SHAPE DETECTION BY PACKING CONTOURS

Qihui Zhu

A DISSERTATION

in

Computer and Information Science

Presented to the Faculties of the University of Pennsylvania in Partial

Fulfillment of the Requirements for the Degree of Doctor of Philosophy

2010

Jianbo Shi
Supervisor of Dissertation

Jianbo Shi
Graduate Group Chairperson

Acknowledgements

First and foremost, I would like to thank my advisor, Prof. Jianbo Shi, for both his insightful advice on research and sincere suggestions on my life and career. Through the years at Penn, Jianbo has greatly broadened my scope on research. On the other hand, he has offered numerous sharp comments that has kept me on the right track in science. Without his persistent guidance and support, I would not have been gone this far in the adventure of computer vision.

I am also grateful to all the members in my thesis committee: Camillo Jose Taylor, Sanjeev Khanna, Jean Gallier at Penn, and Longin Jan Latecki from Temple University. CJ organized the committee and gave extremely helpful advice on the overall presentation. Sanjeev asked deep algorithmic questions that stimulate my further thinking on the topic. Jean kindly provided with many suggestions on mathematics and writing (along with his French humor). Longin inspired me in many aspects of contour based shape detection, which composes a major part of this thesis.

I have interacted with several wonderful faculty members in CIS department: Kostas Daniilidis, Sampath Kannan, Ben Taskar, and Lawrence K. Saul. I would like to thank Kostas for his warm encouragement and continuous support on my research over these years, Sampath and Ben for serving on my WPE-II committee and giving me many valuable advices on primal-dual algorithms, and Lawrence for mentoring me on scientific writing inside and outside his course.

Furthermore, my research work received much help from our talented and friendly group members and alumni. I would like to thank Gang Song for all his spiritual and technical support from day one, Praveen Srinivasan, Liming Wang, and Yang Wu for their excellent collaboration on laying down the foundation of this work, Timothee Cour, Elena Bernardis, Katerina Fragkiadaki, Jeffrey Byrne, Jack Sim, Weiyu Zhang, and Haifeng Gong for their generous help on research and for making the lab a fun and nice place.

I am indebted to my other collaborators including Philippos Mordohai (Stevens), Stella X. Yu (BC), Kilian Q. Weinberger (WUSTL), Fei Sha (USC), and Lawrence K. Saul (again, now at UCSD). Philippos offered various help on the early draft of contour grouping (Chapter 2) and contour packing (Chapter 3), as well as other research projects. Stella guided me on exploring many topics on perceptual organization and graph embedding. The early work with Kilian, Fei and Lawrence on manifold embedding inspired me on the computational solution of region packing (Chapter 6).

In addition, I was lucky enough to be surrounded and helped by current and former GRASP Lab members: Alexander Toshev, Mirko Visontai, Ameesh Makadia, Alexander Patterson IV, Yuanqing Lin, Arvind Bhusnurmath, and Ben Sapp. I also cherish good memories with my friends in CIS department including Tingting Sha, Qian Liu, Yi Feng, Liming Zhao, Peng Li, Liang Huang, Stephen Tse, John Blitzer, Jinsong Tan, Mengmeng Liu, Zhuowei Bao, Jianzhou Zhao, and many many others. I would like to give special thanks to our graduate coordinator Michael Felker and administrative coordinator Charity Payne for their assistance with all kinds of requests I brought up.

Last but also the most importantly, I would like to thank my family. My parents have implanted in me the interest and passion on knowledge since childhood. I am grateful for their endless love and continuous support in my career pursuit for so many years. This thesis is dedicated to them.

ABSTRACT

SHAPE DETECTION BY PACKING CONTOURS

Qihui Zhu

Jianbo Shi

Humans have an amazing ability to localize and recognize object shapes from natural images with various complexities, such as low contrast, overwhelming background clutter, large shape deformation and significant occlusion. We typically recognize object shape as a whole the entire geometric configuration of image tokens and the context they are in. Detecting shape as a global pattern involves two key issues: model representation and bottom-up grouping. A proper model captures long range geometric constraints among image tokens. Contours or regions that are grouped from bottom-up capture correlations of individual image tokens, and often appear as half complete shapes that are easily recognizable. The main challenge of incorporating bottom-up grouping arises from the representation gap between image and model. Fragmented image structures usually do not correspond to semantically meaningful model parts.

This thesis presents *Contour Packing*, a novel framework that detects shapes in a global and integral way, effectively bridging this representation gap. We first develop a grouping mechanism that organizes individual edges into long contours, by encoding Gestalt factors of proximity, continuity, collinearity, and closure in a graph. The contours are characterized by their topologically ordered 1D structures, against otherwise chaotic 2D image clutter. Used as integral shape matching units, they are powerful for preventing accidental alignment to isolated edges, dramatically reducing false shape detections in clutter.

We then propose a set-to-set shape matching paradigm that measures and compares holistic shape configurations. Representing both the model and the image as a set of contours, we seek packing a subset of image contours into a complete shape formed by model contours. The holistic configuration is captured by shape features with a large spatial extent, and the long-range contextual relationships among contours. The unique feature of this approach is the ability to overcome unpredictable contour fragmentations. Computationally, set-to-set matching is a hard combinatorial problem. We propose a linear programming (LP) formulation for efficiently searching over exponentially many contour configurations. We also develop a primal-dual packing algorithm to quickly bound and prune solutions without actually running the LPs.

Finally, we generalize set-to-set shape matching on more sophisticated structures arising from both the model and the image. On the model side, we enrich the representation by compactly encoding part configuration selection in a tree. This makes it applicable to holistic matching of articulated objects with wild poses. On the image side, we extend contour packing to regions, which has a fundamentally different topology. Bipartite graph packing is designed to cope with this change. A formulation by semidefinite program ming (SDP) provides an efficient computational solution to this NP-hard problem, and the flexibility of expressing various bottom-up grouping cues.

Contents

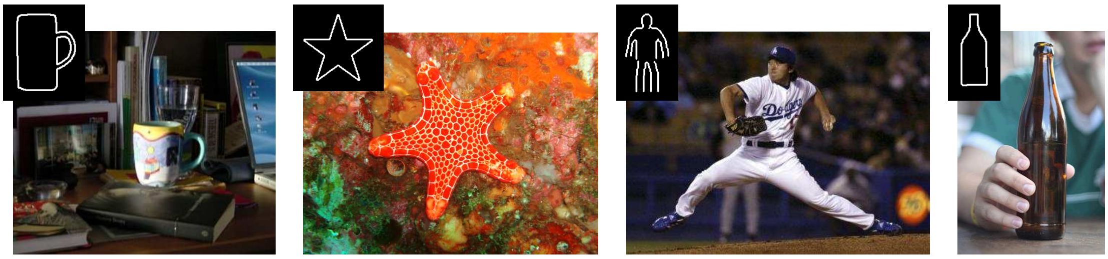
List of Tables

List of Figures

Chapter 1

Introduction

Humans have an amazing ability to localize and recognize object shapes from an image with various complexities, such as low contrast, overwhelming background clutter, large shape deformation, and significant occlusion (see Fig. 1.1). Shape is not only a useful cue for object recognition, but also an important problem by itself because it leads to further understanding of the geometric arrangement of the scene, and functional properties of objects.



(a) Low contrast (b) Background clutter (c) Deformation (d) Occlusion

Figure 1.1: Complexities in real images. In (a), part of the mug is covered by shadow. The contour of the starfish in (b) is surrounded by both clutter in the background, and texture in the foreground. The baseball player in (c) has a very different pose than the canonical model. Part of the bottle in (d) is occluded by a person's hand. Despite all these complexities, a human has no difficulty in locating and matching objects to the target shape models shown at the top left corner.

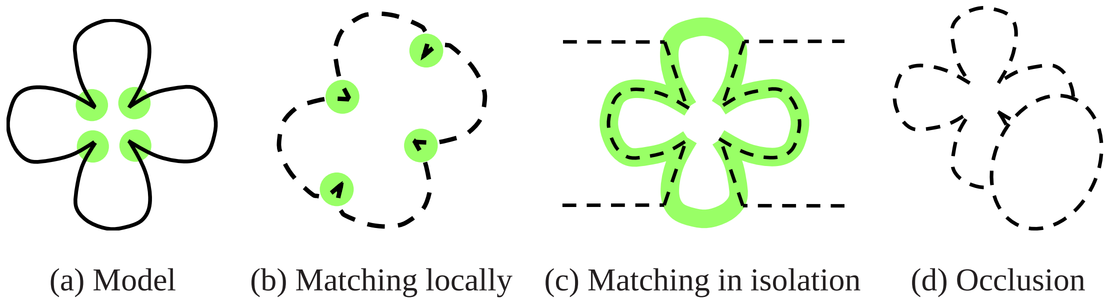


Figure 1.2: The global percept of shapes. (a) presents a simple shape with a silhouette. Image tokens that fits to the target locally could compose a completely different shape as shown in (b). The local neighborhoods of (a) and (b) marked in green have identical junctions, with the curvature of the smooth silhouettes similar in most of the places. Matching shapes by aligning edges independently could contrive false hypotheses as shown in (c). Most of the silhouette in (a) can be aligned to some individual edges in (c). They group with the horizontal lines as integral contours, and those lines do not have matches to the target. In (d), although part of the object silhouette is also missing, most likely the object has the same shape as the target, and missing silhouette is only due to occlusion.

1.1 Motivation

Shape is fundamentally a *global percept* – we typically recognize object shape as a whole. By "global" we mean the following two concepts:

1. *Non-locality.* Shapes are measured by the *entire* geometric configuration of image tokens, rather than their local properties. Unlike other object properties such as texture, a shape hardly has small distinctive parts that can uniquely identify it.

2. *Non-isolation.* Shapes are formed by orderly structures that *link* image tokens together, instead of independent image tokens. Grouping on these tokens provides a context where the shape could be extended, and what could be the other alternatives.

Fig. 1.2 illustrates false shape matching examples ignoring either one of these two aspects. An image hypothesis can locally fit the shape prototype in most of the places, but overall does not resemble the target at all. On the other hand, a subset of individual edges can be aligned to the prototype perfectly, but edges connected to them do not have matches, and cause errors faraway from the matched edges.

In light of the above observations, *model representation* and *bottom-up grouping* are key issues to consider in order to detect shapes robustly from images. A proper model representation handles the non-locality problem by capturing long range geometric constraints. During the search process, image tokens that are far apart can be bound by the model, interpreted and checked via their configurations. Bottom-up image structures such as contours identify the underlying correlation of individual edges, which can be extracted from the image independent of the shape model. Matching with these integral shape tokens avoids many accidental alignments to isolated edges in the clutter.

Previous shape detection and matching approaches can be classified into two groups by model representation: shape primitive based methods and template-based methods.

Shape primitive based methods. These approaches assume that shapes are composed of some high level generic primitives, or volumetric parts that constitute objects via certain basic rules. These components include generalized cylinders (Brooks, 1983), superquadratics (Pentland, 1986), geons (Biederman, 1985), and ribbons (Nevatia & Binford, 1977). Although perceptually these primitives make proper abstraction of the shape models, they are hard to detect from images reliably. The representation gap between the model and the image poses a big challenge: a shape recognition system has to connect raw image edges or pixels into contours or surfaces, and then assemble them into these high level primitives. This results in two typical problems which eclipses the application of these methods in real images. First, previous search procedures such as Interpretation Tree (Grimson & Lozano-Perez, 1987) are insufficient to explore the huge, usually exponential, solution space. Second, many premature hard decisions have to be made before reaching the final output since the primitives are several levels above the image pixels.

Medial axis based representations (Blum, 1967; Peleg & Rosenfeld, 1981; Leymarie & Levine, 1992; Bai *et al.* , 2007) continue on the path of these attempts to develop high level primitives. Several shape descriptions such as Shock Graphs (Siddiqi *et al.* , 1999) and Poisson equation based features (Gorelick *et al.* , 2006) effectively capture global shapes as well as semantical parts. Because medial axes are sensitive to region boundaries, all these approaches assume that object regions and their boundaries have been segmented from the background. However, segmenting foreground objects correctly is a hard problem to solve on its own in shape detection. Medial axis is a useful representation for describing and matching holistic shapes given the foreground regions, but does not provide insights on how to search the target shape from image regions with over-segmentation or under-segmentation.

Template-based methods. A separate path of research has been focusing on building shape templates by low level, and detectable tokens. This essentially brings the model representation all the way down to the image, such that the patterns of model representation are repeatable in images. For example, the tokens can be as simple as edge points. Chamfer matching (Barrow *et al.* , 1977; Shotton *et al.* , 2008) and Hausdorff matching (Huttenlocher *et al.* , 1993) are representatives of when the model is merely a set of unordered points with fixed locations. The tokens can also be keypoints along with local shape or appearance descriptors. Shapes are represented as the spatial configurations of these keypoints, *e.g.* geometric hashing (Lamdan *et al.* , 1990), decision tree (Amit & Wilder, 1997) and Active Shape Models (ASM) (Cootes *et al.* , 1995). However, keypoints alone are insufficient to distinguish objects shapes in cluttered images (Belongie *et al.* , 2002). Recent attempts such as Shape Contexts (SC) (Belongie *et al.* , 2002), Histogram of Gradients (HOG) (Dalal & Triggs, 2005) and Scale Invariant Features (SIFT) (Lowe, 2004) construct tokens from spatial histograms which encode local shape information centered at keypoints or the object center. The model usually employs a graph on the tokens, either a pair-wise connected graph (SC) or a star graph (HOG, SIFT), to capture the long-range geometric constraints of the entire shape (Leordeanu *et al.* , 2007).

Template-based methods have achieved certain success by bringing the model closer to image signals, but sacrificing the generalizability. Because the tokens only contain very local information, the templates made of these tokens are often specific to some instances rather than generic for the whole object category. Therefore, object models result in either a large number of exemplars (Torralba *et al.* , 2009), with each one of them sensitive to shape deformation, or composites from complicated grammatical rules (*e.g.* AND/OR graphs) (Zhu & Mumford, 2006; Han & Zhu, 2009).

Although many model representations have addressed the non-local shape configuration, bottom-up grouping has been missing in most of the previous works. Contour grouping or region segmentation naturally pops out many object shapes. Starting with half complete shapes appearing in grouped contours or region segments greatly reduces the search space of shape matching (Grimson, 1986). In contrast, most template-based methods resort to matching the shape model with individual edges or pixels. Shapes are not perceived by randomly linking edges or pixels, but by organizing them in a simple, regular and orderly form called *Prägnanz* (Palmer, 1999). The principle of Prägnanz, advocated by Gestalt psychologists in the early 20th century (Kohler, 1929; Koffka, 1935; Wertheimer, 1938), involves grouping elements by the laws of proximity, similarity, continuity, closure, symmetry and common fate. Contour grouping or region segmentation organizes the image by integrating several of these factors. The resulting contours or regions are semi-finished products towards forming the entire shape, which save constructing shapes from scratch with edges or pixels.

A deeper consequence of incorporating bottom-up grouping is turning the overall shape matching cost into a non-additive function. This is phrased by the Gestalt principle *"the whole is greater than the sum of the parts"* (Wertheimer, 1938). The additivity of the shape matching cost function has been recognized as a main cause of accidental alignments to clutter (Amir & Lindenbaum, 1998). For example, chamfer matching sums up errors on many edges to a total cost. The additive cost cannot distinguish a similar shape with gaps versus a different shape partially aligned with the model (see Fig. 1.2 (c),(d)). Additivity of local errors implicitly assumes the statistical independence of edges. However, image edges do not occur in isolation, and errors made by the edges tend to be correlated. Bottom-up grouping identifies intermediate structures such as contours and regions that constitute an image and capture the dependency of edges on them. Utilizing these bottom-up image structures can greatly improve the robustness of shape detection against the background clutter.

The main challenge of incorporating bottom-up grouping arises from the *representation gap* between image structures and the shape model. Bottom-up contours or regions do not necessarily correspond to semantically meaningful model parts, and the fragmentations of contours and regions can vary from image to image. At a junction formed by occlusion, a contour could continue to complete the figure, stop for further reasoning, or leak to the background. A contour could also span multiple object parts when edges continue smoothly, with little distraction around. These situations break the one-to-one correspondences between contours and model parts, and hence complicate the shape matching process. This results in either sophisticated construction of the model (Latecki *et al.* , 2008), or expensive search on bottom-up fragmentations (Keselman & Dickinson, 2005).

1.2 Outline and Contributions

This thesis presents *Contour Packing*, a novel framework that detects shapes in a non-local, non-isolated way, addressing the issues of both model representation and bottom-up grouping.

We exploit long and salient contours extracted by bottom-up grouping as shape primitives, instead of using short edges or local patches. These bottom-up contours have a large spatial extent allowing the recognition of global geometry, and capture the correlation of individual edges forming the shape. With both the model and the image represented by contours, we seek a packing of a subset of image contours into a complete global shape similar to the one composed by model contours. The unique feature of contour packing is the ability to describe and match the holistic shape configurations of two contour sets, but neglecting the difference of their fragmentations. In this way, the representation gap between the bottom-up image structures and the top-down shape model is effectively bridged.

In contour packing, the model representation addresses the non-locality aspect of shape in two levels. In the level of shape tokens, these contours themselves encode useful geometric constraints on faraway edges, especially when contours are long and curved. More importantly, the assembly of the contours in the structure level takes into account the global geometric context – contours are packed if all their surrounding contours have the right placement. This work has made the following contributions on shape detection:

1. We develop a grouping mechanism that organizes individual edges into ordered topologically 1D structures, against otherwise chaotic 2D image clutter. Gestalt factors of proximity, continuity, collinearity, and closure on edges are integrated via a directed graph. Our formulation achieves simultaneous segmentation and parameterization of image contours as 1D cycles in this graph. Maintaining contours as integral units for matching can drastically reduce false shape detections in clutter.

2. We propose a set-to-set shape matching paradigm that measures and compares holistic shape configurations formed by two sets of contours. The holistic configuration is captured by shape features with a large spatial extent, and the long-range contextual relationship among contours. Unlike traditional local features that are precomputed before shape matching, our approach adjusts shape features according to figure/ground selection. As a result, it provides an effective way to overcome unpredictable fragmentations on bottom-up contours or regions.

The above principles are achieved by the following computational tools:

1. A complex eigenvector solution for extracting multiple contours as graph cycles;

2. A formulation that searches for a holistic shape matched to the target over combinatorially many subsets of contours;

3. An efficient primal-dual algorithm to search and bound contour packing solutions;

4. Extensions of contour packing to accommodate additional structures including deformable model composition and figure/ground region selection.

We describe the key components to develop in the next few chapters as follows:

First, Chapter 2 translates grouping topologically 1D contours into finding persistent random walks in a weighted directed graph. Representing contours as random walk cycles in the graph captures ordering, the essential property of a topologically 1D structure. We derive the mathematical connection from cycle persistence to complex eigenvalues of the random walk matrix. This connection leads to the solution of computing complex eigenvectors, and tracing cycles in the corresponding complex embedding space.

In Chapter 3, we formulate the maximal, holistic set-to-set matching of shapes as finding the correct figure/ground contour selection, and the optimal correspondences of control points on or around contours. This task is simplified by encoding the feature descriptor algebraically in a linear form of contour figure/ground selection variables. This allows us to formulate set-to-set matching as an instance of linear programming (LP), which enables the efficient search over exponentially many figure/ground contour selections.

The LP arising in the set-to-set matching is reduced to a fractional packing problem in Chapter 4, where contours and feature descriptor bins correspond to items and knapsacks, respectively. We derive a primal-dual combinatorial algorithm for contour packing which exploits the duality of packing and covering. The primal-dual algorithm gives a deeper algorithmic understanding of the search process, and is capable of bounding and pruning suboptimal solutions without running LP to convergence.

In Chapter 5, we enrich the model representation by incorporating part configuration selection, making it applicable to deformation and articulation of object shapes. The model encodes exponentially many configurations through a compact set of selection variables. We extend the LP based set-to-set matching method to this representation, which efficiently searches the combinatorial space formed by image contours and model poses.

In Chapter 6, we extend contour packing further to regions, which have a fundamentally different topology than contours. We propose bipartite graph packing to cope with this variation. Regions are represented by graph nodes and boundary fragments between regions are represented by edges whose weights indicate their contributions to shape.

Packing bipartite edges can be casted as semidefinite programming (SDP) for efficient computation. Several grouping constraints from the graph partitioning setting naturally fit into the formulation, increasing the expressive power of region packing. We demonstrate promising results that simultaneously detect object shapes and their foreground region support.

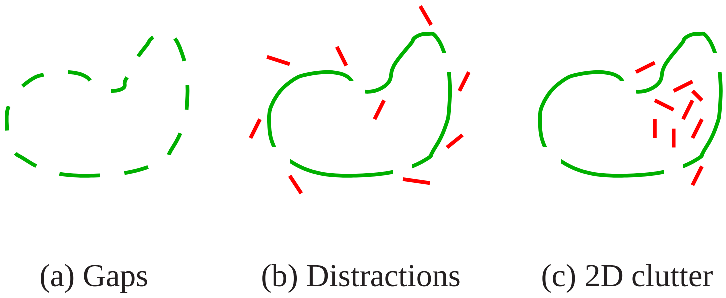
On the theoretical side, contour packing provides an effective solution that can extract and assemble intermediate image structures into shapes composed of high level semantic parts. The set-to-set matching opens up shape detection to an extent that it does not rely on locally distinctive features (and hence the matching does not have to be one-to-one). It also provides a search mechanism on the combinatorial space due to shape composition. On the practical side, our approach resists background clutter in natural images, and generalizes well to object shape deformations even with few training examples. The approach shows promising results on detecting objects like mugs, bottles, and swans and estimating human poses in cluttered images. We believe that the packing based computational paradigm shall have many more applications in computer vision.

Chapter 2

Contour Grouping

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 memory on object shapes, and speed up visual perception (Koffka, 1935). A stable bottom-up salient contour grouping mechanism is extremely helpful to shape detection. Long contours provide global structural information on shapes, which is not captured by individual short edges or local patches (Ullman & Shashua, 1988). Contours also simplify object recognition by aligning model shapes to a few salient structures instead of tremendous edge points in the image (Ullman, 1996).

In this chapter we study contour grouping from a novel perspective of topology. The fundamental distinction between a curve-like contour and a collection of random edges is that a contour must be *topologically 1D* (see Fig. 2.2). By topologically 1D, we mean a set of edge points that have one well defined order, and the connections among them strictly follow that order. To detect contours from images, we need to ask a harder question: does the image contain any 1D curve-like structure, and if so, can we show that it is topologically 1D? Looking at the topology explicitly excludes 2D clutter, *i.e.* region-like structures from our contour search. Regions of 2D clutter can contain short edges with high contrast locally, but does not form a long, contiguous 1D sequence. We formulate contour detection as extracting persistent cycles in a directed weighted graph. These cyclic structures generate periodic random walks, which we found closely related to complex eigenvalues



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

Figure 2.1: 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.

of the graph weight matrix. This observation leads to the efficient computational solution of finding the top complex eigenvectors, and tracing cycles in the corresponding complex embedding space.

2.1 Overview

Detecting salient contours without reporting many false edges remains a challenge for incorporating this bottom-up information into object recognition. Contour grouping methods often start with edge detection, and followed by linking edgels[1] to optimize a saliency measure (Ullman & Shashua, 1988). Finding salient contours is reliable when images are clean, and contours are well separated. Gestalt factors of grouping, such as proximity collinearity, and continuity, define the local likelihood of connecting two nearby edgels. A local greedy search, such as shortest path, guided by the grouping measure can compute an optimal contour efficiently. However, existing contour grouping algorithms often fail on natural images where image clutter is mixed with gaps on contours. Fundamentally it is difficult to distinguish gaps versus background clutter locally (see Fig. 2.1), resulting in many false contours in cluttered regions with texture.

A key notion we introduce for this topological curve detection task is *entanglement*. Intuitively, a set of edgels is entangled if these edges cannot be organized following an

[1] In the rest of this chapter, we call an image edge point *an edgel* to avoid the confusion with *an edge* in the contour graph which connects two edgels.

order without breaking many strongly linked edgel pairs. We provide a graph embedding formulation with a topological curve grouping score which is able to evaluate both separation from the background and entanglement within the curve. Computationally, finding such curves requires *simultaneously* segmenting a subset of edgels and determining their order in the graph. The general task of searching for subgraphs with a specified topology is a much harder combinatorial problem. We translate it into a circular embedding problem in the *complex* domain, where entanglement can be easily encoded and checked. We seek the desired circular embedding by computing complex eigenvectors of the graph weight matrix.

The use of graph formulation for contour grouping has a long history, and we have drawn ideas from many of them (Mahamud *et al.* , 2003; Ullman & Shashua, 1988; Medioni & Guy, 1993; Amir & Lindenbaum, 1998; Alter & Basri, 1996; Sarkar & Soundararajan, 2000; Yu & Shi, 2003; Ren *et al.* , 2005b). The most related work is (Mahamud *et al.* , 2003) which uses 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 (Elder & Zucker, 1996; Jacobs, 1996; Mahamud *et al.* , 2003; Wang *et al.* , 2005) seek salient closed contours. In contrast, we seek closed topological cycles that can include open contours, and are more robust to clutter. We are also motivated by the work of (Fischer & Buhmann, 2003) which showed classical pairwise grouping is insufficient for contour detection. However, their solution using min-max distance is sensitive to outlier and clutter. Our approach computes not only the parameterization, but also the segmentation of contours simultaneously.

The rest of this chapter is organized as follows. In Section 2.2, we define a directed contour grouping graph and outline the three untangling cycle criteria. A novel circular

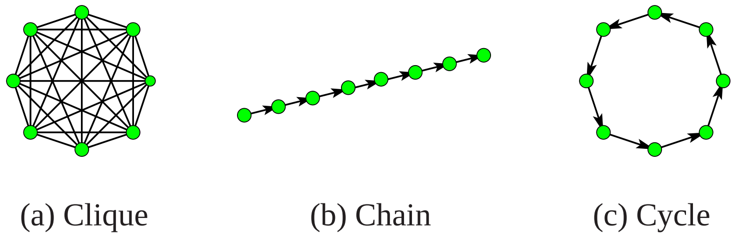

(a) Clique (b) Chain (c) Cycle

Figure 2.2: Distinction of 1D vs 2D topology. (a) The 2D topology (*e.g.* regions) assumes a clique model. In (b), (c) The 1D topology assumes a chain or a cycle model. A ring has a 1D topology but is geometrically embedded in 2D.

embedding is introduced to encode these untangling cycle criteria. We show how a continuous relaxation of the circular embedding leads to computing the complex eigenvectors of the graph weight matrix in Section 2.3. An alternative interpretation using random walk is presented in Section 2.4, with explanations on its close connection to the complex eigenvalues. We summarize our computational solution in Section 2.5 and demonstrate experimental results in Section 2.6. The chapter is concluded by Section 2.7.

2.2 Untangling Cycle Formulation

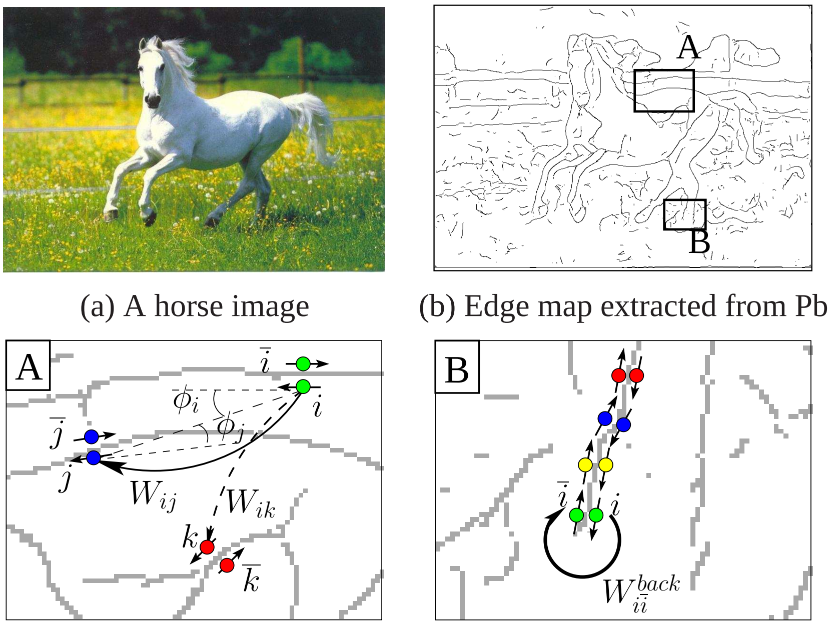
In this section, we formulate the topological requirement of 1D structures as *Untangling Cycle Cut Score* defined on a *directed* contour grouping graph.

2.2.1 Directed Graph and Contour Grouping

We start by introducing the construction of the graph. For contour grouping, we first threshold the output of an edge detector (*e.g.* Probability of Boundary (Pb) (Martin *et al.* , 2001) or (Maire *et al.* , 2008)) to obtain a discrete set of edgels. We define a directed graph on these edgels $\mathbf{G} = (\mathbf{V}, \mathbf{E}, \mathbf{W})$ as follows.

- The set of graph nodes $\mathbf{V}$ corresponds 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 and $\theta + \pi$.

- The set of graph edges $\mathbf{E}$ includes all the pairs of edgels within some distance r_e:



(a) A horse image (b) Edge map extracted from Pb

(c) W_{ij} at non-terminal nodes (d) W_{ij} at terminal nodes

Figure 2.3: Directed graph for contour grouping. Zoom-in views of graph weights W_{ij} in windows A and B are shown in (c) and (d) respectively. 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 $W_{i\bar{i}}^{back}$ to duplicated nodes $i, \bar{i}$ turns a topological chain into a cycle.

$\mathbf{E} = \{(i,j) : \|(x_i, y_i) - (x_j, y_j)\| \leq r_e\}$. Since every edgel is directed, we connect each edgel i only to the neighbors in 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:

$$W_{ij} = e^{-(1-\cos(|\phi_i|+|\phi_j|))/\sigma^2} \quad \text{if } i \longrightarrow j \tag{2.1}$$

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. 2.3(c)).

In this graph, an ideal closed contour forms two directed cycles, one for each duplicated direction. Similarly, an ideal open contour 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. 2.3).

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 the random walk interpretation of Normalized Cut (Meila & Shi, 2000). We define the random walk matrix:

$$P = D^{-1} \cdot W \tag{2.2}$$

where D is diagonal with $D_{ii} = \sum_j W_{ij}$. This amounts to normalizing a 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.

2.2.2 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. 2.2).

Order plays an important role in distinguishing 1D topological grouping. We define **entanglement** as *connection of nodes violating a given order*. Any 1D topological structure can be put into a specific order, such that each graph node connects to exactly one successor and is connected to exactly one predecessor (see Fig. 2.2 (b)(c)). In 2D topological structures, it is impossible to find a good order without entanglement (see Fig. 2.2 (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 order. 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 Table 2.1). 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 order on it such that they maximize the following score:

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

$$C_u(S, \mathcal{O}, k) = \frac{1 - E_{cut}(S) - I_{cut}(S, \mathcal{O}, k)}{T(k)} \tag{2.3}$$

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

$\mathcal{O}$: Cycle order 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:

$$E_{cut}(S) = \frac{1}{|S|} \sum_{i \in S, j \in (V-S)} P_{ij} \tag{2.4}$$

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 (Shi & Malik, 2000), our contour

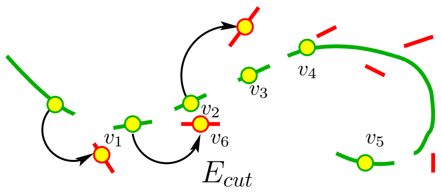
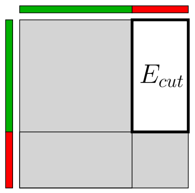
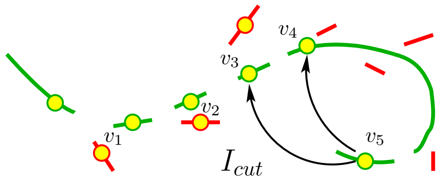
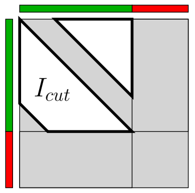
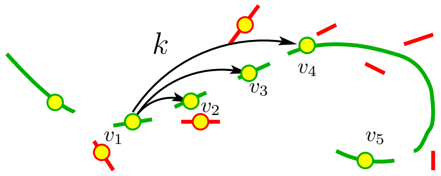
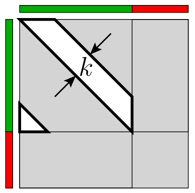
Criterion	Graph Topology	Graph Weight Matrix
External Cut $E_{cut}(S)$	 v_1 v_2 v_6 v_3 v_4 v_5 E_{cut}	
Internal Cut $I_{cut}(S, \mathcal{O}, k)$		
Tube Size $T(k)$		

Table 2.1: Illustration of 1D topological grouping criteria. The middle column visualizes a graph containing a contour (marked in green) and other background clutter edges (marked in red). The graph nodes are sorted in a way that contour nodes come first and background nodes come last, with contour nodes following the right order (see the color bar in the right column). Note that we do not know the partition and the order in advance. External cut measures the strength of connections leaking from contour nodes to background nodes shown in the first row. Internal cut measures the strength of connections within the contour that violates the order, shown in the second row. Tube size refers to how many forward step on the cycle are considered, as shown in the last row. This corresponds to the width of the band formed by contour connections in the weight matrix.

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 order. It requires minimal entanglement between nodes far away in the order. We define the node order as a one-to-one mapping:

$$\mathcal{O} : S \mapsto S = \{1, 2, ..., |S|\} \tag{2.5}$$

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

The "thickness" factor k measures the *maximal step size* defining how much each link can violate the order $\mathcal{O}$. 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 Table 2.1) 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:

$$I_{cut}(S, \mathcal{O}, k) = \frac{1}{|S|} \sum_{(\mathcal{O}(i) \geq \mathcal{O}(j)) \vee (\mathcal{O}(j) > \mathcal{O}(i) + k)} P_{ij} \tag{2.6}$$

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.

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:

$$T(k) = k/|S| \tag{2.7}$$

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 (2.3).

One way to visualize the three criteria is to observe the structures of matrix P (Fig. 2.4(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. (2.3) 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.* (Shi & Malik, 2000)) only consider external cut, but do not take internal cut and cycle thickness into account.

2.2.3 Circular Embedding

Optimizing eq. (2.3) essentially performs segmentation and parameterization on the graph *simultaneously*. We only cut out a subset of nodes with a good parameterization, *i.e.* order. This is a hard combinatorial task. Our strategy is to embed the graph into a circular space, such that the three criteria in (2.3) can be encoded and checked effectively.

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

$$\mathcal{O}_{circ} : V \mapsto (r, \theta) : \mathcal{O}_{circ}(i) = x_i = (r_i, \theta_i) \tag{2.8}$$

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 *order* of graph nodes. $S = \{v_i : r_i = r_0\}$ specifies the nodes being cut out, as in eq. (2.4). Angle θ_i specifies the order. We simplify the embedding by restricting $\theta_i = 2\pi i/|S|$ (see Fig. 2.4), *i.e.* x_i is distributed uniformly on the circle. It is important to force x_i to spread out in the circular embedding. When all of x_i's are mapped to the same point, no order information can be obtained. We also define the maximal jumping angle θ_{max} on how far

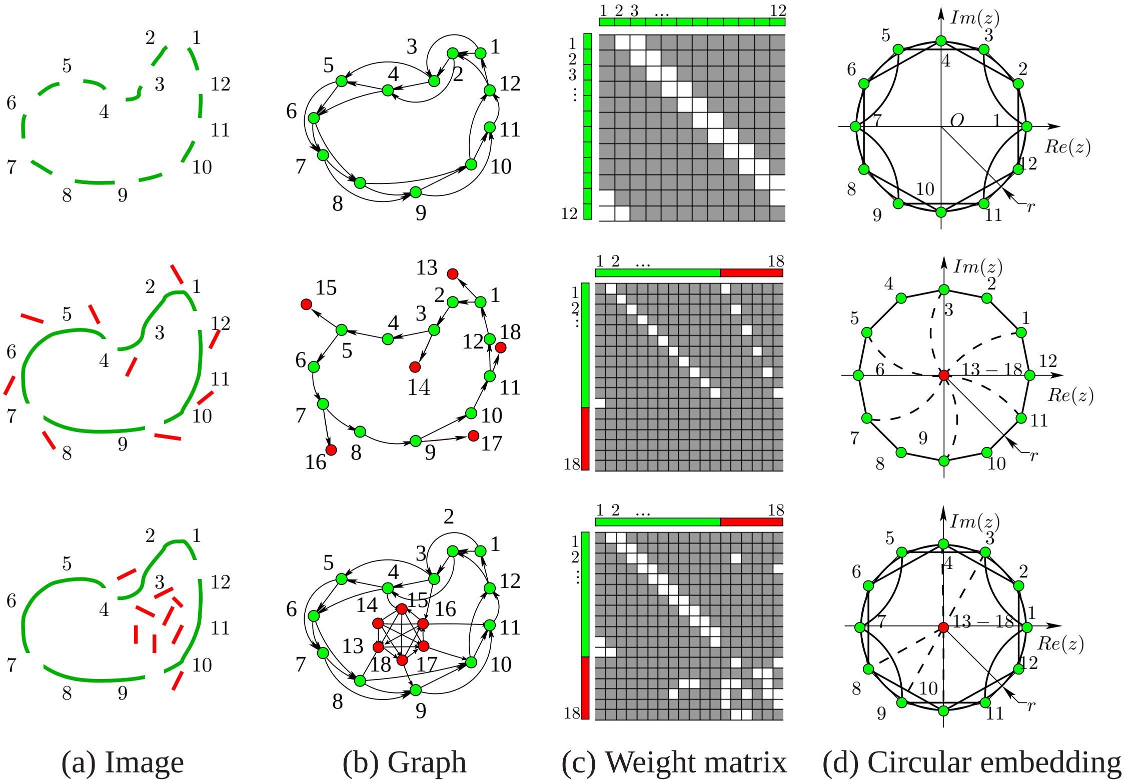


Figure 2.4: Finding 1D topological cycles in circular embedding. Three canonical cases are shown: a perfect cycle (green) shown in row 1, a cycle with sporadic distracting edges (red) in row 2, and with 2D clutter (red) in row 3. (a) Canonical image cases. (b) Directed graph constructed from edgels. (c) Random walk transition matrix P (white for strong links). (d) The optimal circular embedding. Distracting edges and 2D clutter are embedded into the origin.

it can jump from one node to another on the circle.

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, θ, θ_{max})

$$C_e(r, \theta, \theta_{max}) = \sum_{\substack{\theta_i < \theta_j \leq \theta_i + \theta_{max} \\ r_i > 0,\ r_j > 0}} P_{ij}/|S| \cdot \frac{1}{\theta_{max}} \tag{2.9}$$

r: Circle indicator with $r_i \in \{r_0, 0\}$.

θ: Angles on the circle specifying an order.

θ_{max}: Maximal jumping angle.

With the above definition, Circular Embedding Score (eq. (2.9)) is equivalent to Untangling Cycle Cut Score (eq. (2.3)). We interpret the three untangling cycle criteria in the new embedding space 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.

2. *Internal Cut* requires angles spanned by links on the circle to be small. Edges in the original graph are 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 maximal jumping angle θ_{max}. 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 $k \cdot \frac{2\pi}{|S|}$.

$$\theta_{max} = 2\pi \cdot k/S = 2\pi \cdot T(k) \tag{2.10}$$

Now we can rewrite the score function (2.3) in circular embedding, expressed by (r, θ) and the maximal jumping angle θ_{max}. Because P_{ij} is row normalized (eq. (2.2)), $\sum_j P_{ij}/|S| = 1$. 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. (2.3) can be expressed in terms of r, θ and θ_{max}:

$$1 - E_{cut}(r) - I_{cut}(r, \theta, \theta_{max}) = \sum_{\substack{\theta_i < \theta_j \leq \theta_i + \theta_{max} \\ r_i > 0,\ r_j > 0}} \frac{P_{ij}}{|S|} \tag{2.11}$$

The forward links are chords with spanning angles no more than θ_{max}. Combining eq. (2.10), (2.11), maximizing eq. (2.3) reduces to maximizing eq. (2.9) in circular embedding.

2.3 Complex Eigenvectors: A Continuous Relaxation

Now we are ready to derive a computational solution. We generalize the discrete circular embedding (2.8) by mapping the graph into the complex plane. The optimal continuous circular embedding turns out to be given by the complex eigenvectors of the random walk matrix.

First we relax both r and θ in eq. (2.9) 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. (2.9). Here $r_j = \|x_j\|$ and θ_j are magnitude and phase angle of the complex number x_j.

In order to capture the dominant mode of phase angle changes, we introduce the *average jumping angle* of the links as:

$$\Delta\theta = \overline{\theta_j - \theta_i} \tag{2.12}$$

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 order, $\Delta\theta$ describes how far one node is expected to jump through the links.

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. (2.11)). When the angle difference $\theta_j - \theta_i$ equals the 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 (2.11) becomes:

$$\frac{\sum_{ij} P_{ij} \mathrm{Re}(x_i^* x_j \cdot e^{-i\Delta\theta}) \cdot t_0}{\sum_i |x_i|^2} \tag{2.13}$$

where the denominator is exactly $|S|$ in the discrete case. Here $t_0 = 1/\theta_{max}$.

Expressed in a matrix form, eq. (2.13) becomes

$$\max_{\Delta\theta \in \mathbb{R}, x \in \mathbb{C}^n} \frac{\mathrm{Re}(x^{\mathrm{H}} P x \cdot t_0 e^{-i\Delta\theta})}{x^{\mathrm{H}} x} \tag{2.14}$$

Here $X^{\mathrm{H}} = (X^*)^{\mathrm{T}}$ denotes the conjugate transpose of matrix/vector X.

Solving eq. (2.14) is not an easy task. Moreover, we are not only interested in the best solution of eq. (2.14), but all local optima. These local optima will generate all the 1D structures in the graph. Our first step to tackle this problem is to fix $\Delta\theta$ to be a constant.

$$E(\Delta\theta) = \max_{x \in \mathbb{C}^n} \frac{\mathrm{Re}(x^{\mathrm{H}} P x \cdot e^{-i\Delta\theta})}{x^{\mathrm{H}} x} \tag{2.15}$$

The local optima of the orginal problem must also be the local optima of $E(\Delta\theta)$. The restricted problem can be solved by computing the eigenvectors of a matrix parameterized by $\Delta\theta$ as shown by the following theorem:

Theorem 2.1. *The necessary condition for the critical points (local maxima) of the following optimization problem*

$$\max_{x \in \mathbb{C}^n} \frac{\mathrm{Re}(x^{\mathrm{H}} P x \cdot e^{-i\Delta\theta})}{x^{\mathrm{H}} x} \tag{2.16}$$

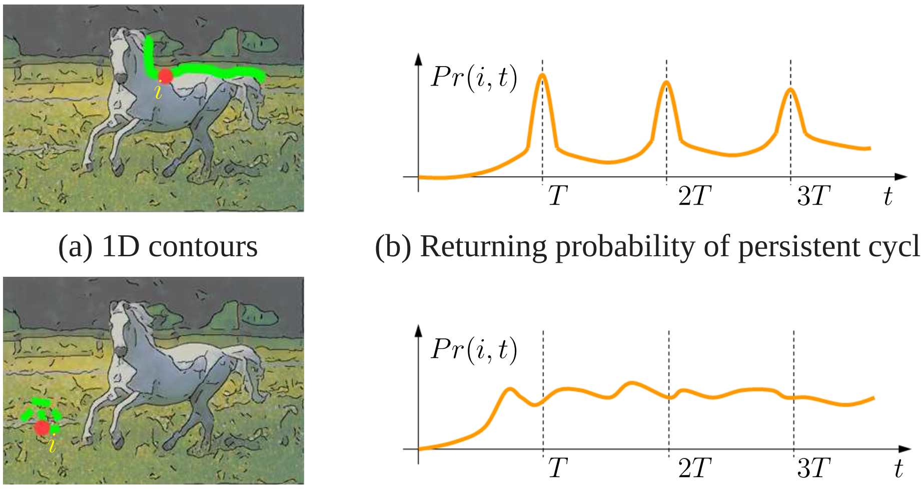
is that x is an eigenvector of

$$M(\Delta\theta) = \frac{1}{2}(P \cdot e^{-i\Delta\theta} + P^{\mathrm{T}} \cdot e^{i\Delta\theta}) \tag{2.17}$$

Moreoever, the corresponding local maximal value is the eigenvalue $\lambda(M(\Delta\theta))$.

Proof. See Appendix. □

One possibility of finding all the local optima of the orginal score function eq. (2.14) is to compute the local maxima of eigenvalues $\lambda(M(\Delta\theta))$ with respect to average jumping



(a) 1D contours

(b) Returning probability of persistent cycles

(c) 2D clutter

(d) Returning probability of non-persistent cycles

Figure 2.5: 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.

angle $\Delta\theta$. However, this approach is computationally intensive. Another alternative is to examine the eigenvectors of P directly as a proxy to the local maxima of the orginal problem. Notice that since P is asymmetric, the left and right eigenvectors (eigenvectors of P^{T}) are in general different. If both P and P^{T} permit x as a (left) eigenvector[2], x is also an eigenvector of $M(\Delta\theta)$ simply because

$$\frac{1}{2}(Pe^{-i\Delta\theta}+P^{\mathrm{T}}e^{i\Delta\theta})x=\frac{1}{2}(Px\cdot e^{-i\Delta\theta}+P^{\mathrm{T}}x\cdot e^{i\Delta\theta})=\frac{1}{2}[\lambda(P)e^{-i\Delta\theta}+\lambda(P^{\mathrm{T}})e^{i\Delta\theta}]x \tag{2.18}$$

Therefore x is indeed a local maximum by Theorem 2.1. In the subsequent sections, we will be focusing on computational solution from embedding space given by eigenvectors of P.

2.4 Random Walk Interpretation

A random walk provides an alternative view to see why complex eigenvectors are useful for untangling cycles. Random walks have been shown to be effective in analyzing region segmentation (Meila & Shi, 2000). 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.

2.4.1 Periodicity

Following traditional random walk analysis, the transition matrix $P = D^{-1}W$ (eq. (2.2)) encodes the probability of switching states. In other words, P_{ij} is the probability that a particle starts from node j and randomly walks to node i in one step. Note that P is asymmetric because the random walk is directional.

According to our graph setup in Section 2.2, 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

$$\Pr(i, t) = \sum_{\ell} \Pr(i, t \mid |\ell| = t) \tag{2.19}$$

Here ℓ is a random walk cycle with length t passing through i. However, this condition alone is not enough to identify 1D cycles. 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*.

[2] Note: this does not mean that P has to be a normal matrix, as only part of its subspaces are diagonalizable.

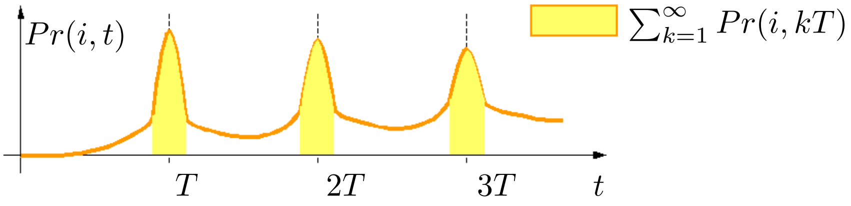


Figure 2.6: Peakness measure. $R(i,T)$ measures the 'peakness' of the returning probability $\mathrm{Pr}(i,T)$ of random walk in the graph. It can be shown that $R(i,T)$ is dominated by complex eigenvalues of the random walk matrix P.

2.4.2 Persistent Cycles

We have found that 1D cycles have a special pattern of returning probability $\mathrm{Pr}(i,t)$ (see Fig. 2.5). From analysis of Section 2.2, one step of random walk on a 1D cycle tends to stay in the cycle (external cut to be small), and move a fixed amount forward in the cyclic order (internal cut to be small). If one starts a random walk from a node in a 1D cycle, it is very likely to return at multiple times of a certain period. We call such 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 (see Fig. 2.6):

$$R(i,T) = \frac{\sum_{k=1}^{\infty} \mathrm{Pr}(i,kT)}{\sum_{k=0}^{\infty} \mathrm{Pr}(i,k)} \tag{2.20}$$

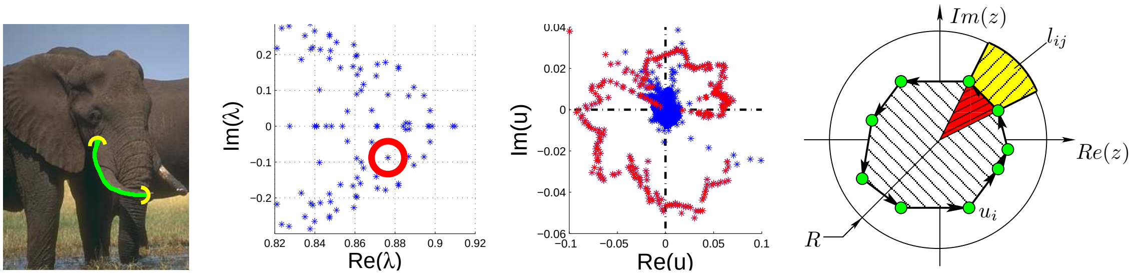
Here we compute the probability that the random walk returns at steps of multiples of T. $R(i,T)$ being high indicates there are 1D cycles passing through node i.

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

Theorem 2.2. *(Peakness of Random Walk Cycles) $R(i,T)$ can be computed by the eigenvalues of transition matrix P:*

$$R(i,T) = \frac{\sum_j \mathrm{Re}(\frac{\lambda_j^T}{1-\lambda_j^T} \cdot U_{ij}V_{ij})}{\sum_j \mathrm{Re}(\frac{1}{1-\lambda_j} \cdot U_{ij}V_{ij})} \tag{2.21}$$

Proof. See Appendix. □



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

Figure 2.7: 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).

Theorem 2.2 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, $\mathrm{Pr}(i,T)$ 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 $\eta^k\mathrm{Pr}(i,k)$ to replace $\mathrm{Pr}(i,k)$. Responses with longer time are weighted lower because the peaks become more and more blurred. This amounts to replacing P by ηP and all the above analysis.

2.5 Tracing Contours

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 will introduce a discretization method and give the overall untangling cycle procedure in this section.

2.5.1 Discretization

For each of the top complex eigenvectors, we seek discrete topological cycles separated from the 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 top eigenvectors with small phase angles.

Once knowing the existence of a 1D cycle, we search for it in its complex eigenvector whose components are $v(1), ...v(2n)$. 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 the re-ordered embedding points $u(1) = v(s(1)), u(2) = v(s(2)), ..., u(h) = v(s(h))$ satisfy two criteria: 1) the magnitudes $|u(1)|, ..., |u(h)|$ are large and; 2) the phase angles $\theta(u(1)), ..., \theta(u(h))$ are in an increasing order. This can be tackled by finding the sequence enclosing the largest area in the complex plane:

$$\max_{s(1),...,s(h)} \sum_{j=1}^{h} A(u(j), u(j+1)) \tag{2.22}$$

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

To accelerate the search, we 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, be 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 (see Fig. 2.7). Notice that the sequence $u(1), ..., u(h), u(h+1) = u(1)$ 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 $\ell_{ij} = \frac{1}{2}[\theta(v(j)) - \theta(v(i))] \cdot R^2 - A(v(i), v(j))$. The number R is chosen sufficiently large to guarantee $\ell_{ij} > 0$ for all i,j. Then eq. (2.22) can be reduced to

$$\pi R^2 - \min_{s(1),\ldots,s(h+1)} \sum_{j=1}^{h} \ell_{s(j)s(j+1)} \tag{2.23}$$

This shortest cycle problem can be broken into two parts: the first shortest path from $s(1)$ in bin B_1 to a node $s(a)$ in bin B_2, and the second one from $s(a)$ back to $s(1)$. Hence, the second term $\min_{s(1),\ldots,s(h+1)} \sum_{j=1}^{h} \ell_{s(j)s(j+1)}$ in eq. (2.23) becomes

$$\min_{\substack{s(1)\in B_1, s(a)\in B_2 \\ s(1),\ldots,s(h+1)}} [\sum_{j=1}^{a-1} \ell_{s(j)s(j+1)} + \sum_{j=a}^{h} \ell_{s(j)s(j+1)}] \tag{2.24}$$

where each summation itself is a shortest path.

2.5.2 Untangling Cycle Algorithm

In summary, our untangled cycle algorithm has three steps:

Algorithm 1 (Untangling Cycle Algorithm)

```
1: GRAPH SETUP. Construct the directed graph $G$ and compute transition matrix $P$ by eq. (2.1) and (2.2).
2: COMPLEX EMBEDDING. Compute the first $n_c$ complex eigenvectors of $P$. Each complex eigenvector produces a complex circular embedding $v(1), v(2), ...v(2n) \in \mathbb{C}$.
3: CYCLE TRACING. For $v(1), v(2), ...v(2n)$, use shortest path to find a cycle $S \subseteq \{1, ..., 2n\}$ minimizing (eq. (2.23)).
```

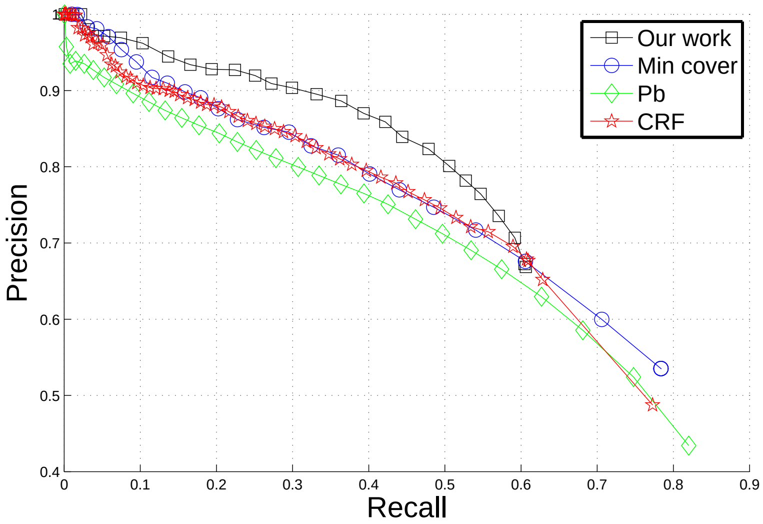


Figure 2.8: Precision recall curve on the Berkeley benchmark, with comparison to Pb, CRF and min cover. We use probability boundary with low threshold to produce graph nodes, and seek untangling 1D topological cycles for contour grouping. The same set of parameters are used to generate all the results.

2.6 Experiments

We tested our untangling cycle algorithm on a variety of challenging real images. The test datasets includes Berkeley Segmentation Dataset (Martin *et al.* , 2001) (see Fig. 2.9), Weizmann horse database (Borenstein & Ullman, 2002) (see Fig. 2.10), Berkeley baseball player dataset (Mori *et al.* , 2004a) (see Fig. 2.11), and ETHZ Shape Classes (Ferrari *et al.* , 2007b) in which we will utilize contours for shape detection in Chapter 3. Our untangling cycle algorithm is capable of extracting contours even when many of the images have significant clutter (see Fig. 2.9). We output all contours that are 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 post-processing. Extensive tests show that our algorithm is effective in discovering one-dimensional topological structures in real images.

The implementation details of the algorithm are explained as follows.

1. Graph Setup. The edgel graph is constructed by thresholding Pb at a low value (0.03) to ensure high recall. Other edge detectors can be applied as long as they output edge

tangents/normals. Graph weights are computed within a 21×21 neighborhood for each edgel. 10% of the weights is added to the reverse edges as backward connection W^{back} to close the open contours in topology. The graph matrix is normalized by column to generate a random walk matrix.

2. Complex Embedding. We compute 200 to 400 eigenvectors of the graph random walk matrix. The real eigenvectors are pruned because they contain no information on the contour ordering, as shown in Section 2.4. Eigenvalues whose phase angle is too large or whose magnitude is too small are also discarded. These indicates bad cycles with untangling cycle cut score. After eliminating one of the eigenvalue in each conjugate pair, typically less than 100 eigenvalues/eigenvectors survive.

3. Cycle Tracing. We run the shortest cycle algorithm eq. (2.22) on the embedding space generated by the remaining eigenvectors. Each complex embedding space is divided uniformly into 8 bins by phase angle. A cycle is broken into two shortest paths as in eq. (2.24): one from bin 1 to bin 2, and the other from bin 2 to bin 8 back to bin 1. We choose the top 5 cycles in each eigenvector, and combine the redundant ones. The final output contains partially overlapping contours due to multiple possibilities at junctions, instead of disjoint contours. These additional hypotheses are very important for constructing shapes in the next chapter.

The current unoptimized Matlab implementation takes about 3 minutes on a 300×400 image. The bottleneck of the computation is solving the complex eigenvectors. Similar to the eigenvalue problem in NCut, techniques of multi-scale graph (Cour *et al.* , 2005) or GPU implementation (Catanzaro *et al.* , 2009) can be explored to accelerate the computation in the future.

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 with two top contour grouping algorithms: CRF (Ren *et al.* , 2005b) and Min Cover (Felzenszwalb & McAllester, 2006). Our results are well above these approaches by about 7% in the medium to high precision part (see Fig. 2.8). Visually our results produce much cleaner contours as shown in Fig. 2.9-2.11. Many of the false positives are shading edges, which are not labelled by humans. However, once they are grouped, they could be easily to pruned in later recognition process. These are the advantages not reflected by the metric in the Berkeley benchmark, which counts matched pixels independently.

2.7 Summary

To our knowledge, this is the first major attack on contour grouping using a topological formulation. Our grouping criterion of untangling cycles exploits the inherent topological 1D 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.

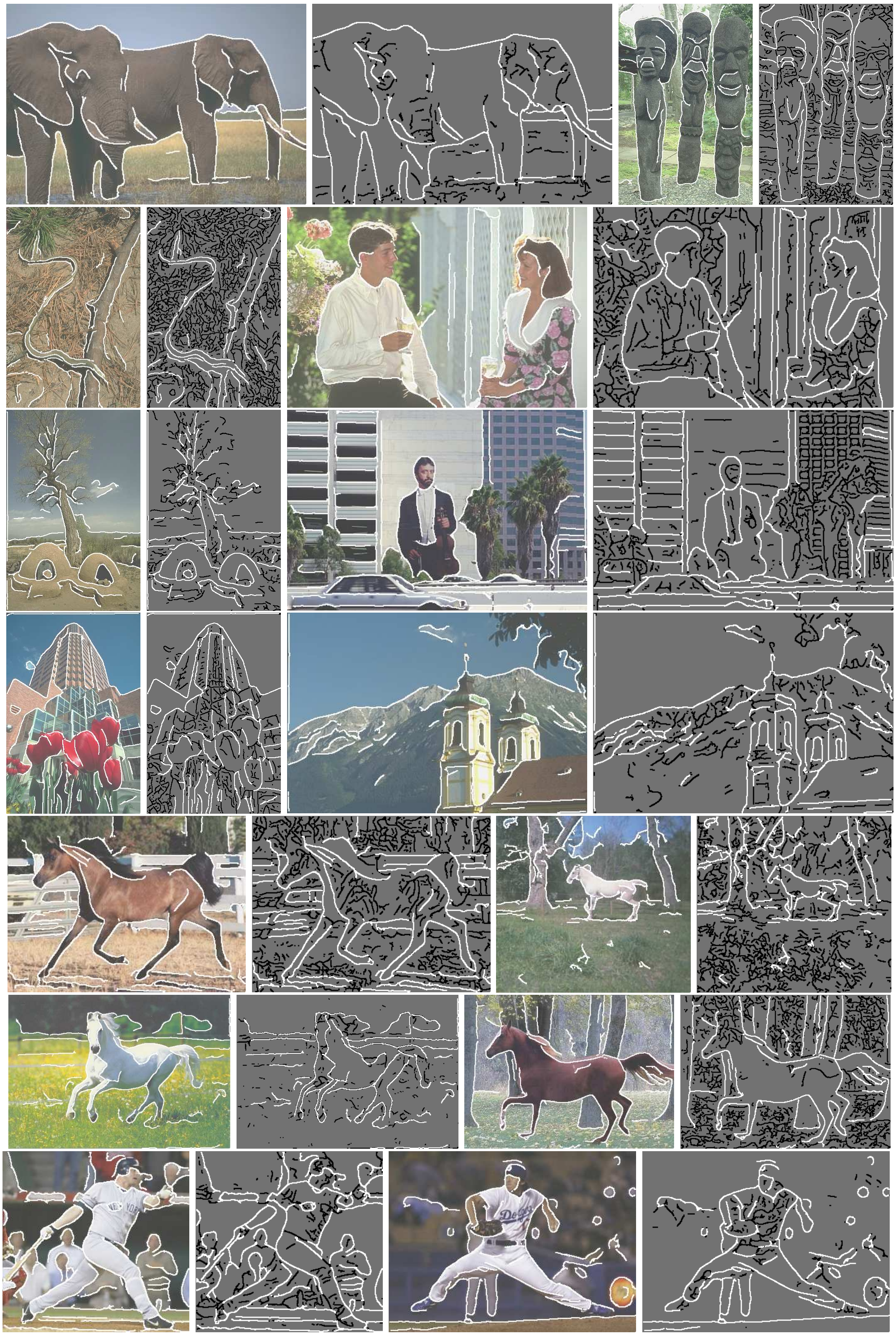


Figure 2.9: Contour grouping results on real images. Our method prunes clutter edges (dark), and groups salient contours (bright). We focus on graph topology, and detect contours that are either open or closed, straight or bended.

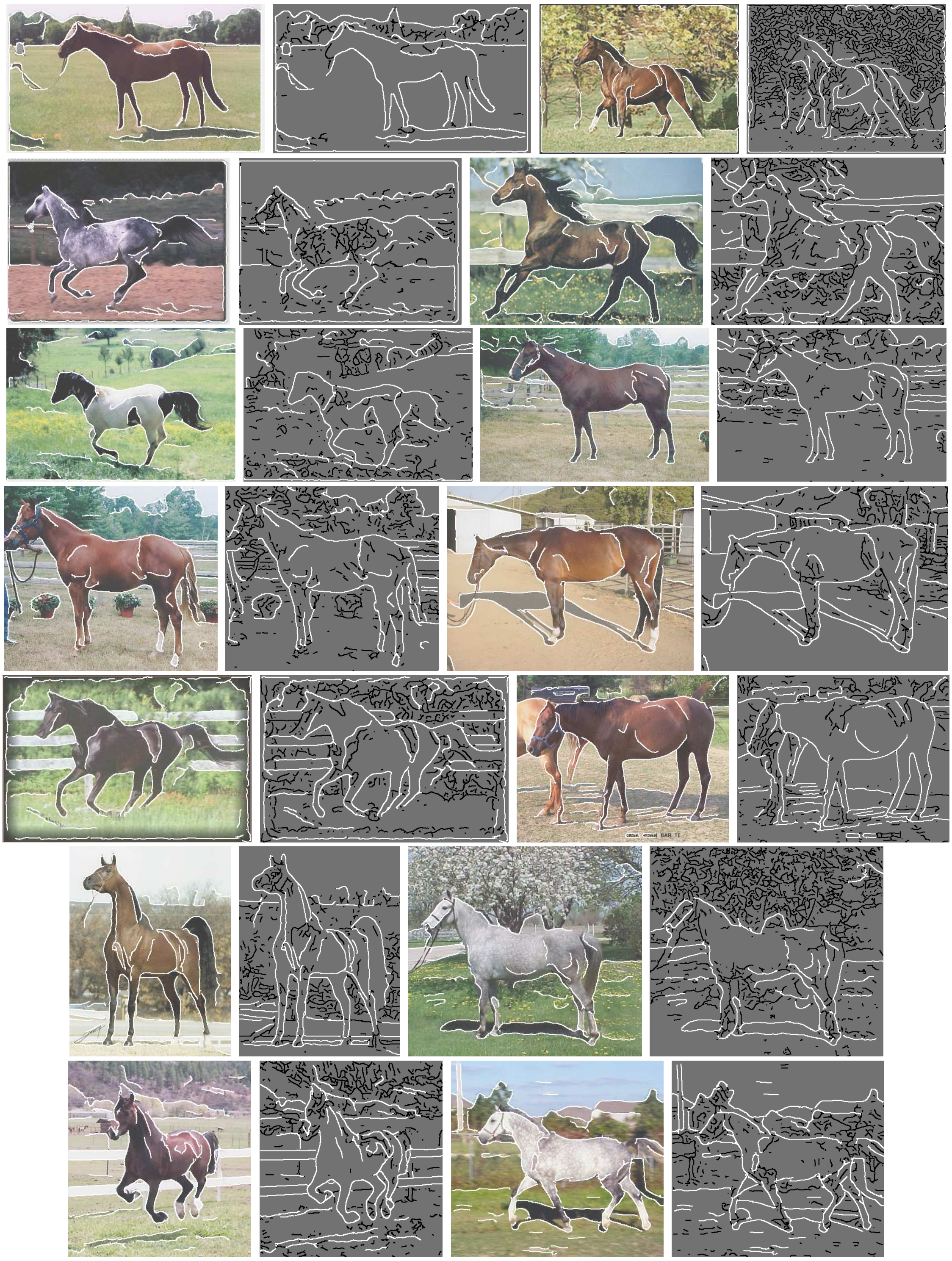


Figure 2.10: Contour grouping results on Weizmann horse database. All detected binary edges are shown (right). Our method prune clutter edges (dark), and groups salient contours (bright). We use no edge magnitude information for grouping, and can detect faint but salient contours under significant clutter. We focus on graph topology, and detect contours that are both open or closed, straight or bent.

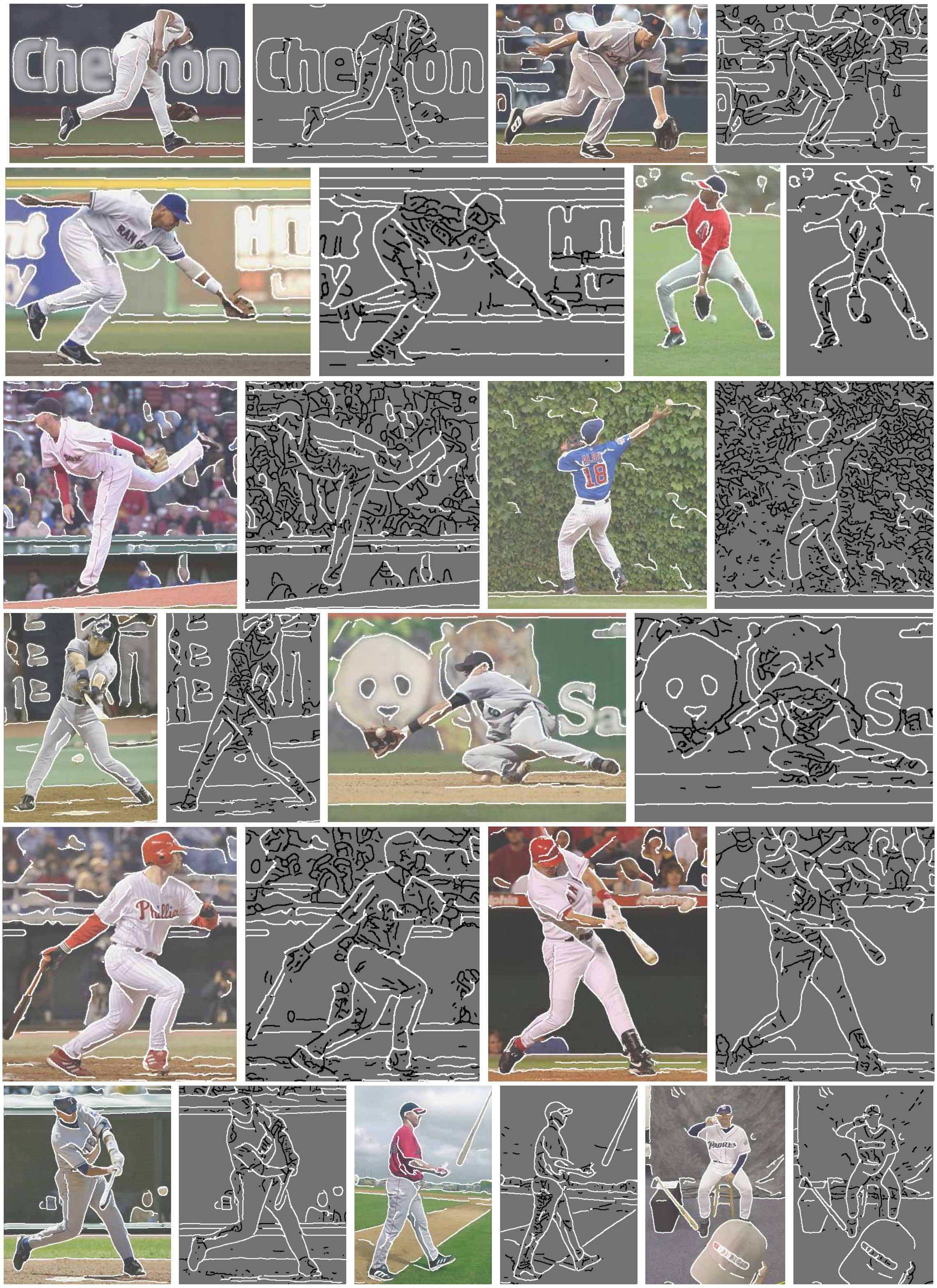


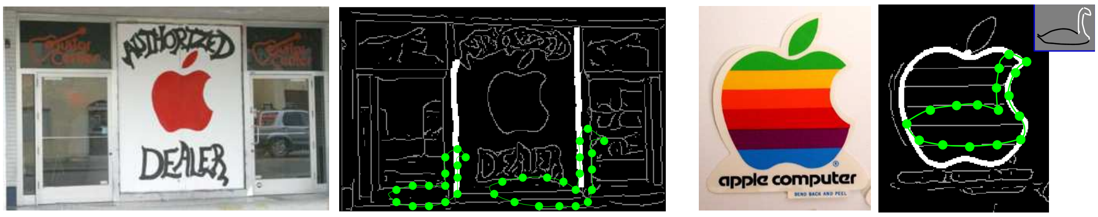
Figure 2.11: Contour grouping results on Berkeley baseball player dataset.

Chapter 3

Contour Packing

Visual objects can be represented in a variety of levels: from the signal level of filter responses to the symbolic level of object parts (Ullman, 1996). We focus on the representation based on shape that is closer to the symbolic level, allowing abstract geometric reasoning of objects. Shape-based object description is invariant to color, texture, and brightness changes, and dramatically reduces the number of training examples required, without sacrificing the detection accuracy.

This chapter presents the contour packing framework that holistically detects and matches a model shape by packing a set of image contours – an intermediate level of object representation. We build this framework on top of our contour grouping approach in Chapter 2, which suppresses 2D clutter and produces long topologically 1D contours. We develop a set-to-set contour matching formulation to bridge the representation gap between the image and the model due to unpredictable fragmentations of bottom-up contours. The global shape configuration of a contour set is characterized by *context selective shape features*, constructed from contours within a large spatial context. Unlike traditional shape features such as (Belongie *et al.* , 2002) which are precomputed regardless of context changes, context selective shape features adjust *on the fly* depending on which set of image contours participate in matching. The generated shape features can be encoded in a linear form of figure/ground contour selection. This enables the combinatorial search arising in set-to-set contour matching to be approximated and solved efficiently by an instance



(a) Accidental alignment

(b) Missing critical parts

Figure 3.1: Typical false positives can be traced to two causes: (1) Accidental alignment shown in (a). Our algorithm prunes it by exploiting contour integrity, *i.e.* requiring contours to be whole-in/whole-out. Contours violating this constraint is marked in white on the image. (2) Missing critical object parts indicates that the matching is a false positive. In (b), after removing the accidental alignment to the apple logo outline (marked in white), only the body can find possible matches and the neck of the swan is completely missing shown at the top-right corner of (b). Our approach rejects this type of detection by checking missing critical model contours after joint contour selection.

of Linear Programming (LP).

3.1 Overview

Detecting objects using shape alone is not an easy task. Most shape matching algorithms are susceptible to *accidental alignment*: hallucinating objects in the clutter by matching random edges (Amir & Lindenbaum, 1998). To avoid foreground clutter (*e.g.* surface marking on objects) and background clutter, shape descriptors are often computed within a window of a limited spatial extent. Local window features are discriminative enough for detecting objects such as faces, cars and bicycles. However, for many objects with simple shapes, such as swans, mugs or bottles, local features are insufficient.

To overcome the accidental alignment, our contour packing consists of the following three key ingredients:

1. **Contour integrity.** We detect salient contours using bottom-up contour grouping. Long contours themselves are more distinctive, and maintaining contours as integral tokens for matching eliminates many false positives due to accidental alignment to unrelated edges.

2. **Holistic shape matching.** We measure shape features from a large spatial extent, as well as long-range contextual relationships among object parts. Accidental alignment of holistic shape descriptors between image and model is unlikely.

3. **Model configuration checking.** We break the model shape into its informative semantic parts, and explicitly check which subset of model parts is matched. Missing critical model parts can signal an accidental alignment between the image and model.

We start with salient contours extracted by bottom-up contour grouping in Chapter 2. Shape matching with contours composed of orderly, grouped edges instead of isolated edges has several advantages. Long salient contours have more distinctive shapes, which leads to efficiency of the search as well as the accuracy of shape matching. Furthermore, by requiring the entire contour to be matched as a whole, we eliminate accidental alignment causing false positive detections shown in Fig. 3.1 (a). Using contour grouping as the starting point of shape matching carries risk as well. Contours could be mis-detected, or accidentally leaking to background. Therefore, a good contour grouping algorithm is essential for shape matching. We have demonstrated the good performance of our contour grouping algorithm in cluttered images. These contours are not disjoint, providing multiple hypotheses at junctions where contours can potentially leak to other objects.

The main technical challenge is that image and model contours do not have one-to-one correspondence. Contours detected from bottom-up grouping and segmentation are different from the semantically meaningful contours in the model. However, as a whole they will have a match (see Fig. 3.2). The holistic matching occurs only by considering a set of “figure” contours together. To formulate this set-to-set matching task, we introduce control points sampled on and around image and model contours. We compute shape features on the control points from the “figure” contours within a large neighborhood (see Fig. 3.2). The task boils down to finding the correct figure/ground contour selection, such that there is an optimal one-to-one matching of the control points. The set-to-set matching potentially requires searching over exponentially many choices of figure/ground selection on contours. We simplify this task by encoding the shape descriptor algebraically in a linear form of contour selection variables, allowing the efficient optimization technique of LP.

To evaluate shape matching, one needs to measure the accuracy of alignment, and more importantly, determine *which* model parts have actually been aligned. For simple shapes, missing a small but critical object part can indicate a complete mismatch (see Fig. 3.1 (b)). We manually divide the model into contours which corresponds to distinctive parts. Just as image contours, we require model contours to be whole-in or whole-out.

The rest of the chapter is organized as follows. Section 3.2 introduces the contour packing formulation and the key concept of context sensitive shape features. We present the computational solution for this framework using Linear Programming (LP) in Section 3.3. Section 3.4 describes related works and comparisons. Section 3.5 demonstrates our approach on the challenging task of detecting non-rectangular and wiry shaped objects, followed by the conclusion in Section 3.6.

3.2 Set-to-Set Contour Matching

In this section we develop the set-to-set contour matching method. The computational task of set-to-set contour matching consists of parallel searches over image contours and model contours to obtain the maximal match of the image and model shapes.

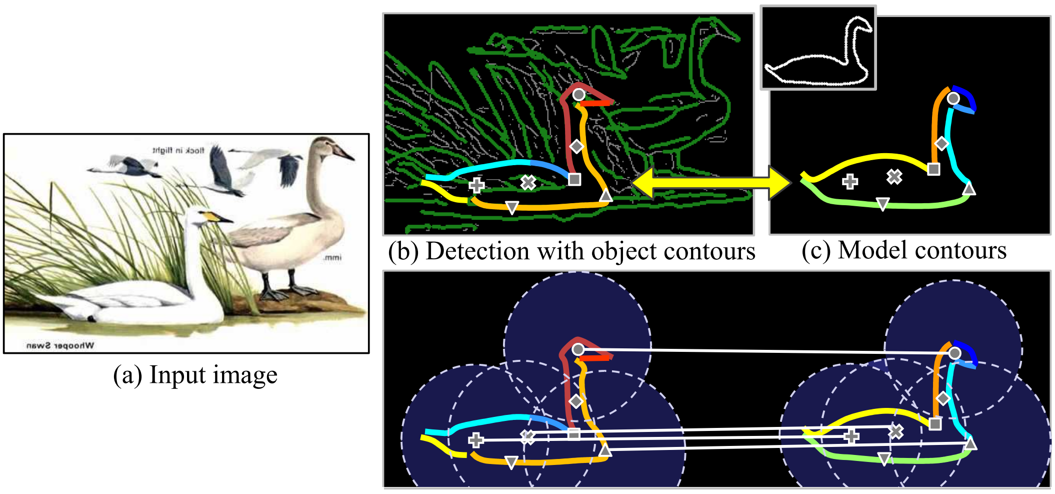
3.2.1 Problem Formulation

We start with formulating the shape detection as the following problem:

Definition of set-to-set contour matching. Given an image $\mathcal{I}$ and a model $\mathcal{M}$ represented by two sets of contours:

- Image: $\mathcal{I} = \{C_1^I, C_2^I, \dots C_{|\mathcal{I}|}^I\}$, C_k^I is the k^{th} contour;
- Model: $\mathcal{M} = \{C_1^M, C_2^M, \dots, C_{|\mathcal{M}|}^M\}$, C_l^M is the l^{th} contour.

we would like to select the maximal contour subsets $\mathcal{I}^{sel} \subseteq \mathcal{I}$ and $\mathcal{M}^{sel} \subseteq \mathcal{M}$, such that object shapes composed by $\mathcal{I}^{sel}$ and $\mathcal{M}^{sel}$ match (see Fig. 3.2 for an image example).



(a) Input image

(b) Detection with object contours

(c) Model contours

(d) Control point correspondence

Figure 3.2: Using a single line drawing object model shown in (c), we detect object instances in images with background clutter in (a) using shape. Bottom-up contour grouping provides tokens of shape matching. Long salient contours in (b) can generate distinctive shape descriptions, allowing both efficient and accurate matching. Image and model contours, shown by different colors in (b) and (c), do not have one-to-one correspondences. We formulate shape detection as a set-to-set matching task in (d) consisting of: (1) correspondences between control points, and (2) selection of contours that contribute contextual shape features to those control points, within a disk neighborhood.

Matching constraint: contour integrity. The above formulation implies that each contour is restricted to be an integral unit in matching. For each contour $C_k^I = \{p_1^{(k)}, p_2^{(k)}, ..., p_c^{(k)}\}$ where $p_i^{(k)}$'s are edge points, there are only two choices: either all the edge points $p_i^{(k)}$ participate in the matching, or none of them are included. Partially matched contours are not allowed. The same constraint applies to model contours in $\mathcal{M}$ as well. We introduce contour selection indicators $x^{sel} \in \{0,1\}^{|I| \times 1}$ in the entire test image and $y^{sel} \in \{0,1\}^{|M| \times 1}$

in the model defined as

$$\text{(Image Contour Selector)} \qquad x_\ell^{sel} = \begin{cases} 1, & \text{if contour } C_\ell^I \text{ is selected} \\ 0, & \text{otherwise.} \end{cases} \tag{3.1}$$

$$\text{(Model Contour Selector)} \qquad y_\ell^{sel} = \begin{cases} 1, & \text{if contour } C_\ell^M \text{ is selected} \\ 0, & \text{otherwise.} \end{cases} \tag{3.2}$$

Control point correspondence. While contours themselves do not correspond one-to-one, their overall shape configuration can be evaluated at nearby control points, and those control points do have one-to-one correspondences. Suppose control points $\{p_1, p_2, \ldots, p_m\}$ are sampled from the image and $\{q_1, q_2, \ldots, q_n\}$ are sampled from the model. We define the correspondence matrix $(U^{cor})_{m\times n}$ from the image to the model as:

$$U_{ij}^{cor} = \begin{cases} 1, & \text{if } p_i \text{ matches } q_j \\ 0, & \text{otherwise.} \end{cases} \tag{3.3}$$

Note that these control points can be located anywhere in the image, not limited to contour points. Computing dense point correspondences is unnecessary. Instead, rough matching of a few control points is sufficient to select and match contour sets $\mathcal{I}^{sel}$ and $\mathcal{M}^{sel}$.

Feature representation: holistic shape features. The important question is, what will be the appropriate shape feature for matching these control points, and how to compute shape dissimilarity/distance D_{ij}. In order to be matched, the shape feature has to share a common description between the image and the model. Since there do not exist one-to-one correspondences between contours, the feature description is more appropriate on the contour set or global shape level rather than on the individual contour level. We propose a holistic shape representation at the control points covering not only nearby contours but also faraway contours (see Fig. 3.3).

The holistic shape representation immediately poses the problem of *figure/ground selection* since figure/ground segmentation is unknown and the shape feature is likely to include both foreground and background contours. Without the correct segmentation, background clutter and contours from other objects can corrupt the shape feature. This poses great difficulties to any shape features with a fixed context. A fixed context feature cannot adapt to the combinatorial possibilities of figure/ground selection, with each generating a different feature. Our strategy is to adjust the context of the holistic shape features during matching depending on the figure/ground selection. Therefore, we are able to compute the right features and determine the figure/ground segmentation simultaneously.

3.2.2 Context Selective Shape Features

We are ready to introduce the holistic shape representation called context selective shape features determined by the figure/ground selection of the contours x^{sel} and y^{sel}. We choose Shape Contexts (SC) (Belongie *et al.* , 2002) as the basic shape feature descriptor. Measuring global shape requires the scope of SC to be large enough to cover the entire object. Define $sc_i^I = [sc_i^I(1), sc_i^I(2), ..., sc_i^I(b)]^{\mathrm{T}}$ to be the vector of SC histogram centered at control point p_i, *i.e.* $sc_i^I(k) = \#$ of points in bin k. We introduce a contribution matrix V_i^I with size (#bin)×(#contour) to encode the contribution of each contour to each bin of sc_i^I:

$$V_i^I(k,l) = \# \text{ of points in bin } k \text{ from contour } C_l \tag{3.4}$$

Similar notations sc_j^M and V_j^M are defined for SC at control point q_j in the model.

The key observation is that shape features sc_i^I will be *different* depending on context x^{sel}, *i.e.* they are not fixed. Since each contour can have 2 choices, either selected or not selected, there exists 2^n possible contexts – exponential in the number of contours n. One advantage of histogram features such as SC is that the exponentially many combinations of contexts can be written in a simple linear form:

$$sc_i^I(k) = \sum_l V_i^I(k,l) \cdot x_l^{sel} = (V^I \cdot x^{sel})_k \tag{3.5}$$

This allows us to cast the complex search as an optimization problem later.

Our goal is to find x^{sel} and y^{sel} such that they produce similar shape features: $V_i^I \cdot x^{sel} \approx V_j^M \cdot y^{sel}$. We evaluate and compare these two features by the context sensitive

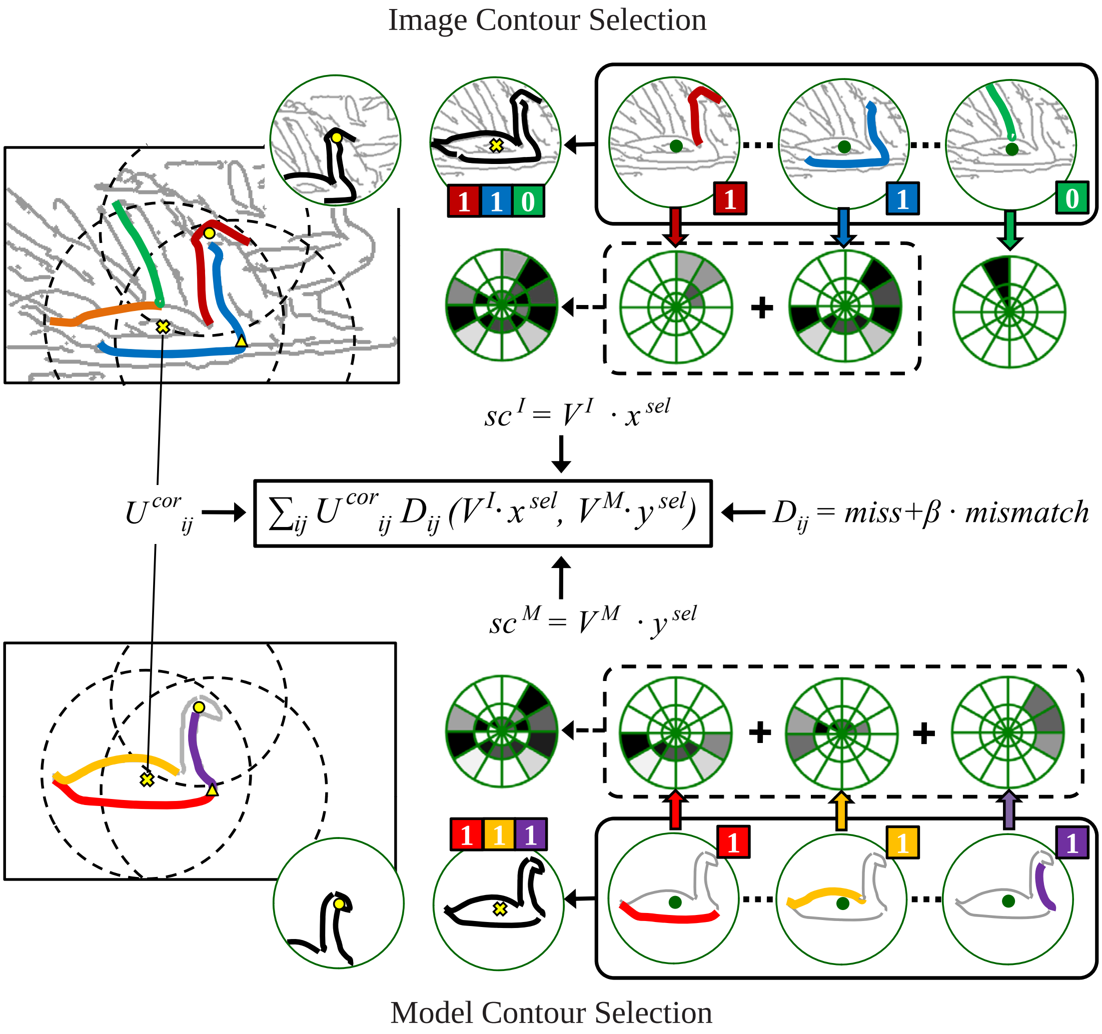


Image Contour Selection
1 1 0
1
1
0
+
$sc^I = V^I \cdot x^{sel}$
U^{cor}_{ij}
$\sum_{ij} U^{cor}_{ij} D_{ij} (V^I \cdot x^{sel}, V^M \cdot y^{sel})$
$D_{ij} = miss + \beta \cdot mismatch$
$sc^M = V^M \cdot y^{sel}$
+
+
1 1 1
1
1
1
Model Contour Selection

Figure 3.3: Illustration of our computational solution for set-to-set contour matching on shape detection example from Fig. 3.2. The top and the bottom row shows the image and model contour candidate sets marked in gray. Each contour contributes its shape information to nearby *control points* in the form of Shape Context histogram, shown on the right. By selecting different contours (x^{sel}, y^{sel}), each control point can take on a set of possible Shape Context descriptions (sc^I, sc^M). With the correct contour selection in the image and model (marked by colors), there is a one-to-one correspondence U^{cor}_{ij} between (a subset of) image and model control points (marked by symbols). This is a computationally difficult search problem. The efficient algorithm we developed is based on an encoding of Shape Context description (which could take on exponentially many possible values) using linear algebraic formulation on the contour selection indicator: $sc^I = V^I \cdot x^{sel}$. This leads to the LP optimization solution.

dissimilarity:

$$\text{(SHAPE DISSIMILARITY)} \qquad D_{ij}(sc_i^I, sc_j^M) = D_{ij}(V_i^I \cdot x^{sel}, V_j^M \cdot y^{sel}) \tag{3.6}$$

The shape dissimilarity D_{ij} not only depends on the local attributes of p_i and q_j, but more importantly, on the context given by x^{sel} and y^{sel}. Matching object shapes boils down to minimizing D_{ij}, which is a combinatorial search problem on x^{sel} and y^{sel}.

3.2.3 Contour Packing Cost

Finding the set-to-set contour matching finally becomes a joint search over correspondences U^{cor} and contour selection x^{sel}, y^{sel} by minimizing the following cost:

(Contour Packing Cost)

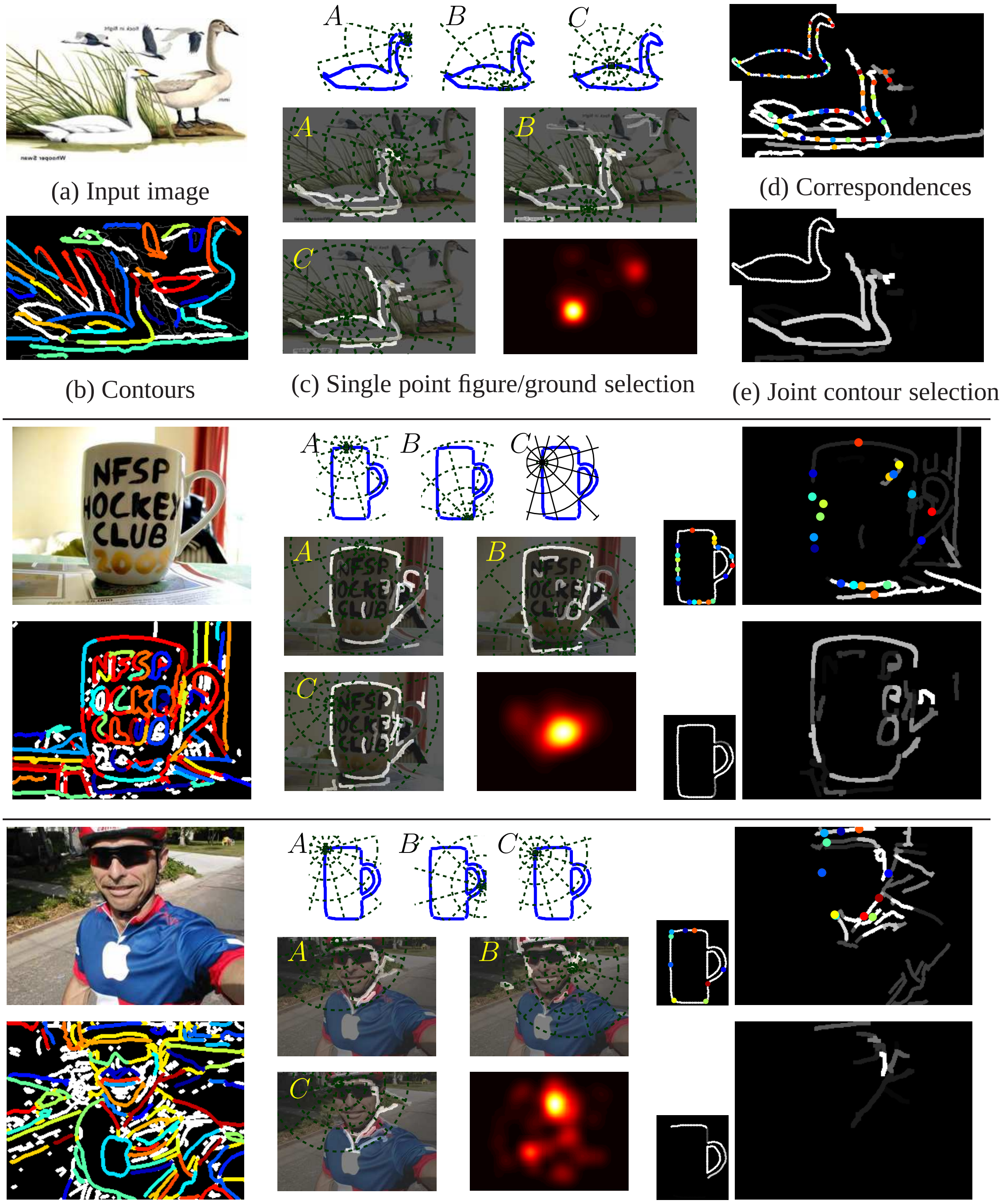
$$\min_{U^{cor}, x^{sel}, y^{sel}} \quad C_{packing}(U^{cor}, x^{sel}, y^{sel}) = \frac{1}{m} \sum_{i,j} U_{ij}^{cor} D_{ij}(V^I x^{sel}, V^M y^{sel}) \tag{3.7}$$

$$\text{s.t.} \quad U^{cor} \in G$$

where $m = \sum_{i,j} U_{ij}^{cor}$ is the number of control point correspondences. Correspondences U^{cor} from different object parts should have geometric consistency. We use a star model graph for checking global geometric consistency. Each correspondence (p_i, q_j) can predict an object center c_{ij}. For the correct set of correspondences, all the predicted centers should form a cluster, *i.e.* close to their average center: $c(U^{cor}) = \sum c_{ij} U_{ij}^{cor} w_{ij} / \sum U_{ij}^{cor} w_{ij}$, where w_{ij}'s are the weights on correspondences. Thus correspondences U^{cor} satisfying the geometric consistency constraint can be expressed as:

$$\text{(GEOMETRIC CONSISTENCY)} \quad G = \{\|c(U^{cor}) - c_{ij} U_{ij}^{cor}\| \leq d_{max} \text{ if } U_{ij}^{cor} = 1\} \tag{3.8}$$

where d_{max} is the maximum distance allowed for deviation from the center.


(a) Input image

(b) Contours

(c) Single point figure/ground selection

(d) Correspondences

(e) Joint contour selection

Figure 3.4: Illustration of contour packing for shape detection. From input image (a), we detect long salient contours shown in (b). For each control point correspondence in (c), we select foreground contours whose global shape is most similar to the model, with selection x^{sel} shown in gray scale (the brighter, the larger x^{sel}). Voting maps in (c) prune geometrically inconsistent correspondences. (d) shows the consistent correspondences marked by different colors. The optimal joint contour selection is shown in (e). Note in the last example, model selection allows us to detect false match on the face.

3.3 Computational Solution via Linear Programming

Direct optimization of contour packing cost function eq. (3.7) is a hard combinatorial search problem. The shape dissimilarity $D_{ij}(V^I \cdot x^{sel}, V^M \cdot y^{sel})$ can only be evaluated given correspondences U^{cor}. However, finding the correct correspondences U^{cor} requires x^{sel} and y^{sel}. Therefore, the inference problem becomes circular. We approximate this joint optimization by breaking the loop into two steps: *single point figure/ground selection* and *joint contour selection* (see Fig. 3.4). The first step focuses on finding reliable correspondences U^{cor} (maybe sparse) by matching image contours to the whole model. Note that even this subroutine is a combinatorial search, with exponentially many combinations of figure/ground selection. The second step selects contours simultaneously from both image contours labelled as figure and all the model contours being matched, based on the correspondences computed in the first step. This section presents the relaxation of both steps as an instance of Linear Programming (LP).

3.3.1 Single Point Figure/Ground Selection

Our first step discovers all potential control point correspondences U_{ij} and computes the corresponding figure/ground selection x^{sel} for them. We fix $y^{sel} = 1$ to encourage matching to the full model as much as possible. In this step, partial matches are undesired since the correspondences they produce are much less reliable. We use the simple L_1-norm as the dissimilarity D_{ij}. Accordingly, the contour packing cost eq. (3.16) reduces the the following problem:

$$\min_{x^{sel}} \ \|V^{\mathcal{I}} \cdot x^{sel} - V^{\mathcal{M}} \cdot y^{sel}\|_1, \ x^{sel} \in \{0,1\}^{|\mathcal{I}|} \tag{3.9}$$

A brute force approach of the above problem is formidable even for mid-size problems with 20 to 30 contours. We compute an approximate solution by relaxing the binary variables x^{sel} to continuous values: $0 \le x^{sel} \le 1$. Since the norm in the cost function is L_1[1].

[1] Besides L_1, other distance functions such as L_2 and χ^2 for shape context can also be used. However, the relaxations will be computationally much more intensive. We will see discussion on L_2 in later this section and Appendix.

By introducing slack variables $b^+, b^- \geq 0$ such that $V^{\mathcal{I}} \cdot x^{sel} - V^{\mathcal{M}} \cdot y^{sel} = b^+ - b^-$, we can reduce the problem to a standard LP:

$$\text{(CONTOUR PACKING LP)} \qquad \begin{aligned} \min_{x^{sel}, b^+, b^-} \quad & 1^{\mathrm{T}} b^+ + 1^{\mathrm{T}} b^- \\ \text{s.t.} \quad & V^{\mathcal{I}} x^{sel} - V^{\mathcal{M}} y^{sel} = b^+ - b^- \\ & 0 \leq x^{sel} \leq 1 \\ & b^+, b^- \geq 0 \end{aligned} \tag{3.10}$$

This LP problem can be solved efficiently by off-the-self LP solvers such as Mosek (Andersen & Andersen, 2000). We will see even more efficient solutions using primal-dual algorithms in the next chapter.

L_2-norm Dissimilarity: A MaxCut Approach

The choice of shape dissimilarity D_{ij} has a significant impact on solving the combinatorial problem of contour packing. One alternative to the L_1-norm used in eq. (3.9) is to have L_2-norm: $\|V^{\mathcal{I}} \cdot x^{sel} - V^{\mathcal{M}} \cdot y^{sel}\|^2$. We have discovered that this can be reduced to MaxCut, with a proved bound on approximation via Semidefinite Programming (SDP) (Goemans & Williamson, 1995). The derivation of this connection is summarized in following theorem:

Theorem 3.1. *Construct a graph $G_{packing} = (V, E, W)$ with $V = \mathcal{I} \cup \mathcal{M} \cup \mathcal{A}$ and $w_{ij} = a_i^T a_j$, where*

$$a_i = \begin{cases} V^I_{(:,i)} & if \text{ node } i \in \mathcal{I} \\ V^M_{(:,i)} & if \text{ node } i \in \mathcal{M} \\ (0, ..., 0, |\sum_k V^I_{ik} - \sum_k V^M_{ik}|, 0, ..., 0)^T & if \text{ node } i \in \mathcal{A} \end{cases} \tag{3.11}$$

Here $V^I(k, i)$ is the feature contribution of contour i to the histogram bin k defined in eq. (3.4). Vectors $V^I_{(:,i)}$ and $V^M_{(:,i)}$ represents the i^{th} columns of V^I and V^M.

The optimal subset S^I_* and S^M_* with the best matching cost $\|V^{\mathcal{I}} \cdot x^{sel} - V^{\mathcal{M}} \cdot y^{sel}\|^2$ is given by the maximum cut of the graph $G_{packing}$. If (C_1, C_2) is the cut with $V_0 \in C_2$, the optimal subsets are given by $S^I_* = \mathcal{I} \cap C_1$ and $S^M_* = \mathcal{M} \cap C_2$.

Proof. Please see Appendix. □

Although the relaxation of SDP provides a tighter approximation in theory, L_2-norm is not as good L_1-norm as a distance function for feature description. L_2-norm is susceptible to large values in the histogram bins, and hence less robust to image outliers and noises. Therefore, the L_1-norm dissimilarity and the LP relaxation is adopted in the subsequent sections. We will revisit the SDP relaxation in Chapter 6, which provides additional expressive power for region packing.

Correspondences found from single point figure/ground selection might not satisfy geometric consistency eq. (3.8). Therefore, we enforce geometric consistency by pruning hypotheses of control point correspondences via a voting procedure (Wang *et al.* , 2007). Each image control point can predict an object center using its best match to model control points computed by eq. (3.9). These predictions generate votes weighted by the shape dissimilarity, and accumulates to a voting map. We extract object centers from the local maxima and further back-trace the voters to identify geometrically consistent correspondences.

3.3.2 Joint Contour Selection

Once obtaining a group of geometrically consistent correspondences, we seek a subset of contours that match well consistently across all correspondences in eq. (3.7). In single point figure/ground selection, the selected contours at different control points are not guaranteed to be the same. The shape feature centered at each control point essentially covers the whole object. However, the sensitivity of shape description differs: close-by shape descriptions are more precise to be discriminative, and the faraway ones are more blurry to tolerate deformations. A unification of these descriptions from different control points can generate an overview of the shape without losing the details. Given a list of control point correspondences $(i(1), j(1)), (i(2), j(2)), ..., (i(k), j(k))$ where $U^{cor}_{i(s),j(s)} = 1$, we can stack the all the contribution matrices for image contours into one matrix, and similarly for the model side:

$$V^I = \begin{pmatrix} V^I_{i(1)} \\ V^I_{i(2)} \\ \vdots \\ V^I_{i(k)} \end{pmatrix}, \;\; V^M = \begin{pmatrix} V^I_{j(1)} \\ V^I_{j(2)} \\ \vdots \\ V^I_{j(k)} \end{pmatrix} \tag{3.12}$$

The contour packing cost in eq. (3.7) can be written in the following matrix form:

$$\sum_{s=1}^{k} \|V^I_{i(s)} \cdot x^{sel} - V^M_{j(s)} \cdot y^{sel}\|_1 = \|V^I \cdot x^{sel} - V^M \cdot y^{sel}\|_1 \tag{3.13}$$

Note that this is an optimization problem with exactly the same form as eq. (3.9). Therefore the previous LP-based computational solution applies directly.

Maximal matching cost. Recall that our problem is to search for the maximal common subsets from the image and model contours such that their shapes are similar. *What is the right matching cost $D_{ij}(V^I_i \cdot x^{sel}, V^M_j \cdot y^{sel})$ that can enforce the maximal condition?*. A straightforward cost function, such as the L_1-norm used previously: $D_{ij}(V^I_i \cdot x^{sel}, V^M_j \cdot y^{sel}) = \|V^I_i \cdot x^{sel} - V^M_j \cdot y^{sel}\|$, will simply result in the trivial solution which chooses empty sets from both sides (*i.e.* $x^{sel} = \mathbf{0}$, $y^{sel} = \mathbf{0}$). In fact all the norms as well as χ^2 distance will suffer from the same problem. We introduces the *maximal matching cost* for D_{ij} which balances the maximal requirement on the contour selection and the quality of the match. We seek to match as many model contours as possible while the difference between image and model contours is small. Before describing the details, we first introduce a few variables. Set

- $sc^{\mathcal{MF}}_j = V^M_j y^{full}$ to be the shape context centered at model point q_j selecting the full model, where $y^{full} = \mathbf{1}_{|\mathcal{M}|}$ means selecting all model contours;

- $sc^I_i = V^I_i x^{sel}$ to be the shape context with selection x^{sel} on image at p_i;

- $sc^M_j = V^M_j y^{sel}$ to be the shape context with selection y^{sel} on model at q_j.

We use $sc^{\mathcal{MF}}_j(k)$, $sc^I_i(k)$, $sc^M_j(k)$ to denote the k^{th} bin in the shape context.

Our maximal matching cost consists of two terms: *miss* and *mismatch* (see Fig. 3.3). To match as many model contours as possible, the following difference between the number of matched points and that of full model points should be minimized:

$$\mathrm{miss}_k^{(ij)} = sc_j^{\mathcal{MF}}(k) - \min(sc_i^I(k), sc_j^M(k)) \tag{3.14}$$

Here $\min(sc_i^I(k), sc_j^M(k))$ counts the number of matched contour points between the image and model in shape context bin k.

The above term $\mathrm{miss}_k^{(ij)}$ alone does not measure how well the selected image contours match to the selected model contours. To ensure the matching quality, the amount of difference between the number of image and model contour points in all shape context bins needs to be minimized:

$$\mathrm{mismatch}_k^{(ij)} = |sc_i^I(k) - sc_j^M(k)| \tag{3.15}$$

By combining eq. (3.14) and eq. (3.15), we have the following dissimilarity:

$$D_{ij} = \frac{\sum_k [\mathrm{miss}_k^{(ij)} + \beta \cdot \mathrm{mismatch}_k^{(ij)}]}{\sum_k sc_j^{\mathcal{MF}}(k)} \tag{3.16}$$

where $\beta > 1$ is a factor balancing the two types of costs. We use $\sum_k sc_j^{\mathcal{MF}}(k)$ to normalize the cost D_{ij} such that it is invariant to the number of contour points.

LP can also be used to solve eq. (3.7) for contour context selection by relaxing x^{sel} and y^{sel} to real value vectors. eq. (3.16) and eq. (3.7) translate to the following problem:

$$\begin{aligned} \min_{x^{sel}, y^{sel}} & \sum_{U_{ij}^{cor}=1} \{ \frac{1}{N_i} \sum_k [sc_i^{\mathcal{MF}}(k) - \min(sc_i^I(k), sc_j^M(k))] + \frac{\beta}{N_i} \| sc_i^I - sc_j^M \|_1 \} \\ \text{s.t. } & sc_i^I = V_i^I \cdot x^{sel}, \;\; sc_j^M = V_j^M \cdot y^{sel} \end{aligned}$$

where $N_i = \sum_k sc_i^{\mathcal{MF}}(k)$ is a normalization constant and $\min(x, y)$ computes the element-wise min of vectors x and y. The two terms in the summation are miss and $\mathrm{mismatch}$ in eq. (3.16) respectively. The above problem can be relaxed to an instance of LP by adding slack variables $s_{ijk} \geq sc_i^I(k)$ and $s_{ijk} \geq sc_j^M(k)$ for $\min(sc_i^I(k), sc_j^M(k))$.

We have obtained the rough correspondences U^{cor} from the previous step. We optimize the contour selection cost eq. (3.7) *w.r.t.* x^{sel}, y^{sel} to prune false positives and detect objects. The outcome includes both the matching cost $C_{packing}$ and model contours actually matched, indicated by y^{sel}. Both of them can be used to prune false positives. Note that it is not required to have a complete correspondence set U^{cor} since the cost eq. (3.16) has been normalized by the number of correspondences.

Model configuration checking. The selected model contours from joint contour selection form a shape configuration that are actually matched to image contours. Because the number of object model contours is typically very limited (usually 6 to 8), we can specify a dictionary of all possible configurations of true positives. Detection of model contours with bad configurations, e.g. missing critical parts, are rejected. This configuration checking together with the matching cost $C_{packing}$ can prune most of the false positives while preserving true positives. The last row in Fig. 3.4 shows such a case.

3.4 Related Work and Discussion

Salient contours and their configurations are more distinctive than individual edge points for shape matching. The works (Ferrari *et al.* , 2007b; Ferrari *et al.* , 2007a) represent objects by learning a codebook of Pairs of Adjacent Segments, which are consecutive roughly straight contour fragments. They achieve detection using a bag-of-words approach. In (Shotton *et al.* , 2005), boosted contour-based shape features are learned for object detection. These efforts utilize mostly short contour fragments, and therefore have to rely on many training examples to boost the discriminative power of shape features. In contrast, our work takes the advantage of contour grouping such as (Zhu *et al.* , 2007) to detect long salient contours, capturing more global geometric information of objects. More importantly, we constrain these long contours to act as a whole unit, *i.e.* they can either be entirely matched to an object, or entirely belong to the background. This characteristic makes shape matching more immune to accidental alignment to background clutter. Similar properties are exploited by grouping-based verification approaches (Amir & Lindenbaum, 1998), and the recent work (Felzenszwalb & Schwartz, 2007).

From a broader perspective, our recognition framework is based on shape matching,

	Applelogos	Bottles	Giraffes	Mugs	Swans
Contour Packing	**49.3%/86.4%**	**65.4%/92.7%**	**69.3%/70.3%**	25.7%/83.4%	**31.3%/93.9%**
Ferrari *et al.*., 07	32.6%/86.4%	33.3%/92.7%	43.9%/70.3%	**40.9%/83.4%**	23.3%/93.9%

Table 3.1: Comparison of Precision/Recall (P/R). We compare the precision of our approach to the precision in (Ferrari *et al.*., 2007) at the same recall (lower recall in (Ferrari *et al.*., 2007)). We convert the result of (Ferrari *et al.*., 2007) reported in DR/FPPI into P/R since the number of images in each class is known. Our performance is significantly better than (Ferrari *et al.*., 2007) in four out of five classes. The other class ”Mugs” have some instances that are too small to be detected by contour grouping. Note that we did not use magnitude information which plays an important role in (Ferrari *et al.*., 2007).

which has a long history in vision. A large amount of research has been done on different levels of shape information. Early works (Zahn & Roskies, 1972; Gdalyahu & Weinshall, 1999) focused on silhouettes which are relatively simple for representing shape. Silhouette-based approaches are limited to objects with a single closed contour without any interior edges with occlusions. Objects in real images are more complex, and may be embedded in heavy clutter. Efforts on dense matching of the edge points often focus on spatial configurations of key points, such as geometric hashing (Lamdan *et al.* , 1990), decision tree (Amit & Wilder, 1997) and Active Shape Models (Cootes *et al.* , 1995). However, key-points alone are insufficient to distinguish objects shapes in cluttered images (Belongie *et al.* , 2002).

Feature representation and shape similarity measurement are the key issues for matching. Shape Context (Belongie *et al.* , 2002) uses spatial distribution of edges points relative to a given point to describe shape. Inner Distance Shape Context (IDSC) refines it to account for articulated objects (Ling & Jacobs, 2005). We build our basic shape feature representation on Shape Context, with contour as the unit. A much larger context window covering the whole object enables our approach to capture global shape configurations. We introduce a novel contour selection mechanism to extract global shape features against background clutter.

3.5 Experiments

We demonstrate our detection approach using only one hand-drawn model without negative training images, To evaluate our performance, we choose the challenging ETHZ Shape Classes (Ferrari *et al.* , 2007a) containing five diverse object categories with 255 images in total. Each image has one or more object instances. All categories have significant scale variances, illumination changes and intra-class variations. Moreover, many objects are surrounded by extensive background clutter and have interior surface markings. We have the same experimental setup as (Ferrari *et al.* , 2007a), using only a single hand-drawn model for each class and all 255 images as test set. To adapt to large scale variance, we resize the model in 5 to 8 scales with a ratio step of 1.3 for each class.

We first use contour grouping developed in Chapter 2 to generate long salient contours from images. Contours can have overlaps due to multiple possible groupings at junctions. The Shape Context (SC) used for contour selection covers the entire model shape with a large spatial extent. The SC histogram has 12 polar bins, 5 radial bins and 4 edge orientation bins. To tolerate shape deformation and eliminate the border artifact of histogram binning, bin counts are blurred as in (Wang *et al.* , 2007). This refinement can be encoded into contribution matrices V^I, V^M as well.

We sample control point hypotheses on image contours uniformly with an interval equal to $1/10$ of the model bounding box diagonal. The number of image control points sampled in each scale ranges from 50 to 400. The numbers of model control points vary from 15 to 30 depending on the complexity of the target shape. LPs arising from single point figure/ground selection as well as joint contour selection are solved by the interior point method. The computation time for each hypothesized correspondence in single point figure/ground selection is within 0.1 sec.

After selecting figure contours, each correspondence votes for the object center with the weight inversely proportional to the shape matching cost. We collect the votes into a voting map and extract its local maxima above certain threshold to generate object hypotheses. Since the correct object scale is unknown beforehand, voting is performed in a

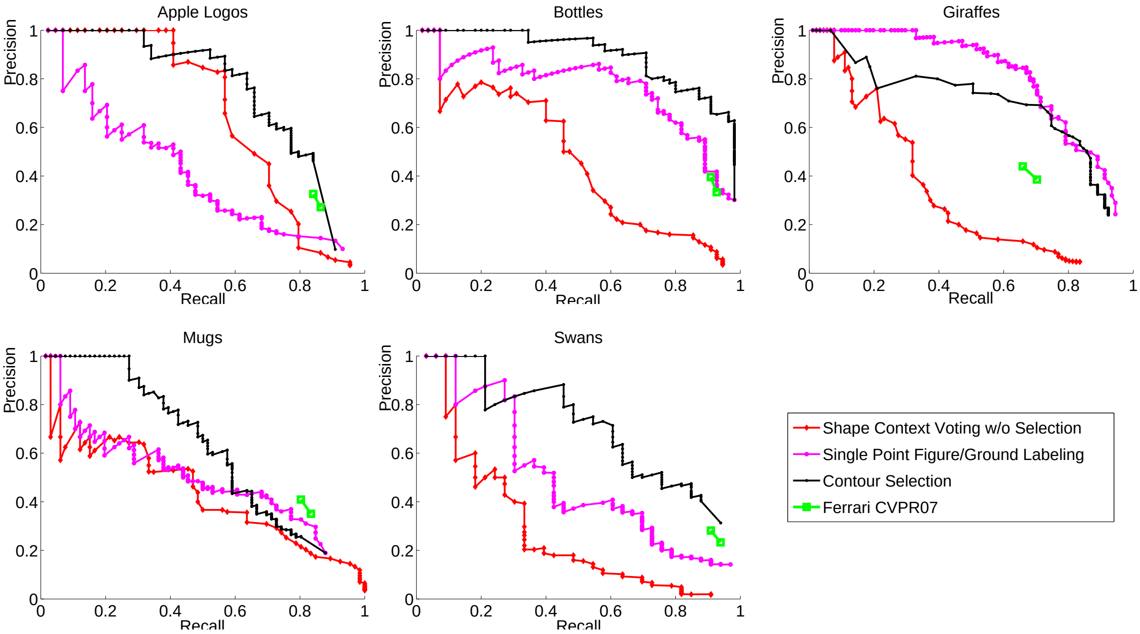


Figure 3.5: Precision vs. recall curves on five classes of ETHZ Shape Classes. Our precisions on ”Applelogos”, ”Bottles”, ”Giraffes” and ”Swans” are considerably better than results in (Ferrari *et al.*., 2007): 49.3%/32.6% (Applelogos), 65.4%/33.3% (Bottles), 69.3%/43.9% (Giraffes) and 31.3%/23.3% (Swans). Also notice that we boost the performance by large amount compared to local shape context voting without contour selection.

multi-scale fashion, with non-maximum suppression on both position and scale.

Currently the model shape is manually decomposed into 6 to 8 contours at high curvature places. The contour partition respects the semantic object parts, *e.g.* two sides of the swan neck and the dent of the applelogo are kept as single model contours. As described in Section 3.3, configurations of matched model contours are used to reject false positives in addition to the packing score. In principle, the dictionary of valid configurations can be automatically learned from detections in training images. Since the shape models usually have very few contours, we manually construct a dictionary of acceptable configurations[2].

Precision vs. recall (P/R) curve is used for quantitative evaluation. To compare with the results in (Ferrari *et al.* , 2007a) which is evaluated by detection rate (DR) vs. false positive per image (FPPI), we translate their results into P/R values. We choose P/R instead

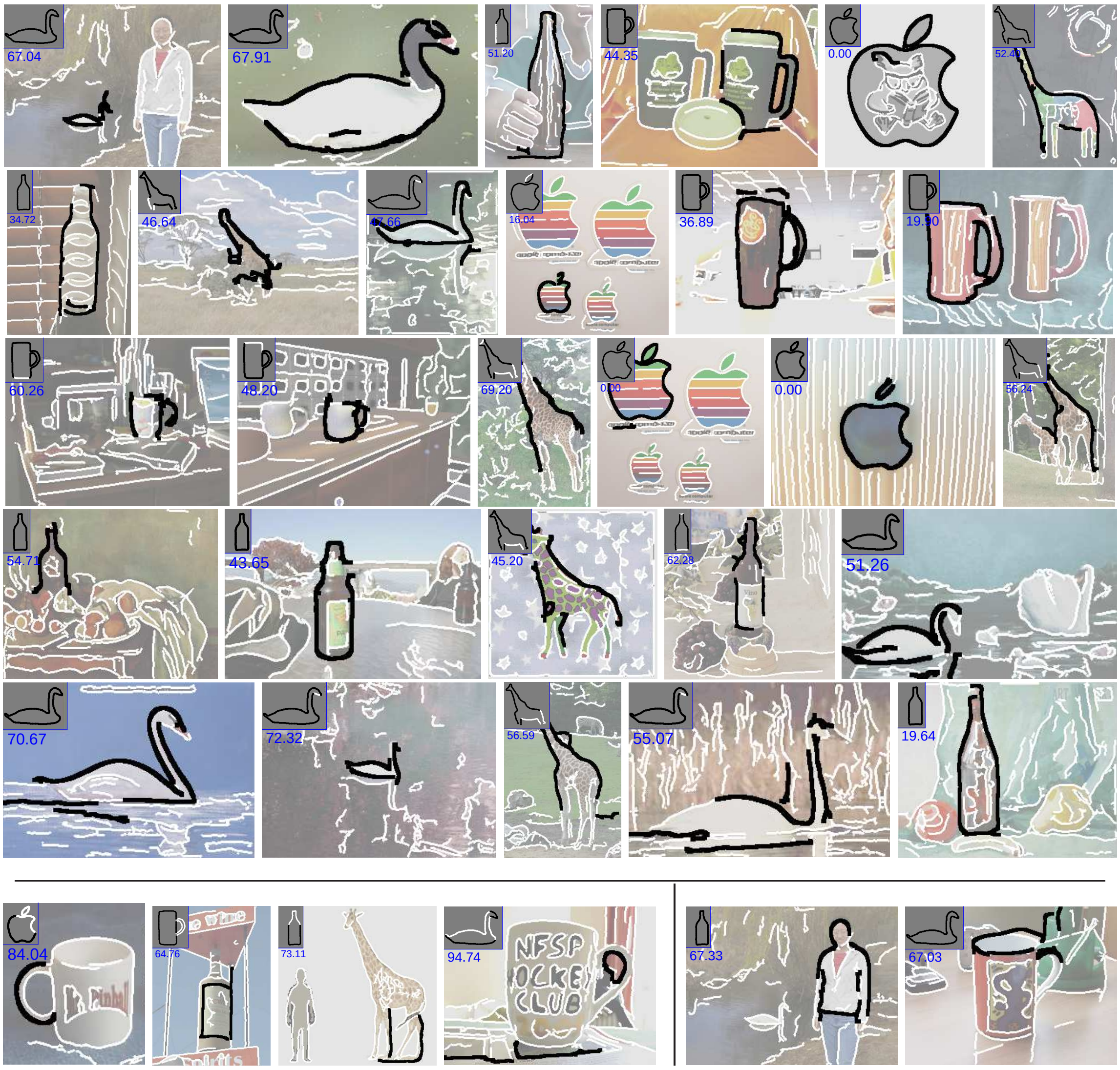
[2] We further bind some model contours, reducing the contour number to a maximum of 6, so that $2^6 = 64$ dictionary entries can be numerated by hand.

of DR/FPPI because DR/FPPI depends on the ratio of the number of positive and negative test images and hence is biased. Our final results on this dataset can be seen in Fig. 3.5. Results of the two steps of our framework are both evaluated. Single point figure/ground labeling only uses matching cost as the final evaluation for detection, while joint contour selection uses both matching cost and the detected shape configuration. Compared to the latest result in (Ferrari *et al.* , 2007a), our performance is considerably better on four classes out of five. We also compare voting using simple local shape context with our first step of contour selection. Contour selection greatly improves detection performances (see Fig. 3.5).

Our shape matching algorithm can reliably extract and select contours of object instances in test images, robust to background clutter and missing contours. Image results of detection with selected object and model contours are demonstrated in Fig. 3.6.

3.6 Summary

We have introduced a novel shape-based recognition framework called *Contour Packing*. We construct context sensitive shape features depending on selected contours and propose a method to search for the best match. Joint selection on both image and model contours ensures detection to be robust to background clutter and accidental alignment. We are able to detect object in cluttered images using only one training example. Experiments on hard object detection task demonstrate promising results.



False positives pruned by model contour selection

Failure cases

Figure 3.6: Examples of contour context selection on model and image contours in ETHZ Shape Classes. The first five rows show detected objects from image with significant background clutter. In the last row, the first four cases are false positives successfully pruned by our algorithm by checking the configurations of selected model contours. The last two are failure cases. Each image only displays one detected object instance.

Chapter 4

The Primal-Dual Packing Algorithm

In the previous chapter, we developed the set-to-set contour matching framework and derived a computational solution based on LP. The core of the solution is to encode the overall shapes at several control points in a linear form of figure/ground contour selection, which do have one-to-one correspondences. Since the control point correspondences are unknown, searching for the correct ones results in many LPs, one for each correspondence hypothesis. A natural question arises: do we really need to solve all LPs for the figure/ground contour selection precisely?

This chapter will show that this is unnecessary for most of the time. We introduce primal-dual combinatorial algorithms which have generated fast algorithms for a large class of packing and covering problems. The contour packing LP can be reduced to a bin covering LP, where these primal-dual ideas can be readily applied. By exploiting the duality between contours and feature bins, the algorithm is able to either find an approximate solution, or declare a lower bound on the optimum of the cost function. Therefore, most suboptimal solutions can be knocked out without running the LP to the end. Each iteration of the primal-dual algorithm only involves a simple operation of sorting the contours, making it very fast to generate approximate solutions.

4.1 Primal-Dual Combinatorial Algorithms

Linear programming (LP) has been widely used for analyzing combinatorial problems and designing fast approximation algorithms. The LP formulation leads to principled approaches for a large class of packing and covering problems (Plotkin *et al.* , 1995; Young, 1995), multi-commodity flow (Plotkin *et al.* , 1995; Leighton *et al.* , 1991), Travelling Salesman Problem (TSP) (Khandekar, 2004), faculty location (Vazirani, 2004), etc. The power of LP-based algorithms is largely attributed to the *duality* which simultaneously considers two different but coupled problems: the primal and the dual. Each one of them serves as a guidance and bound on solving its counterpart, providing a different perspective to the original problem.

In a seminal work (Plotkin *et al.* , 1995), Plotkin *et al.* proposed a *primal-dual combinatorial algorithm* for fractional packing and covering, which greatly outperformed previous approaches on a large set of problems such as minimal cost multi-commodity flow, the Held-Karp bound for TSP, and cutting stock. The key idea is to feed the current estimate of the dual to improve the primal during iterations, and vice versa. On the primal side, one solves an oracle with partial constraints and a simplified cost function induced by dual variables. This provides the freedom of designing oracles adapted to different problems and can employ existing efficient combinatorial algorithms. On the dual side, dual variables are adjusted by a multiplicative update rule according to the ”feedback” from the oracle. The updated dual variables thus give a tighter bound in the next iteration.

The primal-dual formulation provides more insights to the problem than just treating LP as a black-box. Computationally, while solving LP using general purpose solutions (Vaidya, 1996; Nesterov, Y. E. & Nemirovsky, A. S., 1993; Wright, 1997) (*e.g.* interior point methods) has shown some degree of success, *combinatorial* algorithms built on the primal-dual formulation can exploit specific structures, generate much more efficient approximation solutions, and provide explicit manipulation to the computational routine.

The LP formulation has been extensively used in general matching problems. In (Jiang *et al.* , 2007; Jiang & Martin, 2008), an LP relaxation was proposed for metric labeling with L_1-norm regularization in image matching. A simplex-based solution and an efficient successive convexification implementation were developed. Alternatively, interior point method was applied in a related formulation in image registration (Taylor & Bhusnurmath, 2008). The structure of the problem was exploited more effectively in solving the linear systems using specific matrix structures. LP was also used in the inner loop of iterative algorithms of Integer Quadratic Programming (IQP) arising in matching (Ren *et al.* , 2005a; Berg *et al.* , 2005). Although also formulated as an LP, our problem differs from previous ones in that set-to-set matching instead of one-to-one correspondence on feature points is performed. The selection variables in set-to-set matching are more densely related to each other, resulting in a fundamentally different matrix structure.

The rest of this chapter is organized as follows. Section 4.2 will review primal-dual algorithms for general fractional packing and covering problems, and lay down the foundation for applying these ideas subsequently in the contour packing problem. In Section 4.3 we reduce the single point figure/ground selection LP to a covering problem, and propose a primal-dual algorithm that enables pruning suboptimal solutions early. Section 4.4 describes details of how to apply the algorithm to contour packing.

4.2 Primal-Dual Algorithms for Packing and Covering

The packing problem studies how to optimally fill a knapsack by choosing the most valuable objects from a list. Suppose there are n objects whose prices are p_i $(i = 1, ..., n)$. One would like to choose a subset of these items maximizing their total price, subject to m capacity constraints such as weight, dimension, etc. Denote the maximum value of each capacity constraint as c_j and the contribution from item i as W_{ji}. Finding the optimal packing can be written as the following integer programming problem:

$$\text{(PACKING IP)} \qquad \begin{aligned} \max_{x\in\{0,1\}^n} \quad & \sum_i p_i x_i \\ \text{s.t.} \quad & \sum_i W_{ji} x_i \le c_j, \quad j = 1, ..., m \end{aligned} \tag{4.1}$$

where x_i is the $0/1$ indicator of whether object i is selected. By relaxing the integer constraint $x \in \{0,1\}^n$ to $0 \leq x \leq 1$, we obtain a linear program called *fractional packing* which provides a lower bound to eq. (4.2):

$$\text{(PACKING LP)} \qquad \max_{x \in \mathbb{R}^n} \; p^{\mathrm{T}} x \qquad \text{s.t. } A \cdot x \leq c, \;\; x \geq 0 \tag{4.2}$$

Here $A = [W; I]$ and $c = [c_1, ..., c_m, \underbrace{1, ..., 1}_{n}]^{\mathrm{T}}$. Hence the constraint $x \leq 1$ has been folded into the matrix constraint $A \cdot x \leq c$.

The covering problem is to find sets with minimal total cost to cover elements. Let the c_j's be the costs of the n sets. Each set j covers element i for W_{ij} times. The multiplicity of each element i to be covered is required to be at least p_i. Let y_j be the number of copies of set j that are selected (choosing multiple copies are allowed). Similarly to packing, the covering problem can be written as an integer program, and relaxed to *fractional covering*:

$$\text{(COVERING IP)} \qquad \min_{y \in \mathbb{N}^n} \; \sum_j c_i y_j \qquad \text{s.t. } \sum_j W_{ij} y_j \geq p_i, \;\; i = 1, ..., n \tag{4.3}$$

$$\text{(COVERING LP)} \qquad \min_{y \in \mathbb{R}^n} \; c^{\mathrm{T}} y \qquad \text{s.t. } A^{\mathrm{T}} \cdot y \geq p, \;\; y \geq 0 \tag{4.4}$$

The fractional packing problem eq. (4.2) and fractional covering eq. (4.4) are actually Lagrangian duals. By introducing nonnegative Lagrangian multipliers (y, λ) to the constraints $A \cdot x \leq c$ and $x \geq 0$ respectively, the Lagrangian function $\mathcal{L}(x, y, \lambda) = p^{\mathrm{T}} x + y^{\mathrm{T}}(c - Ax) + \lambda^{\mathrm{T}} x$ always serves as an upper bound of the fractional packing cost function $p^{\mathrm{T}} x$, whenever x is feasible or not. Therefore, $\max_x \mathcal{L}(x, y, \lambda)$ bounds the optimum of eq. (4.2). By strong duality of linear program, the optimum of eq. (4.2) and eq. (4.4) coincides (Boyd & Vandenberghe, 2004). Therefore packing and covering are essentially flipped sides of the same coin: solving one implies the other.

Primal and dual formulations provide different perspectives on the problem: for the feasibility version, primal solution serves as "yes" certificate while the dual solution serves as “no” certificate. Just as the divide-and-conquer strategy, one would like to generate a series of yes and no certificates to narrow down the search space. Therefore, primal and dual need to communicate, and use one to update the other.

We start with a feasibility version of the fractional packing problem:

(Feasibility Problem) Given a convex set $P \subseteq \mathbb{R}^n$, an $m \times n$ constraint matrix A and an $n \times 1$ vector c, determine whether there exists $x \in P$ such that

$$a_j^{\mathrm{T}} x - c_j \leq 0, \ \ j = 1, ..., m \tag{4.5}$$

Here a_j^{T} is the jth row of matrix A.

For the packing problem (4.2), the convex set P is a simple polytope:

$$P = \{x : \ p^{\mathrm{T}} x \geq \alpha, \ 0 \leq x \leq 1\} \tag{4.6}$$

where α is a constant. If eq. (4.5) is feasible, then the optimal value μ_p^* of eq. (4.2) is at least α. Otherwise it is less than α. By a binary search on α, one can find a $(1 + \beta)$ approximation to the optimization problem within $O(\log \beta)$. Our discussion will focus on eq. (4.5) in the subsequent sections.

4.2.1 Multiplicative Weight Update: From Primal to Dual

Suppose we are given a primal estimate and its corresponding cost as feedback, how can we update the current dual estimate? We start with considering an online prediction problem.

Online Prediction. There are m experts who make predictions on uncertain events in the world. Our goal is to construct the best strategy over time from these experts. At time t $(t = 0, 1, 2, ...)$, if the prediction from the jth expert is taken, the event (possibly adversarial) incurs a positive *reward* $\mathcal{R}_j^t$ and a negative *loss* $-\mathcal{L}_j^t$. Hence the net *value* gained is $\mathcal{V}_j^t = \mathcal{R}_j^t - \mathcal{L}_j^t$. One can construct a mixed strategy from these experts by linearly combining their predictions. A mixed strategy specifies positives weights $y^t = (y_1, ..., y_m)^{\mathrm{T}}$ on all the experts. The total net value of the strategy will be $\mathcal{V}^t = \sum_j \overline{y}_j^t \mathcal{V}_j^t$

where $\overline{y}^t = y^t / \sum_j y_j^t$ are the normalized weights. Consider the event sequence from time $t = 0$ to T. At time t, the strategy chooses weights y^t on the experts based on all previous observations $\mathcal{R}^k$ and $\mathcal{L}^k$ with $0 \leq k \leq t-1$, and gains a value $\mathcal{V}^t$. One would like to maximize the cumulative value over time $\mathcal{V} = \sum_{t=0}^{T} \mathcal{V}^t$.

Intuitively, experts making correct predictions previously should be up-weighted while experts predicting incorrectly should be down-weighted. In other words, the weights should be updated according to the "feedback" of the experts from the world $\mathcal{V}_j^t$. We introduce a multiplicative weight update scheme to guide the strategy from the feedback:

(Multiplicative Weight Update) Initialize weights $y^{(0)} = (1, ..., 1)^{\mathrm{T}}$. At time t, prediction from expert j produces a value of $\mathcal{V}_j^t \in [-1, 1]$. Given a constant $\epsilon \in (0, 1)$, update the weights y^{t+1} at time $t+1$ by

$$y_j^{t+1} = y_j^t \exp(\epsilon \mathcal{V}_j^t) \tag{4.7}$$

Theorem 4.1. *(Littlestone & Warmuth, 1989) (Perturbed Value of the Strategy) Let $\mathcal{R} = \sum_t \sum_j \overline{y}_j^t \mathcal{R}_j^t$ and $\mathcal{L} = \sum_t \sum_j \overline{y}_j^t \mathcal{L}_j^t$ be the cumulative reward and loss of the strategy using eq. (4.7). The perturbed value of the strategy given by eq. (4.7) is worse than the performance of best pure strategy only by $\frac{\log m}{\epsilon}$, as stated in the following inequality:*

$$\max_j \mathcal{V}_j \leq \exp(\epsilon)\mathcal{R} - \exp(-\epsilon)\mathcal{L} + \frac{\log m}{\epsilon} \tag{4.8}$$

Proof. Please see Appendix. □

Theorem 4.1 is essential in the complexity analysis in the subsequent sections. It proves the quality of the multiplicative update rule (4.7). Since the average strategy given by the update rule cannot exceed the best strategy in the hindsight, we would like the gap between their values $\max_j \sum_t \mathcal{V}_j^t$ and $\sum_t \mathcal{V}^t$ to be small. This value is called *regret* of the strategy. The theorem proves the fact that the regret is as small as $\log m/\epsilon$. We can bound the regret over time by the following corollary:

Corollary 4.2. *(Regret Over Time) If $\mathcal{V}_j^t \in [-\rho, \rho]$ for all j, then we have a bound on the average value $\mathcal{V}/T$:*

$$\max_j \frac{\mathcal{V}_j}{T} \leq \frac{\mathcal{V}}{T} + \frac{\rho \log m}{\epsilon T} + \rho\epsilon \exp(\epsilon) \tag{4.9}$$

Proof. Please see Appendix □

The above bound shows that the regret over time consists of two terms: the term $\frac{\rho \log m}{\epsilon T}$ which can be "washed out" by time and the other term $\rho\epsilon\exp(\epsilon)$ which cannot. If we would like to diminish the regret over time, for example proportional to a small number δ, we can set $\epsilon \sim \delta/\rho$ and $T \sim \rho^2/\delta^2$. However, if $\mathcal{V}$ only contains reward or loss, the result can be strengthened as:

Corollary 4.3. *(Regret for Reward Only) If $\mathcal{V}_j^t \in [0, \rho]$ for all j, i.e. $\mathcal{L}_j^t = 0$ for all t and j, then we have a bound on the average value $\mathcal{V}/T$:*

$$\max_j \frac{\mathcal{V}_j}{T} \leq \exp(\epsilon) \cdot \frac{\mathcal{V}}{T} + \frac{\rho \log m}{\epsilon T} \tag{4.10}$$

The corollary is a direct consequence of eq. (4.8). It makes a stronger claims than Corollary 4.2 since we only need to set $T \sim \rho/\delta$ to make the regret over time small, instead of $T \sim \rho/\delta$. This is the fundamental difference between packing/covering and general LP, in which the latter has higher complexity.

4.2.2 The Oracle: From Dual to Primal

From the dual formulation, we would like to improve the current primal solution by minimizing $\sum_j y_j f_j(x)$.

(Oracle) Given a convex **constraint set** $P \subseteq \mathbb{R}^n$, a dual variable $y \in \mathbb{R}^m$ and a set of functions $\mathcal{V}_j(x)$ $(j = 1, ..., m)$. Optimize the linear combination of $\mathcal{V}_j(x)$ in the constraint set P:

$$\min_{x \in P} \sum_j y_j \mathcal{V}_j(x) \tag{4.11}$$

The constraints in the original problem have been separated into two parts. Constraints easy to check and optimize are pushed into CONSTRAINT SET P, making the oracle efficient to compute. Hard constraints are left outside and are only approximated by the Lagrangian as in eq. (4.11). It is a design choice how to divide the two.

In the case of packing, P is given by eq. (4.6). Define $\mathcal{V}_j(x) = a_j^{\mathrm{T}}x - c_j$ for $j = 1, ..., m$. Notice that $\sum_j y_j \mathcal{V}_j(x) = (A^{\mathrm{T}}y)^{\mathrm{T}}x - c^{\mathrm{T}}y$, given y^{sel}, the oracle becomes

$$\begin{aligned} &\min_x \, (A^{\mathrm{T}}y)^{\mathrm{T}}x \\ &\text{s.t. } c^{\mathrm{T}}x = \alpha, \; 0 \le x \le 1 \end{aligned} \tag{4.12}$$

If $c \ge 0$ and $A \ge 0$, one can solve eq. (4.12) by simply sorting $(A^{\mathrm{T}}y)_j/c_j$ in ascending order, and choosing $x_j = 1$ according to the order until $c^{\mathrm{T}}x = \alpha$ is satisfied. The oracle (4.11) simply reduces to sorting, whose complexity is $O(n \log n)$.

4.2.3 Complexity Analysis

So far we have all the ingredients of primal dual combinatorial algorithms. We summarize the primal-dual algorithm for packing as follows:

Algorithm 2 (Primal Dual Algorithm)

```
Initialize $y^0 = (1, ..., 1)^{\mathrm{T}}$, $t = 0$, $S = 0$, $\epsilon = \delta/3\rho$. Define $f_j(x) = a_j^{\mathrm{T}}x - c_j$.
repeat
    Run oracle (4.11) and obtain the optimum $\mu^t$ and optimal point $x^t$.
    if $\mu^t > 0$ then
        return infeasible.
    end if
    Compute $w^t := 1/\max_j |f_j(x)|$.
    Run multiplicative weight update (4.7): $y_j^{t+1} := y_j^t \exp(\epsilon w^t f_j(x^t))$
    $S := S + w^t$, $t := t + 1$.
until $S \ge 9\rho \log m/\delta^{-2}$
return feasible solution $\bar{x} = \frac{\sum_t w^t x^t}{\sum_t w^t}$.
```

Theorem 4.4. *(Complexity of the Primal Dual Algorithm) Algorithm 2 either declares that the fractional packing eq. (4.2) is infeasible, or outputs an approximate feasible solution $\bar{x}$ satisfying*

$$a_j^{\mathrm{T}} \bar{x} - c_j \leq \delta \tag{4.13}$$

for all $j = 1, ..., m$. The total number of calls to the oracle is $O(\rho^2 \delta^{-2} \log m)$ with $\rho = \max_j \max_{x \in P} |f_j(x)|$.

Proof. Please see Appendix. □

Variant 1. If $A, c \geq 0$, we can improve the running time of Algorithm 2 to $O(\rho\delta^{-1} \log m)$ by changing the termination condition to $S \geq \rho\delta^{-1}\epsilon^{-1} \log m$ and set $f_j(x) = a_j^{\mathrm{T}} x / c_j$.

Variant 2. If $f_j(x) \geq 0$ for $x \in P$, we can improve the running time of Algorithm 2 to $O(\rho\delta^{-1} \log m)$ by changing the termination condition to $S \geq \rho\delta^{-1}\epsilon^{-1} \log m$.

In both cases, we can apply Corollary 4.3. Eq. (A.33) has a tighter bound $\max_j [a_j^{\mathrm{T}} \bar{x} - c_j] \leq \frac{\log m}{\epsilon S}$, the rest of the analysis falls through.

4.3 Primal-Dual Formulation for Contour Packing

This section presents an alternative formulation of contour packing as oppose to the direct LP relaxation in Chapter 3. Applying the primal-dual ideas for general packing/covering in the previous section leads to an efficient, and incremental style search algorithm.

Consider the single point figure/ground selection eq. (3.10) with full model $sc^M = V^M \cdot 1$. We introduce normalized slacks s^+, $s^- \geq 0$ such that the surplus and deficit of the bins are $b^+ = \mathrm{Diag}(sc^M)s^+$ and $b^- = \mathrm{Diag}(sc^M)s^-$ respectively. The main constraint in eq. (3.10) can be written as:

$$V^I x^{sel} - sc^M = \mathrm{Diag}(sc^M)s^+ - \mathrm{Diag}(sc^M)s^- \tag{4.14}$$

The term $\mathrm{Diag}(sc^M)s^+$ represents the amount of over-packed edge points in the feature bins and $\mathrm{Diag}(sc^M)s^-$ represents the amount of the under-packed. Since $sc^M, s^- \geq 0$, we have a covering constraint $V^I x^{sel} + \mathrm{Diag}(sc^M)s^- = sc^M + \mathrm{Diag}(sc^M)s^+ \geq sc^M$.

By substituting $\mathrm{Diag}(sc^M)s^+ = V^I x^{sel} - sc^M + \mathrm{Diag}(sc^M)s^-$, the contour figure/ground selection cost eq. (3.10) becomes

$$\begin{aligned}\|V^I \cdot x^{sel} - sc^M\|_1 &= 1^\mathrm{T}[\mathrm{Diag}(sc^M)s^+ + \mathrm{Diag}(sc^M)s^-] \\ &= 1^\mathrm{T}[2 \cdot \mathrm{Diag}(sc^M)s^- + V^I x^{sel} - sc^M] \\ &= 2 \cdot (sc^M)^\mathrm{T} s^- + 1^\mathrm{T} V^I x^{sel} - 1^\mathrm{T} sc^M\end{aligned}$$

The last term $1^\mathrm{T} sc^M$ is a constant and hence can be dropped. Moreover, the under-packed slack variable s^- is bounded by 1. Notice that at most one of s_i^+ and s_i^- needs to be strictly positive. Otherwise subtract the minimum of s_i^+ and s_i^- will drive one of them down to 0, but with a lower cost. If $s_i^- > 0$, then $s_i^+ = 0$. and the constraint eq. (4.14) implies $sc_i^M s_i^- = sc_i^M - (V^I x^{sel})_i \leq sc_i^M$, which means $s_i^- \leq 1$ for each i. By putting the cost function and the constraints together, we simplify eq. (3.10) to a standard covering problem on the bins:

$$\text{(BIN COVERING)} \qquad \begin{aligned}\min_{x^I, s^-} \;& 1^\mathrm{T} V^I x^{sel} + 2 \cdot (sc^M)^\mathrm{T} s^- \\ s.t. \;& V^I x^{sel} + \mathrm{Diag}(sc^M)s^- \geq sc^M \\ & 0 \leq x^{sel}, s^- \leq 1\end{aligned} \tag{4.15}$$

The primal-dual method iterates between 1) the oracle that solves the packing oracle, which boils down to sorting the contours and bins in this case; 2) the multiplicative update that changes dual variables y by multiplication.

Oracle

The oracle for contour packing has the following form:

$$\text{(ORACLE: PACKING)} \qquad \begin{aligned}\max_{x^{sel}, s^-} \;& y^\mathrm{T}[V^I x^{sel} + \mathrm{Diag}(sc^M)s^-] \\ s.t. \;& 1^\mathrm{T} V^I x^{sel} + 2 \cdot (sc^M)^\mathrm{T} s^- \leq f_0 \\ & 0 \leq x^{sel}, s^- \leq 1\end{aligned} \tag{4.16}$$

Let $x = (x^{sel}; s^-)$, $c = (V^I 1; 2 \cdot sc^M)$ and $A = (V^I, \mathrm{Diag}(sc^M))$. This problem can be written as $\max_x y^\mathrm{T} A x$ subject to $c^\mathrm{T} x \leq f_0$ and $0 \leq x \leq 1$. A greedy algorithm that packs x according to the sorted value/capacity ratio $\frac{(A^{\mathrm{T}}y)_i}{c_i}$ can efficiently acheive the global optimum.

Multiplicative Update

The update is similar to the general packing/covering problem:

$$(\text{U}\text{PDATE}) \qquad y \leftarrow y \cdot \exp(\delta), \;\; \delta = (sc^M - Ax) \cdot \epsilon \tag{4.17}$$

with δ representing how much violation is incurred for each covering constraint.

By combining Algorithm 2, eq. (4.15) and eq. (4.16), we summarize the primal-dual contour packing algorithm as follows:

Algorithm 3 (Primal Dual Contour Packing)

```
 1: Initialize $x = (0, ..., 0)^{\mathrm{T}}$, $y^0 = (1, ..., 1)^{\mathrm{T}}$, $t = 0$, $S = 0$.
 2: for $t = 1, 2, ..., T_{max}$ do
 3:     $u := A^{\mathrm{T}}y$, $x^t := (0, ..., 0)^{\mathrm{T}}$, $f := f_0$.
 4:     Sort $u_i/c_i$ in descending order, with indices $s(1), s(2), ..., s(n)$.
 5:     for $i = 1, 2, ..., n$ and $f > 0$ do
 6:         $k := s(i)$, $x_k^t := x_k^t + \min(f/c_k, 1)$, $f := f - c_k x_k^t$.
 7:     end for
 8:     if $y^{\mathrm{T}}(Ax^t - sc^M) < 0$ then
 9:         return infeasible.
10:     end if
11:     $w^t := 1/\max_j |\delta_j(x)|$.
12:     Run multiplicative weight update: $y_j^{t+1} := y_j^t \exp(\epsilon w^t f_j(x^t))$.
13:     $x := x + w^t x^t$, $S := S + w^t$, $t := t + 1$.
14:     if $c^{\mathrm{T}}x/S < f_0$ then
15:         return feasible with the solution $x/S$.
16:     end if
17: end for
18: return the best primal solution $x/S$.
```

In line 3-7, the algorithm uses sorting to solve the oracle eq. (4.15). Note that each iteration involves only one matrix vector multiplication (in line 3) and one sorting operation (in line 4). This is faster by orders of magnitude compared to one iteration of the standard interior point LP solvers, which involves solving a linear system (Wright, 1997). Additionally, the sorting can be updated from the previous iteration, which provides more speed-up to algorithm. The rest of the algorithm is similar Algorithm 2, except for early stopping via checking the current solution in line 13.

4.4 Implementation

We apply the primal-dual packing algorithm to the single point figure/ground selection in Section 3.3. This is the most time-consuming step because a large amount of LP instances need to be solved in our original formulation. In each scale, n image control points and m model control points will generate $n \times m$ correspondence hypotheses, with each one as an LP. An important observation is that many of these hypotheses are competing with each other. Notice that the correspondence U_{ij}^{cor} in eq. (3.7) has to be one-to-one. If correspondence (i, j) has the best cost (3.9), then all other correspondences $(i, *)$ sharing the same image control point i will be suboptimal and should be discarded from eq. (3.7). In other words, the current estimation on (i, j) provides an upper bound on the optimum, making it possible to prune correspondences $(i, *)$ early. Algorithm 3 we developed in the previous section computes a coarse bound efficiently, and hence is a perfect candidate for this purpose. The above intuitions are summarized in Algorithm 4.

In this template, we leave several steps open for problem specific optimizations.

1. The order enumerating control point pairs (i, j) in line 2 can be arbitrary. The sooner to encounter a good solution, the more correspondences we can prune early. One way is to sort their current best estimation by running Algorithm 3 for just a small fixed number of steps. We found this a good heuristic in practice, because the most important contours tend to be packed first.

2. The bounds B_i in line 1 can be extended to enable more pruning. For example, one could introduce B_j for all the correspondences $(*, j)$ that votes for the same object

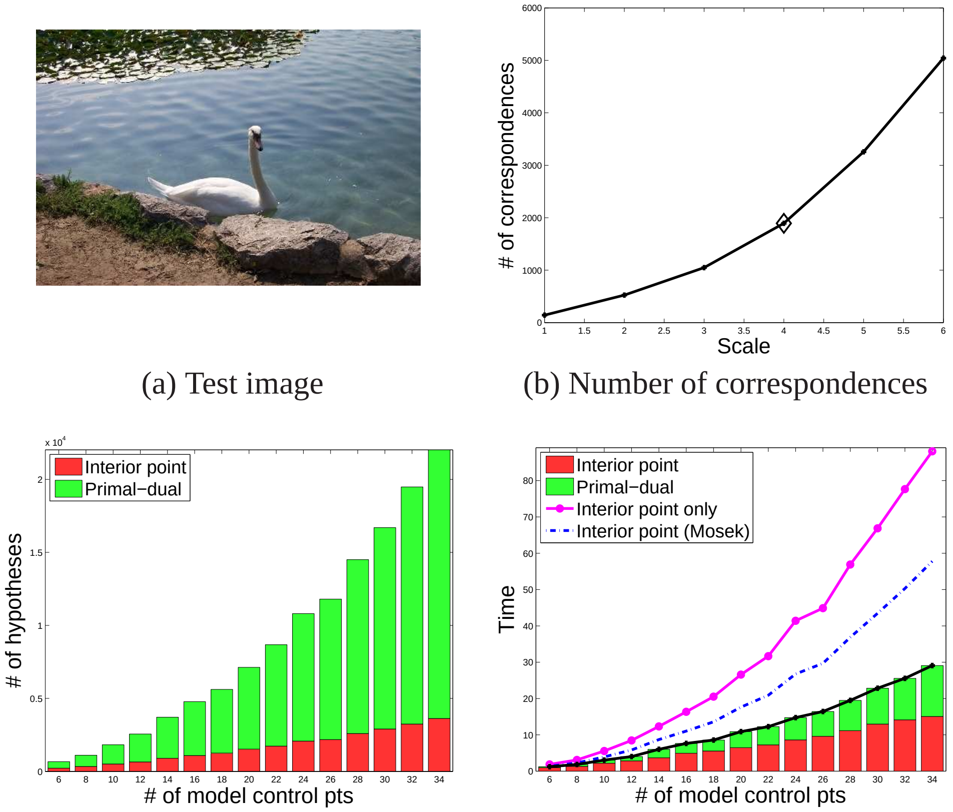
Algorithm 4 (Single Point Figure/Ground Selection – A Faster Version)

```
1:  Initialize $B_i = \inf$, $i = 1, ..., n$.
2:  for $(i, j)$, $1 \leq i \leq n$ and $1 \leq j \leq m$ do
3:      Run Algorithm 3 for $f_0 = B_i$.
4:      if infeasible then
5:          break
6:      else
7:          Compute optimal value $c^*$ in eq. (3.10).
8:          $B_i := \min(B_i, c^*)$.
9:      end if
10: end for
```

center, ensuring unique matching on the model side. Additionally, it can be used to encode non-maximal suppression.

3. The final step of computing an optimal primal solution in line 7 can be any algorithm, include the standard LP solutions. Although in principle the same primal-dual algorithm can be continued, it might requires many more iterations to converge to a final accurate. In practice we adopt a path following interior point method (Wright, 1997). The Newton's iterations in interior point methods are particularly suited for this purpose since it is closer to the optimum, and hence faster convergence can be expected. This results in a hybrid implementation that takes advantages from both sides.

The complexity of Algorithm 4 depends on the portion of correspondences pruned in line 5. How much overall speed up can we gain from this primal-dual packing algorithm? We test it on ETHZ images used in Section 3.5. We plot the number of iterations and time used by primal-dual pruning in line 3 and the interior point method in line 7, varying the number of model control points. As shown in Fig. 4.1, the portion of solutions pruned by the primal-dual packing algorithm increases with more model control points, leading to bigger speed-up. Thanks to the efficient combinatorial oracle, the primal-dual iteration is



(a) Test image (b) Number of correspondences

(c) Percentage (d) Running time

Figure 4.1: Performance of primal-dual packing algorithm. Single point figure/ground selection is run in 6 scales to detect the swan shape in (a). The number of model control points ranges from 5 to 35. (b) shows the number of hypotheses to search in all the scales when the number of model control points is 28, with scale 4 marked in diamond (the scale in which the swan is detected). (c) shows the proportion of correspondences handled by primal-dual iterations (line 3) and interior point iterations (line 7) in Algorithm 4. In (d), the running time of the entire algorithm is shown and compared to the one without primal-dual pruning. Note that the rejection by primal-dual iterations consumes very little time in the algorithm.

at least two orders of magnitude faster than the interior point iteration on average. This makes it suitable for fast pruning suboptimal solutions.

4.5 Summary

We have shown that the LPs arising from contour packing do not need to be solved exactly for most of the time. The contour packing LP is first reduced to a fractional covering problem. We borrow the idea of primal-dual combinatorial algorithms that are able to prune and bound packing and covering problems through duality and efficient oracles. We develop an algorithm applying these ideas to single point figure/ground selection which involves massive LP instances. Most of these LPs can be efficiently pruned by the primal-dual combinatorial algorithm, without resorting to solve the original LPs explicitly. Preliminary results confirm that the primal-dual algorithms can greatly relieve the computational burden from standard LP solvers. We plan to explore more applications of the algorithm in set-to-set matching.

Chapter 5

Contour Packing with Model Selection

So far we have developed a framework that detects a subset of image contours matched to a model shape in a holistic manner. Shape models involved in the previous chapter are simply exemplars composed of a few contours. Although the set-to-set matching method endows the model the ability to accommodate different contour fragmentations, a fixed target shape cannot handle large object deformations in images. Deformations often generate a combinatorial configuration space with exponentially many poses. This makes brute-force search for the best exemplar prohibitive in practice. Moreover, it deepens the discrepancy between the model and image shape descriptions because both side have exponentially many configurations now.

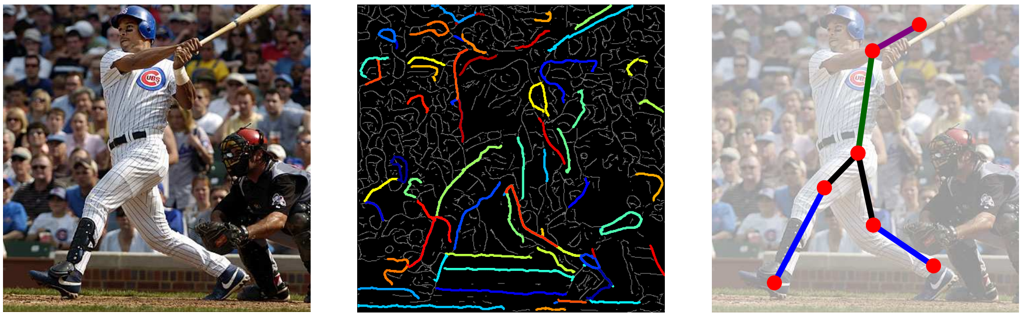
In this chapter we push contour packing further to relate bottom-up contours to top-down deformable parts beyond exemplars, addressing a bigger representation gap. We study the challenging problem of articulated human pose estimation from unsegmented images. A compact model representation is developed to encode exponentially many poses via a few configuration selection variables on a tree. The set-to-set matching method extended for this new model representation can search and compare holistic shape features of both image contours and model parts on the fly. This alleviates the reliance on local shape features of parts, which often causes many false detections in clutter. The parallel search over holistic shape features can be efficiently approximated by an LP-based computational solution. We demonstrate results of human pose estimation on baseball player images with wild pose variations.

5.1 Overview

Estimating poses for deformable or articulated objects is a challenging problem for two reasons. The first reason is the large number of degrees of freedom to be estimated. Due to the extreme pose variations, prior knowledge is of limited use in guiding the search. Second, images are often cluttered and bottom-up detection of parts is usually prone to error. Again this is due to the fact that shape is a global percept – a part is seldom salient without the whole shape.

For articulated objects, contour is a compact and effective shape representation. However, finding the foreground contours and estimating the object poses or articulations is a circular problem. One individual bottom-up contour can hardly cover the entire object by itself. If we know the right set of contours composing the foreground object, then we can recognize the object by matching against a set of candidate models or exemplars. On the other hand, this becomes circular because grouping contours into an object shape requires the correct model. We can think of this problem as a puzzle of two parallel searches, one for finding the right foreground contour grouping and one for generating the correct object model. A naive approach to this would result in an exponential search.

We propose an active search method that finds the correct object contour grouping and model configuration in one step. To encode this search, we extend the selection variables which can be turned ON and OFF in Chapter 3. On the image side, each contour acts as an integral unit that can either be selected or discarded as a whole. On the model side, we deform a decomposable articulated model. Recognition is achieved if the model pose matches the image foreground. We have developed a method for generating a holistic shape descriptor based on these ON/OFF selection variables. Computationally this leads to solving an integer program and a subsequent linear programming relaxation. A discrete solution can be recovered using dynamic programming (DP) to discretize the continuous solution of linear programming relaxation.



(a) Original image (b) Contours on Pb edge map (c) Estimated pose

Figure 5.1: Given an image (a), salient contours are extracted (b) from the edge map of Pb Having contours as our unit, we use a coupled optimization procedure of foreground contour selection and model deformation to recover the pose of an articulated baseball player (c).

The key contribution of our approach is unification of *a holistic shape scoring scheme and a compositional model.* We take advantage of the compositional power of a simple tree structured model while scoring shape similarity in a holistic way during our search. This is in contrast to a typical part-based model, which only measures shape similarity as sum of its local part matches. Matching global shape requires correct foreground contour selection to remove the effect of clutter. Furthermore, our global shape descriptors vary depending on each composition of foreground contours. Searching for the correct segmentation/grouping is a hard combinatorial problem. As far as we are aware, this is the first approach that extracts global shape features without knowing the correct segmentation and modifies the shape descriptors according to the foreground selection at each step of the estimation process, making them robust to background and interior clutter.

The rest of the chapter is organized as follows. Section 5.2 describes related work and comparisons. Section 5.3 and 5.4 present the problem of pose estimation combining foreground search and model deformation and an efficient LP-based computational solution. Section 5.5 demonstrates our approach on the problem of pose estimation on the baseball dataset (Mori *et al.* , 2004b), followed by conclusion in Section 5.6.

5.2 Related Work

Pose estimation of articulated objects remains an important unsolved problem in vision. There has been a large amount of previous work on this topic. Here we review only some of the most representative examples. (Felzenszwalb & Huttenlocher, 2005) developed the well-known pictorial structures (PS) and applied it to human pose estimation. In the original formulation, PS performs probabilistic inference in a tree-structured graphical model. In this model, the overall cost function for a pose decomposes across the edges and nodes of the tree, usually with the torso as the root. Although our method exhibits the compositional power of a similar tree-structured graphical model, our score function *measures shape holistically* and not as the sum of local similarities as (Felzenszwalb & Huttenlocher, 2005; Ramanan, 2007). Many approaches (Mori *et al.* , 2004b; Cour & Shi, 2007; Mori, 2005; Lee & Cohen, 2004; Zhang *et al.* , 2006; Ronfard *et al.* , 2002) are based on part detection and search. Due to the fact that part detectors are prone to error, some authors have used additional cues like skin color, which however limits the generality of the approach. Search approaches need to use heuristics to deal efficiently with the combinatorial nature of the problem. In our method, we are not based on local decision to guide the search. Instead, the model is compared as a whole against the image at each step, and this is done efficiently using an LP formulation. (Srinivasan & Shi, 2007) uses hand written compositional rules for augmenting partial body masks which are compared against exemplars at each stage and correspondences are recomputed. Although the body is measured as a whole, the method suffers from the explosion of the number of hypotheses as in usual search-based parsing approaches, due to the absense of a good heuristic function. (Ren *et al.* , 2005a) used bottom-up detection of parallel lines in the image as part hypotheses, and then combined these hypotheses into a full-body configuration via an integer quadratic program.

Many of the above approaches ignore the representation gap between parts in the model and bottom-up extraction results, and treat the result of a bottom-up process, like segmentation or parallel line detector, as exactly corresponding to body parts. This is far

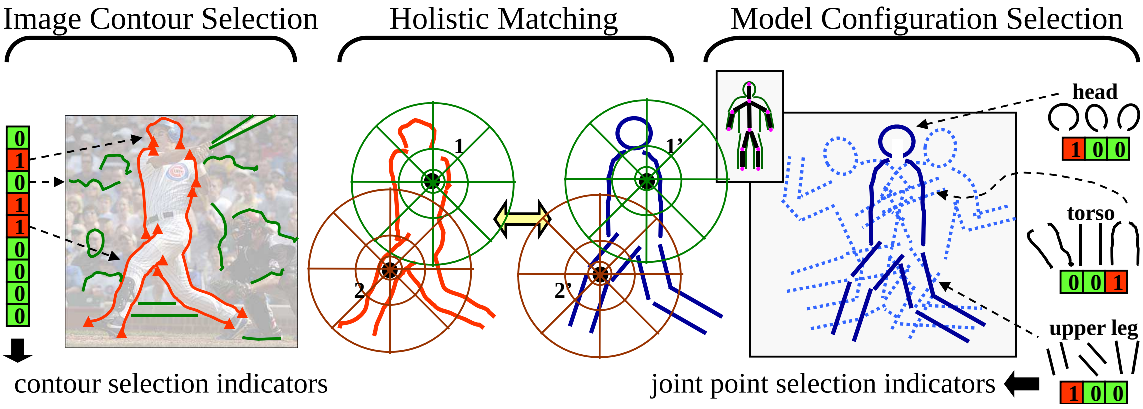


Image Contour Selection
Holistic Matching
Model Configuration Selection
0
1
0
1
1
0
0
0
0
contour selection indicators
1
2
1'
2'
head
1 0 0
torso
0 0 1
upper leg
1 0 0
joint point selection indicators

Figure 5.2: Holistic shape matching. Our search has two parallel process, each encoded by a selection variable. On the image side (left), contour selection variables turn image contours ON and OFF assigning them to foreground or background respectively. This results in all feasible shapes on the image side. On the model side, selection variables assign configurations to each model part in the tree structure. The two shapes, one derived from the image and one from the model, are compared to each other using a holistic shape feature. When the two match, recognition and pose estimation are achieved. Therefore the recognition task amounts to finding the optimal selection on both the image and the model side.

from being true in many cases. For example, in a straight leg you cannot expect to obtain the upper and lower part of the leg separately. Our holistic view of shape surpasses this difficulty,

5.3 Holistic Shape Matching

In this section, we first present the pose estimation formulation in terms of image contours and model parts. Then we introduce our articulated model representation, with an active shape description built in. The design of the active model shape descriptor is the key to holistic shape matching.

5.3.1 Formulation of Pose Estimation Problem

Starting with contours as our basic units in the image, we develop the following formulation.

Pose Estimation Problem. Given image $\mathcal{I}$ represented by a set of contours and model $\mathcal{M}$ represented by a set of parts:

- Image: $\mathcal{I} = \{C_1^I, C_2^I, \ldots C_{|\mathcal{I}|}^I\}$, C_k^I is the k^{th} contour;
- Model: $\mathcal{M} = \{P_1^\Theta, P_2^\Theta, \ldots P_{|\mathcal{M}|}^\Theta\}$ where P_k^Θ is the k^{th} part of the model and Θ is a family of global parameters controlling model deformation.

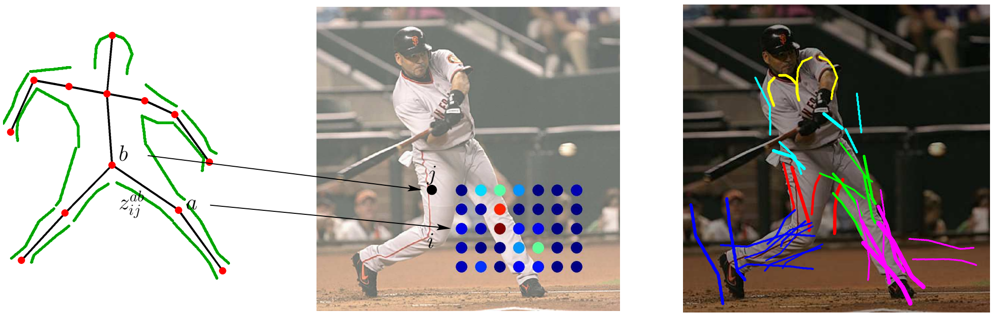
We would like to select the best subset $\mathcal{I}^{sel} \subseteq \mathcal{I}$ and Θ such that the shapes composed by $\mathcal{I}^{sel}$ and model parts P_k^Θ are most similar as scored by global shape descriptors (see Fig. 5.2). Note that this is another *set-to-set* matching since there might not exist a one-to-one mapping between selected image contours and contours of model configurations, even though they have similar overall shapes. For example, elongated contours might span multiple parts. We introduce the contour selection indicator $x^{sel} \in \{0,1\}^{|I| \times 1}$ over all contours in the *entire* test image defined as

$$\text{(Image Contour Selection)} \quad x_\ell^{sel} = \begin{cases} 1 & \text{Contour } C_\ell^I \text{ is selected} \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

Accordingly we introduce a set of configuration selection indicators $y^{part} = \{y_\Theta^k\}$ over all parts P_k^Θ in the model as

$$\text{(Model Configuration Selection)} \quad y_\alpha^k = \begin{cases} 1 & \text{Part } P_k \text{ selects config. } \alpha \in \Theta \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

Notice that since there is an infinite number of poses defined by Θ, resulting in an infinite number of choices for our selection variables. We will show later that the selection y_α^k on model articulation can be decomposed and simplified to limited choices by borrowing the compositional power of a tree structure model. This problem statement is similar to the one in Section 3.2. Parts with different configurations (y^{part}) replace contours (y^{sel}) as tokens in model representation to handle articulation. The shapes generated from the two



(a) The articulated model (b) Sample points of joints (c) Pose sketch

Figure 5.3: Object model and articulation. The model deformation Θ is controlled by joint positions. Once positions of two adjacent joints a and b are determined, shown in i and j in (b), the part can deform accordingly. This type of deformation can be encoded by the selection variable y_{ij}^{ab} on the model side. Continuous relaxation using LP produces sketch-like rough pose estimations of parts, marked by different colors in (c). Note that for most parts, the values of y_{ij}^{ab} are very small. (b) also shows the sum of y_{ij}^{ab} at all the sample locations for one joint, with red for large values and blue for small values. These values give the confidence of the joint locations. In this case, it correctly locates the knee.

independent selection processes are then compared using global shape descriptors (see the middle part of Fig. 5.2).

Unknown segmentation/grouping presents a great challenge to any *fixed* image shape descriptors (*e.g.* shape context). Fixed shape descriptors cannot adapt to the combinatorial possibilities of grouping, each generating a different context. Without the correct grouping, background clutter and contours from other objects can easily corrupt the useful shape information and prevent global shape reasoning.

5.3.2 Generation of Model Active Descriptors

We first construct a model representation to handle the problem of object articulations.

Model representation. We introduce a tree structured part based model anchored by a collection of joint points. For the articulated human body, the set of joint positions J controls the articulation of the model while the rectangle-like parts remain rigid. An example of this model is shown in Fig. 5.3.

Each model part includes two joint points a, b and a set of contours whose relative positions to these joints are fixed. Therefore each model part appears to be a rigid shape template, described by $P_{ab} = \{C^k(a,b)\}$ where $C^k(a,b)$'s are contours as a function of a, b. The image positions $i(a), j(b)$ of the two joint points uniquely determine a rigid transformation (translation, rotation, and scaling) of the model part. In practice, we found it sufficient to describe object deformation, though more joint points could be added in general.

The collection of joint points $a, b, c, ...$ of all model parts uniquely defines a legal pose if the resulting template is *connected* at joint points. For example, the lower joint point of a thigh has to be hooked with the upper joint point of a leg (at the knee). The model participates in the matching process as a set of contours that compose the parts, which are a function of the compatible configuration of the joint points as shown in Fig. 5.3. We need to clarify that it is not important in which way the contours are fragmented on the model side, as long as all together it composes a legal configuration of joint points. Hence the shape is measured as a whole and all the contours on the model side participate in the matching process.

With the exact model representation, we refine our part configuration selection variable y^k_α in eq. (5.2) to encode the selection of a model part configuration as follows:

$$y^{ab}_{ij} = \begin{cases} 1 & \text{Joint } a \text{ is mapped to image sample point } i \text{ and } b \text{ mapped to } j \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

The model can also be defined as a set of part configurations $\mathcal{M} = \{P_{ab}(i,j) : \ a, b \in J, \ i, j \in S\}$ with J and S being the set of joint points and the set of sample points. The sample points are the possible placement of the model joint points. The set S could be as simple as rectangular grid locations. We would like to select a set of legal one-to-one correspondences between J and S, such that the shape of the model resulting from these configurations is as close as possible to the shape composed by the selected image contours.

Now we are ready to express the holistic shapes by these model part configurations.

Shape Contexts (SC) centered at sample points are chosen as our basic shape descriptors, which is ideal for capturing the bending and rotation of body parts such as limbs. A model contribution matrix V_i^M at sample point i is defined similar to the image contribution matrix V_i^I in eq. (3.4):

$$V_i^M(k,l) = \# \text{ of points in bin } k \text{ from part } P_l \tag{5.4}$$

Recall that the image SC is written as follows in eq. (3.5):

$$sc_i^I(k) = (V_i^I \cdot x^{sel})_k \tag{5.5}$$

It is straightforward to see that SC on model sc_i^M can be generated similar to eq. (3.5), depending on exponentially many combinations of model part configurations:

$$sc_i^M(k) = (V_i^M \cdot y^{part})_k \tag{5.6}$$

We treat y^{part} as a selection vector by concatenating all the joint point selection indicators y_{ij}^{ab} in eq. (5.3).

5.4 Computational Solution for Matching Holistic Features

Our goal is to find x^{sel} and y^{part} such that they produce similar global shape context features at the view points considered. For the model with tree structure defined above, we present an efficient computational solution. The holistic matching of selected image contours and model deformation amounts to minimize the difference between sc_i^I and sc_i^M. This can be summarized by putting eq. (3.5) and eq. (5.6) together:

(Contour Packing LP with Model Selection)

$$\min_{x^{sel},y^{part}} \sum_i D_i(sc_i^I, sc_i^M) = \sum_i \|V_i^I \cdot x^{sel} - V_i^M \cdot y^{part}\| \tag{5.7}$$

$$\text{s.t.} \quad \sum_i z_{ij}^{ab} = \sum_k z_{jk}^{bc}, \ \forall j \in J \ \ (\text{Connectivity between parts}) \tag{5.8}$$

$$\sum_{ij} z_{ij}^{ab} = 1, \ \forall a,b \ \ (\text{Uniqueness of part assignment}) \tag{5.9}$$

The first constraint ensures the connectivity between the neighboring parts of the model. The second constraint ensures that each model part is present. We can relax this constraint to account for possibly occluding or missing parts, essentially introducing selection on the model side. We omit this extension for simplicity.

Direct optimization of the integer programming eq. (5.9) is a hard combinatorial search problem. Basically at each step of the search we need to update our shape descriptors according to the current image contour selection and model deformation and compare them using eq. (5.7). To deal with the combinatorial nature of the problem we relax and solve it using linear programming (LP). Essentially we exploit linear form of shape context descriptors to formulate the holistic matching with contour and part selection. This technique enables us to generate the space of all the combinatorial features via precomputing contribution matrices V^I and V^M.

Discretization via Dynamic Programming (DP). Holistic search using the above computational solution produces sketch-style rough estimation of the poses and locations of joints (see Fig. 5.3). Rounding the linear programming solution of y^{part} directly does not guarantee the selected model parts to be connected. Therefore, we search for assignments of joints to image locations with the largest sum of connections y^{part} while maintaining the model structure. We optimize $\sum_{(a,b)\in J} y_{ij}^{ab}$ where y_{ij}^{ab} is the linear programming solution. Since the model has a tree structure, the optimum can be found by a simple DP.

Our treatment is different from performing pictorial structure directly in two aspects. First, searching for the optimal y^{part} has taken into account the global context beyond pairwise part connections. In contrast, the pairwise cost contains much less information and hence has limited discriminative power. Second we are able to utilize salient image structures such as long contours and large regions despite the semantic gap between them and the model parts. Hence we do not need to design part detector which itself could be a much harder problem than recognizing the whole shape.

Bottom-up driven sampling of joint points. The holistic search of pose should not start purely in a top-down sense, and bottom-up grouping should be exploited as much as possible. Contours and regions are grouped into symmetric ribbons. Therefore, we detect termination points on medial axis of these ribbons as candidates of the protrusion points (e.g. foot). We start sampling all possible locations of other joint points w.r.t these points under part rotation and stretching (see Fig. 5.3). These hypotheses suggest possible model part deformations and they are further verified by the holistic search.

5.5 Experiments

Our approach is tested on a challenging dataset of baseball player images collected from the web as well as the one used in (Mori *et al.* , 2004b). The dataset contains a wide range of pose variations and severe background clutter (see Fig. 5.1 for an example). The combination of these two factors makes pose estimation very challenging.

We start with contour grouping described in Chapter 2. It produces 100 contours for each image on average. Since arms are often missing in the bottom-up contour detection due to occlusion and confusion with background, we use the model containing only head, torso, and lower body with 7 joint points. For this experiment, we take rough bounding boxes as inputs since our focus is pose estimation rather than hypothesis generation. We sample candidates of joints in head, torso and upper leg from grid points in the image. Additional sample joint points are extracted from termination point of medial axis. Each joints have roughly 50 sample points, which will generate $50^7 \times 100 = 7.8^{13}$ hypotheses if brute force search was done. Our linear programming search is efficient: typically 20-30 seconds per images by itself.

We run our method using global shape context without image contour selection and the results are much worse due to overwhelming background clutter. We also test our method using a smaller shape context window without selection. The results are better than the global one without selection but worse than large one with selection. This verifies the importance of holistic matching. Active shape features we introduce are robust against clutter and can accurately recover the correct poses. Our results outperform (Ramanan, 2007) which uses iterative PS, as shown in Fig. 5.4 (d), (e).

5.6 Summary and Future Work

We have presented a holistic shape matching technique with a deformable template for pose estimation and segmentation of articulated objects. We introduce the concept of active context features and present an efficient computational framework for their comparison. We demonstrate results in the baseball dataset but our approach is general enough for any other category of articulated objects. Future work includes the incorporation of additional constraints on model deformation to further restrict the search space and the introduction of part selection on the model side to deal with missing parts due to occlusion. Future work also includes the incorporation of further bottom-up cues like segments to help guide the model deformation.

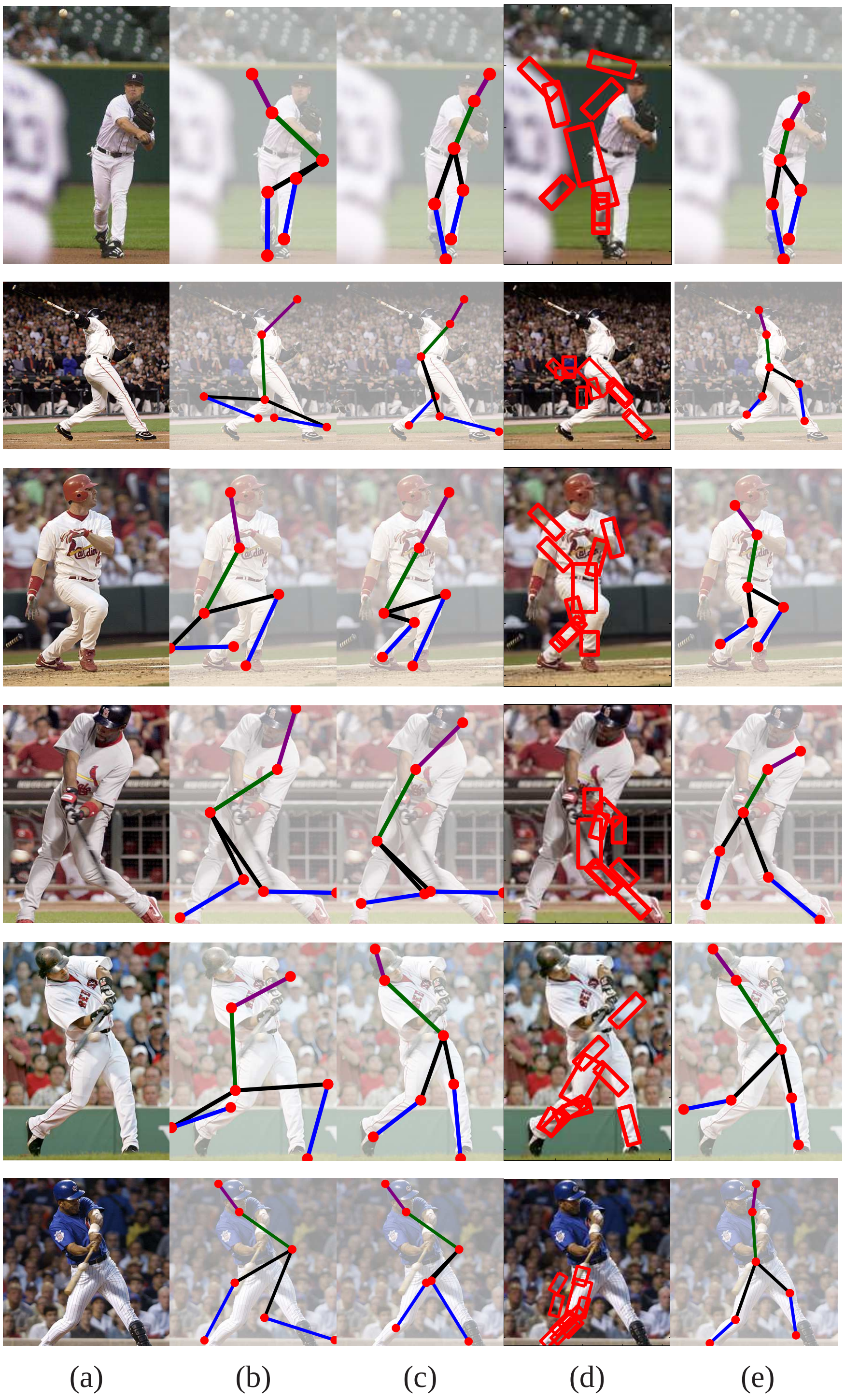


Figure 5.4: Comparison on baseball dataset. Joints with medial axes are displayed on top of the image. Subplots from left to right are: (a) Original image; (b) Results of our approach using large shape context window but without context selection; (c) Results of our approach using a small window again without context selection; (d) Results in (Ramanan, 2007); (e) Results of our approach. Our approach is able to discover the correct rough poses in spite of large pose variations.

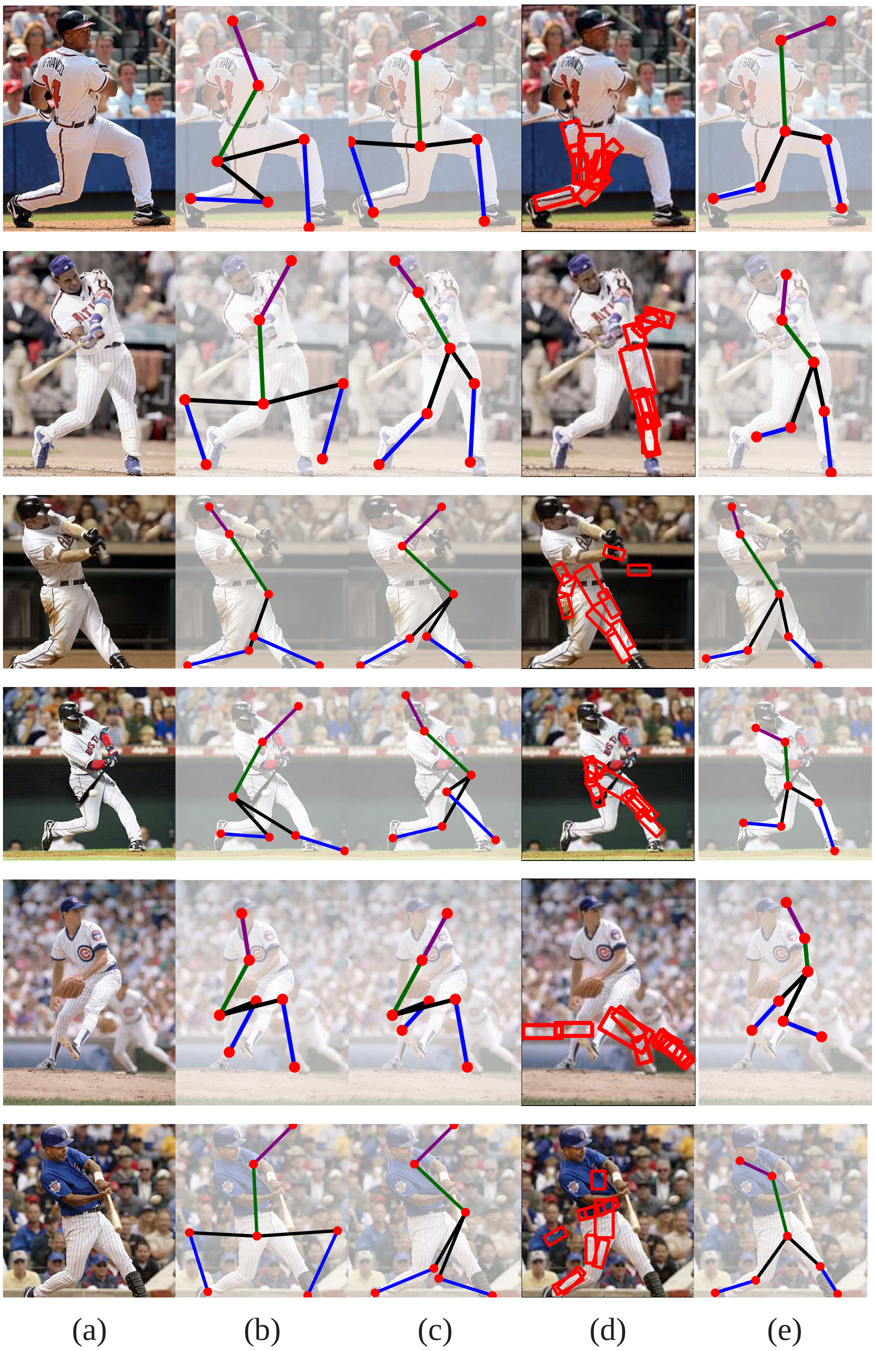


Figure 5.5: More results on baseball dataset. Joints with medial axes are displayed on top of the image. Subplots from left to right are: (a) Original image; (b) Results of our approach using large shape context window but without context selection; (c) Results of our approach using a small window again without context selection; (d) Results in (Ramanan, 2007); (e) Results of our approach. Our approach is able to discover the correct rough poses in spite of large pose variations.

Chapter 6

Region Packing

Salient objects tend to pop out as contiguous *regions* – a group of pixels that delineate themselves from the rest of the image. As a complement to contours, regions play an important role in object detection. First of all, regions convey global shape information which is not available from local image features. Boundaries of regions often contain half complete object silhouettes whose shapes are clearly recognizable. Secondly, unlike contours that could be open ended, regions are closed and therefore specify the figure/ground labeling of the image. The figure/ground segmentation ensures the right spatial support of objects, and blocks irrelevant features from clutter. Thirdly, segmenting the image into regions helps to arouse visual attention to certain objects. Exhaustive search such as scanning the entire image could be avoided by reasoning salient regions and their surroundings.

In this chapter, we develop a packing framework that detects holistic shapes from bottom-up regions, extending contour packing in the previous chapters. Starting from region segments with bags of shape features, we try to pack image and model features into histograms. A subset of regions are matched to the model if they can pack the same set of features as the model. Due to the different topology of regions, the underlying combinatorial problem is relaxed to Semi-Definite Programs (SDP) instead of LPs. This formulation not only tackles the problem of region fragmentations, but is also able to incorporate bottom-up grouping saliency into a unified framework.

6.1 Overview

The importance of regions to object recognition has long been noticed by many researchers (Basri & Jacobs, 1997). Regions along with their boundaries are used extensively to build shape descriptions in medial axis (Blum, 1967), and its successors such as shock graphs (Siddiqi *et al.* , 1999), conformal mapping (Sharon & Mumford, 2006), and Poisson equation based descriptors (Gorelick *et al.* , 2006). Regions provide a global account for object shapes since they are large enough to capture the long-range geometric dependency. They are also shown to be useful for searching and parsing semantical parts (Srinivasan & Shi, 2007), as well as handling object deformation (Ling & Jacobs, 2005). However, all these methods assume that the segmentation of the entire object can be obtained *a priori*, which is rarely the case in detection. The global region-based descriptors change drastically when fragmentations and leakages occur in real images. It is not clear how a shape descriptor can guide the search over exponentially many different segmentations for the desired shape.

Many works based on Bag-of-Features (BoF) exploit regions from bottom-up segmentation as the spatial support of local features (Li *et al.* , 2009; Gupta & Davis, 2008; Galleguillos *et al.* , 2008; Malisiewicz & Efros, 2008). However, geometry as well as object part information is completely missing in BoF. Spatial histogram on local features, *e.g.* HOG (Dalal & Triggs, 2005) has put geometry back to the representation. However, the extraction of these local features is independent of their underlying spatial support. Selecting the right features associated with the foreground relies on discriminative classifiers, which usually requires a large number of training examples. The fixed, rectangular spatial histogram also poses the problem of object alignment. Regions have been used in verifying hypotheses from top-down classifiers in (Wang *et al.* , 2007; Ramanan, 2007), showing the potential of reasoning the spatial support of detection.

Inspired by all the previous approaches, we propose *region packing*, a shape matching method that reasons the holistic shape composed by a set of region segments, and provides an efficient search over their combinations. Region packing bears the same spirit as earlier works that the overall rather than the individual shape of region boundaries should be measured. It incorporates a different shape description than medial axis, *etc*, using spatial histograms with a large spatial extent developed in Chapter 3. This representation enables exploiting the composition and closure of regions, such that combinatorially many segmentations can be encoded compactly, and an efficient search can be performed without enumerating all the hypotheses.

The main technical challenge is the unpredictable fragmentations of region segments. Boundaries between two segments can be either real or fake depending on which segment is foreground. Removing these fake boundaries (and hence merging the regions) is complicated by different fragmentations of images. To overcome this challenge, two recent works (Gu *et al.* , 2009) and (Todorovic & Ahuja, 2008) are most related to our approach. In (Gu *et al.* , 2009), discriminative shape features are learned from some "typical" object segments, and combined in a BoF way. In (Todorovic & Ahuja, 2008), subgraphs in the segmentation hierarchy are explicitly compared during shape matching, which amounts to memorizing all possible different fragmentations. However, structures of these subgraphs might not be repeatable with limited training images. Region packing adjusts shape features according to the set of regions that are merged to form the foreground, and therefore unaffected by fragmentations. Unlike (Gu *et al.* , 2009), we do not assume that individual region segments are simultaneously distinctive and repeatable. We also noticed that regions are not fragmented randomly, hence they should not be merged blindly. The preferences from various bottom-up grouping cues can naturally fit into the framework.

The rest of this chapter is organized as follows. We start with the basic holistic region matching in Section 6.2. This problem is formulated as a bipartite graph packing due to the topology of region. Then we develop an SDP-based approximation which can compactly express bipartite graph packing. In Section 6.3, we show that various grouping cues such as figure/ground, boundary saliency and junction configurations can be readily incorporated into the framework. The proposed approach is tested on the challenging

ETHZ Shape Classes in Section 6.4, producing comparable results to the state-of-the-art region-based methods.

6.2 Holistic Region Matching

The main problem to solve is to match object shapes composed by regions in a holistic way, without knowing which regions belong to foreground. We start by formalizing this problem as follows.

Definition of holistic region matching. Given an image $\mathcal{I}$ and a model $\mathcal{M}$ decomposed into two sets of disjoint regions:

- Image: $\mathcal{I} = R_1^I \cup R_2^I \cup \ldots \cup R_{|\mathcal{I}|}^I$, with R_k^I being the k^{th} region and $R_i^I \cap R_j^I = \emptyset$ for any two regions $i \neq j$;
- Model: $\mathcal{M} = R_1^M \cup R_2^M \cup \ldots \cup R_{|\mathcal{M}|}^M$, with R_l^M being the l^{th} region and $R_i^M \cap R_j^M = \emptyset$ for any two regions $i \neq j$,

we would like to find region subsets $\mathcal{I}^{sel} \subseteq \{R_i^I\}$ and $\mathcal{M}^{sel} \subseteq \{R_i^M\}$, such that their boundary shapes $\mathcal{B}(\mathcal{I}^{sel})$ and $\mathcal{B}(\mathcal{M}^{sel})$ match. Each region R_k^I and R_l^M contains a connected set of pixels. The operator $\mathcal{B}(\cdot)$ is defined as the boundary generated by the mask of a region set. This can be written formally as:

$$\mathcal{B}(\mathcal{R}) = \{x : \ N(x) \cap \bigcup_{R_i \in \mathcal{R}} R_i \neq \emptyset, \ N(x) \cap \mathcal{I} \setminus (\bigcup_{R_i \in \mathcal{R}} R_i) \neq \emptyset\} \tag{6.1}$$

Here x is a pixel and $N(x)$ represents the set of its neighboring pixels (3×3 neighborhood). Since bottom-up region segmentation could also have unpredictable fragmentations that are different from the model (see Fig. 6.1), we adapt the set-to-set matching paradigm developed in Chapter 3 to overcome this representation problem in the following sections.

Before diving into the solutions to the problem, we would like to highlight two key conceptual differences between region packing and contour packing. First, using regions as the basic units in packing exploits *closure*, a stronger constraint than its contour peer: the object boundaries have to be closed. In contrast, a set of open contours could be

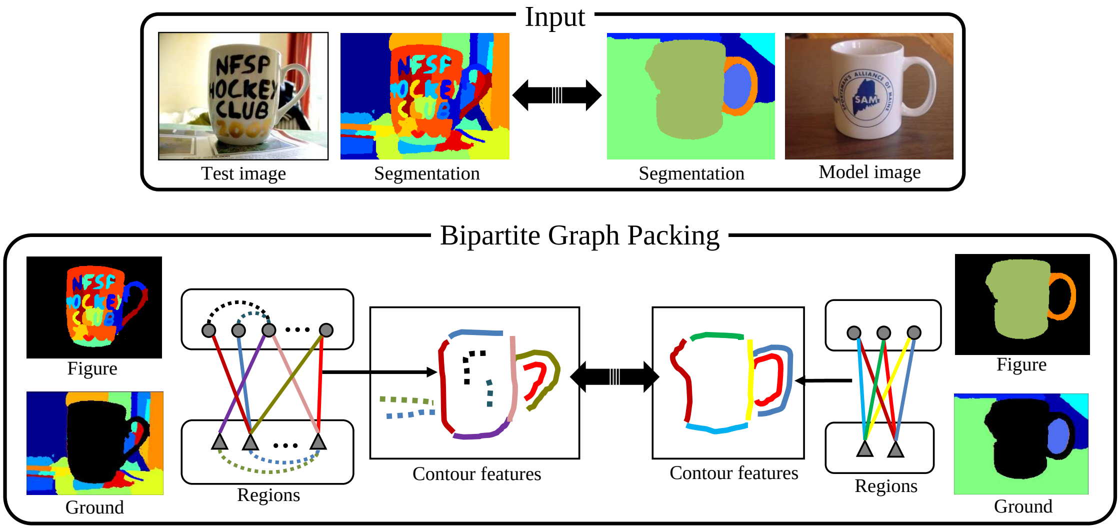


Figure 6.1: Overview of region packing. The first row shows the input image and model with different boundary fragmentations. In the second row, we construct bipartite subgraphs whose nodes are foreground and background regions respectively. The figure/ground partitioning generates bipartite subgraphs, whose edges correspond to boundary fragments (marked with color in the graph). Our goal is to pack these bipartite edges such that the overall shapes from image and model are a good match.

disconnected due to gaps, and susceptible to accidental alignment. Regions rule out this possibility by completing contours into a closed object boundary. Second, regions bind far-away contours that are not linked by bottom-up contour grouping. For example, the contours on the left and the right side of the mug handle can be connected by a region in Fig. 6.1. With these combined contours, ribbon-like shapes become much easier to recognize.

6.2.1 Bipartite Graph Packing

Our goal is to detect a set of object regions whose boundaries form a shape similar to the model. Fundamentally the overall shape of the region set is determined by both of the foreground and background regions. A boundary fragment presents in the shape if and only if exactly one of its two adjacent regions belongs to the foreground. It is this unique topology that brings us to the bipartite graph packing representation.

We consider the following combinatorial problem for holistic region matching:

Definition. Given a graph $G = (V, E)$ where

- Graph nodes $V = \{R_1, R_2, ..., R_n\}$ represent image regions.
- Graph edges $E = \{B_{ij} : \ B_{ij} = \mathcal{B}(R_i) \cap \mathcal{B}(R_j)\}$ correspond to boundary fragments shared by adjacent regions.

Given any partition of regions $V = F \cup \overline{F}$ with F as foreground and $\overline{F}$ as background, we evaluate a shape cost function $C_p(F, \overline{F})$ to measure the shape similarity of boundaries formed by F and $\overline{F}$ compared to the object model. For holistic shape matching, we pose the question: can we find an optimal bipartite subgraph $G_{sub}(F, \overline{F})$ minimizing shape cost $C_p(F, \overline{F})$? We refer to this general problem as *bipartite graph packing* since the cost $C_p(F, \overline{F})$ is determined over a biparitite subgraph.

An appropriate shape cost function $C_p(F, \overline{F})$ plays an important role conceptually and computationally. If there exists one-to-one correspondences between image and model boundaries, one can define $C(F, \overline{F})$ as a linear combination of costs W_{ij} on the edges E_{ij}. Minimizing a linear cost results in standard graph-cut problems (MinCut or MaxCut). Because of the unpredictable fragmentations of image region boundaries (see Fig. 6.1), set-to-set matching on region boundaries arises. A simple linear cost on bipartite graph is insufficient to match the holistic shapes of two set of boundaries. We adopt the Context Selective Shape Features in Chapter 3 as:

$$C_p(F, \overline{F}) = \|V^I \cdot x - sc^M\|_1, \quad x \in \{0, 1\}^{|E|} \tag{6.2}$$

with $x_k = 1$ if and only if edge E_k is a bipartite edge, *i.e.* $E_k \in E(F, \overline{F})$.

The bipartite graph packing with cost eq. (6.2) can be reduced to cardinality constrained and multicriteria cut problems (Bruglieri *et al.* , 2004; Bentz *et al.* , 2009), as stated by the following theorem:

Theorem 6.1. *The bipartite region graph packing problem consists in finding an optimal bipartite subgraph $G_{sub}(F, \overline{F})$ of the region graph G, which minimizes cost $C_p(F, \overline{F})$ defined in eq. (6.2). It can be reduced to a cardinality constrained and multicriteria cut problem on a graph G' associated with R positive edge weight functions $w^{(1)}$,...,$w^{(R)}$ according* *to R criteria. The cardinality constrained and multicriteria cut problem seeks a cut C with cardinality at least d: $\sum_{E_{ij} \in C} 1 \geq d$, and all R criteria are satisfied: $\sum_{E_{ij} \in C} w_{ij}^{(k)} \leq b^{(k)}$ for $k = 1, 2, ..., R$.*

Proof. Please see Appendix for details of the reduction. □

The cardinality constrained and multicriteria multicut problems are in general NP-hard[1], as shown in (Bentz *et al.* , 2009). Therefore, finding a computationally feasible approximation is the key to solve the original problem.

6.2.2 Approximation via Semidefinite Program (SDP)

We seek a relaxation to the above bipartite graph packing formulation via Semidefinite Program (SDP), which has provided polynomial time approximations to many NP-hard problems such as MaxCut (Goemans & Williamson, 1995). In the following sections, we will also demonstrate various constraints such as junction configurations can be conveniently encoded in the SDP formulation.

First we define the region selection indicator $r \in \mathbb{R}^n$ as:

$$\text{(REGION SELECTION INDICATOR)} \quad r_i = \begin{cases} +1, & \text{if region } R_i \in \text{foreground} \\ -1, & \text{otherwise.} \end{cases} \tag{6.3}$$

Note that the definition of r is different from the $0/1$ contour selection indicator in Chapter 3 for simplicity in the subsequent formulation.

Next we introduce a graph indicator matrix $\mathbf{Z} \in \mathbb{R}^{n \times n}$ to be the Gram matrix of the region selection indicator r:

$$\text{(GRAPH INDICATOR)} \qquad Z = rr^{\mathrm{T}} \tag{6.4}$$

Each entry Z_{ij} is also a $+1/-1$ indicator, with the diagonal to be ones: $Z_{ii} = 1$. The graph indicator Z fully characterizes a bipartite subgraph with nodes $F = \{i : \ r_i = 1\}$, $\overline{F} = \{i : \ r_i = -1\}$, and bipartite edges $E(F, \overline{F}) = \{(i,j) : \ Z_{ij} = -1\}$. Moreover, Z is a

[1] However, MinCut which represents a single criteria cut *without* any cardinality contraints, can be solved in polynomial time.

positive semidefinite matrix $Z \succeq 0$ because for any vector u, we have $u^{\mathrm{T}} Z u = u^{\mathrm{T}} r r^{\mathrm{T}} u = (r^{\mathrm{T}} u)^2 \geq 0$. a counterpart of the contour selector, we use a $0/1$ selection indicator x^{sel} to specify figure/ground labels on boundary fragments that are shared by two adjacent regions. These boundary fragments serve as the basic building blocks of the object shapes just as contours in Chapter 3. Boundary fragments behave differently than contours in that they can only be packed if exactly one of its two adjacent regions appears as foreground, *i.e.*

$$x_k^{sel} = 1 \Leftrightarrow (r_i = 1 \wedge r_j = -1) \vee (r_i = -1 \wedge r_j = 1) \tag{6.5}$$

This constraint can be rephrased in terms of **Z**: $(1 - Z_{ij})/2 = x_k^{sel}$ since $Z_{ij} = r_i r_j$.

The overall shape composed by selected regions needs to be holistically matched to the model shape. We adopt the contour packing cost eq. (3.7) as the packing function $C_p(F, \overline{F})$ on bipartite edges, measuring the shape dissimilarity of a set of boundary fragments generated by the selected regions (F). For each control point correspondence, the shape dissimilarity $\|V^I \cdot x^{sel} - sc^M\|_1$ depends on which boundary fragments are selected by x^{sel}, with the contribution matrix of boundary fragments V^I precomputed. We summarize all the above components into the following SDP:

(REGION SELECTION SDP)

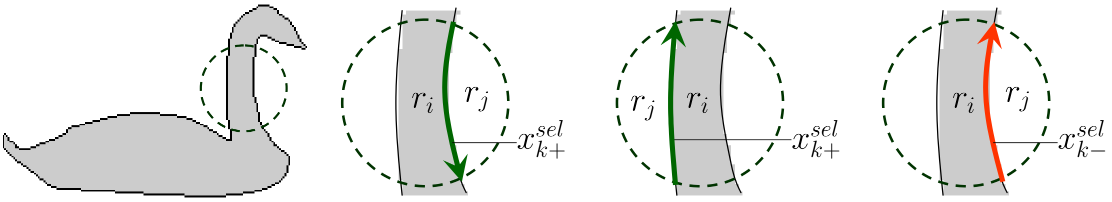
$$\max_{Z,\, x^{sel}} \quad \|V^I \cdot x^{sel} - sc^M\|_1 \tag{6.6}$$

$$\text{s.t.} \quad \frac{1 - Z_{ij}}{2} = x_k^{sel}, \;\; \forall R_i, R_j \text{ separated by fragment } k \tag{6.7}$$

$$\operatorname{diag}(Z) = 1 \tag{6.8}$$

$$Z \succeq 0 \tag{6.9}$$

If the rank of matrix **Z** is 1, the optimal SDP solution is exactly the optimum of bipartite graph packing. The non-convex constraint $\operatorname{rank}(\mathbf{Z}) = 1$ is dropped to obtain an SDP problem, which is solvable by off-the-shelf SDP packages. After solving the optimal graph matrix Z^*, we recover r by computing its largest eigenvector. A binary selection on regions can be obtained by thresholding the continuous eigenvector.



(a) Object boundary (b) True boundary 1 (c) True boundary 2 (d) False boundary

Figure 6.2: Figure/ground labeling on boundaries. The boundary of a swan along with its foreground region is shown in (a). In the circled area, different figure/ground configurations exist and need to be distinguished. Two true boundaries with opposite directions in (b) and (c) appear due to the parallelism. (d) shows a false boundary with incorrect figure/ground labeling.

For the convenience of further discussion, we introduce a vectorization operator $\mathrm{svec} : \mathcal{S}^n \mapsto \mathbb{R}^{n(n+1)/2}$ on the symmetric matrix $Z \in \mathcal{S}^n$ as:

$$\mathrm{svec}(Z) = [Z_{11}, \sqrt{2}Z_{12}, Z_{22}, ..., \sqrt{2}Z_{(n-1)n}, Z_{nn}]^{\mathrm{T}} \tag{6.10}$$

An important property of the operator svec is that it translates matrix inner product into vector inner product: $\mathrm{tr}(YZ) = \mathrm{svec}(Y)^{\mathrm{T}}\mathrm{svec}(Z)$. This allows us to define a transformation matrix $T \in \mathbb{R}^{m \times \frac{n(n+1)}{2}}$ to represent all the linear constraints in eq. (6.7) such that $T \cdot \mathrm{svec}(Z) = x^{sel}$. Note that since $Z_{ii} = 1$, every entry $x^{sel}_k = \frac{1-Z_{ij}}{2}$ in eq. (6.7) can be written as a linear form in $\mathrm{svec}(Z)$. With the above notations, region selection eq. (6.6) can be expressed more compactly as:

$$\max_{Z} \ \|V^I \cdot T \cdot \mathrm{svec}(Z) - sc^M\|_1 \tag{6.11}$$

$$\mathrm{s.t.} \ \ \mathrm{diag}(Z) = 1, \ \ Z \succeq 0$$

6.3 Representing Grouping Constraints

Expressing bipartite graph packing in a SDP form enables several important extensions to bottom-up grouping constraints.

6.3.1 Figure/Ground

Up to this point we have not taken into account the figure/ground labeling of boundary fragments. Selection of a boundary fragment does not specify which side of the fragment belongs to the foreground object. In eq. (6.6), flipping the region indicator from r to $-r$ produces the same Z, and hence does not change the packing cost. This means foreground and background are exchangeable for region packing. To remedy this problem, we add a fictitious node $r_0 = 1$ to represent the foreground. Any regions partitioned to the same side as the fictitious node will be labeled as foreground. This amendment adds one row and one column to **Z** with $Z_{i0} = r_i$. Accordingly, boundary fragments become directional: the foreground region is always located on the right side of the boundary. The boundary selection indicators are split into two copies $x_k^{sel} = x_{k+}^{sel} + x_{k-}^{sel}$ defined as follows:

$$x_{k+}^{sel} = (r_i = 1) \wedge (r_j = -1) = \frac{r_i + 1}{2} \cdot \frac{1 - r_j}{2} = \frac{Z_{i0} - Z_{j0} - Z_{ij} + 1}{4} \tag{6.12}$$

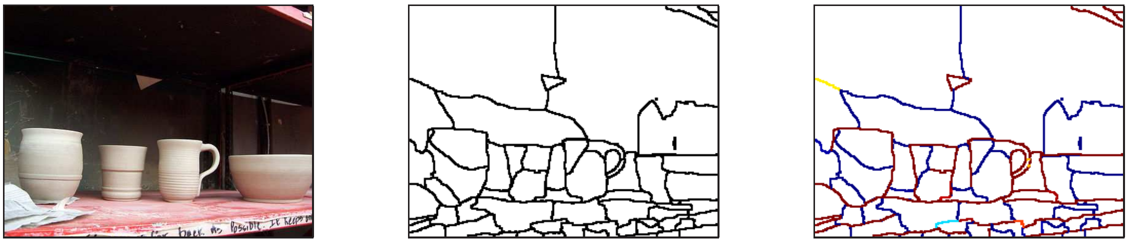
$$x_{k-}^{sel} = (r_i = -1) \wedge (r_j = 1) = \frac{1 - r_i}{2} \cdot \frac{r_j + 1}{2} = \frac{-Z_{i0} + Z_{j0} - Z_{ij} + 1}{4} \tag{6.13}$$

Indices i, j, $k+$, $k-$ are organized in the following way. When traveling along the direction of $k+$, the positive one of r_i, r_j (foreground region) lies on the right side of the boundary; it lies on the left side when traveling along $k-$ (see Fig. 6.3).

The shape features also need changes to be compatible for the figure/ground specification. We split each edge orientation bin of shape context into two bins, encoding edges pointing opposite directions. Now the contributions of x_{k+}^{sel} and x_{k-}^{sel} to the shape descriptors are separated, and therefore a mismatch of figure/ground will be penalized.

6.3.2 Boundary Saliency

True objects not only match model well, but pop out from the background. Saliency of segmentation can reduce many false positives by penalizing randomly packed segments, and favoring segments that can be easily cut out of the background (see Fig. 6.3.2). Therefore, we introduce region grouping edges E_g, whose weights encode how well the regions



(a) Original Image (b) Segmentation with 60 regions (c) Boundary saliency

Figure 6.3: Binary region boundaries alone are insufficient to pop up object shapes. (a) shows an image containing mugs and bowls clearly discernible from background. Restricted to binary region boundaries in (b), objects are surrounded by fake boundaries in the background (lower part of the image), and hence become less salient. In (c), boundary saliency helps to re-group over-fragmentations of objects. Segmentation boundaries are colored by strengths from low (blue) to high (red).

can be grouped together. We denote the bipartite edges previously defined for packing as $E = E_p$. The two different types of edges, packing edges E_p and region grouping edges E_g, encode independent information: one for the global shape similarity to the top-down model, and one for the saliency from bottom-up grouping.

Our goal is to minimize the cost $C_p(F, \overline{F})$ over the packing edges and $C_g(F, \overline{F})$ over the region grouping edges simultaneously, with both defined on the bipartite subgraph $(F, \overline{F})$. The cost $C_g(F, \overline{F})$ is represented as the cut between F and $\overline{F}$ in the graph as in the graph partitioning framework such as NCut (Shi & Malik, 2000). In terms of graph indicator matrix **Z**, the cut cost $C_g(F, \overline{F})$ can be written as $\mathrm{tr}(W_g \cdot Z)$ where W_g is the weight matrix of the region grouping edges. As well known in graph partitioning, the cut cost alone biases on “shorter” boundaries (Shi & Malik, 2000) and smaller regions. We introduce a normalization factor $D_p(F, \overline{F}) = 1^{\mathrm{T}} \cdot V^I \cdot T \cdot \mathrm{svec}(Z)$ analogous to the degree in the graph partitioning setting. The normalization factor measures the total length of selected boundaries, and hence approaches 0 if no foreground regions are selected. In summary, we would like to optimize the following cost which combines packing and grouping:

$$C_{p+g}(F,\overline{F}) = \frac{C_p(F,\overline{F}) + C_g(F,\overline{F})}{D_p(F,\overline{F})} \tag{6.14}$$

$$= \frac{\|V^I \cdot T \cdot \mathrm{svec}(Z) - sc^M\|_1 + \beta \cdot \mathrm{svec}(W_g)^{\mathrm{T}}\mathrm{svec}(Z)}{1^{\mathrm{T}} \cdot V^I \cdot T \cdot \mathrm{svec}(Z)} \tag{6.15}$$

In spite of the normalization, the optimization problem eq. (6.15) can still be formulated as SDP by introducing a normalized matrix $Y = Z/[1^{\mathrm{T}}V^IT \cdot \mathrm{svec}(Z)]$. Because the normalization factor $1^{\mathrm{T}}V^IT \cdot \mathrm{svec}(Z) > 0$, the matrix Y is also positive semidefinite, resulting in the following SDP:

$$\max_{Y} \; \|V^I \cdot T \cdot \mathrm{svec}(Y) - sc^M \cdot Y_{11}\|_1 + \beta \cdot \mathrm{svec}(W_g)^{\mathrm{T}}\mathrm{svec}(Y) \tag{6.16}$$

$$\text{s.t.} \; 1^{\mathrm{T}}V^IT \cdot \mathrm{svec}(Y) = 1 \tag{6.17}$$

$$\mathrm{diag}(Y) = Y_{11} \tag{6.18}$$

$$Y \succeq 0 \tag{6.19}$$

Since we construct the graphs on the region segments rather than image pixels, grouping weights W_g directly include global grouping saliency. The weight $W_g(i,j)$ between region segment r_i and r_j are computed by:

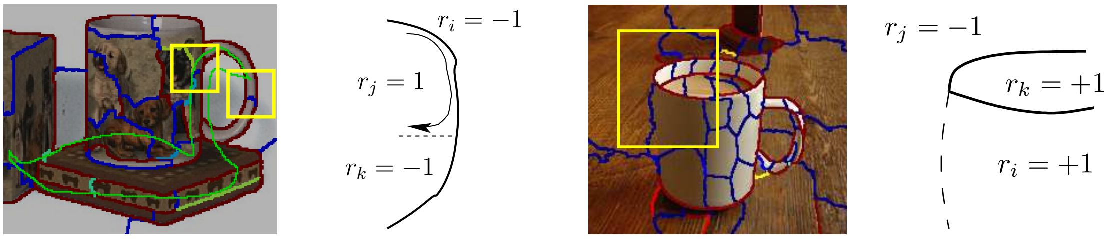
$$W_g(i,j) = \exp(-\frac{d_{ij}^2}{2\sigma^2})|Cut(r_i, r_j)| \tag{6.20}$$

where $|Cut(r_i, r_j)|$ is defined as the boundary length between the two segments. The term d_{ij} is the geodesic distance in the eigenvector embedding space of NCuts between cluster centers of r_i and r_j. The geodesic distance computes the shortest path distance on weights defined as the point density in the embedding space formed by eigenvectors. This measures how well the two regions can be separated. We would like to pointed out the advantage of defining W_g on the output of segmentation rather than original edge magnitude, which makes the overall cost insensitive to image contrast changes. Moreover, because entries in W_g are normalized by the corresponding boundary lengths, the three terms $C_P(F,\overline{F})$, $C_G(F,\overline{F})$, and $D_P(F,\overline{F})$ in eq. (6.14) are balanced.

6.3.3 Junction Configurations

Over-segmentation of regions can cause many false positives. In the case of over-segmentation, the selection on region boundary fragments has too much freedom – the selected boundaries can easily hallucinate a model shape by making arbitrary turns. Boundary saliency cost avoids fake boundaries to some extent, but the additive penalty in eq. (6.20) loses its power when the fake boundaries are short. Fig. 6.4(a) shows a typical example. The shortcut at the boundary fragment on the mug handle enables a false detection. The selection on the boundary fragment only pays a small penalty, yet has a significant effect on the overall shape structure.

Junctions formed by several adjacent regions are good places to inspect. We have noticed that the undesired shortcut usually occurs at junctions formed by two salient boundaries and one weak boundary (see Fig. 6.4(b)), This indicates that the two regions separated by the weak boundary tend to merge in the coarser level of segmentation. Restricting the region selection not to segment the two regions may reduce many false positives. This



(a) Image I (b) Invalid junction (c) Image II (d) Valid junction

Figure 6.4: Illustration of the junction configuration and a false positive. (a) shows an accidental alignment of the swan, where the region boundaries make a wrong turn without paying large penalties (marked in yellow rectangles). The boundary strengths computed by eq. (6.20) are also displayed on the figure, increasing from blue to red. A schematic diagram of regions is shown in (b). Region packing only chooses region r_j ($+1/-1$ means foreground/background). This creates an incorrect boundary fragment and makes the strong boundary leak to the background. Note that a strong boundary leaking to the foreground is very likely due to a salient object part (top part of the mug in (c), (d)), and a weak object boundary.

grouping cue is asymmetric for figure and ground. The strong boundaries are more likely to extend to the foreground when it surrounds a salient object part (see Fig. 6.4(c)), than leak to the background. Leakage to the background could occur if a salient object in the background is occluded by another object with a weak boundary. But in practice this scenario is very rare.

This figure/ground constraint can be written as a logic statement on the neighboring regions. Let $r_i, r_j, r_k \in \{\pm 1\}$ be the selection indicators on the incident regions at the junction, with regions R_j, R_k separated by a weak boundary fragment. Then a valid configuration satisfies:

$$(r_i = -1) \Rightarrow (r_j = r_k) \tag{6.21}$$

The above logic statement rules out cases where $r_i = 1$ and exactly one of r_j and r_k belongs to the background ($r_j \neq r_k$), implying the strong boundary leaks to the background. Expressed by the graph indicator Z, this becomes a simple linear constraint:

$$Z_{0i} + Z_{jk} \geq 0 \tag{6.22}$$

An alternative to the above constraint is to utilize the cue in the cost function. This can be done by adding slack variables to eq. (6.22) and minimizing the sum of these slacks in addition to the original cost.

Generally, other types checking on junction configurations are possible. Any cost function involving a 2-CNF (conjunctive normal form) logic statement over the regions can be tightly encoded in SDP (Goemans & Williamson, 1995), since Z_{ij} and $1 - Z_{i0}$ represent XOR and NOT logic respectively. Higher order CNFs can always decomposed into 2-CNF via auxiliary variables, but with weaker relaxations and more expensive computations.

6.4 Experiments

Region packing is demonstrated by detection using only shape features on ETHZ Shape Classes (Ferrari *et al.* , 2007a). A similar experimental setup as Chapter 3 is adopted for this task.

6.4.1 Implementation

We start with region segmentation from multi-scale Normalized Cuts (Cour *et al.* , 2005). Boundary saliency of regions defined in Section 6.3 is used in addition to binary region boundaries. For the finest scale of detection, 60 segments are used for region packing to capture small objects. The number of segments are inversely proportional to the detection scale, down to 30 segments for the coarsest scale. The large window shape context descriptor consists of 12 polar angles, 5 radial bins and 8 edge orientations. Note that edge orientations different by π encode the same boundary fragments with opposite figure/ground labels. Hence the number of edge orientations is doubled compared to the one in contour packing.

We generate object hypotheses by a voting process. Control points are uniformly sampled on image region boundaries as well as the model shape boundary. The correspondences of these control points give alignment of the model shape to the image. The spatial extent of regions gives great advantages on the search over the correspondences. Regions which have a signification portion of boundary outside the object bounding box can be pruned. Selection on the leftover segments can be evaluated exhaustively if their number is small (≤ 12). This enables reduction of correspondence hypothesis evaluation from around 4000 down to under 500 on average per scale. For each remaining correspondence, we use the publicly available solver SeDuMi (Sturm, 1999) to compute the SDP solution in eq. (6.6). To adapt to scale variance, voting of object centers is performed in 5 to 7 scales for each category. After identifying object center hypotheses from the voting map, regions are selected jointly across all correspondences that agree on the object center, similar to eq. (3.12). The final region packing cost is computed using these consistently selected foreground regions.

Region boundaries do not contribute equally to the holistic object shape – some parts are more salient than the others. For example, the handle of the mug is critical for recognizing its shape. The region packing cost from different control points and shape context bins should reflect this distinction. We borrow the idea from latent SVM (Felzenszwalb *et al.* , 2008) to learn shape feature weights that are most discriminative for classifying positives and negatives. The feature weights are defined on under-packed and over-packed values b^+, b^- at each bin. Note that b^+, b^- depend on the region selection. We learn the weights in a coordinate descent way which optimizes feature weights and region selections alternatively. The feature weights are optimized by:

$$\begin{aligned} \min_{w=(w^+;w^-)} \quad & \frac{1}{2}\|w\|^2 + C\sum_j \xi_j \\ \text{s.t.} \quad & y_j \cdot [(w^+)^{\mathrm{T}} b_j^+ + (w^-)^{\mathrm{T}} b_j^-] \geq 1 - \xi_j \\ & w^+,\ w^- \geq 0 \end{aligned} \tag{6.23}$$

The iterations converge in 3 to 5 steps. We split the dataset into training and test set in the following way. For each category, half of the positive images are used for training, with the other half for testing. The same number of negative images are added to the training set, sampled uniformly from the other 4 negative categories.

6.4.2 Quantitative Comparison

We quantitatively evaluate the performance of region packing and compare with state-of-the-art via Precision vs. Recall (P/R) curve [2]. Region packing achieves overall results superior or on par with the previous state-of-the-art works (Maji & Malik, 2009; Gu *et al.* , 2009; Felzenszwalb *et al.* , 2008; Lu *et al.* , 2009). Table 6.1 summarizes the Average Precision (AP) on each category and the whole dataset. Among these works, (Gu *et al.* , 2009) is most related to our approach since it is also region-based. Unlike (Gu *et al.* , 2009) which has texture and color features in addition to shape, region packing only uses shape feature. This shows that our framework does capture the global shape of region segments despite different fragmentations, because shape alone on individual segments is not distinctive. If necessary, other features such as texture and color can be incorporated to region packing in the same way. Also we would like to point it out that our training set

[2] We choose Precision vs. Recall (P/R) instead of Detection Rate vs. False Positive Per Image (DR/FPPI) because DR/FPPI depends on the ratio of the number of positive and negative test images and hence could introduce bias to the measure.

	Applelogos	Bottles	Giraffes	Mugs	Swans	**Average**
Region Packing†	0.866	0.902	0.715	0.786	0.730	0.800
Region Packing (50% split)§	0.878	0.908	0.772	0.829	0.890	0.855
(Srinivasan *et al.* , 2010)	0.845	0.916	0.787	0.888	0.922	0.872
(Toshev *et al.* , 2010)	0.983	0.936	0.713	0.718	0.973	0.865
(Maji & Malik, 2009)	0.869	0.724	0.742	0.806	0.716	0.771
(Gu *et al.* , 2009)	0.772	0.906	0.742	0.760	0.606	0.757
(Lu *et al.* , 2009)	0.844	0.641	0.617	0.643	0.798	0.709
(Felzenszwalb *et al.* , 2008)	0.891	0.950	0.608	0.721	0.391	0.712

Table 6.1: Comparison of region packing and the latest shape detection works on average precision (AP). †: Same train/test split as (Srinivasan *et al.* 2010), *i.e.* taking 50% positives as training examples, with the same number of negatives randomly sampled from other categories. §: Same region packing algorithm as †, but split train/test as (Toshev *et al.* 2010), which includes 50% images as training set (larger than (Srinivasan *et al.* 2010)).

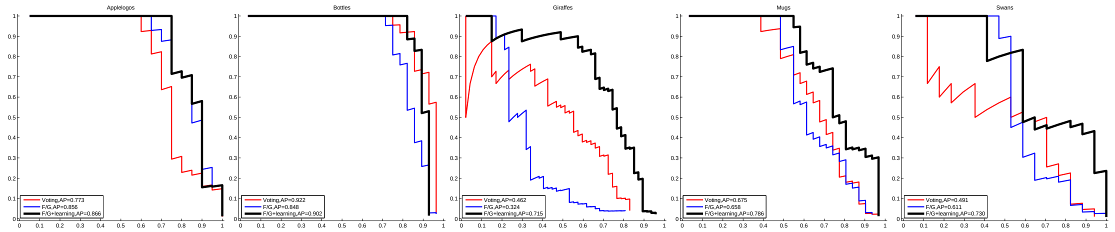
is smaller than (Gu *et al.* , 2009) (but the same as (Maji & Malik, 2009)), containing fewer negative and the same number of positives. This means that region packing will have better P/R if the train/test split follows (Gu *et al.* , 2009). The recent work of (Srinivasan *et al.* , 2010) uses contour packing presented in Chapter 3, but with discriminative SVM training. Contours give a strong boost to objects with elongated structures such as Swans and hence outperform its region counterpart (see Table 6.1). Also it includes an extra refinement stage on control point correspondences to better handle large object deformations, such as aspect changes (Mugs) and articulations (Swans and Giraffes).

Region packing presented in this chapter is conceptually similar to boundary structure segmentation in (Toshev *et al.* , 2010), but developed independently. Both approaches leverage regions as integral tokens for object shape recognition and match region boundaries using holistic shape features. Computationally Semidefinite Programs (SDP) provide an approximated solution to the combinatorial matching problem. Also both uses SVM on top of holistic shape matching to boost the discriminative power of the shape descriptor.

The major differences between the two methods are: 1) the boundary feature in (Toshev *et al.* , 2010) is a correspondence-less spatial histogram, while shape contexts in region packing depend on the correspondence of the center point. Compared to shape contexts, the boundary feature in (Toshev *et al.* , 2010) imposes a coarser binning to the spatial relationship of contour points. Hence it has the advantage of efficient detection without the burden of an explicit correspondence search. On the other hand, its discriminative power on shape can be limited because unrelated pairwise spatial relations can fall into the same bin. 2) our region packing feature does not include local edge contrast as in (Toshev *et al.* , 2010), which is sensitive to specific datasets. Note that the embedding distance in Section 6.3.2 is a global boundary measure rather than a local one, and immune to image contrast change. Due to the common philosophy and algorithm design, the two methods achieve comparable results on ETHZ dataset with the same train/test split, as shown in Table 6.1.

Region packing successfully identifies the correct figure/ground selection in most images (see Fig. 6.6, Fig. 6.7, Fig. 6.8, Fig. 6.9, Fig. 6.10 for top detections). The selected foreground regions generate a boundary shape that is visually similar to the target shape, and follows the grouping preference as well. In several cases such as bottles and mugs, regions break into many segments with complicated shapes due to interior marking of the objects. Local shapes are insufficient to choose the right foreground, and reasoning boundary continuity is easily confused by numerous junctions. Typical false positives have similar global shape to the model, but lacking the right detailed shapes, or violating region connectivity. We expect a significant improvement if refinement on the correspondence search and detailed shape matching is employed. Most misses occur due to large shape deformations as shown in Fig. 6.11.

We also tested influences of different components in region packing in Table 6.2 and Fig. 6.5. Latent SVM learning significantly improves the average AP from 0.665 (voting only) and 0.659 (with grouping cue) to 0.800 (with both). Note that the figure/ground group cue could hurt the precision for deformable objects such as Giraffes. However,



(a) Applelogos (b) Bottles (c) Giraffes (d) Mugs (e) Swans

Figure 6.5: Precision vs. Recall curves (PR).

	w/o SVM	w/o grouping	Full system
Overall AP	0.665	0.659	0.800

Table 6.2: The effect of different factors in region packing.

since the constraint regularizes the region selection, it makes learning feature weights easier and hence gain significant boost after training.

6.5 Summary

In this chapter, we have proposed a novel feature packing framework using bottom-up regions to recognize shapes. Starting from fragmented regions, we try to assemble a subset of them into the model shape such that their overall boundary shapes are similar. A subset of regions are holistically matched to the model if they can pack the same set of shape boundary features as the model. Due to the topological relationship between regions and their boundaries, the holistic shape matching is formulated as a bipartite graph packing problem. The combinatorial search of bipartite graph packing can be approximated and solved efficiently via SDP. We extend the formulation to incorporate various grouping cues, and unify all these components in the graph partitioning setting. The framework has shown results on ETHZ Shape Classes comparable with the state-of-the-art region-based methods, with less reliance on features other than shape. The promising results are largely attributed to the ability to overcome arbitrary region fragmentation and utilize region-based grouping cues.

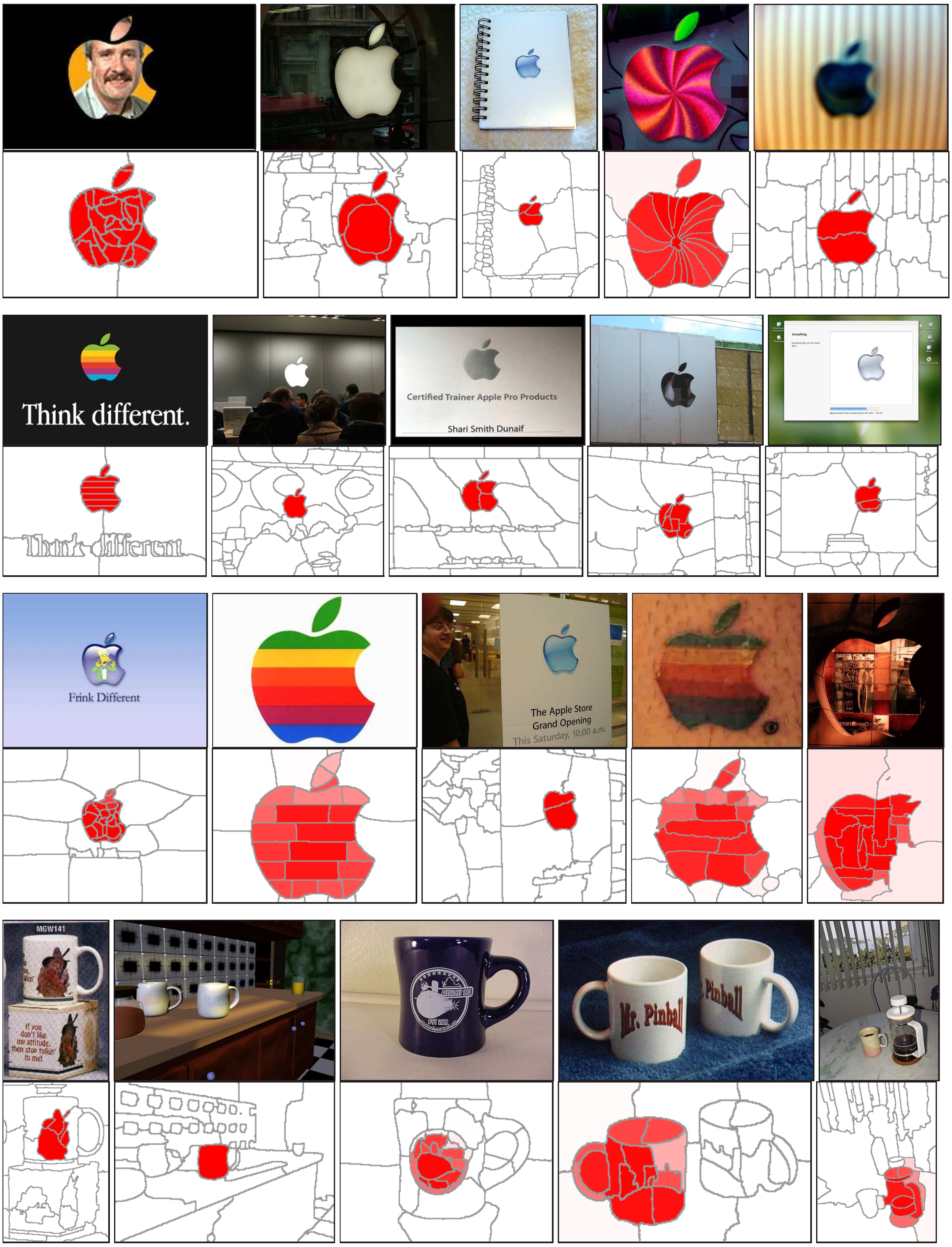

Think different.
Certified Trainer Apple Pro Products
Shari Smith Dunaif
Frink Different
The Apple Store
Grand Opening
This Saturday, 10:00 a.m.
MGW141
If you don't like my attitude, then stop talkin' to me!
Mr. Pinball
Pinball

Figure 6.6: Top 20 detections on Applelogos. Detections are sorted by scores from high to low. The continuous values of region selection indicator are colored on the corresponding regions from white (-1) to red (1).

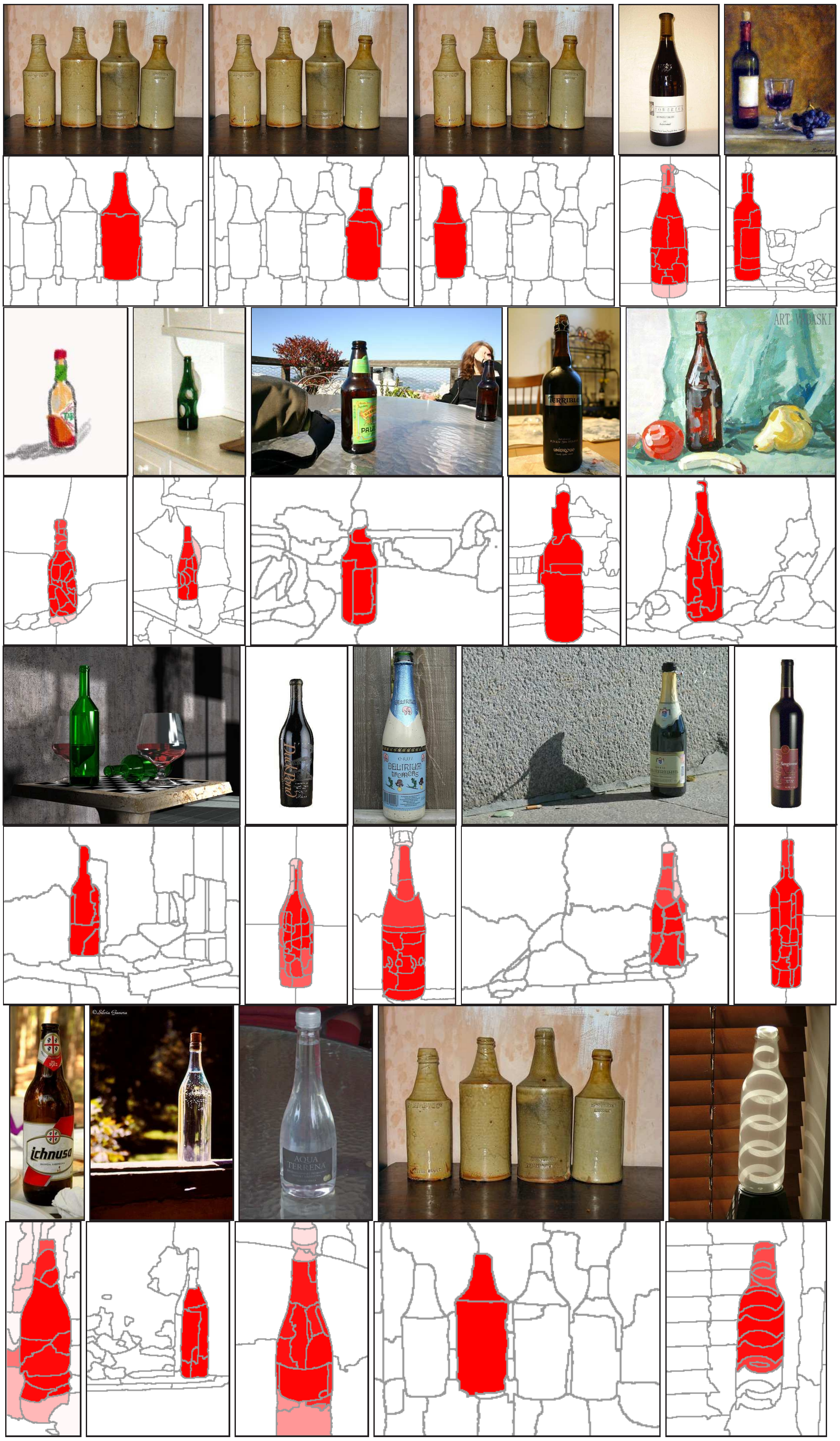


Figure 6.7: Top 20 detections for Bottles. Detections are sorted by scores from high to low. The continuous values of region selection indicator are colored on the corresponding regions from white (-1) to red (1).

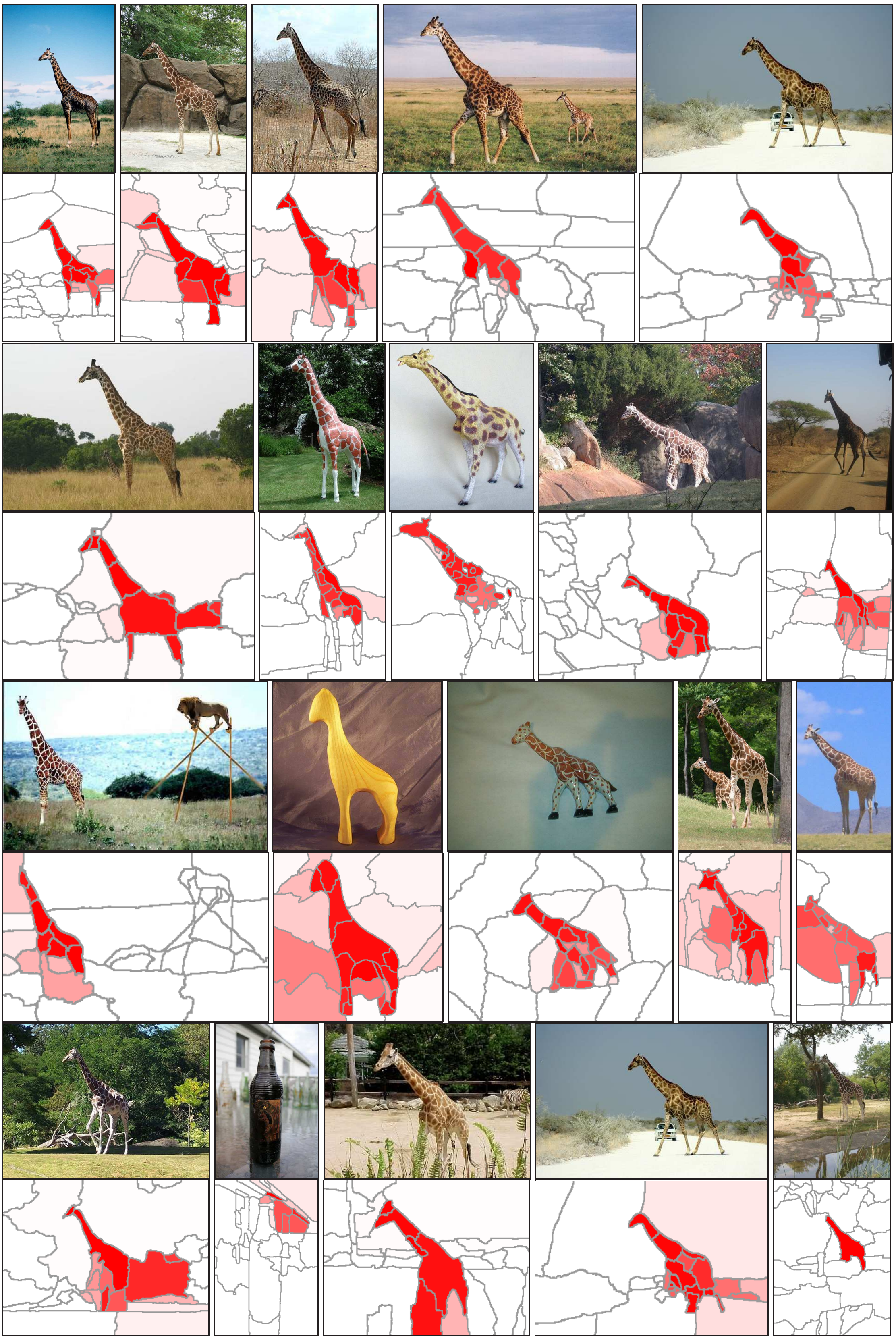


Figure 6.8: Top 20 detections for Giraffes. Detections are sorted by scores from high to low. The continuous values of region selection indicator are colored on the corresponding regions from white (-1) to red (1).

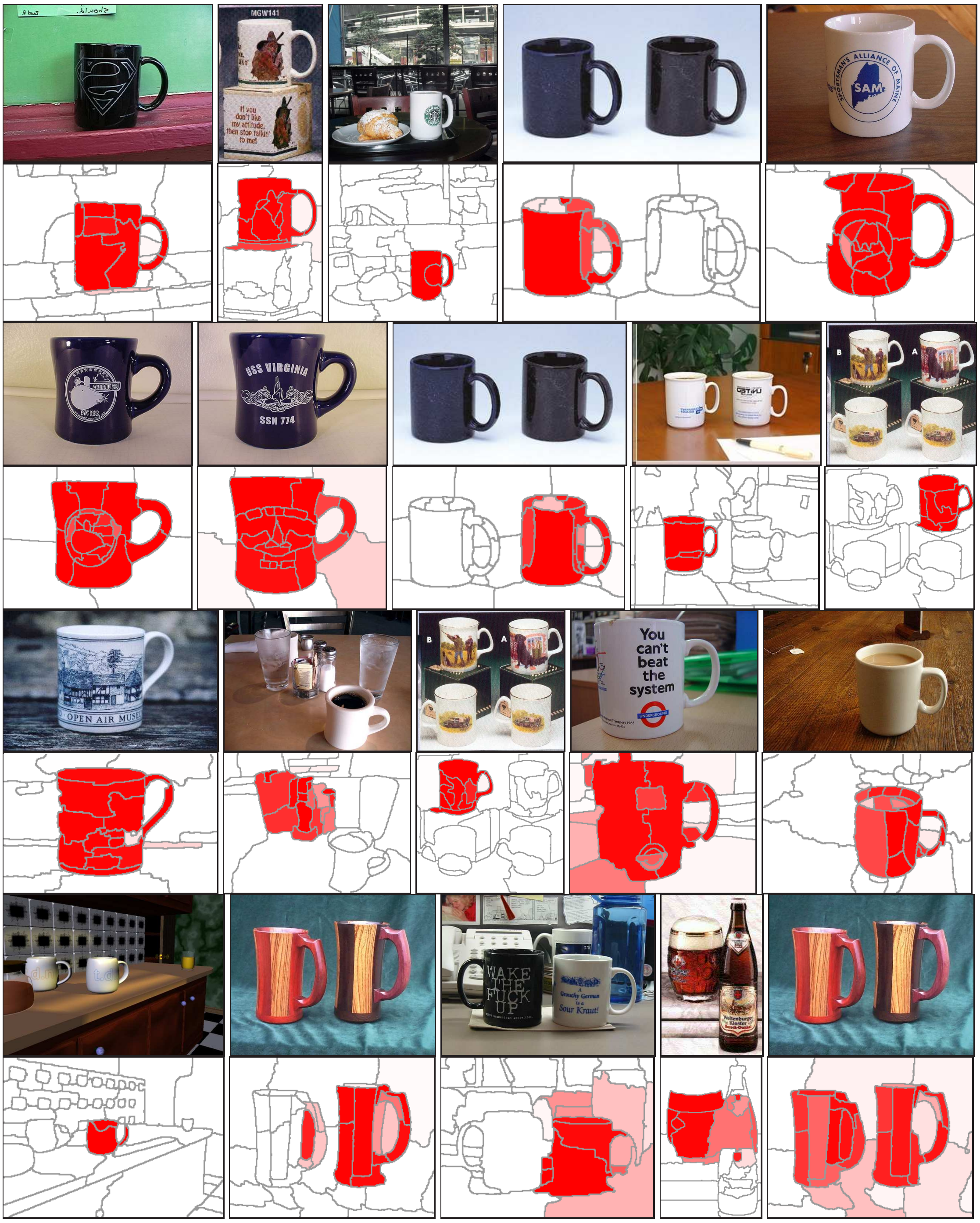

MGW141
If you don't like my attitude, then stop talkin' to me!
SAM
USS VIRGINIA
SSN 774
OPEN AIR MUSE
You can't beat the system
WAKE THE FUCK UP
Sour Kraut!

Figure 6.9: Top 20 detections for Mugs. Detections are sorted by scores from high to low. The continuous values of region selection indicator are colored on the corresponding regions from white (-1) to red (1).

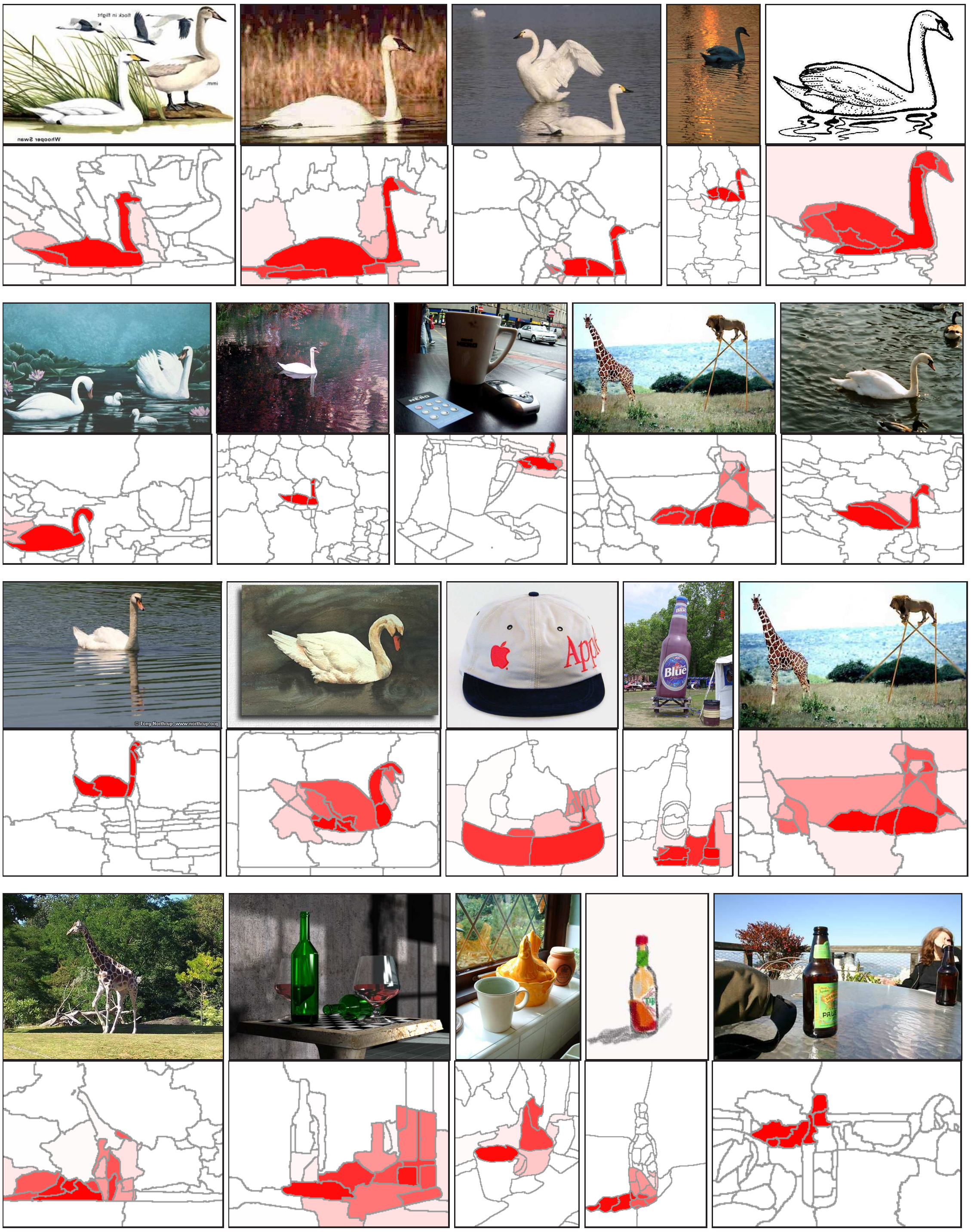
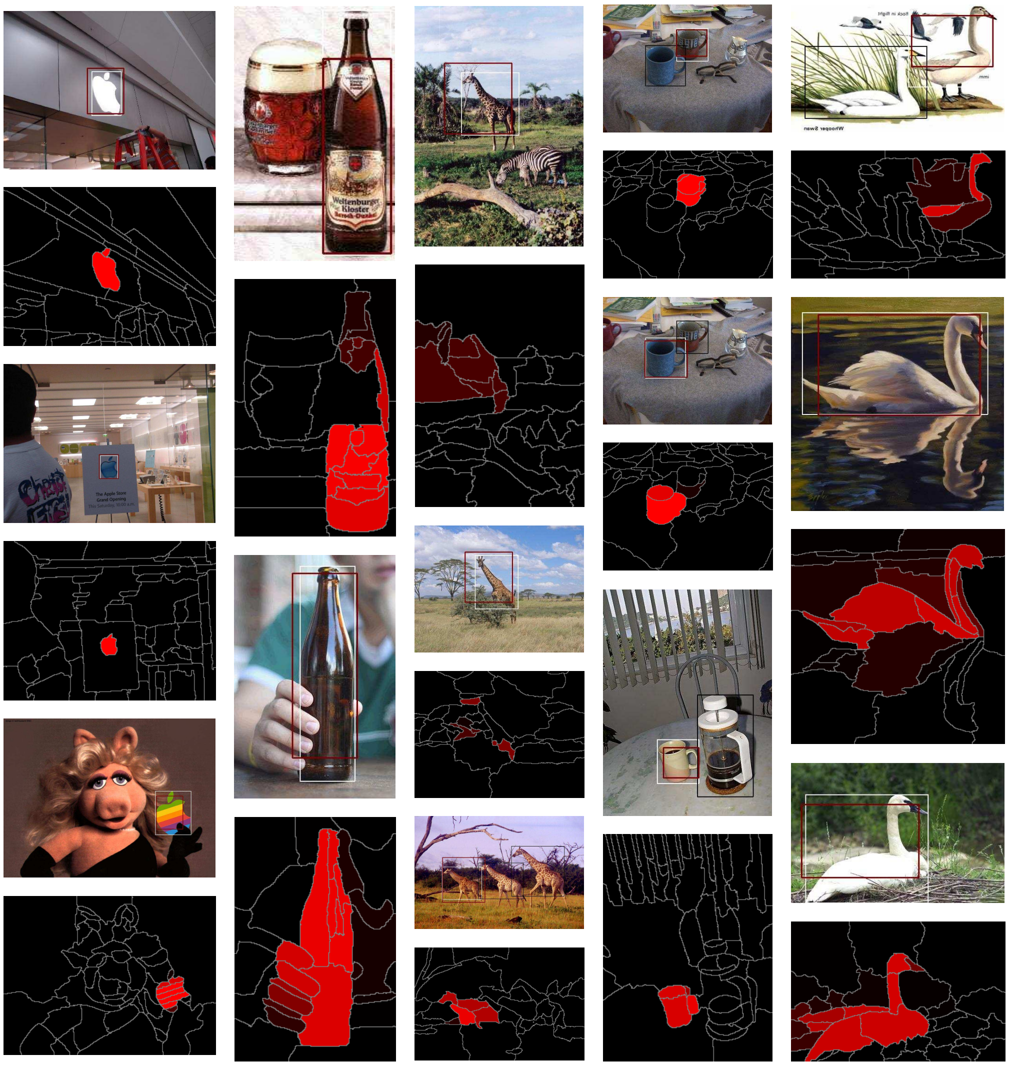


Figure 6.10: Top 20 detections for Swans. Detections are sorted by scores from high to low. The continuous values of region selection indicator are colored on the corresponding regions from white (-1) to red (1).



(a) Applelogos (b) Bottles (c) Giraffes (d) Mugs (e) Swans

Figure 6.11: Typical misses for all five categories. True positives with the lowest scores. The figures are sorted by score in ascending order from top to bottom.

Chapter 7

Conclusion

Exploiting global contexts to detect and recognize complex patterns while keeping the search computationally tractable has been a fundamental issue not only in computer vision, but also in the broad area of artificial intelligence. In this thesis, we consider this problem in the setting of detecting shapes from natural images with various complexities. Unlike other patterns such as textures which may be locally recognizable, shape is typically perceived as a whole – it is fundamentally about the global geometric arrangement of a set of entities. With few distinctive local shape features, reasoning on individual entities without examining their surroundings is bound to be unreliable.

Traditional contextual models such as Markov Random Fields (MRF) face two difficulties on this problem. First, only short range contextual relations are usually considered in these models. Pixels are connected within a small neighborhood, and model parts have constraints only if they are nearby (*e.g.* pictorial structures). This limited scope is caused by either the fact that background can corrupt the long range relations, or lacking cues to generate such constraints. Second, the contextual relations are often restricted to pairwise constraints to ensure computational tractability. However, most shape configurations cannot be decomposed into the summation of pairwise checks. The simplest case is a straight line whose valid verification involves at least three points. Any pair of two points can form a line and therefore does not give any information on the hypothesis. In general, robustly matching a shape requires simultaneous reasoning over many entities. In this thesis, we have developed a principled approach that addresses the context issue from the following aspects:

1. We identifies the underlying generic structures that capture the inherent correlations of a long sequence of points, independent of the model. Specifically, Chapter 2 introduces a novel topological formulation for grouping contours. The mechanism is able to extract topologically 1D image contours robust to clutter and broken edges, and generally applicable to grouping and segmenting data forming a parameterized structure (*i.e.* a manifold). Part of the work in Chapter 2 was published in (Zhu *et al.* , 2007).

2. The set-to-set matching method we developed in Chapter 3 opens a path towards utilizing the context arising from a set, going beyond the traditional pairwise constraints on tokens. This was made feasible by a holistic shape feature that can be adjusted on-the-fly according to the context from figure/ground selection. The resulting combinatorial problem of matching can be optimized and bounded by LP-based primal-dual algorithms presented in Chapter 4. Part of the work in Chapter 3 was published in (Zhu *et al.* , 2008; Srinivasan *et al.* , 2010). The review on primal dual algorithms in Chapter 4 is based on (Zhu, 2009).

3. Additionally, we are able to incorporate more sophisticated structures into the contextual shape reasoning. Chapter 5 extends the holistic approach to match image contours with an articulation model represented by a *tree*. In Chapter 6, the basic shape tokens, *i.e.* regions, do not generate shape features by themselves. It is the *difference* of a region and its neighbors in terms of figure/ground selection produce boundaries forming object shapes. This property brings in bipartite graph packing.

We have noticed several future directions worthy of further exploration:

1. Interaction between grouping and shape matching. Although the holistic shape reasoning requires extraction of discrete, big structures from bottom-up grouping, this

does not mean that grouping and shape matching have to be performed in a sequential, feed-forward way. The feedback from top-down shape matching can potentially resolve ambiguities in bottom-up grouping. For example, a well matched incomplete shape can guide the search for missing segments due to faint boundaries and leakages. The integration of the decisions on the two processes is preferred.

2. Integration of regions and contours into the packing framework. We have developed and demonstrated contour packing and region packing separately in Chapter 3 and Chapter 6. Contours express elongated boundary structures while regions capture boundary closure and figure/ground segregation. The complementary role of contours and regions suggests that combining the two into a single computational framework would further reduce false shape detections.

3. Designing better deformable model representation. The tree-based model we used in Chapter 5 is a special case of AND/OR graph (Zhu & Mumford, 2006), which is more suitable for representing models with multiple prototypes and occlusions. It is also important to consider how to exploit features generated from the intermediate level of AND/OR graph.

4. Finding common shapes in multiple images. In all the computational paradigms, we dealt with holistic matching between only two shapes. Discovering common shapes from multiple images would be interesting from both practical and theoretical point of views. In addition to spatial context contained within each individual image, context across all the images needs to be investigated for this problem.

5. Extension of primal-dual algorithms to model selection and region packing. We have merely scratched the surface of employing these ideas to search and bound the resulting general packing problem. Additional structures such as bipartite graph on the image side and tree or AND/OR graph on the model side are not exploited. We believe that more efficient combinatorial algorithms and procedures can be designed by incorporating these new structures into the oracle.

Appendix

A.1 Proof of Theorem 2.1

Theorem 2.1 *The necessary condition for the critical points (local maxima) of the following optimization problem*

$$\max_{x\in\mathbb{C}^n} \frac{\mathrm{Re}(x^{\mathrm{H}}Px \cdot e^{-i\Delta\theta})}{x^{\mathrm{H}}x} \tag{A.1}$$

is that x is an eigenvector of

$$M(\Delta\theta) = \frac{1}{2}(P \cdot e^{-i\Delta\theta} + P^T \cdot e^{i\Delta\theta}) \tag{A.2}$$

Moreover, the corresponding local maximal value is the eigenvalue $\lambda(M(\Delta\theta))$.

Proof. Let $x = x_r + i \cdot x_c$ where x_r and x_c are the real and imaginary parts of x. The original problem can be rewritten as

$$\max_{x_r,x_c} \ (x_r^{\mathrm{T}}Px_r + x_c^{\mathrm{T}}Px_c)\cos\Delta\theta + (x_r^{\mathrm{T}}Px_c - x_c^{\mathrm{T}}Px_r)\sin\Delta\theta \tag{A.3}$$

$$s.t. \quad x_r^{\mathrm{T}}x_r + x_c^{\mathrm{T}}x_c = 1 \tag{A.4}$$

$$x_r, x_c \in \mathbb{R}^n \tag{A.5}$$

Hence, the Lagrangian has the following form with λ as the multiplier on the constraint:

$$L = (x_r^{\mathrm{T}}Px_r + x_c^{\mathrm{T}}Px_c)\cos\Delta\theta + (x_r^{\mathrm{T}}Px_c - x_c^{\mathrm{T}}Px_r)\sin\Delta\theta + \lambda(x_r^{\mathrm{T}}x_r + x_c^{\mathrm{T}}x_c - 1)$$

By taking derivatives of the Lagrangian, we have

$$\frac{\partial L}{\partial x_r} = (P^T + P)\cos\Delta\theta \cdot x_r + (P - P^T)\sin\Delta\theta \cdot x_c + 2\lambda x_r = 0 \tag{A.6}$$

$$\frac{\partial L}{\partial x_c} = (P^T + P)\cos\Delta\theta \cdot x_c + (P^T - P)\sin\Delta\theta \cdot x_r + 2\lambda x_c = 0 \tag{A.7}$$

Setting the above derivatives to 0 gives all the local maxima of the original problem (2.1). Notice that P is a real matrix, we obtain the following equation by combining eq. (A.6) and eq. (A.7):

$$[\ \frac{P + P^T}{2} \cdot \cos\Delta\theta + i \cdot \frac{P^T - P}{2} \cdot \sin\Delta\theta\ ] \cdot (x_r + i \cdot x_c) = -\lambda(x_r + i \cdot x_c) \tag{A.8}$$

Therefore $x = x_r + i \cdot x_c$ is a real eigenvector of matrix:

$$M(\Delta\theta) = \frac{P + P^T}{2} \cdot \cos\Delta\theta + i \cdot \frac{P^T - P}{2} \cdot \sin\Delta\theta \tag{A.9}$$

$$= \frac{1}{2}(P \cdot e^{-i\Delta\theta} + P^T \cdot e^{i\Delta\theta}) \tag{A.10}$$

with eigenvalue $-\lambda$. Notice that $M(\Delta\theta)$ is a Hermitian matrix and hence all its eigenvalues are real. By substituting eq. (A.6) and eq. (A.7) back to the original cost function we have

$$(x_r^{\mathrm{T}} P x_r + x_c^{\mathrm{T}} P x_c)\cos\Delta\theta + (x_r^{\mathrm{T}} P x_c - x_c^{\mathrm{T}} P x_r)\sin\Delta\theta = -\lambda(x_r^{\mathrm{T}} x_r + x_c^{\mathrm{T}} x_c) = -\lambda \tag{A.11}$$

The local optimal values are exactly the corresponding eigenvalues of $M(\Delta\theta)$.

□

A.2 Proof of Theorem 2.2

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]*:*

$$Pr(i, m) = \sum_{\lambda_j\ real} \lambda_j^m U_{ij} V_{ij} + \sum_{\lambda_j\ complex} \mathrm{Re}(\lambda_j^m U_{ij} V_{ij}) \tag{A.12}$$

[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* $\|P - Q\| < \epsilon$*.*

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

$$Pr(i, m) = (P^m)_{ii} \tag{A.13}$$

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

Consider the eigenvalue decomposition of P

$$P = U \Sigma U^{-1} \tag{A.14}$$

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.* $(U^{-1})^{\mathrm{T}} = V$. The power of P can be easily computed by

$$P^m = U \Sigma^m U^{-1} \tag{A.15}$$

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

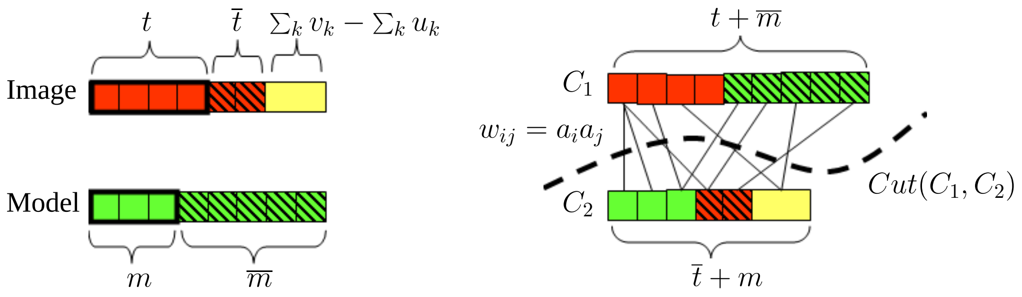
$$\begin{aligned} (P^m)_{ii} &= (U \Sigma^m U^{-1})_{ii} && \text{(A.16)} \\ &= \sum_j U_{ij} \cdot \lambda_j^m \cdot V_{ij} && \text{(A.17)} \\ &= \sum_{\lambda_j \ real} \lambda_j^m U_{ij} V_{ij} + \sum_{\lambda_j \ complex} \mathrm{Re}(\lambda_j^m U_{ij} V_{ij}) && \text{(A.18)} \end{aligned}$$

Eq (A.18) 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 2.2 *(Peakness of Random Walk Cycles) $R(i, T)$ can be computed by the eigenvalues of transition matrix P:*

$$R(i, T) = \frac{\sum_j \mathrm{Re}(\frac{\lambda_j^T}{1-\lambda_j^T} \cdot U_{ij} V_{ij})}{\sum_j \mathrm{Re}(\frac{1}{1-\lambda_j} \cdot U_{ij} V_{ij})} \tag{A.19}$$



(a) Packing one bin

(b) The corresponding graph cut

Figure A.1: Reduction from packing to MaxCut. (a) is a simple case where there is only one bin. The red blocks represent image contours nodes $\mathcal{I}$. The green blocks are nodes for model parts $\mathcal{M}$ and the yellow nodes is the fictitious node $\{V_0\}$. Image or model background nodes are shaded. (b) shows the corresponding graph cut of the packing.

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

$$\sum_{k=1}^{\infty} Pr(i, kT) = \sum_j \text{Re}(\lambda_j^T/(1-\lambda_j^{\text{T}}) \cdot U_{ij}V_{ij}) \tag{A.20}$$

$$\sum_{k=1}^{\infty} Pr(i, k) = \sum_j \text{Re}(1/(1-\lambda_j) \cdot U_{ij}V_{ij}) \tag{A.21}$$

Finally we have

$$R(i,T) = \frac{\sum_j \text{Re}(\frac{\lambda_j^T}{1-\lambda_j^{\text{T}}} \cdot U_{ij}V_{ij})}{\sum_j \text{Re}(\frac{1}{1-\lambda_j} \cdot U_{ij}V_{ij})} \tag{A.22}$$

□

A.3 Proof of Theorem 3.1

In this section we show that the contour packing problem can be reduced to MaxCut when the dissimilarity function $D_{ij}(\cdot)$ in eq. (3.7) is L_2. This reformulation leads to a computational solution via SDP, with a proved bound on the optimal cost.

A simple example with one bin

First we start with the simplified case containing one bin only. In this case the bin contains one single value of feature counts. For convenience, we denote:

- $t = \sum_{i \in S^I} v_i$ to be the total contribution of selected image contours S^I to the bin;

- $\overline{t} = \sum_{i \notin S^I} v_i$ to be the contribution from *unselected* contours $\mathcal{I} \setminus S^I$;

- $m = \sum_{i \in S^M} u_i$ to be the total contribution of selected model parts S^M;

- $\overline{m} = \sum_{i \notin S^I} u_i$ to be the contribution from *unselected* model parts $\mathcal{M} \setminus S^M$.

With the above notations, optimizing eq. (**??**) can be reduced to minimizing:

$$(t - m)^2 = (\sum_{i \in S^I} v_i - \sum_{i \in S^M} u_i)^2 \tag{A.23}$$

We balance the total contributions of the image and model side to the bin by adding a dummy node V_0. Without loss of generality, we assume $\sum_i u_i \geq \sum_i v_i$ and the contribution of V_0 to the bin is $\sum_i u_i - \sum_i v_i$. V_0 can be regarded as a virtual contour which can *never* be packed. By including this special node, we are ready to establish the connection between the packing and MaxCut:

Lemma A.1. *Set graph $G_{packing} = (V, E, W)$ with $V = \mathcal{I} \cup \mathcal{M} \cup \{V_0\}$ and $w_{ij} = a_i a_j$, where*

$$a_i = \begin{cases} v_i & if\ V_i \in \mathcal{I} \\ u_i & if\ V_i \in \mathcal{M} \\ \sum_k u_k - \sum_k v_k & if\ V_i = V_0 \end{cases}$$

The optimal subset S^I_* and S^M_* with the best matching cost $(t - m)^2$ in eq. (A.23) is given by the maximum cut of the packing graph $G_{packing}$. If (C_1, C_2) is the cut with $V_0 \in C_2$, the optimal subsets are given by $S^I_* = \mathcal{I} \cap C_1$ and $S^M_* = \mathcal{M} \cap C_2$ (see Fig. A.3).

Proof. Since the total contributions of $\mathcal{I} \cup \{V_0\}$ and $\mathcal{M}$ are the same to the bin, we can simply include V_0 into $\mathcal{I}$. Any cut (C_1, C_2) of the graph $G_{packing}$ with $V_0 \in C_2$ uniquely defines the selection on $\mathcal{I}$ and $\mathcal{M}$ as $S^I = \mathcal{I} \cap C_1$ and $S^M = \mathcal{M} \cap C_2$. Also notice that $C_1 = S^I \cup (\mathcal{M} \setminus S^M)$ and $C_2 = S^M \cup (\mathcal{I} \setminus S^I)$. Recall that t, $\overline{t}$, m and $\overline{m}$ represent the total contributions from S^I, $\mathcal{I} \setminus S^I$, S^M and $\mathcal{M} \setminus S^M$ respectively. Because V_0 contributes to $\overline{t}$, we can set $c = t + \overline{t} = m + \overline{m}$.

The cut value $Cut(C_1, C_2)$ can be computed by

$$\begin{aligned} Cut(C_1, C_2) = & \sum_{i \in C_1, j \in C_2} w_{ij} = \sum_{i \in C_1, j \in C_2} a_i a_j \\ = & (\sum_{i \in C_1} a_i)(\sum_{j \in C_2} a_j) = (t + \overline{m})(\overline{t} + m) \end{aligned} \tag{A.24}$$

$\sum_{i \in C_1} a_i = t + \overline{m}$ comes from equalities $C_1 = S^I \cup (\mathcal{M} \setminus S^M)$, $t = \sum_{i \in S^I} a_i$ and $\overline{m} = \sum_{i \notin S^M} a_i$. Similarly we can prove $\sum_{j \in C_2} a_j = \overline{t} + m$.

Finally, a simple calculation shows that the cut value and the matching cost sum up to a constant c^2:

$$(t + \overline{m})(\overline{t} + m) = c^2 - (t - m)^2$$

Therefore, minimizing $(t - m)^2$ is equivalent to finding the maximum cut on $G_{packing}$, whose cut value is given by $(t + \overline{m})(\overline{t} + m)$. □

Note that without any constraint, the system can choose trivial solution of packing nothing from image and model. This corresponds to the cut between $\mathcal{I}$ and $\mathcal{M}$. This can be alleviated by fixing the model nodes since we know what to pack on the model side. We also have the freedom of multiple choices on model nodes, which is essential for articulation model in Section 4.2. These modifications can all be encoded as hard constraints on the MaxCut.

Reduction of the full problem

Lemma A.1 can be naturally generalized to multiple knapsacks. Each bin in H_j introduces an extra node. Set $\mathcal{A}$ to be the set of all these nodes. Now we would like to consider the cut on the graph with nodes $\mathcal{I}$, $\mathcal{M}$ and $\mathcal{A}$. This is captured by Theorem 3.1:

Construct a graph $G_{packing} = (V, E, W)$ with $V = \mathcal{I} \cup \mathcal{M} \cup \mathcal{A}$ and $w_{ij} = a_i^T a_j$, where

$$a_i = \begin{cases} V^I_{(:,i)} & if \text{ node } i \in \mathcal{I} \\ V^M_{(:,i)} & if \text{ node } i \in \mathcal{M} \\ (0, ..., 0, |\sum_k V^I_{ik} - \sum_k V^M_{ik}|, 0, ..., 0)^T & if \text{ node } i \in \mathcal{A} \end{cases} \tag{A.25}$$

Here $V_I(k,i)$ is the feature contribution of image segment i to the histogram bin k. $V^M(k,i)$ is defined similarly. $V^I_{(:,i)}$ and $V^M_{(:,i)}$ are the i^{th} columns of V^I and V^M.

The optimal subset S^I_* and S^M_* with the best matching cost $\sum_k (t_k - m_k)^2$ in eq. (A.23) is given by the maximum cut of the graph $G_{packing}$. If (C_1, C_2) is the cut with $V_0 \in C_2$, the optimal subsets are given by $S^I_* = \mathcal{I} \cap C_1$ and $S^M_* = \mathcal{M} \cap C_2$.

Proof. Let $G_{packing} = G_1 \cup ... \cup G_l$ where G_k's are graphs induced by bin k defined in Lemma A.1. Applying Lemma A.1 to all these subgraphs. □

A.4 Proof of Theorem 4.1

Theorem A.2. *(Littlestone & Warmuth, 1989) (Perturbed Value of the Strategy) Let $\mathcal{R} = \sum_t \sum_j \overline{y}^t_j \mathcal{R}^t_j$ and $\mathcal{L} = \sum_t \sum_j \overline{y}^t_j \mathcal{L}^t_j$ be the cumulative reward and loss of the strategy using eq. (4.7). The perturbed value of the strategy given by eq. (4.7) is worse than the performance of best pure strategy only by $\frac{\log m}{\epsilon}$, as stated in the following inequality:*

$$\max_j \mathcal{V}_j \leq \exp(\epsilon)\mathcal{R} - \exp(-\epsilon)\mathcal{L} + \frac{\log m}{\epsilon} \tag{A.26}$$

Proof. Consider the potential function $\Phi^t = \sum_j y^t_j$.

On the one hand, we can compute it using the update rule:

$$\begin{aligned}
\Phi^t &= \sum_j y^t_j \\
&= \sum_j y^{(0)} \prod_{k=1}^{t} \exp[\epsilon \mathcal{V}^k_j] && \text{(Update rule (4.7))} \\
&= \sum_j \exp[\epsilon \sum_{k=1}^{t} \mathcal{V}^k_j] && (y^{(0)}_j = 1) \\
&\geq \exp[\epsilon \cdot \sum_{k=1}^{t} \mathcal{V}^k_j] && \text{(A.27)}
\end{aligned}$$

Note the above inequality holds for any j. Therefore, Φ^t is bounded below by

$$\Phi^t \geq \exp[\epsilon \cdot \max_j \mathcal{V}_j] \tag{A.28}$$

On the other hand, we have

$$\begin{aligned}
y_j^{t+1} - y_j^t &= y^t[\exp(\epsilon \mathcal{V}_j^t) - 1] \\
&\leq y^t \cdot (\epsilon \mathcal{V}_j^t) \cdot \exp(\epsilon \mathcal{V}_j^t) \\
&= y^t[\epsilon \exp(\epsilon \mathcal{V}_j^t)\mathcal{R}_j^t - \epsilon \exp(\epsilon \mathcal{V}_j^t)\mathcal{L}_j^t] \\
&\leq y^t[\epsilon \exp(\epsilon)\mathcal{R}_j^t - \epsilon \exp(-\epsilon)\mathcal{L}_j^t] \\
&= y^t \epsilon \widetilde{\mathcal{V}}_j^t
\end{aligned}$$

Here $\widetilde{\mathcal{V}}_j^t = \exp(\epsilon)\mathcal{R}_j^t - \exp(-\epsilon)\mathcal{L}_j^t$ is the "perturbed" version of value $\mathcal{V}_j^t$. The first inequality holds because $\exp(x) - 1 \leq x \cdot \exp(x)$ for any x. The second inequality is due to the fact that $\mathcal{V}_j^t \in [-1, 1]$.

By summing up the above inequality over j, we have

$$\begin{aligned}
\Phi^{t+1} &= \sum_j (y_j^{t+1} - y_j^t) + \Phi^t \\
&\leq \sum_j y_j^t \epsilon \widetilde{\mathcal{V}}_j^t + \Phi^t \\
&= \epsilon \Phi^t \cdot \sum_j y_j^t \widetilde{\mathcal{V}}_j^t / \sum_j y_j^t + \Phi^t \\
&= \Phi^t (1 + \epsilon \widetilde{\mathcal{V}}^t) \\
&\leq \Phi^t \cdot \exp(\epsilon \widetilde{\mathcal{V}}^t) && (1 + x \leq \exp(x))
\end{aligned}$$

Using induction over t and $\Phi^0 = m$, we bound Φ^t above by

$$\Phi^t \leq m \cdot \exp(\sum_k \epsilon \widetilde{\mathcal{V}}^k) \tag{A.29}$$

Finally combining eq. (A.28), (A.29) yields

$$\epsilon \cdot \max_j \mathcal{V}_j \leq \log m + \sum_k \epsilon \widetilde{\mathcal{V}}^k \tag{A.30}$$

which is equivalent to eq. (4.8).

□

A.5 Proof of Corollary 4.2

Corollary A.3. *(Regret Over Time) If $\mathcal{V}_j^t \in [-\rho, \rho]$ for all j, then we have a bound on the average value $\mathcal{V}/T$:*

$$\max_j \frac{\mathcal{V}_j}{T} \leq \frac{\mathcal{V}}{T} + \frac{\rho \log m}{\epsilon T} + \rho\epsilon \exp(\epsilon) \tag{A.31}$$

Proof. Since $\mathcal{V}_j^t \in [-\rho, \rho]$, we can substitute $\mathcal{V}_j^t$ by $\mathcal{V}_j^t/\rho$ and prove the following inequality for $\mathcal{V}_j^t \in [-1, 1]$:

$$\max_j \mathcal{V}_j \leq \mathcal{V} + \frac{\log m}{\epsilon} + T\epsilon \exp(\epsilon)$$

We set $\mathcal{R}_j^t = \max(0, \mathcal{V}_j^t)$ and $\mathcal{L}_j^t = \max(0, -\mathcal{V}_j^t)$, which satisfies $\mathcal{V}_j^t = \mathcal{R}_j^t - \mathcal{L}_j^t$.

Under these simplifications, we can apply Theorem 4.1 on $\mathcal{V}$:

$$\begin{aligned}
\max_j \mathcal{V}_j &\leq \widetilde{\mathcal{V}} + \frac{\log m}{\epsilon} \\
&= \mathcal{V} + \frac{\log m}{\epsilon} + (\exp(\epsilon) - 1)\mathcal{R} - (\exp(-\epsilon) - 1)\mathcal{L} \\
&\leq \mathcal{V} + \frac{\log m}{\epsilon} + \epsilon \exp(\epsilon)|\mathcal{V}| \\
&\leq \mathcal{V} + \frac{\log m}{\epsilon} + \epsilon \exp(\epsilon)T
\end{aligned}$$

The first inequality uses the fact that $|\mathcal{V}| = \mathcal{R} + \mathcal{L}$, $\exp(\epsilon) - 1 \leq \epsilon \exp(\epsilon)$ and $1 - \exp(-\epsilon) \leq \epsilon < \epsilon \exp(\epsilon)$. □

A.6 Proof of Theorem 4.4

Theorem A.4. *(Complexity of the Primal Dual Algorithm) Algorithm 2 either declares that the fractional packing eq. (4.2) is infeasible, or outputs an approximate feasible solution $\bar{x}$ satisfying*

$$a_j^{\mathrm{T}} \bar{x} - c_j \leq \delta \tag{A.32}$$

for all $j = 1, ..., m$. The total number of calls to the oracle is $O(\rho^2 \delta^{-2} \log m)$ with $\rho = \max_j \max_{x \in P} |f_j(x)|$.

Proof. We build our proof based on Corollary 4.2. First notice that if $\mu^t > 0$ at some time t, then the eq. (4.2) is indeed infeasible. Otherwise suppose there exists x^t such that $f_j(x^t) = a_j^{\mathrm{T}} x^t - c_j \leq 0$ for all j. Because $y^t \geq 0$ throughout the algorithm, $\mu^t \leq \sum_j y_j^t f_j(x^t) \leq 0$, a contradiction.

Suppose the algorithm runs to the end and outputs $\bar{x}$. Let $\mathcal{V}_j^t = w^t f_j(x^t)$ be the value incurred by the update. Notice that $\mathcal{V}_j^t \in [-1, 1]$. By applying Corollary 4.2, we have

$$\begin{aligned}
\max_j [a_j^{\mathrm{T}} \bar{x} - c_j] &= \max_j \frac{\sum_t w^t (a_j^{\mathrm{T}} x^t - c_j)}{\sum_t w^t} \\
&= \max_j \frac{\sum_t \mathcal{V}_j^t}{\sum_t w^t} \\
&\leq \frac{1}{\sum_t w^t} [\mathcal{V} + \frac{\log m}{\epsilon} + \epsilon T \exp(\epsilon)] \\
&\leq \frac{1}{\sum_t w^t} [\frac{\log m}{\epsilon} + \epsilon T \exp(\epsilon)] \\
&= \frac{1}{S} [\frac{\log m}{\epsilon} + \epsilon T \exp(\epsilon)] \\
&\leq \delta
\end{aligned} \tag{A.33}$$

The first inequality uses the fact that $\mathcal{V}^t = (w^t / \sum_j y_j^t) \sum_j y_j^t f_j(x^t) = w^t \mu^t / \sum_j y_j^t \leq 0$ for every t since the oracle never fails. The last inequality is due to the termination condition $S \geq 9\rho \log m / \delta^{-2}$, $T/S = T / \sum_t w^t \leq \rho$ and $\epsilon = 3\delta/\rho$.

Therefore, x returned by the algorithm satisfies the approximate feasibility eq. (4.13). Finally, each time the algorithm collects $w^t \geq 1/\rho$ and it terminates when $S = \sum_t w_t \geq S \geq 9\rho \log m / \delta^{-2}$, so the total number of iterations is at most $O(\rho^2 \delta^{-2} \log m)$. □

A.7 Proof of Theorem 6.1

Theorem A.5. *The bipartite region graph packing problem consists in finding an optimal bipartite subgraph $G_{sub}(F, \overline{F})$ of the region graph G, which minimizes cost $C_p(F, \overline{F})$ defined in eq. (6.2). It can be reduced to a cardinality constrained and multicriteria cut problem on a graph G' associated with R positive edge weight functions $w^{(1)}$,...,$w^{(R)}$ according to R criteria. The cardinality constrained and multicriteria cut problem seeks a cut C with* *cardinality at least d: $\sum_{E_{ij} \in C} 1 \geq d$, and all R criteria are satisfied: $\sum_{E_{ij} \in C} w_{ij}^{(k)} \leq b^{(k)}$ for $k = 1, 2, ..., R$.*

Proof. We first transform bipartite region graph packing problem into a simpler linear form, and notice that the main hurdle is the bipartite graph packing cost $C_p(F, \overline{F})$ is an L_1-norm. Using a similar technique which converts contour packing into primal-dual packing in eq. (4.15), we have:

$$\min_{x,s^+,s^-} \|V^I \cdot x - sc^M\|_1 = 1^{\mathrm{T}}[\mathrm{Diag}(sc^M)s^+ + \mathrm{Diag}(sc^M)s^-] \tag{A.34}$$

$$\text{s.t. } V^I x - sc^M = \mathrm{Diag}(sc^M)s^+ - \mathrm{Diag}(sc^M)s^- \tag{A.35}$$

$$x \in \{0,1\}^{|E(G)|},\ s^+, s^- \in [0,1]^m \tag{A.36}$$

Here s^+ and s^- are normalized slack variables on the feature bins. Furthermore, this can be rewritten as:

$$\max_{x,s^+} V^I + 2 \cdot 1^{\mathrm{T}}\mathrm{Diag}(sc^M)(1 - s^+) \tag{A.37}$$

$$\text{s.t. } V^I x + \mathrm{Diag}(sc^M)(1 - s^+) \leq sc^M \tag{A.38}$$

$$x \in \{0,1\}^{|E(G)|},\ s^+ \in [0,1]^m \tag{A.39}$$

by substituting the constraint in eq. (A.35) and using the fact that s^- is nonnegative. We can further make the continuous slack variable $(1 - s^+) \in [0,1]^m$ a binary one by splitting it into units of 1,2,4,...,2^ℓ pixels for each bin. Since ultimately the cost is measured as multiples of a pixel, the binary representation is sufficient to reproduce any integer slack. We group these slack variables into a single vector s.

If one would like to bound the objective function eq. (A.37), a feasibility problem arises by changing the objective function into a constraint $V^I + 2 \cdot 1^{\mathrm{T}}\mathrm{Diag}(sc^M)(1 - s^+) \geq c$ for a constant c:

$$\mathrm{Feasibility}(x,s): \quad V^I + 2 \cdot p^{\mathrm{T}} s \geq c \tag{A.40}$$

$$V^I x + p^{\mathrm{T}} s \leq sc^M \tag{A.41}$$

$$x \in \{0,1\}^{|E(G)|},\ s \in [0,1]^m \tag{A.42}$$

where p_i is the number of pixels included in slack s_i^+. Now the feasibility problem appears to be the same as a cardinality constrained and multicriteria cut problem except that the binary indicators x and s have to be defined on graph edges and (x, s) must represent a cut to the graph.

Construct a graph G' with additional nodes $V(G') = \{V_f, V_b\} \cup V(G) \cup S$ with following specifications: 1) Two V_f,V_b are the source and sink terminals of the graph representing foreground and background respectively; 2) $V(G)$ are the nodes from the region graph G and a node belongs to foreground if on the same side as V_f in the cut; 3) S denotes the bin slack variables s and the slack is applied if on the same side as V_f in the cut. Define edge weight functions $w^{(i)}$ to be V_{ik}^I for edge E_k in G[2], and p_i for edge between s_i and V_b. The left side of each constraint in $\mathrm{Feasibility}(x, s)$ is the sum of weights in a cut on G'.

The above problem is exactly a cardinality constrained and multicriteria cut problem with cardinality defined by the cost function and criteria defined by the feature bins. □

[2] Unary terms used in Section 6.3 can be represented as edges between $V(G)$ and $\{V_f, V_b\}$

References

ALTER, T. D., & BASRI, RONEN. 1996. Extracting Salient Curves from Images: An Analysis of the Saliency Network. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

AMIR, ARNON, & LINDENBAUM, MICHAEL. 1998. Grouping-Based Nonadditive Verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **20**(2), 186–192.

AMIT, YALI, & WILDER, KENNETH. 1997. Joint Induction of Shape Features and Tree Classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **19**(11), 1300–1305.

ANDERSEN, E. D., & ANDERSEN, K. D. 2000. The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm. *Pages 197–232 of: et al.*, H. FRENK (ed), *High Performance Optimization*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

BAI, X., LATECKI, L.J., & LIU, W.Y. 2007. Skeleton Pruning by Contour Partitioning with Discrete Curve Evolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **29**(3), 449–462.

BARROW, H. G., TENENBAUM, J. M., BOLLES, R. C., & WOLF, H. C. 1977. Parametric Correspondence and Chamfer Matching: Two New Techniques for Image Matching. *International Joint Conference on Artificial Intelligence (IJCAI)*, 659–663.

BASRI, R., & JACOBS, D. W. 1997. Recognition Using Region Correspondences. *International Journal of Computer Vision (IJCV)*, **25**(2), 145–166.

BELONGIE, SERGE, MALIK, JITENDRA, & PUZICHA, JAN. 2002. Shape Matching and Object Recognition Using Shape Contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.

BENTZ, C., COSTA, M. C., DERHY, N., & ROUPIN, F. 2009. Cardinality constrained and multicriteria (multi)cut problems. *J. of Discrete Algorithms*, **7**(March), 102–111.

BERG, A. C., BERG, T. L., & MALIK, J. 2005. Shape Matching and Object Recognition Using Low Distortion Correspondences. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, I: 26–33.

BIEDERMAN, I. 1985. Human Image Understanding: Recent Research and a Theory. *Computer Vision, Graphics, and Image Processing (CVGIP)*, **32**, 29–73.

BLUM, H. 1967. A Transformation for Extracting new Descriptors of Shape. *Pages 362–380 of:* WATHEN-DUNN (ed), *Models for the Perception of Speech and Visual Form*. MIT-Press.

BORENSTEIN, E., & ULLMAN, S. 2002. Class-Specific, Top-Down Segmentation. *European Conference on Computer Vision (ECCV)*.

BOYD, STEPHEN, & VANDENBERGHE, LIEVEN. 2004. *Convex Optimization*. Cambridge: Cambridge University Press.

BROOKS, R. 1983. Model-Based 3-D Interpretations of 2-D Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **5**(2), 140–150.

BRUGLIERI, MAURIZIO, MAFFIOLI, FRANCESCO, & EHRGOTT, MATTHIAS. 2004. Cardinality constrained minimum cut problems: complexity and algorithms. *Discrete Appl. Math.*, **137**(March), 311–341.

CATANZARO, BRYAN, SU, BOR-YIING, SUNDARAM, NARAYANAN, LEE, YUNSUP, MURPHY, MARK, & KEUTZER, KURT. 2009. Efficient, High-Quality Image Contour Detection. *IEEE International Conference on Computer Vision (ICCV)*.

COOTES, T. F., TAYLOR, C. J., COOPER, D. H., & GRAHAM, J. 1995. Active Shape Models: Their Training and Application. *Computer Vision and Image Understanding (CVIU)*, **61**(1), 38–59.

COUR, TIMOTHEE, & SHI, JIANBO. 2007. Solving Markov Random Fields with Spectral Relaxation. *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS)*, **11**.

COUR, TIMOTHEE, BENEZIT, FLORENCE, & SHI, JIANBO. 2005. Spectral Segmentation with Multiscale Graph Decomposition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

DALAL, NAVNEET, & TRIGGS, BILL. 2005. Histograms of Oriented Gradients for Human Detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 886–893.

ELDER, J. H., & ZUCKER, S. W. 1996. Computing contour closure. *Lecture Notes in Computer Science*, **1064**.

FELZENSZWALB, P. F., & MCALLESTER, D. 2006. A Min-Cover Approach for Finding Salient Curves. *IEEE Computer Society Workshop on Perceptual Organization in Computer Vision (WPOCV)*, 185.

FELZENSZWALB, P. F., MCALLESTER, D., & RAMANAN, D. 2008. A discriminatively trained, multiscale, deformable part model. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–8.

FELZENSZWALB, PEDRO F., & SCHWARTZ, JOSHUA D. 2007. Hierarchical Matching of Deformable Shapes. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

FELZENSZWALB, P.F., & HUTTENLOCHER, D.P. 2005. Pictorial Structures for Object Recognition. *International Journal of Computer Vision (IJCV)*.

FERRARI, VITTORIO, JURIE, FRÉDÉRIC, & SCHMID, CORDELIA. 2007a. Accurate Object Detection with Deformable Shape Models Learnt from Images. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

FERRARI, VITTORIO, FEVRIER, L, JURIE, FRÉDÉRIC, & SCHMID, CORDELIA. 2007b. Groups of Adjacent Contour Segments for Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.

FISCHER, BERND, & BUHMANN, JOACHIM M. 2003. Path-Based Clustering for Grouping of Smooth Curves and Texture Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **25**(4), 513–518.

GALLEGUILLOS, CAROLINA, BABENKO, BORIS, RABINOVICH, ANDREW, & BELONGIE, SERGE. 2008. Weakly Supervised Object Recognition and Localization with Stable Segmentations. *European Conference on Computer Vision (ECCV)*.

GDALYAHU, YORAM, & WEINSHALL, DAPHNA. 1999. Flexible Syntactic Matching of Curves and Its Application to Automatic Hierarchical Classification of Silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.

GOEMANS, M. X., & WILLIAMSON, D.P. 1995. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *Journal of the ACM*, **42**, 1115–1145.

GORELICK, LENA, GALUN, MEIRAV, SHARON, EITAN, BASRI, RONEN, & BRANDT, ACHI. 2006. Shape Representation and Classification Using the Poisson Equation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **28**(12), 1991–2005.

GRIMSON, ERIC. 1986. The Combinatorics of Local Constraints in Model-Based Recognition and Localization from Sparse Data. *JACM: Journal of the ACM*, **33**.

GRIMSON, W. E. L., & LOZANO-PEREZ, T. 1987. Localizing Overlapping Parts by Searching the Interpretation Tree. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **9**, 469–482.

GU, CHUNHUI, LIM, JOSEPH J., ARBELAEZ, PABLO, & MALIK, JITENDRA. 2009. Recognition using Regions. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

GUPTA, ABHINAV, & DAVIS, LARRY S. 2008. Beyond Nouns: Exploiting Prepositions and Comparative Adjectives for Learning Visual Classifiers. *European Conference on Computer Vision (ECCV)*, **5302**, 16–29.

HAN, F., & ZHU, S. C. 2009. Bottom-Up/Top-Down Image Parsing with Attribute Grammar. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **31**(1), 59–73.

HUTTENLOCHER, KLANDERMAN, & RUCKLIDGE. 1993. Comparing Images Using the Hausdorff Distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **15**.

JACOBS, DAVID W. 1996. Robust and Efficient Detection of Salient Convex Groups. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **18**(1), 23–37.

JIANG, H., & MARTIN, D. R. 2008. Global pose estimation using non-tree models. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–8.

JIANG, HAO, DREW, MARK S., & LI, ZE-NIAN. 2007. Matching by Linear Programming and Successive Convexification. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **29**(6), 959–975.

KESELMAN, YAKOV, & DICKINSON, SVEN J. 2005. Generic Model Abstraction from Examples. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **27**(7), 1141–1156.

KHANDEKAR, ROHIT. 2004. *Lagrangian Relaxation Based Algorithms for Convex Programming Problems*. Ph.D. thesis, IIT Delhi.

KOFFKA, KURT. 1935. *Principles of Gestalt Psychology*.

KOHLER, W. 1929. *Gestalt psychology*. New York: Liveright.

LAMDAN, Y., SCHWARTZ, J. T., & WOLFSON, H. J. 1990. Affine Invariant Model-Based Object Recognition. *IEEE Transactions on Robotics and Automation*, **6**, 578–589.

LATECKI, LONGIN JAN, LU, CHENGEN, SOBEL, MARC, & BAI, XIANG. 2008. Multiscale Random Fields with Application to Contour Grouping. *Advances in Neural Information Processing Systems (NIPS)*, 913–920.

LEE, MUN WAI, & COHEN, ISAAC. 2004. Proposal Maps Driven MCMC for Estimating Human Body Pose in Static Images. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

LEIGHTON, T., MAKEDON, F., PLOTKIN, S., STEIN, C., TARDOS, E., & TRAGOUDAS, S. 1991. Fast Approximation Algorithms for Multicommodity Flow Problems. *23rd Annual ACM Symposium on Theory of Computing (STOC)*, 101–111.

LEORDEANU, MARIUS, HEBERT, MARTIAL, & SUKTHANKAR, RAHUL. 2007. Beyond Local Appearance: Category Recognition from Pairwise Interactions of Simple Features. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

LEYMARIE, F.F., & LEVINE, M.D. 1992. Simulating the Grassfire Transform Using an Active Contour Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **14**(1), 56–75.

LI, L.J., SOCHER, R., & FEI FEI, L. 2009. Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2036–2043.

LING, HAIBIN, & JACOBS, DAVID W. 2005. Using the Inner-Distance for Classification of Articulated Shapes. *IEEE International Conference on Computer Vision (ICCV)*.

LITTLESTONE, & WARMUTH. 1989. The Weighted Majority Algorithm. *FOCS: IEEE Symposium on Foundations of Computer Science (FOCS)*.

LOWE, DAVID G. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision (IJCV)*, **60**(2), 91–110.

LU, CHENGEN, LATECKI, LONGIN JAN, ADLURU, NAGESH, YANG, XINGWEI, & LING, HAIBIN. 2009. Shape Guided Contour Grouping with Particle Filters. *IEEE International Conference on Computer Vision (ICCV)*.

MAHAMUD, S., WILLIAMS, L., THORNBER, K., & XU, K. 2003. Segmentation of multiple salient closed contours from real images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.

MAIRE, M., ARBELAEZ, P., FOWLKES, C., & MALIK, J. 2008. Using contours to detect and localize junctions in natural images.

MAJI, S., & MALIK, J. 2009. Object detection using a max-margin Hough transform. *CVPR*, 1038–1045.

MALISIEWICZ, TOMASZ, & EFROS, ALEXEI A. 2008. Recognition by Association via Learning Per-exemplar Distances. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June.

MARTIN, DAVID R., FOWLKES, CHARLESS, TAL, DORON, & MALIK, JITENDRA. 2001. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. *IEEE International Conference on Computer Vision (ICCV)*, 416–425.

MEDIONI, G. G., & GUY, G. 1993. Inferring Global Perceptual Contours from Local Features. *Image Understanding Workshop (IUW)*.

MEILA, MARINA, & SHI, JIANBO. 2000. Learning Segmentation by Random Walks. *Advances in Neural Information Processing Systems (NIPS)*, 873–879.

MORI, G. 2005. Guiding Model Search Using Segmentation. *IEEE International Conference on Computer Vision (ICCV)*.

MORI, GREG, REN, XIAOFENG, EFROS, ALEXEI A., & MALIK, JITENDRA. 2004a. Recovering Human Body Configurations: Combining Segmentation and Recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 326–333.

MORI, GREG, REN, XIAOFENG, EFROS, ALEXEI A., & MALIK, JITENDRA. 2004b. Recovering human body configurations: combining segmentation and recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

NESTEROV, Y. E., & NEMIROVSKY, A. S. 1993. *Interior Point Polynomial Methods in Convex Programming : Theory and Algorithms*. SIAM Publishing.

NEVATIA, R., & BINFORD, T. O. 1977. Description and Recognition of Curved Objects. *Artificial Intelligence*, **8**(1), 77–98.

PALMER, S. 1999. *Vision Science: Photons to Phenomenology*. MIT Press.

PELEG, S., & ROSENFELD, A. 1981. A Min-Max Medial Axis Transformation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **3**(2), 208–210.

PENTLAND, A. P. 1986. Perceptual Organization and the Representation of Natural Form. *Artificial Intelligence*, **28**, 293–331.

PLOTKIN, SERGE A., SHMOYS, DAVID B., & TARDOS, EVA. 1995. Fast Approximation Algorithms for Fractional Packing and Covering Problems. *Mathematics of Operations Research*, **20**, 257–301.

RAMANAN, DEVA. 2007. Learning to parse images of articulated bodies. *Advances in Neural Information Processing Systems (NIPS)*.

REN, XIAOFENG, BERG, ALEXANDER C., & MALIK, JITENDRA. 2005a. Recovering human body configurations using pairwise constraints between parts. *IEEE International Conference on Computer Vision (ICCV)*.

REN, XIAOFENG, FOWLKES, CHARLESS, & MALIK, JITENDRA. 2005b. Scale-Invariant Contour Completion Using Conditional Random Fields. *IEEE International Conference on Computer Vision (ICCV)*, 1214–1221.

RONFARD, RÉMI, SCHMID, CORDELIA, & TRIGGS, BILL. 2002. Learning to Parse Pictures of People. *European Conference on Computer Vision (ECCV)*.

SARKAR, SUDEEP, & SOUNDARARAJAN, PADMANABHAN. 2000. Supervised Learning of Large Perceptual Organization: Graph Spectral Partitioning and Learning Automata. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **22**(5), 504–525.

SHARON, E., & MUMFORD, D. 2006. 2D-Shape Analysis Using Conformal Mapping. *International Journal of Computer Vision*, **70**(1), 55–75.

SHI, J., & MALIK, J. 2000. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **22**(8), 888–905.

SHOTTON, J. D. J., BLAKE, A., & CIPOLLA, R. 2008. Multiscale Categorical Object Recognition Using Contour Fragments. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **30**(7), 1270–1281.

SHOTTON, JAMIE, BLAKE, ANDREW, & CIPOLLA, ROBERTO. 2005. Contour-Based Learning for Object Detection. *IEEE International Conference on Computer Vision (ICCV)*.

SIDDIQI, K., SHOKOUFANDEH, A., DICKINSON, S. J., & ZUCKER, S. W. 1999. Shock Graphs and Shape Matching. *International Journal of Computer Vision*, **35**(1), 13–32.

SRINIVASAN, PRAVEEN, & SHI, JIANBO. 2007. Bottom-up Recognition and Parsing of the Human Body. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

SRINIVASAN, PRAVEEN, ZHU, QIHUI, & SHI, JIANBO. 2010. Many-to-one Contour Matching for Describing and Discriminating Object Shape. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

STURM, J. F. 1999. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, **11**, 625–653.

TAYLOR, CAMILLO J., & BHUSNURMATH, ARVIND. 2008. Solving Image Registration Problems Using Interior Point Methods. *European Conference on Computer Vision (ECCV)*, October, 638–651.

TODOROVIC, S., & AHUJA, N. 2008. Region-Based Hierarchical Image Matching. *International Journal of Computer Vision (IJCV)*, **78**(1), 47–66.

TORRALBA, A., RUSSELL, B. C., & YUEN, J. 2009. LabelMe: online image annotation and applications. *MIT CSAIL Technical Report*.

TOSHEV, ALEXANDER, TASKAR, BEN, & DANIILIDIS, KOSTAS. 2010. Object Detection via Boundary Structure Segmentation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

ULLMAN, S. 1996. *High-Level Vision: Object Recognition and Visual Cognition*. MIT Press.

ULLMAN, S., & SHASHUA, A. 1988. Structural Saliency: The Detection of Globally Salient Structures Using a Locally Connected Network. *MIT AI Memo*.

VAIDYA, PRAVIN M. 1996. A new algorithm for minimizing convex functions over convex sets. *Math. Program.*, **73**(3), 291–341.

VAZIRANI, V. V. 2004. *Approximation Algorithms*. Springer.

WANG, LIMING, SHI, JIANBO, SONG, GANG, & SHEN, I-FAN. 2007. Object Detection Combining Recognition and Segmentation. *Asian Conference on Computer Vision (ACCV)*.

WANG, SONG, KUBOTA, TOSHIRO, SISKIND, JEFFREY MARK, & WANG, JUN. 2005. Salient Closed Boundary Extraction with Ratio Contour. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.

WERTHEIMER, M. 1938. Principles of perceptual organisation. *In:* ELLIS, W. H. (ed), *Source Book of Gestalt Psychology*.

WRIGHT, STEPHEN J. 1997. *Primal-dual interior-point methods*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics.

YOUNG, NEAL E. 1995. Randomized rounding without solving the linear program. *SODA '95: Proceedings of the sixth annual ACM-SIAM symposium on Discrete algorithms*, 170–178.

YU, STELLA X., & SHI, JIANBO. 2003. Multiclass Spectral Clustering. *IEEE International Conference on Computer Vision (ICCV)*, 313–319.

ZAHN, C. T., & ROSKIES, R. S. 1972. Fourier descriptors for plane closed curves. *IEEE Transactions on Computing*.

ZHANG, JIAYONG, LUO, JIEBO, COLLINS, ROBERT, & LIU, YANXI. 2006. Body Localization in Still Images Using Hierarchical Models and Hybrid Search. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

ZHU, QIHUI. 2009. Primal Dual Combinatorial Algorithms. *CIS WPE-II Report, University of Pennsylvania*.

ZHU, QIHUI, SONG, GANG, & SHI, JIANBO. 2007. Untangling Cycles for Contour Grouping. *IEEE International Conference on Computer Vision (ICCV)*.

ZHU, QIHUI, WANG, LIMING, WU, YANG, & SHI, JIANBO. 2008. Contour Context Selection for Object Detection: A Set-to-Set Contour Matching Approach. *European Conference on Computer Vision (ECCV)*.

ZHU, SONG-CHUN, & MUMFORD, DAVID. 2006. A stochastic grammar of images. *Foundations and Trends in Computer Graphics and Vision*, **2**(4), 259–362.