Chapter 1
The Role of Mid-Level Shape Priors in Perceptual Grouping and Image Abstraction

Sven J. Dickinson, Alex Levinshtein, Pablo Sala, and Cristian Sminchisescu

1.1 Introduction

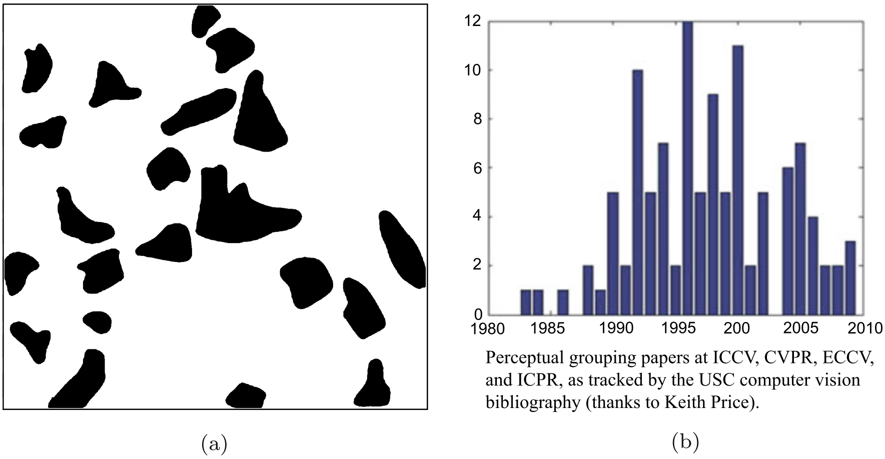
Have a look at the image in Fig. 1.1(a) (taken from [29]) and don't read any further until you recognize the object(s) in the scene. For most people, the image of a horse and rider quickly emerges. This is remarkable considering that each individual black fragment is practically meaningless in terms of its indexing power to suggest a horse or rider (or *any* object, for that matter). Only when the fragments are *grouped* together and *abstracted* to yield meaningful parts and relations do the objects begin to emerge. Moreover, these grouping and abstraction processes are primarily bottom-up, and do not require a priori knowledge of scene content. Nobody told you what object to look for, and you certainly didn't run through tens of thousands of category detectors to decide that it was a horse and rider and not a table and chair. Somehow, your visual system grouped the fragments to form a set of abstract parts, then grouped those parts into larger configurations, then "queried" your visual memory for similar configurations, and only then used a priori knowledge of a promising candidate to "detect", i.e., verify, the object.

Perceptual grouping is a critical function in the human visual system, offering a powerful heuristic for grouping together causally related image features in support of both figure-ground segmentation and 3-D inference. In the mid-to-late 1990s, perceptual grouping was a thriving subcommunity in computer vision, as illustrated in Fig. 1.1(b). However, over the past 10 years, there's been a steady decline in the number of perceptual grouping papers appearing in the computer vision community's main conferences. The reason for this is the reformulation of object recogni-

S.J. Dickinson (✉) · A. Levinshtein · P. Sala
Department of Computer Science, University of Toronto, Toronto, Canada
e-mail: sven@cs.toronto.edu

C. Sminchisescu
Institute of Numerical Simulation, University of Bonn, Bonn, Germany

S.J. Dickinson, Z. Pizlo (eds.), *Shape Perception in Human and Computer Vision*,
Advances in Computer Vision and Pattern Recognition,
DOI 10.1007/978-1-4471-5195-1_1, © Springer-Verlag London 2013



Perceptual grouping papers at ICCV, CVPR, ECCV, and ICPR, as tracked by the USC computer vision bibliography (thanks to Keith Price).

Fig. 1.1 (**a**) An illustration of the power of perceptual grouping. Individually, the black, amorphous blobs carry very little information. However, when grouped into parts, the emergent part structure allows the scene (horse and rider) to be quickly interpreted *without* any a priori knowledge of scene content (figure reproduced with kind permission from Teachers College Press, Columbia University, New York: *A gestalt completion test: a study of a cross section of intellect*, 1931, Roy F. Street, p. 55, Fig. 8); (**b**) The rise and fall of perceptual grouping. Tracking perceptual grouping papers in the computer vision community's four main conferences indicates a growing interest in perceptual grouping, peaking in the late 1990s. However, since then, interest in this critically important problem has waned

tion, historically cast as the problem of recognizing an object from a large database, as a detection problem, cast as the search for a particular target object.

The classical formulation of the object recognition problem, which defined the mainstream from the mid-1960s through to the late-1990s, was the recognition of an unexpected object from a database of objects. As illustrated in Fig. 1.2, the feature extraction process began by extracting categorical or generic features, as the recognition community aspired to recognize categories, not exemplars. As far back as the seminal work of Roberts [23] in the mid-1960s, the recognition community understood that across the exemplars that belong to a category, shape is a more invariant property than appearance. As a result, the majority of recognition systems from the mid-1960s to the late 1990s attempted to extract shape features, typically beginning with the extraction of edges, for at occluding boundaries and surface discontinuities, edges capture shape information. However, unlike today's distinctive local image features, e.g., SIFT [20], a local edgel carries very little information with which to index into a database of objects in an attempt to select a small number of promising object models that might account for the edgels.

The need for perceptual grouping in these early systems was critical, for only when the edgels were grouped into longer contours, perhaps parsed at high-curvature points, and grouped with other causally related contours, did distinctive indexing features emerge. Lowe's thesis [21] was the first to introduce computational models of perceptual grouping processes, e.g., proximity, collinearity, and

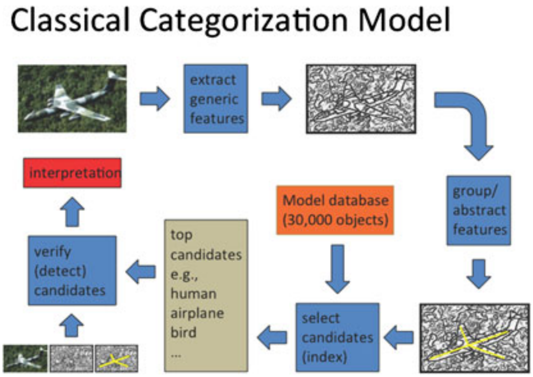

Classical Categorization Model
extract generic features
group/ abstract features
Model database (30,000 objects)
select candidates (index)
top candidates e.g., human airplane bird ...
verify (detect) candidates
interpretation

Fig. 1.2 In the classical recognition model, the desire to extract shape features, considered more generic than appearance, began with edge detection. Because edgels were not discriminative, they were perceptually grouped and abstracted to form distinctive indexing structures that could prune a large database of objects down to a small number of promising candidates. (figure reproduced with kind permission from Springer Science+Business Media: *Proceedings, 4th Mexican Conference on Pattern Recognition (MCPR)*, Perceptual Grouping using Superpixels, 2012, S. Dickinson, A. Levinshtein, and C. Sminchisescu, p. 14, Fig. 1)

parallelism, derived from image statistics. By grouping contour features into more distinctive groups (in Lowe's case, proximity followed by collinearity followed by parallelism), more discriminating indexing (using parallel lines instead of, say, triples of corners [11]) was possible. The more that features were grouped, perhaps first into parts and then into multipart groups [8, 9], the more powerful the resulting indexing structure and the fewer candidates that ultimately needed to be verified. Each candidate was verified, yielding a score (typically reflecting the degree to which a model could be aligned with image features), and the top-scoring candidate, if sufficiently strong, gave the final interpretation.

The formulation of object recognition as the detection of a specific target object has dominated the recognition community over the past 10 years. As illustrated in Fig. 1.3(top) and working backwards from the verification module, instead of having to verify a number of candidate object hypotheses, the detection problem identifies only a single hypothesis that needs to be verified (or detected). This, in turn, means that the indexing step, in which a large database of candidate objects is pruned down to a small set of candidates for verification, is superfluous, as the database effectively has a single object (target). Continuing to work our way backwards, as illustrated in Fig. 1.3(middle), if discriminative indexing features are not required to select promising candidates, the perceptual grouping stage is also superfluous. Instead, as illustrated in Fig. 1.3(bottom), the detector, i.e., verification, can be applied directly to the edgels, e.g., [6], to give the final score, thereby short-circuiting the entire perceptual grouping process.

The existence of an object detector, representing a strong shape prior, eliminates the need for perceptual grouping, representing a much weaker, domain-independent shape prior. However, as the categorization community moves from single object

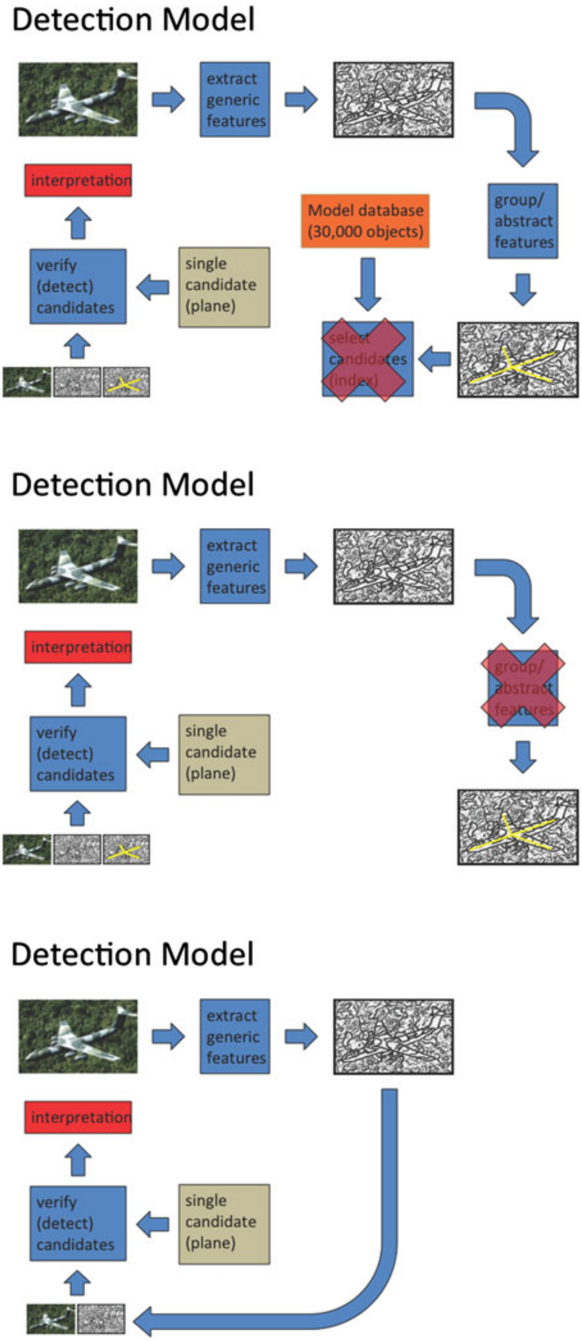


Fig. 1.3 The classical formulation of object recognition from a large database has given way to a more recent formulation of object recognition as target detection: (*top*) rather than verifying a number of candidates, the target candidate is known, rendering the process of indexing (or model selection) obsolete. (*Middle*) Without the need for domain-independent recovery, grouping, and abstraction of structure in order to prune a large database down to a small number of promising candidates, perceptual grouping is unnecessary. (*Bottom*) As a result, verification (detection) can be applied directly to the ungrouped, low-level edge features. (Figure reproduced with kind permission from Springer Science+Business Media: *Proceedings, 4th Mexican Conference on Pattern Recognition (MCPR)*, Perceptual Grouping using Superpixels, 2012, S. Dickinson, A. Levinshtein, and C. Sminchisescu, p. 14, Fig. 1)

detection back to recognition from large databases, detection methods, typically formulated as template matching (or "sliding windows"), simply won't scale, and a linear search through thousands of templates is intractable, especially when an object can be viewed arbitrarily, it can articulate, and it can undergo significant within-class shape deformation. Verification (or detection) must be highly sublinear in the size of the database, demanding that discriminative indexing features be recovered *without knowledge of which object is being imaged*. Such domain-independent, bottom-up perceptual grouping is essential in the absence of an object prior.

In this chapter, we briefly review our recent progress on three classical problems in perceptual grouping using three mid-level shape priors: symmetry, closure, and parts. We begin by describing a framework that first groups superpixels into symmetric parts, and then groups the symmetric parts into multipart structures [13]. Symmetry has played a prominent role in shape modeling for object recognition since the 2-D medial axis transform (MAT) of Blum [2] and the 3-D generalized cylinder (GC) of Binford [1]. By detecting a set of symmetric parts and their attachments from a cluttered image of real objects, we recover a powerful shape index that can serve to prune a large database of objects down to a small number of promising candidates. Next, we address the classical problem of contour closure, i.e., finding a cycle of edgels in the image that separates figure from ground. We describe a framework that looks for groups of superpixels whose collective boundary has strong edgel support in the image [14, 15]. The resulting shape boundary, or silhouette, can yield a structured, parts-based representation, e.g., [27], that can also be used to prune a large database down to a small number of promising candidates. Finally, we use a vocabulary of simple shape parts (which, in turn, can be used to construct an infinite number of objects) to not only guide the perceptual grouping of superpixels into regions representing parts, but use the part vocabulary to regularize, or *abstract*, the shapes of the regions.

1.2 Symmetric Part Detection and Grouping

In [13], we introduced a novel approach to recovering the symmetric part structure of an object from a cluttered image, as outlined in Fig. 1.4. Drawing on the principle that a skeleton is defined as the locus of *medial points*, i.e., centers of maximally inscribed disks, we first hypothesize a sparse set of medial points at multiple scales by segmenting the image (Fig. 1.4(a)) into compact superpixels at different superpixel resolutions [17] (Fig. 1.4(b)). Superpixels are adequate for this task, balancing a data-driven component that's attracted to shape boundaries while maintaining a high degree of compactness. The superpixels (medial point hypotheses) at each scale are linked into a graph, with edges adjoining adjacent superpixels. Each edge is assigned an affinity that reflects the degree to which two adjacent superpixels represent medial points belonging to the same symmetric part (medial branch) (Fig. 1.4(c)). The affinities are learned from a set of training images whose symmetric parts have been manually identified. A standard graph-based segmentation

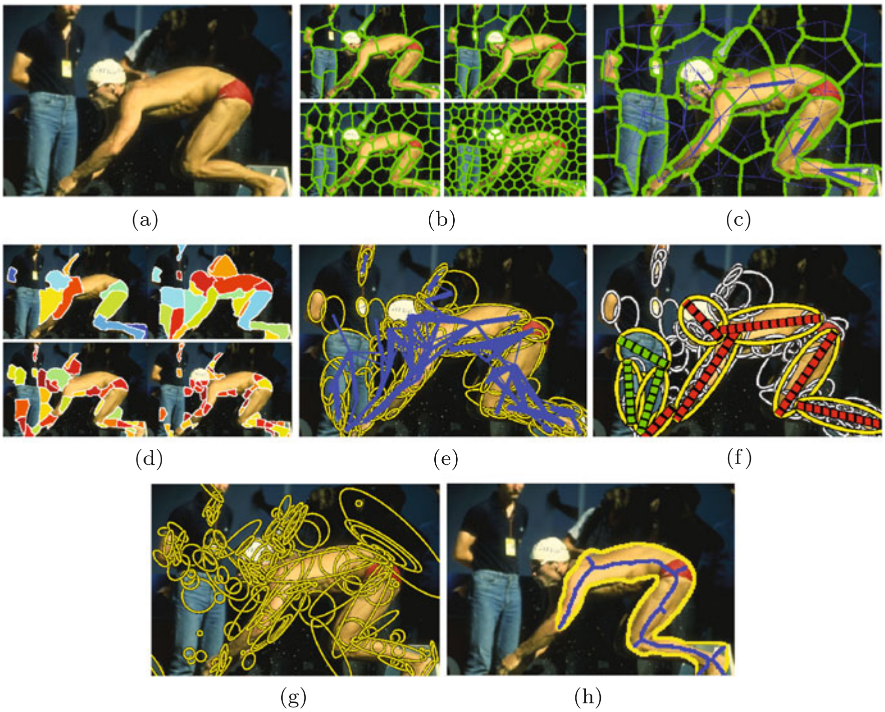


Fig. 1.4 Overview of our approach for multiscale symmetric part detection and grouping: (**a**) original image; (**b**) set of multiscale superpixel segmentations (different superpixel resolutions); (**c**) the graph of affinities shown for one scale (superpixel resolution); (**d**) the set of regularized symmetric parts extracted from all scales through a standard graph-based segmentation algorithm; (**e**) the graph of affinities between nearby symmetric parts (all scales); (**f**) the most prominent part clusters extracted from a standard graph-based segmentation algorithm, with abstracted symmetry axes overlaid onto the abstracted parts; (**g**) in contrast, a Laplacian-based multiscale blob and ridge decomposition, such as that computed by [19], shown, yields many false positive and false negative parts; (**h**) in contrast, classical skeletonization algorithms require a closed contour which, for real images, must be approximated by a region boundary. In this case, the parameters of the N-cuts algorithm [26] were tuned to give the best region (maximal size without region undersegmentation) for the swimmer. A standard medial axis extraction algorithm applied to the smoothed silhouette produces a skeleton (shown in *blue*) that contains spurious branches, branch instability, and poor part delineation. (Figure reproduced with kind permission from Springer Science+Business Media: *Proceedings, 4th Mexican Conference on Pattern Recognition (MCPR)*, Perceptual Grouping using Superpixels, 2012, S. Dickinson, A. Levinshtein, and C. Sminchisescu, p. 17, Fig. 2)

algorithm applied to each scale yields a set of superpixel clusters which, in turn, yield a set of regularized symmetric parts (Fig. 1.4(d)).

In the second phase of our approach, we address the problem of perceptually grouping symmetric parts arising in the first phase. Like in any grouping problem, our goal is to identify sets of parts that are causally related, i.e., unlikely to co-occur by accident. Again, we adopt a graph-based approach in which the set of symmetric parts across all scales are connected in a graph, with edges adjoining parts in close spatial proximity (Fig. 1.4(e)). Each edge is assigned an affinity, this time reflecting the degree to which two nearby parts are believed to be physically attached. Like in the first phase, the associated, higher granularity affinities are learned from the regularities of attached symmetric parts identified in training data. A graph segmentation yields a set of part clusters, each representing a set of regularized symmetric elements and their hypothesized attachments (Fig. 1.4(f)).

Our approach offers clear advantages over competing approaches. For example, classical multiscale blob and ridge detectors, such as [19] (Fig. 1.4(g)), yield many spurious parts, a challenging form of noise for any graph-based indexing or matching strategy. And even if an opportunistic setting of a region segmenter's parameters yields a decent object silhouette (Fig. 1.4(h)), the resulting skeleton may exhibit spurious branches and may fail to clearly delineate the part structure. From a cluttered image, our two-phase approach recovers, abstracts, and groups a set of medial branches into an approximation to an object's skeletal part structure, enabling the application of skeleton-based categorization systems to more realistic imagery. Details of the approach can be found in [13].

Some qualitative results are shown in Fig. 1.5. Proceeding left to right, top to bottom, we see excellent part recovery and grouping for the starfish, the plane, the windmill, and the runner, respectively. In the case of the windmill, a second, singleton cluster, representing the entire body of the human, is recovered; however, the distant windmills are not recovered, for their scale is smaller than the smallest superpixel scale. The final two figures represent failure modes. In the case of the lizard, the curved symmetric tail is oversegmented into piecewise linear symmetric parts. In the case of the lake scene, the symmetric parts making up the horizon tree line are incorrectly grouped with the dock structure due to a lack of apparent occlusion boundary between the dock structure and the tree line parts.

1.3 Contour Closure

In this section, we review our framework for efficiently searching for optimal contour closure; details can be found in [14, 15]. Figure 1.6 illustrates an overview of our approach to computing contour closure. Given an image of extracted contours (Fig. 1.6(a)), we begin by restricting contour closures to pass along boundaries of superpixels computed over the contour image (Fig. 1.6(b)). In this way, our first contribution is to reformulate the problem of searching for cycles of contours as the problem of searching for a subset of superpixels whose collective boundary has strong contour support in the contour image; the assumption we make is that those salient contours that define the boundary of the object (our target closure) will align well with superpixel boundaries. However, while a cycle of contours represents a single contour closure, our reformulation requires a mechanism to encourage superpixel subsets that are spatially coherent.

Spatial coherence is an inherent property of a cost function that computes the ratio of perimeter to area. We modify the ratio cost function of Stahl and Wang [28]

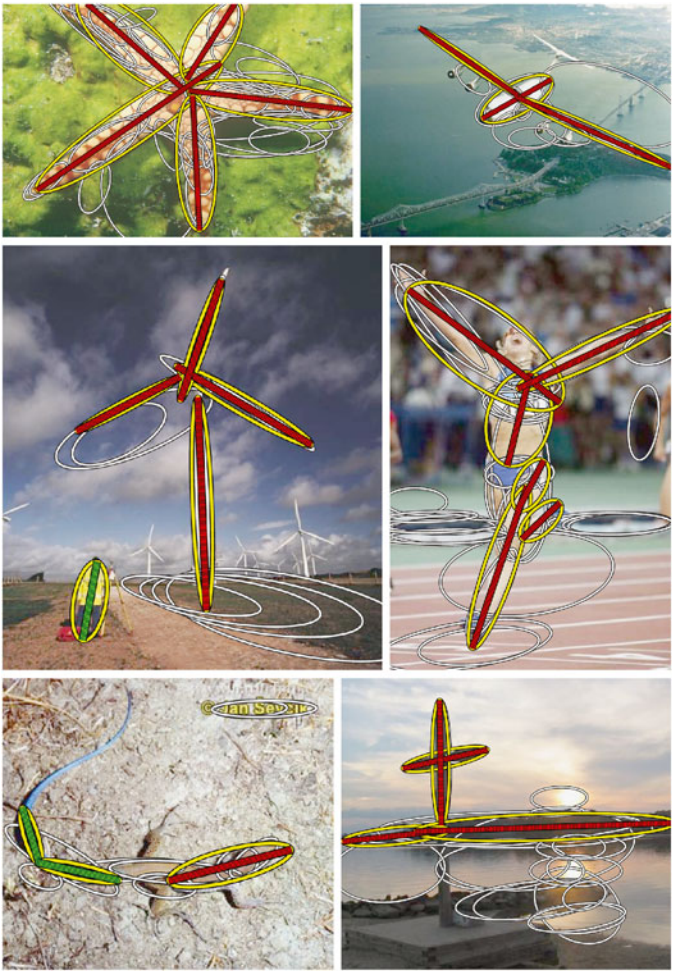
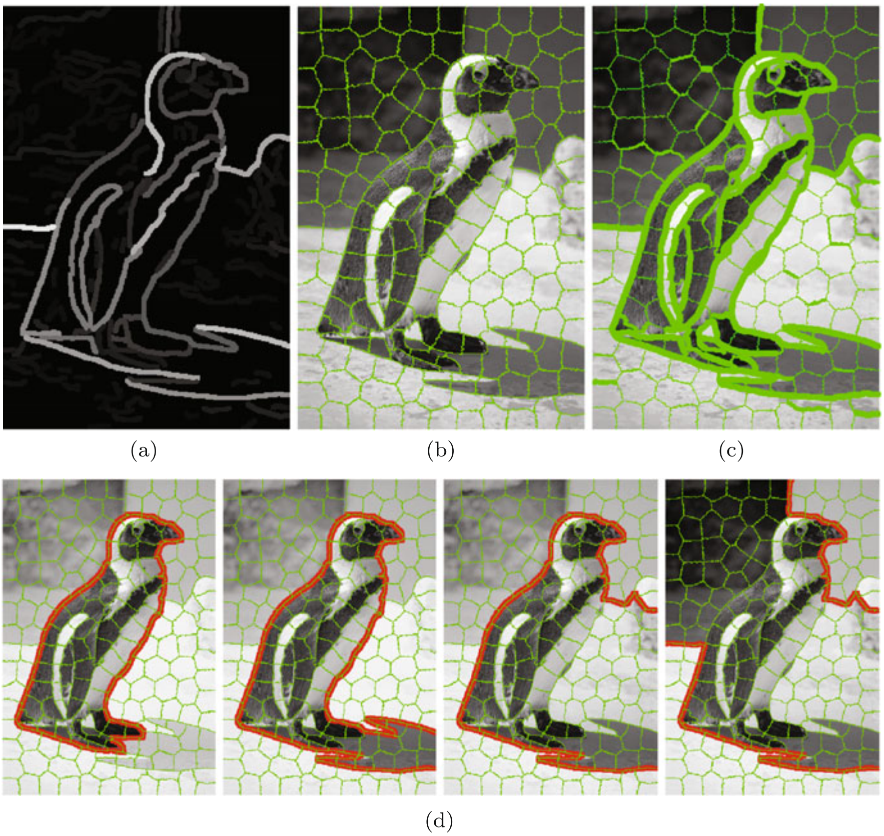


Fig. 1.5 Detected medial parts and their clusters. Parts with the same color axis have been grouped together (through high attachment affinities) and are hypothesized to belong to the same object. (Figure reproduced with kind permission from Springer Science+Business Media: *Proceedings, 4th Mexican Conference on Pattern Recognition (MCPR)*, Perceptual Grouping using Superpixels, 2012, S. Dickinson, A. Levinshtein, and C. Sminchisescu, p. 18, Fig. 3)

to operate on superpixels rather than contours, and extend it to yield a cost function that: (1) promotes spatially coherent selections of superpixels; (2) favors larger closures over smaller closures; and (3) introduces a novel, learned gap function that accounts for how much agreement there is between the boundary of the selection and the contours in the image. The third property adds cost as the number and sizes of gaps between contours increase. Given a superpixel boundary fragment (e.g., a side of a superpixel) representing a hypothesized closure component, we assign a gap cost that's a function of the proximity of nearby image contours, their strength,


(a) (b) (c)

(d)

Fig. 1.6 Overview of our approach for image closure: (**a**) contour image: while we take as input only this contour image, we will overlay the original image in the subsequent figures to ease visualization; (**b**) superpixel segmentation of contour image, in which superpixel resolution is chosen to ensure that target boundaries are reasonably well approximated by superpixel boundaries; (**c**) a novel, learned measure of gap reflects the extent to which the superpixel boundary is supported by evidence of a real image contour (line thickness corresponds to the amount of agreement between superpixel boundaries and image contours); (**d**) our cost function can be globally optimized to yield the largest set of superpixels bounded by contours that have the least gaps. In this case the solutions, in increasing cost (decreasing quality), are organized left to right. (Figure reproduced with kind permission from Springer Science+Business Media: *Proceedings, 4th Mexican Conference on Pattern Recognition (MCPR)*, Perceptual Grouping using Superpixels, 2012, S. Dickinson, A. Levinshtein, and C. Sminchisescu, p. 19, Fig. 4)

and their orientation (Fig. 1.6(c)). It is in this third property that our superpixel reformulation plays a second important role—by providing an appropriate scope of contour over which our gap analysis can be conducted.

In our third contribution, the two components of our cost function, i.e., area and gap, are combined in a simple ratio that can be efficiently optimized using parametric maxflow [12] to yield the global optimum. The optimal solution yields the

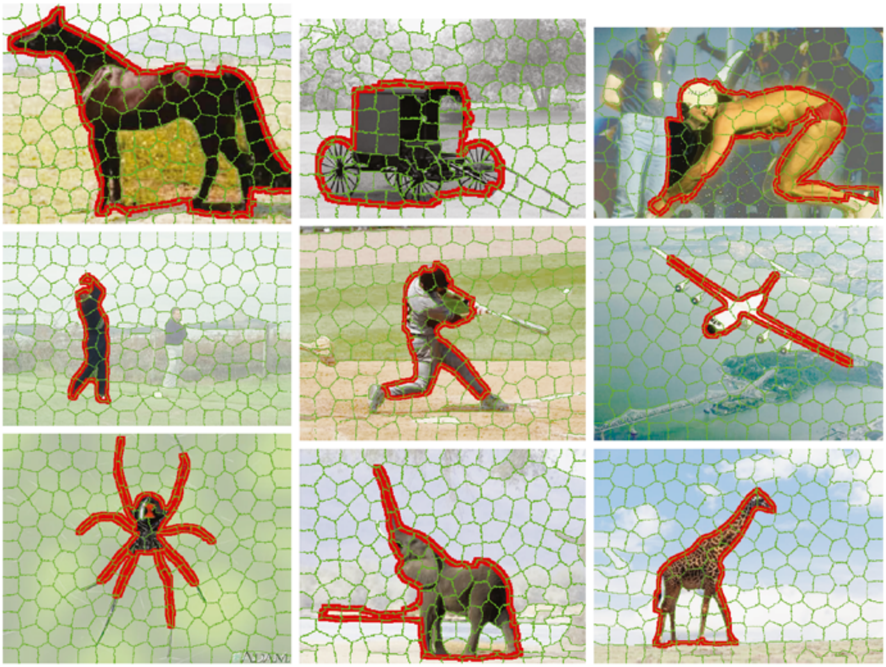


Fig. 1.7 Example results of superpixel closure. (Figure reproduced with kind permission from Springer Science+Business Media: *Proceedings, 4th Mexican Conference on Pattern Recognition (MCPR)*, Perceptual Grouping using Superpixels, 2012, S. Dickinson, A. Levinshtein, and C. Sminchisescu, p. 20, Fig. 5)

largest set of superpixels bounded by contours that have the least gaps (Fig. 1.6(d)). Moreover, parametric maxflow can be used to yield the top k solutions (see [4], for example). In an object recognition setting, generating a small set of such solutions can be thought of as generating a small set of promising shape hypotheses which, through an indexing process, could invoke candidate models that could be verified (detected). The use of such multiscale hypotheses was shown to facilitate state-of-the-art object recognition in images [18].

In Fig. 1.7, we illustrate results of our superpixel closure (SC) method. In the case of the carriage, swimmer, plane, golfer, baseball player, plane, and spider, we see that the algorithm nearly correctly segments figure from background, and is able to capture the deep concavities of the object, which is particularly visible with the spider. In the case of the horse, elephant, and giraffe, we see evidence of undersegmentation due to the properties of the objective function that we're optimizing. In each case, there are false boundaries (e.g., horizon) that can increase the area of the figure without introducing additional gap. In other words, if the algorithm can follow a gap-free contour that yields a larger area, e.g., following the contour between ground and sky in the giraffe image, it will do so, yielding a bias towards compact objects.

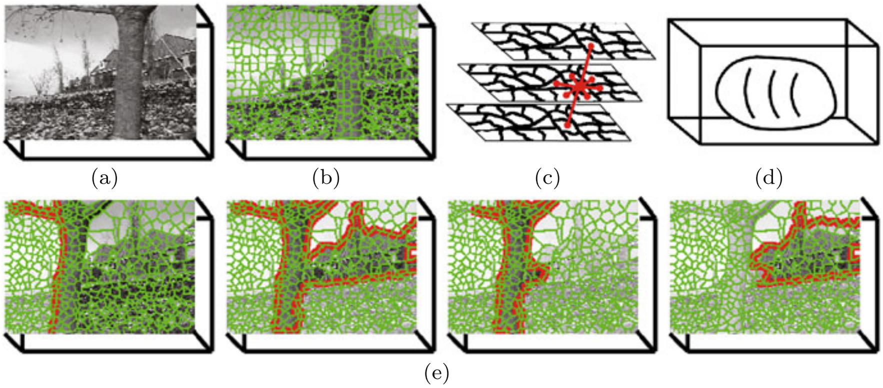


Fig. 1.8 Overview of our approach for spatiotemporal closure. (**a**) Spatiotemporal volume; (**b**) spatiotemporal superpixels; (**c**) superpixel graph with edges encoding appearance and motion affinity; (**d**) optimizing our spatiotemporal closure corresponds to finding a closed surface cutting low affinity graph edges; (**e**) our optimization framework results in multiple multiscale hypotheses, corresponding to objects, objects with their context, and object parts. (Figure reproduced with kind permission from Springer Science+Business Media: *Proceedings, 10th Asian Conference on Computer Vision (ACCV)*, 4th Mexican Conference on Pattern Recognition (MCPR), Spatiotemporal Closure, 2010, A. Levinshtein, C. Sminchisescu, and S. Dickinson, p. 370, Fig. 1)

We have extended this framework to detect spatiotemporal closure [15, 16]. Similar to detecting contour closure in images, we formulate spatiotemporal closure detection inside a spatiotemporal volume (Fig. 1.8(a)) as selecting a subset of spatiotemporal superpixels whose collective boundary falls on such discontinuities (Fig. 1.8(b)). Our spatiotemporal superpixels, extending our superpixel framework in [17], provide good spatiotemporal support regions for the extraction of appearance and motion features, while limiting the undersegmentation effects exhibited by other superpixel extraction techniques due to their lack of compactness and temporal stability.

We proceed by forming a superpixel graph whose edges encode appearance and motion similarity of adjacent superpixels (Fig. 1.8(c)). Next, we formulate spatiotemporal closure. The notion of contour gap from image closure detection is generalized to the cost of a cut of a set of spatiotemporal superpixels from the rest of the spatiotemporal volume, where the cut cost is low for superpixel boundaries that cross appearance and motion boundaries. Similarly, instead of normalization by area, we choose to normalize by a measure of internal motion and appearance homogeneity of the selection, which is more appropriate for video segmentation. The cost is again minimized using parametric maxflow [12] which is not only able to efficiently find a globally optimal closure solution, but returns multiple closure hypotheses (Fig. 1.8(e)). This not only eliminates the need for estimating the number of objects in a video sequence, as all objects with the best closure are extracted, but can result in hypotheses that oversegment objects into parts or merge adjacent objects. Multiple spatiotemporal segmentation hypotheses can serve tasks such as action recognition, video synopsis, and indexing [22].

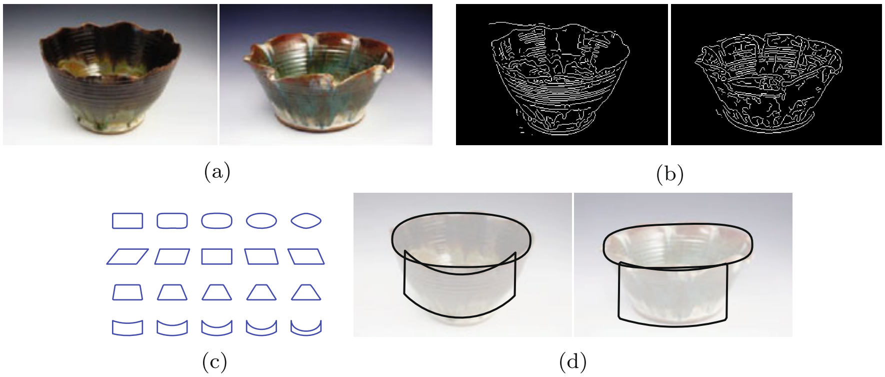


Fig. 1.9 Recovering abstract shape parts from an image: (**a**) input image of two exemplars that show considerable within-class variation; (**b**) extracted contours: note that corresponding contour-based features are seldom in one-to-one correspondence; (**c**) a simple example vocabulary of 2-D part models that will drive the perceptual grouping and shape abstraction processes; (**d**) the resulting abstract surfaces recovered by our framework; contour correspondence exists not at the level of individual contours, but at a much higher level of abstraction. (Figure reproduced with kind permission from Springer Science+Business Media: *Proceedings, 11th European Conference on Computer Vision (ECCVC)*, Contour Grouping and Abstraction using Simple Part Models, 2010, P. Sala and S. Dickinson, p. 604, Fig. 1)

1.4 Abstract Part Recovery

In the previous two sections, we reviewed approaches based on traditional Gestalt grouping principles such as symmetry and closure. But consider Fig. 1.9(a), which shows images of two object exemplars belonging to the same class (bowl). If we examine their extracted contours, shown in Fig. 1.9(b), we notice that corresponding contour-based features are seldom in one-to-one correspondence. Despite this lack of contour correspondence, the two objects are perceived as having similar shape *without any a priori knowledge of object class*, i.e., you did not run a successful bowl detector on both images. Somehow, you not only grouped this plethora of contours into surfaces, but *abstracted* the groups to yield emergent shapes that were common to both images. While cues such as symmetry and closure are indeed powerful mid-level regularities that could drive perceptual grouping of these contours, the complexity of the contours begs the question: Is there some sort of higher-level regularity, lying somewhere between low-level perceptual grouping and knowledge of the target object, that can be used to not only group the contours but recover their abstract shape?

In this third and final section of this chapter, we review our approach to the perceptual grouping and abstraction of image contours using a set of 2-D part models; details can be found in [24]. We assume no object-level prior knowledge and, like the perceptual grouping community, assume only a mid-level shape prior. However, our shape prior is slightly stronger than such classical Gestalt features as symmetry, parallelism, proximity, collinearity, etc. Specifically, our mid-level shape prior takes the form of a user-defined vocabulary of simple 2-D shape models, representing a

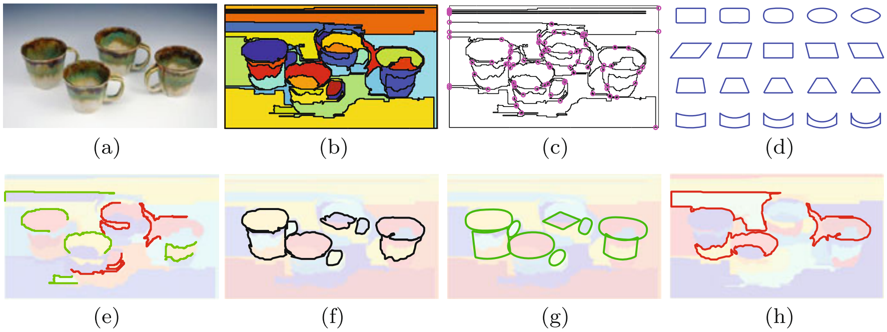


Fig. 1.10 Problem formulation: (**a**) input image; (**b**) region oversegmentation; (**c**) region boundary graph; (**d**) example vocabulary of shape models (used in our experiments); (**e**) example paths through the region boundary graph that are consistent (*green*) and inconsistent (*red*); (**f**) example detected cycles that are consistent with some model in the vocabulary; (**g**) abstractions of cycles consistent with some model; (**h**) example cycles inconsistent with all models. (Figure reproduced with kind permission from Springer Science+Business Media: *Proceedings, 11th European Conference on Computer Vision (ECCVC)*, Contour Grouping and Abstraction using Simple Part Models, 2010, P. Sala and S. Dickinson, p. 606, Fig. 2)

fixed set of parts from which a large database of object models can be constructed. In that sense, our vocabulary can be seen as a high-level nonaccidental regularity—a common denominator set of part shapes that can be used to model a large collection of objects in the world [7–9]. But since different domains may demand different vocabularies of parts, it's essential that our framework be *independent* of the part vocabulary; therefore, the vocabulary is an input to our framework.

Returning to our illustrative example, in Fig. 1.9(c), we show sample instances from a simple, example vocabulary of 2-D shapes that will be used to group and abstract the contours in Fig. 1.9(b). In Fig. 1.9(d), we overlay the abstract shapes recovered by our algorithm. It is at this level, i.e., the abstracted parts and their relations, that commonality exists between the two images. Moreover, the boundaries of these abstract parts may not correspond to explicit image boundaries in the image. Rather, they can be viewed as *hallucinations* of the actual image boundaries, after they're appropriately selected and grouped.

Our approach begins by computing a region oversegmentation (Fig. 1.10(b)) of the input image (Fig. 1.10(a)). The resulting region boundaries yield a *region boundary graph* (Fig. 1.10(c)), in which nodes represent region boundary junctions where three or more regions meet, and edges represent the region boundaries between nodes; the region boundary graph is a multigraph, since there may be multiple edges between two nodes. Our approach can be formulated as finding simple cycles in the region boundary graph whose shape is consistent with one of the model shapes in the input vocabulary (Fig. 1.10(d)); these are called *consistent cycles*. There is an exponential number of simple cycles in a planar graph [3], and simply enumerating all cycles (e.g., [30]) and comparing their shapes to the model shapes is intractable. Instead, we start from an initial set of starting edges and extend these paths, called *consistent paths* (or CPs), as long as their shapes are consistent with a part of *some* model. To determine whether a given path is consistent (and therefore extendable), we approximate the path at multiple scales with a set of polylines (piecewise linear approximations), and classify each polyline using a one-class classifier trained on the set of training shapes (Fig. 1.10(e)). When a consistent path is also a simple cycle, it is added to the set of output consistent cycles (Fig. 1.10(f)).

Figure 1.10(d) shows the input vocabulary used in our experiments: four part classes (superellipses plus sheared, tapered, and bent rectangles, representing the rows) along with a few examples of their many within-class deformations (representing the columns). Each shape model is allowed to anisotropically scale in the horizontal and vertical directions as well as rotate in the image plane. Since we employ scale-, rotation-, and translation-invariant features to train the classifiers, we need to generate only (approximately) 1,500 instances (by varying the aspect ratio and deformation parameters) belonging to these four shape classes. A *single* classifier is trained on all the component polylines (computed at multiple scales) of length (i.e., number of piecewise linear segments) k spanning the *complete* set of shape models and their deformations. Therefore, if K is the upper bound on the length of a polyline approximating a shape in the vocabulary, then K classifiers are trained. An ideal vocabulary defines a small set of "building blocks" common to a large database of objects. As such, the complexity of the vocabulary shapes is low, and even at the finest scale of polyline partitioning of a vocabulary shape's contour, K remains low; for our vocabulary, K is 13.

The algorithm outputs cycles of contours that are consistent with one of the model (training) shapes. A cycle consists of actual contours (edges in the region boundary graph) in the image, and therefore does not explicitly capture the abstract shape of the contours. Moreover, the cycle has not yet been categorized according to the shapes in the vocabulary. To abstract (or regularize) the shape of a cycle and to categorize it, we employ an active shape model (ASM) [5] trained on about 600,000 model instances (generated by varying their aspect ratio, orientation, and a finer sweeping of the deformation parameters than the one used to train the polyline classifiers). We iterate over the classical two-step ASM procedure, consecutively aligning and deforming the mean training shape with the cycle until convergence. However, we depart from a standard ASM framework in two key ways.

In a standard ASM framework, the training shapes belong to a single shape class, and the allowable, often limited, deformations are typically captured (using PCA) in a low-dimensional shape space that can be approximated by a multidimensional Gaussian distribution. Moreover, at run time, the model must be properly initialized, for if the model is grossly misaligned, the deformations required to warp the model into the image may fall outside the space of allowable deformations. In our case, given a consistent cycle, we don't know which category of vocabulary shape it belongs to, and hence which ASM model to apply (if we assumed one model per category in the vocabulary). Moreover, even if we knew its category, we assume no correct or near-correct initial landmark correspondence. We overcome the first problem by having a single ASM that's trained on all instances of all the shapes in the vocabulary, and overcome the second problem by training on all possible landmark correspondences (alignments) across these shapes.

After ASM convergence, the training shape closest to the deformed model identifies the category of the cycle. In the previous step, the consistent cycle classifier's precision rate is never 100 % at reasonable recall rates, and some of the recovered consistent cycles (of contours) may yield shapes that are qualitatively different from those in the vocabulary. Therefore, following ASM convergence, shapes that are still significantly different from the training shapes are discarded. Figure 1.10(g) illustrates the vocabulary shapes abstracted from the consistent cycles in Fig. 1.10(f); for each detected shape, the algorithm also yields its shape category. Finally, Fig. 1.10(h) illustrates some of the false positives discarded by the shape abstraction process.

In order to evaluate our framework, we created an annotated dataset with 67 images containing object exemplars whose 3-D shape can be qualitatively described by cylinders and bent or tapered cubic prisms. The abstract visible surfaces of each 3-D shape were hand-labeled using 2-D models drawn from our vocabulary. Figure 1.11 illustrates the output of our system on a number of examples in the dataset: column (a) shows the input image; column (b) shows the region oversegmentation used as input to our algorithm, computed using the local variation approach by Felzenszwalb and Huttenlocher [10] with a fixed parameterization on all images; column (c) shows the consistent cycles from which the shapes in column (d) were abstracted, representing the recovered parts closest to the ground truth in column (e). The numbers inside recovered abstract parts in column (d) indicate the rank of the part among all recovered parts in that image, computed as a function of the distance to the contours of the cycles that they are abstracting. The target regions can sometimes rank low if their degree of abstraction is high compared to non-target regions in the image (whether real or segmentation artifacts) that require less abstraction. Note that in some cases, e.g., the blender body in row 8, the ideal ground truth part (e.g., corresponding to the projection of the body of a tapered cylinder) did not exist in the vocabulary.

Exploring the results in more detail, we see that Fig. 1.11(d) shows the ability of our approach to abstract object surfaces that are locally highly irregular due to noise or within-class variation, but capture a model shape at a higher level of abstraction. In some cases (e.g., rows 5, 6, and 8), we see misalignment with a neighboring shape. This can be due to two reasons: (1) the vocabulary may not contain the appropriate shape to model the surface; and (2) the shapes are recovered independently, with no alignment constraints exploited; such constraints, as well as other constraints, will play an aggressive role in pruning/aligning hypotheses in our future work. In all the examples, we can see that the model abstraction process is able to cope with region undersegmentation when it is restricted to a relatively small section of the contour.

Our ability to abstract the shape of a cycle of contours with high local irregularity (shape "noise") means that many false positive parts will be recovered. In [25], we addressed this precision problem by moving the camera and exploiting spatiotemporal constraints in the grouping process. We introduced a novel probabilistic, graph-theoretic formulation of the problem of spatiotemporal contour grouping, in which the spatiotemporal consistency of a perceptual group under camera motion is

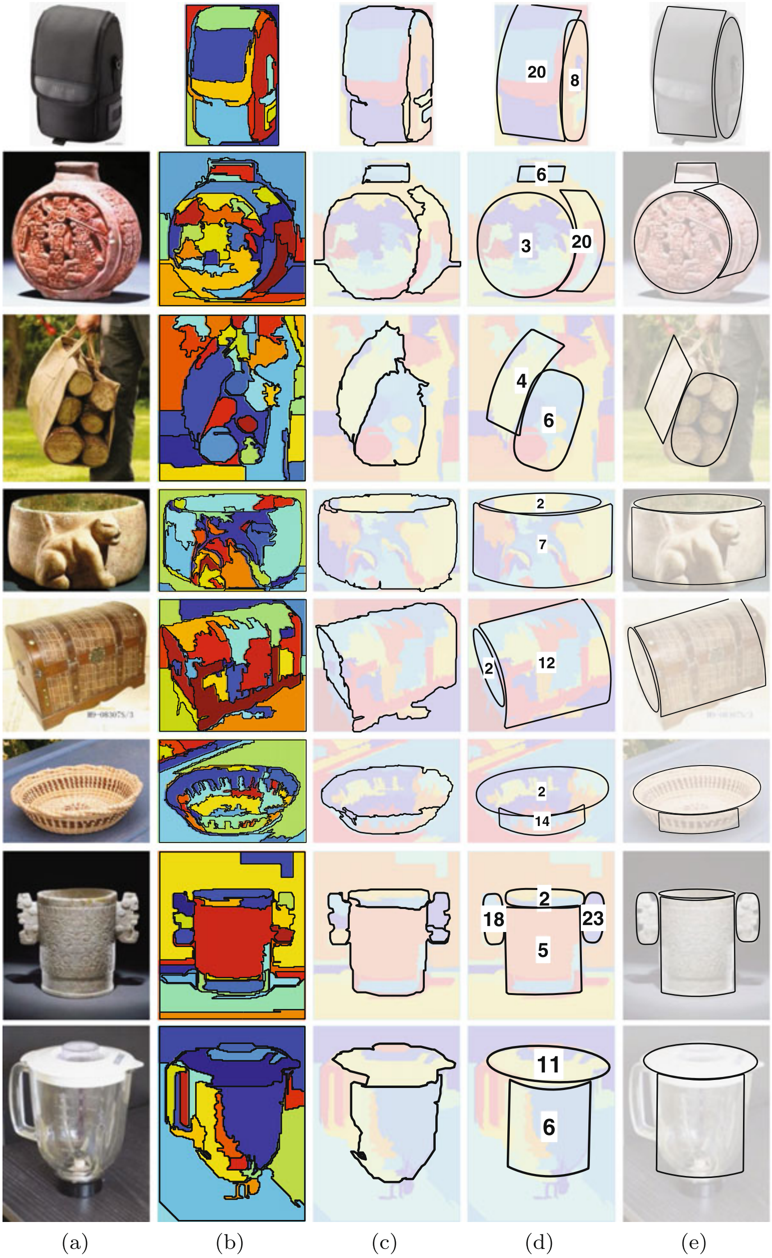


Fig. 1.11 Abstract part recovery (see text for discussion). (Figure reproduced with kind permission from Springer Science+Business Media: *Proceedings, 11th European Conference on Computer Vision (ECCVC)*, Contour Grouping and Abstraction using Simple Part Models, 2010, P. Sala and S. Dickinson, p. 613, Fig. 4)

learned from a set of training sequences. In future work, we plan to explore powerful contextual relations, including proximity, alignment, and 3-D shape information to prune many of these false positives. For example, if the surfaces in our images can indeed be the projections of volumetric parts, such as cylinders or prisms, then there are strong constraints on the shapes and relations of the component faces (parts) of their aspects. Other constraints are also possible, such as pruning smaller surfaces that are subsumed by larger surfaces.

1.5 Conclusions

The perceptual grouping of contours has long been a problem of interest to human and computer vision researchers alike. In computer vision, classical approaches have addressed the problem by first extracting contours and then grouping the contours, leading to prohibitive combinatorial complexity. We have explored this problem from the dual standpoint of region-based grouping, where regions are superpixels that minimize undersegmentation. In the case of symmetry-based grouping, the superpixels represent deformable, maximally inscribed disks (medial points), and we learn to group them when they belong to the same symmetric part. In the case of closure-based grouping, the superpixels represent "chunks" of boundary, and when the right subset of superpixels is found, those chunks of boundary will form a closure with minimal gap. Finally, in the case of part-based grouping and abstraction, the superpixels define an intractable space of contour cycles from which those whose coarse shape matches a model part are efficiently found. In each case, oversegmented regions, or superpixels, not only help manage the combinatorial complexity of traditional contour grouping, but support the inclusion of appearance information.

As the community moves from single category detection to recognition from very large databases, the strong priors provided by object detectors will have to give way to domain-independent intermediate shape priors that can yield discriminative shape structures that, in turn, are required for efficient indexing. These mid-level shape priors represent a return to perceptual grouping, and we expect research activity in this area of critical importance to rise again. Shape is clearly the most powerful and the most invariant feature of most categories, but a single shape part, unlike a SIFT feature, carries very little distinctiveness. Only when shape primitives are nonaccidentally grouped together do the resulting higher-order structures possess the indexing power required to prune a large database down to a few promising candidates. In each of the frameworks reviewed in this chapter, the perceptual grouping of superpixels yields a rich shape structure (in the case of a closed contour, a set of parts and relations can be easily extracted [27]) that will support powerful shape indexing and categorization.

Acknowledgements The authors gratefully acknowledge the support of NSERC, Mitacs, and DARPA. Sven Dickinson would like to thank Keith Price for providing the data in Fig. 1.1(b).

References

1. Binford TO (1971) Visual perception by computer. In: Proceedings, IEEE conference on systems and control, Miami, FL
2. Blum H (1967) A transformation for extracting new descriptors of shape. In: Wathen-Dunn W (ed) Models for the perception of speech and visual form. MIT Press, Cambridge, pp 362–380
3. Buchin K, Knauer C, Kriegel K, Schulz A, Seidel R (2007) On the number of cycles in planar graphs. In: In proceedings, COCOON, LNCS, vol 4598. Springer, Berlin, pp 97–107
4. Carreira J, Sminchisescu C (2012) Cpmc: automatic object segmentation using constrained parametric min-cuts. IEEE Trans Pattern Anal Mach Intell 34(7):1312–1328
5. Cootes TF, Taylor CJ, Cooper DH, Graham J (1995) Active shape models–their training and application. Comput Vis Image Underst 61(1):38–59
6. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: CVPR, pp 886–893
7. Dickinson S, Pentland A, Rosenfeld A (1990) A representation for qualitative 3-d object recognition integrating object-centered and viewer-centered models. In: Leibovic K (ed) Vision: a convergence of disciplines. Springer, New York
8. Dickinson S, Pentland A, Rosenfeld A (1992) From volumes to views: an approach to 3-d object recognition. CVGIP, Image Underst 55(2):130–154
9. Dickinson S, Pentland A, Rosenfeld A (1992) 3-d shape recovery using distributed aspect matching. IEEE Trans Pattern Anal Mach Intell 14(2):174–198
10. Felzenszwalb P, Huttenlocher D (2004) Efficient graph-based image segmentation. Int J Comput Vis 59(2):167–181
11. Huttenlocher D, Ullman S (1990) Recognizing solid objects by alignment with an image. Int J Comput Vis 5(2):195–212
12. Kolmogorov V, Boykov YY, Rother C (2007) Applications of parametric maxflow in computer vision. In: IEEE international conference on computer vision, pp 1–8
13. Levinshtein A, Dickinson S, Sminchisescu C (2009) Multiscale symmetric part detection and grouping. In: IEEE international conference on computer vision, September 2009
14. Levinshtein A, Sminchisescu C, Dickinson S (2010) Optimal contour closure by superpixel grouping. In: ECCV, pp 480–493
15. Levinshtein A, Sminchisescu C, Dickinson S (2012) Optimal image and video closure by superpixel grouping. Int J Comput Vis 100(1):99–119
16. Levinshtein A, Sminchisescu C, Dickinson SJ (2010) Spatiotemporal closure. In: ACCV, pp 369–382
17. Levinshtein A, Stere A, Kutulakos KN, Fleet DJ, Dickinson SJ, Siddiqi K (2009) Turbopixels: fast superpixels using geometric flows. IEEE Trans Pattern Anal Mach Intell 31(12):2290–2297
18. Li F, Carreira J, Sminchisescu C (2010) Object recognition as ranking holistic figure-ground hypotheses. In: CVPR, June 2010
19. Lindeberg T, Bretzner L (2003) Real-time scale selection in hybrid multi-scale representations. In: Scale-space. LNCS, vol 2695. Springer, Berlin, pp 148–163
20. Lowe D (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
21. Lowe DG (1985) Perceptual organization and visual recognition. Kluwer Academic, Norwell
22. Pritch Y, Rav-Acha A, Peleg S (2008) Nonchronological video synopsis and indexing. IEEE Trans Pattern Anal Mach Intell 30:1971–1984
23. Roberts L (1965) Machine perception of three-dimensional solids. In: Tippett J et al (eds) Optical and electro-optical information processing. MIT Press, Cambridge, pp 159–197
24. Sala P, Dickinson S (2010) Contour grouping and abstraction using simple part models. In: Proceedings, European conference on computer vision (ECCV), Crete, Greece, September 2010

25. Sala P, Macrini D, Dickinson S (2010) Spatiotemporal contour grouping using abstract part models. In: Proceedings, Asian conference on computer vision (ACCV), Queenstown, New Zealand, November 2010
26. Shi J, Malik J (2000) Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell 22(8):888–905
27. Siddiqi K, Shokoufandeh A, Dickinson S, Zucker S (1999) Shock graphs and shape matching. Int J Comput Vis 35:13–32
28. Stahl JS, Wang S (2007) Edge grouping combining boundary and region information. IEEE Trans Image Process 16(10):2590–2606
29. Street R (1931) A gestalt completion test: a study of a cross section of intellect. Teachers College Press, Columbia University, New York
30. Tiernan J (1970) An efficient search algorithm to find the elementary circuits of a graph. Commun ACM 13(12):722–726