Mid-level Cues for Bottom-up Grouping

by

Tom Lee

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto

Abstract

Mid-level Cues for Bottom-up Grouping

Tom Lee

Doctor of Philosophy

Graduate Department of Computer Science

University of Toronto

2016

Bottom-up perceptual grouping is an essential but often elusive component of computer vision that occurs in support of object recognition. We base our analysis on Gestalt grouping principles that range from low-level cues, which lack contextual scope but are computationally attractive, to mid-level cues, which cover larger scope but are difficult to incorporate in a computationally efficient manner. Our thesis begins with object categorization as motivating context for perceptual grouping, highlighting the issues of complexity and learning, and the importance of feature representation in handling variability. We then make inroads into bottom-up grouping in three steps. First, we focus on the mid-level cue of symmetry, which we extend to handle curvature and taper. We make effective use of symmetry to handle a wide range of input variability, and demonstrate significant improvements from formulating symmetry-based grouping as an optimization problem. Second, we develop an energy-based superpixel grouping framework that has expressive power to accommodate multiple grouping cues that range from low-level to mid-level, and from contour-based to region-based. We demonstrate the benefit of combining multiple mid-level cues to eliminate false positives. Finally, we reformulate bottom-up perceptual grouping as a prediction task to enable us to use the framework of structured prediction to tackle grouping as a single, unified problem. We bring performance to a level competitive with recent state-of-the-art baselines to close the gap between bottom-up grouping and recognition.

Acknowledgements

I would like to express my gratitude to Sven Dickinson and Sanja Fidler, who in their roles as advisors provided me with dedicated support during the course of my graduate studies. Sven's knowledge and experience led to interactions that were intellectually stimulating, and his energetic approach impacted me in a positive way. Sanja was always ready to offer her valuable insight and advice, and my mentorship would not have been complete without her. For those remaining on my thesis committee, thanks to Allan Jepson for his consistently generous and thoughtful advice, and to Richard Zemel for his consistently clear and informative advice.

I am honoured to have met many exceptional people at the Department of Computer Science at the University of Toronto in professional and personal contexts. Alex Levinshtein was a valued co-author and mentor in many ways. I also thank Marcus Brubaker and Raquel Urtasun, and graduate students and members of the computer vision and machine learning groups with whom I've interacted, including Pablo, Mike, Shenlong, Jian, Kaustav, Yanshuai, and Mohammad.

Earlier mentors who left an impact on me include Suzanne Stevenson at the University of Toronto, my undergraduate advisors and lecturers at the University of Waterloo, and my high school teachers at Silverthorn Collegiate Institute.

Finally, thanks to my brother, my father and late mother, my aunts and uncles, my friends and musician friends, and my cousins for their love and understanding.

Contents

Chapter 1

Introduction

Perceptual grouping is an essential aspect of human vision that largely occurs behind the scenes. Take a moment to consider Figure 1.1, which depicts a scene that helps to reveal some of its inner workings. Although it is not difficult to identify the contents of the scene, a first glance likely required significant attention before the objects emerged from the brown and white patches. You may have noticed vertically symmetric figures in the snow and grouped them together as legs, and followed the contours along the back and neck, and all the way around before finally seeing an entire horse. The purpose of this example is to illustrate the "bottom-up" grouping process, whereby regions and contours are locally grouped by mid-level cues, like contour closure and symmetry, into object parts and whole objects, and eventually into a global interpretation of the scene. Since bottom-up grouping occurs without (but in support of) object recognition, the only grouping cues that are involved are ones that are not specific to any particular object. For example, a grouping cue that expects the shape of the side view of a horse plays no role in a purely bottom-up process. This leads to the "top-down" counterpart in object recognition, the process that occurs when you identify the horses. Despite the importance of identifying objects, however, it is not our focus here. The focus of this thesis is to take bottom-up cues—what humans already use successfully to make sense

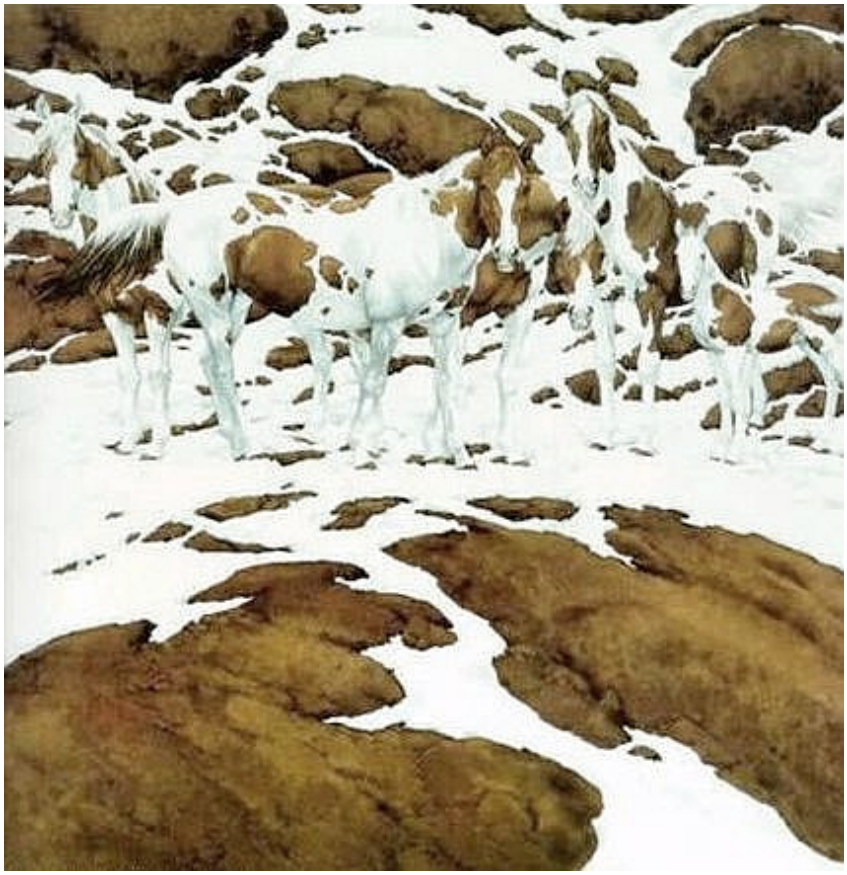


Figure 1.1: Demonstrating the bottom-up process of perceptual grouping.

of visual scenes—and give computers the same ability to take advantage of them.

Humans use bottom-up cues effortlessly, however it remains incredibly difficult to implement them in a computer. Perceptual grouping is a sub-area of computer vision that has been addressed by much prior work and has seen much progress (see Chapter 2). However, there remains a long path ahead that reflects the complexity of the issues still to be addressed. In this thesis we will consider the following issues:

- **Complexity**: Seen computationally, perceptual grouping involves a search through a space of possible scene interpretations. Using Figure 1.1 to illustrate, consider the possible combinations of patches that one could have taken, and the possible turns of contour one could have taken at each junction, and it quickly becomes clear that we are dealing with a combinatorial search space of intractable size. Aside from

developing an evaluative function that distinguishes good from bad interpretations, we need a strategy to efficiently guide us to the best one or few interpretations.

- **Variability**: A good perceptual grouping system is robust to a wide range of possible scene inputs. Such a system produces accurate interpretations not only for the current scene at hand, but for almost all other scenes as well. For example, a system that groups image elements by color alone may work well for many scenes of horses, but will fail for the scene in Figure 1.1. It is difficult to achieve robustness because there is a huge amount of variability in the range of possible inputs, and it is all too easy to design a method that works well for only a subclass of scenes but not others. We thus need grouping cues that capture properties at a broad enough level, such as shape, without being sensitive to detailed properties that are specific only to the example at hand.

- **Learning**: A perceptual grouping system that is designed in a fully manual way may be most amenable to human interpretation, however the alternative of learning such a system from training data offers the benefit of directly optimizing for accuracy. Specifically, when exposed to a set of scenes paired with their correct interpretations, a perceptual grouping system that is able to automatically adjust itself (via its parameters) to produce interpretations that better match the correct ones is more likely able to achieve accuracy in a cost-effective way. Here, such a system could remain independent of object categories throughout learning, and would, for example, allow for navigation around the horses in Figure 1.1 without needing to recognize them as horses.

We have illustrated above some basic issues that arise in designing a perceptual grouping system. Before we address them, however, we will review prior work that attempted these same issues, which will allow us to form a foundation on which we can build and advance.

We begin with the important early work from the Gestalt school of psychology, which produced the **principles of perceptual grouping** that capture regularities of 3D objects as they appear in a 2D image. Visual perception is governed by these principles in the sense that we organize the elements of an image into groups (*i.e.*, figure-ground segmentations) according to these principles. Wertheimer, 1923 [51] produced such a list of grouping principles:

proximity tendency to group elements that are near each other

similarity tendency to group elements that are like each other

continuity tendency to form smooth contours

common fate tendency to group elements with the same motion

closure tendency to close gaps along boundaries

symmetry tendency to group elements that are symmetric to each other

Recognizing that these principles, even when combined, may not uniquely constrain the image's interpretation, Koffka, 1935 [21] proposed the concept that "of several geometrically possible organizations that one will occur which possesses the best, simplest and most stable shape. This is, of course, nothing but our law of prägnanz".

In addition, Wertheimer also proposed the principle of **familiarity**, which applies when elements can be grouped into a recognizable, meaningful whole.

familiarity tendency to group elements that are familiar together

As already mentioned, we draw an important distinction between familiarity and the other principles. Specifically, our thesis is not concerned with the high-level cue of familiarity associated with the top-down process, but focuses on grouping with low-level and mid-level cues like proximity, similarity, closure, and symmetry.

Of the cues above, of particular interest are mid-level cues, which grasp larger spatial scope and thus offer greater context, yet are harder to implement for tasks like figure-ground segmentation. Like other cues, mid-level cues capture non-accidental relations between image elements that are exhibited by all objects. They are less specific than a high-level object model, yet more discriminative than low-level cues like appearance similarity and contour continuity. There are two mid-level cues of interest to us. **Closure** [28, 47] is a regularity that favors regions that are enclosed by strong boundaries. Bottom-up approaches to finding closure vary in the types of cues used, and may include continuity and convexity. **Symmetry** [77, 84, 114] is a ubiquitous and powerful regularity with scope that spans entire objects or their parts. Perceptual grouping literature contains varied symmetry representations like the medial axis transform [10], generalized cylinders [9], superquadrics [90], and geons [8]. Later approaches applied symmetry toward cluttered and occluded image domains, and fall into filter-based and contour-based grouping approaches. The challenge of managing scene complexity is reflected in both closure and symmetry approaches to grouping.

More recent developments saw a rise in the use of low-level cues, especially pairwise color affinities, in the form of image segmentation algorithms [103, 32], while mid-level cues remained relatively difficult to implement. However, even bottom-up segmentation lost much of its appeal when machine learning algorithms rapidly advanced and spread, leveraging training data to optimize performance, and hugely benefiting object detection. Classifiers took fixed-length feature representations for which bounding boxes were convenient, and the community adopted the sliding window framework, in which classifiers were evaluated at every bounding box location. By performing an exhaustive but feasible bottom-up search of all possible bounding boxes, the adoption of sliding windows effectively displaced the role of perceptual grouping in object recognition.

Although bounding boxes are an effective way to reduce the search space, they have the disadvantage of being unable to capture boundary shape. As a result, the sliding

window approach offers little support for shape-based recognition. To illustrate the potential that is otherwise lost, we devote the entirety of Chapter 3 to a demonstration of the advantage of shape features for recognition. By building a system on perceptually grouped contours that are coarser than pixels yet sufficiently local, we learn and recognize object categories without relying on expensive bounding box annotations or exhaustive sliding windows.

The most recent developments have seen a comeback of bottom-up segmentation in the form of **region proposals**. It was proposed that even if the image could not be perfectly partitioned into the correct objects, one could still hypothesize a set of regions that include them. By outputting multiple hypotheses, individual mistakes mattered less, and hypotheses from the bottom-up stage could be disambiguated by a later stage that had access to higher-level information. The framework thereby enabled one to access the focus of attention provided by bottom-up segmentation without being limited by segmentation errors. This development led to a recoupling of state-of-the-art object detection with such variable-size region proposals [42], which arguably represents a return of bottom-up grouping cues in support of recognition. It is therefore an exciting time for perceptual grouping, in particular mid-level grouping cues, to make an impact on a wide range of applications.

Having equipped ourselves with the background of the field, we can now supply the issues listed above with the context in which they will be addressed:

- **Complexity**: In designing an approach to propose object locations in a way that provides better spatial support than the sliding window approach, potential ultimately lies in perceptual grouping. Current region proposal methods rely only on low-level grouping cues to generate regions, however, low-level cues can only grasp a small spatial context, and we need to turn to mid-level cues to improve the precision of the proposal set. If we can exploit cues like closure and symmetry, which have greater grasp of the image contents, using methods like discrete optimization

for combinatorial search, we can advance along this approach.

- **Variability**: We have shown that objects could vary in their appearance in many ways including different surface markings and distinct subparts, and that a grouping strategy as simple as one using color similarity alone is not enough. Similarly, we have also seen that a color model is likely over-sensitive to the details of a particular example and likely of little support for object categorization. Shape cues, however, capture image properties at a broader level and are a more robust alternative to appearance cues. There are, however, many distinct shape cues, such as the Gestalt principles, each of which provides a different source of information. Due to the wide variability in input, no single cue can always provide the best interpretation, and thus the need arises to combine information from different cues. Such cue integration will provide the potential to widen the domain of applicability beyond examples limited to the dataset at hand.

- **Learning**: Recent focus on the complementary problem of object detection has meant that once a perceptual grouping system is exposed to annotated training data, the dominant learning paradigm consists of single-output frameworks such as classification and structured prediction. In particular, there is a lack of methods that capture the multiple-output structure of bottom-up grouping, such as region proposals. Furthermore, previous perceptual grouping approaches tend to learn only small system components at a time and independently of each other, such as the pixel-wise probability estimates of contour maps or individual pairwise affinity functions in energy minimization approaches. In contrast, the approach of learning to group in the full image with multiple cues remains relatively unexplored. Until we can express the whole system in a trainable form, we will not have learned to do perceptual grouping.

1.1 Thesis outline

In order to address the above issues within a fixed context, we will begin with a chapter on object categorization. Chapter 3 introduces an object categorization framework, specifically highlighting issues of using appearance and shape in feature representations. By showing dramatic improvements from the use of shape features, the chapter illustrates the importance of feature representation in addressing the issue of variability. The chapter also addresses the prohibitive cost of manual bounding box annotation during learning by adopting automatic perceptual grouping, thus motivating the issue of computational complexity. With the motivating context for how bottom-up and top-down process might interact, we then proceed with the main outline of our thesis.

First, as summarized below in Section 1.2, we focus on symmetry, which is not only a powerful grouping cue, but a powerful shape regularity that has long been exploited to recover part structure without prior knowledge of scene content. We will address limitations of prior work in robustness to common variations in symmetry, notably by adding insensitivity to bending/tapering variations to widen the domain of applicability. We additionally formulate an optimization problem that is solved by a dynamic programming algorithm, resulting in an efficient search that makes fewer grouping errors than previous work.

Next, as summarized in Section 1.3, we address the problem of combining mid-level cues that are poorly or not integrated in previous work. Combining the cues of symmetry and closure allows us to extend the domain of applicability beyond images of purely symmetric objects or images purely of objects with closed boundaries. Additionally, we model cue combination as a trainable parameter in a predictive model that covers the full image. Thus, while other methods must manually design the combination or make predictions for only small parts of the image at a time, we learn automatically and jointly in the whole image.

Finally, as summarized in Section 1.4, we address the challenge of formulating a

learning framework for bottom-up grouping. In particular, we consider models commonly used for region proposal methods that produce multiple output hypotheses, such as minimizing a parametric energy function using parametric maxflow. When viewed as the output of a prediction function, it becomes clear that there are multiple outputs and a corresponding lack of suitable learning models for this type of prediction. To address the need for learning multiple-output models for bottom-up grouping, we will build on existing structured prediction frameworks for single-output models such as the structured support vector machine (S-SVM).

The text in Chapters 3, 4, 5, 6 is based on [61], [58], [59], [60], respectively.

1.2 Symmetric part detection

One of the most powerful indexing structures is a configuration of parts, in which a set of spatially related parts from the same object is recovered without prior knowledge of scene content. Such bottom-up recovery of generic parts can be traced back to the medial axis transform (MAT) [10], generalized cylinders [9], superquadrics [90], and geons [8], all representations that are grounded on symmetry, not only in its capacity as a grouping cue, but also as a structure for part recovery.

We build on previous work [65] that leverages medial symmetry for segmenting an image. Models of medial symmetry (*e.g.* the medial axis transform (MAT)) have proven to be a powerful object representation, in particular owing to its ability to provide a decomposition of the object into its parts. Despite its advantage in part modelling, however, such models assume segmented input and lack the ability to handle unsegmented images [106].

To this end, we provide a rigorous analysis of the link between maximally inscribed discs and superpixels (from the MAT and a superpixel oversegmentation of the image, respectively) as proposed in [65]. As shown in Figure 1.2, grouping the superpixels that

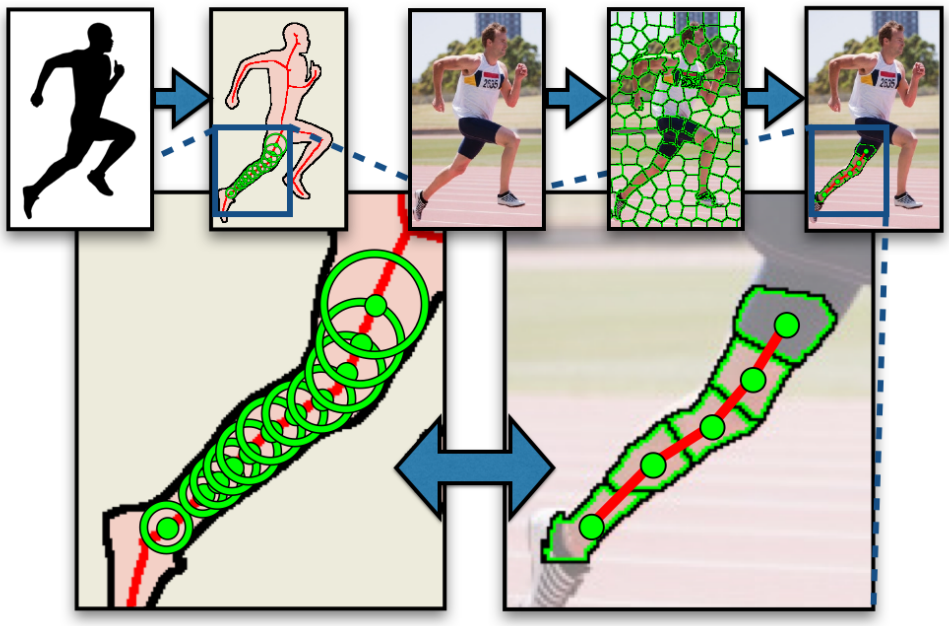


Figure 1.2: Our representation of symmetric parts: On the left, the shape of the runner's body is transformed into its medial axis (red), a skeleton-like structure that decomposes the shape into branch-like segments, *e.g.* the leg. The leg's shape is swept out by a sequence of discs (green) lying along the medial axis. On the right, the shape of the same leg is composed from superpixels that correspond to the sequence of discs.

compose an object part (*e.g.* a runner's leg) corresponds to tracing the maximal discs that sweep out the symmetric part.

While [65] outperformed previous approaches, it suffered from some limitations. First, we improve robustness by accommodating common forms of variability of symmetric parts appearing in diverse environments. While a straight medial axis was previously assumed, we introduce features that are insensitive to bending and tapering, thereby extending the domain of applicability to a wider class of inputs.

Additional value lies in allowing the grouping process to integrate disc hypotheses from multiple scales, in contrast with the previous restriction that discs could only be grouped when they came from the same scale.

Finally, we address limitations in the greedy nature of the grouping algorithm by

deriving a superpixel grouping problem from the framework that is solved by dynamic programming. Due to the correspondence between superpixels and maximal discs, the problem is framed as finding optimal sequences of maximal discs that sweep out symmetric parts. The new framework is demonstrated on two datasets, and shown to significantly improve performance.

1.3 Multi-cue grouping

Bottom-up grouping has re-emerged in the form of region proposals [15, 116], which typically start with a generation stage that uses a bottom-up grouping algorithm to output a diverse set of proposals, which are then passed to a ranking stage where they are evaluated by a trained scoring function. In doing so, region proposal methods forward bottom-up ambiguity from the generation stage to the ranking stage in the form of proposals, at which point stronger cues are available to reduce the ambiguity.

Typical methods like [15, 116], however, rely on only low-level appearance and contour cues to generate proposals, and as a result must diversify their proposals in large quantities to preserve recall. We present a complementary approach to diversification that uses mid-level grouping cues to resolve ambiguity at an early stage to avoid the need to generate proposals in excessive quantities.

By approaching the problem as figure-ground separation, we draw on a large body of work in perceptual grouping. Mid-level cues capture non-accidental relations between image elements that are exhibited by all objects. They are less specific than a high-level object model, yet more discriminative than low-level cues like appearance similarity and contour continuity.

Closure [28, 47] is a regularity that favors regions that are enclosed by strong contour evidence along the boundary. There are many bottom-up approaches to finding closure and involve different cues like continuity and convexity. A common way to formulate the

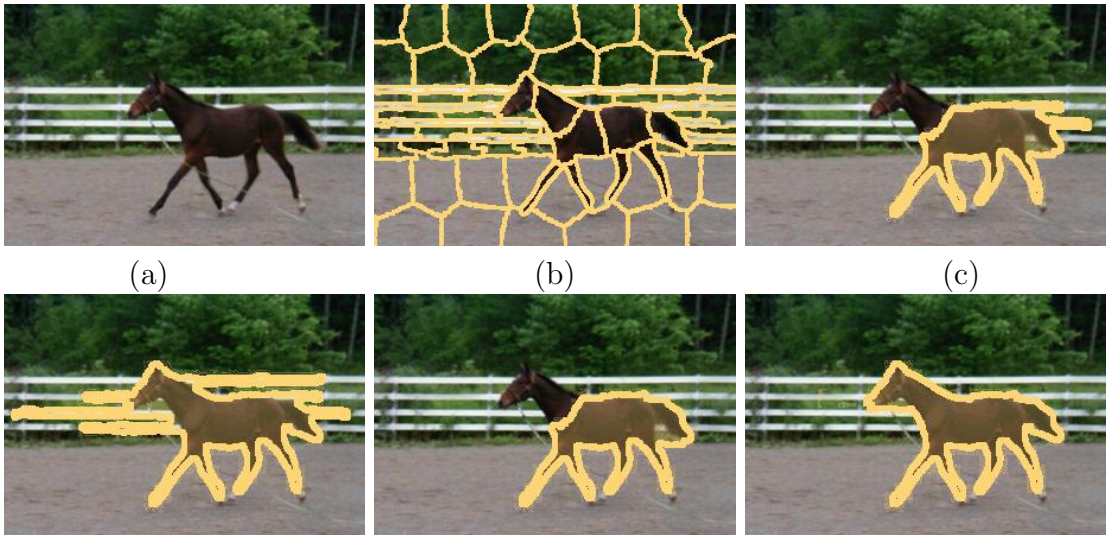


Figure 1.3: Given an input image as shown in (a), our method first oversegments into superpixels in (b), which are to be grouped into regions based on a combination of perceptual grouping cues. In this example, both the horse and the fence are relatively homogeneous in color and exhibit contrasting boundaries, however the horse's neck is slightly darker than its torso. As shown in (c), low-level appearance alone oversegments the horse at the neck where a large gap in contour is attempted. When including contour closure in (d), the boundary correctly encloses the head, but elsewhere strays along the fence. Conversely in (e), including symmetry without closure separates the fence from the horse, but fails to enclose the head. With closure and symmetry together in (f), the entire horse is correctly segmented.

problem is to find a cycle of graph edges in a very large space.

Symmetry [77, 84, 114] is a ubiquitous and powerful regularity with scope that spans entire objects or their parts. Since the early days, perceptual grouping research has produced numerous representations such as the medial axis transform [10], generalized cylinders [9], superquadrics [90], and geons [8]. Later approaches applied symmetry toward cluttered and occluded image domains, which are much more difficult to handle due to increased noise in the background.

We combine low-level appearance with mid-level cues of symmetry and closure, which are otherwise poorly or not integrated in previous work. As shown in Figure 1.3, mid-level cue combination allows situational use of symmetry and closure, thereby extending the domain of applicability beyond images of purely symmetric objects or images purely

of objects with closed boundaries.

Additionally, we model cue combination as a trainable parameter in a predictive model. Thus, while other methods must manually design the combination, we are able to leverage powerful learning algorithms to obtain the best combination of cues. Specifically, cues are combined by summing their respective energy terms in a parametric energy function of the following form:

$$E^{\lambda}(y; w) = \sum_{p} U^{\lambda}(y_p; w) + \sum_{p,q} V(y_p, y_q; w) \tag{1.1}$$

To learn the parameters w of (1.1), we view the minimization of (1.1) (for all values of parameter λ simultaneously) as the inference step of a structured prediction framework. Grouping cues are combined into (1.1) which is learned as a whole, in contrast to previous methods that learn energy terms independently from each other.

1.4 Learning to generate mid-level region proposals

In Chapter 6, we build on the previous chapter to address the challenge of formulating bottom-up grouping as a learning problem. Specifically, we develop a new learning framework that accurately captures the ambiguity of bottom-up grouping. Ambiguity is an important characteristic of bottom-up grouping that defers the verification of hypotheses to a later stage that has access to more information, such as object-level knowledge. This is more generally known as the principle of least commitment, and is reflected in the region proposals framework.

We will formulate a learning problem for and in the context of the parametric energy model in Equation (1.1). In particular, we argue that it is important that the loss function capture quality not based solely on a single answer, but based on the set of multiple output hypotheses as a whole. This will require some innovation as conventional learning models do not support the required problem structure.

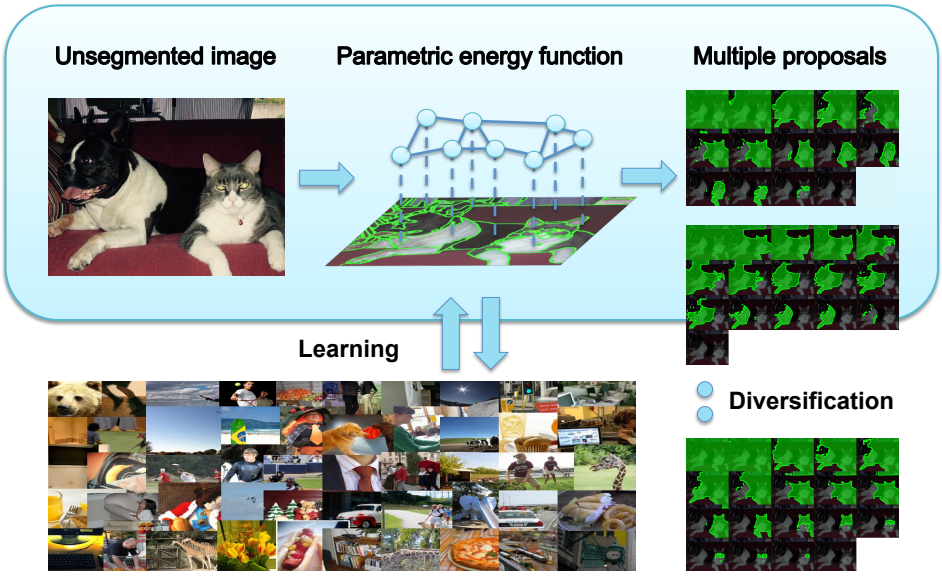


Figure 1.4: Our approach takes an input image, partitions it into superpixels, and groups superpixels into region proposals using a novel structured learning framework for parametric energy functions, called Parametric Min-Loss (PML). The parametric energy function combines mid-level cues with weights that are trained to generate multiple region proposals. Finally, we diversify the energy function to generate a diverse set of region proposals.

Specifically, a conventional loss function $\ell(y, y)$ that measures the discrepancy between a single prediction and the correct answer, such as that used in [111], is not directly suitable for prediction functions that output multiple hypotheses. This includes our model-output model of interest, namely, the prediction function that minimizes $E^{\lambda}(y; w)$ in y to yield a distinct solution for each value of λ. Consequently, approaches that train $E^{\lambda}(y; w)$ with a conventional loss function ℓ, like [59], do not use the model in a way that it was trained for.

To develop a suitable loss function, we take an approach similar to Multiple Choice Learning (MCL) [45], which proposed a loss function $\mathcal{L}(Y, y)$ that captures the problem structure by measuring the discrepancy between a set Y of output hypotheses and the

correct answer, and implemented the loss $\mathcal{L}$ in the structured support vector machine (S-SVM) framework. The resulting learning problem yielded a minimization problem in two blocks of variables, in turn yielding a block-coordinate descent learning algorithm. In our work, we follow this approach to derive a learning model for (1.1), and show that it yields a block-coordinate descent learning algorithm that decomposes into simpler subproblems.

In the context of bottom-up grouping, our learning formulation apparently contradicts the diversification component that drives the prototypical region proposal method: we seek a single model with minimum weight, while diversification keeps all hypotheses that are explored. On the contrary, we show that our learning formulation is not only compatible with, but complementary to diversification. Specifically, the model (1.1), for a particular setting of weights w, specifies a way using λ to diversify its outputs, and thus learning enables us to choose the best way to diversify output hypotheses to include the correct outputs.

Finally, we note that the learning model is developed for (1.1), a model that forms the basis of many region proposal methods, and in particular supports the modelling of grouping cues as component energy terms. Thus the learning framework is consistent with perceptual grouping using grouping cues such as symmetry and closure. An illustration of our full approach is provided in Figure 1.4.

1.5 Review of contributions

Our thesis contributions are supported by an initial chapter on object categorization:

- An object categorization framework that includes automatic perceptual grouping throughout learning and detection, eliminating the dependence on expensive manual annotations.

- Dramatic improvements from the use of shape features for object categorization demonstrate the importance of feature representation in accommodating variability.

As part of our main contributions, we proceed with our work on symmetric part detection:

- A rigorous analysis of the link between maximally inscribed discs (of the MAT) and superpixels (in a superpixel oversegmentation).
- A derivation of a superpixel grouping problem framed as finding optimal sequences of maximal discs that sweep out symmetric parts, and efficiently solved by dynamic programming.
- An improvement in robustness by accommodating common forms of variability of symmetric parts, namely bending and tapering, that appear in diverse environments.

In our work in combining multiple mid-level grouping cues, we made the following contributions:

- A method for combining mid-level cues of symmetry and closure, which were previously poorly or not integrated, and thus allows the domain of applicability to be extended beyond purely symmetric objects or objects fully bounded by contours.
- Modelling cue combination as a trainable parameter in a predictive model that jointly combines all cues, which allows the use of machine learning algorithms to find the best combination of cues.

Our final contributions lie in our work in learning to generate bottom-up regions:

- Parametric Min-Loss (PML), a novel learning framework for the parametric energy function (1.1) that uses a loss function to score multiple output hypotheses, enabling us to correctly capture the structure of bottom-up grouping.
  - An exposition in standard notation supports the application of PML to other domains that use parameterized energy functions of the form (1.1).

- An efficient diversification strategy suitable for the model, demonstrating that our learning framework is not only compatible with, but complementary to the technique of diversification.

- An overall novel learning approach for perceptual grouping, demonstrated by modelling mid-level grouping cues in the model (1.1) and achieving results comparable with recent region proposal methods.

Chapter 2

Related Work

Approaches to bottom-up grouping differ not only in the type of perceptual cues they use, but also how they are used to group elements together, and how they interact with other cues. We begin with an overview of perceptual grouping cues in Section 2.1, followed by contour-based methods in Section 2.2 which utilize geometric relations, then region-based methods in Section 2.3 which emphasize photometric features, and finally symmetry-based methods in Section 2.5 which are of interest due to their higher-order scope.

2.1 Perceptual grouping cues

Scientists have long studied bottom-up grouping as a phenomenon of human vision. To explain how we automatically focus on certain visual structures in a scene image (whether still or moving), the Gestalt school identified sets of features common to those structures. Wertheimer, 1923 [119] produced such a list: "proximity, similarity, uniform density, common fate, direction, closure, good curve, past experience, habit", which were illustrated with monochrome drawings of shapes and patterns. The Gestaltists thus hypothesized geometric cues, independent of specific objects, that are used to group elements together. In an age before computers were widely available, their theory lacked

an effective computational design.

Since the Gestaltists, a number of works have investigated an underlying principle for these object-independent cues. Witkin & Tenenbaum, 1983 [122] argued that spatiotemporal coherence and regularity are reflected in these cues, and that they are non-accidental occurrences that indicate an underlying cause. Suppose that we measure a feature s (*e.g.*, parallelism), and consider the likelihoods $P(s|\text{obj})$, representing the relative frequency of s in the world as caused by an object, and $P(s|\text{acc})$ in the world as occurring by accident. Since $P(s|\text{acc})$ is so low, the alternative explanation almost certainly holds, that it arises from an object. Lowe, 1985 [75] notes that structures are themselves projected from the 3D world, and thus viewpoint-invariance plays a role in the likelihood $P(s|\text{obj})$. Uncertainty also arises from the fact that observed structures are never perfect (*e.g.*, near-symmetry), yet there is a point when the resemblance is so rough that it is likely to be disregarded as an instance of the structure (asymmetry).

Koffka, 1935 [52] proposed the concept that "of several geometrically possible organizations that one will occur which possesses the best, simplest and most stable shape. This is, of course, nothing but our law of prägnanz." This leads to another principle for seeking one of alternative structures $\{\sigma_1, \sigma_2, \ldots\}$ to explain the same set x of elements. The minimum description length (MDL) principle selects the simplest alternative with respect to some uniform description of the alternatives. Modelling the probability that x is explained as a particular structure as $P(\sigma|x) \propto P(x|\sigma)P(\sigma)$, the simplicity criterion is expressed by the prior $P(\sigma)$. While the MDL principle achieves the best trade-off between goodness-of-fit and overfitting, it however does not provide a way to come up with the description itself.

The Gestalt school spawned some key ideas of perceptual grouping, but lacked a computational procedure. In addition, we must work with real images and tackle the problem of measuring such geometric features, which is far from straightforward. Their ideas, however, point forward to probability theory and optimization as ways to formulate

and solve the problem.

2.2 Contour-based grouping

Many approaches use contours as a starting point for perceptual grouping. Coherent objects in space project to bounded regions in images, suggesting that objects can be segmented by finding their bounding contours. Typically, an initial stage computes an indication of contour at each pixel. Even when restricted to contour pixels, the search space is too large for exhaustive search, and most approaches use contour proximity and continuity (curvilinearity) to group contour pixels together.

2.2.1 Contour pixel detection

The method of Canny, 1986 [13] uses a photometric model to derive an oriented edge filter. Since the filter is not aware of alternate sources of edges such as illumination and surface marking boundaries, the method is an edge detector. However, contour pixels can be found among the detected edges. Modelling an ideal edge as a step function with additive Gaussian sensor noise, the method uses signal-to-noise and localization criteria to derive the filter. The method convolves the intensity image with a Gaussian-smoothed derivative filter in x and y, and combines the results to obtain the intensity gradient at each pixel. Non-maximum responses are suppressed along the orthogonal direction to find peaks, and a technique called hysteresis thresholding is performed to form chains of edgels using both adjacency and response level information. As an intensity gradient-based method, the Canny detector has low recall where boundaries are faint, and low precision where non-object boundaries are found, thus yielding fragmented and spurious contours.

More sophisticated contour pixel detectors extract a richer set of features to compute the response at each pixel. The Pb detector of Martin *et al.*, 2004 [81] extracts brightness,

color, and texture features in oriented neighbourhoods around each pixel by dividing a disc into halves at multiple orientations. The features are combined into the probability of boundary at each pixel, via parameters that are trained on a set of images annotated with ground truth boundaries. This method uses wider-scoped features in comparison to [13], and outperforms the latter on the BSDS benchmark. An improved method, gPb, in Maire *et al.*, 2008 [80] incorporates global spectral cues from a contour map derived from solutions to a generalized eigenvalue problem for a modified affinity matrix.

2.2.2 Contour grouping

We now cover contour-finding methods that minimize an energy, typically over paths defined on a graph. For example, the active contour framework of Kass *et al.*, 1988 [50] defines an energy function over splines (called snakes) that decomposes into low-order derivative terms (for smoothness and curvature) and image feature terms (attraction to intensity edges). The method requires the user to supply an initial snake, though multiple initializations could be generated automatically from higher-level knowledge.

The global structural saliency method of Ullman and Sha'ashua, 1988 [117] places a grid of vertices on image pixels and sparsely connects nearby vertices with an edge weighted by a filter response. A saliency objective Φ over curves is defined to sum over unary terms that favor high weights and binary terms that favor low curvature. The algorithm iterates locally additive operations N times to maximize the curve of length N ending at each edge, and to compute the curve saliency at each edge. While the method is efficient, it requires the value of N to be supplied, and the objective Φ is biased towards longer curves. The bias is removed by normalizing by curve length, as done by the cost density objective in Felzenszwalb & McAllester, 2006 [34], for which a dynamic programming algorithm computes the global minimum.

While the above methods are principled in combining multiple criteria such as contour pixel strength and contour continuity into a single optimization objective, the objective

Figure 2.1: Felzenszwalb & McAllester, 2006 [34]: Given a colour image, minimum cost density curves are extracted.

does not guarantee that the contour solutions bound a region. Resulting contours are likely to be open due to missing contour pixels and branching ambiguities. Nonetheless, fragmented contours can still be grouped or identified by a higher-level process, for example shape-based object recognition in Ferrari *et al.*, 2006 [36], and object discovery in Payet & Todorovic, 2010 [88].

2.2.3 Bounding contours

Graph-theoretic methods that detect bounding contours find cycles. The method of Elder & Zucker, 1996 [28] fits tangent lines of maximal length to edgel chains, and forms a graph of tangents where edges connect each tangent to the 6 best neighboring tangents, and each edge is assigned a link likelihood based on rectilinear completion features. The maximum likelihood cycle in the graph is found by an efficient solution to the shortest-paths problem between an edge's vertices for every edge. Cycles are found quickly, however the objective is biased towards shorter cycles.

The spectral method of Zhu *et al.*, 2007 [125] also uses shortest-paths to find closed contours, but in an abstract space defined by the first complex eigenvectors of a modified edgel affinity matrix. Representing the eigenvector's entries on the complex plane, the method looks for a circular structure among the entries that corresponds to a closed

contour.

A unique approach is taken by the convexity method of Jacobs, 1996 [47]. Directed line segments are fitted to edgel chains, and gap line segments are inserted between existing line segments. A backtracking algorithm explores promising extensions to paths of line segments while enforcing a global convexity constraint over the line segments. As a non-accidental cue, convexity encourages figure-ground separation even in the absence of other cues. However, since not all objects appear convex, the method suffers from low recall.

Jermyn & Ishikawa, 2001 [49] proposes a class of energy functions of cycles on graphs that have efficient global solutions. The energy function takes a ratio form with a numerator that encourages the boundary slope to be orthogonal to an underlying vector field, normalized by the boundary length. A discrete minimization algorithm is given that computes the global minimum in polynomial time, along with an alternative algorithm of lower complexity for a denominator of restricted form. The method is optimal for a general class of energies, however its effectiveness depends on reliable image features.

Contour continuity is a boundary regularity that influences contour grouping. As an enclosure of a region, contour closure is an important grouping cue, but open contours can also be used for higher-level processing. A drawback of contour-based methods, however, is a reliance on an accurate contour map, which is difficult to obtain without higher-level knowledge itself. Additionally, potentially useful information from the region interior does not enter the grouping process at all.

2.3 Region-based grouping

While contour-based methods focus on geometric relations among contour pixels, region-based methods exploit photometric relations among pixels to cluster them together. Typical approaches use colour and texture similarity, often in the form of pairwise affinities,

to group pixels together.

2.3.1 Image partitioning

Feature vector clustering. Viewing an image of pixels as a set of feature vectors, a clustering algorithm can be applied to obtain a partition into groups. For example, the EM algorithm can be used to fit a mixture of K Gaussians to the feature vectors. The method, however, is sensitive to initialization and requires K to be supplied. A more flexible method called mean-shift in Comaniciu & Meer, 2002 [20] finds individual densities without requiring K. A specified kernel function is used at each iteration to weight nearby points for re-estimating the mean, and two points belong to the same cluster if their mean-shift trajectories converge to the same mean. While the feature space approach allows a range of powerful tools to be applied to segmentation, they are not the best at expressing boundary criteria between segments.

Watershed transform. A graph-theoretic view of pixels as vertices encodes adjacency relations on edges which, for example, can be placed to connect each pixel to 4 or 8 neighboring pixels. The watershed transform computes a segmentation map from an input map by viewing it as an elevation map. The transformation is computed by finding pixels that locally minimize the elevation, and "flooding" the elevation surface into disjoint basins that are eventually bounded by watersheds. In the linear-time implementation of Vincent & Soille, 1991 [118], pixels are sorted and labelled in order of increasing elevation. Each iteration at a given elevation preserves the property that all pixels below the given height are already labelled by basin membership. New pixels are enqueued to gradually expand the basin by letting pixels neighbouring existing basins take on the same label, and pixels found to be equidistant from two minima are labelled as watershed boundaries. Despite its efficiency, the transform will produce as many over-segmented regions as the number of local minima in the elevation map, and thus its effectiveness crucially relies upon the input map.

Figure 2.2: Arbeláez *et al.*, 2011 [4]: Given a colour image, the gPb contour map is computed (in heat map colours) and the watershed transform run from the local minima (red points).

The method of Arbeláez *et al.*, 2011 [4] uses the watershed transform to produce a hierarchical segmentation. The gPb detector [80] is used to compute the input, which yields an oversegmentation into small watershed regions, which are then greedily merged. Affinities between regions depend on contour strength, and are refined by enforcing orientation consistency between contours and watershed arcs. The method produces a hierarchy of nested segmentations ranging from fine to coarse scales.

Graph partitioning. Graph-theoretic approaches commonly solve a graph partitioning problem to compute a segmentation. For example, a bipartition of vertices into figure and ground segments can be computed by solving the source-sink minimum cut problem in polynomial-time. The solution is efficient, but this approach requires the source and sink vertices to be supplied by the user, though there are methods that automatically propose source-sink pairs, *e.g.* using spectral embedding techniques in Estrada

et al., 2004 [30]. Since the cut cost is biased toward small regions, however, a disadvantage of this approach is that it can lead to oversegmentation.

The normalized cut cost of Shi & Malik, 2000 [103] removes the bias and is approximately minimized by a spectral clustering algorithm. (There are faster approximations to the intractable problem, such as the linear-time recursive coarsening method of Sharon *et al.*, 2000 [102], however the spectral approximation is more widely known.) The generalized eigenvalue problem for a modified affinity matrix is solved, and the second eigenvector is used to compute a bipartition, and resulting partitions are recursively bipartitioned. The affinity matrix captures brightness, colour, and texture similarity between two elements, and has also been extended to account for the intensity gradient in Leung & Malik, 1998 [64]. A pairwise affinity between two points is computed in terms of the maximum response along the line joining the two elements. Curvilinear continuity is incorporated by propagating scores along the contour and updating the affinities. The framework is general and has a polynomial-time solution with appealing theoretical properties, however the time complexity requires sparse affinities, limiting the scope over which to compute perceptual cues.

An efficient $O(N \log N)$ method is proposed in Felzenszwalb & Huttenlocher, 2004 [32] that greedily merges elements together. The algorithm iterates through all edges in sorted order, maintaining a forest of minimum spanning trees (MSTs). An edge joins two MSTs only if its weight does not increase a "regularized" local variation of either MST. The algorithm is fast, but the resulting partitions tend to undersegment objects over faint boundaries.

Energy minimization. A general approach segments an image by assigning a label y to each element $x \in X$ that indicates its membership to a particular segment ($y \in \{1, 0\}$ in the case of figure-ground labeling), by minimizing an energy function $f(Y; X)$ that maps each possible labeling Y of X to its energy. A common form decomposes f into potential functions of up to two labels at a time, where each unary potential $f(y_i; x_i)$

assigns a cost for labeling x_i (*e.g.* via a discriminative classifier with colour features), and each binary potential $f(y_i, y_j; x_i, x_j)$ assigns a cost for combinations of neighboring labels (*e.g.* to favour contrast-sensitive smoothness). Kolmogorov & Zabih, 2011 [54] showed that when an energy of binary-valued labels decomposes into submodular potentials of order 2 or 3, the energy can be globally minimized by solving source-sink minimum cut. Efficient algorithms for other forms of energies exist, but a low-order decomposition is generally needed to get an efficient solution, trading off the representational power of higher-order potentials. While one can experiment with algorithms for denser connectivity and higher-order potentials, an alternative strategy is to label entire superpixels at a time, not only reducing the number of labels but providing spatial scope for higher-level cues.

2.3.2 Superpixel oversegmentation

While segmentation methods can quickly provide a compact representation of an image, unfortunately, they are generally marred by over- and undersegmentation errors. These errors reduce the suitability of the resulting segments for direct use as object hypotheses. Segmentation methods, however, remain useful in the form of "superpixels", which places high importance on boundary recall, and thus will allow for as much oversegmentation as needed in order to eliminate as much undersegmentation as possible. In other words, even though precision is lowered by spurious boundaries, superpixels have the advantage of potentially capturing all boundaries in an image.

Some of the above methods, such as Felzenszwalb & Huttenlocher, 2004 [32] and Arbeláez *et al.*, 2011 [4] are used to compute superpixels by biasing parameters toward finer segments, while superpixel algorithms typically enforce compact, uniform size by adapting one of the methods above. For example, the feature space clustering method SLIC of Achanta *et al.*, 2010 [1] is based on k-means; the graph-theoretic approach of Ren & Malik, 2003 [95] is based on normalized cuts; and the region-growing method of

Figure 2.3: Achanta *et al.*, 2010 [1]: Superpixels of increasing uniform size.

Levinshtein *et al.*, 2009 [69] is based on geometric flow. Superpixels offer a compact, "lossless" representation of an image that reduces the grouping complexity of further processes.

2.4 Region proposals

Region proposals is a framework that, unlike the segmentation methods discussed above, aims to produce multiple groupings (*i.e.*, figure-ground segmentations) rather than a *single image partition*. This can be motivated from the principle of *least commitment*, which states that decisions should be delayed for as long as needed, so that when they are taken their probability of correctness is maximized. In the context of bottom-up grouping, one can argue that image partitioning requires a final decision to assign each pixel to a unique segment without using higher-level information (*e.g.*, an object category), thus violating the principle. The correct approach would therefore be to produce multiple groupings as a hypothesis set, and propagate these forward to a subsequent stage that

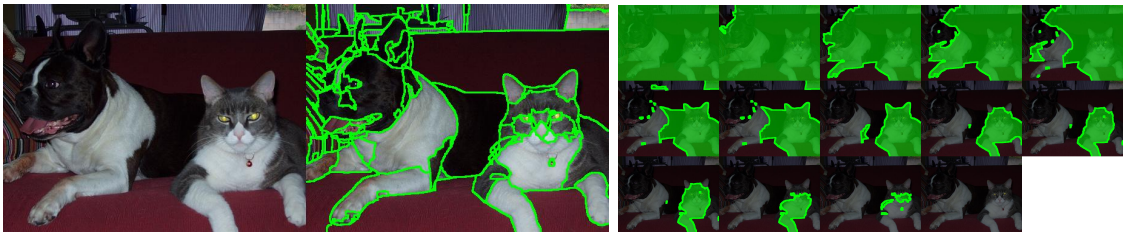


Figure 2.4: Clarifying the difference from grouping hypotheses from segmentations: a) in input image, b) a segmentation, and c) region proposals. Note that segmentation and regprops both provide hypotheses of objects at different image locations, segmentation forces a fully unambiguous bottom-up grouping of pixels, while regprops captures the ambiguity inherent in bottom-up grouping by allowing groupings to overlap.

does have access to higher-level information (*e.g.*, an object detector). See Figure 2.4 for a visual comparison of region proposals and image partitioning.

Viewing region proposals as object hypotheses for recognition, we begin by broadening our scope to include methods designed for sliding window detectors. Among these, the *objectness* detector of Alexe *et al.* [2] computes low-level features on superpixels [32] to score sampled image boxes. *Selective search* of Uijlings *et al.* [116] outputs boxes that bound regions generated from agglomerative clustering of superpixels [32]. The method accumulates a pool of regions over each step of region-merging until all regions are merged together, and ensures diversity by pooling results over multiple color and texture feature spaces. The method is very fast, yet is based on low-level appearance alone.

Arbelaez *et al.* [5] produces regions by merging superpixels of [4] over multiple scales. The method considers a limited number of all pairs, triples, and quadruples of adjacent superpixels. Our approach is different in that we operate on a single layer of compact superpixels, and define a set of low-level and mid-level cues that quantify the likelihood of grouping.

The *shape sharing* method of Kim & Grauman [51] matches part-level regions in a given image to a bank of exemplars, which project object-level information back into the image to help with segmentation. The *category-independent proposals* of Endres & Hoiem [29] develops a CRF model to label superpixels based on segment seeds. The

resulting region proposals are ranked using structured learning on grouping cues. The energy potentials are pairwise and submodular, and inference is done by graph cuts. While we use a similar procedure to generate regions, we combine mid-level cues at the front-end without seeding from a fixed hierarchical segmentation.

The *CPMC* method of Carreira & Sminchisescu [15] generates regions directly from the image rather than deriving them from a fixed segmentation. The method solves multiple parametric min-cut instances over color seeds. Regions are re-ranked by regressing on overlap with region-scoped features, including mid-level features such as convexity and eccentricity. The emphasis is on ranking rather than the front-end grouping, which samples color seed models over millions of pixels. Our approach is qualitatively different from the above methods as we focus on bottom-up grouping, however our mid-level front-end is complementary to the ranking stage.

2.5 Grouping by symmetry

Contour- and region-based methods typically resolve ambiguity using low-level cues such as contour continuity and colour and texture similarity. These cues, however, do not accurately constrain the grouping problem on their own, and a large number of hypotheses must be provided to achieve sufficient detection recall. It is thus interesting to pursue higher-order cues, such as symmetry, which can verify the strength of lower-level hypotheses without assuming object-specific knowledge.

Symmetry combines lower-level elements such as region and contour features, ideally into part- or object-level scopes that lead directly to part or object representations. The classic medial axis transform (MAT) in Blum, 1967 [10] composes an object from its symmetric parts, and leads to powerful representations as surveyed in Siddiqi & Pizer, 2008 [106]. The higher-order scope of symmetry, however, while difficult to encode into a computationally efficient framework, has been shown to improve performance in transfer

learning [110].

Notions of reflectional symmetry can be traced back to the MAT: in the set of maximally inscribed discs that defines the transform, each disc is tangent to a pair of points on the shape's boundary. While the MAT is an invertible transform that can serve as a powerful shape representation [106], it suffers from high sensitivity to slight changes in the boundary. A stable alternative to the MAT is offered by smoothed local symmetries (SLS) in Brady & Asada, 1984 [12], in which a pair of corresponding boundary points is related by a line segment. Common to these symmetry transforms, however, is the unrealistic assumption that a closed region is already available for applying the transform. Hence, these methods are not directly useful for detecting symmetry in real, cluttered images. Nonetheless they are conceptually a part of different models for symmetry detection, which we divide into three groups: 1) reflection axis models, 2) medial point models, and 3) symmetric shape models.

2.5.1 Reflection axis models

The first type of symmetry model describes an axis of reflection that relates pairs of elements across the axis. An element can be any local feature or a contour point. Having generated a set of hypothetical pairs of available elements, each pair is then used as evidence for a specific axis. As an example, Mohan & Nevatia, 1992 [84] finds symmetric pairs of curves. After grouping edgels into curves, pairs are hypothesized by considering all possible pairs, and using symmetry-based heuristics to prune out implausible pairs. Each pair defines an axis, which is encoded as a node in a constraint satisfaction network. Each node is weighted by the quality of the axis, and a global cost function is formulated over the network for selecting a consistent set of axes. Similarly, Saint-Marc *et al.*, 1993 [97] uses a B-spline parameterization of curves to calculate symmetry from each pair of curves, including skew symmetry axes. These approaches both work with curves and therefore reduce grouping complexity and take advantage of their geometric

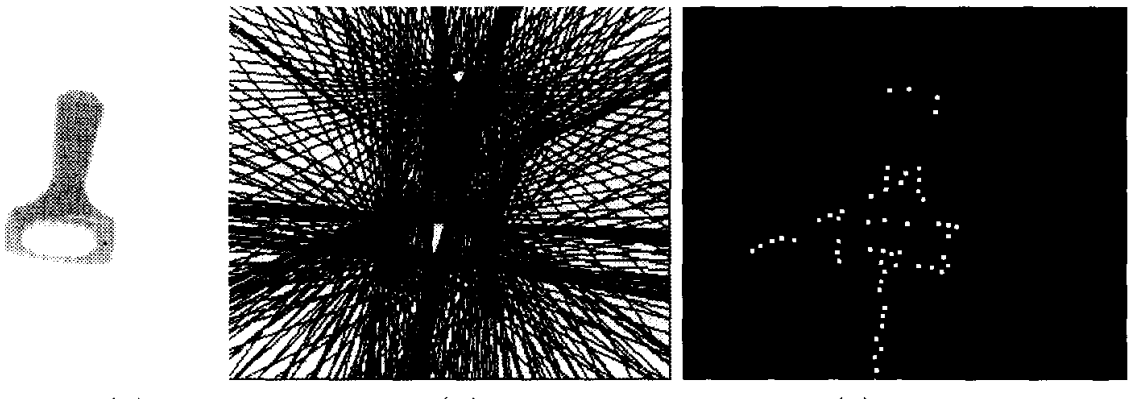


Figure 2.5: Cham & Cipolla, 1995 [17]: Given an intensity image, tangent lines from all oriented edgels are intersected, producing points on possible symmetry axes.

representation, but they depend on accurate bottom-up grouping of contours, a difficult problem in itself.

Other methods work directly with contour points and find a sequence of point pairs along the same axis. Cham & Cipolla, 1995 [17] use a voting framework in which each pair of contour points (sampled from B-spline curves) assigns linear axis parameters and an angle of skew to a local skewed symmetry field. An additional heuristic is applied using a linear-time voting scheme for an axis based on intersecting tangent lines. While the pruning heuristic is fast, it does not respond to parallel symmetry. Taking a different view, Liu *et al.*, 1998 [74] frames an optimization problem over sequences of pairs of edgels. All possible pairs are captured as graph vertices, and edge weights favour smoothness along the axis and boundary, axis perpendicularity, and gradient magnitude. The solution to the shortest-paths problem is used to return an optimal sequence, however a manual initialization is required.

An intermediate approach works with line segment approximations of contours. Ylä-Jääski & Ade, 1996 [124] finds a sequence of line segment pairs (LSP) by defining a series of pruning rules and agglomeratively grouping combinations of LSPs. Stahl & Wang, 2008 [109] similarly finds a sequence of trapezoids, defined by line segment pairs, and

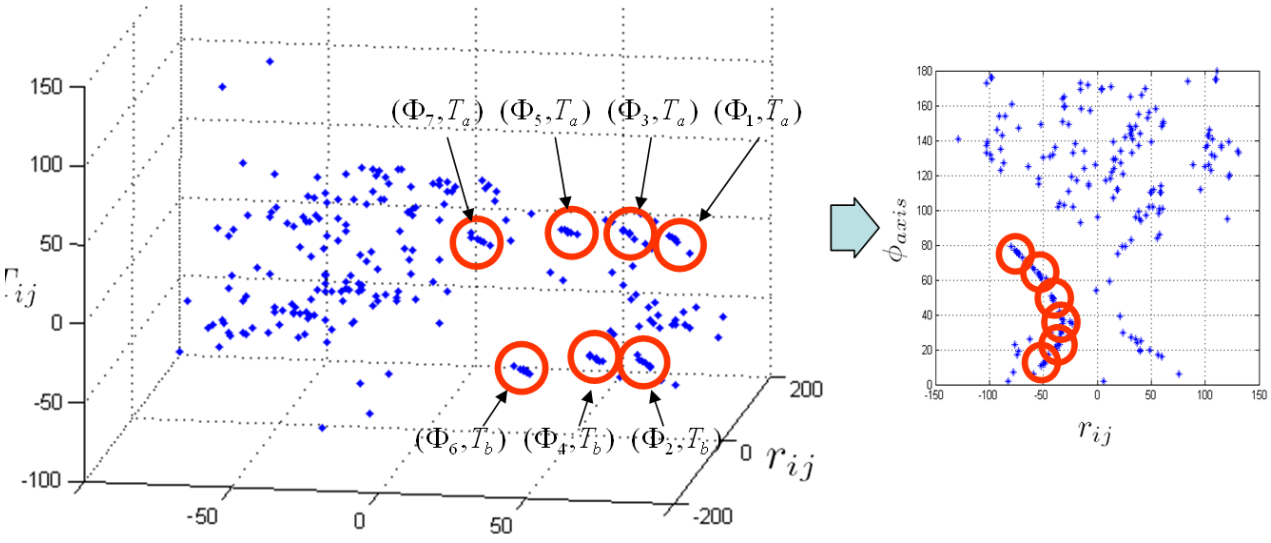


Figure 2.6: Lee & Liu, 2012 [57]: Votes casted (blue points) from glide symmetry units form dense clusters (red circles), and when projected into 2-space can be easily grouped.

gap-filling quadrilaterals, and incorporates this into a more principled, ratio-cut cost that is globally minimized. While the cost function incorporates closure, an important cue, it is still built on a prohibitively large number of possible trapezoids and quadrilaterals.

With the advent of repeatable interest point detectors and their matching capability, a line of methods uses symmetry to group them. Loy & Eklundh, 2006 [77] use a voting framework based on matching SIFT features within the same image. From all possible pairs of SIFT features, the best matching "mirrored" pairs are selected, and each pair votes for a linear axis. Lee & Liu, 2012 [57] builds on this method by handling glide reflection using an extra translation parameter, and curved symmetry with piecewise linear approximation. While the distinctiveness of local features offers an effective way to narrow down promising pairs, they critically depend on interest point detection, which typically does not respond well in regions of low-texture. On the other hand, computing features more densely will explode the pairing complexity. Common to all approaches using an axis of reflection is the challenge of enumerating promising pairs of elements. Without the help of other cues, the number of hypotheses grows quadratically, constraining the efficiency of the grouping algorithm.

2.5.2 Medial point models

The medial point approach to symmetry detection decomposes a symmetric structure into a locus of maximal discs, and groups together disc hypotheses that lie on the same medial axis. The detected symmetric parts can then be grouped to compose an object.

Methods in this category vary in the structure of the underlying discs. Filter-based methods detect disc hypotheses by thresholding the response at every point. For example, Crowley & Parker, 1984 [23] computes a pyramid of filter responses, over which peaks and ridges are located, corresponding to medial axes, and linked into a tree structure. Shokoufandeh *et al.*, 2006 [104] presents a recognition framework that uses a directed acyclic graph over a multi-scale blob decomposition of an image. To reduce grouping complexity, medial point approaches can compute disc hypotheses using interest point detectors like the auto-scale blobs and ridges of Lindeberg, 1998 [72], and affine-invariant interest point features of Mikolajczyk & Schmid, 2002 [83]. The initial step of proposing disc hypotheses critically affects the performance of medial point approaches. Since many filters are point-based, they tend to discard region boundary information, yielding both false positives and false negatives.

The more recent method of Tsogkas & Kokkinos, 2012 [114] defines a geometric filter composed of a 3-rectangle template, which is used to compute the probability of a skeleton point at every pixel using a classifier with symmetry-based features. A new dataset with annotated skeleton points was introduced, used to train the template classifier and to demonstrate an improvement in computing a skeleton filter map. While the method is parallelizable, the bottleneck lies in trying all possible orientations and scales of the template.

An alternative way of generating disc hypotheses is by detecting regions corresponding to discs. For example, the method of Levinshtein *et al.*, 2009 [65] hypothesizes discs directly from a superpixel segmentation of the image. Superpixel disc hypotheses are represented by a graph, and a trained affinity is assigned between adjacent discs, allowing

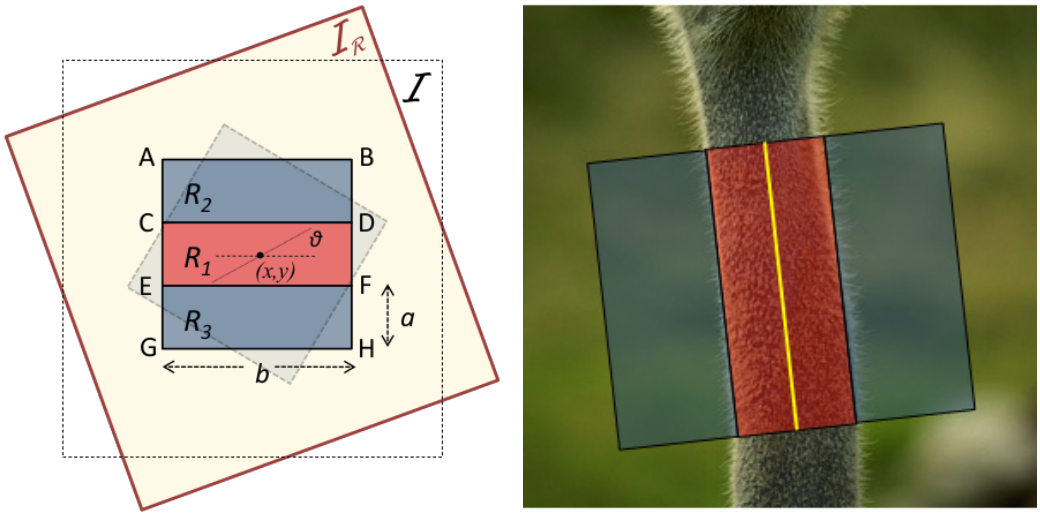


Figure 2.7: Tsogkas & Kokkinos, 2012 [114]: A symmetry template matched at a given point at the correct scale and orientation.

discs to be grouped by a solution to a graph partitioning problem. Because symmetric parts exist at unknown scales, the graph partitioning step is repeated over superpixels of different scales. As with the filter-based methods, performance depends on the quality of the underlying superpixels.

The method of Narayanan & Kimia, 2012 [86] detects medial fragments which, like superpixels, roughly correspond to maximal discs. They are computed by the shock graph defined by the contours of the entire image. Due to the presence of false positives and false negatives in contour maps, however, the resulting medial fragments are over- and undersegmented, and the method merges and splits them according to a pre-defined set of transformations such as gap completions, loop removals, and occlusion removal. A containment graph manages the exponential number of possible series of transforms by bounding the likelihood and detecting duplicates. A limitation of this method is that the

number of transformations required to detect a part is highly sensitive to contour noise.

2.5.3 Symmetric shape models

The third and final type of symmetry model is a high-level model of a symmetric shape. It is high-level in the sense that it abstracts the set of symmetric elements into a specific class, and can therefore narrow down the set of possible objects that it belongs to. Superpixels are useful in this approach, providing a reduced set of oversegmented regions to combine into hypothetical shapes. While all possible hypotheses should, in theory, be tested before selecting the most favourable ones, the practical challenge faced by this approach is the effective pruning of implausible hypotheses in order to keep the search space down to a manageable size.

An example method taking this approach is Sala & Dickinson, 2010 [98], which matches cycles of oversegmentation boundaries to parts from a pre-defined vocabulary of symmetric part shapes. In general, models are not limited to symmetric shape, and the category can be expanded to include similar methods. For example, Sclaroff & Liu, 2001 [100] learn a statistical deformable polygon model and apply it to combinations of over-segmented regions. Despite ruling out implausible combinations of regions by enforcing adjacency and smoothness across the boundary, the method still finds a difficult trade-off between speed and accuracy.

2.6 Conclusion

Methods in computer vision have progressed from clean, segmented inputs to natural images of scenes containing multiple cluttered objects. Likewise, object recognition can now handle important sources of variability such as part deformation. More recently, we have seen a shift from sliding window proposals to region proposals of arbitrary shape and size. The assumption of object-specific knowledge, however, has deflected attention away

from the role of object-independent knowledge, resulting in bottom-up grouping methods that need to return a large quantity of proposals to achieve satisfactory recall. One way to move forward is to develop methods to generate a smaller number of more reliable proposals. The power of higher-level cues suggests a path to pursue, however more research is needed on how to incorporate these cues without giving up computational efficiency.

Chapter 3

Shape-based learning and detection

As outlined above, we begin with a chapter on object categorization as a motivating context for our main thesis. Learning visual category models from training images is now standard practice in the object categorization community [33, 38]. Such systems typically rely on a strong degree of supervision, including cluttered scenes with labeled bounding boxes placed around objects of interest, or alternatively, scenes in which the labeled object of interest is largely front and centre (allowing the image boundary to serve as an effective bounding box). However, as the scope of the recognition task scales up to many thousands of objects, the burden of manually annotating the large number of required training images becomes prohibitive.

Captioned images are ubiquitous on the web and in certain image collections, and offer a powerful semi-supervised mechanism for learning category models without the need for labeled bounding boxes or image cropping. Unfortunately, any given image-caption pair may be unreliable, providing very weak or even erroneous training data. For example, the caption's nouns may refer to objects that don't appear in the image, while the more salient objects in the image might not even be referred to in the caption. However, across a large training set, recurring correspondences between particular objects appearing in the images and particular nouns appearing in the captions of those same images can be

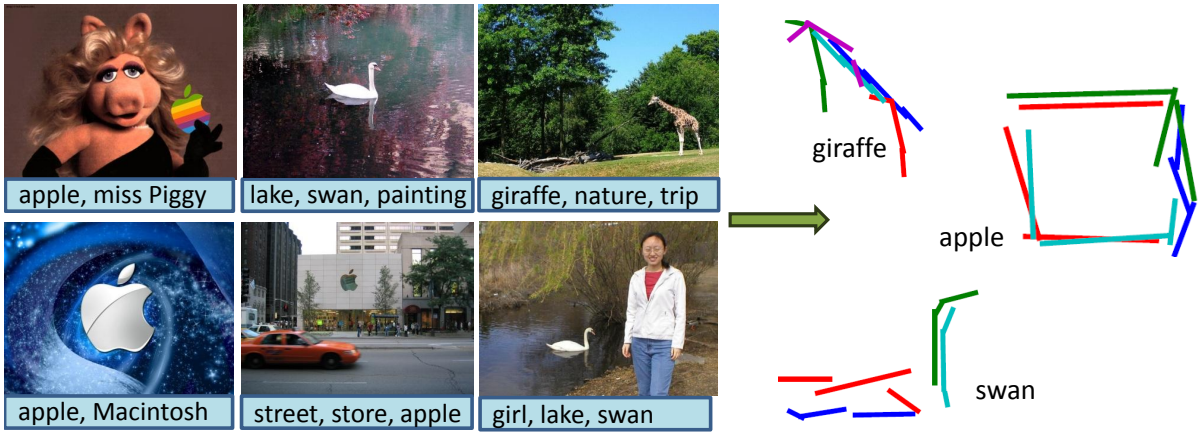


Figure 3.1: Given a set of captioned images of objects appearing in different positions and scales, we learn named contour models without bounding boxes.

assumed to be salient. Such correspondences can therefore be analyzed to yield visual category models as well as the names of those categories.

In [48], such a framework was proposed which learned structured visual object models from captioned training data. A learned visual model was captured in a graph, in which nodes represent SIFT features and edges represent spatial relations among the SIFT features, including relative position, orientation, and scale. When applied to captioned image collections, the choice of SIFT to characterize a node ultimately constrained the system to learn the structure and names of exemplars rather than categories. For example, the system learned the logos (and team names) of NHL teams from captioned action images (containing players whose jerseys contained the logos) taken from the NHL website, and learned the models and names of famous buildings and landmarks from around the world from captioned image landmark collections. The question we pose is whether such a framework can be extended to learn named categorical models based on shape rather than appearance.

Extending the framework of Jamieson *et al.* [48] poses a number of significant challenges, including 1) the choice of a suitable structured shape representation that can accommodate within-class deformation, articulation, and occlusion, and 2) coping with

the tremendous ambiguity of local shape features relative to appearance-based features such as SIFT. For a categorical shape representation, we adopt and extend the contour representation introduced by Ferrari *et al.* [37], replacing the star graph with a more generic graph with spatial relations between any pair of nodes, and adding multi-scale feature extraction. Like Jamieson *et al.*, we "grow" visual models that repeatedly co-occur with caption words across training images. However, for any pair of spatially related contour features representing an initial model, there may be many false positives across training images due to their inherent lack of specificity compared to appearance-based features. As a result, we introduce a powerful bottom-up heuristic that can focus search for recurring shape features that are likely to represent the boundaries of objects.

We evaluate our approach head-to-head with Jamieson *et al.*, and demonstrate that on a standard benchmark, it clearly outperforms Jamieson *et al.* in terms of learning visual categories in which shape is more invariant than appearance. Since we have based our shape representation on that of Ferrari *et al.*, we demonstrate that for the task of learning visual models with correct object labels but without the aid of bounding boxes, our approach outperforms Ferrari *et al.*'s approach, which depends heavily on the strong supervision offered by a bounding box. Finally, we demonstrate the robustness of our approach under image caption noise.

3.1 Related Work

There is a vast literature on language-vision integration, and the related problems of object category modelling, recognition and localization. While it is beyond the scope of this chapter to provide a full review of these topics, we will focus on the two subfields most related to our work: 1) using language or text to discover associations between visual and textual features, and 2) weakly- or semi-supervised object category learning using part-based models in support of image annotation.

Automatic image annotation systems attempt to discover correlations between words and visual features in a set of image-text pairs, *e.g.*, Barnard *et al.* [7] and Duygulu *et al.* [27]. Such systems typically model objects as a mixture of appearance-based features in which common configurations are not captured by explicit spatial relations but rather by co-occurrence statistics, *e.g.*, Carneiro *et al.* [14], Monay and Gatica-Perez [85], and Quattoni *et al.* [93]. The relatively high dimensionality of appearance-based features, *e.g.*, SIFT, means that image features are relatively unambiguous and therefore explicit relations are often unnecessary. The most similar work to ours is Jamieson *et al.* [48], whose framework used language to recover an explicit graph-based structural appearance model from captions training images. While the structural model captured explicit relations, its reliance on appearance-based local features rendered it far more suitable for learning named exemplars rather than named categories. We attempt to extend that framework to support the learning of visual object categories based on a structural shape representation, which is far more invariant to within-class variation than a structured appearance representation.

Learning a visual category model in the absence of image captions has received considerable attention from the recognition community. Most current approaches assume that bounding boxes around the objects are given [33, 92, 38]. In our domain, we want to avoid such strict supervision, and learn objects from cluttered scenes without any a priori information about location and scale. Moreover, labeling is assumed to be noisy in the sense that a noun in the caption may or may not refer to an object in its associated image, and an object in the image may or may not be referred to in the caption. Given coarse location and scale information, a number of frameworks have learned structured models in terms of parts and relations without requiring part labeling [22, 35, 6]. However, most of these approaches, like Jamieson *et al.* [48], rely on the distinctiveness of local appearance-based features, such as image patches or SIFT, because such features allow the search space to be aggressively pruned, thereby reducing the complexity of the

task. Furthermore, [22, 35, 6] constrain object models to a star structure, while in our work object features can be connected using a denser graph whose number of vertices and edge structure are inferred from images without supervision.

There are a few approaches which attempt to learn object models without the strong supervision provided by a bounding box. For example, Todorovic and Ahuja [112] propose a powerful framework for unsupervised modelling based on tree matching using a region-based object representation. Lee and Grauman [62] perform object discovery over images of multiple categories, using matching local appearance patches to anchor an initial set of edge fragments. In our approach, we do not rely on sparse discriminative features, and instead use only dense contour features which capture object categories better. Both Leordeanu *et al.* [63] and Payet and Todorovic [88] use only shape features to learn object models with weak supervision. In [63], object models consisting of hundreds of fully interconnected features require class labels for learning, while in [88], clusters of matching pairs of contour features and spatial relations are found in an unsupervised manner. While our visual representation also consists only of contours, we take an integrated approach where learning is guided by both bottom-up segmentation and image caption text to achieve a comparatively efficient way to initialize visual clusters among multiple categories. The key concept here is that in our approach, we focus on the construction of only those models that are referred to (i.e., named) in the captions, as opposed to mining a much larger space of possible regularities across a set of images, regardless of whether they are salient or not.

3.2 Overview

Given a set of captioned images, we learn object models that co-occur with words, and use the learned models to detect and annotate objects in uncaptioned images. In Section 3.3, we describe the object model, a graph in which vertices are local contour features

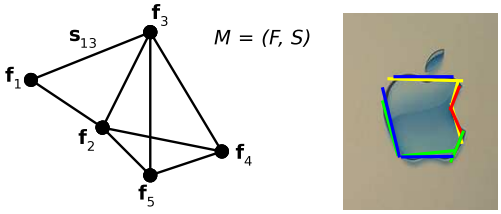
and edges encode pairwise spatial relations between features. Section 3.4 describes how object models are detected in an image by matching the model's contour features to those in the image. In a cluttered image, typically a large number of ambiguous features match individually to model features, creating an intractable search space. Efficient detection is achieved by using the model's spatial relations to prune out unlikely matches.

Section 3.5 describes how we learn object models that co-occur with words. In a graph-growing process, an initial model representing a small part of the object is iteratively grown, primarily along the object's boundary, to cover the object. Features found in the vicinity of existing model matches are added to the model if they recur with spatially consistent relations, thus making the model more object-specific and strengthening its co-occurrence with the given word. When no more consistently recurring features can be found in the vicinity, co-occurrence can no longer be improved and the final model is returned.

Model growth strongly depends on the initial model representing a small part of the object. Whereas in [48] a spatially related pair of appearance-based features was distinctive enough to represent a salient part, a related pair of contour features is relatively ambiguous, *i.e.*, a contour representation of an object part is often very similar to, and thus easily confused with, contours from background clutter or even other objects. To ensure that model growth begins from an object part, we use bottom-up segmentation as a powerful heuristic to focus search on contours that are likely to represent an object boundary.

3.3 Object model

An object model is denoted as $M = (F, S)$, where M is a graph with a vertex set of contour features $F = \{\mathbf{f}_1, \ldots, \mathbf{f}_T\}$ where T is unknown, and an edge set of pairwise relations $S \subseteq \{\mathbf{s}_{ij} : 0 \leq i < j \leq T\}$. Figure 3.2(a) shows an example of such a structure.



(a) Object model graph (b) Model instance

Figure 3.2: An object is modeled as a graph M over contour features F with pairwise spatial relations S.

A (undirected) spatial relation $\mathbf{s}_{ij}$ between features $\mathbf{f}_i$ and $\mathbf{f}_j$ may exist for any feature pair. To maintain a spatially coherent object description, we require that M be a connected graph. Typically, a dense set of relations is learned for objects having spatially consistent features and relations, resulting in a relatively distinctive model, while more deformable and articulating objects result in a sparser, more flexible model.

Feature extraction and description. Local contour features are extracted from an image using the method of Ferrari *et al.* [36]. Features are scale-invariant descriptions of line-segment abstractions of image contours, where edgel-chains are partitioned into line segments (*e.g.*, Figure 3.2(b)). Because linear partitioning is scale-dependent, we extract features at multiple scales to obtain a robust bottom-up description of an image. Efficient matching (Section 3.4) and learning (Section 3.5) is facilitated by a discrete vocabulary of codewords [36], whereby similar contour features are represented by the same codeword. By performing feature comparisons at the codeword level, many feature similarity computations are saved.

Spatial relations. A spatial relation $\mathbf{s}_{ij}$ encodes the distance u_{ij}, relative direction v_{ij}, and relative scale w_{ij} between features $\mathbf{f}_i$ and $\mathbf{f}_j$, *i.e.*, $\mathbf{s}_{ij} = (u_{ij}, v_{ij}, w_{ij})$. Letting $\mathbf{x}_i$ and s_i denote feature position and scale with respect to the image as in [36], respectively,

the three components are defined as follows:

$$u_{ij} = \frac{1}{\lambda}||\mathbf{x}_j - \mathbf{x}_i|| \qquad \text{(distance)}$$

$$v_{ij} = \arctan(\mathbf{x}_j - \mathbf{x}_i) \qquad \text{(relative direction)}$$

$$w_{ij} = \frac{1}{\lambda}(s_j - s_i) \qquad \text{(relative scale)}$$

where $\lambda(s_i, s_j)$ normalizes for the feature scales [48].

While features and spatial relations originate from the image, the object model is a prototypical description of the object using these elements, averaging over natural variations present in example images, *e.g.*, due to deformation or viewpoint variation. The following section explains how the model matches to image features under these variations.

3.4 Detecting objects

Occurrences of $M = (F, S)$ are detected by matching model features $F = \{\mathbf{f}_1, \ldots, \mathbf{f}_T\}$ to image features subject to spatial relations S. Due to occlusion and feature extraction errors, only a minimum number of model features are required to match. To detect occurrences in complex, cluttered images efficiently, features are matched sequentially, using S to prune out unlikely combinations at each stage.

Each occurrence is associated with a detection score D that measures the confidence of the match, and can be thresholded to obtain a level of precision. Suppose F' is a set of image features which is a potential match to a set of model features F. The detection score with respect to M is defined as a ratio of two quantities (similarly to [35]):

$$D = \frac{p(F'|M)}{p(F'|\text{bg})}. \tag{3.1}$$

The numerator is the probability that F' is a true instance of the object (approximated

by M), while the denominator is the probability that F' is not an instance of the object.

We can prune out unlikely matches by testing $D \geq \tau$, where $\tau = 1$ is a natural threshold. While the quantity $p(F'|\text{bg})$ determines the level of pruning, and is set independently of the object, $\tau > 1$ provides a tighter, object-specific threshold for D.

Let a partial matching be denoted using $\mathbf{c}$, a list of T binary indicators, where c_i is 1 exactly when the model feature $\mathbf{f}_i$ is matched. We use $F(\mathbf{c}) \subseteq F$ to indicate the subset of matching model features, and $S(\mathbf{c}) \subseteq S$ to indicate the subset of model relations between any pair of features in $F(\mathbf{c})$. Furthermore, given a set of matching image features F', let S' denote the set of pairwise relations among F'.

We let the object probability $p(F'|M)$ factorize into a feature and spatial component:

$$p(F'|M) = p(F'|F)p(S'|S). \tag{3.2}$$

Next, we proceed to define the two components. Let $\mathbf{f}'$ denote the image feature matching the model feature $\mathbf{f}$. The feature component $p(F'|F)$ is defined only over $\mathbf{f} \in F(\mathbf{c})$, as follows:

$$p(F'|F) = \prod_{\mathbf{f} \in F(\mathbf{c})} p(\mathbf{f}'|\mathbf{f}), \tag{3.3}$$

where $p(\mathbf{f}'|\mathbf{f})$ is a Gaussian density over the feature dissimilarity measure $d(\mathbf{f}', \mathbf{f})$ given in [36], with variance $\sigma_f^2 = 2.0$ and zero mean. The dissimilarity $d(\mathbf{f}', \mathbf{f})$ compares two contours using their line segment abstractions, in particular their internal relative positions, orientations, and lengths.

The spatial component $p(S'|S)$ considers only model relations between matched features, *i.e.*, $\mathbf{s}_{ij} \in S(\mathbf{c})$. Given a pair of matching features, let $\mathbf{s}'_{ij}$ denote the spatial relation

in the image corresponding to $\mathbf{s}_{ij}$. The spatial component is then defined as:

$$p(S'|S) = \prod_{\mathbf{s}_{ij} \in S(\mathbf{c})} p(\mathbf{s}'_{ij}|\mathbf{s}_{ij}), \tag{3.4}$$

where $p(\mathbf{s}'_{ij}|\mathbf{s}_{ij})$ factors into its distance, relative direction, and relative scale components:

$$p(\mathbf{s}'_{ij}|\mathbf{s}_{ij}) = p(u'_{ij}|u_{ij})p(v'_{ij}|v_{ij})p(w'_{ij}|w_{ij}), \tag{3.5}$$

each of which is a Gaussian density with means u_{ij}, v_{ij}, w_{ij} and fixed variances $\sigma_u^2 = 0.35, \sigma_v^2 = 0.3, \sigma_w^2 = 0.8$, respectively. The spatial component accounts for variations in spatial relations, *e.g.*, due to deformation or viewpoint.

The background probability $p(F'|\text{bg})$ represents a pruning threshold and is similarly composed only of the matching components as follows:

$$p(F'|\text{bg}) = \prod_{\mathbf{f} \in F(\mathbf{c})} p(\mathbf{f}'|\text{bg}_f) \prod_{\mathbf{s}_{ij} \in S(\mathbf{c})} p(\mathbf{s}'_{ij}|\text{bg}_s), \tag{3.6}$$

where $p(\mathbf{f}'|\text{bg}_f)$ and $p(\mathbf{s}'_{ij}|\text{bg}_s)$ are fixed values that in the product offset the ratio D.

Given the above definitions, the detection score is equivalent to:

$$D = \prod_{\mathbf{f} \in F(\mathbf{c})} \frac{p(\mathbf{f}'|\mathbf{f})}{p(\mathbf{f}'|\text{bg}_f)} \prod_{\mathbf{s}_{ij} \in S(\mathbf{c})} \frac{p(\mathbf{s}'_{ij}|\mathbf{s}_{ij})}{p(\mathbf{s}'_{ij}|\text{bg}_s)}. \tag{3.7}$$

Note that since features and spatial relations must individually pass the pruning threshold given by $p(\mathbf{f}'|\text{bg}_f)$ and $p(\mathbf{s}'_{ij}|\text{bg}_s)$, each factor in Equation 3.7 is greater than 1, *i.e.*, each matching component accumulates evidence by increasing the detection score. Since a partial match has fewer components, it is penalized with a lower score.

Detection algorithm. As in [48], matching is done efficiently by using S as a constraint to prune out unlikely feature combinations. A set of potentially matching features F' is iteratively grown until all model features are matched, or no more matching

features can be found. Failed partial matches are rejected and search resumes with a new initialization. Multiple occurrences in the same image are detected by repeating the search over remaining image features.

3.5 Learning objects

The graph-growing algorithm in [48] iteratively adds model features to cover the object (*e.g.*, in Figure 3.3), making the model increasingly object-specific and simultaneously increasing its co-occurrence with the word. Since word-object co-occurrence is our objective in finding salient object models in captioned images, we measure co-occurrence using the score $C_{M,W}$, defined as follows. Given N captioned training images, occurrences of words and objects are summarized in two vectors of length N:

$$\mathbf{w} = \{w_1, \ldots, w_N\}, w_n \in \{0, 1\}$$

indicating the occurrence of W in each image caption, and

$$\mathbf{m} = \{m_1, \ldots, m_N\}, m_n \in [0, 1]$$

indicating the occurrence of M in each image. While word occurrences are binary, object occurrences have soft scores weighted by their detection scores D. When there are multiple object occurrences in one image, the highest-scoring detection is considered.

While object and word occurrences are obviously correlated, objects may or may not be referred to in the caption, and words in the caption may or may not refer to objects in the image. The co-occurrence score $C_{M,W}$ has a probabilistic formulation in which the occurrences $\mathbf{m}, \mathbf{w}$ are generated from the presence ($c = 1$) or absence ($c = 0$) of a

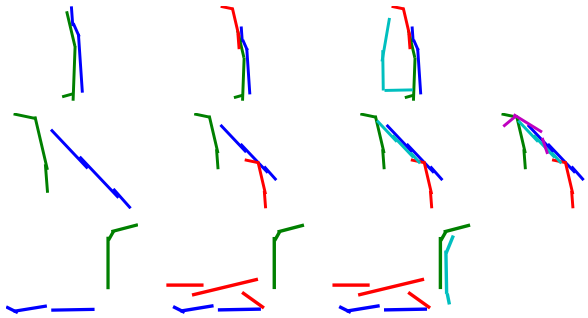


Figure 3.3: Initial models grow (left to right) to cover the object, increasing in distinctiveness and co-occurrence with words 'bottle', 'giraffe', and 'swan', shown (top to bottom).

common object, defined similarly to [48] as

$$C_{M,W} = \frac{p(\mathbf{m}, \mathbf{w}|c=1)}{p(\mathbf{m}, \mathbf{w}|c=0)}. \tag{3.8}$$

The numerator is defined as:

$$\begin{aligned} p(\mathbf{m}, \mathbf{w}|c=1) &= \prod_{n=1}^{N} p(m_n, w_n|c=1) \qquad (3.9) \\ &= \prod_{n=1}^{N} \sum_{o_n=0,1} p(m_n, w_n|o_n, c=1) p(o_n|c=1) \\ &= \prod_{n=1}^{N} \sum_{o_n=0,1} p(m_n|o_n, c=1) p(w_n|o_n, c=1) p(o_n|c=1) \end{aligned}$$

where o_n is a hidden variable indicating the presence ($o_n = 1$) or absence ($o_n = 0$) of the common object. The quantities $\alpha = p(w|o=1)$ and $\beta = p(w|o=0)$ are the probability that W occurs in the caption when the common object is present or absent, respectively. Similarly $\mu = p(m|o=1)$ and $\nu = p(m|o=0)$ represent the probability of the model M occurring in an image when the common object is present or absent. These parameters control the degree to which $C_{M,W}$ is sensitive to caption noise.

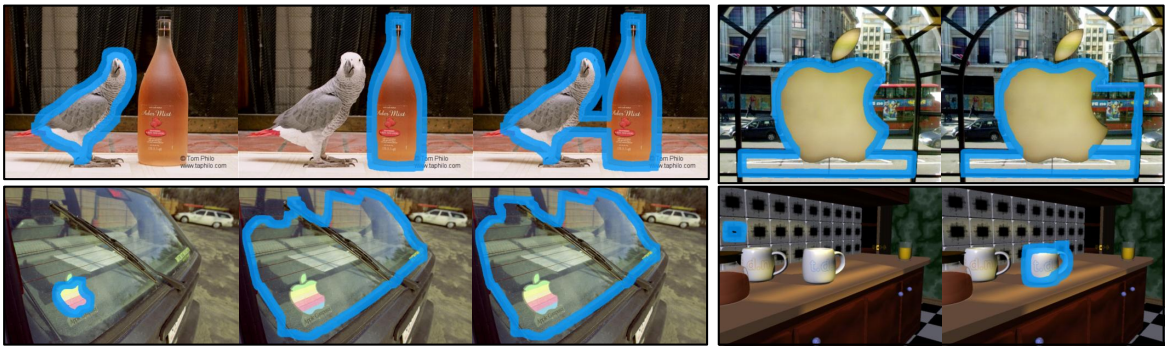


Figure 3.4: Superpixel Closure [66] returns multiple figure-ground segmentations per image (shown above for 4 images). Model initialization is constrained to contour features that fall along the boundaries of segmentation regions.

The denominator is defined as:

$$p(\mathbf{m}, \mathbf{w}|c = 0) = \prod_{n=1}^{N} p(m_n|c = 0)p(w_n|c = 0) \tag{3.10}$$

and acts as a bias term to offset the score $C_{M,W}$.

Graph-growing algorithm. Given an initial model $M^{(0)}$ of two spatially related features representing a small object part, a sequence of successively larger models $M^{(1)}, M^{(2)}, M^{(3)}, \ldots$ is found such that co-occurrence $C_{M^{(k)},W}$ is increasing for $k \geq 0$. More specifically, given occurrences of $M^{(k)}$, a search is performed for a feature in the vicinity that repeats in a consistent spatial relation with respect to the occurrences. Given a candidate shortlist of such features, the candidate with the highest $C_{M^{(k+1)},W}$ is chosen, where $M^{(k+1)}$ is the model with the added candidate feature, provided that $C_{M^{(k+1)},W} > C_{M^{(k)},W}$. When no such candidate exists, the current model is returned as the final model. While the approach described above is greedy, in practice we keep a list of the best few candidates at each iteration to explore in a backtracking fashion.

Model initialization. It is crucial that $M^{(0)}$ initially represents part of an object so the model can be expanded. The ambiguity of contour features, however, makes it difficult to distinguish salient boundary portions from accidental contours. Bottom-

up segmentation offers a powerful heuristic for focusing search over features likely to represent object boundaries. For each image containing the word W, we use Superpixel Closure [66] to extract multiple figure-ground segmentation hypotheses at multiple scales (Figure 3.4). The boundaries of a figure-ground segmentation hypothesis are used as constraints over features, where only contour features that fall within a small, fixed distance from the region boundary are selected. By initializing $M^{(0)}$ over this subset of features, the heuristic is used to guide the search for promising object parts that can be added to the model.

3.6 Evaluation

Following a discussion of the strengths and limitations of our method in Section 3.6.1, we present a head-to-head comparison of our approach to Jamieson *et al.* [48] in Section 3.6.2, and two experiments comparing our approach to Ferrari *et al.* [38], on which we have based our shape representation, in Section 3.6.3. Finally in Section 3.6.4, we train object models under image caption noise.

Experiments are conducted on the benchmark ETHZ dataset [36], which consists of 255 images of 5 diverse categories labeled with the words 'apple logo', 'bottle', 'giraffe', 'mug', and 'swan'. While bounding boxes are provided with the ETHZ dataset, they are used only for evaluating object localization and *not* for training, unless otherwise noted.

In our evaluation, we have used only the single final model M^* having the highest co-occurrence score $C_{M^*,W}$, in the same manner as in [48]. A possibility is to incorporate combinations of multiple learned models into detection for improved robustness (*e.g.*, models corresponding to different viewpoints), although we have not done this.

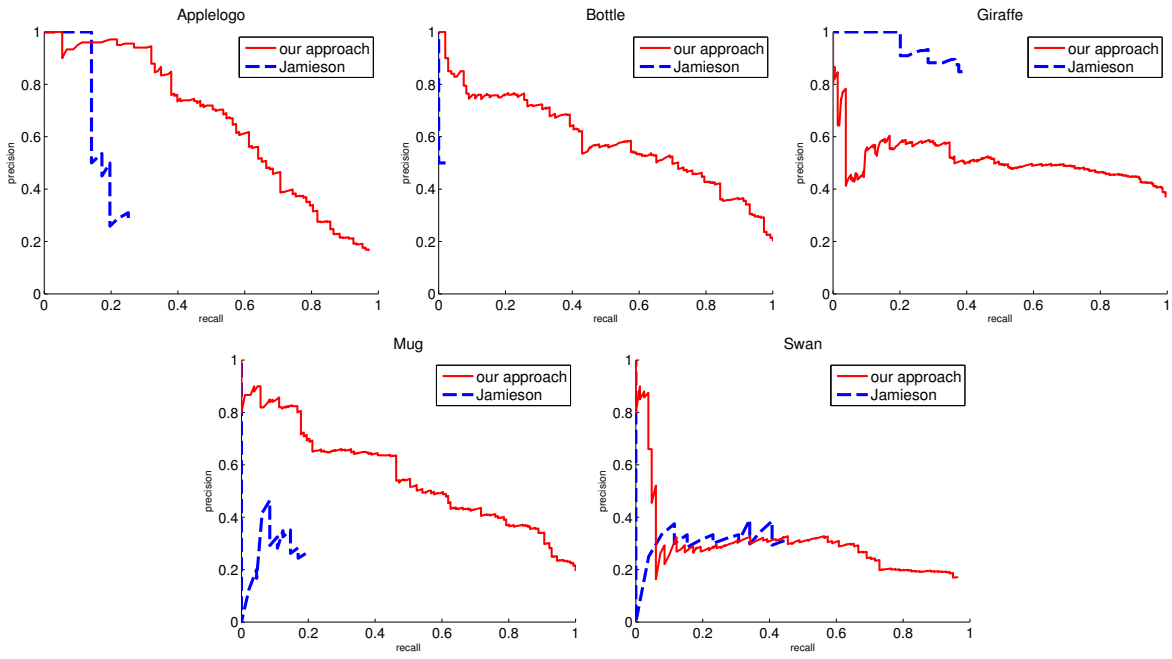


Figure 3.5: Comparison with Jamieson *et al.* Contour features capture object categories more effectively than appearance-based features.

3.6.1 Qualitative evaluation

In Figure 3.9 we present sample detections of learned object models for each ETHZ category. Our method is able to localize objects under large variations in scale and minor changes in viewpoint, orientation, deformation, and articulation. We achieve our best performance with 'apple logo', which features stable contours; our system missed only instances that were severely occluded or rotated by a large amount. 'Bottle' and 'mug' objects have tremendous variation in their surface markings: these objects are similar only in their shape, and our method correctly captures their characteristic contours. One source of false positives, however, is brand labels on bottles, which are easily confused with the bottle boundary due to their proximity. 'Giraffe' and 'swan' pose a significant challenge to our system due to their deformation and articulation. The line-based abstraction of Ferrari *et al.*'s contour features does not always respond in a stable manner to slight changes in curvature, as curve partitioning breakpoints suddenly appear or disappear. When our system encounters highly varying features during the graph-growing

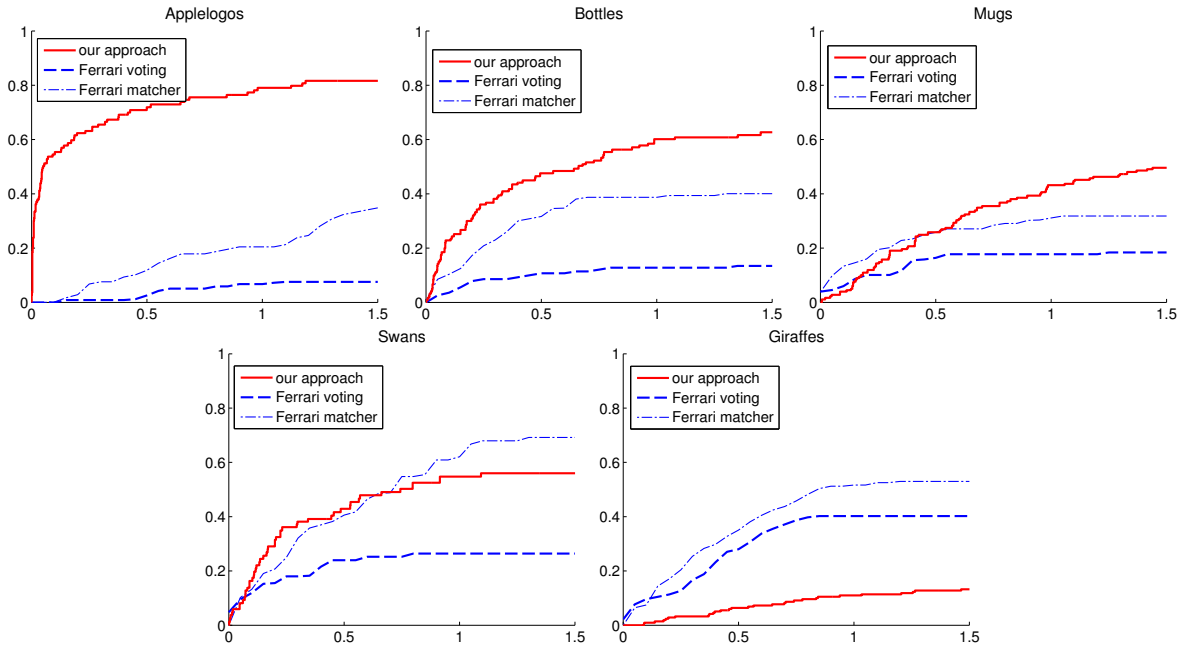


Figure 3.6: Comparison with Ferrari *et al.*, training without bounding boxes.

process, it tends to terminate growth early to avoid overfitting.

3.6.2 Shape *vs.* appearance models

A comparison with Jamieson *et al.* [48] allows us to study the effectiveness of shape-based *vs.* appearance-based features for learning named object categories. Our preliminary experiments support our hypothesis that shape is more effective. Figure 3.5 shows the performance of both approaches using precision-recall over correct image annotation, where an annotated image is counted as a true positive if the annotation is correct, *i.e.*, a detected occurrence is consistent with the true word label; and counted as a false positive if a detected occurrence is inconsistent with the true label.

Results show that the appearance-based system has difficulty finding recurring object models, especially for 'bottle' and 'mug'. These categories exhibit virtually no regularities in colour, texture, or surface markings. Instead, the appearance-based system found recurring texture on the body of the giraffe, and company slogan text appearing in a limited number of 'applelogo' images. Recurring patterns in the water were found for

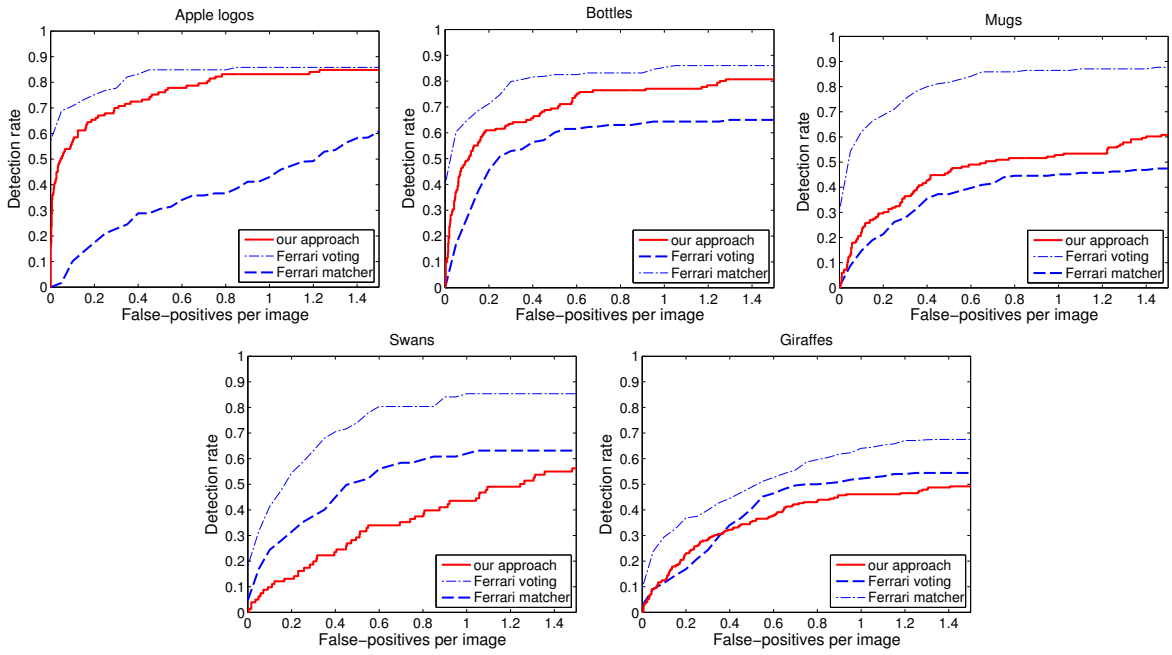


Figure 3.7: Comparison with Ferrari *et al.*, training with bounding boxes.

'swan' images, but these were similar to parallel strokes in leaves and grass, giving rise to poor precision. While appearance may have a limited ability to describe some object categories, contours were found to be more effective overall.

3.6.3 Training without bounding boxes

Our next experiments compare our approach with that of Ferrari *et al.* [38] under two settings described below. We follow the strict evaluation protocol in [38] (Pascal criterion of 50% intersection-over-union bounding box overlap), and report in Figure 3.6 detection rate (DR) against false positives per image (FPPI). For interest, we include performance for Ferrari *et al.*'s initial hough voting stage (lower curve) and the final verification stage (higher curve).

In the first setting we train under realistic conditions where manual annotations (bounding boxes) are not available to the system. Results in Figure 3.6 show that we generally outperform Ferrari *et al.*, which is unable to handle training images where objects do not appear in consistent positions and scales, namely, 'apple logo', 'bottle',

and 'mug'. It is clear that Ferrari *et al.*'s method strongly depends on the availability of manually annotated bounding boxes. Ferrari *et al.* performs better for 'giraffe' and 'swan', as the respective objects typically occupy most of the image, *i.e.*, the image boundary serves as an effective bounding box.

Our second setting examines the scenario where bounding boxes are provided for training. Although our system is not designed to use the information given by bounding boxes, it is interesting to include a discussion of the results. While our method works well for categories with relatively stable contour representations, *e.g.*, 'apple logo', the high variation in deformable and articulating categories present a significant challenge to our model-growing algorithm, and our performance is worse than that of Ferrari *et al.*'s (Figure 3.7). Since models grow incrementally in size, the algorithm is faced with the choice of adding weak, ambiguous contours, and hence may terminate growth early to avoid overfitting, resulting in the strict bounding box overlap criterion not being met. In contrast, Ferrari *et al.*'s Hough-based method has a more global view by having each model feature vote independently for the object centroid. However, this explicitly takes advantage of information from the bounding box, and thus performs an easier task than ours.

3.6.4 Image caption noise

In our final experiment, we approximate real-world captions by adding noise to the ETHZ word labels (referred to below as captions). Recall from Section 3.5 that the co-occurrence score assumes that an object is likely to be referenced in the caption with probability α, and an object has a chance of being referenced even when it is absent, with probability β. We corrupt the 5 original words in the ETHZ captions as follows. First, with probability α, words are substituted with a random word (not restricted to the 5 original words). For example, some 'apple logo' images no longer have the word 'apple logo' in the captions. Secondly, with probability β, an original word is appended to captions not originally

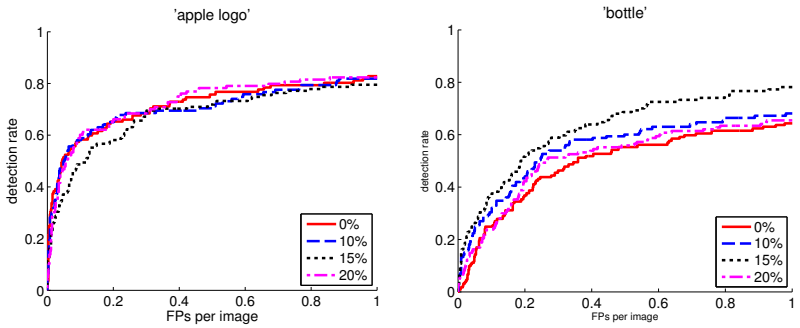


Figure 3.8: Localization performance of 'apple logo' and 'bottle' as caption noise is increased by corrupting ETHZ word labels. Results show stability under significant noise (up to 20%).

containing that word. For example, some non-'apple logo' images now have the word 'apple logo' in their captions. Distinct levels of noise are quantified with a percentage p, where $p\%$ reflects the value $\alpha = 1 - \frac{p}{100}$ and the value $\beta = \frac{p}{100}$. We run experiments at different noise levels, where captions are corrupted prior to training. Figure 3.8 reports localization performance as a function of noise levels, and shows that performance for 'apple logo' and 'bottle' remains stable under significant noise.

3.7 Conclusion

We have extended the framework of Jamieson *et al.* [48] to learn named categorical models based on shape, and added a focus-of-attention heuristic to cope with the ambiguity of contour features. Using a standard benchmark we have demonstrated that our method is able to handle large variations in scale and minor changes in viewpoint, deformation and articulation. A comparison with Jamieson *et al.* [48] showed that object categories are better captured by shape than appearance. Additionally we outperform methods such as Ferrari *et al.* [38] when training without bounding boxes, and showed that such methods have a strong dependence on manual annotations.

To conclude our chapter on object categorization as motivating context, we end with a few remarks on future work. We would like to extend the method in different ways

Figure 3.9: Sample contour detections of 'applelogo', 'bottle', 'giraffe', 'mug', and 'swan' (each in a unique color) in the ETHZ dataset.

to make it applicable to larger datasets. Ultimately, our goal is to learn object class models directly from the web. This entails having a richer representation (shape as well as appearance, more flexibility in the shape features), dealing with multiple classes in a scalable way as well as a more involved text model that could better deal with noisy image tags and surrounding web text.

Chapter 4

Symmetric part detection

Symmetry is a long-standing, interdisciplinary form that spans across the arts and sciences, covering fields as disparate as mathematics, biology, architecture, and music [70]. The roles played by symmetry are equally diverse, and can involve being an abstract object of analysis, a balancing structure in nature, or an attractor of visual attention. With object categorization in mind, the common thread in all of the above is that symmetry is ubiquitously present in both natural objects and artificial objects. It is no accident that we constantly encounter symmetry through our eyesight, and in fact, Gestalt psychologists [120] of the previous century proposed that symmetry is a physical regularity in our world that has been exploited by the human visual system to yield a powerful perceptual grouping mechanism. Experiments show evidence that we respond to symmetry before being consciously aware of it [115].

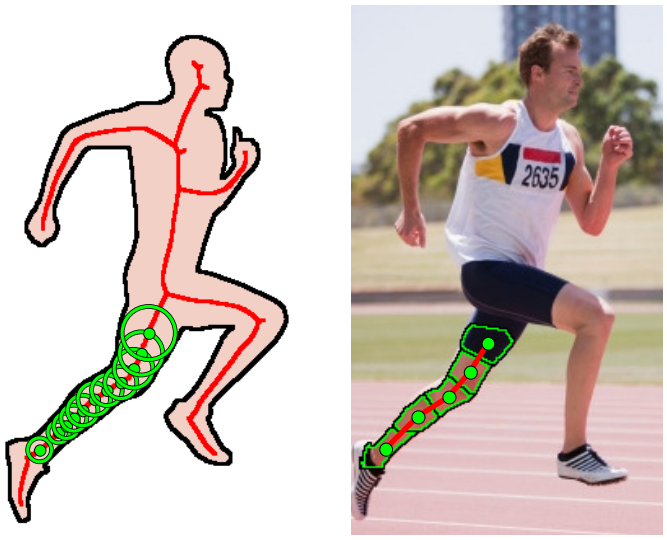
Inspired by a computational understanding of human vision, perceptual grouping played a prominent role in support of early object recognition systems, which typically took an input image and a set of shape models, and identified which of the models was visible in the image. Mid-level shape priors were crucial in grouping causally related shape features into discriminative shape indices that were used to prune the set down to a few promising candidates that might account for a query. Of these shape priors, one

of the most powerful is a configuration of parts, in which a set of related parts belonging to the same object is recovered without any prior knowledge of scene content.

The use of symmetry to recover generic parts from an image can be traced back to the earliest days of computer vision, and includes the medial axis transform (MAT) of Blum (1967) [10], generalized cylinders of Binford (1971) [9], superquadrics of Pentland (1986) [90], and geons of Biederman (1985) [8], to name just a few examples. Central to a large body of approaches based on *medial symmetry* is the MAT, which decomposes a closed 2D shape into a set of connected medial branches corresponding to a configuration of parts, providing a powerful parts-based decomposition of the shape suitable for shape matching, *e.g.* Siddiqi *et al.* (1999) [107] and Sebastian *et al.* (2004) [101]. For a definitive survey on medial symmetry, see Siddiqi *et al.* (2008) [106].

In more recent years, the field of computer vision has shifted in focus toward the object detection problem, in which the input image is searched for a specific target object. One reason for this lies in the development of machine learning algorithms that can leverage large amounts of training data to produce robust classification results. This led to rapid progress in the development of object detection systems, enabling them to handle increasing levels of background noise, occlusion, and variability in input images [42]. This development established the standard practice of working with input domains of real images of cluttered scenes, significantly increasing the applicability of object recognition systems to real problems.

A parallel advance in perceptual grouping, however, did not occur for a simple reason: With the target object already known, indexing is not required to select it, and perceptual grouping is not required to construct a discriminative shape index. As a result, perceptual grouping activity at major conferences has diminished along with the supporting role of symmetry [25, 26]. However, with the increasing popularity of evaluation datasets like VOC [31] and COCO [71], the community is starting to move from detecting objects of a single class to multi-class object detection. We expect shape-based perceptual grouping



(a) (b)

Figure 4.1: Our representation of symmetric parts: In (a), the shape of the runner's body is transformed into its medial axis (red), a skeleton-like structure that decomposes the shape into branch-like segments, *e.g.* the leg. The leg's shape is swept out by a sequence of discs (green) lying along the medial axis. In (b), the shape of the same leg is composed from superpixels that correspond to the sequence of discs. The scope of this chapter's framework is limited to detecting symmetric parts corresponding to individual branches.

to play a critical role in facilitating this transition.

In attempting to bring back medial symmetry in support of perceptual grouping, we observe that the subcommunity's efforts have not kept pace with mainstream object recognition. Specifically, medial symmetry approaches typically assume that the input image is a foreground object free from occluding and background objects and, accordingly, lack the ability to segment foreground from background, an ingredient crucial for tackling contemporary datasets. It is clear that the MAT cannot be reintroduced without combining it with an approach for figure-ground segmentation. This chapter is based on Lee *et al.* (2013) [58], which is the most current work along this trajectory, itself based on the earlier work of Levinshtein *et al.* (2009) [65, 68].

In the context of symmetric part detection, [58] introduced an approach that leveraged earlier work [65] to build a MAT-based superpixel grouping method. Since the proposed representation is central to our approach, we proceed with a brief overview of

the latter. A bottom-up method was introduced which first detected symmetric parts, then grouped them non-accidentally to form a discriminative shape index. By establishing a correspondence between superpixels and maximally inscribed discs, the method formulated a superpixel grouping problem that exploited symmetry as a grouping cue. The method thus recovered symmetric parts by grouping superpixels that represented discs of the same part. An evaluation was presented to show a significant improvement over other symmetry-based approaches.

Subsequently, [58] furthered the development of the above ideas on two complementary fronts. First, the medial representation was used to derive a sequence optimization problem for grouping, whose solution was shown to bring significant improvements in results. The approach uses a grouping algorithm that is principled and more effective than in [65]. Second, symmetry was captured more accurately by increasing the number of model parameters. While a limited number of parameters previously captured scale and orientation, the method's invariance was improved by additionally capturing bending and tapering. The resulting affinity function was also shown to support an improvement.

This chapter takes a high-level view of the work in reintroducing the MAT with figure-ground segmentation capability, enabling us to draw insights from a higher vantage point. We first develop the necessary background to trace the development from its origins in the MAT, through [65], and finally to [58]. In doing so, we establish a framework that makes clear the connections among previous work. For example, it follows from our exposition that [65] is an alternative instance of our framework. More generally, our unified framework benefits from the rich structure of the MAT while directly tackling the challenge of segmenting out background noise in a cluttered scene. Our model is discriminatively trained and stands out from typical perceptual grouping methods that use predefined grouping rules. Using experimental image data, we present both qualitative results and a quantitative metric evaluation to support the development of the components of our approach.

4.1 Related work

Symmetry is one of several important Gestalt cues that contribute to perceptual grouping. Symmetry plays neither an exclusive nor an isolated role in the presence of other cues. Contour closure, for example, is another mid-level cue whose role will increase as the community relies more on bottom-up segmentation in the absence of a strong object prior, *e.g.* [67]. Symmetry may also be effectively combined with other mid-level cues, *e.g.* [96, 59]. For brevity, we restrict our survey of related work to symmetry detection.

The MAT, along with its many descendant representations such as the shock graph [101, 107, 89, 24] and bone graph [78, 79], provides an elegant decomposition of an object's shape into symmetric parts; however, it made the unrealistic assumption that the shape was segmented, and is thus not directly suitable for today's image domains. For symmetry approaches in the cluttered image domain, we first consider the *filter-based* approach, which first attempts to detect local symmetries, in the form of parts, and then finds non-accidental groupings of the detected parts to form indexing structures. Example approaches in this domain include the multiscale peak paths of Crowley & Parker (1984) [23], the multiscale blobs of Shokoufandeh *et al.* (1999) [105], the ridge detectors of Mikolajczyk & Schmid (2002) [83], and the multiscale blobs and ridges of Lindeberg & Bretzner (2003) [73], and Shokoufandeh *et al.* (2006) [104]. Unfortunately, these filter-based approaches yield many false positive and false negative symmetric part detections, and the lack of explicit part boundary extraction makes part attachment detection unreliable.

A more powerful filter-based approach was recently proposed by Tsogkas & Kokkinos (2012) [114], in which integral images are applied to an edge map to efficiently compute discriminating features, including a novel spectral symmetry feature, at each pixel at each of multiple scales. Multiple instance learning is used to train a detector that combines these features to yield a probability map which, after non-maximum suppression, yields a set of medial points. The method is computationally intensive yet parallelizable, and

the medial points still need to be parsed and grouped into parts. But the method shows promise in recovering an approximation to a medial axis transform of an image.

The *contour-based* approach is a less holistic approach that addresses the combinatorial challenge of grouping extracted contours. Examples include Brady & Asada (1984) [12], Connell & Brady (1987) [21], Ponce (1990) [91], Cham & Cipolla (1995, 1996) [17, 16], Saint-Marc *et al.* (1993) [97], Liu *et al.* (1998) [74], Ylä-Jääski & Ade (1996) [124], Stahl & Wang (2008) [109], and Fidler *et al.* (2014) [39]. Since these methods are contour-based, they have to deal with the issue of computational complexity of contour grouping, particularly when cluttered scenes contain many extraneous edges. Some require smooth contours or initialization, while others were designed to detect symmetric objects and cannot detect and group the symmetric parts that make up an asymmetric object. A more recent line of methods extract interest point features, such as SIFT [76], and group them across an unknown symmetry axis [77, 57]. While these methods exploit distinctive pairwise correspondences among local features, they critically depend on reliable feature extraction.

A recent approach by Narayanan and Kimia [86] proposes an elegant framework for grouping medial fragments into meaningful groups. Rather than assuming a figure-ground segmentation, the approach computes a shock graph over the entire image of a cluttered scene, and then applies a sequence of medial transforms to the medial fragments, maintaining a large space of grouping hypotheses. While the method compares favorably to figure-ground segmentation and fragment generation approaches, the high computational complexity of the approach restricts it to images with no more than 20 contours.

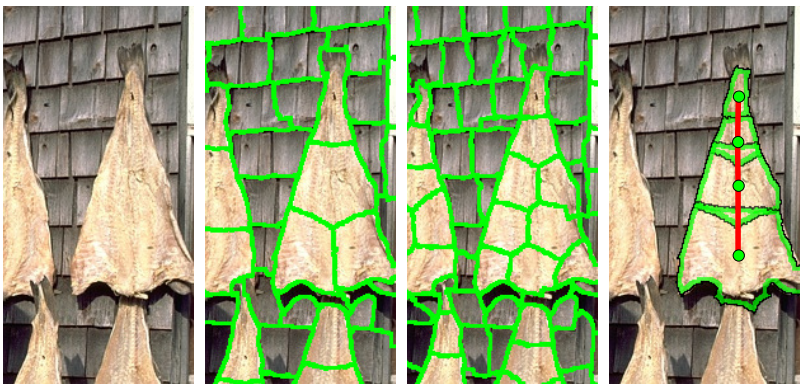
Our approach, represented in the literature by [58, 65, 68], is qualitatively different from both filter-based and contour-based approaches, offering a *region-based* approach which perceptually groups together compact regions (segmented at multiple scales using superpixels) representing deformable maximal discs into symmetric parts. We note that

while [65] has an additional step that groups symmetric parts into full objects, the scope of our framework is limited to detecting symmetric parts. In doing so, we avoid the low precision that often plagues the filter-based approaches, along with the high complexity that often plagues the contour-based approaches.

4.2 Representing symmetric parts

Our approach rests on the combination of medial symmetry and superpixel grouping [58, 65, 68], and in this section we formally connect the two ideas together. We proceed with the medial axis transform (MAT) [10] of an object's shape, as illustrated with the runner in Figure 4.1. The set of maximally inscribed discs plays the central role, whose centers (called *medial points*) trace out the skeleton-like *medial axis* of the object. We can identify the object's parts by decomposing the medial axis into its branch-like linear segments. We note that each object part is swept out by the sequence of maximally inscribed discs along the corresponding segment of the medial axis. For details on the relationship between the medial axis and the simpler reflective axis of symmetry, see Siddiqi *et al.* (2008) [106].

The link between discs and superpixels is established by recently developed approaches that oversegment an image into *superpixels* of compact and uniform scale. In order to view superpixels as discs, we note that just as superpixels are attracted to parts' boundaries, we imagine removing the circular constraint on discs and allowing them to deform to the boundary, resulting in "deformable discs". We will henceforth use the terms "superpixel" and "disc" interchangeably. The disc's shape deforms to the boundary provided that it remains compact (not too long and thin), resulting in a subregion that aligns well with the part's boundary on either side, when such a boundary exists. In contrast with the maximal disc, which is only bitangent to the boundary, as shown in Figure 4.1, the number of discs required to compose a part's shape is far less than the number required



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

Figure 4.2: To compose a part's shape from superpixels in a given input image (a), we compute superpixels at multiple scales (b-c), and combine superpixels from different scales (d).

using maximal discs.

In an input image domain of cluttered scenes, the vast majority of superpixels will *not* correspond to true discs of an object's parts, and thus it is suitable to treat superpixels as a set of *candidate* discs. Furthermore, a superpixel that is too fine or too coarse for a given symmetric part fails to relate its opposing boundaries together into a true disc, and a tapered part may be composed of discs of different sizes, as shown in Figure 4.2. Since we have no prior knowledge of a part's scale, and an input image may contain object parts of different scales, we compute superpixels at different scales, and take their union as a set of candidate discs.

Our goal is to perceptually group discs that belong to the same part. To facilitate grouping decisions, we will define a pairwise affinity function to capture non-accidental relations between discs. Since the vast majority of superpixels will not correspond to true discs, however, we must manage the complexity of the search space. By restricting affinities to *adjacent* discs, we exploit one of the most basic grouping cues, namely, *proximity*, which dictates that nearby discs are more likely to belong to the same medial part. We enlist the help of more sophisticated cues, however, to separate those pairs of discs that belong to the same part from those that do not. Viewing superpixels as discs allows

us to directly exploit the structure of medial symmetry to define the affinity. In Section 4.3, we motivate and define the affinity function from perceptual grouping principles to set up a weighted graph $\mathcal{G}$ of disc candidates. In Section 4.4, we discuss alternative graph-based algorithms for grouping discs into medial parts. Section 4.5 presents qualitative and quantitative experiments, while Section 5.7 draws some conclusions about the framework.

4.3 Disc affinity

Since bottom-up grouping is category-agnostic, a supporting disc affinity must accommodate variations across objects of all types. The affinity $A(d_i, d_j)$ between discs d_i and d_j must be robust against variability not only within object categories, but also variability between object categories. For a discriminatively trained affinity, it is helpful to extract features that reduce the variability for the classifier. In the following sections we define both shape and appearance features on the region scope defined by d_i and d_j.

4.3.1 Shape features

The local shape of discs is captured by a spatial histogram of gradient pixels, as illustrated in Figure 4.3. By encoding the distribution of the boundary edgels of the region defined by the union of the two discs, we capture mid-level shape while avoiding features specific to the given exemplar. This representation offers us a degree of robustness that is helpful for training the classifier, however it is not perfect—it remains sensitive to variations like scale and orientation, to name a few, and can thus allow the classifier to overfit to training examples.

We turn to medial symmetry to capture these unwanted variations, as the first step in making the feature invariant to such changes. Specifically, we locally model the shape by fitting the parameters of a symmetric shape to the region. We refer to a vector $\mathbf{w}$ of

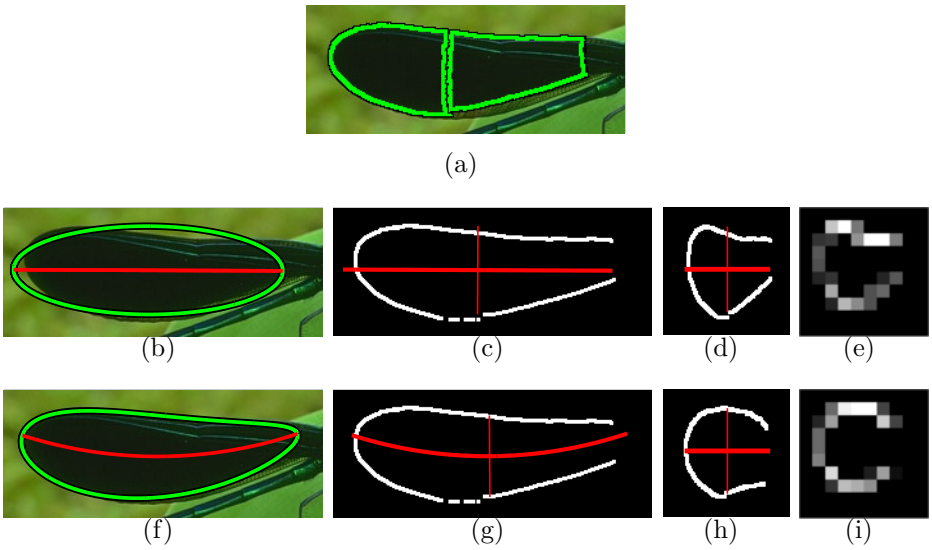


Figure 4.3: Improving invariance with a deformable ellipse: given two adjacent candidate discs, the first step is to fit the ellipse parameters to the region defined by their corresponding superpixels in (a). The top row shows invariance achieved with a standard ellipse. The ellipse's fit is visualized with the major axis in (b), the region's boundary edgels before (c) and after (d) warping out the unwanted variations, and the resulting spatial histogram of gradient pixels (e). See text for details. The bottom row shows the corresponding steps (f-i) obtained by the deformable ellipse. Comparing the results, a visually more symmetric feature is obtained by the deformable ellipse, which fits tightly around the region's boundary as compared with the standard ellipse.

warping parameters that subsequently define a warping function $W : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ that is used to remove the variations from the space, in effect normalizing the local coordinate system. Figure 4.3 visualizes the parameters $\mathbf{w}$ of a deformable ellipse fit to a local region, the medial axis before and after the local curvature was "warped out" from the coordinate system, and the spatial histogram computed on the normalized coordinate system.

Before describing the spatial histogram in detail, we discuss a class of ellipse-based models for modelling the local medial symmetry. Ellipses represent ideal shapes of an object's parts, and in particular are shapes that are symmetric about their major axes. A standard ellipse is parameterized by $\mathbf{w}_e = (\mathbf{p}, \theta, \mathbf{a})$, where $\mathbf{p}$ denotes its position, θ its orientation, and $\mathbf{a} = (a_x, a_y)$ the lengths of its major and minor axes. The parameter vector $\mathbf{w}_e$ is analytically fit to the local region and is used to define the corresponding

warping function $W_e(\mathbf{w}_e)$.

Historically, we first obtained the warping parameters with an ellipse [65]. While the advantages of using the ellipse lie in its simplicity and ease of fitting, shortcomings were identified in its tendency to provide too coarse a fit to the boundary to yield an accurate enough warping function. Accordingly in [58], we added deformation parameters to obtain a better overall fit across all examples. Despite a higher computational cost of fitting, the deformable model was shown to yield quantitative improvements.

Specifically, we obtain invariance to bending and tapering deformations by augmenting the ellipse parameters as follows: $\mathbf{w}_d = (\mathbf{p}, \theta, \mathbf{a}, b, t)$ with the bending radius b along the major axis and tapering slope t along the major axis. The parameter vector $\mathbf{w}_d$ is fit by initializing as a standard ellipse and iteratively fitting it to the local region's boundary with a non-linear least-squares algorithm. The fitted parameters are then used to define the warping function $W_d(\mathbf{w}_d)$ corresponding to the deformable ellipse.

Parameters $\mathbf{w}_d$ are fit to input points $\mathcal{X} = \{\mathbf{x}\}$ in the least-squares minimization problem

$$\arg\min_{\mathbf{w}} \sum_{\mathbf{x}\in\mathcal{X}} r(\mathbf{x};\mathbf{w})^2, \tag{4.1}$$

with error function

$$r(\mathbf{x};\mathbf{w}) = c \cdot \left(W(\mathbf{x};\mathbf{w}) \cdot W(\mathbf{x};\mathbf{w}) - 1\right), \tag{4.2}$$

where $c = \sqrt{a_x a_y}$ is a regularization term that penalizes highly elongated axes [108]. The least-squares estimate $\hat{\mathbf{w}}$ is obtained using the Matlab function `lsqcurvefit` with box constraints $0 < a_x, a_y \leq 1000, -2\pi \leq \theta \leq 2\pi, 0.5 \leq b \leq 0.5, 0.5 \leq t \leq 0.5$. The algorithm is initialized with a standard elliptical fit with zero deformation.

Only once the warping function $W(\mathbf{w})$ is fit to the local region $\mathcal{X}$ and applied to normalize the local coordinate system do we compute the spatial histogram feature. We

place a 10×10 grid on the warped region, and focusing on the model fit to the union of the two discs, we scale the grid to cover the area $[-1.5a_x, 1.5a_x] \times [-1.5a_y, 1.5a_y]$. Using the grid, we compute a 2D histogram on the normalized boundary coordinates weighted by the edge strength of each boundary pixel. Figure 4.3 illustrates the shape feature computed for the disc pair. We train a SVM classifier on this 100-dimensional feature using our manually labeled superpixel pairs, labeled as belonging to the same part or not. The margin from the classifier is fed into a logistic regressor in order to obtain the shape affinity $A_{shape}(d_i, d_j)$ in the range [0,1].

4.3.2 Appearance features

Aside from medial symmetry, we include appearance similarity as an additional grouping cue. While object parts may vary widely in color and texture, regions of similar appearance tend to belong to the same part. We extract an appearance feature on the discs d_i, d_j that encodes their dissimilarity in color and texture. Specifically, we compute the absolute difference in mean RGB color, absolute difference in mean HSV color, RGB and HSV color variances in both discs, and histogram distance in HSV space, yielding a 27-dimensional appearance feature. To improve classification, we compute quadratic kernel features, resulting in a 406-dimensional appearance feature. We train a logistic regressor with L1-regularization to prevent overfitting on a relatively small dataset, while emphasizing the weights of more important features. This yields an appearance affinity function between two discs $A_{app}(d_i, d_j)$. Training the appearance affinity is easier than training the shape affinity. For positive examples, we choose pairs of adjacent superpixels that are contained inside a figure in the figure-ground segmentation, whereas for negative examples, we choose pairs of adjacent superpixels that span figure-ground boundaries.

We combine the shape and appearance affinities using a logistic regressor to obtain the final pairwise affinity $A(d_i, d_j)$. Both the shape and the appearance affinities, as well as the final affinity $A(d_i, d_j)$, were trained with a regularization parameter of 0.5 on the

L1-norm of the logistic coefficients.

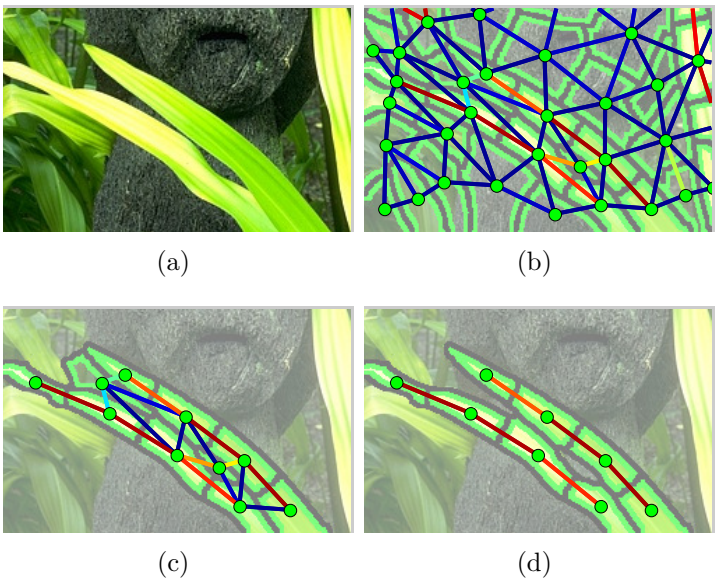
4.4 Grouping discs

Given a graph $\mathcal{G}$ of discs weighted by affinities, the final step is to group discs that belong to the same symmetric part. If two adjacent discs correspond to medial points belonging to the same medial axis, they can be combined to extend the symmetry. This is the basis for defining the pairwise affinities in $\mathcal{G}$, and it is how we exploit our medial representation of symmetric parts for grouping. Specifically, the affinity between two adjacent discs reflects the degree to which it is believed that they not only non-accidentally relate the two opposing boundaries together, but that they are centered along the same medial axis. In this section, we adapt and discuss two alternative graph-based algorithms, namely, the agglomerative clustering algorithm of Felzenszwalb & Huttenlocher (2004) [32], and the sequence-finding algorithm in the salient curve detection method of Felzenszwalb & McAllester (2006) [34].

4.4.1 Agglomerative clustering

Our first grouping approach is based on agglomerative clustering [32]. The algorithm takes as input the weighted graph $\mathcal{G}$ and merges edges in increasing order of weights. Each merge represents a grouping of discs, and the connected components that result correspond to symmetric parts. Grouping is performed efficiently in $O(e \log e)$ time, where e is the number of edges in $\mathcal{G}$. We refer the reader to [65] for details on the algorithm's adaptation to the setting of grouping discs.

The greedy approach, while fast, is unfortunately underconstrained in allowing merges to occur between branch-structured clusters, resulting in tree-like clusters as illustrated in Figure 4.4. These types of clusters can occur as frequently as spuriously high affinity values (false positives) occur, thus motivating the need to constrain the growth of clusters



(a) (b)

(c) (d)

Figure 4.4: In our approach, the (a) input image of foreground leaves is oversegmented into superpixels, and a (b) weighted graph $\mathcal{G}$ is built that captures the pairwise affinities that are computed among the superpixels. A graph-based grouping algorithm takes as input the graph $\mathcal{G}$, which may contain false positive affinities between the leaves, as shown in (b). In this figure, we illustrate the relative advantage of (d) sequence optimization over (c) agglomerative clustering. In (c), merging the vertices in $\mathcal{G}$ results in a cluster that undersegments the leaves, combining them into a single symmetric part that violates the assumption that a part is composed of a linear sequence of discs. In (d), the branching constraint is built into the sequence-finding algorithm which prevents symmetric parts from having tree-structured discs, and correctly segments the leaves into two distinct parts.

within medial branches.

4.4.2 Sequence optimization by dynamic programming

Our second approach is dynamic programming used in [58], which observes that each symmetric part is swept out by an *ordered sequence* of discs. Discs along the same medial axis are thus not only combined in pairs, but can be traced out linearly. This allows us to reformulate the problem of superpixel grouping as finding sequences of discs in a weighted graph $\mathcal{G}$ that belong to the same symmetric part. We thus obtain a grouping approach in which the desired branching constraint is inherent in the problem formulation. As illustrated in Figure 4.4, the algorithm applied to the same graph prevents the resulting

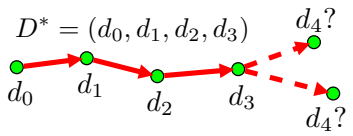


Figure 4.5: Grouping by dynamic programming: The iterative step of the algorithm grows sequences by extracting a sequence D^* from the priority queue, and returning longer sequences to the queue obtained by extending the end of D^* with adjacent discs. See text for details.

clusters from violating the branching constraint.

Before describing the steps of the dynamic programming algorithm, we note that it solves a discrete optimization problem, and thus represents a principled reformulation of our grouping problem. This includes defining an objective function that captures the goal of the problem, which is missing from the first approach, and making use of dynamic programming that efficiently solves for a global optimum. We specifically borrow from the optimization framework used for salient curve detection in Felzenszwalb & McAllester [34] and adapt it for symmetric part detection.

The application of [34] to our setting is best explained via their method for curve detection. The method takes as input a graph with weights defined on edges, and "transition weights" defined on pairs of adjacent edges. A salient curve is modeled as a valid sequence of edges, and a regularized cost function is defined on valid sequences that includes a normalized sum of the weights along the given sequence. Salient curves are found by globally optimizing the cost function using a dynamic programming algorithm.

In our setting, the graph $\mathcal{G}$ supplies weights between adjacent discs, and we define a valid sequence of discs (of variable length) by $D = (d_0, d_1, \ldots, d_n)$, which represents a symmetric part. The criteria that we want to optimize—good symmetry along the medial axis and a maximally long axis—is provided by the affinity graph $\mathcal{G}$. The regularized cost function, $\text{cost}(D)$, is defined correspondingly, and favors good internal affinity with a normalized sum over the affinities along the given sequence, and encourages longer sequences with a regularization term. Affinities defined over longer subsequences corresponding to

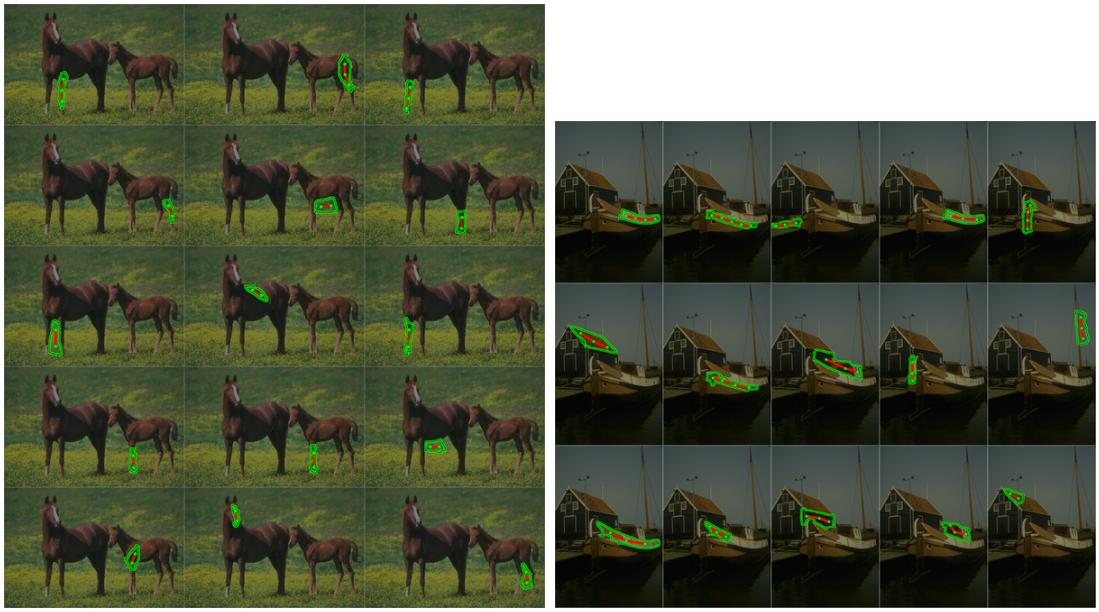
the transition weights have a smoothing effect on the preferred sequences.

We find optimal sequences $D_1, D_2, \ldots$ over $\mathcal{G}$ using the global cost defined as follows:

$$\text{cost}(D) = \frac{A}{n} + \frac{\sum_{i=1}^{n} w(d_{i-1}, d_i)}{n} + \frac{\sum_{i=1}^{n-1} w(d_{i-1}, d_i, d_{i+1})}{n-1}, \tag{4.3}$$

which consists of a sum over pairwise and triplewise weights, and a length bias A. The total cost is normalized by the length n of the sequence of edges. Compact superpixels are uniform in size and allows us to assume that a sequence consists of superpixels of approximately equal length. When the term A is strictly positive, the cost favors longer sequences and has the overall effect of encouraging sequences to grow during grouping (we have set $A = 0.1$). A sequence of discs corresponding to a symmetric part is ideally reflected by a low sum of weights. In our implementation, we have set pairwise weights to $1 - A(d_{i-1}, d_i)$, and experimented with setting the triplewise weights to a uniform value of zero, and to $1 - A(d_{i-1}, d_i, d_{i+1})$, in which the affinity is evaluated over the region defined by the union of three discs. In the latter case, the triplewise weights have a smoothing effect by evaluating symmetry over longer subsequences.

We now summarize the dynamic programming steps for globally minimizing cost(D). The core step is illustrated in Figure 4.5, and details can be found in [34]. The algorithm initializes a priority queue Q of candidate sequences with all possible sequences of unit length, then pursues a best-first search strategy of iteratively extending the cheapest candidate sequences. Each edge (d_{i-1}, d_i) is directed such that a sequence of edges terminating at d_i can be extended with an edge starting at d_i. At each iteration, as shown in Figure 4.5, the most promising sequence D^* is removed from Q, and new candidate sequences are proposed by extending the end of D^* with adjacent discs. If an extended sequence ending at an edge improves the cost of an existing sequence ending at the same edge, it is added back into Q. To find multiple sequences from the graph corresponding to different symmetric parts, we iteratively remove sequences that are already found and



(a) (b)

Figure 4.6: Multiple symmetric parts: for each image (a), (b) below we show the top 15 masks detected as symmetric parts. Each mask is detected as a sequence of discs, whose centers are plotted in green and connected by a sequence of red line segments that represents the medial axis.

re-minimize the cost, until a maximum cost is reached.

4.5 Results

We present an evaluation of our approach, first qualitatively in Section 4.5.1, then quantitatively in Section 4.5.2. Our qualitative results are drawn from sample input images and illustrate particular strengths and weaknesses of our approach. In our quantitative evaluation, we use performance metrics on two different datasets to gauge the contributions of different components in our approach. Figure 4.6 visualizes detected masks returned by our method, specifically showing the top 15 detected parts on sample input images. Parts are ranked by the optimization objective function. On each part's mask, we indicate the associated disc centers and the medial axis via connecting line segments. All results reported are generated with superpixels computed using normalized cuts [103],

at multiple scales corresponding to 25, 50, 100, and 200 superpixels per image.

Our evaluation employs two image datasets of cluttered scenes. The first dataset is a subset of 81 images from the Weizmann Horse Database (WHD) [11], in which each image contains one or more horses. Aside from color variation, the dataset exhibits variations in scale, position, articulation of horse joints. The second dataset was created by Lee *et al.* [58] from the Berkeley Segmentation Database (BSD) [82]. This set is denoted as BSD-Parts and contains 36 BSD images which are annotated with ground-truth masks corresponding to the symmetric parts of prominent objects (*e.g.*, duck, horse, deer, snake, boat, dome, amphitheater). This contains a variety of natural and artificial objects and offers a balancing counterpart to the horse dataset.

Both WHD and BSD-Parts are annotated with ground-truth masks corresponding to object parts in the image. The learning component of our approach requires ground-truth masks as input, for which we have held a subset of training images away from testing. Specifically, we trained our classifier on 20 WHD images and used for evaluation the remaining 61 WHD images and all 36 BSD-Parts images. This methodology supports a key point of our approach, which is that of *mid-level transfer*: by increasing feature invariance against image variability, we help prevent the classifier from overfitting to the objects on which it is trained. By training our model on horse images and applying it on other types of objects, we thus demonstrate the ability of our model to transfer symmetric part detection from one object class to another.

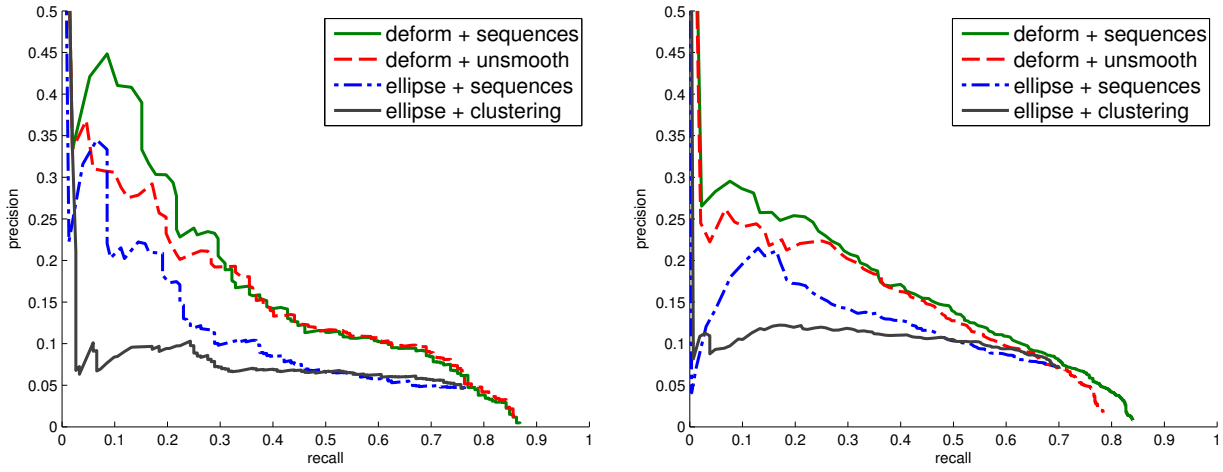
4.5.1 Qualitative results

Figure 4.7 presents our results on a sample of input images. For each image, the set of ground-truth masks are shown, followed by the top several detection masks. (Detection masks are indicated with the associated sequence of discs.) For clarity, individual detections are shown in separate images. The tiger image demonstrates successful detection of its parts, which vary in curvature and taper. In the next example, vertical segments

Figure 4.7: Example detections on a sample of images from BSD-Parts. Columns left to right: input image, ground-truth masks, top 4 detection masks. Note that many images have more ground-truth masks than detections that can be shown here.

of the Florentine dome are detected by the same method. The next example shows recovered parts of the boat. When suitably pruned, a configuration of parts hypothesized from a cluttered image can provide an index into a bank of part-based shape models.

In the image of the fly, noise along the abdomen was captured by the affinity function at finer superpixel scales, resulting in multiple overlapping oversegmentations. The leaf was not detected, however, due to its symmetry being occluded. In the first snake image, low contrast along its tail yielded imperfect superpixels that could not support correct segmentation, however the invariance to bending is impressive. The second snake



(a) BSD-Parts (b) WHD

Figure 4.8: Performance curves for corresponding to different settings of the components of our approach on (a) BSD-Parts and (b) WHD. See text for details.

is accompanied with a second thin detection along its shadow. We conclude with the starfish whose complex texture was not a difficult challenge for our method. We have demonstrated that symmetry is a powerful shape regularity that is ubiquitous across different objects.

4.5.2 Quantitative results

In the quantitative part of our evaluation, we use standard dataset metrics to evaluate the components of our approach. Specifically, we demonstrate the improvement contributed by formulating part grouping as sequence optimization, and by using invariant features to train the classifier. Results are computed on the subset of WHD held out from training, and on BSD-Parts. To evaluate the quality of our detected symmetric parts, we compare them in the form of detection masks to the ground-truth masks using the standard intersection-over-union metric (IoU). A detection mask m_{det} is counted as a hit if its overlap with the ground-truth mask m_{gt} is greater than 0.4, where overlap is measured by IoU $|m_{det} \cap m_{gt}|/|m_{det} \cup m_{gt}|$. We obtain a precision-recall curve by varying the threshold over the cost (weight) of detected parts.

Figure 4.8 presents the performance curves corresponding to 4 different settings under our framework, evaluated on both WHD and BSD-Parts: 1) *ellipse+clustering* combines the ellipse-warped affinity with agglomerative clustering and corresponds to [65]. We note that low precision is partly due to the lack of annotations on many background objects in both datasets; 2) *ellipse+sequences* combines the ellipse-warped affinity with sequence optimization; 3) *deform+sequences* combines deformable warping with sequence optimization, and corresponds to [58]; and 4) *deform+unsmooth* sets the triplewise weights in cost(D) uniformly to zero rather than using the affinity as done in the previous setting. A corresponding drop in performance shows that smoothness is an important feature of symmetric parts. In summary, experimental results confirm that both the added deformations and sequence optimization are individually effective at improving the accuracy of our approach.

4.6 Conclusion

Symmetry figured prominently in early object recognition systems, but the potential of this powerful cue is largely overlooked in contemporary computer vision. In this chapter, we have reviewed a framework that attempts to reintroduce medial symmetry into the current research landscape. The key concept behind the framework is remodelling the discs of the MAT as compact superpixels, learning a pairwise affinity function between discs with a symmetry-invariant transform, and formulating a discrete optimization problem to find the best sequences of discs. We have summarized quantitative results that encourage further exploration of using symmetry for object recognition.

We have reviewed ways in which we overcame the early limitations of our approach, such as using additional deformation parameters to improve warping accuracy, and reformulating grouping as a discrete optimization problem to improve results. There are also current limitations to be addressed in future work. To briefly mention two, we first note

that the success of using Gestalt grouping cues such as symmetry depends on effectively combining multiple cues together. To improve the robustness of our system, we are thus exploring how to incorporate additional mid-level cues such as contour closure. This will help our system more accurately resolve cases where different features provide conflicting cues, and thus improve the overall performance. Secondly, our scope is bottom-up detection and thus is agnostic of object categories. However, in a detection or verification task, top-down cues may be available. We are thus investigating ways of integrating top-down cues into our framework.

In conclusion, we have reviewed an approach for reintroducing the MAT back into contemporary computer vision, by leveraging the formulation of maximal discs as compact superpixels to derive symmetry-based affinity function and grouping algorithms. Quantitative results encourage further development of the framework to recover medial-based parts from cluttered scenes. Finally, as initial explored in [68], detected parts must be non-accidentally grouped before they yield the distinctiveness required for object recognition. Having looked in-depth at the grouping cue of symmetry, we now move on to combining symmetry with other mid-level cues.

Chapter 5

Grouping with multiple mid-level cues

Bottom-up grouping has re-emerged in the form of class-independent *region proposals* [15, 116] which are increasingly combined with object detectors and have been shown to improve performance on competitive challenges [40]. Region proposal methods typically start with a generation stage that uses a bottom-up grouping algorithm to output a diverse set of proposals, which are then passed to a ranking stage where they are evaluated by a trained scoring function. The ranked proposals have richer structure than sliding windows, which are typically fixed in aspect ratio, and have higher precision than sliding windows, whose proposals number in the millions. In contrast, region proposal methods achieve state-of-the-art results with only thousands of proposals.

Region proposal methods forward bottom-up ambiguity from the generation stage to the ranking stage in the form of proposals, at which point stronger cues are available to reduce the ambiguity. Unlike hierarchy-based models [39], proposals are often explicitly isolated from object class labels. Typical methods like [15, 116], however, rely on only low-level appearance and contour cues to generate proposals, and as a result must diversify their proposals in large quantities to preserve recall. In this chapter, we

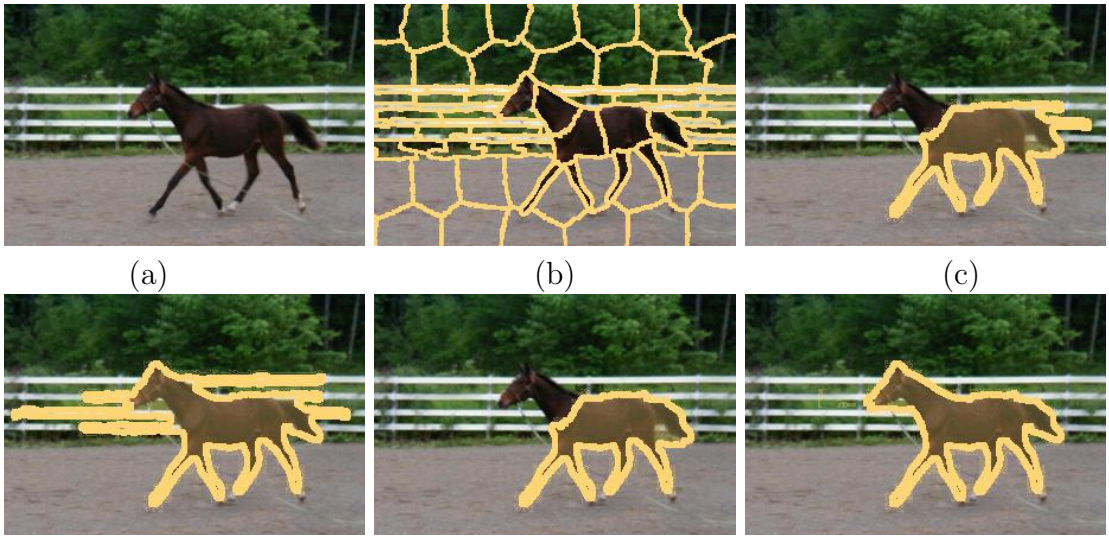


Figure 5.1: Given an input image as shown in (a), our method first oversegments into superpixels in (b), which are to be grouped into regions based on a combination of perceptual grouping cues. In this example, both the horse and the fence are relatively homogeneous in color and exhibit contrasting boundaries, however the horse's neck is slightly darker than its torso. As shown in (c), low-level appearance alone oversegments the horse at the neck where a large gap in contour is attempted. When including contour closure in (d), the boundary correctly encloses the head, but elsewhere strays along the fence. Conversely in (e), including symmetry without closure separates the fence from the horse, but fails to enclose the head. With closure and symmetry together in (f), the entire horse is correctly segmented.

present a complementary approach to diversification that uses mid-level grouping cues to resolve ambiguity at an early stage to avoid the need to generate proposals in excessive quantities.

By approaching the problem as figure-ground separation, we draw on a large body of work in perceptual grouping. Mid-level cues capture non-accidental relations between image elements that are exhibited by all objects. They are less specific than a high-level object model, yet more discriminative than low-level cues like appearance similarity and contour continuity. Here we highlight two mid-level cues of interest:

Closure [28, 47] is a regularity that favors regions that are enclosed by strong contour evidence along the boundary. Bottom-up approaches to finding closure vary in the types of cues used, and may include continuity and convexity. The problem is often cast as

finding a cycle of graph edges in a very large space, and is exacerbated when allowing for gaps in the closure (an illustrative example being the Kanizsa triangle).

Symmetry [77, 84, 114], as discussed in Chapter 4, is a ubiquitous and powerful regularity with scope that spans entire objects or their parts. Since the early days, perceptual grouping research has produced such varied representations as the medial axis transform [10], generalized cylinders [9], superquadrics [90], and geons [8]. Later approaches applied symmetry toward cluttered and occluded image domains, which present the challenge of searching for symmetrically related elements in an intractably large space.

Like other bottom-up cues, closure and symmetry govern the perception of figure and ground. Our method, as illustrated in Figure 5.1, groups regions by leveraging mid-level and low-level cues in combination. An input image (a) is oversegmented into superpixels (b) to be grouped together into regions. The example shown contains a horse as foreground, for which multiple grouping cues will help to separate from the background. Relying on a limited number of cues, as subsequently shown, may result in a segmentation that is overly sensitive to detailed changes in the image. In (c), low-level appearance alone oversegments the horse at the slightly darker neck, while jumping a large gap in contour. Including contour closure in (d) attracts the boundary to pixels with strong contour evidence and encloses the head, but elsewhere strays along the fence. Symmetry is a regularity that groups objects, such as the fence, into its coherent parts, but as shown in (e), does not group the head with the horse. In (f), closure and symmetry combine their strengths to correctly segment the horse.

Mid-level cues extend beyond any particular object, and symmetry and closure, in particular, are ubiquitous over most objects. Since our model is aware of objects only at the mid-level and unaware of their specific appearance, the model can easily transfer from one object to another. In this chapter, as a case in point, we learn our model on the Weizmann Horse Dataset (WHD) [11], and then apply it to diverse non-horse objects from the Weizmann Segmentation Dataset (WSD) [3]. Quantitative experiments

are performed on WHD to 1) establish the usefulness of each cue by demonstrating improvement as they are incrementally added, and to 2) demonstrate improvement on two leading region proposal methods with a limited budget of proposals. The contributions of our chapter are summarized as follows:

1. **Perceptual search.** We focus on the front-end stage where the generation of region proposals is driven by bottom-up grouping. We argue that stronger mid-level cues play an important role in reducing the number of proposals.

2. **Mid-level cue combination.** We improve upon previous approaches that lack mid-level knowledge or combine only one mid-level cue with low-level cues, by leveraging the combination of mid-level closure and mid-level symmetry to group regions together.

3. **Trained cue combination:** While perceptual grouping methods often make ad hoc grouping decisions, we capture all cues in a single energy function and jointly learn their weighted combination.

5.1 Related work

Viewing region proposals as object hypotheses for recognition, we begin by broadening our scope to include methods designed for sliding window detectors. Among these, the *objectness* detector of Alexe *et al.* [2] computes low-level features on superpixels [32] to score sampled image boxes. *Selective search* of Uijlings *et al.* [116] outputs boxes that bound regions generated from agglomerative clustering of superpixels [32]. The method accumulates a pool of regions over each step of region-merging until all regions are merged together, and ensures diversity by pooling results over multiple color and texture feature spaces. The method is very fast, yet is based on low-level appearance alone.

Arbelaez *et al.* [5] produces regions by merging superpixels of [4] over multiple scales. The method considers a limited number of all pairs, triples, and quadruples of adjacent

superpixels. Our approach is different in that we operate on a single layer of compact superpixels, and define a set of low-level and mid-level cues that quantify the likelihood of grouping.

The *shape sharing* method of Kim & Grauman [51] matches part-level regions in a given image to a bank of exemplars, which project object-level information back into the image to help with segmentation. The *category-independent proposals* of Endres & Hoiem [29] develops a CRF model to label superpixels based on segment seeds. The resulting region proposals are ranked using structured learning on grouping cues. The energy potentials are pairwise and submodular, and inference is done by graph cuts. While we use a similar procedure to generate regions, we combine mid-level cues at the front-end without seeding from a fixed hierarchical segmentation.

The *CPMC* method of Carreira & Sminchisescu [15] generates regions directly from the image rather than deriving them from a fixed segmentation. The method solves multiple parametric min-cut instances over color seeds. Regions are re-ranked by regressing on overlap with region-scoped features, including mid-level features such as convexity and eccentricity. The emphasis is on ranking rather than the front-end grouping, which samples color seed models over millions of pixels. Our approach is qualitatively different from the above methods as we focus on bottom-up grouping, however our mid-level front-end is complementary to the ranking stage.

Viewing region proposals as figure-ground labeling calls on a large literature covering low-level and mid-level Gestalt cues. Rather than covering methods on individual mid-level cues like symmetry [77, 84, 114] and closure [28, 47], we consider holistic approaches that combine low- and/or mid-level cues. The *region competition* approach of Zhu & Yuille [126] combines the objectives of snakes and region growing into a single Bayes criterion, effectively integrating the relative strengths of contour-based and region-based cues. An algorithm for optimizing the new criterion was introduced, however only guaranteed convergence to a local minimum. Our approach differs in using superpixels

which, providing access to both contours and regions, serves as a convenient basis for combining their respective cues, independently from the optimization approach.

Cue combination is alternately formulated as a linear combination of terms that make up a cost or scoring function. Graph-based image partitioning [32, 103] requires an affinity function to be specified between pairs of pixels and therefore falls under this category. For example, the *intervening contour* method of Leung & Malik [64] includes a contour-based term into the appearance-based affinity and solves the normalized cut problem. Like [64], we combine cues in a linear combination of terms, but differ in the overall grouping approach and use different cues on superpixels.

Inspired by random field models, the *cue integration* method of Ren *et al.* [96] develops an energy function that integrates appearance similarity, contour continuity, contour closure, and object familiarity on triangular tokens. The model was trained and solved using loopy belief propagation. Like [96], we combine multiple grouping cues over adjacent regions, but we take the approach of expressing the energy potentials in a form that allows efficient and exact solutions.

Our approach is most similar to Levinshtein *et al.* [67], which elegantly formulated contour closure as finding minimum energy labelings, and used parametric min-cut to find globally optimal solutions. A gap cost was trained on superpixel boundary features and incorporated into a gap-to-area ratio cost. We differ from [67] by combining multiple cues, among which contour closure counts as only one, and furthermore we learn to combine cues in a random field energy model.

5.2 Approach overview

We develop an energy function over superpixel labelings that captures a combination of low-level and mid-level grouping cues. In Section 5.3, we motivate the cues of low-level appearance, mid-level closure, and mid-level symmetry from perceptual grouping

principles and define their corresponding energy potentials. We use a mathematical form that is flexible enough to accommodate additional cues, yet conforms to a structure that can be exploited to obtain efficient and exact solutions. In Section 5.4, we introduce a scaling term in the energy that represents ambiguity in scale, and use it to obtain multiple solutions. Section 5.5 formulates the loss-based framework with which we train the weights of the energy function. We present and discuss results in Section 5.6 and conclude in Section 5.7.

5.3 Grouping cues

Our method operates on superpixels as grouping primitives from which regions are composed. Superpixels provide a rich topology of regions and boundaries on which a diverse set of cues can be defined to capture different grouping relations. Specifically, an input image x is oversegmented into P superpixels, where each superpixel p is assigned a binary label $y_p \in \{1 = \text{figure}, 0 = \text{ground}\}$. The labeling space $\mathcal{Y} = \{1, 0\}^P$ contains all possible vectors $\mathbf{y} = \{y_1, \ldots, y_P\}$ of superpixel labels and thus represents all possible groupings. An energy function $E(x, \mathbf{y})$ is defined on $\mathcal{Y}$ that favors labelings based on a combination of cues observed on the image x, and captures this combination as a decomposition into potentials corresponding to different cues:

$$E(x, \mathbf{y}) = \sum_{cue} \sum_{I \in \mathcal{N}^{cue}} E_I^{cue}(x_I, \mathbf{y}_I) \tag{5.1}$$

In (5.1), *cue* varies over low-level appearance (*app*), mid-level closure (*clo*), and mid-level symmetry (*sym*). The set $\mathcal{N}^{cue}$ of neighborhoods for a particular cue defines the local subsets of superpixels on which the cue is defined. Potentials in our model are restricted to pairwise order. By finding a labeling that globally minimizes the energy, we obtain a region that exhibits strong grouping relations. In this section, we discuss the contributions of the cues of symmetry, closure, and appearance and define their

corresponding energy potentials.

5.3.1 Appearance similarity

Similarity is a basic perceptual grouping cue that we capture in the form of color and texture similarity. We note that even objects of heterogeneous appearance are often composed of homogeneous parts. For each superpixel p, we compute a d-dimensional normalized histogram descriptor $\mathbf{h}^p$ that summarizes its appearance. We then compute the similarity between a pair p, q of adjacent superpixels using the histogram intersection kernel:

$$s^{p,q} = \sum_{i=1}^{d} \min(h_i^p, h_i^q).$$

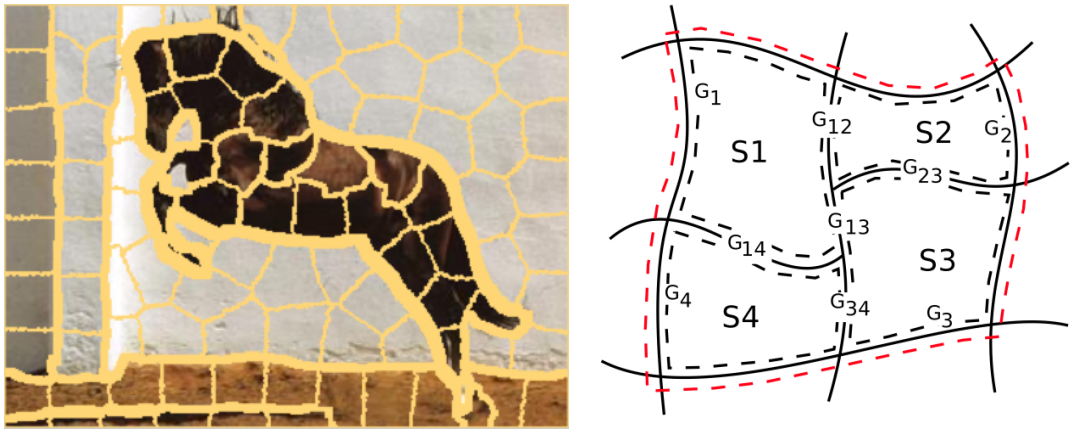
Color and texture are captured with different histograms $\mathbf{h}_c, \mathbf{h}_t$ which are computed in the manner of Uijlings *et al.* [116] using multiple color channels and SIFT-like features. Similarity is computed for both histograms to obtain the two-dimensional feature:

$$\phi_{p,q}^{app}(x) = (s_c^{p,q}, s_t^{p,q}).$$

The pairwise appearance potential for each adjacent pair p, q combines the cues and is defined as follows:

$$E_{p,q}^{app}(x, \mathbf{y}_{p,q}) = \begin{cases} \mathbf{w}_{app}^{\mathsf{T}} \phi_{p,q}^{app}(x) & y_p \neq y_q \\ 0 & y_p = y_q. \end{cases} \tag{5.2}$$

We note that this discourages adjacent superpixels of similar appearance from splitting, but allows superpixels of dissimilar appearance to merge.



(a) (b)

Figure 5.2: To support the cue of mid-level closure, contour evidence is computed along superpixel boundaries as shown in (a), where thickness indicates the degree of contour evidence (lack of gap) [67]. In (b), the gap cost $G(\mathbf{y})$ for a hypothetical labeling $\mathbf{y}$ over the corresponding region's boundary $\partial(\mathbf{y})$ is shown in dashed red and consists of superpixels S1-S4. Unary potentials sum gap along the corresponding boundaries G1-G4, and pairwise potentials sum gap along the shared boundaries G12-G34. The total gap $G(\mathbf{y})$ along the dashed red is obtained by subtracting twice the pairwise potentials from the unary potentials. (We thank the authors of [67] for permission to reproduce figure (b)).

5.3.2 Contour closure

Contour closure is a key challenge of perceptual grouping. One of the key ingredients of closure is strong contour evidence along the boundary that separates figure from ground. Since we prefer boundaries that avoid large gaps of contour (weak evidence), we define for any given labeling $\mathbf{y}$ the gap cost $G(\mathbf{y})$ in terms of the corresponding region's boundary $\partial(\mathbf{y})$:

$$G(x, \mathbf{y}) = \sum_{x \in \partial(\mathbf{y})} g(b).$$

This cost accumulates a positive gap $g(b)$ over all boundary pixels $x \in \partial(\mathbf{y})$. We compute $g(b) \in [0, 1]$ at every boundary pixel using the trained measure of [67], which accounts for discrepancy between actual image boundaries and superpixel boundaries in location and orientation.

We directly incorporate $G(\mathbf{y})$ into our energy function by expressing it in terms of

unary and pairwise potentials over $\mathbf{y}$. We encode the potentials as in [67], for which a schematic example is provided in Figure 5.2. Unary potentials are defined to sum gap along the corresponding superpixel's boundary $\partial(p)$ when $y_p = 1$. Pairwise potentials between p and q sum gap only along the boundary $\bar{\partial}(p,q)$ shared by *both* superpixels, when $y_p = y_q = 1$:

$$E_p^{clo}(y_p) = \begin{cases} \sum_{b \in \partial(p)} g(b) & y_p = 1 \\ 0 & y_p = 0 \end{cases} \quad E_{p,q}^{clo}(\mathbf{y}_{p,q}) = \begin{cases} \sum_{b \in \bar{\partial}(p,q)} g(b) & y_p = y_q = 1 \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

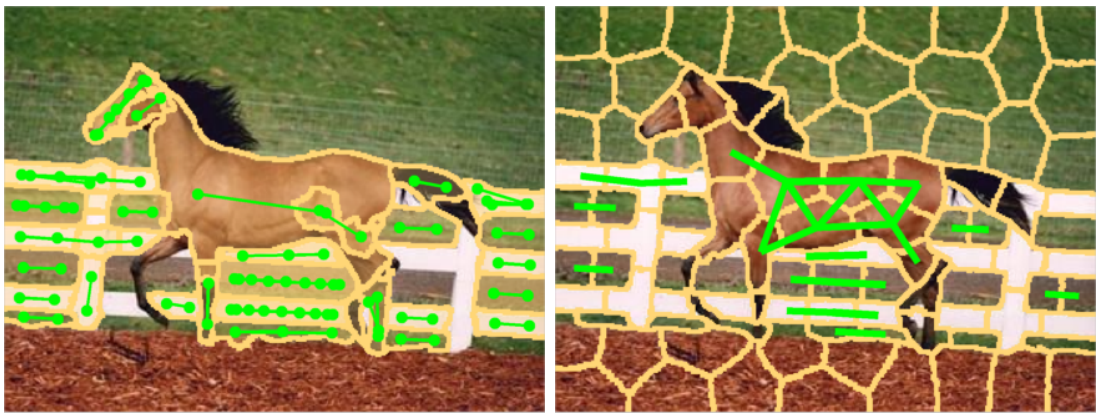
As illustrated in Figure 5.2(b), unary potentials sum gap along their superpixel boundaries. For a region consisting of a single superpixel, the unary potential reflects the correct gap cost. However, for a region consisting of multiple superpixels, simply summing the corresponding unary potentials will double count the gaps along the boundaries shared by adjacent superpixels in the region, which are exactly those counted by the pairwise potentials. The gap $G(\mathbf{y})$ along the true boundary of the region can thus be easily expressed as the sum of the unary potentials, minus twice the pairwise potentials:

$$E^{clo}(x, \mathbf{y}) = w_{clo} \cdot \left(\sum_p E_p^{clo}(x, y_p) - 2 \sum_{p,q} E_{p,q}^{clo}(x, \mathbf{y}_{p,q}) \right) \tag{5.4}$$

5.3.3 Symmetry

Symmetry relates together local features that span the entire object or its parts and, as such, is a powerful mid-level cue. Its large spatial scope, however, makes the associated grouping problem combinatorially hard. In the context of our representation in the labeling space $\mathcal{Y}$, the region corresponding to an object or its part can be composed from any number of superpixels, and thus induces dependencies of arbitrarily high order.

Our method draws on the approach of Lee *et al.* [58] for finding symmetrically related features, which circumvents the above difficulty by leveraging the scope of large superpix-



(a) (b)

Figure 5.3: Symmetric parts detected by [58] as sequences of medial points represented as region masks, as shown in (a). In (b), straight lines indicate strong pairwise affinities between superpixels that belong to the same symmetric part.

els. By operating on successively coarser superpixels, pairwise combinations are able to cover successively larger regions, effectively achieving higher orders of dependency. This allows local sections of symmetry of the same object part to be composed from a *sequence* of pairwise superpixels at the correct scale. Furthermore, [58] finds optimal sequences of superpixels that lie along the symmetry axes of object parts, as shown in Figure 5.3.

We incorporate the symmetry cue in the above form into our method by favoring the grouping of superpixels that belong, with high likelihood, to the same symmetric part. In practice, we run the sequence optimization of [58] independently on multiscale superpixels to obtain a set S of symmetric parts, as shown in Figure 5.3(a), and define pairwise potentials that favor grouping of superpixels that belong to the same symmetric part, as shown in 5.3(b).

For each pair of adjacent superpixels p, q, we define the feature:

$$\phi^{sym}_{p,q}(x) = \max_{s \in S(p,q)} \text{score}(s),$$

which is the score of the best scoring symmetric part $s \in S(p,q)$, where $S(p,q) \subseteq S$ is the subset of symmetric parts for which the overlap with p and q both exceed $\tau = 0.75$. When $S(p,q)$ is empty, the feature takes on a value of zero. The value $\text{score}(s) \in [0,1]$ is

the part's detection score, which we interpret as positive grouping evidence. We perform non-maximum suppression over all superpixels pairs so that each pairwise relation is influenced by at most one symmetric part. The symmetry potential penalizes labeled regions that cut through symmetric parts, and is defined for each pair (p, q) of adjacent superpixels as:

$$E^{sym}_{p,q}(x, \mathbf{y}_{p,q}) = \begin{cases} w_{sym} \cdot \phi^{sym}_{p,q}(x) & y_p \neq y_q \\ 0 & y_p = y_q. \end{cases} \tag{5.5}$$

5.4 Figure-ground labeling

We incorporate the potentials corresponding to the grouping cues into our final energy function as follows:

$$E(\mathbf{y}) = \sum_{p,q} E^{app}_{p,q}(\mathbf{y}) + \sum_{p} E^{clo}_{p}(\mathbf{y}) - 2\sum_{p,q} E^{clo}_{p,q}(\mathbf{y}) + \sum_{p,q} E^{sym}_{p,q}(\mathbf{y}) + \lambda \sum_{p} \phi_p(\mathbf{y}). \tag{5.6}$$

In (5.6), the grouping cues are rescaled by a scaling potential $\phi_p(\mathbf{y})$ by a factor of $\lambda > 0$ that is defined as follows:

$$\phi_p(\mathbf{y}) = \begin{cases} -\text{area}(p) & y_p = 1 \\ 0 & y_p = 0. \end{cases} \tag{5.7}$$

The scaling potential removes trivial solutions associated with the empty grouping with zero energy. Furthermore, as λ increases, the scaling potential favors labelings of larger area, and thus λ adjusts the energy's preference for regions of smaller or larger scale.

To minimize (5.6), we rewrite it as a sum of unary and pairwise potentials:

$$E(x, \mathbf{y}) = \sum_{p} \mathbf{w}_1^{\mathsf{T}} \phi^{\lambda}_p(\mathbf{y}, x) + \sum_{p,q} \mathbf{w}_2^{\mathsf{T}} \phi_{p,q}(\mathbf{y}, x), \tag{5.8}$$

noting that the pairwise potentials are submodular when weights are non-negative (features are non-negative). When λ is fixed, (5.8) can be minimized efficiently with a maxflow algorithm. In our model, λ is an unknown variable that represents the scale of an object, and so we minimize (5.8) for all values $\lambda \in \Lambda$, for $\Lambda \subset \mathbb{R}$. This is known as the parametric maxflow problem [53], which yields a finite number of solutions as λ varies over Λ. The set of globally optimal solutions can be found with a linear number of calls to the maxflow algorithm. We use $\Lambda = [0, 1]$ to yield a dozen solutions on average per image, thereby obtaining multiple proposals varying in scale.

5.5 Learning

We train the weights of the energy function (5.8) by incorporating it into the Structured SVM framework. Learning will be thoroughly addressed in Chapter 6, however for this chapter, we will restrict learning to standard S-SVM for tractability. The framework is instantiated with the loss function:

$$\Delta(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{\text{area}(x)} \sum_p \text{area}(p) \cdot \phi_p^{\Delta}(\hat{y}_p, y_p) \qquad \phi_p^{\Delta}(\hat{y}, y) = \begin{cases} 1 - \alpha_p & \hat{y} = 1 \\ \alpha_p & \hat{y} = 0 \end{cases} \tag{5.9}$$

where $\alpha_p \in [0, 1]$ is the fraction of pixels inside superpixel p labeled by the ground truth pixel mask. Weights are optimized using StructSVMCP [99] and constrained to be non-negative. We note that the learning step assumes that the loss for a particular example is obtained by minimizing the corresponding energy with a particular value of λ. For simplicity, we have fixed $\lambda = 0.01$ for all training examples. During testing, however, we vary λ over Λ for each example.

5.6 Evaluation

A key point of our approach is that our model being mid-level enables it to directly transfer from one object class to another. To illustrate this point, we use the Weizmann Horse Dataset (WHD) [11] to build our model, while applying it on diverse non-horse objects from the Weizmann Segmentation Dataset (WSD) [3]. Section 5.6.2 describes the qualitative results obtained on WSD. We additionally perform quantitative experiments to study the individual contributions of our grouping cues, and to demonstrate an improvement over two leading region proposal methods. Results are presented in Sections 5.6.1 and 5.6.3, respectively.

Contained in WHD are 328 images, each annotated with a ground truth mask. We train on the first 200 images, and hold out the remainder for test. As an evaluation metric, we compute the average best overlap [116]:

$$\mathcal{O}(\mathcal{G}, \mathcal{R}; k) = \frac{1}{|\mathcal{G}|} \sum_{(g,i)\in\mathcal{G}} \max_{r\in\mathcal{R}(i;k)} o(r; g),$$

where $\mathcal{G}$ and $\mathcal{R}$ are the ground truth and region proposal masks, respectively, and the quantity k is the number of top-ranked proposals. Intersection-over-union overlap between a region r and the ground truth mask g is denoted by $o(r; g)$. We plot overlap against k to measure the trade-off between overlap and k.

5.6.1 Cue combination

We study the effect of incrementally combining the cues of appearance, closure and symmetry, by including their respective potentials in the energy function (5.6). Each cue observes a different type of grouping evidence, and we expect the best results from combining the strengths of all cues. Figure 5.4 shows the effect of incrementally adding closure and symmetry to appearance, as well as using mid-level cues without appearance. We observe that closure and appearance work well together, while symmetry helps

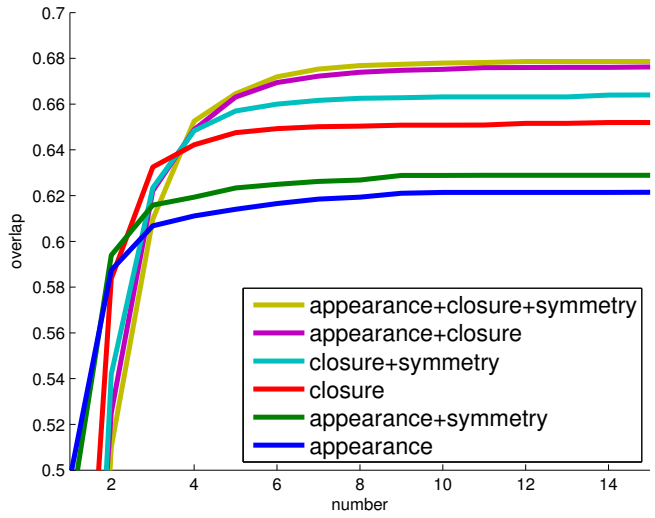


Figure 5.4: Improvement in recall as grouping cues are incrementally added to the energy.

for all combinations. The results confirm our hypothesis that each cue individually contributes useful information, with the best result from combining all cues. Our symmetry cue contributed a smaller than expected improvement on WHD. We expect symmetry's contribution to be better reflected in more challenging datasets of objects whose regions cannot be as easily computed with the remaining cues alone.

5.6.2 Qualitative results

We present qualitative results for a diverse set of objects in Figure 5.6, where each row shows the top proposed regions produced from a given input image, along with the corresponding ground truth mask. Our method successfully separates horses from cluttered and occluded backgrounds. We observe that alternative regions often arise when there are spurious contours, particularly within the horse and shadows under the horse. False negative contours, however, can cause undersegmentation, *e.g.*, in row 4. Symmetry of occluded fences is often sufficient to prevent undersegmentation. We note that while our appearance cue favors grouping regions of similar color, it does not penalize regions of heterogeneous color and correctly segments the horse in row 5. The remaining

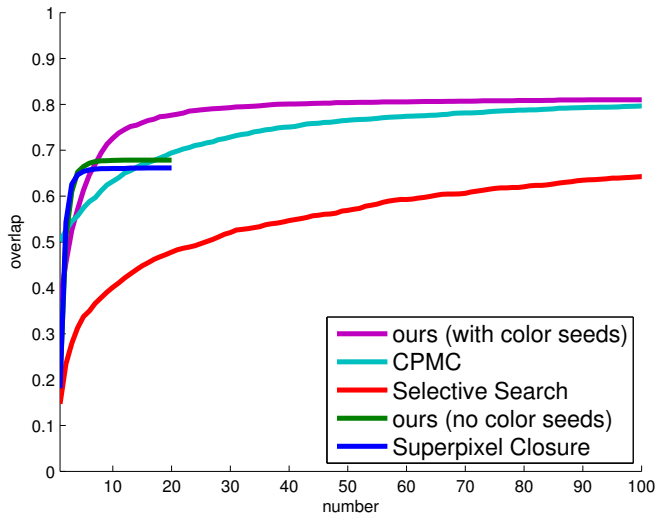


Figure 5.5: Improvements over CPMC [15] and Selective Search [116] with a limited budget of proposals, and improvement over Superpixel Closure [67]. Our method is evaluated with and without color seeds. See text for details.

rows show results on different objects from WSD and demonstrate that our method is class-independent, and that our mid-level cues trained on WHD transfer well to objects of different classes.

5.6.3 Comparison with region proposals

We demonstrate the advantage of our mid-level method with respect to Selective Search [116] and CPMC [15] in Figure 5.5. For comparison with [116], we have measured overlap with respect to the agglomerated regions (rather than their bounding boxes), pooled over color types, similarity measures, and the parameter of [32]. The quantitative comparison demonstrates an improvement on [116] with a budget of a hundred proposals. We note that our method focuses on resolving ambiguity and generates 20-30 proposals per image. In contrast, [116] relies on diversity of proposals and requires over 100 proposals to achieve the same recall. Our method is thus more effective for a limited budget of proposals.

For comparison with [15], we have measured overlap with their regions produced using color seeds, where a color seed model is fit to sampled locations. For this comparison,

we have also added color seeds to our energy function (5.6). Specifically, for any given test image, we fit a Gaussian mixture model to the image's RGB distribution to obtain a compact set of color seed models corresponding to each mixture component (we obtain 4-6 clusters per image). This differs from [15] which densely samples color seeds over a grid. For each pair of color seeds as a foreground-background hypothesis, we bias our energy function (5.6) with a unary potential that scores the corresponding superpixel's log likelihood ratio between the foreground seed and the background seed, as done in [15]. Parametric min-cut is solved for each pair of color seeds, and the resulting regions are pooled with the original (unbiased) regions, obtaining several hundred proposals per image. Our method with color seeds improves on [15] with a budget of a hundred proposals.

5.7 Conclusion

Bottom-up grouping is regaining momentum as a counterpart to object detection, and is a promising area in which to explore the importance of mid-level grouping cues. Mid-level cues are ubiquitous and transcend individual object classes, yet can be leveraged effectively only in combination. We have presented a method to combine appearance, closure, and symmetry, and demonstrated the usefulness of each cue. We have also demonstrated the effectiveness of using mid-level cues to resolve ambiguity with a limited budget of proposals, and have shown that our model complements diversification techniques when a large number of proposals is affordable. Our next chapter will address learning in depth by adapting S-SVM to the context of region proposals.

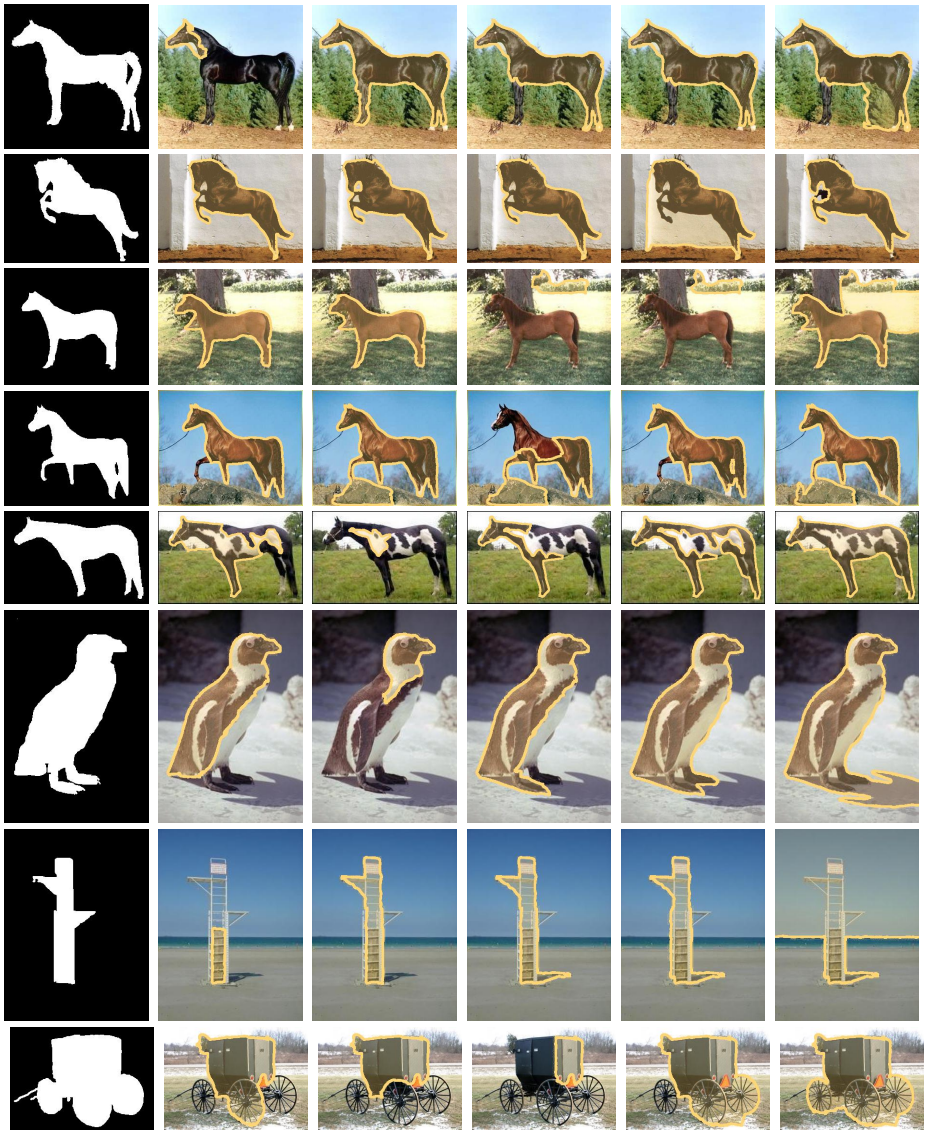


Figure 5.6: Top region proposals from our method from different images ranked by increasing energy. Leftmost column shows corresponding ground truth masks and remaining columns show region proposals. Rows 1-5 correspond to images from the Weizmann Horse Database (WHD), and rows 6-8 correspond to images from the Weizmann Segmentation Database (WSD). See text for details.

Chapter 6

Learning to generate grouping hypotheses

The return to bottom-up region segmentation is an invitation to integrate the many mid-level cues that ultimately play a role in such perceptual grouping, including proximity, symmetry, closure, similarity, and continuity, to name a few. But even with computational models of such cues, how should they be combined and what is their relative importance? As shown in Figure 6.1, we explore these issues within the framework of graphical models, which encode contextual relationships such as grouping cues between adjacent image regions. Computationally, graphical models can be solved exactly in a tractable manner, *e.g.* by minimizing a pairwise submodular energy function of binary variables with a maxflow algorithm. They can be discriminatively trained to predict structured outputs, offering a consistent learning and inference framework for bottom-up segmentation [96, 111].

Until recently, discriminative graphical models for segmentation have been restricted to single-output predictions, and lacked a framework for learning to predict diverse multiple outputs, *e.g.* as introduced in Multiple Choice Learning (MCL) [45]. Multiple-output models are especially important for region proposals due to the principle of least commit-

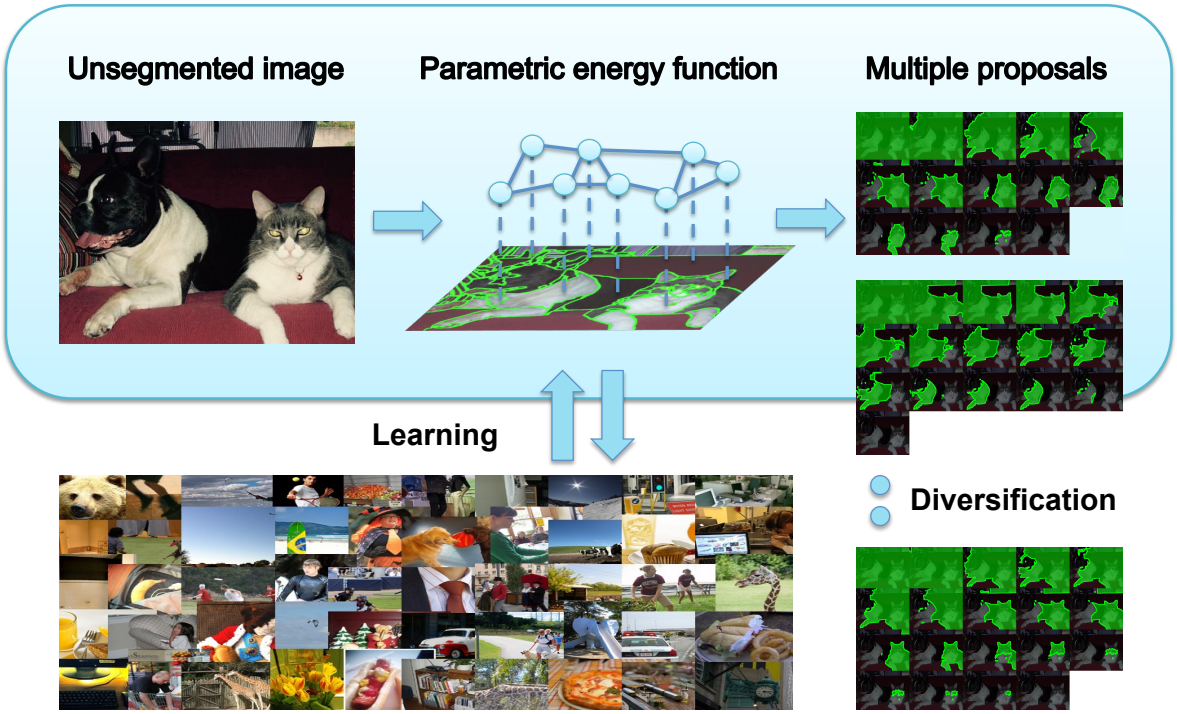


Figure 6.1: Our approach takes an input image, partitions it into superpixels, and groups superpixels into region proposals using a novel structured learning framework for parametric energy functions, called Parametric Min-Loss (PML). The parametric energy function combines mid-level cues with weights that are trained to generate multiple region proposals. Finally, we diversify the energy function to generate a diverse set of region proposals.

ment needed for bottom-up grouping. One such tool that has emerged in the computer vision community is parametric maxflow [53], which is used to minimize an energy function $E^{\lambda}(y)$ for multiple values of parameter λ, generating multiple solutions at a time. Parametric maxflow was applied to proposing object regions in CPMC [15] and in subsequent variants [46, 94].

Despite frequent use in region proposal generation, however, parametric energy functions are, as a rule, not trained to predict multiple outputs, but rather trained to predict a single output. To the best of our knowledge, this thesis is the first to bridge the gap between learning and inference for parametric maxflow. Our formulation is inspired by MCL, which models multiple-output learning with a loss function that evaluates multiple outputs against a correct output. Our model, however, differs from MCL in 1) having a

single parametric energy function, and in 2) automatically adapting the number of output solutions to the input image. Despite these significant differences, we find that MCL's block-coordinate descent strategy applies to parametric maxflow and yields a solution that decomposes into simple alternating steps.

In summary, we introduce Parametric Min-Loss (PML), a novel model and algorithm for structured multiple-output learning using parametric maxflow. We demonstrate its use for learning to combine a set of mid-level grouping cues to yield a set of region proposals with high recall. Besides having applications to perceptual grouping, the model bridges the disparity between learning and inference for parametric energy functions and can be applied to any domain that uses parametric maxflow. While learning accounts for diverse multiple outputs, we include a complementary diversification step that allows the proposals to adapt to different conditions. With a large-scale experimental validation, we cast mid-level cue combination in a structured learning environment, representing an exciting new direction for perceptual grouping.

6.1 Related work

Perceptual grouping has often been formulated as an energy minimization problem, *e.g.* [43, 117, 121, 19, 126, 50, 103], yielding a single region or (possibly) closed contour, or a partition into regions. In the more recent context of generating region proposals, a *parametric* energy minimization problem is often formulated (*e.g.* CPMC [15]) in which the energy is parameterized by λ and minimized for multiple values of λ using parametric maxflow, yielding multiple solutions. Such an approach is an extension of energy minimization from predicting a single output to predicting multiple outputs in support of the principle of least commitment, and has been refined by subsequent variants [46, 94]. However, the combination of cues is typically specified manually in the energy or not trained jointly in the energy.

Moreover, a gap has emerged between learning and inference for parametric maxflow because prediction has been extended to multiple outputs while learning has not. This disparity exists in general for multiple-output models, an example of which is the M-Best MAP approach for generating multiple hypotheses [41]. Recently, Multiple Choice Learning (MCL) [45] addressed this gap in a tractable way using an M-tuple of independent structured predictors that predicts M outputs. The model is efficient and minimizes the loss of only the most accurate prediction in the set of outputs. Subsequent improvements included an explicit criterion to encourage diversity among the predictors [44], however the model remains fundamentally different from parametric maxflow, which solves a single parametric energy function that accounts for multiple outputs, and whose number of outputs is adaptive and does not need to be pre-specified. Our method for parametric maxflow, however, is similar to MCL in using a block-coordinate descent strategy in a large-margin formulation to close the train-test gap.

Approaches for region proposals typically consist of a generation stage for hypothesizing proposals, followed by a ranking stage that attempts to order them by "objectness". A diversity of approaches exist in which many generate proposals in the form of bounding boxes, *e.g.* Objectness [2] and Edge Boxes [127]. In such methods, a sliding window suffices as no explicit grouping is required, and they are suitable for box-based detectors even though proposals do not explicitly capture the underlying shape of the objects. Selective Search [116] efficiently generates region-based proposals based on greedily merging superpixels and was subsequently improved with trained affinity functions [123]. The approach is similar to ours in using region-based similarity cues, however the agglomerative grouping procedure is brute force.

In approaches for region-based proposals, such as GOP [55], RIGOR [46], MCG [5], the principle of least commitment is typically not built into learning. Only very recently was such a method proposed [56] that minimized the loss of the most accurate region proposal, with efficient runtime at test time and achieving competitive results. In our

work, we minimize the same loss function, however one of our key aims is to develop a graphical model that is unified across learning and inference. Another recent work [18] also uses learning to combine several cues for generating object proposals in 3D, but it does not use parametric energies. Earlier methods gave a significant role to learning in the ranking stage, *e.g.* [5, 29, 94]. CPMC [15] uses parametric maxflow to generate proposals and is most similar to ours in spirit, however we perform grouping at superpixel-level rather than pixel-level. This allows access to region-based mid-level cues during the generation stage. In contrast to the above methods, our approach emphasizes the generation stage over the ranking stage, and emphasizes the role of learning to group using mid-level cues. The closest methods to our approach are Superpixel Closure [66], which uses mid-level closure, but does not combine other cues, and Multicue [59], which combines mid-level cues in a parametric energy function, but only trains the energy to generate a single proposal.

6.2 Perceptual grouping cues

Our method begins by segmenting the input image x into a single layer of superpixels that forms the basis of feature extraction, labeling, and grouping. Superpixels reduce search complexity while providing access to local region and contour scope. At the same time, we are restricting regions to superpixel boundaries, so it is important to preserve boundary recall. The resulting strategy is to oversegment the image into superpixels which remain to be grouped.

Formally, we partition the image x into a set S of superpixels, from which we seek a subset $R \subset S$ that represents an object. Equivalently, we represent R as a binary labeling $\mathbf{y} \in \{0,1\}^{|S|}$, where $y_p = 1$ exactly when superpixel p is in R, for $p = 1, \ldots, |S|$, hence $R = \{p : y_p = 1\}$. The space of possible regions lies in $\mathcal{Y} = \{0,1\}^{|S|}$.

Given an image x, we seek a minimum energy region $\mathbf{y} \in \mathcal{Y}$ with respect to the

energy $E^\lambda(x) : \mathcal{Y} \to \mathbb{R}$ which is defined for the image and a parameter λ. Specifically, we minimize the energy function:

$$E^\lambda(x, \mathbf{y}) = \lambda \sum_p \phi_0(x, y_p) + \mathbf{w}_1^\mathsf{T} \sum_p \phi_1(x, y_p) + \mathbf{w}_2^\mathsf{T} \sum_{p,q} \phi_2(x, \mathbf{y}_{p,q}), \tag{6.1}$$

whose terms here are grouped by weighted features ϕ_0, ϕ_1, ϕ_2. This energy can be minimized for multiple values of λ by parametric maxflow under further constraints (see [53]), however the goal of this section is to model mid-level grouping cues in the energy. To do so, we regroup the energy (6.1) into subenergies that model their respective cues:

$$E^\lambda(x, \mathbf{y}) = E_{\text{app}}(x, \mathbf{y}) + E_{\text{clo}}(x, \mathbf{y}) + E_{\text{sym}}(x, \mathbf{y}) + E^\lambda_{\text{scale}}(x, \mathbf{y}). \tag{6.2}$$

The following sections will define the subenergies above.

6.2.1 Proximity

The grouping cue of proximity is a basic image relation that is preserved through image projection. Since pairwise potentials encode grouping relations, proximity is reflected in placing a potential on every pair of *adjacent* superpixels, thereby defining the edge set $\mathcal{A}(S) \subset S^2$.

6.2.2 Appearance similarity

Appearance similarity is a non-accidental regularity of objects—the more similar a group of elements are to each other, the more likely they belong to the same object. We extract a color histogram $\mathbf{h}_p^{\text{col}}$ of d^{col} dimensions for every superpixel p, and define a similarity

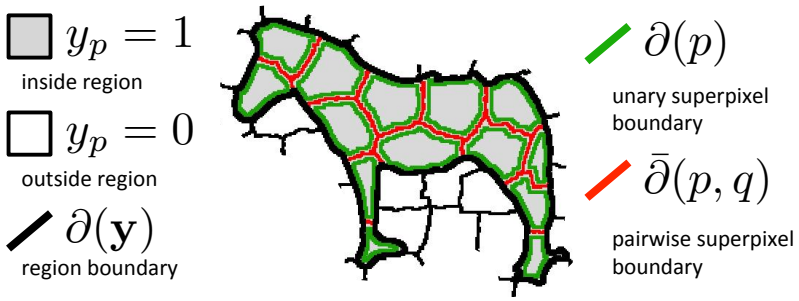


Figure 6.2: Given a region defined by $\mathbf{y} \in \{0,1\}^{|S|}$, the closure cue sums gap along its boundary $\partial(\mathbf{y})$. Summation is regrouped into unary superpixel boundaries $\partial(p)$ and pairwise superpixel boundaries $\bar{\partial}(p,q)$ for superpixels inside the region (see text for details).

for every pair $(p,q) \in \mathcal{A}(S)$ using the histogram intersection kernel [116]:

$$\text{sim}_{p,q}(\mathbf{h}) = \sum_{i=1}^{d} \min(\mathbf{h}_p(i), \mathbf{h}_q(i)) \tag{6.3}$$

We similarly define similarity for a texture histogram $\mathbf{h}_p^{\text{text}}$ of d^{text} dimensions. Appearance similarity is encoded into our energy as a 2-dimensional feature consisting of color and texture:

$$\phi_{\text{app}}(x, \mathbf{y}_{p,q}) = \mathbb{1}_{[y_p \neq y_q]}(\text{sim}_{p,q}(\mathbf{h}^{\text{col}}), \text{sim}_{p,q}(\mathbf{h}^{\text{text}})) \tag{6.4}$$

We note that ϕ_{app} contributes a cost only when neighboring superpixels with similar appearance are labeled differently, while there is no cost when neighboring superpixels with dissimilar appearance are labeled the same. The potentials are weighted by $\mathbf{w}_{\text{app}}$ which is trained and shared across all superpixel pairs, and overall contributes to the following energy:

$$E_{\text{app}}(x, \mathbf{y}) = \mathbf{w}_{\text{app}}^{\mathsf{T}} \sum_{p,q} \phi_{\text{app}}(x, \mathbf{y}_{p,q}). \tag{6.5}$$

6.2.3 Contour closure

Contour closure is a non-accidental regularity of objects, in which object coherence in 3D projects to a closed boundary in 2D. The more contour evidence there is along the boundary of a given region $\mathbf{y}$, the more likely it is to enclose an object. We use a cost function that sums contour gap $G(x, \mathbf{y}) = \sum_{b \in \partial(\mathbf{y})} g(x, b)$ along the region boundary $\partial(\mathbf{y})$, where $g(x, b)$ is the gap (lack of contour) evaluated at pixel b of image x.

To express the cost $G(x, \mathbf{y})$ in the form of unary and pairwise features [66], we first define a unary feature:

$$\phi_{\text{clo}}(x, y_p) = \sum_{b \in \partial(p)} \mathbb{1}_{[y_p=1]} g(x, b) \tag{6.6}$$

that sums gap along selected superpixel boundaries. For a region consisting of a single superpixel, the unary feature sums the correct gap cost. However, as shown in Figure 6.2, for a region consisting of multiple superpixels, simply summing the unary features will double count the gaps along the internal boundaries shared by adjacent superpixels. We thus define pairwise features to cancel them out:

$$\phi_{\text{clo}}(x, \mathbf{y}_{p,q}) = \sum_{b \in \bar{\partial}(p,q)} \mathbb{1}_{[y_p=y_q=1]} g(x, b) \tag{6.7}$$

The gap $G(x, \mathbf{y})$ of region $\mathbf{y}$ is thus the sum of the unary features, minus twice the pairwise features. In summary, the closure cue contributes the weighted energy:

$$E_{\text{clo}}(x, \mathbf{y}) = w_{\text{clo}}^{\mathsf{T}} \left(\sum_p \phi_{\text{clo}}(x, y_p) - 2 \sum_{p,q} \phi_{\text{clo}}(x, \mathbf{y}_{p,q}) \right) \tag{6.8}$$

6.2.4 Symmetry

Symmetry is a powerful regularity in objects. While symmetry captures interactions among all parts of an object, this must be balanced with the need for a low-order energy.

Coarse superpixels help by expanding the spatial scope of each unit, however superpixel size must also be limited in order to preserve boundary recall. Overall, it is a computational challenge to capture grouping by symmetry.

We follow the approach of [59] of "outsourcing" symmetry to a region-based symmetry detector [58], and biasing our energy to detected symmetric parts. Formally, given a set T of region-scoped, scored symmetric parts, we define pairwise potentials that prefer to merge superpixels when they fall in the same symmetric part. For every pair $(p, q) \in \mathcal{A}(S)$ we define:

$$\phi_{\text{sym}}(x, \mathbf{y}_{p,q}) = \mathbb{1}_{[y_p \neq y_q]} \max_{s \in S(p,q)} \text{score}(s), \tag{6.9}$$

where the max considers symmetric parts $T(p, q) \subseteq T$ that overlap p and q by at least $\tau = 0.75$, and selects the best-scoring one. A value of zero is assigned when $T(p, q)$ is empty. Non-maximum suppression is applied over all superpixel pairs so that at most one symmetric part contributes to each pair. Overall, the symmetry cue penalizes labeled regions that cut through symmetric parts, via the weighted energy:

$$E_{\text{sym}}(x, \mathbf{y}) = w_{\text{sym}}^{\mathsf{T}} \sum_{p,q} \phi_{\text{sym}}(x, \mathbf{y}_{p,q}). \tag{6.10}$$

6.2.5 Object scale

The grouping energies above accumulate higher costs for regions with more superpixels, and thus the energy is artificially biased toward smaller regions and needs to be normalized by the region's size. To do so, we subtract unary features ϕ_{area} scaled by a factor $|\lambda|$ from the energy, with the effect of accommodating larger regions as $|\lambda|$ increases. Practically, a non-zero λ is necessary to remove trivial solutions. We define

$\phi_{\text{area}}(x, y_p) = \mathbb{1}_{[y_p=1]}\text{area}(p)$, which contributes the negative quantity:

$$E^{\lambda}_{\text{scale}}(x, \mathbf{y}) = \lambda \sum_{p} \phi_{\text{area}}(x, y_p). \tag{6.11}$$

A diverse set of solutions can be obtained with different values of λ. Note that the cost of selecting an individual superpixel is influenced by the magnitude of λ against other potentials: a very large $|\lambda|$ will more than offset the other potentials and cause all superpixels to be selected. Since potentials are empirically below 1, we can obtain all solutions by varying λ within $[-1, 0]$.

6.3 Parametric energy minimization

The domain $\mathcal{Y}$ over which the energy (6.1) is minimized is too large for exhaustive search, but when written as a sum of unary and pairwise potentials, the energy is seen to have the required structure for an efficient solution. When submodular pairwise potentials are guaranteed by requiring $\mathbf{w} \geq 0$, and λ is held at a fixed non-positive value, the problem

$$\hat{\mathbf{y}} = \arg\min_{\mathbf{y}} E^{\lambda}(x, \mathbf{y}, \mathbf{w}) \tag{6.12}$$

can be solved exactly by a maxflow algorithm. Solving (6.12) for all values $\lambda \in [\lambda^{\min}, \lambda^{\max}]$ simultaneously is known as a parametric problem, and can be done via parametric maxflow. Furthermore, since the linear term ϕ_0 measures area, the monotonicity property is satisfied that guarantees a solution set of size linear in $|S|$ [53]. See Figure 6.3 for a visualization of the solution set in an input image.

We rewrite (6.1) in a linear form that is amenable to large-margin learning [113] by stacking the features and weights of individual cues together. Specifically, we define a weight vector $\mathbf{w} = (\mathbf{w}_0, \mathbf{w}_1, \mathbf{w}_2)$ where $\mathbf{w}_0 = 1$, and a feature vector $\phi^{\lambda} = (-\lambda\Phi_0, -\Phi_1, -\Phi_2)$. We can then rewrite $E^{\lambda}(x, \mathbf{y}, \mathbf{w}) = -\mathbf{w}^{\mathsf{T}}\phi^{\lambda}(x, \mathbf{y})$ and thus rewrite

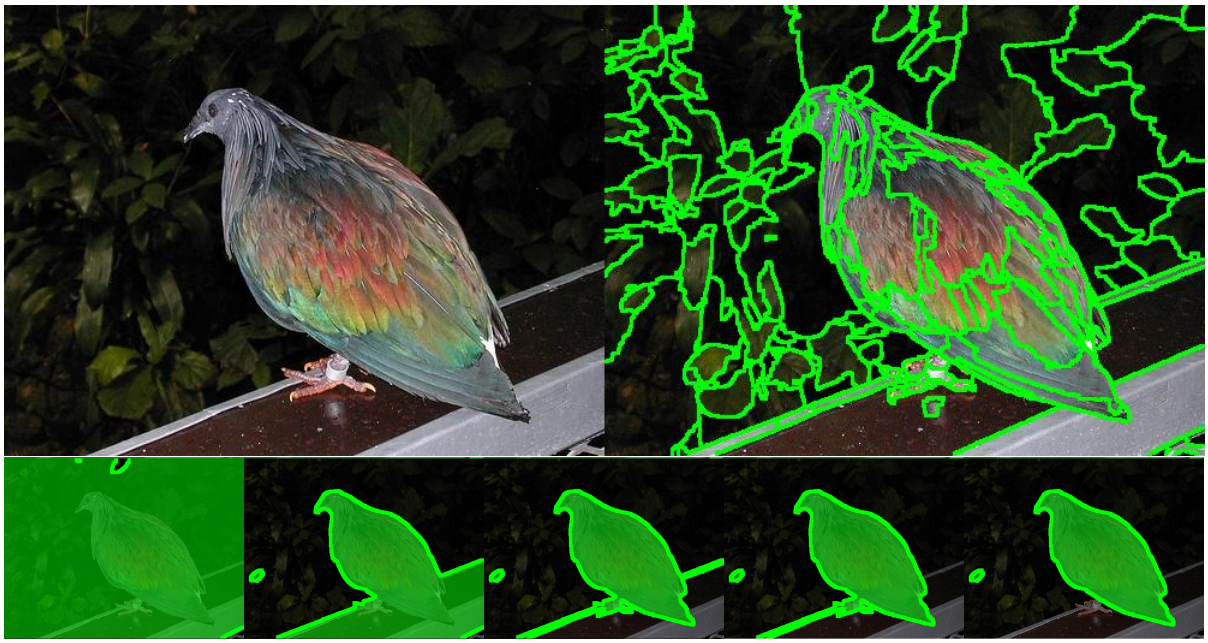


Figure 6.3: Given the input image and superpixel segmentation shown in the first row, our approach defines a parametric problem whose solution set is shown in the second row. Optimal labelings are listed in order of increasing $\lambda \in [-1, 0]$.

(6.12) as:

$$\hat{\mathbf{y}}(x, \mathbf{w}) = \arg\max_{\mathbf{y}} \mathbf{w}^{\mathsf{T}} \phi^{\lambda}(x, \mathbf{y}). \tag{6.13}$$

Finally, the structured prediction function (6.13) is generalized to a set of solutions over a range of λ:

$$\hat{Y}(x, \mathbf{w}) = \{\hat{\mathbf{y}}^{\lambda}(x, \mathbf{w}) : \lambda \in [-1, 0]\}. \tag{6.14}$$

6.4 Parametric Min-Loss learning

When a ground truth region g annotates an object in input image x, the quality of the set $\hat{Y}(x, \mathbf{w})$ of predicted regions can be evaluated against g. In the evaluation of region proposals, for example, Jaccard similarity is considered by the Average Best Overlap

(ABO) metric [116]. In S-SVM learning [113], a task loss $\ell(\hat{\mathbf{y}}, \mathbf{y})$ measures the mismatch of a structured prediction $\hat{\mathbf{y}}$ against $\mathbf{y}$. To measure the mismatch of a set $\hat{Y}$ of structured predictions, however, we generalize the task loss to a set in the following way:

$$\mathcal{L}(\hat{Y}, \mathbf{y}) = \min_{\hat{\mathbf{y}} \in \hat{Y}} \ell(\hat{\mathbf{y}}, \mathbf{y}), \tag{6.15}$$

where the min-task loss (6.15) says that the quality of the entire set $\hat{Y}$ of predictions is the quality of the best prediction. As in standard S-SVM, the task loss ℓ is defined to be amenable to loss-augmented inference [113] and decomposes into a sum of unary losses. Each unary loss uses v_p, as defined below, to measure the mismatch of superpixel p against the ground truth region g as follows:

$$\ell(\hat{\mathbf{y}}, \mathbf{y}(g)) = \frac{1}{|g|} \sum_p |p| \begin{cases} v_p & \hat{y}_p = 0 \\ 1 - v_p & \hat{y}_p = 1, \end{cases} \tag{6.16}$$

where v_p is the fraction of p's pixels that lie in g.

The weights of (6.1) are ideally learned by minimizing $\mathcal{L}(\hat{Y}, y)$ in $\mathbf{w}$, but in order to circumvent difficulties arising from non-convexity and discontinuities, we develop a related loss function $H(\mathbf{w})$ that is easier to minimize. Our derivation of $H(\mathbf{w})$ follows a strategy based on the (structured) hinge loss (see, *e.g.* Def. (6.1) in [87]): as the hinge loss is an upper bound of the task loss, we derive a min-hinge loss that is an upper bound of the min-task loss [45]. We first write the hinge loss for parametric maxflow as follows, with dependence on the training example $(x, \mathbf{y})$ omitted for brevity:

$$h(\mathbf{w}, \lambda) = \max_{\hat{\mathbf{y}}} \ell(\hat{\mathbf{y}}, \mathbf{y}) + \mathbf{w}^\mathsf{T} \phi^\lambda(x, \hat{\mathbf{y}}) - \mathbf{w}^\mathsf{T} \phi^\lambda(x, \mathbf{y}). \tag{6.17}$$

Algorithm 1 Parametric Min-Loss

Require: $\{\phi^{\lambda}(x,\cdot),\mathbf{y},\ell\}_{n=1}^{N}$ ▷ energy potentials, ground truth, task loss
Ensure: $\mathbf{w}^{*} \geq 0$ ▷ positive weights for pairwise submodularity
$\tau \leftarrow 0$ ▷ set iteration counter
repeat
  $\tau \leftarrow \tau + 1$ ▷ increment iteration counter
  $\mathbf{w}^{(\tau)} \leftarrow \text{StructuredSVM}(\{\phi^{\lambda^{(\tau)}}(x,\cdot),\mathbf{y},\ell\}_{n=1}^{N})$ ▷ solve S-SVM
  for $n \leftarrow 1 \ldots N$ **do**
    $\{(\lambda_i,\mathbf{y}_i)\} \leftarrow \text{ParametricMaxflow}(-\ell(\mathbf{y}_n) - \mathbf{w}^{\mathsf{T}}\phi(\mathbf{x}_n),[-1,0])$ ▷ solve min-hinge
    $h_i \leftarrow \ell(\mathbf{y}_n,\mathbf{y}_i) + \mathbf{w}^{\mathsf{T}}[\phi^{\lambda_i}(\mathbf{x}_n,\mathbf{y}_i) - \phi^{\lambda_i}(\mathbf{x}_n,\mathbf{y}_n)], \forall i$ ▷ solve min-hinge
    $\lambda_n^{(\tau)} \leftarrow \lambda_{\arg\min h_i}$ ▷ solve min-hinge
  end for
until converged or maxed out
return $\mathbf{w}^{*} \leftarrow \mathbf{w}^{(\tau)}$

The min-hinge loss $H(\mathbf{w})$ then takes the minimum of $h(\mathbf{w},\lambda)$ over a range of λ:

$$H(\mathbf{w}) = \min_{\lambda\in[-1,0]} h(\mathbf{w},\lambda). \tag{6.18}$$

Unlike in standard S-SVM, the loss function $H(\mathbf{w})$ is not guaranteed to be convex, however it is shown that H is an upper bound for $\mathcal{L}$ [45].

Accounting for all training examples $\{(x_n,\mathbf{y}_n)\}_{n=1}^{N}$ where each $\mathbf{y}_n$ is a ground truth region, we obtain the regularized min-hinge minimization problem:

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + \frac{C}{N}\sum_{n=1}^{N} \min_{\lambda_n\in[-1,0]} h_n(\mathbf{w},\lambda_n). \tag{6.19}$$

Although solving (6.19) is an NP-hard problem, we can derive an efficient solution by rewriting the problem as:

$$\min_{\mathbf{w}} \min_{\{\lambda_n\}} \frac{1}{2}||\mathbf{w}||^2 + \frac{C}{N}\sum_{n=1}^{N} h_n(\mathbf{w},\lambda_n) \tag{6.20}$$

and decomposing it into two simpler problems that we identify as λ-update and standard S-SVM. Our algorithm, summarized in Algorithm 1, alternates between holding $\mathbf{w}$ fixed and optimizing the λ's, and holding the λ's fixed and solving S-SVM.

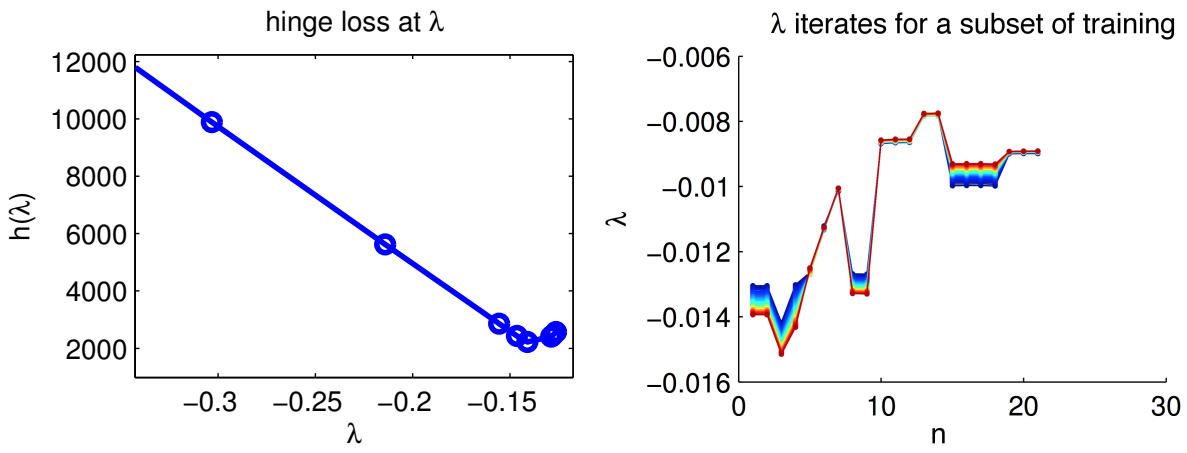


Figure 6.4: The convex function $h(\lambda)$ sampled at breakpoints (left), and iterates of $\lambda_n^{(\tau)}$ over time τ for each n in a subset of training examples, where later iterations (higher τ) are represented by warmer colors (right).

Fixing $\mathbf{w}$, we obtain N independent problems that can be solved in parallel. Each problem amounts to solving parametric minimization of the hinge-loss:

$$\arg \min_{\lambda \in [-1,0]} h(\mathbf{w}, \lambda) \tag{6.21}$$

Since the function $h(\mathbf{w}, \lambda)$ is the maximum of $2^{|S|}$ linear functions, it is convex and piecewise-linear. It follows that $h(\mathbf{w}, \lambda)$ reaches its minimum value at one of the λ breakpoints, and so we need only to search for the breakpoint that evaluates to a minimum. The set of breakpoints $\{\lambda_i\}$ and their solutions $\{\mathbf{y}_i\}$ are found by solving the parametric maxflow problem:

$$\forall \lambda \in [-1, 0], \min_{\hat{\mathbf{y}}} -\ell(\hat{\mathbf{y}}, \mathbf{y}) - \mathbf{w}^\mathsf{T} \phi^\lambda(x, \hat{\mathbf{y}}) \tag{6.22}$$

To solve (6.21), we exhaustively evaluate $h(\mathbf{w}, \lambda)$ for each λ_i using $\mathbf{y}_i$. We note that λ has monotonic coefficients and thus there are at most $O(|S|)$ breakpoints [53] containing the solution. See Figure 6.4 for an illustration.

With $\{\lambda_n\}$ fixed, problem (6.19) reduces to a single, standard S-SVM problem:

$$\mathbf{w} = \arg\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + \frac{C}{N}\sum_{n=1}^{N} h_n(\mathbf{w}, \lambda_n) \tag{6.23}$$

We solve (6.23) with the constraint $\mathbf{w} \geq 0$ using the cutting-plane implementation of [99].

Although the learning algorithm alternates between minimizing $\mathbf{w}$ and λ, the learning goal for region proposals is to optimize the weights. Minimization in λ reflects the selection of the best region by a ground truth oracle, and prediction has no access to such an oracle. Moreover, in the absence of a specified object category, bottom-up grouping cues are the only means of predicting region proposals.

6.5 Diversification

Diversification is an important step toward achieving recall without a specified object category. Typically, diverse proposals are generated based on local features sampled from different parts of the image, often called seeds [15, 46, 94]. In a given image, we sample seeds that make assumptions about object properties such as image location and color distribution. An energy function that is biased with a particular seed will then yield proposals that are customized toward a particular location or color distribution. Pooling together proposals associated with different seeds allows us to cover a wide range of conditions with greater precision.

Location. We use individual superpixels to seed image locations, with a total of $|S|$ seeds. As shown in Figure 6.5, each seed p defines unary features that discourage selecting $y_q = 1$ depending on q's "distance" from p. This is encoded for any superpixel q with a cost based on the maximum distance of q from p. For non-compact superpixels, this promotes compactness by encouraging smaller, nearby regions to be "annexed" first (regardless of their color).

Figure 6.5: A location-based seed is sampled on the motorcyclist's back, which induces the unary potentials as shown. Warmer colors represent higher costs.

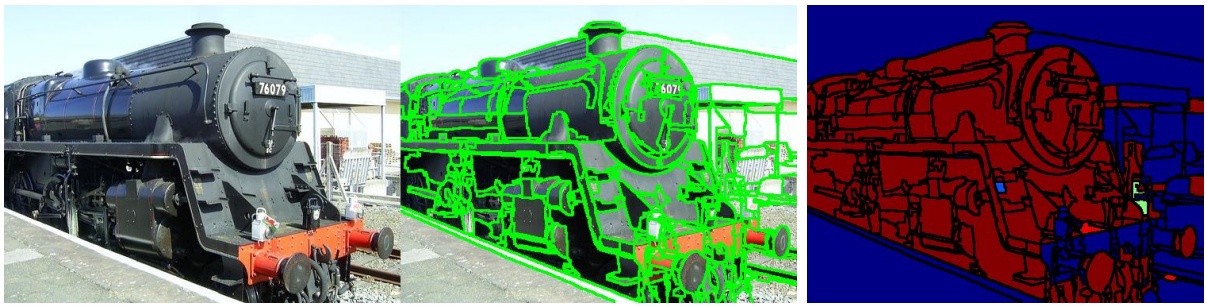


Figure 6.6: A color-based seed is sampled on the pair of foreground-background color distributions, which induces the unary potentials as shown.

Color similarity. Here, we seed image colors (without considering proximity) using a Gaussian mixture model applied to the color space. Specifically, we seed foreground-background pairs of color distributions. For any given seed, each superpixel q has a likelihood under the foreground distribution and a likelihood under the background distribution. The higher the likelihood ratio, the lower the cost of assigning $y_q = 1$. An example is provided in Figure 6.6.

In our application, diversification is complementary to PML. While learning accounts for diversity over scale in λ, here the energy is further diversified in location and color. Moreover, learning and diversification balance each other out, as the grouping cues combined in the energy have a tempering effect on diversification seeds, *e.g.* by helping a seed centered on a location to adapt to irregular shapes.

6.6 Postprocessing

All solutions pooled over diversification seeds enter a pipeline of postprocessing steps. First, we process each solution $\mathbf{y} \in \mathcal{Y}$ to ensure that contiguous regions are considered for recall. We find connected components efficiently in superpixel space, and include the largest $M = 2$ connected components as region proposals.

We then remove artefactual regions in the form of empty labelings and labelings that are within a very high percentage of the image's total area. We filter out redundant regions in the form of duplicate labelings and clusters of labelings that are similar in overlap. Similarly to [15], we perform agglomerative clustering of labelings by intersection-over-union overlap, and consider clusters of labelings that exceed a very high overlap threshold. For each cluster, we keep the labeling with the best closure and discard the rest. Closure is efficiently computed using the gap cost $G(x, \mathbf{y})$.

Finally, we rank the proposals to allow a small number of proposals to be selected. We cast this as a problem of assigning a classification score to each region that indicates how object-like it is. Unlike the perceptual grouping problem above, this is a verification step in which higher-order relations are more easily captured over the full region scope. We turn to convolutional neural networks as they yield good categorization results. The final network layers are fully connected, and can be thought of as learned, mid-level features that encode category-independent information that is relevant for categorization. Specifically, to extract a feature vector for a given region proposal R, we place a cropping box tightly around R, and warp the cropped image to normalized dimensions. After normalizing pixel values, we evaluate OxfordNet and retain layer 20 as a 4096-dimensional feature vector. Like R-CNN [42], we then trained a SVM classifier on the feature to assign 'object' or 'non-object' to each R, and trained a logistic regressor to map the output margin to a score between 0 and 1.

We obtained features for positive and negative training examples by sampling from the training images of the VOC 2012 segmentation subset. For each image, we use

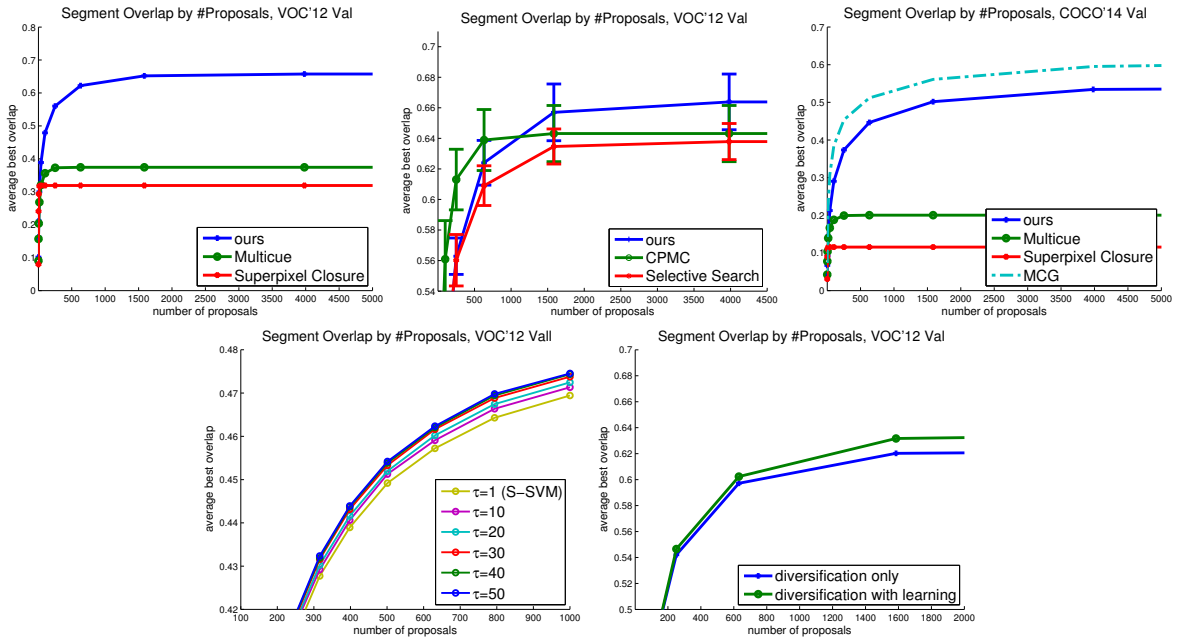


Figure 6.7: Comparison with perceptual grouping methods Multicue [59] and Superpixel Closure [66] (far left), and CPMC [15] and Selective Search [116] (left) on VOC'12. Comparison with MCG [5] and perceptual grouping methods on COCO'14 (middle). See text for comparisons with more recent methods like GOP [55] and RIGOR [46]. Parametric Min-Loss learning improves segment overlap as weights evolve (right). Diversification alone, using location- and color-based seeds, is able to bring us quite far, however the remaining performance gap is achieved by learning a combination of mid-level cues (far right).

the ground truth boxes as positive examples, and a matching number of random boxes as negative examples.

6.7 Results

For quantitative evaluation, the SEGMENTATION subset of VOC 2012 provides a set of images containing different objects annotated with at least one ground truth region per image. We apply Parametric Min-Loss on the TRAIN subset and evaluate our trained method on the VAL subset. We use $\mathcal{P}$ to denote the set of proposed regions to be evaluated, and $\mathcal{G}$ to denote the corresponding set of ground truth regions. For all pairs $(p \in \mathcal{P}, g \in \mathcal{G})$ contained by the same image, we consider the Jaccard similarity $\mathcal{J}(p, g) =$

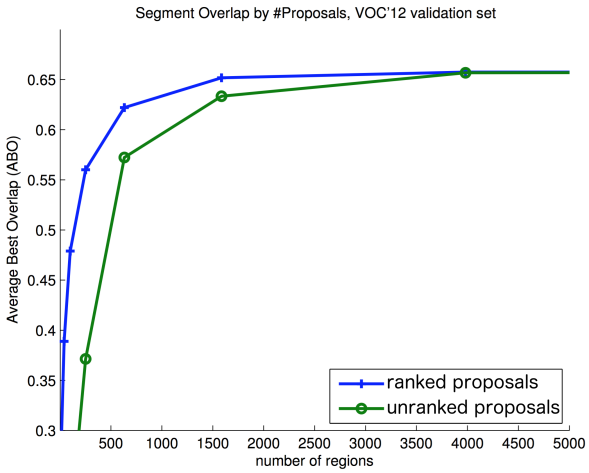


Figure 6.8: Accuracy improvement due to ranking the proposals.

$\frac{|p \cap g|}{|p \cup g|}$ to score the quality of a potential match. The Average Best Overlap (ABO) metric [116] is defined over all images as follows:

$$\text{ABO}(k) = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \max_{p \in \mathcal{P}(k)} \mathcal{J}(p, g),$$

where $\mathcal{P}(k)$ represents the top k regions proposed for g's image. Plots sample ABO on increasing values of k to show the trade-off between recall and the number of proposals. The improvement in recall due to ranking is shown in Figure 6.8.

Our method requires a single superpixel layer as preprocessing. We found that non-compact superpixels, *e.g.* Felzenszwalb & Huttenlocher [32], yielded better results than compact superpixels, *e.g.* SLIC [1]. Results in this paper were generated using UCM [4], thresholded at $k = 0.1$.

Overall results. We first compare our method with two recent perceptual grouping methods most similar to ours, in Figure 6.7 (far left). Superpixel Closure [66] used parametric maxflow to group superpixels into regions of minimum closure cost, while Multicue [59] used S-SVM to train a parametric energy function that combines appearance, closure, and symmetry cues. We improve on these methods via a holistic learning framework and effective diversification. While both methods were quantitatively evaluated only on

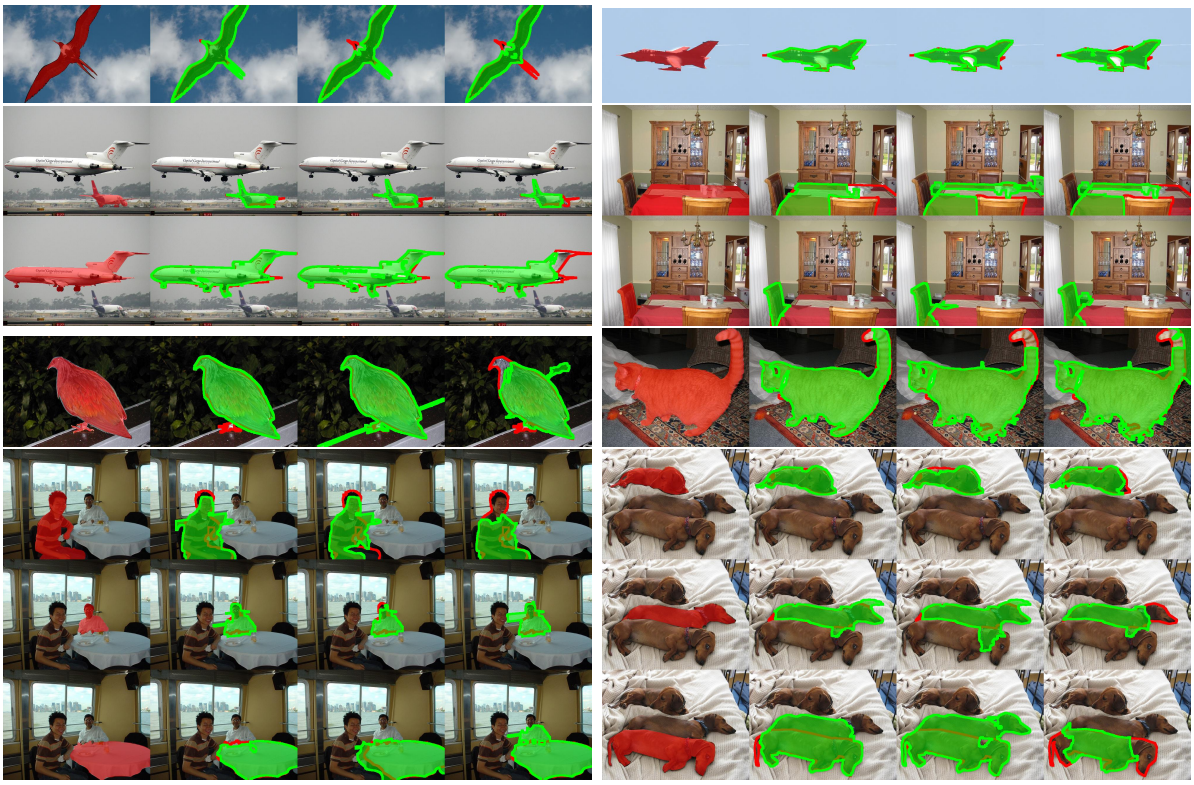


Figure 6.9: Example region proposals found for images from VOC 2012. Red masks denote ground truth, green masks denote the corresponding top proposals (from left to right).

the Weizmann Horse Database, we evaluate on VOC 2012 val segmentations and COCO 2014 val segmentations.

Our results are comparable with leading region proposal methods such as Selective Search [116] and CPMC [15] on VOC 2012 val, as shown in Figure 6.7 (left). At 1585 proposals, GOP [55], RIGOR [46], MCG [5], and CMPC achieve 75.1, 74.4, 69.8, and 64.9 ABO, respectively, while ours achieves 65.2 ABO, so we are outperformed by the most recent methods. Ours takes a similar approach to Selective Search in using regional features to group superpixels, however we train a combination of cues and find minimum energy regions, allowing "better focused" proposals. Our level of recall, however, is more comparable to that of CPMC's for higher numbers of proposals. While we achieve higher recall with learning and effective diversification, our simple ranking procedure with a SVM classifier does less for precision than the sophisticated overlap regressors of CPMC.

The runtime of our unoptimized Matlab implementation is, on average, several seconds per image, excluding preprocessing (gPb-UCM superpixels). Although this is not competitive with state-of-the-art runtime, there are unexploited opportunities to further optimize and parallelize the implementation of mid-level cues to obtain speed-ups.

In Figure 6.9, we show some example region proposals. For the images of the airplane, bird, and cat, our approach does well at separating figure from ground. For images of more complex scenes such as the dinner table, over- and undersegmentation occurred due to low contrast or in objects of highly heterogeneous appearance.

Learning. The second part of our results focuses on learning. We note that S-SVM is a natural baseline for Parametric Min-Loss due to the structure of the iterative algorithm. Specifically, we track the energy functions corresponding to weights as they evolve over iterations (indexed by τ), where the first iteration corresponds to S-SVM with initial λ values. We initialize λ's to -0.01, as done in Multicue [59]. As shown in Figure 6.7 (right), successive energy functions yield better recall, iteratively improving on the S-SVM baseline. Additionally, since recall is measured by segment overlap, the result also shows that Parametric Min-Loss and its surrogate are effective approximating training objectives. Finally, in Figure 6.7 (far right), we show the impact of learning a combination of mid-level cues as distinguished from our diversification steps based on location and color seeds. Diversification alone brings us quite far, however the remaining gap in performance is achieved by learning a combination of mid-level cues. This shows that learning is complementary to diversification.

6.8 Conclusion

We introduced Parametric Min-Loss (PML), a novel structured learning framework for parametric energy functions, and demonstrated it in the context of region proposal generation. Our perceptual grouping method learns how to combine multiple cues to generate

a set of figure-ground region proposals. By applying the MCL optimization strategy to parametric maxflow, we bridge the gap between learning and inference for parametric energy functions. Moreover, our framework supports efficient superpixel-based diversification that yields a diverse set of region proposals that competes favorably with recent state of the art on VOC 2012. In future work, we plan to use our general framework to learn how we can integrate other classical grouping cues to improve region proposal generation.

Chapter 7

Conclusions

We began in Chapter 1 with an historical overview of perceptual grouping, starting with the bottom-up cues of the Gestalt principles. Subsequently, we saw that methods in computer vision have progressed from clean, segmented inputs to natural images of scenes containing multiple cluttered objects. Likewise, object recognition can now handle important sources of variability such as part deformation. More recently, we have seen a shift from sliding window proposals to region proposals of arbitrary shape and size. Bottom-up grouping is regaining momentum as a counterpart to object detection, and is a promising area in which to explore the importance of mid-level grouping cues.

We traced older and more recent developments in the field to equip ourselves with an intellectual basis for advancing the field. In doing so, we identified a set of issues in perceptual grouping that we set out to tackle. These are 1) the challenge of managing the complexity of searching for plausible figure-ground segmentations in cluttered and occluded scenes, and more specifically, the lack of mid-level knowledge in current region proposal methods, 2) the challenge of handling a wide range of input variability in a robust way, notably reducing the sensitivity to properties of lower importance such as object appearance, and combining information from multiple grouping cues to widen the domain of applicability, and 3) the challenge of learning to do perceptual grouping in an

image from training data, in particular by performing prediction jointly in the full image rather than at independent image locations, and addressing the lack of multiple-output models that are necessary to capture the structure of bottom-up grouping.

We proceeded to Chapter 3 on object categorization to provide additional support to the issues of perceptual grouping. Specifically, we adopted the approach of automatic perceptual grouping as a scalable alternative to expensive manual annotations, and addressed the issue of computational complexity. Secondly, we handled within-category variation by choosing a shape feature that yielded dramatic improvements, demonstrating the importance of feature representation when dealing with the issue of variability.

We then outlined the plan of attack in our thesis in three steps, and presented our work starting with Chapter 4. Like all Gestalt grouping cues, symmetry is a ubiquitous regularity that our visual system has evolved to focus on to help us perceptually organize images of scenes. In the absence of an object prior, symmetry is a powerful cue for detecting parts whose configurations, in turn, can help manage search in a large-scale recognition task. The symmetric part detection framework of [65] draws on the power of the medial axis while avoiding its pitfalls. However, it suffers from some serious limitations that limit its ability to detect more general classes of symmetric parts. In our first contribution, we have addressed these limitations by introducing a number of extensions to [65], including both a richer yet more flexible model for symmetry, a multi-scale framework, and a discrete optimization formulation based on the regularity of good continuation. The resulting framework significantly outperforms that of [65], offering an improved perceptual grouping framework for recovering symmetric parts without a priori knowledge of scene content.

In Chapter 5, we proceeded up a level of abstraction to express multiple grouping cues in a unified framework. Mid-level cues are ubiquitous and transcend individual object classes, yet can be leveraged effectively only in combination. We have presented a method to combine appearance, closure, and symmetry, and demonstrated the usefulness

of each cue. We have also demonstrated the effectiveness of using mid-level cues to resolve ambiguity with a limited budget of proposals, and shown that our model complements diversification techniques when a large number of proposals is affordable.

Finally in Chapter 6, we introduced a novel structured learning framework for perceptual grouping that learns how to combine multiple cues to generate a set of figure-ground region proposals. By adapting a loss function from MCL, we enable parametric energy functions to be trained to generate the best set of region proposals. Moreover, our framework supports an efficient diversification strategy that yields a diverse set of region proposals that competes favorably with the state of the art. Additionally, our learning formulation was developed specifically for bottom-up grouping, however the framework could be applied to other "multiple diverse" problems as well.

In summary, we have addressed the issues of complexity, variability, and learning in the following ways. Our work in symmetric part detection and multi-cue grouping combined mid-level cues of symmetry and closure and addressed the challenge of computational complexity with dynamic programming and parametric maxflow, respectively. This work also addressed the challenge of handling variability by adding invariance to important classes of variation, namely bending and tapering, and by integrating information from multiple grouping cues. Finally, our work in multi-cue grouping and learning to generate proposals addressed the challenge of learning to combine cues jointly over the full image, and developing a multiple-output learning framework that is suitable for bottom-up grouping. Quantitative results with respect to baselines demonstrate that we have taken encouraging steps forward, however a long path ahead still remains.

7.1 Limitations and future work

As we have mentioned, we have by no means reached our destination, but have only taken a few steps forward with many more to follow. Just as we outlined our steps forward at

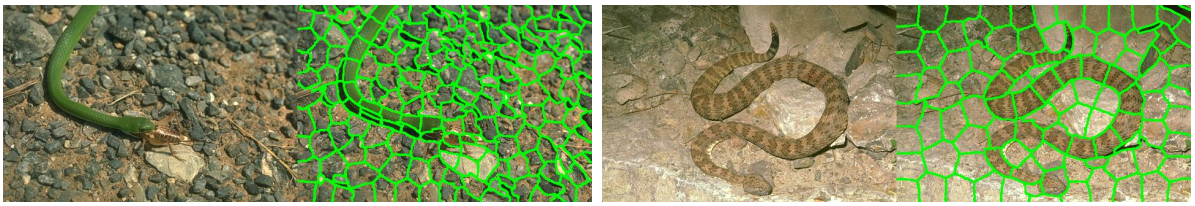


Figure 7.1: In addressing the challenge of obtaining superpixels that accurately model maximal discs, we compare image superpixels that yield correct discs (left), with a more difficult image that yields the correct discs only for areas of sufficient color contrast. Our method in Chapter 4 depends on the segmentation accuracy of superpixels as maximal discs.

the outset in Chapter 1, having the results behind us allows us now to identify current limitations, and outline possible steps to continue forward in future work.

Our work in Chapter 4 added invariance to bending and tapering to improve robustness, and thereby widened the domain of applicability. There are, however, other types of robustness to be addressed, such as occlusion. With symmetry comes the benefit of compositionality (of object parts), representing another direction in which to move forward. Aside from other types of symmetry like rotation and translation, medial symmetry composes a larger structure called the medial axis transform (MAT) that can serve as a powerful part-based representation of an object. The part-based approach to modelling objects has the advantage of being able to recognize the parts of an object that survive occlusion. Future work will thus address the problem of part grouping to yield part configurations whose relational information offers the discriminative power to prune a large database down to a small number of promising candidates.

There is an additional limitation of the superpixel grouping framework used in [58] and [65] that can be addressed. By grouping superpixels under the medial axis model, these methods assume that a symmetric part is composed from regions, each of which is compact in size, yet fully spans the part along its width, *i.e.* from the boundary on one side of the axis to the boundary on the other. While it is convenient to use existing superpixel algorithms to produce the required regions, superpixels do not always satisfy

Figure 7.2: The top few region proposals in an image with little background clutter (left), compared with the top few region proposals in an image with background clutter (right). The latter proposals undersegment the horse by including parts of the background fence. Looking closely at the T-junctions where the horse occludes the fence, however, shows that the proposed segment violates foreground-background regularities. This example suggests that more advanced features, such as T-junction regularities, could have helped to avoid undersegmenting proposals and improve precision.

the requirements above. Specifically, over- and undersegmentation errors that violate the requirements will limit the recall achievable by the framework (see Figure 7.1 for an example). Superpixels, however, remain a useful preprocessing step, and future work can try to contain the effects of superpixel errors within the medial axis framework. We have seen how region proposals successfully used diversification to handle bottom-up segmentation errors. One can thus follow the same strategy by formulating and solving the subproblem of producing "maximal disc" proposals. Another way is to take the contour-based approach, which is inherently free from region-based errors.

Another direction to pursue is more cue integration with other cues. While we followed this approach in Chapter 5 with non-symmetry cues, there are other types of symmetry that are worth exploiting. For example, detection of rotational and translation symmetry [57] is an active research area that explores regularities that are as ubiquitous as reflective symmetry. These approaches, however, are typically followed in isolation from other cues and evaluated on specialized datasets, and thus offer little or no cue integration. Overall, there is a long way to go in developing symmetry as a unified framework for segmentation and recognition. Continued progress, however, will yield many benefits.

In Chapter 5, we combined the cues of appearance, closure, and symmetry, and thereby widened the domain of applicability beyond those consisting purely of symmetric

shapes or shapes with closed boundaries. Therefore, incorporating more cues will further extend the domain of applicability and thereby improve robustness. For example in classic line drawings, various restrictions on local combinations of line types appearing at their junctions have been applied as a useful constraint for 3D interpretation (*e.g.*, Waltz labeling). Junction cues can thus be incorporated as cues for figure-ground labeling in real images of cluttered scenes. Of particular interest are T-junctions due to their simplicity and association with occluding boundaries. Generally, features can be designed and extracted at superpixel intersection points to measure various properties of junctions (see Figure 7.2 for an example).

Modelling cue combination as a trainable parameter allowed us to leverage the methodology of machine learning, however the formulation used a loss function that did not account for multiple output hypotheses. This is an open problem and an emerging research area known as "multiple diverse" problems, and spans areas as varied as natural language systems and assistive technologies. In the context of bottom-up grouping, there is no standard way to model such a loss function, however Chapter 6 proposed such a way.

In Chapter 6, we designed a loss function for parametric energy minimization for learning to group bottom-up. It was demonstrated that the learning framework was complementary to diversification, a technique that is crucial for recall. However, this was demonstrated only for a subset of the proposed diversification, in particular only up to what λ allowed us to model. In our case, we used the area feature ϕ_0 as the coefficient of λ to model diversity in object scale. In light of competing region proposal methods (like Selective Search) where diversification is employed in multiple variables (different color spaces and segmentation parameters), one can imagine a generalization of (1.1) to replace the scalar parameter λ with a vector parameter. Accordingly, one can explore the possibility of extending the parametric maxflow algorithm to minimize the resulting energy function for multiple values of the vector parameter.

Figure 7.3: We compare a dinner table that is homogeneously white and easy to group (left), with a dinner table that is multi-colored with objects placed on top that is difficult to group (right). One approach to this combinatorial problem is to continue to develop higher-order cues, like symmetry, that can potentially do more intelligent bottom-up grouping.

Apart from generalizing parametric maxflow to multiple parameters, one can also explore different ways of modelling λ. In our case we have used the area feature ϕ_0, however ϕ_0 need only satisfy mild mathematical constraints, and one could try to learn a feature ϕ_0 that maximizes recall. Generally speaking, diversification allows a method to adapt to different conditions by exhaustively testing different parameter values, with the drawback of dramatically lowering precision. Hence, if we can leverage the learning approach to diversify with better precision, we can significantly advance the approach of region proposals.

As a method for region proposals that uses diversification, our approach has the same limitations as any diversification technique that exhaustively explores possible groupings. As illustrated in Figure 7.3, objects that are highly heterogeneous are likely highly oversegmented, making it unlikely for the object to be proposed due to the number of segments that need to be correctly grouped. One way to view this problem is a lack of higher-order cues, like symmetry, that can perform bottom-up grouping of segments more accurately.

These are some possible directions to follow, however there are potentially many other promising directions. What is certain is that there is a long way to go in managing search complexity, improving robustness to input variability, and optimizing performance, before we can give computers the benefit of perceptual grouping.

Bibliography

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels. *Ecole Polytechnique Fédéral de Lausssanne (EPFL), Tech. Rep*, 2:3, 2010.

[2] B Alexe, T Deselaers, and V Ferrari. Measuring the objectness of image windows. *PAMI*, Jan 2012.

[3] S Alpert, M Galun, R Basri, and A Brandt. Image segmentation by probabilistic bottom-up aggregation and cue integration. *CVPR*, Jan 2007.

[4] P Arbeláez, M Maire, C Fowlkes, and J Malik. Contour detection and hierarchical image segmentation. *PAMI*, 33(5):898 – 916, 2011.

[5] Pablo Arbeláez, Jordi Pont-Tuset, Jonathan Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping. *CVPR*, 2014.

[6] S. Bagon, O. Brostovsky, M. Galun, and M. Irani. Detecting and sketching the common. In *CVPR*, 2010.

[7] K Barnard, P Duygulu, D Forsyth, N De Freitas, DM Blei, and MI Jordan. Matching words and pictures. *JMLR*, 3:1107–1135, 2003.

[8] I Biederman. Human image understanding: Recent research and a theory. *CVGIP*, Jan 1985.

[9] TO Binford. Visual perception by computer. *ICSC*, 1971.

[10] H Blum. A transformation for extracting new descriptors of shape. *Models for the perception of speech and visual form*, 19(5):362–380, 1967.

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

[12] M Brady and H Asada. Smoothed local symmetries and their implementation. *IJRR*, 3(3):36–61, 1984.

[13] J Canny. A computational approach to edge detection. *PAMI*, 8(6):679–697, Jan 1986.

[14] G. Carneiro, A. Chan, P. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *PAMI*, 2007.

[15] Joao Carreira and Cristian Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *PAMI*, 34(7):1312–1328, 2012.

[16] T Cham and R Cipolla. Geometric saliency of curve correspondences and grouping of symmetric contours. *ECCV*, Jan 1996.

[17] Tat-Jen Cham and Roberto Cipolla. Symmetry detection through local skewed symmetries. *IVC*, 13(5):439–450, 1995.

[18] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals for accurate object class detection. In *NIPS*, 2015.

[19] L Cohen and T Deschamps. Multiple contour finding and perceptual grouping as a set of energy minimizing paths. *EMMCVPR*, Jan 2001.

[20] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 24(5):603–619, 2002.

[21] J Connell and M Brady. Generating and generalizing models of visual objects. *Artificial Intelligence*, Jan 1987.

[22] D Crandall and D Huttenlocher. Weakly supervised learning of part-based spatial models for visual object recognition. *ECCV*, pages 16–29, 2006.

[23] James Crowley and Alice Parker. A representation for shape based on peaks and ridges in the difference of low-pass transform. *PAMI*, 6(2):156–170, 1984.

[24] F Demirci, A Shokoufandeh, and S Dickinson. Skeletal shape abstraction from examples. *PAMI*, 31(5):944, 2009.

[25] S Dickinson. The evolution of object categorization and the challenge of image abstraction. *Object categorization: computer and human vision perspectives*, pages 1–37, Jan 2009.

[26] S Dickinson, A Levinshtein, P Sala, and C Sminchisescu. The role of mid-level shape priors in perceptual grouping and image abstraction. *Shape Perception in Human and Computer Vision: An Interdisciplinary Perspective*, Jan 2013.

[27] P Duygulu, K Barnard, J De Freitas, and D Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. *ECCV*, pages 349–354, 2002.

[28] James Elder and Steven Zucker. Computing contour closure. *ECCV*, pages 399–412, 1996.

[29] I Endres and D Hoiem. Category independent object proposals. *ECCV*, Jan 2010.

[30] Francisco Estrada, Allan Jepson, and Chakra Chennubhotla. Spectral embedding and min cut for image segmentation. pages 1–10, 2004.

[31] M Everingham, L V Gool, C Williams, J Winn, and A Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010.

[32] P Felzenswalb and D Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, 2004.

[33] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 32(9), 2010.

[34] P Felzenszwalb and D McAllester. A min-cover approach for finding salient curves. *WPOCV*, 2006.

[35] R Fergus, P Perona, and A Zisserman. Weakly supervised scale-invariant learning of models for visual recognition. *IJCV*, 71(3):273–303, 2007.

[36] V Ferrari, T Tuytelaars, and L Van Gool. Object detection by contour segment networks. *ECCV*, pages 14–28, 2006.

[37] Vittorio Ferrari, Loic Fevrier, Frederic Jurie, and Cordelia Schmid. Groups of adjacent contour segments for object detection. *PAMI*, pages 1–16, Nov 2008.

[38] Vittorio Ferrari, Frederic Jurie, and Cordelia Schmid. From images to shape models for object detection. *IJCV*, 87(3):284–303, May 2010.

[39] S Fidler, M Boben, and A Leonardis. Learning a hierarchical compositional shape vocabulary for multi-class object representation. *ArXiv:1408.5516*, Jan 2014.

[40] Sanja Fidler, Roozbeh Mottaghi, Alan Yuille, and Raquel Urtasun. Bottom-up segmentation for top-down detection. *CVPR*, pages 3294–3301, 2013.

[41] M Fromer and A Globerson. An lp view of the m-best map problem. *NIPS*, Jan 2009.

[42] R Girshick, J Donahue, T Darrell, and J Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CVPR*, Jan 2014.

[43] G Guy and G Medioni. Inferring global perceptual contours from local features. *CVPR*, Jan 1993.

[44] A Guzman-Rivera, P Kohli, Dhruv Batra, and Rob Rutenbar. Efficiently enforcing diversity in multi-output structured prediction. *AISTATS*, Jan 2014.

[45] Abner Guzman-Rivera, Dhruv Batra, and Pushmeet Kohli. Multiple choice learning: Learning to produce multiple structured outputs. *NIPS*, pages 1799–1807, 2012.

[46] A Humayun, F Li, and J Rehg. Rigor: Reusing inference in graph cuts for generating object regions. *CVPR*, Jan 2014.

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

[48] M Jamieson, A Fazly, S Stevenson, S Dickinson, and S Wachsmuth. Using language to learn structured appearance models for image annotation. *PAMI*, 32(1):148–164, 2010.

[49] Ian Jermyn and Hiroshi Ishikawa. Globally optimal regions and boundaries as minimum ratio weight cycles. *PAMI*, 23(10):1075–1088, 2001.

[50] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *IJCV*, 1(4):321–331, 1988.

[51] J Kim and K Grauman. Boundary preserving dense local regions. *CVPR*, Jan 2011.

[52] K Koffka. Principles of gestalt psychology. *gestalttheory.net*, Jan 1935.

[53] V Kolmogorov, Y Boykov, and C Rother. Applications of parametric maxflow in computer vision. *ICCV*, 8, 2007.

[54] Vladimir Kolmogorov and Ramin Zabin. What energy functions can be minimized via graph cuts? *PAMI*, 26(2):147–159, 2004.

[55] P Krähenbühl and V Koltun. Geodesic object proposals. *ECCV*, Jan 2014.

[56] P Krähenbühl and V Koltun. Learning to propose objects. *CVPR*, 2015.

[57] Seungkyu Lee and Yanxi Liu. Curved glide-reflection symmetry detection. *PAMI*, 34(2):266–278, 2012.

[58] T Lee, S Fidler, and S Dickinson. Detecting curved symmetric parts using a deformable disc model. *ICCV*, 2013.

[59] T Lee, S Fidler, and S Dickinson. Multi-cue mid-level grouping. *ACCV*, 2014.

[60] T Lee, S Fidler, and S Dickinson. Learning to combine mid-level cues for object proposal generation. *ICCV*, 2015.

[61] T Lee, S Fidler, A Levinshtein, and S Dickinson. Learning categorical shape from captioned images. *CRV*, 2012.

[62] Y Lee and K Grauman. Shape discovery from unlabeled image collections. *CVPR*, 2009.

[63] M Leordeanu, M Hebert, and R Sukthankar. Beyond local appearance: Category recognition from pairwise interactions of simple features. *CVPR*, 2007.

[64] Thomas Leung and Jitendra Malik. Contour continuity in region based image segmentation. *ECCV*, pages 544–559, 1998.

[65] A Levinshtein, S Dickinson, and C Sminchisescu. Multiscale symmetric part detection and grouping. *ICCV*, 2009.

[66] A Levinshtein, C Sminchisescu, and S Dickinson. Optimal contour closure by superpixel grouping. *ECCV*, pages 480–493, 2010.

[67] A Levinshtein, C Sminchisescu, and S Dickinson. Optimal image and video closure by superpixel grouping. *IJCV*, 2012.

[68] A Levinshtein, C Sminchisescu, and S Dickinson. Multiscale symmetric part detection and grouping. *IJCV*, 104(2):117–134, 2013.

[69] A Levinshtein, A Stere, KN Kutulakos, DJ Fleet, SJ Dickinson, and K Siddiqi. Turbopixels: Fast superpixels using geometric flows. *PAMI*, pages 2290–2297, 2009.

[70] M Leyton. Symmetry, causality, mind. *MIT Press*, 1992.

[71] T Lin, M Maire, S Belongie, J Hays, P Perona, D Ramanan, P Dollar, and C Zitnick. Microsoft coco: Common objects in context. *ECCV*, 2014.

[72] T Lindeberg. Edge detection and ridge detection with automatic scale selection. *IJCV*, Jan 1998.

[73] T Lindeberg and L Bretzner. Real-time scale selection in hybrid multi-scale representations. *Scale Space Methods in Computer Vision*, Jan 2003.

[74] T Liu, D Geiger, and A Yuille. Segmenting by seeking the symmetry axis. *ICPR*, Jan 1998.

[75] D Lowe. Perceptual organization and visual recognition. 1985.

[76] D.G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[77] G Loy and J.O Eklundh. Detecting symmetry and symmetric constellations of features. *ECCV*, 2006.

[78] D Macrini, S Dickinson, D Fleet, and K Siddiqi. Object categorization using bone graphs. *CVIU*, Jan 2011.

[79] Diego Macrini, Sven Dickinson, David Fleet, and Kaleem Siddiqi. Bone graphs: Medial shape parsing and abstraction. *CVIU*, Apr 2011.

[80] M Maire, P Arbeláez, C Fowlkes, and J Malik. Using contours to detect and localize junctions in natural images. *ICCV*, pages 1–8, Apr 2008.

[81] D Martin, C Fowlkes, and J Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, pages 1–20, Jul 2004.

[82] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *ICCV*, Jan 2001.

[83] K Mikolajczyk and C Schmid. An affine invariant interest point detector. *ECCV*, 2002.

[84] Rakesh Mohan and Ramakant Nevatia. Perceptual organization for scene segmentation and description. *PAMI*, 14(6):616–635, 1992.

[85] F. Monay and D. Gatica-Perez. Modeling semantic aspects for cross-media image indexing. *PAMI*, 2007.

[86] M Narayanan and B Kimia. Bottom-up perceptual organization of images into object part hypotheses. *ECCV*, Jan 2012.

[87] S Nowozin and C Lampert. Structured learning and prediction in computer vision. *Foundations and Trends in Computer Graphics and Vision*, 2011.

[88] N Payet and S Todorovic. From a set of shapes to object discovery. *ECCV*, pages 57–70, 2010.

[89] M Pelillo, K Siddiqi, and S Zucker. Matching hierarchical structures using association graphs. *PAMI*, 21(11):1105–1120, Jan 1999.

[90] AP Pentland. Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28(3):293–331, 1986.

[91] J Ponce. On characterizing ribbons and finding skewed symmetries. *CVGIP*, 52(3):328–340, 1990.

[92] T. Quack, V. Ferrari, B. Leibe, and L. Van Gool. Efficient mining of frequent and distinctive feature configurations. In *ICCV*, 2007.

[93] A. Quattoni, M. Collins, and T. Darrell. Learning visual representations using images with captions. In *CVPR*, 2007.

[94] P Rantalankila, J Kannala, and E Rahtu. Generating object segmentation proposals using global and local search. *CVPR*, Jan 2014.

[95] X Ren and J Malik. Learning a classification model for segmentation. *ICCV*, 2003.

[96] Xiaofeng Ren, Charless Fowlkes, and Jitendra Malik. Cue integration for figure/ground labeling. *NIPS*, 2005.

[97] P Saint-Marc, H Rom, and G Medioni. B-spline contour representation and symmetry detection. *PAMI*, 15(11):1191–1197, 1993.

[98] P Sala and S Dickinson. Contour grouping and abstraction using simple part models. *ECCV*, pages 603–616, 2010.

[99] A Schwing, S Fidler, M Pollefeys, and R Urtasun. Box in the box: Joint 3d layout and object reasoning from single images. *ICCV*, Jan 2013.

[100] S Sclaroff and L Liu. Deformable shape detection and description via model-based region grouping. *PAMI*, Jan 2001.

[101] T Sebastian, P Klein, and B Kimia. Recognition of shapes by editing their shock graphs. *PAMI*, 26(5):550–571, Jul 2004.

[102] Eitan Sharon, Achi Brandt, and Ronen Basri. Fast multiscale image segmentation. 1:70–77, 2000.

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

[104] A Shokoufandeh, L Bretzner, D Macrini, M Fatih Demirci, C Jönsson, and S Dickinson. The representation and matching of categorical shape. *CVIU*, 103(2):139–154, 2006.

[105] A Shokoufandeh, I Marsic, and SJ Dickinson. View-based object recognition using saliency maps. *IVC*, 17(5-6):445–460, 1999.

[106] K Siddiqi and S Pizer. Medial representations: mathematics, algorithms and applications. *Springer*, Jan 2008.

[107] K Siddiqi, A Shokoufandeh, S Dickinson, and S Zucker. Shock graphs and shape matching. *IJCV*, 35(1):13–32, 1999.

[108] F Solina and R Bajcsy. Recovery of parametric models from range images: The case for superquadrics with global deformations. *PAMI*, 12(2):131–147, 1990.

[109] J Stahl and S Wang. Globally optimal grouping for symmetric closed boundaries by combining boundary and region information. *PAMI*, 30(3):395–411, 2008.

[110] M Stark, M Goesele, and B Schiele. A shape-based object class model for knowledge transfer. *Computer Vision, 2009 IEEE 12th International Conference on*, pages 373–380, 2009.

[111] M Szummer, P Kohli, and D Hoiem. Learning crfs using graph cuts. *ECCV*, Jan 2008.

[112] S. Todorovic and N. Ahuja. Unsupervised category modeling, recognition, and segmentation in images. *PAMI*, 2009.

[113] I Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. *JMLR*, Jan 2005.

[114] S Tsogkas and I Kokkinos. Learning-based symmetry detection in natural images. *ECCV*, Jan 2012.

[115] C Tyler. Human symmetry perception and its computational analysis. *Taylor & Francis*, Jan 2002.

[116] Jasper Uijlings, Koen van de Sande, Theo Gevers, and Arnold Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013.

[117] Shimon Ullman and Amnon Sha'ashua. Structural saliency: The detection of globally salient structures using a locally connected network. 1988.

[118] Luc Vincent and Pierre Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *PAMI*, 13(6):583–598, 1991.

[119] M Wertheimer. Untersuchungen zur lehre von der gestalt. ii. *Psychological Research*, Jan 1923.

[120] M Wertheimer. Laws of organization in perceptual forms. *Source Book of Gestalt Psychology*, Jan 1938.

[121] L Williams and D Jacobs. Stochastic completion fields: A neural model of illusory contour shape and salience. *Neural Computation*, Jan 1997.

[122] AP Witkin and JM Tenenbaum. On the role of structure in vision. *Human and Machine Vision*, pages 481–543, 1983.

[123] V Yanulevskaya, J Uijlings, and N Sebe. Learning to group objects. *CVPR*, Jan 2014.

[124] Antti Ylä-Jääski and Frank Ade. Grouping symmetrical structures for object segmentation and description. *CVIU*, 63(3):399–417, 1996.

[125] Q Zhu, G Song, and J Shi. Untangling cycles for contour grouping. *ICCV*, pages 1–8, 2007.

[126] Song Zhu and Alan Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *PAMI*, 18(9):884–900, 1996.

[127] C Zitnick and P Dollár. Edge boxes: Locating object proposals from edges. *ECCV*, Jan 2014.