

Optimal Contour Closure by Superpixel Grouping

Alex Levinshtein¹, Cristian Sminchisescu², and Sven Dickinson¹

¹ University of Toronto
{babalex,sven}@cs.toronto.edu

² University of Bonn
cristian.sminchisescu@ins.uni-bonn.de

Abstract. Detecting contour closure, i.e., finding a cycle of disconnected contour fragments that separates an object from its background, is an important problem in perceptual grouping. Searching the entire space of possible groupings is intractable, and previous approaches have adopted powerful perceptual grouping heuristics, such as proximity and co-curvilinearity, to manage the search. We introduce a new formulation of the problem, by transforming the problem of finding cycles of contour fragments to finding subsets of superpixels whose collective boundary has strong edge support in the image. Our cost function, a ratio of a novel learned boundary gap measure to area, promotes spatially coherent sets of superpixels. Moreover, its properties support a global optimization procedure using parametric maxflow. We evaluate our framework by comparing it to two leading contour closure approaches, and find that it yields improved performance.

1 Introduction

One of the key challenges in perceptual grouping is computing contour closure, i.e., linking together a set of fragmented contours into a cycle that separates an object from its background. What makes the problem particularly hard is the intractable number of cycles that may exist in the contours extracted from an image of a real scene. Early perceptual grouping researchers [1] identified a set of nonaccidental contour relations, such as symmetry, parallelism, collinearity, co-curvilinearity, etc., that can be used to link together causally related contours. Such nonaccidental grouping rules can serve as powerful heuristics to help manage the complexity of greedily searching for a contour closure that is unlikely to have arisen by chance [2,3]. However, the space of possible closures is still overwhelming, particularly when one allows larger and larger boundary gaps in a closure. Finding an optimal solution is intractable without somehow reducing the complexity of the problem.

In this paper, we introduce a novel framework for efficiently searching for an optimal closure. Fig. 1 illustrates an overview of our approach. Given an image of extracted contours (Fig. 1(a)), we begin by restricting contour closures to pass along boundaries of superpixels computed over the contour image (Fig. 1(b)).

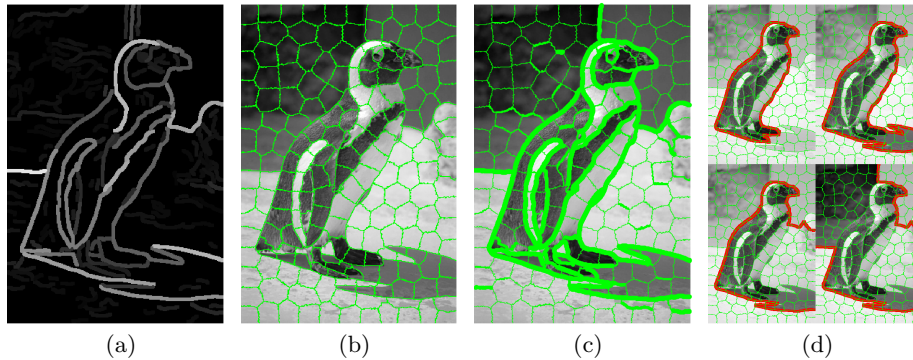


Fig. 1. Approach Overview: (a) contour image – while we take only contours as input, we will overlay the original image in the subsequent figures for clarity; (b) superpixel segmentation, in which superpixel resolution is chosen to ensure that target boundaries are reasonably well approximated by superpixel 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 is inversely proportional to gap); (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, top to bottom.

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 border has strong contour support in the contour image; the assumption we make here 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 needs a mechanism to prefer 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 [4] 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 ones; and 3) introduces a novel, learned gap function measuring the agreement between the boundary of the selection and image contours. 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, their orientation, and their curvature (Fig. 1(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 and final 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 [5] to yield the global optimum. The optimal solution yields the largest set of superpixels bounded by contours that have the least gaps (Fig. 1(d)). Moreover, parametric maxflow will yield the top k solutions (see [6], 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).

In the following sections, we begin by reviewing related work on contour closure (Sec. 2). Next, in Sec. 3, we introduce our problem formulation that transforms the problem of finding optimal cycles of contour fragments into the problem of finding an optimal subset of superpixels. It is here that our cost function is described. In Sec. 4, we describe our process for learning our gap function from training data, and in Sec. 5, we present an efficient procedure for finding the global minimum of our cost function using parametric maxflow. In Sec. 6, we evaluate our framework, comparing it to two competing approaches for computing closure, and discuss the strengths and weaknesses of our approach. We also illustrate the third important role of our superpixel reformulation of providing an appropriate scope over which appearance can be analyzed, and show how our grouping framework can easily be augmented to include both contour and region information. Finally, in Sec. 7, we draw conclusions and outline our plans for future work.

2 Related Work

Detecting closed contours in an image has been addressed by many researchers in different ways. One possible taxonomy for categorizing related work is based on the nature of the prior information used to constrain the grouping process. We will stop short of reviewing methods which assume object-level priors, for it is unclear how to make such methods scale up to very large databases. Instead, we focus on methods that make no assumptions about scene content, although as we will see, many make assumptions about the nature of parts that make up the objects in the scene. In fact, some methods incorporate low-, mid-, and high-level shape priors, as exemplified by Ren et al. [7]. We will also stop short of reviewing methods focused solely on contour completion, e.g., Ren et al. [8] and Williams and Jacobs [9], although the regularities exploited by such approaches can clearly play a powerful role in detecting closure.

Many researchers have exploited the classical Gestalt cues of parallelism and symmetry to group contours. Lowe's [10] early work on perceptual grouping was one of the first to develop a computational model for parallelism, collinearity, and proximity. Many computational models exist for symmetry-based grouping, including Brady and Asada [11], Cham and Cipolla [12], Saint-Marc et al. [13], Ylä-Jääski and Ade [14], and more recently, Stahl and Wang [15]. One significant challenge faced by these systems is the complexity of pairwise contour grouping to detect symmetry-related contour pairs. Levinshtein et al. [16] attempt to overcome this computational complexity limitation by constraining the

symmetric parts to be collections of superpixels. We will draw on this idea of grouping superpixels, but will focus on the more generic cue of closure.

Further down the spectrum of prior knowledge are methods based on weaker shape priors than parallelism and symmetry. For example, Jacobs [17] uses convexity as well as gap to extract closed contours by grouping straight line segments. A less restrictive measure is that of compactness, which can be attained by normalizing the gap by area (Estrada and Jepson [2,3], Stahl and Wang [4]). Some measure of internal homogeneity can also be used (Estrada and Jepson [3], Stahl and Wang [4]), provided that the inside of the region is easily accessible.

Finally we come to the most general methods that compute closure using only weak shape priors, such as continuity and proximity. The most basic closure-based cost function uses a notion of boundary gap, which is a measure of missing image edges along the closed contour. Elder and Zucker [18] model the probability of a connection between two adjacent contour fragments, and find contour cycles using a shortest path algorithm. Wang et al. [19] optimize a measure of average gap using the ratio cut approach. However, a measure based purely on the total boundary gap is insufficient for perceptual closure, and Elder and Zucker [20] argue that the distribution of gaps along the contour is also very important. Williams and Hanson [21] addressed the problem of perceptual completion of occluded surfaces, formulated as the problem of computing a labeled knot-diagram representing a set of occluded surfaces from observed image contours. While formulated as an elegant combinatorial optimization problem, for which an optimal solution was available, the approach was not tested on real scenes.

All the above methods suffer from the high complexity of choosing the right closure from a sea of contour fragments. To cope with this complexity, they either resort to heuristics to prune the search (e.g., [17]) or constrain the search space by other means (e.g., restricting the closure to alternating gap/non-gap cycles [4]). Zhu et al. [22] propose to solve this hard grouping problem by embedding the edge fragments into polar coordinates such that closed contours correspond to circles in that space; however, their goal is to better detect object contours, and they stop short of grouping the contours into closed boundaries. The method of Jermyn and Ishikawa [23] is perhaps the closest to our work. Similar to [19,4], they minimize closure costs using ratio cuts, but unlike [19,4] who operate on contour fragments, [23] works directly with pixels in a 4-connected image grid. It enables the authors to minimize many different closure costs (including our own) by globally minimizing ratio cuts in a simply connected planar graph. However, individual pixels provide poor scope for gap computation. In contrast, our superpixels not only provide greater scope for gap computation (which, in our case, is learned), they provide greater scope for the incorporation of internal appearance-based affinity. Moreover, while their solution is optimal, it does not provide a set of optimal solutions that capture closures at multiple scales.

In this paper, our goal is to find closed contour groups in an efficient manner. Drawing on [16], we use superpixels to constrain the search space of the resulting closures. Superpixels also provide an easy way to access internal region information (such as region area). Moreover, superpixel boundaries provide better scope

for gap computation, as opposed to most previous methods that linearize the output of an edge detector or operate directly on image pixels. On the optimization side, we show that parametric maxflow [5] can be used not only to recover the global optimum of closure costs similar to that of Stahl and Wang [4] and Jermyn and Ishikawa [23], but can also be used to recover a multiscale set of closure hypotheses.

3 Problem Formulation

As mentioned in Sec. 1, our framework reduces grouping complexity by restricting closure to lie along superpixel boundaries. Given a contour image $I(x, y)$ ¹, we first segment it into N superpixels using a modified version of the superpixel segmentation method of Mori et al. [25] ([25] uses the Pb edge detector [26], while we use globalPb [24]). If we let X_i be a binary indicator variable for the i -th superpixel, the vector \mathbf{X} yields a full labeling of the superpixels of I as figure (1) or ground (0). Recall that our goal will be to select a maximal set of superpixels which have high spatial coherence and whose boundary has strong contour support in the image. Drawing on Stahl and Wang [4], we define our closure cost to be $C(\mathbf{X}) = \frac{G(\mathbf{X})}{A(\mathbf{X})}$, where $G(\mathbf{X})$ is the boundary gap along the perimeter of (the “on” superpixels of) \mathbf{X} , and $A(\mathbf{X})$ is its area. Boundary gap is a measure of the disagreement between the boundary of \mathbf{X} and is defined to be $G(\mathbf{X}) = P(\mathbf{X}) - E(\mathbf{X})$, where $P(\mathbf{X})$ is the perimeter of \mathbf{X} and $E(\mathbf{X})$ is the “edginess” of the boundary of \mathbf{X} . Out of the total number of pixels along the boundary of \mathbf{X} , $P(\mathbf{X})$, edginess is the number of edge pixels, with the edginess of image boundary pixels defined to be 0.

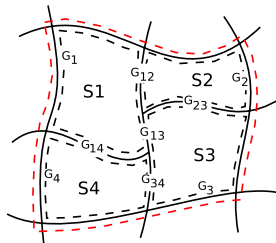


Fig. 2. Boundary gap computation over superpixel graph. $S_1, S_2, S_3,$ and S_4 correspond to superpixels that were selected. G_i and G_{ij} are the boundary gap of superpixel i and the gap on the edge between superpixels i and j respectively. The gap along the boundary of the selection (red) is then $G_{1234} = \sum_{i=1}^4 G_i - 2(G_{12} + G_{13} + G_{14} + G_{23} + G_{34})$.

In order to facilitate the optimization of this cost using an optimal graph cut-based approach (see Sec. 5), we must decompose the cost function into unary and pairwise terms of the variables in X . Let P_i be the perimeter length of

¹ The contour image takes the form of a globalPb image [24].

superpixel i and let P_{ij} be the length of the shared edge between superpixels i and j . Similarly, let E_i be the edginess of superpixel i 's boundary, and E_{ij} be the edginess for the shared boundary between superpixels. Let the superpixel and superpixel edge gaps be $G_i = P_i - E_i$ and $G_{ij} = P_{ij} - E_{ij}$ respectively. Finally, let A_i be the area of superpixel i . Our closure cost becomes:

$$C(\mathbf{X}) = \frac{\sum_i G_i X_i - 2 \sum_{i < j} G_{ij} X_i X_j}{\sum_i A_i X_i} \quad (1)$$

The denominator in the above ratio simply adds the individual areas of all the superpixels that were selected. Normalization by area not only promotes spatial coherence but also promotes compactness; as we shall see in Sec. 6, given two possible paths (with strong edge support) a closure may take, it will prefer a compact path over one with deep concavities. The numerator in the above cost is more complicated. To compute the gap along the perimeter, we first add the individual gaps of all the selected superpixels. However, for selected superpixels that share boundaries, adding individual superpixel gaps would add gaps that are not on the boundary of the selection. For every internal boundary, the gap over that boundary was counted twice (once for each of the superpixels that share the boundary). Therefore, we subtract the gap twice for all internal boundaries. Note that if two superpixels do not have a shared boundary, then both P_{ij} and E_{ij} (and thus G_{ij}) will be 0. Fig. 2 gives an example of gap computation over a simple superpixel graph. In the next section, we introduce our gap measure, and show how it can be learned from training data.

4 Learning the Gap Measure

Most approaches to detecting contour closure (e.g., [4]) typically define gap as simply the length of the missing contour fragments, i.e., the length of that portion of the closure for which no image edges exist. In order to ground our gap measure using image evidence, as well as incorporate multiple contour features for gap computation, we choose to learn the gap from ground truth. Remember from Sec. 3 that for a pair of superpixels i and j , the gap on the edge between them is $G_{ij} = P_{ij} - E_{ij}$. Specifically, if \mathbf{EP}_{ij} is the set of pixels on the superpixel edge (i, j) , then $P_{ij} = |\mathbf{EP}_{ij}|$ and $E_{ij} = \sum_{p \in \mathbf{EP}_{ij}} E_{ij}^p$, where $E_{ij}^p = [P(\mathbf{f}^p) > T_e]$ is an edge indicator for pixel p ($P(\cdot)$ is a logistic regressor and \mathbf{f}^p is a feature vector for the pixel p). T_e is a necessary threshold on the edginess measure. Since the distribution of edges in the training set is not necessarily the same as that for test images, this parameter controls the contribution of weak edges. Decreasing it results in many smaller structures being detected and causes more potential solutions to be generated. We analyze the performance of our method as a function of this parameter in Sec. 6.

Given a pixel p on the superpixel boundary, the feature vector \mathbf{f}^p is a function of both the local geometry of the superpixel boundary and the detected image edge evidence in the neighborhood of the superpixel boundary pixel. This feature vector consists of four features (see Fig. 3(a)):

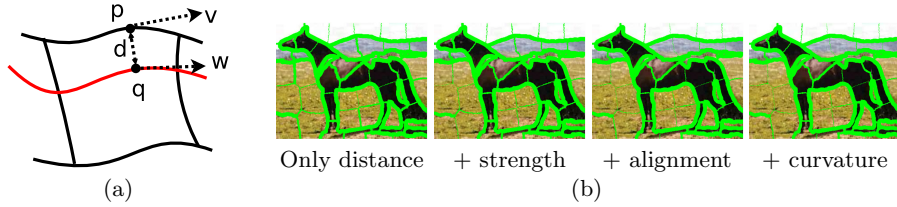


Fig. 3. (a) Contour features for learning the gap measure. Black curves correspond to superpixel boundaries, while the red curve corresponds to detected image edges. The features that are used for edge probability computation at superpixel boundary pixel p are: 1) distance d between p and q , where q is the closest point to p among the detected image edges; 2) image edge strength at q ; 3) the alignment, computed as the absolute value of the cosine of the angle between v and w ; and 4) the smoothness, computed as the squared curvature at p . (b) Effect of different features on gap (ordered left to right). For example, the superpixel edges that cross the legs weaken as alignment is added and the shadow edge on the body weakens as strength is added.

1. Distance to the nearest image edge; closer edges provide stronger evidence.
2. Strength of the nearest image edge; stronger edges provide stronger evidence.
3. Alignment between the tangent to the superpixel boundary pixel and the tangent to the nearest image edge; aligned edges provide stronger evidence.
4. Squared curvature of the superpixel edge at a point.

Given a dataset of images with manually labeled figure/ground, we map the ground truth onto superpixels. Our training set is composed of all the pixels falling on superpixel boundaries and is used to train a logistic classifier over a feature vector f^p . In addition to learning from all four of the above features, we tried learning from subsets of the features. Fig. 3(b) illustrates the effect of incrementally adding more features; the thickness of each superpixel edge corresponds to the average edge probability of its superpixel boundary pixels. Using all four features results in the best performance, in terms of retaining object boundary edges while suppressing other edges.

5 Optimization Framework

It has been known for some time that ratios of real variables that adhere to certain constraints can be minimized globally. Instead of minimizing the ratio $R(x) = \frac{P(x)}{Q(x)}$ directly, one can minimize a parametrized difference $E(x, \lambda) = P(x) - \lambda Q(x)$. It can be shown that the optimal λ corresponds to the optimal ratio $\frac{P(x)}{Q(x)}$. The constraints on the ratio guarantee that the resulting difference is concave and thus can be minimized globally.

In the case of binary variables, ratio minimization can be reduced to solving a parametric maxflow problem. Kolmogorov et al. [5] showed that under certain constraints on the ratio $R(x)$, the energy $E(x, \lambda)$ is submodular and can thus be

minimized globally in polynomial time using min-cuts. Converting our closure cost $C(\mathbf{X})$ in Eqn. 1 to a parametrized difference results in a submodular cost $C(\mathbf{X}, \lambda)$, making the method in [5] applicable for minimizing the ratio $C(\mathbf{X})$.

In fact, the method in [5] does not simply optimize the ratio $R(x)$, but finds all intervals of λ (and the corresponding x) for which the solution x remains constant. The interval boundaries are called breakpoints, and while the smallest breakpoint λ_0 corresponds to the optimal ratio $R(x)$, consecutively larger breakpoints $\lambda_1, \lambda_2, \dots$ are also related to ratio optimization. Kolmogorov et al. show that the optimal solution x^* of $E(x, \lambda)$ in the interval $[\lambda_i, \lambda_{i+1}]$, is also an optimal solution of $\min_{Q(x) \geq T} R(x)$, where $T = Q(x^*)$. In case of optimizing the closure cost in Eqn. 1, using parametric maxflow results in a multiscale set of optimal closure solutions under increasing area thresholds.

The method in [5] can be exponential if the number of breakpoints is exponential, but is polynomial for obtaining a global optimum. In our experiments, a solution is obtained in a fraction of a second for a superpixel graph of 200 superpixels, as there are typically less than 10 breakpoints.

6 Evaluation

We compare our work, which we refer to as *superpixel closure* (SC), to two other contour grouping methods: Estrada and Jepson (EJ) [3] and a version of ratio contours (RRC) from Stahl and Wang [4]. We provide a qualitative evaluation on various images (see Fig. 6), as well as a quantitative evaluation on two datasets, including the Weizmann Horse Database (WHD) [27] and the Weizmann Segmentation Database (WSD) [28]. Learning the gap measure (Sec. 4) is accomplished on the first 30 images from WHD. For testing, we use 170 additional images from WHD and all 100 images from WSD.

6.1 Quantitative Evaluation

For a quantitative evaluation of the results, we use the **F-measure**, $F = \frac{2RP}{R+P}$, where R and P are recall and precision, respectively, of the solution relative to the ground truth. Specifically, if A is the set of pixels corresponding to the solution and A_{gt} is the ground truth, then $R = \frac{|A \cap A_{gt}|}{|A_{gt}|}$ and $P = \frac{|A \cap A_{gt}|}{|A|}$. Given K solutions, we select the solution with the best F-measure relative to the ground truth. We average the “per-image” F-measure for all the images (and three ground truth segmentations in WSD) in a dataset and report the result.

Fig. 4 shows the results of the three methods for increasing values of K . We chose the best parameters for all three algorithms and fixed them for the entire experiment. For EJ, we used a Normalized Affinity Threshold (τ_{affty}) of 0.01, with the line segments generated by fitting the globalPb output. For RRC, we used $\lambda = 0$ and $\alpha = 1$. Here, we could not give the algorithm globalPb-based line segments, and thus use the method’s own line segments generated from a Canny edge response. For our method, we fixed the number of superpixels to 200 and set $T_e = 0.05$, giving us best performance at the high range of K .

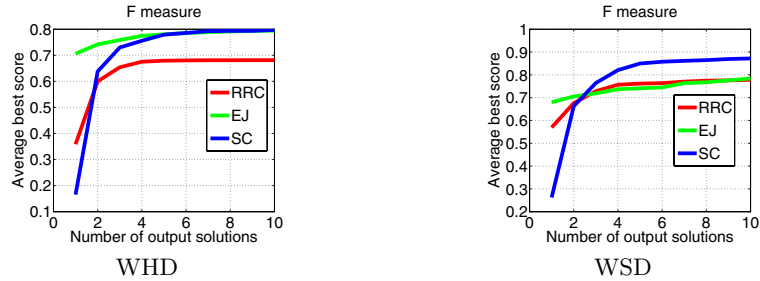


Fig. 4. Quantitative results. We compare our results (SC) to two other algorithms: Estrada and Jepson [3] (EJ) and Ratio Contours [4] (RRC).

Since the resulting solutions can be thought of as shape hypotheses for object recognition, we believe that the performance for some reasonably small value of $K > 1$ is more important than aiming to obtain a single best contour ($K = 1$)². For $K = 10$, SC (EJ, RRC) obtains an average F-measure of 79.72% (79.44%, 68.13%) on WHD and 87.19%³ (78.44%, 77.82%) on WSD.

We outperform the competing approaches on both datasets for a setting of $K = 10$ (obtaining a comparable performance to EJ on the horses dataset), which we attribute to the superpixel formulation, as well as the optimal closure finding method in our framework. On the WHD, both SC and RRC perform significantly worse than on WSD, while EJ performs the same. This is likely due to the lower compactness of objects in the horse dataset (average isoperimetric ratio of 0.15, compared to 0.4 in WSD). Moreover, in many images there is a more compact path that includes the gap between the horse’s legs due to shadow or ground edges. In addition, a significant number of images in the horse dataset have a picture frame boundary around the image. These boundaries provide the largest and most compact solutions, and are therefore found by SC instead of finding the horse. Finally, an interesting fact is that EJ still performs well on the horse dataset (unlike SC and RRC). This is most likely due to its reliance on internal appearance, which is definitely homogeneous in the case of horses.

Since T_e is set so low, our performance is poor for low values of K , but it is better for high K ’s. Fig. 5 shows the change in performance of our algorithm as we change the number of superpixels and vary T_e . Fig. 5(a) illustrates that, in general, higher superpixel density results in a very marginal performance gain for large values of K , while for low values of K , coarser superpixel segmentations prevent very small objects from being detected. Increasing the threshold T_e (Fig. 5(b)) reduces the detection of small objects and improves performance at

² SC can be tuned (see Fig. 5) to perform better for $K = 1$ at a small expense of performance for higher K ’s.

³ For WSD, there are three ground truth segmentations per image. If we instead choose the closest of the three ground truth segmentations per image (as opposed to taking the average), our score on WSD improves to 88.76%.

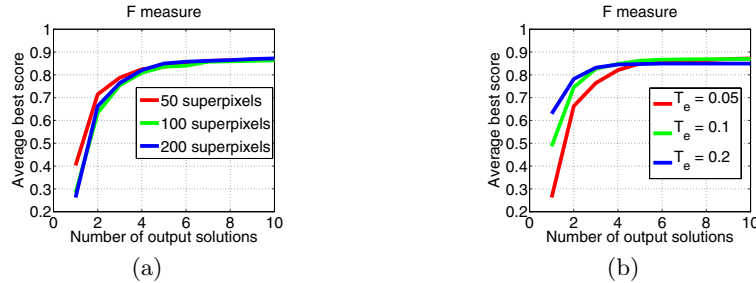


Fig. 5. Varying the parameters of our method. (a) Varying the number of superpixels for a fixed edge threshold $T_e = 0.05$. (b) Varying the edge threshold T_e for a fixed number of superpixels (200).

the low range of K , but also hurts the detection of objects with weak edges and thus results in slightly poorer performance at the high range of K .

We also compare the running times of the three methods on WSD (average image size of 300×290 pixels). On a 2.6GHz Dual Core Intel CPU with 4GB of RAM, setting the methods to retrieve $K = 10$ best contours, the average running times per image are: SC (not including edge detection and superpixel extraction) – 1.3 sec, EJ (not including edge detection) – 23 sec, RRC – 59 sec.

6.2 Qualitative Evaluation

In addition to the quantitative evaluation, we also provide a qualitative evaluation of our method by testing it on images from the two datasets, as well as other images obtained from the internet. Fig. 6 illustrates the performance of our method compared to the two competing approaches⁴. We manually select the best of 10 solutions for each method. Notice that the detected contours in our framework lie closer to the true object contours since the superpixel edges, even in the presence of a gap, lie closer to object edges than the linearized contours detected by the other algorithms. We pleasantly observed that our framework is not constrained to obtain compact solutions as is usually the case when one is normalizing perimeter by area. This is clearly visible in the image of a spider, where very thin legs are segmented since that represents the best closure solution. However, this is not always the case, for if there is a more compact contour that is not losing on gap, it will be preferred. This is the reason for the filled gap between the horse's legs or the filled gap between the carriage's wheels in the first two images. Note that for the horse image, EJ obtains a better solution by relying on the homogeneous appearance inside the horse. Finally, our method relies on superpixels to oversegment the object. We still detect thin structures, such as the spider's legs, if good superpixels were found due to strong image edges. For weaker edges, however, thin structures are harder to capture (the bat of the baseball player, for example).

⁴ Supplementary material contains the results of SC for all the images in both datasets.

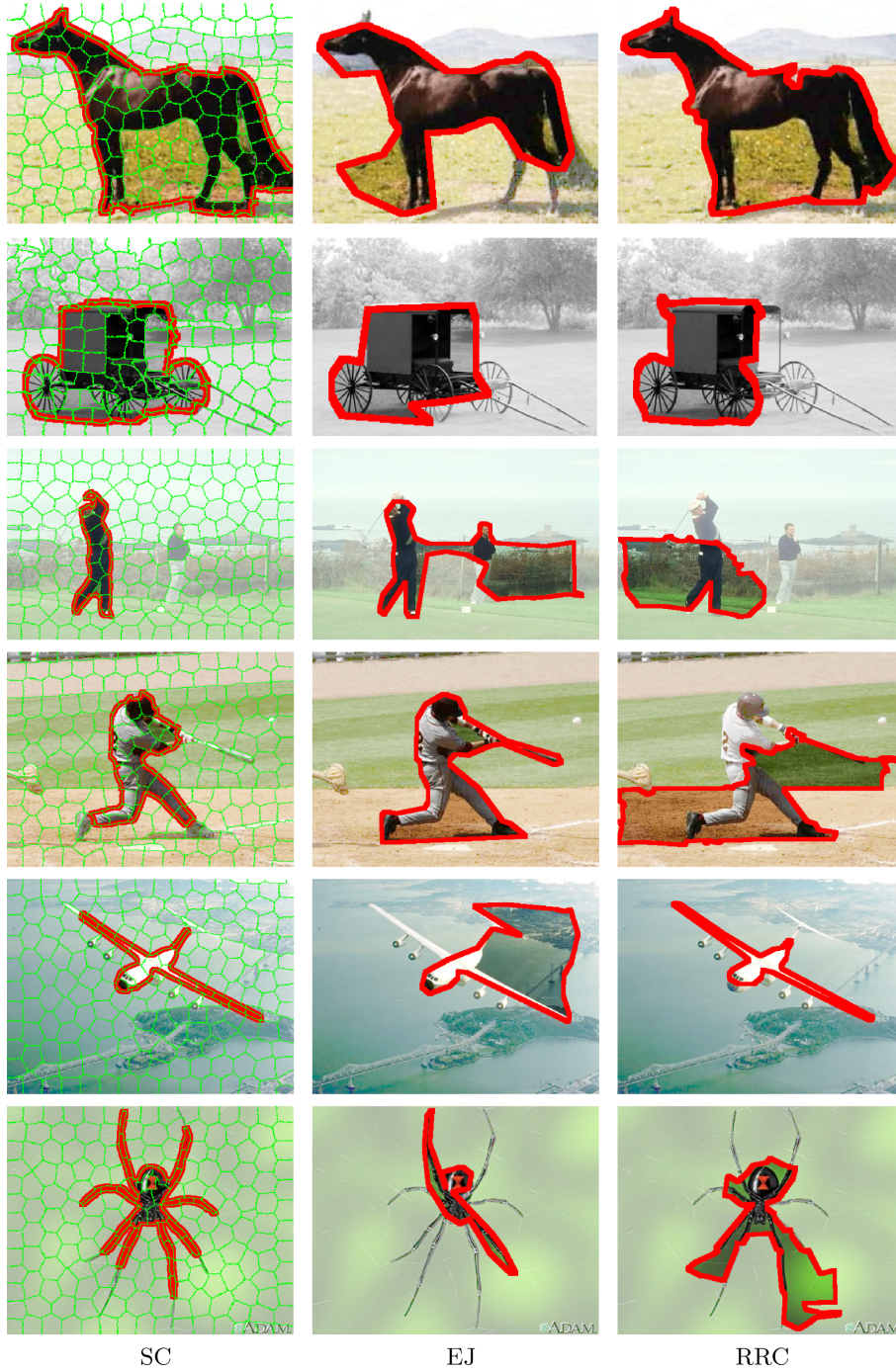


Fig. 6. Qualitative results. We compare our results (left) to two other algorithms: Estrada and Jepson [3] (middle) and Ratio Contours [4] (right).

6.3 Using Internal Homogeneity

As mentioned in Sec. 1, our superpixel formulation also facilitates the incorporation of appearance information, when it is both available and appropriate. The cost function in Eqn. 1 can be easily modified to incorporate a term which reflects the degree to which adjacent superpixels *inside* the superpixel selection, i.e., inside the closed contour, have high affinity. Assuming that we are given an affinity matrix W , such that W_{ij} is the affinity between two superpixels i and j , we modify our closure cost to be:

$$C_{affty}(\mathbf{X}) = \frac{\sum_i G_i X_i - 2 \sum_{i < j} G_{ij} X_i X_j}{\sum_{i < j} W_{ij} X_i X_j} \quad (2)$$

Compared to the cost in Eqn. 1, the numerator remains the same while the denominator changes to an internal homogeneity measure instead of the total object area. Minimizing this ratio results in minimizing the gap while maximizing the total affinity between the selected superpixels. Fig. 7 shows an example where better results were achieved by exploiting appearance homogeneity.

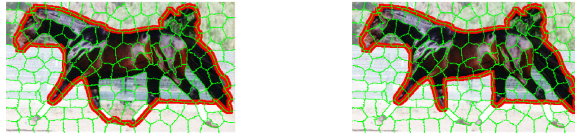


Fig. 7. Using internal appearance homogeneity. For objects with strong internal homogeneity of appearance, optimizing the cost in Eqn. 2 is better (right) than optimizing the cost in Eqn. 1 (left). Note that the gap between the horse's legs was not included on the right due to its heterogeneous appearance w.r.t. the rest of the horse.

6.4 Multiple Superpixel Scales

Though it might seem that the more superpixels we use, the better SC will perform, it is not always so. As seen in Fig. 5(a), coarser superpixel scales constrain the solution more and thus perform better for low values of K . However, there is one additional advantage of using coarser superpixel scales. Since our superpixel algorithm does not produce hierarchical superpixels (since new superpixel boundaries may be introduced from finer to coarser scales), it is possible to occasionally have less undersegmentation at coarser scales. Fig. 8 illustrates a situation where an object was segmented better at a coarser scale and consequently detected by SC.

We tried a simple multiscale version of SC where we merge the results from all scales. Specifically, we run SC at four superpixel scales, obtaining 25, 50, 100, and 200 superpixels for each image. Setting $K = 10$ for each scale results in 40 solutions once the results are merged together. Since the performance of SC for a given scale does not significantly vary for $K > 10$, we do not select 10 of 40 solutions for the multiscale version, but instead retain all 40. Using the multiscale version increases the performance on WSD from 87.19% to 89.53%.

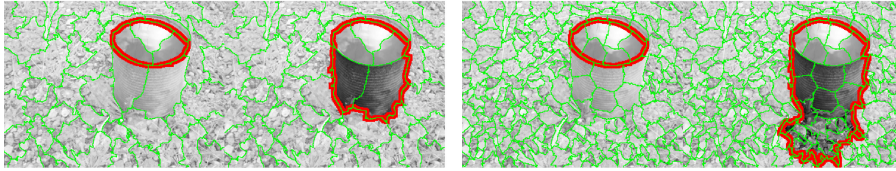


Fig. 8. Multiscale results. Choosing the $K = 2$ top solutions yields better results in the case of 50 superpixels (left) than in the case of 200 superpixels (right).

7 Conclusions

Our reformulation of the problem of finding cycles of contours as the problem of finding spatially coherent subsets of superpixels, whose collective boundary has strong image edge evidence yields an optimal framework for closure detection that compares favorably with two leading prior approaches. While superpixels provide an ideal scope for learning a gap measure from training data, they offer a number of additional advantages that we are currently exploring. We plan to use superpixel junctions to learn an affinity measure between pairs of superpixels that are both inside and adjacent to the boundary. Such an affinity measure can encode a learned measure of continuity and T-junction, and could significantly strengthen our cost function. Superpixels also provide a convenient mechanism for incorporating appearance information, if appropriate and if available. For example, if the object was known to be homogeneous in appearance, our modified cost function can easily incorporate such a prior, as discussed in Sec. 6.3. Our framework is flexible, and can easily accommodate many classical non-accidental regularities. In the future, we also plan to pursue a more elegant coarse-to-fine framework for finding contour closure using multiple superpixel scales.

Acknowledgements

We thank Allan Jepson for discussion about closure cost functions and optimization procedures, and Yuri Boykov and Vladimir Kolmogorov for providing their parametric maxflow implementation. This work was supported in part by the European Commission under a Marie Curie Excellence Grant MCEXT-025481 (Cristian Sminchisescu) and NSERC (Alex Levinshtein, Sven Dickinson).

References

1. Wertheimer, M.: Laws of organization in perceptual forms. In: Ellis, W. (ed.) *Source Book of Gestalt Psychology*. Harcourt, Brace (1938)
2. Estrada, F.J., Jepson, A.D.: Perceptual grouping for contour extraction. In: *ICPR*, pp. 32–35 (2004)
3. Estrada, F.J., Jepson, A.D.: Robust boundary detection with adaptive grouping. In: *POCV*, p. 184 (2006)

4. Stahl, J., Wang, S.: Edge grouping combining boundary and region information. *IEEE Transactions on Image Processing* 16, 2590–2606 (2007)
5. Kolmogorov, V., Boykov, Y., Rother, C.: Applications of parametric maxflow in computer vision. In: *ICCV* (2007)
6. Carreira, J., Sminchisescu, C.: Constrained parametric min-cuts for automatic object segmentation. In: *CVPR* (2010)
7. Ren, X., Fowlkes, C.C., Malik, J.: Cue integration in figure/ground labeling. In: *NIPS*, pp. 1121–1128 (2005)
8. Ren, X., Fowlkes, C.C., Malik, J.: Scale-invariant contour completion using conditional random fields. In: *ICCV*, pp. 1214–1221 (2005)
9. Williams, L.R., Jacobs, D.W.: Stochastic completion fields: a neural model of illusory contour shape and salience. In: *ICCV*, p. 408 (1995)
10. Lowe, D.G.: *Perceptual Organization and Visual Recognition*. Kluwer Academic Publishers, Norwell (1985)
11. Brady, M., Asada, H.: Smoothed local symmetries and their implementation. *IJRR* 3, 36–61 (1984)
12. Cham, T.J., Cipolla, R.: Geometric saliency of curve correspondances and grouping of symmetric comntours. In: Buxton, B.F., Cipolla, R. (eds.) *ECCV 1996*. LNCS, vol. 1065, pp. 385–398. Springer, Heidelberg (1996)
13. Saint-Marc, P., Rom, H., Medioni, G.: B-spline contour representation and symmetry detection. *PAMI* 15, 1191–1197 (1993)
14. Ylä-Jääski, A., Ade, F.: Grouping symmetrical structures for object segmentation and description. *CVIU* 63, 399–417 (1996)
15. Stahl, J.S., Wang, S.: Globally optimal grouping for symmetric closed boundaries by combining boundary and region information. *PAMI* 30, 395–411 (2008)
16. Levinshtein, A., Dickinson, S., Sminchisescu, C.: Multiscale Symmetric Part Detection and Grouping. In: *ICCV* (2009)
17. Jacobs, D.W.: Robust and efficient detection of salient convex groups. *PAMI* 18, 23–37 (1996)
18. Elder, J.H., Zucker, S.W.: Computing contour closure. In: Buxton, B.F., Cipolla, R. (eds.) *ECCV 1996*. LNCS, vol. 1065, pp. 399–412. Springer, Heidelberg (1996)
19. Wang, S., Kubota, T., Siskind, J.M., Wang, J.: Salient closed boundary extraction with ratio contour. *PAMI* 27, 546–561 (2005)
20. Elder, J., Zucker, S.: A measure of closure. *Vision Research* 34, 3361–3369 (1994)
21. Williams, L.R., Hanson, A.R.: Perceptual completion of occluded surfaces. *CVIU* 64, 1–20 (1996)
22. Zhu, Q., Song, G., Shi, J.: Untangling cycles for contour grouping. In: *ICCV* (2007)
23. Jermyn, I., Ishikawa, H.: Globally optimal regions and boundaries as minimum ratio weight cycles. *PAMI* 23, 1075–1088 (2001)
24. Maire, M., Arbelaez, P., Fowlkes, C., Malik, J.: Using contours to detect and localize junctions in natural images. In: *CVPR* (2008)
25. Mori, G., Ren, X., Efros, A.A., Malik, J.: Recovering human body configurations: Combining segmentation and recognition. In: *CVPR*, pp. 326–333 (2004)
26. Martin, D.R., Fowlkes, C.C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI* 26, 530–549 (2004)
27. Borenstein, E., Ullman, S.: Class-specific, top-down segmentation. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002*. LNCS, vol. 2351, pp. 109–124. Springer, Heidelberg (2002)
28. Alpert, S., Galun, M., Basri, R., Brandt, A.: Image segmentation by probabilistic bottom-up aggregation and cue integration. In: *CVPR* (2007)