2007

Untangling Cycles for Contour Grouping

Qihui Zhu  
*University of Pennsylvania*

Gang Song  
*University of Pennsylvania*

Jianbo Shi  
*University of Pennsylvania, jshi@cis.upenn.edu*

Follow this and additional works at: [http://repository.upenn.edu/cis_papers](http://repository.upenn.edu/cis_papers)

Part of the [Computer Sciences Commons](http://repository.upenn.edu/cis_papers)

Recommended Citation

Untangling Cycles for Contour Grouping

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.

Disciplines
Computer Sciences

Comments
Wang, L., Shi, J., & Song, G. IEEE Conference on Computer Vision.

©2011 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

DOI: 10.1109/ICCV.2007.4408929

This conference paper is available at ScholarlyCommons: http://repository.upenn.edu/cis_papers/523
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.

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 circular embedding in the complex domain, where disentan-
We define a directed graph for contour grouping using three directed cycle formulation. 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, 16, 11, 2, 1, 14, 18]. 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, 17] 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 \( G = (V, E, W) \) as follows.

Graph nodes \( V \) correspond to all edgels. Since the edge orientation is ambiguous up to \( \pi \), we duplicate every edgell into two copies \( i \) and \( j \) with opposite directions \( \theta, \theta + \pi \).

Graph edges \( E \) include all the pairs of edgels within some distance \( r_c \): \( E = \{(i, j) : \| (x_i, y_i) - (x_j, y_j) \| \leq r_c \} \).

Since every edgell is directed, we connect each edgell \( i \) only to the neighbors in its direction.

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

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

Here \( i \rightarrow j \) means that \( j \) is in forward direction of \( i \). \( W_{ij} > 0 \) implies that \( W_{ji} = 0 \), \( \phi_i \) and \( \phi_j \) 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 \( j \). 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-tale 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 to 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 $S, O, k$)**

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

$S$: Subset of graph nodes V, i.e. $S \subseteq V$.

$O$: Cycle ordering on $S$.

$k$: Cycle thickness.

**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 $V$:

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

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 [15], 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:

$$O : S \mapsto S = \{1, 2, ..., |S|\}$$

where $O$ 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 $O$. Edge $(i, j)$ is forward if $0 < O(j) - O(i) \leq k$; backward if $-|S|/2 \leq O(j) - O(i) < 0$; 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 $S$, $O$ and $k$, we define internal cut as the total entangled random walk transition probability:

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

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

\[ T(k) = k/|S| \]  

(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 \( O \), 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. [15]) 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:

\[ O_{\text{circ}} : V \mapsto (r, \theta) : O_{\text{circ}}(i) = x_i = (r_i, \theta_i) \]  

(8)

Here \( r_i \) is the circle radius which can only take a positive fixed value \( r_0 \) or 0. \( \theta_i \) 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 \( \theta \) 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:

\[ \Delta \theta = \bar{\theta}_j - \bar{\theta}_i \]  

(9)

Note that the average only counts \((i, j)\) where there is an edge \((i, j)\) in the original contour grouping graph. Since angle \( \theta \) 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:

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

(a) Untangling cycle criteria (b) Circular embedding

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

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. Computational solution

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.

5.1. A continuous relaxation solution

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

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

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

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

\[
\sum_{i,j} P_{ij} \Re((x_i x_j e^{-i \Delta \theta})/r_0^2) \cdot t_0
\]

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

\[
\max_{\Delta \theta \in \mathbb{R}, x \in \mathbb{C}^n} \frac{\Re(x^H P x \cdot t_0 e^{-i \Delta \theta})}{x^H x}
\]

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 \( \Re(\langle u^H P v \cdot c \rangle / \langle u^H v \rangle) \) with \( u, v \in \mathbb{C}^n \). Furthermore, it is equivalent to the following optimization problem:
fore, we should search for a sequence enclosing the largest area in the complex plane:

$$\max_{u, v \in \mathbb{C}} \mathcal{R}(u^H P v) \quad s.t. \quad u^H v = c$$

(15)

This problem leads exactly to $P$’s complex eigenvectors.

**Theorem 1.** All the critical points (local maxima) $(u_{\text{max}}, v_{\text{max}})$ of the optimization problem (15) are given by the left and right eigenvectors of $P$ respectively, i.e.,

$$P v_{\text{max}} = \lambda v_{\text{max}} \quad \text{and} \quad B^T u_{\text{max}} = \lambda u_{\text{max}}.$$  Furthermore, the corresponding maximal value is $\max_{\lambda}(\mathcal{R}(\lambda \cdot c))$ where $\lambda$ is one eigenvalue of $P$.

**Proof.** Please see [19].

The complex eigenvectors give 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 $\lambda$ because $\mathcal{R}(\lambda \cdot c)$ reaches its maximum when the phase angles of $\lambda^*$ and $c = t_0 e^{-i \Delta \theta}$ are most similar. Notice that the complex eigenvector is only 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.

5.2. 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 $u_1, \ldots, u_{2n}$. 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. Therein this embedding space. The larger the cycle is geometrically, the better the 1D graph cycle is topologically. Therefore, we should search for a sequence $s_1, s_2, \ldots, s_h, s_{h+1} = s_1$ such that $|u_{s_1}|, \ldots, |u_{s_h}|$ are large and $\theta(u_{s_2}), \ldots, \theta(u_{s_h})$ are in an increasing order. This can be tackled by finding the sequence enclosing the largest area in the complex plane:

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

(16)

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

To simplify the search, we can pack $u_i$ into bins $B_1, \ldots, B_m$ 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 $u_{s_1}, \ldots, u_{s_h}$ 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. (16) can be reduced to

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

(17)

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. (17) becomes

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

(18)

where each summation itself is a shortest path.

5.3. Algorithm

In summary, our untangled cycle algorithm has 3 steps:
6. Experiments

We test our untangling cycle algorithm on a variety of challenging real images, including Berkeley Segmentation Dataset [10], Weizmann horse database [3] and Berkeley baseball player dataset [13]. 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 [13] 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.

7. 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 \( P \). Each complex eigenvector produces a complex circular embedding \( u_1, u_2, ..., u_{2n} \in \mathbb{C} \).

**Algorithm 1 Untangling cycles**

1. **Graph setup** Construct the directed graph \( G \) and compute transition matrix \( P \) by Eq. (1) and (2).
2. **Complex embedding** Compute the first \( n_c \) complex eigenvectors of \( P \). Each complex eigenvector produces a complex circular embedding \( u_1, u_2, ..., u_{2n} \in \mathbb{C} \).
3. **Cycle tracing** For \( u_1, u_2, ..., u_{2n} \), use shortest path to find a cycle \( S \subseteq \{1, ..., 2n\} \) minimizing (Eq. (17)).

**References**


Figure 8. Precision recall curve on the Berkeley benchmark, with comparison to Pb [10], CRF [13] 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.

Figure 9. 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.