Int J Comput Vis
DOI 10.1007/s11263-013-0614-3

Multiscale Symmetric Part Detection and Grouping

Alex Levinshtein · Cristian Sminchisescu · Sven Dickinson

Received: 1 January 2012 / Accepted: 10 January 2013

Abstract Skeletonization algorithms typically decompose an object's silhouette into a set of symmetric parts, offering a powerful representation for shape categorization. However, having access to an object's silhouette assumes correct figure-ground segmentation, leading to a disconnect with the mainstream categorization community, which attempts to recognize objects from cluttered images. In this paper, we present a novel approach to recovering and grouping the symmetric parts of an object from a cluttered scene. We begin by using a multiresolution superpixel segmentation to generate medial point hypotheses, and use a learned affinity function to perceptually group nearby medial points likely to belong to the same medial branch. In the next stage, we learn higher granularity affinity functions to group the resulting medial branches likely to belong to the same object. The resulting framework yields a skeletal approximation that is free of many of the instabilities that occur with traditional skeletons. More importantly, it does not require a closed contour, enabling the application of skeleton-based categorization systems to more realistic imagery.

Electronic supplementary material The online version of this article (doi:10.1007/s11263-013-0614-3) contains supplementary material, which is available to authorized users.

A. Levinshtein (✉) · S. Dickinson
University of Toronto, Toronto, USA
e-mail: babalex@cs.toronto.edu

S. Dickinson
e-mail: sven@cs.toronto.edu

C. Sminchisescu
University of Bonn, Bonn, Germany
e-mail: cristian.sminchisescu@ins.uni-bonn.de

C. Sminchisescu
Institute of Mathematics of the Romanian Academy, Bucharest, Romania

Keywords Perceptual grouping · Symmetry · Part detection · Segmentation

1 Introduction

The medial axis transform (Blum 1967) decomposes a closed 2-D shape into a set of skeletal parts and their connections, providing a powerful parts-based decomposition of the shape that's suitable for shape matching (Siddiqi et al. 1999; Sebastian et al. 2004). While the medial axis-based research community is both active and diverse, it has not kept pace with the mainstream object recognition (categorization) community that seeks to recognize objects from cluttered scenes. One reason for this disconnect is the limiting assumption that the silhouette of an object is available—that the open problem of figure-ground segmentation has somehow been solved. Even if it were possible to segment the figure from the ground in all circumstances, a second source of concern arises around the instability of the resulting skeleton—the skeletal branches often do not map one-to-one to the object's coarse symmetric parts. However, these limitations should in no way deter us from the goal of recovering an object's symmetric part structure from images. We simply need an alternative approach that does not assume figure-ground segmentation and does not introduce skeletal instability.

In this paper, we introduce a novel approach to recovering the symmetric part structure of an object from a cluttered image, as outlined in Fig. 1. 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. 1a) into compact superpixels at different superpixel resolutions (Fig. 1b). Superpixels are adequate for this task, balancing a data-driven component that is attracted

Published online: 08 February 2013

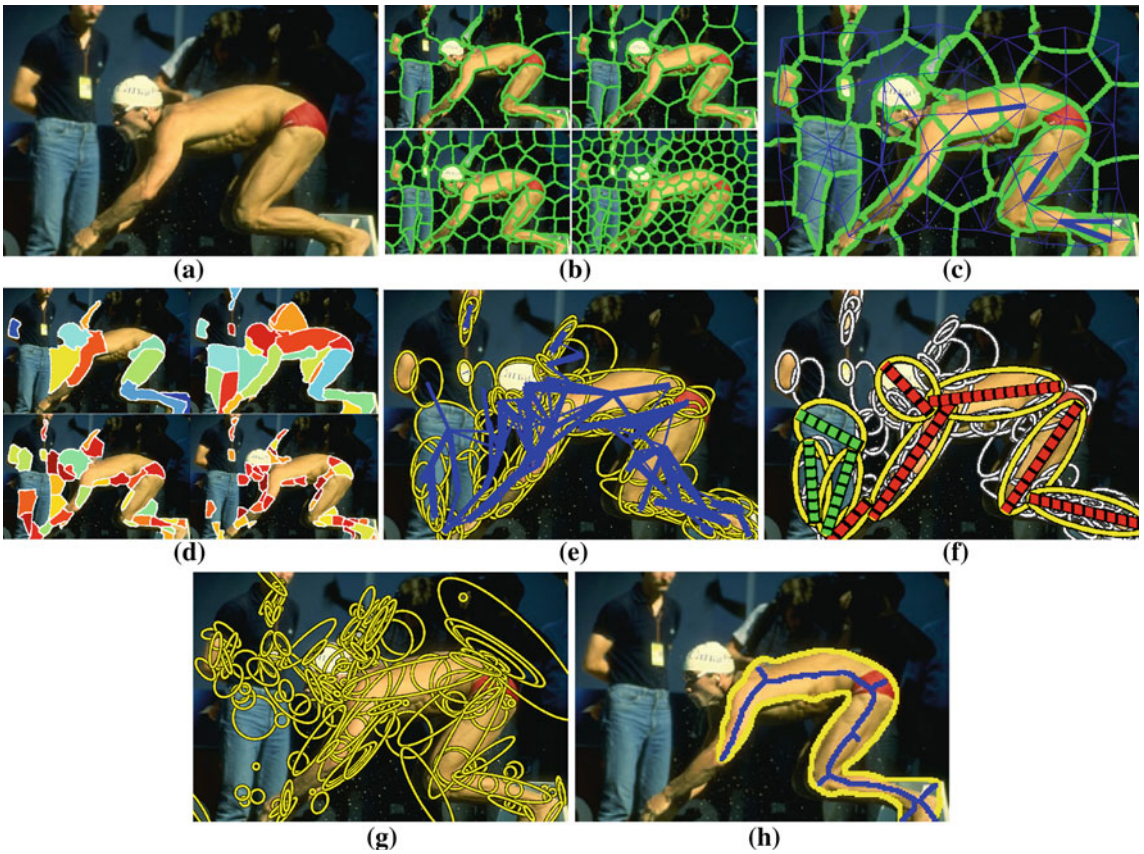


Fig. 1 Overview of our approach for multiscale symmetric part detection and grouping: **a** original image, **b** set of multiscale superpixel segmentations (different superpixel resolutions), **c** the graph of affinities shown for one scale (superpixel resolution), **d** the set of regularized symmetric 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 Lindeberg and Bretzner (2003), 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 (Shi and Malik 2000) 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, unstable branches, and poor part delineation

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 connecting 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. 1c). The affinities are learned from a set of training images whose symmetric parts have been identified by human annotators. A standard graph-based segmentation algorithm applied to each scale yields a set of superpixel clusters which, in turn, yield a set of regularized symmetric parts (Fig. 1d).

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. 1e). Each edge is assigned an affinity, this time reflecting the degree to which two nearby parts are likely 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. Consequently, we explore two graph-based methods for grouping the detected parts. The first method is the same greedy approach that was used to cluster superpixels into parts. The second method employs parametric maxflow Kolmogorov et al. (2007) to globally minimize an unbalanced normalized cuts criterion over the part graph. Both methods yield part clusters, each representing a set of regularized symmetric elements and their hypothesized attachments (Fig. 1f).

Our approach offers clear advantages over competing methods. For example, classical multiscale blob and ridge detectors, such as Lindeberg and Bretzner (2003) (Fig. 1g), yield many spurious parts, a challenging form of input noise

for any graph-based indexing or matching strategy. And even if an opportunistic setting of a region segmenter's parameters yields a sufficiently good object silhouette (Fig. 1h), 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. This is an extension of the work in Levinshtein et al. (2009).

2 Related Work

Symmetry can serve as a powerful basis for defining object parts. While not all objects can be easily represented using a small set of symmetric parts, various symmetry-based part vocabularies have been shown to be sufficient to represent a large set of objects, e.g., Biederman's geons (Biederman 1987). Restricting objects to be modeled using such parts drawn from a small vocabulary offers many advantages over global models or templates, including the ability to cope with occlusion and part articulation. Top-down, model-based recognition (detection), in which models represent entire objects may not scale to large databases. Instead, by defining models first at the part level, a small vocabulary can drive part recovery, with recovered parts and relations used to drive object recognition. The only way a recognition system can scale to support tens of thousands of objects is to recover a set of domain independent parts and relations which, in turn, can prune a large database down to a small number of candidates, which can then be verified using object detectors. Over the last 40 years of recognition systems, symmetry has been the most common basis for defining a part shape vocabulary.

In 3D, symmetry is the principle behind (Binford's 1971) generalized cylinders, as well as (Pentland's 1986) superquadrics and (Biederman's 1985) geons , both restricted forms of generalized cylinders but no less symmetric. Each of these paradigms demonstrated that a small vocabulary of parts could be combined to form a very large vocabulary of objects. In 2D, symmetry has been present in the vision community even longer. Blum's (1967) medial axis transform (MAT) computed the skeleton of an object given its silhouette. The skeleton produced by the MAT is the locus of centers of maximally inscribed discs of the given shape, and can be easily computed using a distance transform. Skeletons have been widely used in the recognition community. For example, Siddiqi et al. (1999) computed the singularities (shocks) in the inward evolution of a silhouette's boundary and organized them into a shock graph, in which nodes represented medial branches (or symmetric parts) and edges captured part adjacency. A shock subgraph (a small subset of symmetric parts and their relations) can be used to prune a large database down to a small set of promising candidates (Shokoufandeh et al. 2005), and the canadiate models can be matched with the input shock graph (Siddiqi et al. 1999; Pelillo et al. 1999; Sebastian et al. 2004). An analogous 3-D medial (symmetric) representation computed from a 3-D mesh, yielding a graph of medial surface-based parts, can also form the basis of a recognition system (Siddiqi et al. 2008).

One issue with skeletonization techniques is that the resulting skeleton is sensitive to small perturbations in the shape of the object. Some skeleton extraction algorithms ignore this, passing the problem on to the recognition module. Others try to address the issue by smoothing either the original contour or the resulting medial axis. Recently, Macrini et al. (2011) proposed a more principled approach for the regularization and abstraction of a skeleton by analyzing its ligature structure, generating a *bone graph*, offering improved stability and recognition performance over a shock graph (Macrini et al. 2011). While this represents an important step toward shape modeling and categorization, an even more serious MAT-based approaches is that a silhouette of an object is typically necessary as input, effectively requiring a figure-ground segmentation input. Figure-ground segmentation without an object prior is, in general, an open problem, although some progress has been made recently Carreira and Sminchisescu (2010), Levinshtein et al. (2010). Therefore, we focus on the domain of symmetric part recovery from a complex scene (as opposed to a segmented object), and review mainly methods that are applicable to cluttered images. These methods can be grouped into three categories: (1) filter-based part extraction, (2) contour-based part extraction, and (3) model-based part extraction. The remainder of this section will cover related work in each of these categories.

2.1 Filter-Based Symmetric Part Extraction

The filter-based part extraction category refers to methods that pass an image through a bank of low-level filters and analyze the filter responses to find symmetric regions (Lindeberg 1996; Lindeberg and Bretzner 2003; Crowley and Parker 1984; Lowe 2004; Mikolajczyk et al. 2005). These regions, referred to as multiscale blobs and ridges, provide a coarse segmentation of an image into a set of symmetric parts. The use of filter-based symmetric part detection for multiscale abstract part extraction was proposed by Crowley and Parker (1984), Crowley and Sanderson (1987), who detected peaks (rotational symmetries) and ridges (elongated symmetries) as local maxima in a Laplacian pyramid, linked together by spatial overlap to form a tree structure. Object matching was then formulated as comparing paths through two trees. Shokoufandeh et al. (1999) extended this framework with improved blob response and linked the blobs in a full graph, reformulating object categorization into a coarse-to-fine graph

matching problem. Shokoufandeh et al. (2006) proposed a more elaborate matching framework based on Lindeberg and Bretzner's (2003) multiscale blob model, an extension of Lindeberg's older work (Lindeberg 1996). Finally, Levinshtein et al. (2005) demonstrate that multiscale blobs and ridges can serve as a promising basis for automatically learning symmetric, part-based, categorical shape models.

More recently, filter-based symmetric regions have been used to define robust feature descriptors for keypoint features for stereo matching or exemplar-based object recognition, made popular by the semi-local SIFT model (Lowe 2004). Motivated by the work of Lindeberg (1996) where blobs are detected as the maxima of the scale-normalized Laplacian of the image, Lowe uses a difference of Gaussians (DoG) as an approximation to the Laplacian for computational efficiency. Mikolajczyk et al. (2005) compare a number of these and other region extractors. Alternatives to using a Laplacian or a DoG include the determinant of the Hessian or the Harris operator for blob detection. All of the aforementioned blob detectors can be adapted to detect ridges using a procedure described by Mikolajczyk and Schmid (2002). While blobs/ridges have proven to be a strong basis for local feature matching, robustly capturing small homogeneous regions, the same is not true for object categorization, where large parts with possibly heterogeneous appearance need to be extracted. Simply detecting parts as local maxima in a set of multiscale filter responses leads to many false positives and false negatives, suggesting that successful part extraction requires paying closer attention to image contours.

2.2 Contour-Based Symmetric Part Extraction

The second category of symmetry detection approaches is comprised of methods that find symmetry by grouping image contours. Most of the techniques in this category are similar in that they start by detecting contours and then group the contours into one or more symmetric regions, representing symmetric parts. Unlike the approaches in the previous category that overcome the complexity of finding symmetric regions through the application of coarse filters, contour-based approaches are faced with the task of finding symmetry in a vast collection of image contours. Coping with the high complexity of contour grouping has led this subcommunity to consider heuristics inspired by the perceptual grouping community, such as proximity, cotermination, collinearity and co-curvilinearity, to manage the high grouping complexity. Symmetry is one such grouping constraint.

Early contour-grouping approaches, motivated by work in skeletonization, attempt to extract skeleton-like representations from real images. An example of such a technique is the work of Brady and Asada (1984), showing how smooth local symmetries (SLS) can be detected. The SLS skeletons are composed of midpoints of line segments forming the same angle with both sides of the bounding contour, and are visually similar to the MAT skeletons. In contrast to skeletonization approaches that assume the availability of an object's contour, Brady and Asada use circular arcs and straight line fragments fitted to Canny edges. The final set of local symmetries can be quite noisy and fragmented. Connell and Brady (1987) describe not only how to "clean" up this noisy set of symmetries using perceptual grouping, but also introduce a system that uses the resulting groups to model objects. In, Ponce's (1990) theoretical analysis examines various skeleton formulations and shows, for example, that the MAT skeletons are a more constrained set than the SLS skeletons of Brady and Asada. Ponce's analysis also results in yet another skeleton definition, accounting for skewed symmetries that arise from 3D projection. Yet despite the improvement of Ponce's and Connell's approach over the original technique of Brady and Asada, all three methods avoid the complexity of grouping image edges arising from real scenes by working with simplified imagery. Working with more realistic scenarios introduces significantly more edgels, making the extraction of closed contours, as well as the detection and grouping of symmetric parts, far more difficult.

Moving into the domain of more real world imagery, Saint-Marc et al. (1993) show how symmetries can be extracted by building on a B-spline representation of image contours. They fit B-splines to image edges and show how a variety of symmetries can be extracted by imposing constraints over pairs of B-splines. Unlike previous approaches, restricted to the detection of local symmetries, Cham and Cipolla (1995) employ a similar B-spline representation for the detection of global skewed symmetries. In Cham and Cipolla (1996), the they describe a different approach for solving the same problem, this time using their measure of "geometric saliency". In, Liu et al. (1998) propose a more principled approach to symmetry axis extraction by grouping pairs of points using Dijkstra's algorithm. Their method does not require the contour to be available a priori, but it does require an initialization with a pair of points and produces an open boundary. Moreover, no preprocessing into image curves is performed. Therefore, to cope with the complexity of finding the best sequence among all pairs of points in the image, the authors resort to using hashing techniques.

Note that all the approaches described so far extract global unbounded symmetry axes, or extract symmetric sections by analyzing closed contours or pairs of image curves. The global symmetry recovered by the first set of approaches is useful for scene analysis but is insufficient for parts-based object categorization. On the other hand, the type of symmetries extracted by the second set of approaches are ideal for object representation, but are shown to operate on a very restrictive set of images, usually containing a single object with homogeneous appearance. In real images containing multiple objects with heterogeneous appearance

imaged against a complex background, it is unlikely that meaningful closed contours could be recovered bottom-up or that objects (or their parts) would be bounded by a pair of extracted image curves. One likely scenario is that an object's contour would correspond to far more than a single closed image contour or a pair of image curves, requiring more elaborate grouping strategies.

Ylä-Jääski and Ade (1996) provide such a method by finding partial symmetries between straight edge segments and then grouping them together into complete axial descriptions. Stahl and Wang (2008) take a similar approach but use a much more principled grouping algorithm based on ratio cuts to obtain their symmetries. The authors start by extracting linear edge segments and construct symmetric quadrilaterals which are then used for grouping. The algorithm finds the best sequence of quadrilaterals that minimizes the gap in boundary edges, while maintaining a smooth symmetry axis as well as a compact internal region for the resulting symmetric part. Even though grouping is polynomial in the number of graph edges, the number of quadrilaterals, as well as the possible ways of filling the gaps between them, are prohibitive. Authors resort to heuristics to reduce the complexity of the problem, and they also provide an iterative approach for extracting multiple symmetric regions, by finding the best region and repeating the process after removing all the quadrilaterals associated with that region. Still, quadrilaterals typically number in the thousands and the running time is on the order of several minutes per image.

Although great advances were made in contour-based symmetry detection, the complexity of contour grouping remains the main challenge faced by all methods. Early work reduced this complexity by constraining the symmetry representation or working with simplified images, while recent approaches work under less constrained scenarios but have to rely on suboptimal grouping algorithms and/or grouping heuristics. Nonetheless, being dominantly data-driven in nature, compared top–down coarse filtering techniques, such approaches prove to be much more suitable for part-based symmetry detection despite their problematic complexity. For a more complete survey of symmetry detection from real imagery, the reader is encouraged to consult the definitive survey on symmetry detection by Liu et al. (2010).

2.3 Model-Based Symmetric Part Extraction

The third category of bottom-up symmetry detection, called model-based grouping, refers to methods that employ a top-down deformable symmetric shape model during the part extraction process. In fact, any filter-based technique can be seen as a model-based approach, as it detects parts by employing a top-down coarse shape model. However, unlike the case of filter-based techniques, where model detection is approximated by analyzing low-level filter responses, the models here are matched against image contours. Moreover, the assumption of having a model is made much more explicit, with models ranging from simple symmetric parts to complex arbitrary shapes (e.g., entire objects). While most techniques in this category model object shape, some address the domain of perceptual grouping and part shape. That said, the object detection approaches can sometimes be modified to detect simple shapes, and thus provide useful insight into symmetric part detection.

Work with deformable shape models has its roots in Kass et al. (1988). Their model restricts the shape to have a smooth boundary with strong underlying image edge support. Given such a weak shape model, the method is more suitable as a low-level segmentation approach. Moreover, the algorithm requires a rough global initialization of the model prior to image alignment. Unlike Kass et al. (1988) and Cootes et al. (1995) use object-specific deformable models in their work. However, the model still needs to be initialized close to the object for the approach to work. Examples of more practical techniques, attempting to automatically find multiple instances of a deformable shape model include Pentland (1990), Sala and Dickinson (2008), Sclaroff and Liu (2001). Pentland (1990) solves the problem using a filter-like approach. Representing object parts using 3D superquadrics, part templates are constructed a priori for different part projections and different settings of the deformation parameters. A final selection step chooses from among a rich set of detected part hypotheses. Sala and Dickinson (2008) employed a similar approach for symmetric part detection. They also represent 2-D parts as projections of the surfaces of a small vocabulary of deformable, symmetric 3-D volumetric parts. However, unlike Pentland (1990), who worked with range images and thereby avoided the complexity of heterogeneous object appearance, Sala and Dickinson were able to extract symmetric parts from real 2-D images. In recent work (Sala and Dickinson 2010), they use a vocabulary of symmetric parts models to both group image contours into a set of abstract symmetric parts with no knowledge of scene content.

In contrast to previous approaches, Sclaroff and Liu (2001) define their shape model as a deformable polygon, and while they use it for segmenting specific objects, the model can be easily adapted for symmetric part extraction. They pose the grouping problem as finding a subset of oversegmented image regions that satisfy their shape model. Similar approaches were employed by Ren and Malik (2003) to group superpixels for generic segmentation, and by Mori (2005) to group superpixels into human body parts. (Sclaroff and Liu 2001), the authors show that despite pruning the search space using various heuristics, a brute force approach still exhibits a prohibitive grouping complexity, forcing an approximate solution of the problem by using a greedy algorithm. Similar to contour grouping techniques, model-based methods are

faced with prohibitive algorithm complexity, this time arising from matching models to image data. Overcoming this issue, while accurately detecting symmetric regions, is the subject of ongoing research.

In summary, we reviewed three categories of symmetry detection approaches that illustrate the tradeoff between fast, less inaccurate methods that rely on low-level filter responses, and high-complexity contour grouping or model-based approaches that are much more data driven. All the aforementioned techniques share an additional recurring weakness. With few exceptions, symmetric parts are usually not grouped together. For example, in skeleton-based approaches, a skeleton already corresponds to a whole object. However, in order to use it for efficient object recognition, it needs to be parsed into stable branches that correspond to object parts—a challenging task as skeletons are sensitive to small shape perturbations. Unlike the case of skeletons, bottom-up symmetry extraction techniques result in a disconnected set of symmetric parts. While grouping them is perhaps easier than grouping low-level features, such as pixels, into whole objects, it is still the subject of ongoing research and not commonly addressed. Whole object skeletons or collections of unrelated symmetric parts undoubtedly simplify generic scene analysis. However, symmetry alone is not enough. Objects parts need to be related and/or grouped together, calling for the use of additional perceptual grouping rules.

The approach we present below addresses some of the limitations of the methods mentioned above. Compared to filter-based approaches, our symmetric parts are more data-driven since they are composed of superpixels, resulting in more precise parts with fewer false positives. On the complexity issue faced by contour- and model-based methods, by adopting a region-based approach with an efficient clustering methodology, our superpixels (medial point hypotheses) effectively group together nearby contours that enclose a region of homogeneous appearance. Drawing on the concept of extracting blobs at multiple scales, symmetric parts will map to "chains" of medial points sampled at their appropriate scale. Our goal will be to group together the members of such chains, ignoring those superpixels (the vast majority) that don't represent good medial point hypotheses. On issue of the smoothness and precise correspondences that are often required of contour grouping methods, we will learn from noisy training data the probability that two adjacent superpixels represent medial point approximations that belong to the same symmetric part; this probability forms the basis of our affinity function used to cluster medial points into chains. Finally, the affinity function that will form the basis of nonaccidental part attachment will be learnt from noisy training data. Addressing these issues yields a novel technique that aims to narrow the gap between work in the segmentation and medial axis extraction.

3 Medial Part Detection

The first phase of our algorithm detects medial parts by hypothesizing a sparse set of multiscale medial point hypotheses and grouping those that are non-accidentally related. In the following subsections, we detail the two components.

3.1 Hypothesizing Medial Points

Medial point hypotheses are generated by compact superpixels which, on one hand, adapt to boundary structure, while on the other hand, enforce a weak shape compactness constraint. In this way, superpixels whose scale is comparable to the width of a part can be seen as deformable maximal disks, "pushing out" toward part boundaries while maintaining compactness. If the superpixels are sampled too finely or too coarsely for a given part, they will not relate together the opposing boundaries of a symmetric part, and represent poor medial point hypotheses. Thus, we generate compact superpixels at a number of resolutions corresponding to the different scales at which we expect parts to occur; as can be seen in Fig. 1b, we segment an image into 25, 50, 100 and 200 superpixels. To generate superpixels at each scale, we employ a modified version (Mori et al. 2004) of the normalized cuts algorithm (Shi and Malik 2000) since it yields compact superpixels.

Each superpixel segmentation yields a superpixel graph, where nodes represent superpixels and edges represent superpixel adjacencies. If a superpixel represents a good medial point hypothesis, it will extend to (and follow) the opposing boundaries of a symmetric part, effectively coupling the two boundaries through two key forms of perceptual grouping: (1) *continuity*, where the intervening region must be locally homogeneous in appearance, and (2) *symmetry*, in that the notion of maximal disk bitangency translates to two opposing sections of a superpixel's boundary. Figure 2b illustrates

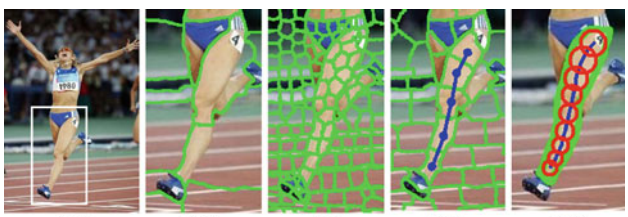


Fig. 2 Superpixels as medial point samples: **a** a region of interest focusing on the athlete's leg **b** superpixels undersample the scale of the symmetric part, **c** superpixels oversample the scale of the symmetric part, **d** superpixels appropriately sample the scale of the symmetric part, non-accidentally relating, through continuity and symmetry, the two opposing contours of the part, **e** the medial point hypotheses that effectively capture the scale of the part represent a sparse approximation to the locus of medial points that comprise the traditional skeleton

a symmetry section (blow-up of the subimage in Fig. 2a containing the athlete's leg) whose medial point hypotheses are too large (undersampled), while in Fig. 2c, the medial point hypotheses are too small (oversampled). When they are correctly sampled, as in Fig. 2d, they can be viewed as a sparse approximation to the locus of medial points making up a skeletal branch, as seen in Fig. 2e.

3.2 Clustering Medial Points

If two adjacent superpixels represent two medial points belonging to the same symmetric section, they can be combined to extend the symmetry. This is the basis for defining the edge weights in the superpixel graph corresponding to each resolution. Specifically, the affinity between two adjacent superpixels represents the probability that their corresponding medial point hypotheses not only capture nonaccidental relations between the two boundaries, but that they represent medial points that belong to the same skeletal branch. Given these affinities, a standard graph-based clustering algorithm applied independently to each scale yields groups of medial points, each representing a medial branch at that scale. In Sect. 4, we group nonaccidentally related medial branches by object, yielding an approximation to an object's skeletal part structure.

The affinity $A_s(i, j)$ between two adjacent superpixels R_i and R_j at a given scale has both shape (A_{shape}) and appearance ($A_{appearance}$) components. We learn the parameters of each component and their weighting from training data (Fig. 3). Note, that while we train our affinities on images from the Weizmann Horse dataset (Borenstein and Ullman 2002), we are careful to encode only relative information between superpixels and not absolute information such as superpixel color. As a result, our affinities are generic and can be used to detect symmetric parts in any domain, which we show in Sect. 5. To generate training examples, we segment an image into superpixels at multiple scales, and identify adjacent superpixels that represent medial points that belong to the same medial branch as positive evidence (e.g., superpixels corresponding to good medial points of the horse's leg are shown in Fig. 3); negative pairs are samples in which one or both medial point hypotheses are incorrect or, if both are valid medial points, belong to different but adjacent parts (e.g, superpixels spanning figure/ground boundaries are shown in Fig. 3). The boundary of the union of each superpixel pair defines a hypothesized boundary in the image (which may or may not have local gradient support).

To compute the shape-based affinity, we fit an ellipse to the union of two adjacent superpixels (we find an ellipse with the same second moments as the superpixel union region). We assign an edge strength to each boundary pixel equal to its Pb score (Martin et al. 2004) in the original image. Each boundary pixel is mapped to a normalized coordinate system

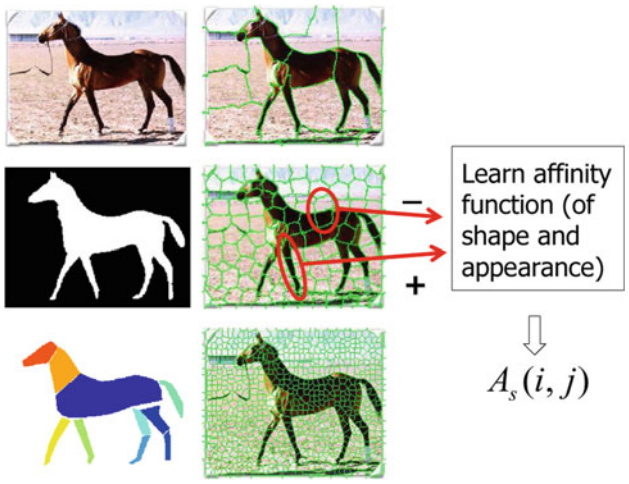


Fig. 3 Learning superpixel affinities. *Left column*: The training image (*top*), with ground truth figure/ground (*middle*) and ground truth parts (*bottom*). *Middle column*: Examples of positive and negative training data for superpixel clustering. Adjacent superpixels that represent medial points belonging to the same part serve as positive evidence (superpixels belonging to the leg at the middle superpixel resolution). Adjacent superpixels that span figure/ground boundaries (as shown for the middle image), belong to different parts, or belong to the same part but do not represent medial points (e.g., small superpixels on the horse's torso at the middle and fine scales) serve as negative evidence. The evidence is used to learn the appearance and the shape superpixel affinity components, which are combined into a single superpixel affinity $A_s(i, j)$

by projecting its coordinates onto the major and minor axes of the fitted ellipse, yielding a scale- and orientation-invariant representation of the region boundary. We partition our normalized coordinate system into rectangular bins of size $0.3a \times 0.3b$, where a and b are half the length of the major and minor ellipse axes, respectively. Using these bins, we compute a 2-D histogram on the normalized boundary coordinates weighted by the edge strength of each boundary pixel. Focusing on the superpixel pair union and its local neighborhood, we only consider bins in the range $[-1.5a, 1.5a]$ and $[-1.5b, 1.5b]$, resulting in a 10×10 histogram. This yields a shape context-like feature that reflects the distribution of edges along the presumed boundary of adjacent superpixels. Figure 4 illustrates the shape feature computed for the superpixel pair from Fig. 1c, corresponding to the thigh of the swimmer.

We train a classifier on this 100-dimensional feature using our manually labeled superpixel pairs, marked as belonging to the same part or not. The margin from the classifier (an SVM with RBF kernel) is fed into a logistic regressor in order to obtain the shape affinity $A_{shape}(R_1, R_2)$ whose range is $[0, 1]$. Table 1 compares various approaches for computing the shape affinity. Training and testing for these results employed our training dataset in a hold-one-out strategy (train on all but one images and report average results

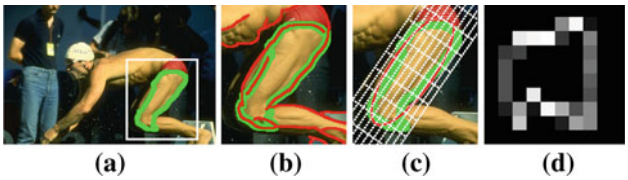


Fig. 4 Superpixel shape feature: **a** boundary of two adjacent superpixels representing two medial point hypotheses, **b** a blow-up of the two superpixels, in which the boundary of their union (*green*) defines a section of a hypothesized symmetric part which may or may not have underlying image edge support (*red*), **c** the normalized scale- and orientation-invariant coordinate system (grid in *white*) based on the ellipse (*red*) fitted to the superpixel union, **d** the shape-context-like feature that projects image edgels, weighted by edge strength, into this coordinate system

Table 1 Shape affinity comparison according to two measures: $F_{measure}$ and mean precision evaluated on test pairs of superpixels

	SVM-R	SVM-H	CC	HI
$F_{measure}$	0.75	0.75	0.42	0.44
Mean precision	0.79	0.79	0.29	0.31

We evaluate 4 methods: SVM with RBF kernel (SVM-R), SVM with histogram intersection kernel (SVM-H), as well as cross correlation (CC) and histogram intersection (HI) against a mean histogram of all positive training pairs.

for the held out image). The SVM with RBF kernel and the SVM with a histogram intersection kernel yield the highest performance. In the remainder of the paper, we used the RBF kernel with a sigma of 1, trained with a slack constant of $C = 10$.

For the appearance component of the affinity, we compute the absolute difference in mean RGB color, absolute difference in mean HSV color, RGB and HSV color variances of both regions, 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 our relatively small training dataset while emphasizing the weights of the more important features. This yields an appearance affinity measure between two regions

($A_{appearance}(R_1, R_2)$). 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 region affinity $A_s(i, j)$[1]. The resulting graph is used in conjunction with an efficient agglomerative clustering algorithm based on Felzenszwalb and Huttenlocher (2004) (complexity: $O(|S|)$, where S is a set of all superpixels) to obtain medial parts (medial point clusters). As the algorithm relies on edge weights that measure the dissimilarity between pairs of elements, we first convert the superpixel affinity to edge weights as $W(i, j) = \frac{1}{A_s(i,j)}$. The clustering algorithm initializes all medial point hypotheses as singletons, and maintains a global priority queue of edges by increasing edge weight (decreasing affinity A_s). At each iteration, the edge with the lowest weight (highest affinity) is removed from the queue, and the two clusters that span the edge are hypothesized as belonging to the same part. If each of the two clusters is a singleton, they are merged if the affinity is sufficiently high (the affinity captures the degree to which the union is symmetric). If one or both clusters contain multiple medial points (superpixels), the global symmetry A_s of the union is verified (in the same manner as a pair is verified, i.e., based on the same shape feature built over the union and the same logistic regressor) before the merge is accepted. Thus, while local affinities define the order in which parts are grown, more global information on part symmetry actually governs their growth. The result is a set of symmetric parts from each scale, where each part defines a set of medial points (superpixels). Combining the parts from all scales, we obtain the set $Part_1, Part_2, \ldots, Part_n$. Figure 1d shows the parts extracted at four scales. Our modified greedy clustering method has several parameters: k in Eq. 5 in Felzenszwalb and Huttenlocher (2004), minimum group size, and the minimum group affinity A_s during the merging of two superpixel groups. We set these parameters based on empirical observation to $k = 5$, minimum group size of 2 superpixels and minimum group affinity of 0.1.

4 Assembling the Medial Parts

Medial part detection yields a set of skeletal branches at different scales. The goal of grouping is to assemble the medial branches that belong to the same object. Drawing on the non-accidental relation of proximity, we define a single graph over the union of elements computed at all scales, with nodes representing medial parts and edges linking pairs in close proximity. Assigned to each edge will be an affinity that reflects the likelihood that the two nearby parts are not only part of the same object, but attached. Two different graph-based clustering techniques are then explored to detect part clusters. However, since some parts may be redundant across scales, a final selection step is applied to yield the final cluster of medial branches, representing an approximation to the object's skeletal part structure. The following two subsections describe these steps.

[1] Both the shape and the appearance affinities, as well as final affinity A_s, were trained with a regularization parameter of 0.5 on the L1-norm of the logistic coefficients.

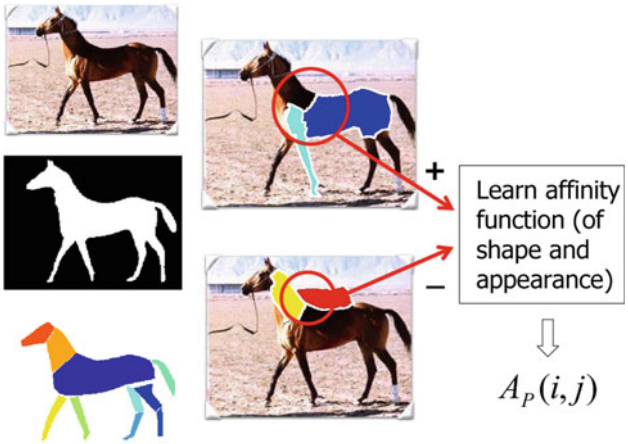


Fig. 5 Learning part affinities. *Left column* the training image (*top*), with ground truth figure/ground (*middle*) and ground truth parts (*bottom*). *Middle column* examples of positive (*top*) and negative (*bottom*) training data for part clustering. Adjacent detected parts that map to two attached ground truth parts serve as positive evidence (e.g., torso and leg). Adjacent detected parts that do not map to two attached ground truth parts (e.g., neck and a background part) serve as negative evidence. Combining appearance and shape features, the evidence is used to learn the part affinity $A_p(i, j)$

4.1 Medial Part Clustering

A minimal requirement for clustering two parts is their close proximity. While the projections of two attached parts in 3-D must be adjacent in 2-D (if both are visible), the converse is not necessarily true, i.e., adjacency in 2-D does not necessarily imply attachment in 3-D (e.g., occlusion). Still, the space of possible part attachments can be first pruned to those that *may* be attached in 3-D. Two parts are hypothesized as attached if one overlaps a scale-invariant dilation of the other (the part is dilated by the size of the minor axis of the ellipse fitted to it, in our implementation).

The edges in the graph can be seen as weak local attachment hypotheses. We seek edge affinities that better reflect the probability of real attachments. We learn the affinity function from training data—in this case, a set of ground truth parts and their attachments, labeled in an image training set (bottom left of Fig. 5). For each training image, we detect parts at multiple scales, hypothesize connections (i.e., form the graph), and map detected parts into the image of ground truth parts, retaining those parts that have good overlap with ground truth (Fig. 5). Positive training example pairs consist of two adjacent detected parts (joined by an edge in the graph) that map to attached parts in the ground truth (e.g., the torso and the leg of the horse in Fig. 5 middle). Negative training example pairs consist of two adjacent detected parts that do not map to adjacent ground truth parts (the neck and a background part in Fig. 5 middle).

As mentioned earlier, our multiscale part detection algorithm may yield redundant parts, obtained at different scales, but covering the same object entity. One solution would be to assign low affinities between such parts. However, this would mean that only one part in a redundant set could be added to any given cluster, making the cluster more sensitive to noisy part affinities. The decision as to which part in a redundant set survives in a cluster is an important one that is best made in the context of the entire cluster. Therefore, we assign a high affinity between redundant parts, and deal with the issue in a separate part selection step.

Formally, our part affinity is defined as:

$$A_p(i, j) = P_r(i, j) + (1 - P_r(i, j))A_{p,\neg r}(i, j) \tag{1}$$

where $P_r(i, j)$ is the probability that parts i and j are redundant, and $A_{p,\neg r}(i, j)$ is the affinity between the parts given non-redundancy. $P_r(i, j)$ is computed by training a quadratic logistic classifier over a number of features: overlap (in area) of the two parts (O_{ij}), defined as the overlap area normalized by the area of the smaller part, overlap of the two parts' boundaries (B_{ij}), and appearance similarity (A_{ij}) of the two parts. The features are defined as follows:

$$\begin{aligned} O_{ij} &= \frac{|Part_i \cap Part_j|}{\min\left\{|Part_i|, |Part_j|\right\}} \\ B_{ij} &= \frac{|\partial(Part_i \cap Part_j)|}{\min\left\{|\partial Part_i|, |\partial Part_j|\right\}} \\ A_{ij} &= A_{appearance}(Part_i, Part_j) \end{aligned} \tag{2}$$

where $|\cdot|$ is the region area and $|\partial(\cdot)|$ is the region perimeter.

Figure 6 gives examples of these four attachment types.

The affinity $A_{p,\neg r}(i, j)$ between non-redundant parts i and j, like affinities between medial points, includes both shape and appearance components. The components are best analyzed based on how the two parts are attached. Given an elliptical approximation to each part, we first compute the intersection of their major axes. The location is normalized by half of the length of the major axis, to yield a scale-invariant attachment position r for each part. We define three qualitative attachment "regions" to distinguish between four attachment types: inside ($|r| < 0.5$) , endpoint ($0.5 < |r| < 1.5$), or outside ($|r| > 1.5$). Our four apparent attachment categories can be specified as follows:

1. end-to-end ($J_{ij} = 1$)—The intersection lies in the endpoint region of both parts.
2. end-to-side ($J_{ij} = 2$)—The intersection lies in the inside region of one part and in the endpoint region of the other part.
3. crossing ($J_{ij} = 3$)—The intersection lies in the inside region of both parts.
4. non-attached ($J_{ij} = 4$)—The intersection lies in the outside region of one or both parts.

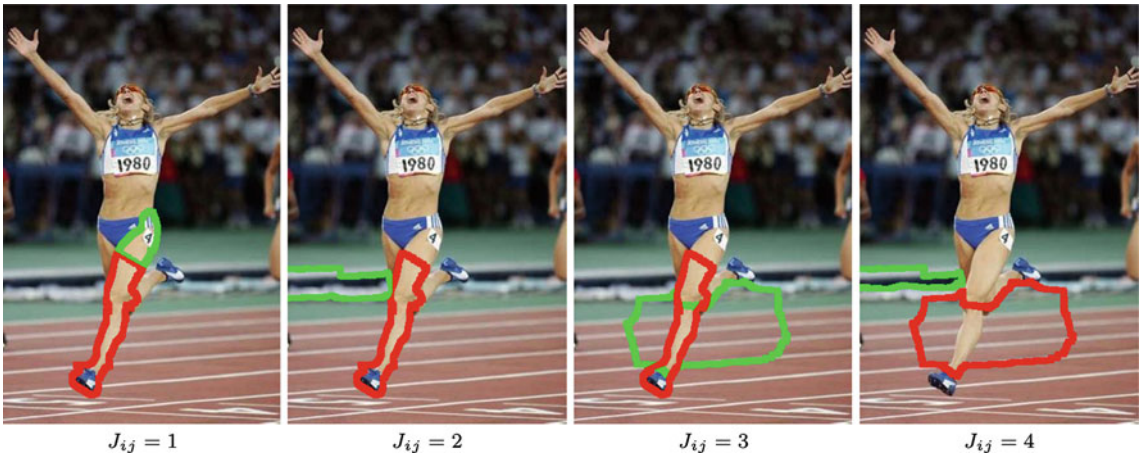


Fig. 6 Attachment categories. The four different attachment categories of parts (*green* and *red*): end-to-end ($J_{ij} = 1$), end-to-side ($J_{ij} = 2$), crossing ($J_{ij} = 3$), and non-attached ($J_{ij} = 4$). The affinity between any two parts depends on their attachment category

The shape component of our affinity is based on the principle that when one part attaches to (interpenetrates) another, it introduces a pair of concave (discontinuities Hoffman et al. (1984) principle of transversality), reflected as a pair of opposing L-junctions (i.e., concave discontinuities) marking the attachment. In contrast, when one part occludes another, the L-junctions are replaced by T-junctions, reflecting an occlusion boundary. This is a heuristic, for there could be an appearance boundary between two attached parts, misinterpreted as an occlusion boundary.

Since extracting and classifying contour junctions is challenging in the presence of noise, we will instead focus on the evidence of an occlusion boundary between two parts, based on image edges (E_{ij}) along the attachment boundary between parts i and j. Once the attachment boundary is found, evidence is accumulated as the average Pb (Martin et al. 2004) of the boundary pixels. Finding the attachment boundary is non trivial since the parts may be sufficiently close but not touching, due to segmentation errors.

The attachment boundary is computed similarly for all four attachment categories. For a pair of attached parts, we first select the part P_1 with the smaller $|r|$ and find the intersection of its boundary with the major axis of the other part P_2. The attachment boundary is centered at the intersection and extends along the boundary of P_1 in both directions, to an extent equal to the length of the minor axis (width) of P_2. For end-to-side attachments, this is illustrated in Fig. 7.

Given the attachment category J_{ij}, the attachment boundary evidence E_{ij}, and the appearance similarity A_{ij}, we can define the part affinity $A_{p,\neg r}(i, j)$. One logistic classifier is trained for end-to-end junctions ($A_1(i, j)$), whereas another is trained for end-to-side junctions ($A_2(i, j)$). For crossing and non-attached junctions, we set the affinity to 0 because we empirically found that none of the attached part pairs in the training set exhibited such attachment categories. Note,

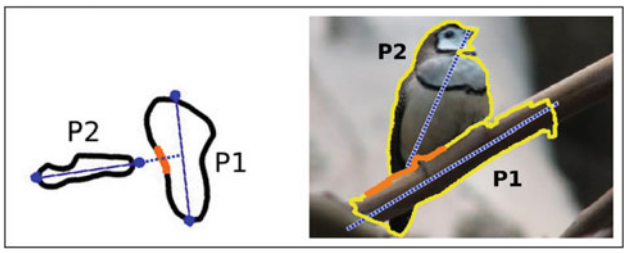


Fig. 7 Locating the attachment boundary between two parts in the case of an end-to-side attachment. The attachment boundary (*orange*) between the two parts P_1 and P_2 is centered at the intersection of the major axis of P_2 with the boundary of P_1, and extends along the boundary of P_1 a total distance equal to the length of the minor axis of P_2. *Left*—illustration of the attachment boundary. *Right*—attachment boundary between two parts in a real image

that the same features (see 3) are used for training $A_1(i, j)$) and $A_2(i, j)$). The difference is only in the training data that is used (A_1 is trained on end-to-end part pairs, while A_2 is trained on end-to-side part pairs). Our affinity for non-redundant parts becomes:

$$A_{p,\neg r}(i, j) = \left[J_{ij} = 1\right] \cdot A_1(i, j) + \left[J_{ij} = 2\right] \cdot A_2(i, j) \quad (3)$$

Having defined all the components of the affinity function $A_p(i, j)$ [2] (Eq. 1), we use these affinities to cluster parts that are attached.

We explore two graph-based approaches for part clustering. Our first approach is the same algorithm (Felzenszwalb and Huttenlocher 2004) used to cluster medial points into parts. Since this technique is greedy in nature, it is susceptible to undersegmentation given noisy part affinities. We there-

[2] All the logistic regressors for part affinities were trained with a regularization parameter of 0.1 on the L1-norm of the logistic coefficients.

fore explore a second, globally optimal, technique that makes use of parametric (Kolmogorov et al. 2007). Given the goal of finding a well-separated part cluster, we formulate the part clustering problem as finding an optimal unbalanced normalized cut (UNC), which is a measure of cluster dissimilarity to the rest of the graph relative to its internal similarity. Formally, given the part affinities $A_p(i, j)$, $D_i = \sum_j A_p(i, j)$, and a binary indicator vector $\mathbf{X}$ over parts,

$$UNC(\mathbf{X}) = \frac{cut(\mathbf{X})}{volume(\mathbf{X})} = \frac{\sum_{ij} X_i(1 - X_j)A_p(i, j)}{\sum_i X_i D_i}, \quad (4)$$

where $cut(\mathbf{X})$ is the sum of the affinities of all the edges between selected and unselected parts, and $volume(\mathbf{X})$, or the affinity volume, is the sum of all the affinities originating from the selected parts.

This cost can be globally minimized using parametric maxflow (Kolmogorov et al. 2007), returning multiple solutions with minimal cost under increasing affinity volume constraints. As it stands, however, the cost has a trivial minimizer $\mathbf{X} = 1$ that selects all the parts. To avoid this trivial solution, we modify the cost and add a small fixed penalty α_p for adding parts to the numerator. Moreover, note that our affinities $A_p(i, j)$ measure part attachment and not similarity in a clustering sense. Two parts in the graph are similar and should be clustered together if they are attached to the same object, meaning that there is a high attachment affinity path between them. To that end, we first compute a shortest path distance $D_p(i, j)$ for all pairs of parts based on their attachment affinities, and convert it into part similarity $W_p(i, j) = e^{-\frac{D_p(i,j)}{\sigma_p}}$. Letting $D'_i = \sum_j W_p(i, j)$, our final unbalanced Ncuts cost becomes:

$$UNC(\mathbf{X}) = \frac{cut(\mathbf{X}) + penalty(\mathbf{X})}{volume(\mathbf{X})} = \frac{\sum_{ij} X_i(1 - X_j)W_p(i, j) + \alpha_p \sum_i X_i}{\sum_i X_i D'_i}. \quad (5)$$

In the results section, we will compare the two part clustering approaches, showing that the second method achieves slightly better performance.

4.2 Medial Part Selection

Our affinity-based grouping yields a set of part clusters, each presumed to correspond to a set of attached parts belonging to a single object. However, any given cluster may contain one or more redundant parts. While such parts clearly belong to the same object, we prune redundancies to produce the final approximation to an object's skeletal part structure. Our objective function selects a minimal number of parts from each cluster that cover the largest amount of image, while at the same time minimizing overlap between the parts. The problem is formulated as minimizing a quadratic energy over binary variables. Let $X_i \in \{0, 1\}$ be an indicator variable representing the presence of the i^{th} part in a cluster. We seek the subset of parts that minimizes the following energy:

$$E = \sum_i X_i (K - |Part_i|) + \sum_{i,j} X_i X_j O_{ij} \quad (6)$$

where K controls the penalty of adding parts. In our experiments, we found that $K = 0.1 \cdot median\{|Part_i|\}$ is an effective setting for this parameter. We find the optimal X by solving a relaxed quadratic programming problem, in which real values are rounded to 0 or 1 Pentland (1990).

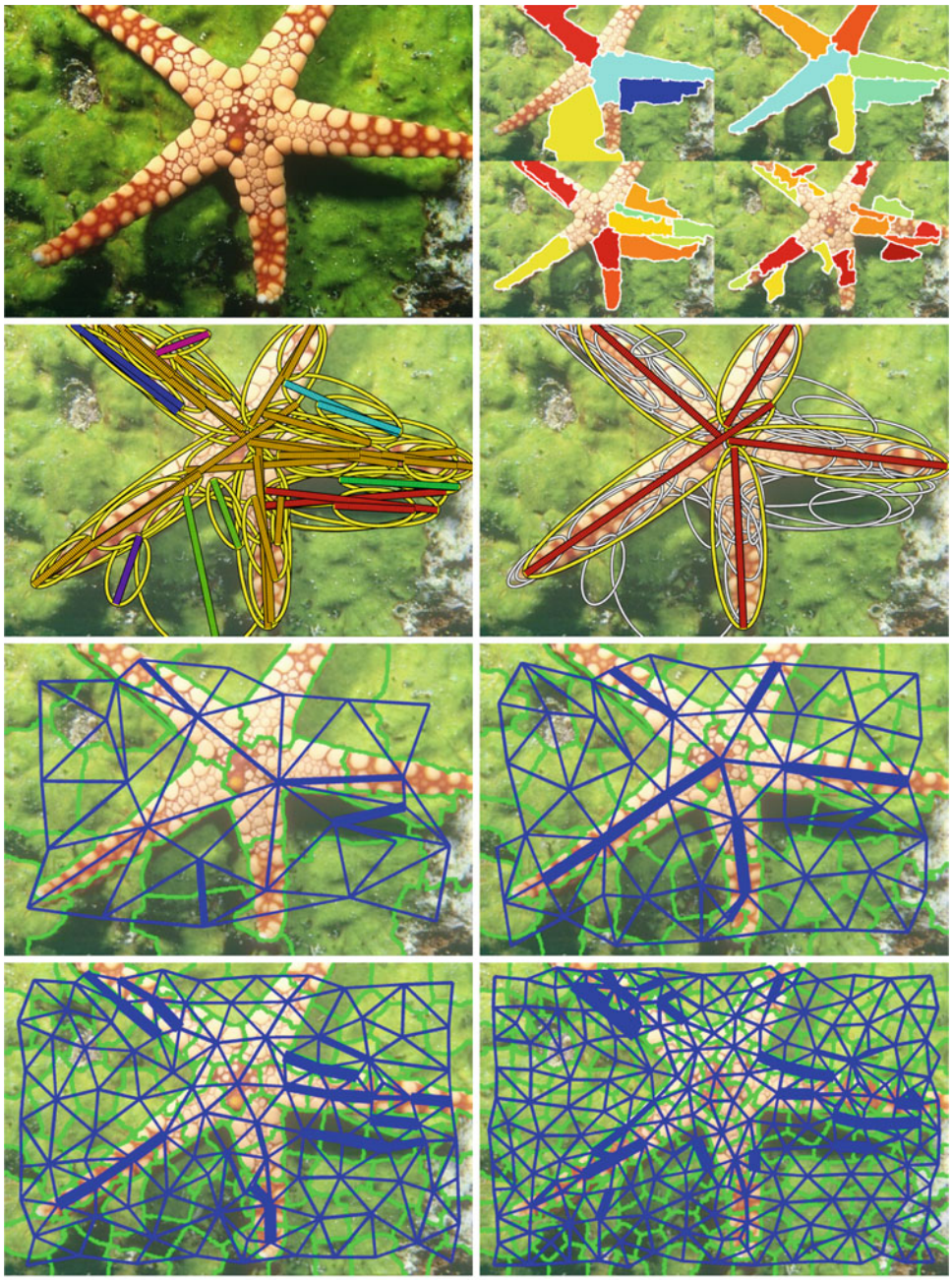
5 Results

To evaluate the method, we train the various components using a subset of the Weizmann Horse Database (Borenstein and Ullman 2002), consisting of images of horses together with figure-ground segmentations; in addition, we manually mark the elongated parts of the horses, together with their attachment relations. Our modified horse part dataset [3] contains 81 images, split into 20 training images, which we use to train all the components of our system, and 61 test images used for the quantitative evaluation below. The left column of both Figs. 3 and 5 illustrates an example training image and its ground truth segmentations. Once trained, we first qualitatively evaluate the system on images of objects with well-defined symmetric parts drawn from *different* (i.e., non-horse) image domains, reflecting our assumption that both symmetry and part attachment are highly generic.

Figure 8 shows qualitative results of our algorithm applied to a number of different image domains.[4] For all the results in this figure, the greedy technique Felzenszwalb and Huttenlocher (2004) was used for part clustering. In the first three cases (a–c), the figure shows the input image, the most prominent groupings of medial branches, as well as the results of intermediate stages in our system. For the remaining cases (d–m) only the most prominent groupings of medial branches are shown. For part cluster visualization, in order to avoid clutter, the abstractions of the parts in each cluster are shown as ellipses with their major axes (medial branch regularizations) depicted by dotted lines. All other parts are shown with faint (gray) elliptical part abstractions (without axes, for clarity), illustrating the ability of our algorithm to correctly group medial branches.

[3] The dataset can be downloaded from http://www.cs.toronto.edu/~babalex/horse_parts_dataset.tgz.

[4] Supplementary material (http://www.cs.toronto.edu/~babalex/symmetry_supplementary.tgz) contains additional examples.



(a)

Fig. 8 Detected medial parts and their clusters. For the first three test cases, we show the original image in true color, followed by the output of different stages overlaid on brightened images for better contrast. The results (ordered *left*-to-*right* and *top*-to-*bottom*) illustrate the recovered parts from each superpixel scale, part clusters with axis color indicating cluster membership, and the most prominent part clusters after the part selection stage (*yellow* ellipses correspond to selected parts, while others correspond to either unselected parts or parts from other clusters). The *bottom half* of the results shows the extracted superpixels and their affinities at our 4 scales. For the remaining test cases we show the most prominent part clusters only, without showing the output from intermediate stages

We organize the results in decreasing order of success, with Fig. 8a–f corresponding to more successful applications of our system and Fig. 8g–m illustrating some constraints and failure modes. Examining the results, Fig. 8a presents an ideal example of our system's operation. All the tentacles of the starfish were successfully recovered and grouped, with the small exception of the center being a part of one of the tentacles. This is a perhaps an easy example since the tentacles are of the same scale, exhibit strong appearance homogeneity and similarity, and are contrasted from the background. Indeed, paying closer attention to the superpixels and the affinities, we see that the second scale not only provides perfect medial disc approximations for all the parts, but the affinities between the superpixels of each tentacle are strong.

In Fig. 8b, we show that our system has successfully extracted the major parts of the athlete, including the torso, which exhibits heterogeneous appearance, and correctly grouped them together. Figure 8c illustrates not only that the parts of the windmill were successfully recovered and clustered, but that the person was also recovered as a separate single-part cluster. The smaller windmills undetected in

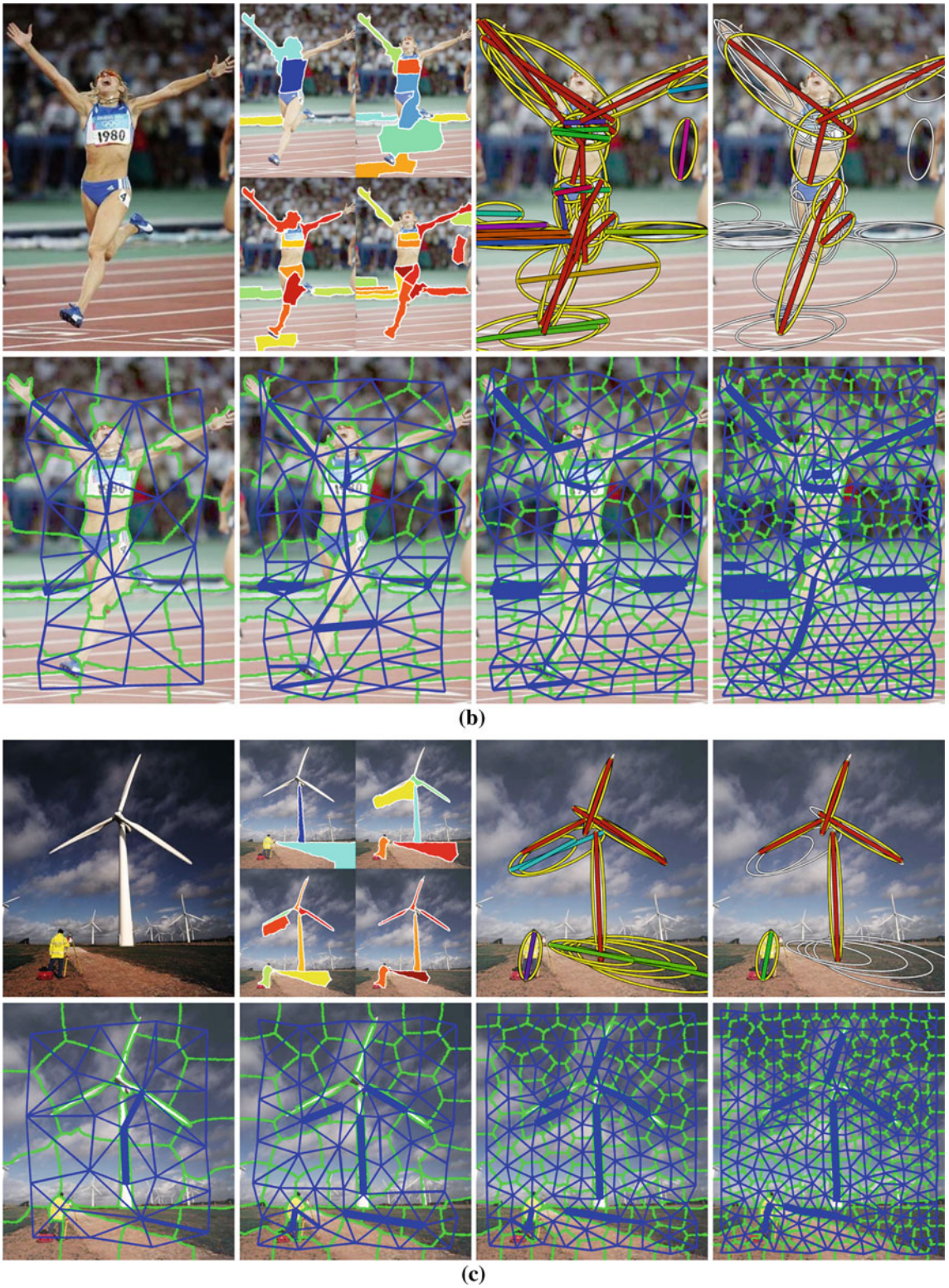


Fig. 8 continued

the background contain parts whose scale was smaller than our finest sampled scale. Figure 8d–f show other examples of our system's success, in which the major medial parts of a plane, swan, and statue, respectively, were recovered and grouped to yield an approximation to an object's skeletal part structure.

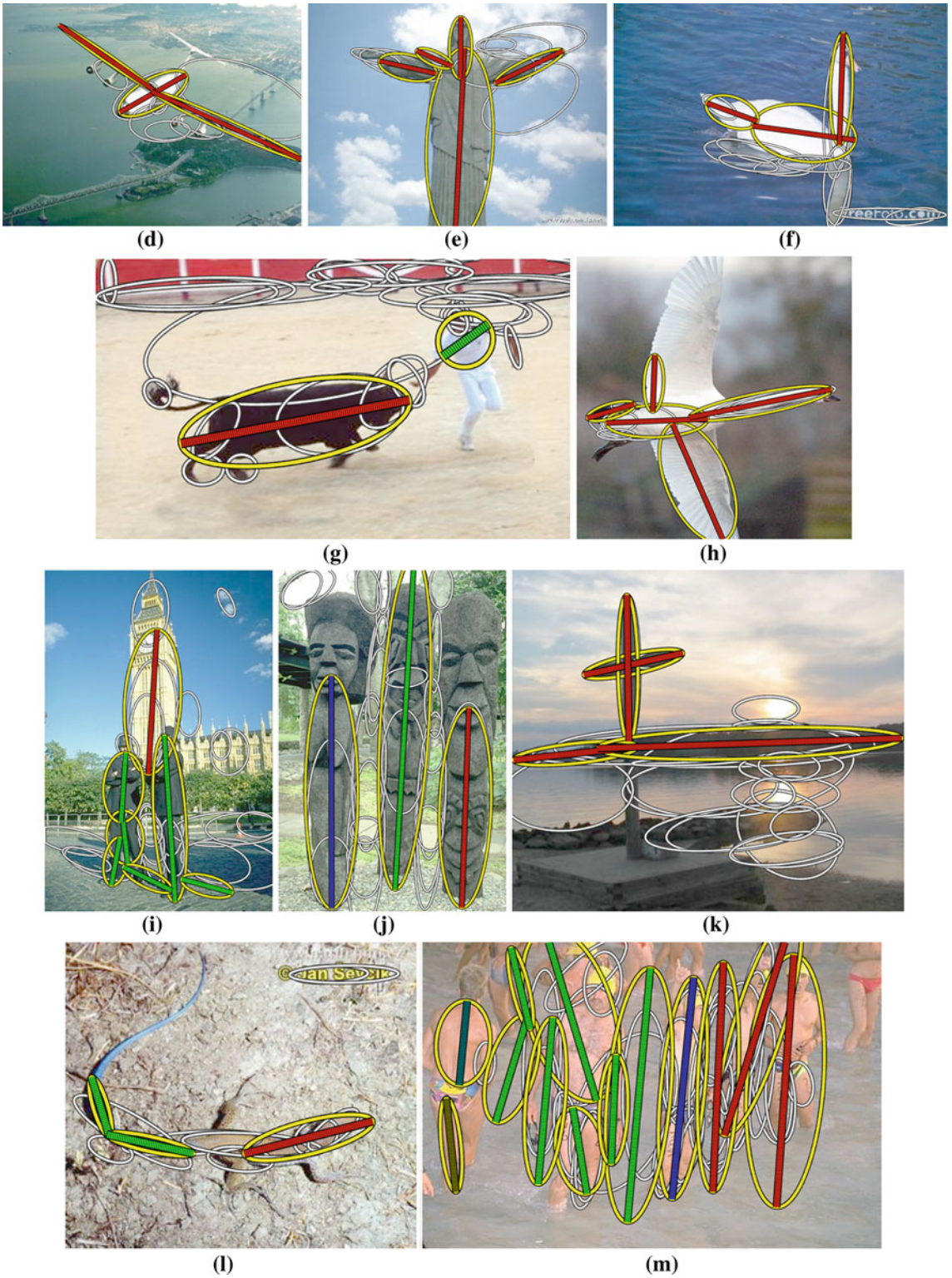


Fig. 8 continued

Figure 8g–m illustrate some limitations of our approach. Our framework relies on the assumption that good medial disc approximations can be extracted bottom-up, and our greatest failure mode occurs when this assumption is broken. While Fig. 8g shows that the bull is correctly detected as a symmetric object, despite irregularities in its contour, the

legs of the bull, as well as the arms and legs of the human, are not detected due to insufficient superpixel resolution or low contrast on part boundaries. These are the two main reasons for failing to extract good medial disc approximations and these failures recur in a number of test cases. For example, in Fig. 8h, one of the swan's wings is not properly detected. Due to insufficient contrast between the wing and the background, the superpixel boundaries fail to capture the part at any scale. Still, the remaining part structure of the swan may provide a sufficiently discriminative shape index to select a small set of candidate models, including the swan model, which could be used top-down to overcome segmentation problems. A similar effect can be seen in Fig. 8j, k, where the top parts of the statues and the center of the cross do not have sufficient contrast with their backgrounds. Figure 8d, l illustrate the second failure mode, where the tail of an airplane and the tail of a lizard are not captured since they are too thin to be well-represented by even our finest superpixel scale. This may be resolved by working at finer superpixel scales or by using different superpixel extraction strategies.

Additional issues in part detection arise when parts are tapered or have a curved axis. Figure 8l shows that although the main parts of the lizard are found, the tail is not composed of a single part since our system assumes parts with straight symmetry axes. Part tapering, and other deformations from roughly parallel part boundaries, also hinder detection. Figure 8e, f, i illustrate this effect. Tapering can result in wide sections of a part being captured at a coarse superpixel scale, and thinner sections being captured at a finer scale. Object extremities, such as the hand of the Jesus statue (Fig. 8e), the tail of the swan (Fig. 8f), and the tip of the tower (Fig. 8i), are all better represented at a finer scale than the remainder of these corresponding parts. Even if this effect is overcome and there is a single superpixel scale at which the whole part is well-captured, superpixel affinities are still negatively affected since most symmetric parts in our training set have parallel boundaries. This limitation may be overcome by a more diverse set of parts and a larger training set.

The part grouping stage of our system also suffers from some limitations. Starting the discussion from our part selection stage, we see that it is not perfect, occasionally resulting in suboptimal part groups. Figure 8b, e show examples where more precise arm parts were discarded over lower quality arm representations. However, our greatest failure mode is part clustering itself. While the affinities at the superpixel level consist of strong appearance and shape components, our part attachment affinities include weaker constraints and are more vulnerable to low contrast between objects and their background. In combination with our greedy part clustering approach, this can often lead to bleeding. In Fig. 8k, part undersegmentation occurs due to a lack of contrast at their attachment boundaries (symmetric strip of horizon landscape accidentally grouped with vertical mast). Like Fig. 8h, a candidate model may be required to resolve such ambiguous part attachments. Finally, while Fig. 8m shows that most swimmers and/or their parts were successfully detected, the vast number of resulting parts and their proximity overwhelm our greedy part grouping approach.

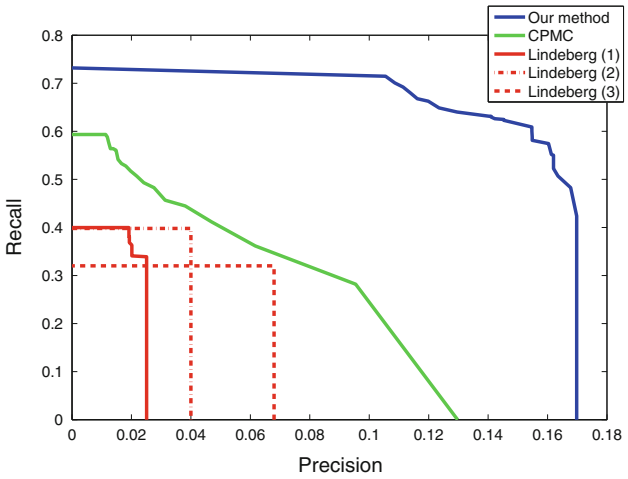


Fig. 9 Precision versus recall of part detection. We compare our method (*blue*) to Lindeberg and Bretzner (2003) (*red*) and Carreira and Sminchisescu (2012) (*green*). Due to the low precision of Lindeberg and Bretzner (2003), we prune small parts to increase precision: (1) no pruning, (2) prune parts whose major axis is less than 10 pixels, (3) prune parts whose major axis is less than 20 pixels. For CPMC, we return a fraction of all the segmentation candidates. We vary this fraction to get a PR curve

To provide a quantitative evaluation of our part detection strategy, we compare its precision and recall to the method of Lindeberg et al. (2003), used to generate the symmetric parts shown in Fig. 1g. We also compare to state-of-the-art generic segmentation algorithm of Carreira and Sminchisescu (2012). All methods are evaluated on 61 test images from the Weizmann Horse Dataset (Borenstein and Ullman 2002). A ground truth part is considered to be recovered if its normalized overlap (in area) with one of the detected parts is above a threshold (0.4). Our part detection offers a significant improvement in both precision and recall (Fig. 9). Compared to Lindeberg and Bretzner (2003) our method has a much stronger data-driven component and thus is able to recover parts much more accurately; and unlike Carreira and Sminchisescu (2012), we have a strong model of part shape to guide part recovery. Moreover, in Lindeberg and Bretzner (2003), no effort is made to distinguish part occlusion from part attachment; parts are simply grouped if they overlap. Note that both methods achieve low precision. This is partially due to the fact that there are other symmetric parts in the images, in addition to the horse body parts, that were not marked in the ground truth. Moreover, due to our multiscale part detection approach, the same ground truth part may be recovered at multiple scales, hindering precision in the absence of some redundancy removal step. Note that our

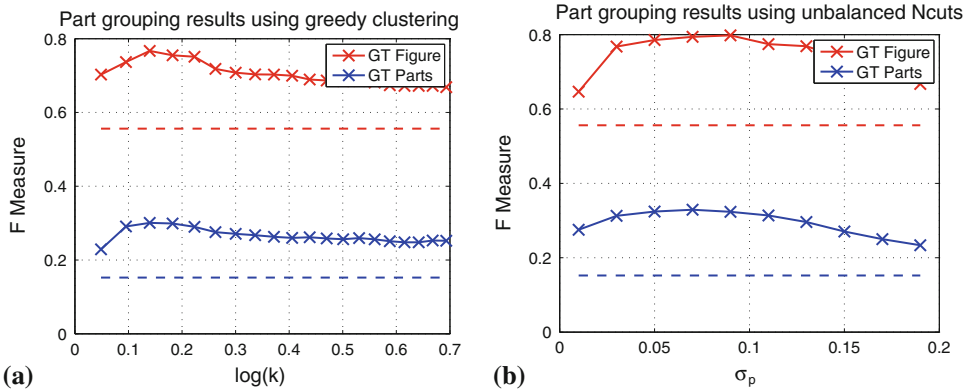


Fig. 10 Part grouping performance. **a** Part grouping using the greedy method in Felzenszwalb and Huttenlocher (2004), **b** Part grouping by minimizing unbalanced Ncuts using parametric maxflow. We evaluate both the segmentation accuracy by comparing to segmentation ground truth (*red*) and the part selection accuracy by comparing to object parts ground truth (*blue*). In both **a**, **b**, performance is measured as a function of method parameters (k in Felzenszwalb and Huttenlocher (2004) and σ_p in unbalanced NCuts). *Dotted lines* provide the baselines for the corresponding evaluation tasks, measuring performance when all the detected parts are selected

part selection stage does consider part redundancy and helps improving part detection precision.

We also provide quantitative evaluation for part grouping. Given a user-specified parameter setting (k in Eq. 5, Felzenszwalb and Huttenlocher 2004), our first, greedy method groups the parts into multiple disjoint clusters. Our second, unbalanced Ncuts-based approach based on parametric maxflow, generates potentially overlapping clusters of parts given parameters α_p (which we fix at 1 for this experiment) and σ_p (that was used to convert part attachment distance into similarity). Not using for now the part selection step, we compare the two part clustering approaches on the tasks of figure/ground segmentation and part grouping. In the first task, our goal was to select a set of parts that covers the object as closely as possible. The second task is motivated by generic object recognition, where correct part groups would be needed to index into a dataset of objects. Here it is less important to achieve a good pixel-wise covering of the object rather than obtain largest possible groups of parts mapping to parts on the object.

Given an image, both methods return multiple clusters of parts corresponding to group hypotheses that would be used for high-level tasks. What is important is that at least one of these clusters corresponds to an object of interest. Thus, for each image we first compute the F measure using the ground truth for both the figure/ground segmentation ("GT-Figure" in Fig. 10 to evaluate segmentation and ground truth for object parts "GT-Parts" in Fig. 10) to evaluate part grouping. We chose the solution with the best F measure for each image. We average the best per-image F measures across all images, giving us a mean F measure for each parameter setting for the two methods. Figure 10 shows the performance of both methods as a function of their parameters. Notice that the unbalanced Ncuts approach achieves slightly better performance. Our main motivation is to establish how usable are the selected parts for generic object recognition. Nevertheless, since we do get a figure/ground segmentation as a by-product of part selection, we performed further comparison of our method to CPMC, a generic segmentation method of Carreira and Sminchisescu (2012). For this comparison, we selected the top 50 ranked segments in CPMC as on average our method returns less that 50 different part groups. The average best per-image F-measure of CPMC is 0.8521, while for both our grouping methods it is close to 0.8. As expected, our method performs slightly worse in this task as it was not designed for segmentation, yet it is not far off from a state-of-the-art segmentation method that was particularly designed for figure/ground segmentation.

6 Limitations and Future Work

A number of limitations of the current framework will be addressed in future research. We have shown that successful part recovery strongly depends on our ability to obtain good medial disc approximations. To improve the quality of medial point hypotheses, we are currently exploring a more powerful superpixel extraction framework that allows greater control over compactness, along with a multiscale Pb detector. In addition, our current system is restricted to grouping superpixels independently at each scale. Removing this constraint would increase grouping complexity and complicate affinity computation, but will also make our part model more flexible and is therefore a subject of future work. We also intend to relax our linear axis model to include curved shapes; for example, the ellipse model could easily be replaced by a deformable superellipse, and a more sophisticated training set can be defined.

The choice of clustering approach can also limit performance. Both stages of our framework, part recovery and part grouping, rely on only locally optimal clustering methods. This allows us to overcome computational issues in extracting and grouping parts bottom-up, but can result in over- or undersegmentation in both part recovery and part clustering. The issue is of greater concern in the second stage, where affinities provide only a crude guideline to true part attachments. Exploring global optimization techniques may lead to a better solution of this grouping problem. The use of unbalanced Ncuts, instead of our main greedy approach, has already been shown to be a promising step in this direction. In future work, we plan to further address this issue through better optimization and the incorporation of more global constraints, such as closure.

Related to the above issue is the issue of learning superpixel and part affinities. Better affinities will directly improve the overall performance of the system. In the current work, we manually trained a multistage classifier for affinities, first training appearance and shape affinities independently and then combining them together using another classifier. In future work, we will put more emphasis on either training all the stages jointly or bootstrapping the different stages by initially holding out some of the training data and adding it gradually for the higher stages in the classification hierarchy (see Munoz et al. 2010).

Finally, we will strive to improve our part detection and grouping evaluation procedure. What is a good part? This is a difficult question to answer. Ultimately, it is a task specific question. If parts are used for object detection, then their quality is measured by the quality of the object detection results. It is a similar question to asking what is a good segmentation? To evaluate such an algorithm independently of the task it is used for, one either needs some manually defined ground truth or some hard-coded objective function to measure the results. In this paper we opted for the first approach, and manually delineated horse parts. However, the parts in our dataset were annotated by a single user, unlike the BSDS, for example, where multiple users were performing the annotation. Moreover, unlike the case of object segmentation, in part segmentation not all part boundaries contain image information for delineating the parts. As a result, some ground truth part boundaries are quite arbitrary (similar to superpixel boundaries inside an object). For example, what is the correct boundary between the head and the neck? There are no image edges to guide the decision. To account for this ambiguity, both in dataset collection process and the observed image information, in this work we relaxed the evaluation criteria for part detection by setting the overlap threshold to 0.4. In the future, however, a more diverse part dataset with multiple user annotations needs to be acquired.

7 Conclusions

We have presented a constructive approach to detecting symmetric parts at different scales by grouping small compact regions that can be interpreted as deformable versions of maximal disks whose centers make up a skeletal branch. In this way, symmetry is *integrated* into the region segmentation process through a compactness constraint, while region *merging* is driven by symmetry-based affinities learned from training data. Detected parts are assembled into objects by exploiting the regularities of part attachments in supervised training data. The resulting framework can, in principle, recover a skeletal-like decomposition of an object from real images without any prior knowledge of scene content and without figure-ground segmentation.

Acknowledgments We thank David Fleet, Allan Jepson, and James Elder for providing valuable advice as members of the thesis committee. We also thank Yuri Boykov and Vladimir Kolmogorov for providing their parametric maxow implementation. This research was sponsored in part by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-10-2-0060. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either express or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation herein. This work was also supported by the European Commission under a Marie Curie Excellence Grant MCEXT-025481 (Cristian Sminchisescu), CNCSIS-UEFISCU under project number PN II- RU-RC-2/2009 (Cristian Sminchisescu), NSERC (Alex Levinshtein, Sven Dickinson), MITACs (Alex Levinshtein).

References

Biederman, I. (1985). Human image understanding: Recent research and a theory. *Computer Vision, Graphics and Image Processing*, *32*, 29–73.

Biederman, I. (1987). Recognition-by-components: A theory of human image understanding. *Psychological Review*, *94*, 115–147.

Binford, T. O. (1971). *Visual perception by computer*. In: Proceedings, IEEE Conference on Systems and Control. Miami.

Blum, H. A. (1967). Transformation for extracting new descriptors of shape. In W. Wathen-Dunn (Ed.), *Models for the perception of speech and visual form* (pp. 362–380). Cambridge: MIT Press.

Borenstein, E., & Ullman, S. (2002). *Class-specific, top-down segmentation*. In: European Conference on Computer Vision (pp. 109–124).

Brady, M., & Asada, H. (1984). Smoothed local symmetries and their implementation. *International Journal of Robotics Research*, *3*(3), 36–61.

Carreira, J., & Sminchisescu, C. (2010). *Constrained parametric min-cuts for automatic object segmentation*. In: IEEE International Conference on Computer Vision and Pattern Recognition.

Carreira, J. Sminchisescu, C. (2012). *CPMC: Automatic object segmentation using constrained parametric min-cuts*. IEEE Transactions on Pattern Analysis and Machine Intelligence.

Cham, T. J., & Cipolla, R. (1995). Symmetry detection through local skewed symmetries. *Image Vision Computer*, *13*(5), 439–450.

Cham, T. J., & Cipolla, R. (1996). *Geometric saliency of curve correspondences and grouping of symmetric contours*. In: European Conference on Computer Vision (pp. 385–398). Florence.

Connell, J. H., & Brady, M. (1987). Generating and generalizing models of visual objects. *Artificial Intelligence*, *31*(2), 159–183.

Cootes, T. F., Taylor, C. J., Cooper, D. H., & Graham, J. (1995). Active shape models-their training and application. *Computer Vision and Image Understanding*, *61*(1), 38–59.

Crowley, J., & Parker, A. (1984). A representation for shape based on peaks and ridges in the difference of low-pass transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *6*(2), 156–169.

Crowley, J., & Sanderson, A. C. (1987). Multiple resolution representation and probabilistic matching of 2-D gray-scale shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *9*(1), 113–121.

Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision*, *59*(2), 167–181.

Hoffman, D. D., Richards, W., Pentland, A., Rubin, J., & Scheuhammer, J. (1984). Parts of recognition. *Cognition*, *18*, 65–96.

Kass, M., Witkin, A., & Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, *1*(4), 321–331.

Kolmogorov, V., Boykov, Y., & Rother, C. (2007). *Applications of parametric maxflow in computer vision*. In: IEEE International Conference on Computer Vision (pp. 1–8).

Levinshtein, A., Dickinson, S., & Sminchisescu, C. (2009). *Multiscale symmetric part detection and grouping*. In: IEEE International Conference on Computer Vision.

Levinshtein, A., Sminchisescu, C., & Dickinson, S. (2005). *Learning hierarchical shape models from examples*. In: International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition (pp. 251–267).

Levinshtein, A., Sminchisescu, C., & Dickinson, S. (2010). *Optimal contour closure by superpixel grouping*. In: ECCV.

Lindeberg, T. (1996). *Edge detection and ridge detection with automatic scale selection*. In: IEEE International Conference on Computer Vision and Pattern Recognition (pp. 465–470).

Lindeberg, T., & Bretzner, L. (2003). Real-time scale selection in hybrid multi-scale representations. In: Scale-space vol. 2695, (pp. 148–163). Springer LNCS.

Liu, T., Geiger, D., & Yuille, A. (1998). Segmenting by seeking the symmetry axis. In: IEEE International Conference on Pattern Recognition, vol. 2, (pp. 994–998).

Liu, Y., Hel-Or, H., Kaplan, C. S., & Gool, L. V. (2010). Computational symmetry in computer vision and computer graphics: A survey. *Foundations and Trends in Computer Graphics and Vision*, *5*(2), 1–195.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Macrini, D., Dickinson, S., Fleet, D., & Siddiqi, K. (2011). Bone graphs: Medial shape parsing and abstraction. *Computer Vision and Image Understanding*, *115*, 1044–1061.

Macrini, D., Dickinson, S., Fleet, D., & Siddiqi, K. (2011). Object categorization using bone graphs. *Computer Vision and Image Understanding*, *115*, 1187–1206.

Martin, D. R., Fowlkes, C. C., & Malik, J. (2004). Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*, 530–549.

Mikolajczyk, K., & Schmid, C. (2002). *An affine invariant interest point detector*. European Conference on Computer Vision, (pp. 128–142). London: Springer.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., et al. (2005). A comparison of affine region detectors. *International Journal of Computer Vision*, *65*(1–2), 43–72.

Mori, G. (2005). *Guiding model search using segmentation*. In: IEEE International Conference on Computer Vision (pp. 1417–1423).

Mori, G., Ren, X., Efros, A. A., & Malik, J. (2004). *Recovering human body configurations: Combining segmentation and recognition*. In: IEEE International Conference on Computer Vision and Pattern Recognition (pp. 326–333).

Munoz, D., Bagnell, J. A., & Hebert, M. (2010). *Stacked hierarchical labeling*. In: ECCV.

Pelillo, M., Siddiqi, K., & Zucker, S. (1999). Matching hierarchical structures using association graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *21*(11), 1105–1120.

Pentland, A. (1986). Perceptual organization and the representation of natural form. *Artificial Intelligence*, *28*, 293–331.

Pentland, A. P. (1990). Automatic extraction of deformable part models. *International Journal of Computer Vision*, *4*(2), 107–126.

Ponce, J. (1990). On characterizing ribbons and finding skewed symmetries. *Computer Vision, Graphics and Image Processing*, *52*(3), 328–340.

Ren, X., & Malik, J. (2003). *Learning a classification model for segmentation*. In: IEEE International Conference on Computer Vision (pp. 10–17).

Saint-Marc, P., Rom, H., & Medioni, G. (1993). B-spline contour representation and symmetry detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *15*(11), 1191–1197.

Sala, P., & Dickinson, S. (2008). *Model-based perceptual grouping and shape abstraction*. In: Proceedings, Sixth IEEE Computer Society Workshop on Perceptual Organization in Computer Vision.

Sala, P., Dickinson, S. (2010). *Contour grouping and abstraction using simple part models*. In: Proceedings, European Conference on Computer Vision (ECCV). Crete.

Sclaroff, S., & Liu, L. (2001). Deformable shape detection and description via model-based region grouping. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*(5), 475–489.

Sebastian, T., Klein, P., & Kimia, B. (2004). Recognition of shapes by editing their shock graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(5), 550–571.

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(8), 888–905.

Shokoufandeh, A., Bretzner, L. D., Demirci, M. F., Jönsson, C., & Dickinson, S. (2006). The representation and matching of categorical shape. *Computer Vision and Image Understanding*, *103*(2), 139–154.

Shokoufandeh, A., Macrini, D., Dickinson, S., Siddiqi, K., & Zucker, S. W. (2005). Indexing hierarchical structures using graph spectra. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(7), 1125–1140.

Shokoufandeh, A., Marsic, I., & Dickinson, S. (1999). View-based object recognition using saliency maps. *Image and Vision Computing*, *17*(5–6), 445–460.

Siddiqi, K., Shokoufandeh, A., & Dickinson, S. J. Y. S. W. Z. (1999). Shock graphs and shape matching. *International Journal of Computer Vision*, *35*, 13–32.

Siddiqi, K., Zhang, J., Macrini, D., Shokoufandeh, A., Bioux, S., & Dickinson, S. (2008). Retrieving articulated 3-d models using medial surfaces. *Machine Vision and Applications*, *19*(4), 261–275.

Stahl, J., & Wang, S. (2008). Globally optimal grouping for symmetric closed boundaries by combining boundary and region information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(3), 395–411.

Ylä-Jääski, A., & Ade, F. (1996). Grouping symmetrical structures for object segmentation and description. *Computer Vision and Image Understanding*, *63*(3), 399–417.