 1$. Since non-forward links are either included in $E_{cut}(S)$ or $I_{cut}(S, \mathcal{O}, k)$, $1 - E_{cut}(S) - I_{cut}(S, \mathcal{O}, k)$ is essentially counting how many forward links are left. The numerator of Eq. (3) can be expressed in terms of r, θ and $\Delta\theta$:

$$1 - E_{cut}(r) - I_{cut}(r, \theta, \Delta\theta) = \sum_{\substack{\theta_i < \theta_j \leq \theta_i + 2\Delta\theta \\ r_i > 0, r_j > 0}} \frac{P_{ij}}{|S|} \tag{12}$$

The forward links are chords with spanning angles no more than $2\Delta\theta$. Combining Eq. (11), (12), maximizing Eq. (3) reduces to the maximizing Eq. (10) in circular embedding.

5. Complex eigenvectors: a continuous relaxation

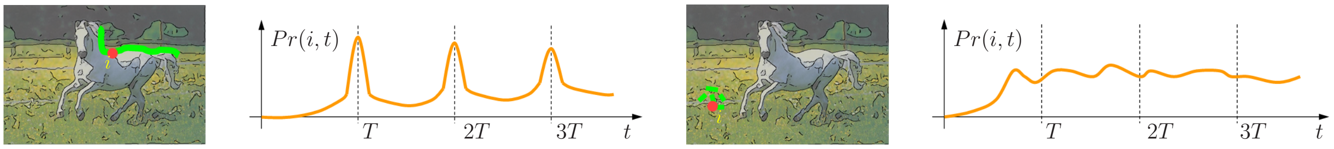
Now we are ready to derive a computational solution. We generalize the discrete circular embedding (8) by mapping the graph into the continuous complex plane. The optimal continuous circular embedding turns out to be the complex eigenvectors of the random walk matrix.

First we relax both r and θ in Eq. (10) to continuous values. Our goal is to find the optimal mapping $\mathcal{O}_{cmpl} : V \mapsto \mathbb{C}$, $\mathcal{O}_{cmpl}(v_j) = x_j = r_j e^{i\theta_j}$, which approximates the optimal r and θ in Eq. (10). Here $r_j = \|x_j\|$ and θ_j are magnitude and phase angle of the complex number x_j.

In the desired embedding with a fixed $\Delta\theta$, the term

$$\sum_{i,j} P_{ij} \cos(\theta_j - \theta_i - \Delta\theta) = \sum_{i,j} P_{ij} \mathrm{Re}(x_i^* x_j \cdot e^{-i\Delta\theta})/r_0^2$$

is a good approximation of the sum of forward links (numerator in Eq. (12)). When the angle difference $\theta_j - \theta_i$ equals average jumping angle $\Delta\theta$, the weight reaches the maximum of 1. When $\theta_j - \theta_i$ deviates from $\Delta\theta$, the weight



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

Figure 7. Persistent cycles. (a) 1D contours correspond to good cycles. (b) Returning probability $Pr(i,t)$ on 1D contours has period peaks since random walk on it tends to return in a fixed time. (c) 2D clutter corresponds to bad cycles. (d) Returning probability $Pr(i,t)$ of random walk on 2D clutter is flat.

gradually dies off. Then the score function (12) becomes:

$$\frac{\sum_{ij} P_{ij}\mathrm{Re}(x_i^* x_j \cdot e^{-i\Delta\theta}) \cdot t_0}{\sum_i |x_i|^2} \tag{13}$$

where the denominator is exactly $|S|$ in the discrete case. Here $t_0 = 1/\Delta\theta$ relates to x as well.

Expressed in a matrix form, Eq. (13) becomes

$$\max_{\Delta\theta\in\mathbb{R}, x\in\mathbb{C}^n} \frac{\mathrm{Re}(x^H Px \cdot t_0 e^{-i\Delta\theta})}{x^H x} \tag{14}$$

Solving Eq. (14) is not an easy task. Moreover, we are not only interested in the best solution of Eq. (14), but all the locally optimal solutions. These local optima will give all the 1D structures in the graph. We find a relaxation by setting $u = x$, $v = u \cdot e^{-i\Delta\theta}$. We set $c = t_0 e^{-i\Delta\theta}$ to be a constant. Eq. (14) can be rewritten as maximizing $\mathrm{Re}((u^H Pv \cdot c)/(u^H v))$ with $u, v \in \mathbb{C}^n$. Furthermore, it is equivalent to the following optimization problem:

$$\max_{u,v\in\mathbb{C}^n} \quad \mathrm{Re}(u^H Pv) \qquad s.t.\ u^H v = c \tag{15}$$

This problem leads exactly to P's complex eigenvectors.

Theorem 1. *All the critical points (local maxima) (u_{max}, v_{max}) of the optimization problem (15) are given by the left and right eigenvectors of P respectively,* i.e., *$Pv_{max} = \lambda v_{max}$ and $P^T u^*_{max} = \lambda u^*_{max}$. Furthermore, the corresponding maximal value is $\max_\lambda(\mathrm{Re}(\lambda \cdot c))$ where λ is one eigenvalue of P.*

Proof. Please see Appendix. □

The complex eigenvectors gives us the ordering of 1D cycles, encoded in the phase angle of u. The average jumping angle $\Delta\theta$ is given by the phase angle of λ because $Re(\lambda \cdot c)$ reaches its maximum when the phase angles of λ^* and $c = t_0 e^{-i\Delta\theta}$ are most similar.

6. Random walk interpretation

Random walk provides an alternative view to see why complex eigenvector is useful for untangling cycles. Random walk has been shown to be effective analyzing region segmentation [12]. Unlike traditional random walk analysis, we are interested in periodicity of the states rather than the convergence behavior. Periodicity is a good indication that there exist persistent cycles in the graph.

6.1. Random walk

Following traditional random walk analysis, transition matrix $P = D^{-1}W$ (Eq. (2)) encodes the probability of switching states. In other words, P_{ij} is the probability that a particle starts from node j and randomly walk to node i in one step. Note that P is asymmetric because the random walk is directional.

According to our graph setup (see Section 3), both open and closed image contours become directed cycles in the contour graph. Finding image contours amounts to searching cycles in this directed graph. However, there are numerous graph cycles and not all cycles correspond to 1D image contours. Now the key question is: *What is the appropriate saliency measure for good cycles (1D contour) and bad cycles (2D clutter)?*

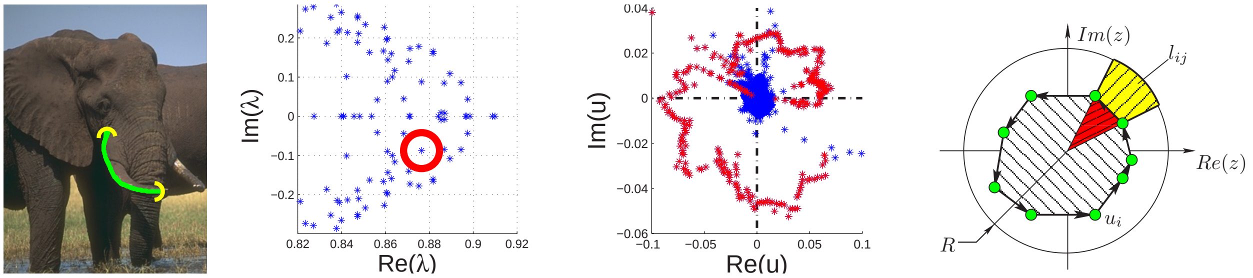
We first notice an obvious necessary condition. If the random walk starting at a node comes back to itself with high probability, then it is likely that there is a cycle passing through it. We denote the returning probability by

$$Pr(i,t) = \sum_\ell Pr(i,t \mid |\ell| = t) \tag{16}$$

Here ℓ is a random walk cycle with length t passing i. However, this condition alone is not enough to identify 1D cycles. 2D clutter can also introduce Consider the case where there are many distracting branches of the main cycle. In this case, paths through the branches will still return to the same node but with different path lengths. Therefore, it is not sufficient to require the paths to return only, but return in the *same period*.

6.2. Persistent cycles

We have found that 1D cycles have a special pattern of returning probability $Pr(i,t)$ (see Fig. 7). From analysis of Section 4, one step of random walk on 1D cycles tends to stay in the cycles (external cut to be small), and move a fixed amount forward in the cyclic ordering (internal cut



(a) Image (b) Eigenvalues (c) One eigenvector (d) Maximum circular cover

Figure 8. Illustration of computational solution. (a) An elephant with a detected contour grouping (green) and endpoints (yellow) on its tusk. (b) The top n_c eigenvalues sorted by their real components. Their phase angles relate to the 1D thickness of cycles. We look for complex ones with large magnitudes but small phase angles indicating the existence of thin 1D structures. (c) The complex eigenvector corresponding to the selected eigenvalue in (b) (red circle) is plotted. The detected tusk contour is embedded into a geometric cycle plotted in red. We find discretization in this embedding space by seeking the maximum circular cover shown in (d). See text for more details.

to be small). If one starts a random walk from a node in 1D cycles, it is very likely to return at multiple times of a certain period. We call these cycles *persistent* cycles. Our task is to separate persistent cycles from other random walk cycles.

To quantify the above observation, we introduce the following 'peakness' measure of the random walk probability pattern:

$$R(i,T) = \frac{\sum_{k=1}^{\infty} Pr(i,kT)}{\sum_{k=0}^{\infty} Pr(i,k)} \tag{17}$$

Here we compute the probability that the random walk returns at steps of multiples of T. $R(i,T)$ being high indicates there are 1D cycles passing through node i.

The key observation is that $R(i,T)$ closely relates to complex eigenvalues of P, instead of real eigenvalues.

Theorem 2. *(Peakness of Random Walk Cycles) $R(i,T)$ can be computed by the eigenvalues of transition matrix P:*

$$R(i,T) = \frac{\sum_j \mathrm{Re}(\frac{\lambda_j^T}{1-\lambda_j^T} \cdot U_{ij}V_{ij})}{\sum_j \mathrm{Re}(\frac{1}{1-\lambda_j} \cdot U_{ij}V_{ij})} \tag{18}$$

Proof. See Appendix. □

Theorem 2 shows that $R(i,T)$ is the "average" of $f(\lambda_j,T) = \mathrm{Re}(\frac{\lambda_j^T}{1-\lambda_j^T} \cdot U_{ij}V_{ij})/\mathrm{Re}(\frac{1}{1-\lambda_j} \cdot U_{ij}V_{ij})$. For real λ_j, $f(\lambda_j,T) \leq 1/T$. For complex λ_j, $f(\lambda_j,T)$ can be large. For example, when $\lambda_j = s \cdot e^{i2\pi/T}$, $s \to 1$, $U_{ij} = V_{ij} = a \in \mathbb{R}$, $f(\lambda_j,T) \to \infty$. Hence it is the complex eigenvalue with proper phase angle and magnitude that leads to repeated peaks. Complex eigenvalues and eigenvectors of P indeed carry important information on persistent 1D cycles.

Because the random walk will eventually converge to the steady state, $Pr(i,T)$ converges to a constant. This means that $R(i,T) \to 1/T$ no matter what the graph structure is. We can alleviate this technical issue by multiplying a decay factor η. Namely, we use $\eta^k Pr(i,k)$ to replace $Pr(i,k)$. Responses with longer time are weighted lower because the peaks become more and more blurred. This amounts to replace P by ηP and all the above analysis.

7. Computational solution

The complex eigenvector is an approximation of the optimal circular embedding and will not produce exact 1D cycles. Therefore, we still need to search for 1D cycles in this space. We introduce a discretization method and give the overall untangling cycle procedure in this section.

7.1. Discretization

For each of the top complex eigenvectors, we seek discrete topological cycle(s) separated from background. First, we can read off the tube size directly from the phase angle of its corresponding eigenvalue. This determines the "thickness" k of our cycle. Since we prefer thin 1D cycles, we will only examine eigenvectors with small phase angles.

Once we know a 1D cycle exists, we search for it in its complex eigenvector whose components are $u_1, ...u_{2n}$. The topological graph cycles are mapped to the geometric cycles in this embedding space. The larger the cycle is geometrically, the better the 1D graph cycle is topologically. Therefore, we should search for a sequence $s_1, s_2, ..., s_h, s_{h+1} = s_1$ such that $|u_{s_1}|, ..., |u_{s_h}|$ are large and $\theta(u_{s_1}), ..., \theta(u_{s_h})$ are in an increasing order. This can be tackled by finding the sequence enclosing the largest area in the complex plane:

$$\max_{s_1,...,s_h} \sum_{j=1}^{h} A(u_{s_j}, u_{s_{j+1}}) \tag{19}$$

Here $A(u_{s_j}, u_{s_{j+1}}) = \frac{1}{2}\mathrm{Im}(u_{s_j}^* \cdot u_{s_{j+1}})$ is the signed area of the triangle spanned by $u_{s_j}, u_{s_{j+1}}$ and 0.

To simplify the search, we can pack u_i into bins $B_1, ..., B_m$ according to their phase angles. Suppose there is an edge (i, j) in the original graph. If u_i is in a properly ordered cycle, the phase angle difference $\theta(u_j) - \theta(u_i)$ will, on average, equal to $\Delta\theta$. Hence, we can safely assume that all its neighbors u_j are at most one bin apart from u_i if the bin size is chosen properly (*e.g.* $2\Delta\theta$). Furthermore, we group nodes within the same bin by their spatial connectivity. This greatly reduces the computational cost.

The maximal enclosed area problem can be solved by the shortest path algorithm. Notice that the sequence $u_{s_1}, ..., u_{s_h}, u_{s_{h+1}} = u_{s_1}$ produces a closed loop around the origin. Suppose it only wraps around the origin once. For each pair of i, j in neighboring bins, set $\ell_{ij} = \frac{1}{2}[\theta(u_j) - \theta(u_i)] \cdot R^2 - A(u_i, u_j)$. R is chosen sufficiently large to guarantee $\ell_{ij} > 0$ for all i,j. Then Eq. (19) can be reduced to

$$\pi R^2 - \min_{s_1,\ldots,s_{h+1}} \sum_{j=1}^{h} \ell_{s_j s_{j+1}} \tag{20}$$

This shortest cycle problem can be broken into two parts: the first shortest path from s_1 in bin B_1 to a node s_u in bin B_2, and the second one from s_u back to s_1. Hence, $\min_{s_1,\ldots,s_{h+1}} \sum_{j=1}^{h} \ell_{s_j s_{j+1}}$ in Eq. (20) becomes

$$\min_{\substack{s_1 \in B_1, s_u \in B_2 \\ s_1,\ldots,s_{h+1}}} [\sum_{j=1}^{u-1} \ell_{s_j s_{j+1}} + \sum_{j=u}^{h} \ell_{s_j s_{j+1}}] \tag{21}$$

where each summation itself is a shortest path.

7.2. Algorithm

In summary, our untangled cycle algorithm has 3 steps:

Algorithm 1 Untangling cycles

```
Graph setup Construct the directed graph $G$ and compute transition matrix $P$ by Eq. (1) and (2).
Complex embedding Compute the first $n_c$ complex eigenvectors of $P$. Each complex eigenvector produces a complex circular embedding $u_1, u_2, ...u_{2n} \in \mathbb{C}$.
Cycle tracing For $u_1, u_2, ...u_{2n}$, use shortest path to find a cycle $S \subseteq \{1, ..., 2n\}$ minimizing (Eq. (20)).
```

8. Experiments

We test our *untangling cycle* algorithm on a variety of challenging real images, including Berkeley Segmentation Dataset [10] (see Fig. 13), Weizmann horse database [3] (see Fig. 11) and Berkeley baseball player dataset [13] (see Fig. 12). Our untangling cycle algorithm is capable of finding one-dimensional contours even when many of the images have significant clutter (see Fig. 10). We output contours that are both open or closed, straight or bent. These experiments are performed using the same set of parameters and we show all the detected contours without any postprocessing. Extensive tests show that our algorithm is effective in discovering one-dimensional topological structures in real images.

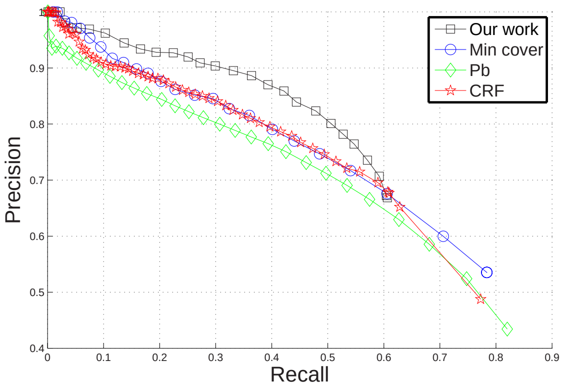


Figure 9. Precision recall curve on the Berkeley benchmark, with comparison to Pb [10], CRF [14] and min cover [5]. We use probability boundary [10] with low threshold to produce graph nodes, and seek untangling 1D topological cycles for contour grouping. The same set of parameters are used to generate all the results.

Our results are significantly better than those of state-of-the-art, particularly on cluttered images. To quantify our performance, we compare our precision-recall curve on the Berkeley benchmark set with two top algorithms: CRF [14] and min cover [5] on this test. Our result is well above these approaches by about 7% in the medium to high precision part. Visually our results produce much cleaner contours. Many of the false positives are shading edges, which are not labelled by humans. However, once they are grouped, they could be easy to prune in later recognition process. These are the advantages not reflected by the metric in the Berkeley benchmark, which counts matched pixels independently.

9. Conclusion

To our knowledge, this is the first major attack on contour grouping using topological formulation. Our grouping criterion, untangling cycles, exploits the inherent 1D topological structure of salient contours to extract them from the otherwise 2D image clutter. We made this precise by defining a directed graph linking local edgels. We encode the untangling cycle criterion by circular embedding. Computationally, this reduces to finding the top complex eigenvectors of the random walk matrix. We demonstrate significant improvements over state-of-the-art approaches on challenging real images.

Acknowledgement We thank Praveen Srinivasan and Elena Bernardis for discussions on shaping the main concept. This work is partially supported by NSF grants: NSF-

IIS-04-47953(CAREER), NSF-IIS-03-33036(IDLP) and NSF-IIS004-31070.

References

[1] T. D. Alter and R. Basri. Extracting salient curves from images: An analysis of the saliency network. *CVPR*, 1996.

[2] A. Amir and M. Lindenbaum. Grouping-based nonadditive verification. *PAMI*, 20(2):186–192, 1998.

[3] E. Borenstein and S. Ullman. Class-specific, top-down segmentation. In *ECCV*, 2002.

[4] J. H. Elder and S. W. Zucker. Computing contour closure. *Lecture Notes in Computer Science*, 1064, 1996.

[5] P. F. Felzenszwalb and D. McAllester. A min-cover approach for finding salient curves. In *WPOCV*, page 185, 2006.

[6] B. Fischer and J. M. Buhmann. Path-based clustering for grouping of smooth curves and texture segmentation. *PAMI*, 25(4):513–518, 2003.

[7] D. W. Jacobs. Robust and efficient detection of salient convex groups. *PAMI*, 18(1):23–37, 1996.

[8] K. Koffka. *Principles of Gestalt Psychology*. 1935.

[9] S. Mahamud, L. Williams, K. Thornber, and K. Xu. Segmentation of multiple salient closed contours from real images, 2003.

[10] D. R. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, pages 416–425, 2001.

[11] G. G. Medioni and G. Guy. Inferring global perceptual contours from local features. In *IUW*, 1993.

[12] M. Meila and J. Shi. Learning segmentation by random walks. In *NIPS*, pages 873–879. MIT Press, 2000.

[13] G. Mori, X. Ren, A. A. Efros, and J. Malik. Recovering human body configurations: Combining segmentation and recognition. In *CVPR (2)*, pages 326–333, 2004.

[14] X. Ren, C. Fowlkes, and J. Malik. Scale-invariant contour completion using conditional random fields. In *ICCV*, pages 1214–1221. IEEE Computer Society, 2005.

[15] S. Sarkar and P. Soundararajan. Supervised learning of large perceptual organization: Graph spectral partitioning and learning automata. *PAMI*, 22(5):504–525, 2000.

[16] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 22(8):888–905, Aug. 2000.

[17] S. Ullman and A. Shashua. Structural saliency: The detection of globally salient structures using a locally connected network. In *MIT AI Memo*, 1988.

[18] S. Wang, T. Kubota, J. M. Siskind, and J. Wang. Salient closed boundary extraction with ratio contour. *PAMI*, 2005.

[19] S. X. Yu and J. Shi. Multiclass spectral clustering. In *ICCV*, pages 313–319. IEEE Computer Society, 2003.

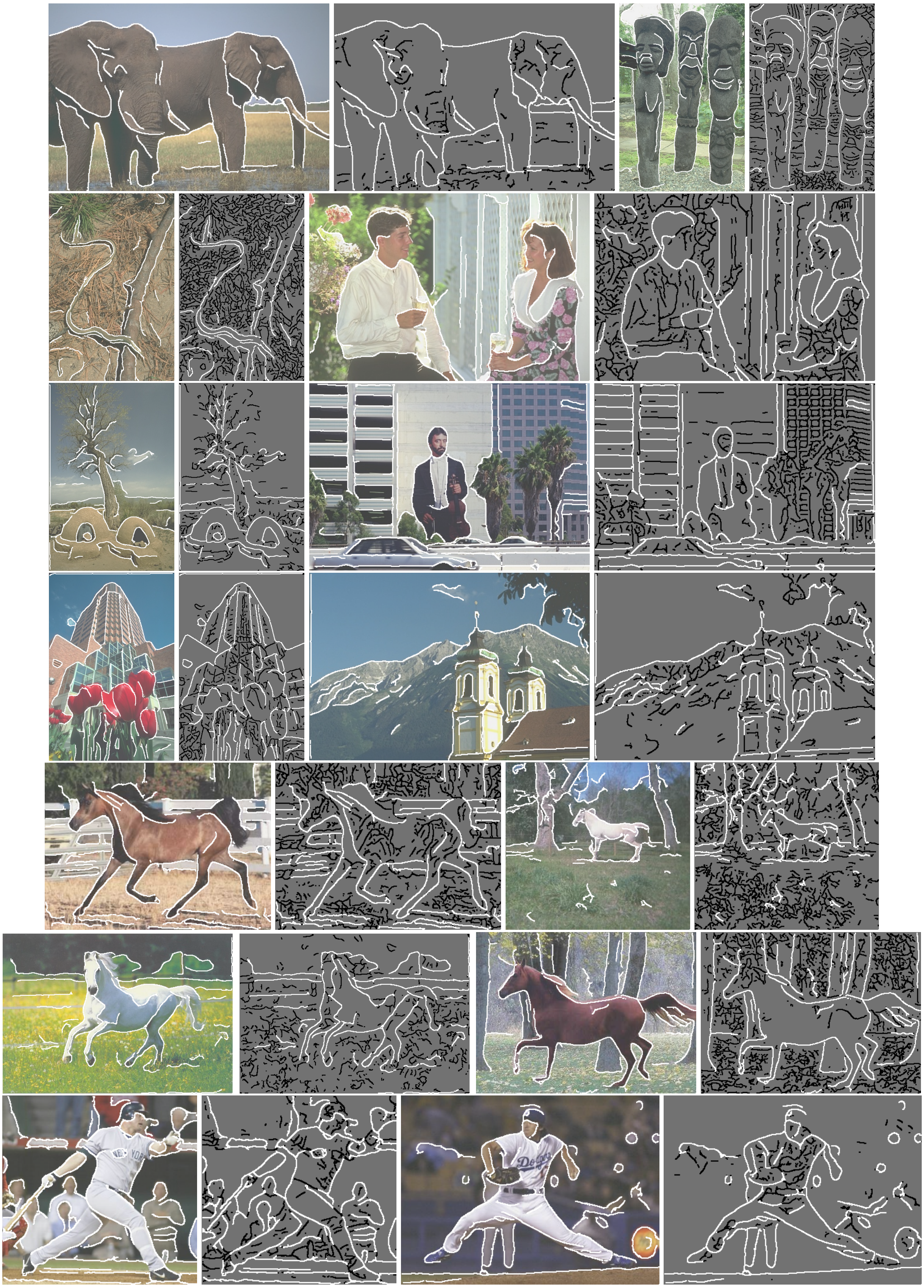


Figure 10. Contour grouping result on real images. All detected binary edges are shown (right). Our method prunes clutter edges (dark), and groups salient contours (bright). We use no edge magnitude information for grouping, and can detect faint but salient contours under significant clutter. We focus on graph topology, and detect contours that are either open or closed, straight or bended.

Appendix

A. Proof of Theorem 1

Theorem 1 *All the critical points (local maxima) (u_{max}, v_{max}) of the following optimization problem is given by the left and right eigenvectors of P respectively,* i.e., *$Pv_{max} = \lambda v_{max}$ and $P^T u^*_{max} = \lambda u^*_{max}$.*

$$\max_{u,v\in\mathbb{C}^n} \ \mathrm{Re}(u^H P v) \tag{22}$$

$$s.t. \ \ u^H v = c \tag{23}$$

Furthermore, the corresponding maximal value is $\max_\lambda(\mathrm{Re}(\lambda \cdot c))$ *where λ is one eigenvalue of P.*

Proof. Let $u = u_r + iu_c$ and $v = v_r + iv_c$. By splitting into real and imaginary parts, the original problem can be written as

$$\max_{u_r,u_c,v_r,v_c} \ u_r^T P v_r + u_c^T P v_c \tag{24}$$

$$s.t. \ \ u_r^T v_r + u_c^T v_c = \mathrm{Re}(c) \tag{25}$$

$$u_r^T v_c - u_c^T v_r = \mathrm{Im}(c) \tag{26}$$

$$u_r, u_c, v_r, v_c \in \mathbb{R}^n \tag{27}$$

Hence, the Lagrangian has the following form with μ and ν as the multipliers:

$$\begin{aligned} L =& u_r^T P v_r + u_c^T P v_c + \mu[u_r^T v_r + u_c^T v_c - \mathrm{Re}(c)] \\ &+ \nu[u_r^T v_c - u_c^T v_r - \mathrm{Im}(c)] \end{aligned}$$

By taking derivatives of the Lagrangian, we have

$$\frac{\partial L}{\partial u_r} = Pv_r + \mu v_r + \nu v_c \tag{28}$$

$$\frac{\partial L}{\partial u_c} = Pv_c + \mu v_c - \nu v_r \tag{29}$$

$$\frac{\partial L}{\partial v_r} = P^T u_r + \mu u_r - \nu u_c \tag{30}$$

$$\frac{\partial L}{\partial v_c} = P^T u_c + \mu u_c + \nu u_r \tag{31}$$

Setting the above derivatives to 0 gives all the local maxima of the original problem (22). After combining the equations, we obtain

$$P(v_r + iv_c) = (-\mu + i\nu)(v_r + iv_c) \tag{32}$$

$$P^T(u_r + iu_c) = (-\mu - i\nu)(u_r + iu_c) \tag{33}$$

Since P is real, Eq. (33) is equivalent to $P^T(u_r - iu_c) = (-\mu + i\nu)(u_r - iu_c)$. This means that all the critical points $(u^*, v) = (u_r - iu_c, v_r + iv_c)$ are left and right eigenvectors of P. Let $\lambda = -\mu + i\nu$. By substituting $Pv = \lambda v$ into the objective function, we conclude that the optimal value is $\max_\lambda(\mathrm{Re}(\lambda \cdot c))$.

□

B. Proof of Theorem 2

First we prove the following lemma:

Lemma 1 *$Pr(i, m)$ can be expressed in terms of eigenvalues and eigenvectors of transition matrix P* [1]*:*

$$Pr(i,m) = \sum_{\lambda_j\ real} \lambda_j^m U_{ij} V_{ij} + \sum_{\lambda_j\ complex} \mathrm{Re}(\lambda_j^m U_{ij} V_{ij}) \tag{34}$$

where λ_j is the j^{th} eigenvalues of P and U_{ij} is the i^{th} entry of the j^{th} right eigenvector and V_{ij} is the i^{th} entry of the j^{th} left eigenvector.

Proof. By simple induction one can prove that

$$Pr(i,m) = (P^m)_{ii} \tag{35}$$

Here $(P^m)_{ij}$ represents the entry at row i and column j.

Consider the eigenvalue decomposition of P

$$P = U\Sigma U^{-1} \tag{36}$$

Here $\Sigma = diag(\lambda_1, ..., \lambda_n)$ and U is a nonsingular complex matrix whose columns are corresponding eigenvectors $u_1, ..., u_n$. Since eigenvectors are not necessarily orthogonal, U^{-1} is not equal to U^H in general. However, rows of U^{-1} are left eigenvectors of P, *i.e.* $(U^{-1})^T = V$. The power of P can be easily computed by

$$P^m = U\Sigma^m U^{-1} \tag{37}$$

We can write $(P^m)_{ii}$ as

$$(P^m)_{ii} = (U\Sigma^m U^{-1})_{ii} \tag{38}$$

$$= \sum_j U_{ij} \cdot \lambda_j^m \cdot V_{ij} \tag{39}$$

$$= \sum_{\lambda_j\ real} \lambda_j^m U_{ij} V_{ij} + \sum_{\lambda_j\ complex} \mathrm{Re}(\lambda_j^m U_{ij} V_{ij}) \tag{40}$$

Eq (40) comes from the fact that U_{ij} and V_{ij} are all real if λ_j is real and all complex eigenvalues appear in pairs. □

With *Lemma 1*, we can easily prove *Theorem 2*.

Theorem 2 *(Peakness of Random Walk Cycles) $R(i, T)$ can be computed by the eigenvalues of transition matrix P:*

$$R(i,T) = \frac{\sum_j \mathrm{Re}(\frac{\lambda_j^T}{1-\lambda_j^T} \cdot U_{ij} V_{ij})}{\sum_j \mathrm{Re}(\frac{1}{1-\lambda_j} \cdot U_{ij} V_{ij})} \tag{41}$$

[1] To simplify the analysis, we assume that P is diagonalizable in $\mathbb{C}^{n\times n}$ and achieve this by perturbing P. For any $\epsilon \in \mathbb{R}$, there exists diagonalizable Q such that $||P - Q|| < \epsilon$.

Proof. From *Lemma 1*, it is straight forward to get

$$\sum_{k=1}^{\infty} Pr(i,kT) = \sum_j \mathrm{Re}(\lambda_j^T/(1-\lambda_j^T) \cdot U_{ij}V_{ij}) \tag{42}$$

$$\sum_{k=1}^{\infty} Pr(i,k) = \sum_j \mathrm{Re}(1/(1-\lambda_j) \cdot U_{ij}V_{ij}) \tag{43}$$

Finally we have

$$R(i,T) = \frac{\sum_j \mathrm{Re}(\frac{\lambda_j^T}{1-\lambda_j^T} \cdot U_{ij}V_{ij})}{\sum_j \mathrm{Re}(\frac{1}{1-\lambda_j} \cdot U_{ij}V_{ij})} \tag{44}$$

□

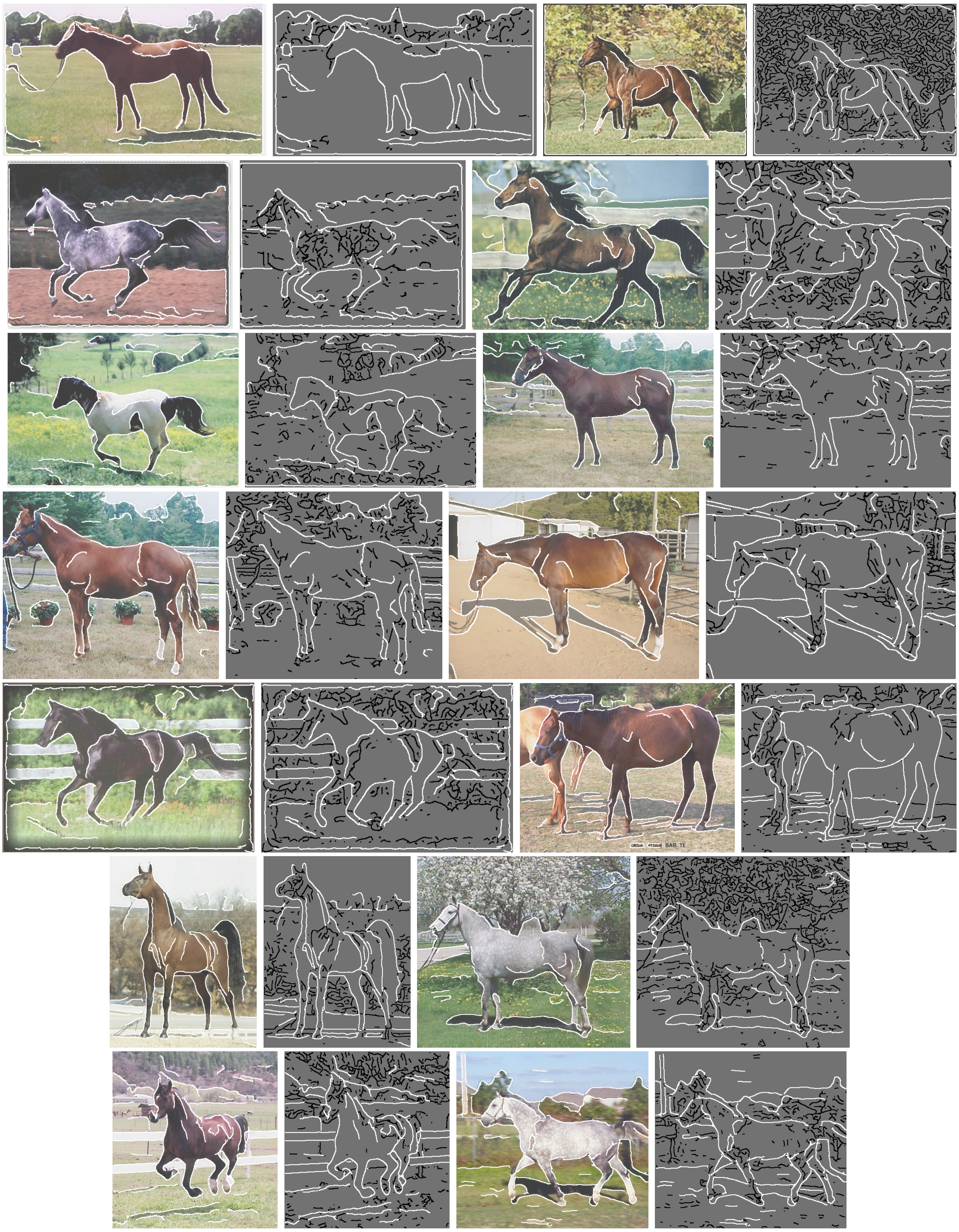


Figure 11. Contour grouping result on Weizmann horse database. All detected binary edges are shown (right). Our method prune clutter edges (dark), and groups salient contours (bright). We use no edge magnitude information for grouping, and can detect faint but salient contours under significant clutter. We focus on graph topology, and detect contours that are both open or closed, straight or bent.

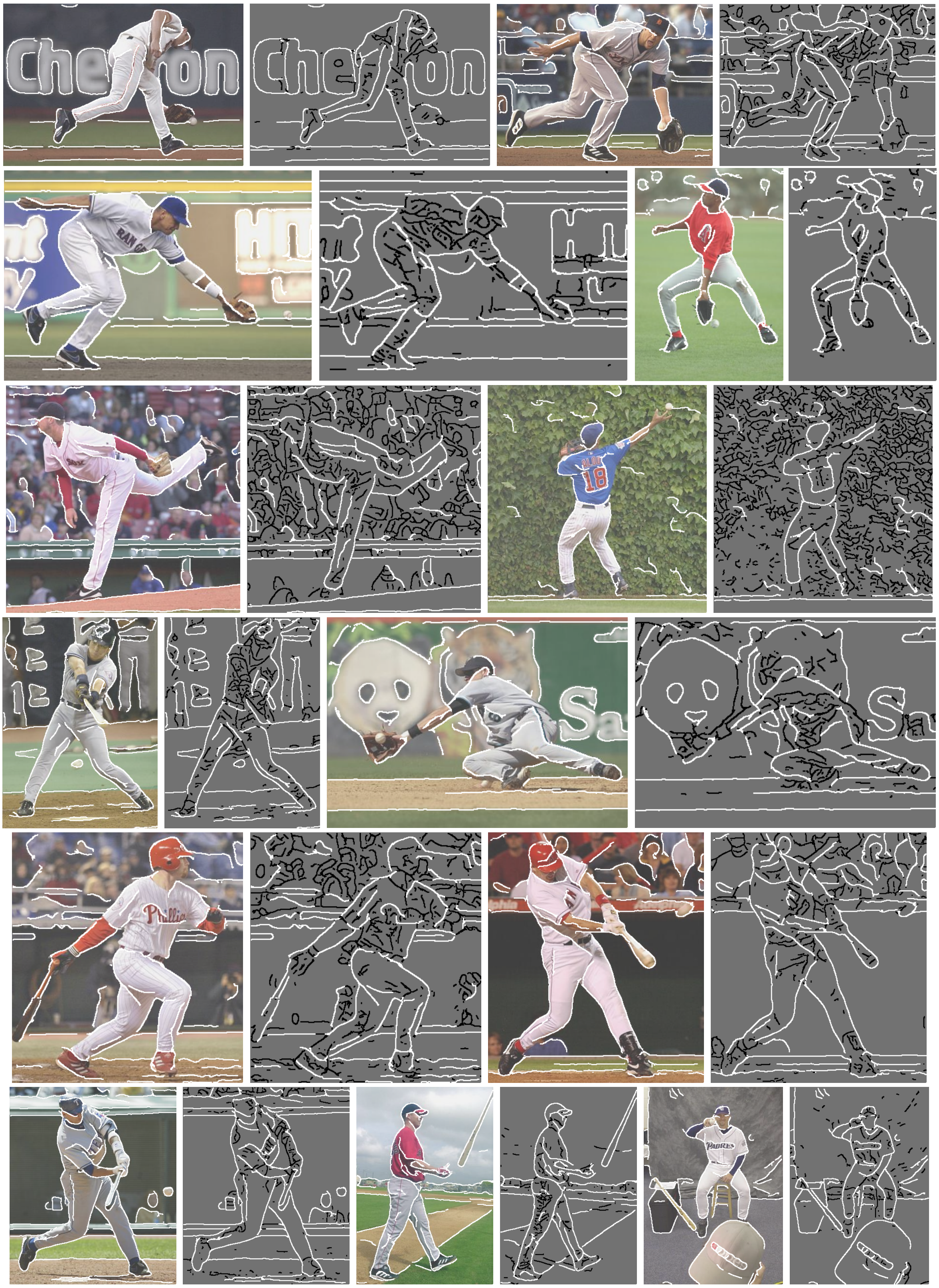


Figure 12. Contour grouping result on Berkeley baseball player dataset. See the caption of Figure 11 for description.

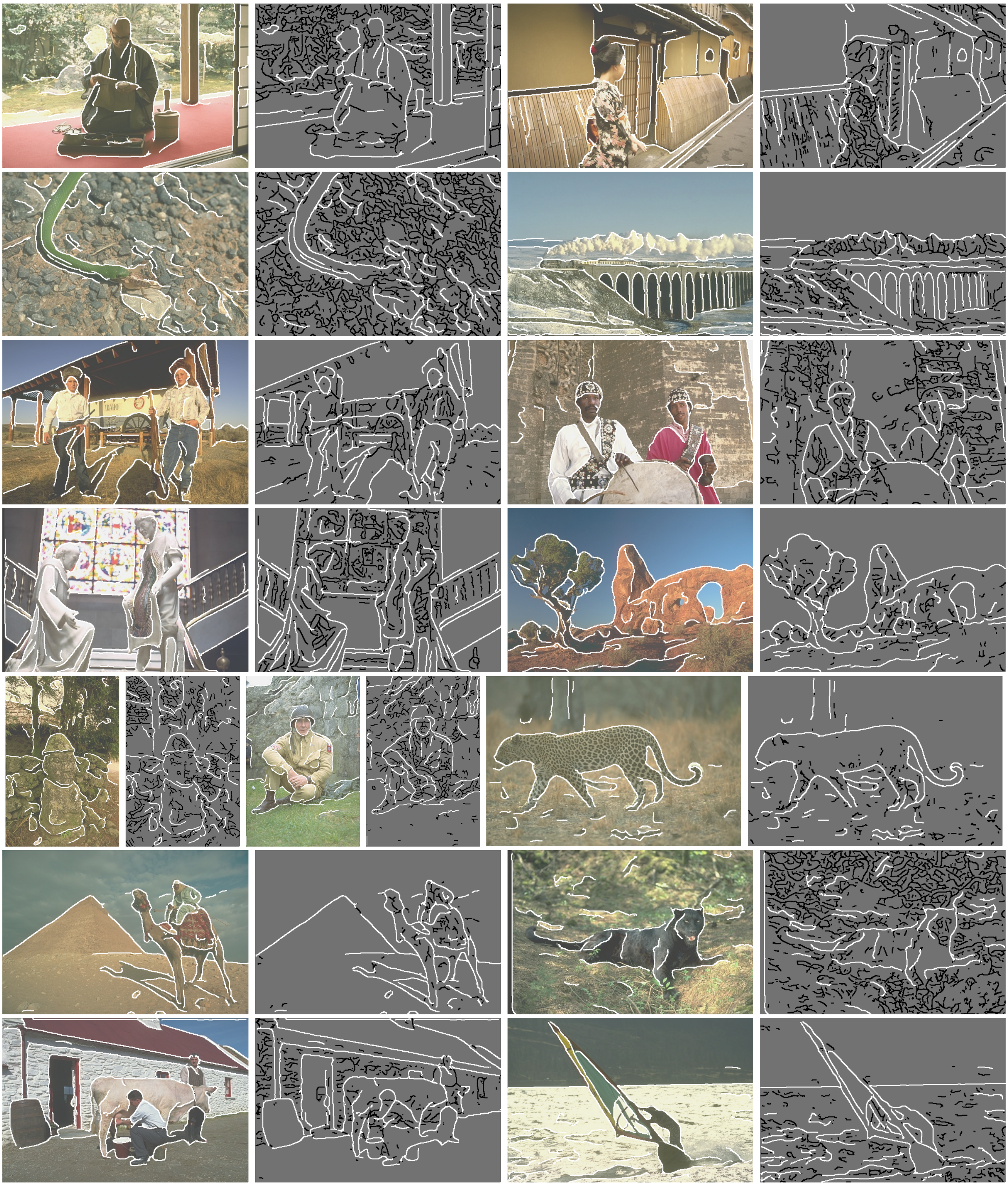


Figure 13. Contour grouping result on Berkeley Segmentation Dataset. See the caption of Figure 11 for description.