Contour Box: Rejecting Object Proposals Without Explicit Closed Contours*

Cewu Lu[‡,♯] Shu Liu[†] Jiaya Jia[†] Chi-Keung Tang[§]

[§]The Hong Kong University of Science and Technology [‡]Stanford University

[†]The Chinese University of Hong Kong [♯]Shanghai Jiao Tong University

Abstract

Closed contour is an important objectness indicator. We propose a new measure subject to the completeness and tightness constraints, where the optimized closed contour should be tightly bounded within an object proposal. The closed contour measure is defined using closed path integral, and we solve the optimization problem efficiently in polar coordinate system with a global optimum guaranteed. Extensive experiments show that our method can reject a large number of false proposals, and achieve over 6% *improvement in object recall at the challenging overlap threshold 0.8 on the VOC 2007 test dataset.*

1. Introduction

Object detection benefits from object proposal generation. A reasonable number of object proposals can speed up the detection process and avoids exhaustive sliding window search.

Closed contour is a useful cue in generating object proposals. To find image contours, in [5], the observation that objects with closed boundaries can be discriminated by the norm of gradients in patches was made use of. In [26], a simple box objectness score was proposed to measure the number of edges in the box excluding those who overlap with the box boundary. Albeit effective, these methods do not explicitly output a closed contour, because of the intrinsic challenge that there exist an exponential number of ways for pixels to connect to form closed curves.

This paper explicitly seeks an optimal closed contour for object proposal generation. There are two issues to consider, i.e., the mathematical definition of a closed contour in the context of object proposals, and an efficient optimization algorithm to compute the optimal closed contour given the large number of candidates within a given image window.

The subclass of closed contours we interested in is modeled using the closed path integral subject to the completeness and tightness constraints. We compute the optimal contour by transforming the solution space to the polar coordinate system. A spatial closed contour becomes a continuous path in polar coordinate system. Interestingly, this problem becomes the shortest path searching, which can be solved exactly and efficiently by dynamic programming. The result is guaranteed to be a global optimum with the time cost insignificant compared to naive search and comparison. Every optimized contour is assigned a contour score to reflect the confidence.

Our second contribution is to employ the contour scores in object proposal generation. Unlike previous work, which prefers high-scoring proposals to indicate objectness, we use the contour score to prune non-object proposals. The reason for not accepting high-score proposals is that a closed contour satisfying both the completeness and tightness constraints does not always correspond to a semantic object. On the other hand, if a proposal does not have such a closed contour, it stands a high probability that this is a non-object proposal. Figure 1 shows a number of examples. Our empirical study over 50K proposals shows that 99.1% of low-score windows are regarded as non-object proposals. Based on this finding, we propose a new objectness scheme that rejects proposals unlikely to be an object. At the time of submitting this paper, our method yields an improvement of 6% over the best one regarding object recall with the intersection-over-union (IoU) overlap set at 0.8. Our evaluation is done on the PASCAL VOC dataset [9].

2. Related Work

The goal of generating object proposals is to produce a set of reliable bounding boxes that likely to contain objects. With a significant reduction in the number of candidate boxes, computationally expensive classifiers [21, 23, 10] can be employed in object detection to achieve state-of-the-art detection performance. This pipeline has been widely adopted in object detection, which further motivates research on improving object proposal generation. Previous methods can

*This research is supported by the Research Grant Council of the Hong Kong Special Administrative Region under grant numbers 619711 and 412911.

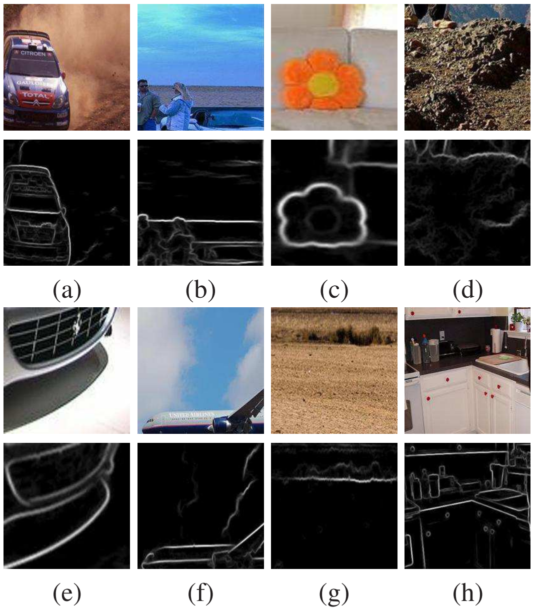


Figure 1. If there is no closed contour near the boundary, the chance that this window contains a complete semantic object is small. A few examples are shown here with their edge maps. Note that (c) is not an object window since it does not tightly enclose the the flower pillow.

be classified into two categories: producing a large number of proposals in the viewpoint of segmentation and the use of objectness ranking.

Selective search [21] is a well-known method in the first category, which merges super-pixels in a greedy manner. When two super-pixels are merged, a bounding box is created to tightly bound them. This process continues until there is only one super-pixel left. In order to generate a number of good bounding boxes, several strategies were proposed for reasonable diversity. The selective search work was extended in [25], where more features are added and weights are learned. In [15], randomness is considered during the merging process. In [4, 8, 19, 24], segmentation is employed while different object proposals are generated by enumerating many foreground-background masks. Their computation cost is generally high.

The method related to ours [16] uses a fixation point to define the polar space for finding the optimal contour where the focus is active visual segmentation. A feedback process is used to establish the relationship between edges and regions, whereas our contour box only uses an edge map. In [12], the structure of super-pixel graph is reused to speed up computation. A fast normalized-cut algorithm was proposed in [2] to generate object proposals by grouping image segments. Most of these methods can generate high-quality segmentation masks. However, without a good objectness score, it is difficult to reduce the large number of boxes into a reasonable number to facilitate subsequent processes.

For the second category, they seek different cues to describe objects. In [1], a number of objectness cues, such as object contour, color, contrast, edge, edge density and salient map are used to score a bounding box. Then the sampled bounding boxes are ranked by the generated objectness scores. This method was modified in [18] by changing the features and classifiers. In [4], proposals were ranked by performing regression between features and their actual overlap with an object.

In [5], a linear classifier was proposed to calculate the objectness score based on gradient features, while in [26] edges were used to calculate the objectness score by counting the number of contours that are wholly contained in a bounding box and removing those who straddle the box boundary. In [17], a category-independent proposal generation method was proposed. In [3], Blaschko *et al*. computed additional objectness features at a low cost based a non-maximal suppression (NMS) scheme on the cascade architecture. Still, the method does not exploit the cue of object contour explicitly. Differently, we discover object contours and score them in an optimization framework.

We also review other methods related to closed curves using dynamic programming in polar coordinate system. In [20], a dynamic programming-based algorithm was proposed to extract multiple paths using constrained expanded trellis (CET) for feature extraction and object segmentation. In [6], Gorce *et al*. proposed dichotomic multiple search (DMS) to find the global minimum as the circular shortest path. While we share some similarity in terms of dynamic programming and polar coordinates, the new completeness and tightness constraints are important and unique to make our method work for proposal generation.

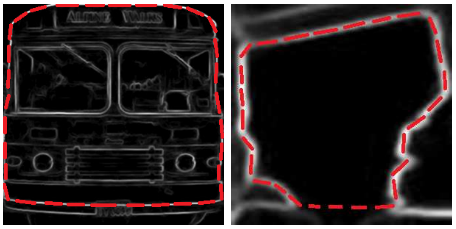
3. Contour Box: Optimizing Closed Contours

In this section, we describe our method to optimize a closed curve that predicts an object silhouette tightly bounded by an image window. We first describe the way to model a closed contour subject to the completeness and tightness constraints. Then an efficient solution is presented to compute it.

3.1. Closed Path Integral

The edge intensity at pixel location $\mathbf{x}$ is denoted by $e(\mathbf{x})$. Our optimized contour is defined as a closed curve. In essence, edge evidence along the curve should be as complete as possible while the curve should be close to the window border. In our implementation, we compute edges using the method of [7].

Completeness constraint An intuitive definition of complete edge evidence requires the mean edge strength along a closed curve to be sufficiently large. However, this definition cannot tell whether salient edges along a closed curve



(a) object (b) non-object

Figure 2. Two closed curves (in red) optimized with and without adjusted edge evidence. Without the adjustment, the mean edge intensities of the curves are similar, which are 0.6054 and 0.5957 respectively. After the adjustment, the mean edge intensities of the right curve drops to 0.3418 while the left remains unchanged.

are missing or not. One example is shown in Figure 2(b), where the mean edge strength along the red curve is reasonably strong. But there is a segment missing in order to form the closed curve. We propose a simple adjustment of the edge map as

$$\phi[y] = \begin{cases} y & y > \tau \\ -\gamma & y \leq \tau \end{cases} \tag{1}$$

where $\tau = 0.001$ and $\gamma = 1$ are constants. Empirically, γ is in range $[0.8, 2]$. This serves to penalize edge intensity below τ or missing edges. The parameter τ is not critical in our experiments.

With the adjusted edge map, we define the completeness condition as maximizing the objective function

$$\frac{\oint_c \phi[e(\mathbf{x})]d\mathbf{x}}{\oint_c 1 d\mathbf{x}}, \tag{2}$$

where c is a closed curve and $\oint_c 1d\mathbf{x}$ is a path length normalization factor. Without this factor, maximizing Eq. (2) would produce a unreasonably long curve that maximizes the total edge values.

Tightness constraint According to the definition of objects, every window should tightly enclose an object. Therefore, we expect the closed curve is near or on window borders. Put differently, we prefer a closed curve whose pixels are reasonably far from centroid of the window.

To this end, we define the normalized distance to the window centroid for every pixel as $u(\mathbf{x}) = \|[2/h, 2/w] \cdot (\mathbf{x} - \mathbf{x_0})\|_2$, where $\mathbf{x_0}$ is the centroid, '$\cdot$' is the point-wise product, and w and h are respectively the width and height of the given image window. We adjust $u(\mathbf{x})$ using a nonlinear function that penalizes non-closed curves as

$$\varphi[y] = \min\{y^2, 0.7\} \tag{3}$$

to update the results. While there exist other non-linear adjustment functions, Eq. (3) is good enough in our experiments. To impose the tightness constraint, we maximize

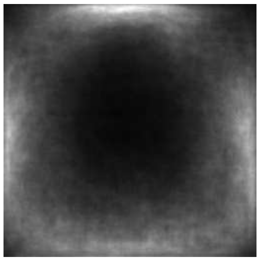


Figure 3. Average map of all object boundaries in the VOC 2012 segmentation dataset. All object images are resized into 256 × 256 before averaging.

$$\frac{\oint_c \varphi[u(\mathbf{x})]d\mathbf{x}}{\oint_c 1 d\mathbf{x}}. \tag{4}$$

Our object proposal generator is category-independent without any presumed shape priors, while the tightness constraint implicitly prefers a round (or elliptical) shape, which is a soft constraint. To justify our choice, Figure 3 shows the *average* map of all the object boundaries in the VOC 2012 segmentation dataset. A rough annulus shape near the boundary can be observed.

Objective function We combine the two terms described above to form the objective function

$$\max_{c \in \mathcal{C}} \frac{1}{\oint_c 1 d\mathbf{x}} \{ \oint_c \phi[e(\mathbf{x})]d\mathbf{x} + \lambda \oint_c \varphi[u(\mathbf{x})]d\mathbf{x} \}, \tag{5}$$

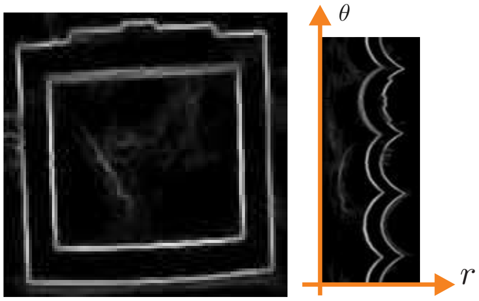
where $\mathcal{C}$ is the set of all closed paths in the window, and λ is a parameter to be set during training. If Eq. (5) outputs a small energy, there is no suitable object contour that is complete and tight.

3.2. Solution in Polar Coordinates

We now describe how to solve the energy function (5). Note the solution space of Eq. (5) is prohibitively large due to the large number of possible closed paths in the set $\mathcal{C}$. Further, due to the normalized factor $\oint_c 1d\mathbf{x}$, popular path optimization algorithms cannot be applied directly.

To address these issues, we map our configuration from the Cartesian coordinate system to the polar coordinate system. An example is illustrated in Figure 4. A closed path in x-y coordinate system is mapped to a continuous one-way path in the polar coordinate system, where the radial coordinate r and angular coordinate θ are obtained with respect to centroid of the image window.

We use a linear interpolation approach to achieve discrete mapping. Our polar coordinate map is stored as a $m \times n$ matrix. That is, we quantize 360° into m bins and radii into n bins. We use $m = 90$ and $n = 50$ in our experiments.



(a) (b)

Figure 4. Mapping a curve from (a) Euclidean coordinates to (b) polar coordinates.

We then rewrite Eq. (5) as

$$\max_{c \in \mathcal{C}} \frac{1}{\oint_c 1 d\mathbf{x}} \{ \oint_c M(\mathbf{x}) d\mathbf{x} \}, \tag{6}$$

where

$$M(\mathbf{x}) = \phi[e(\mathbf{x})] + \lambda \varphi[u(\mathbf{x})]. \tag{7}$$

In polar coordinates, $M(\mathbf{x})$ is denoted as $g(r, \theta)$.

Interestingly, the problem of optimizing our complete and tight closed contour in Eq. (5) is equivalent to seeking the optimal continuous path in polar coordinates from top to bottom in our $m \times n$ matrix. Thus we optimize a continuous path starting at $(1, k)$ in polar coordinates and ending at a point near (m, k), which ensures that the optimized path is closed. We solve the optimization problem of

$$\begin{aligned} E(k, g) = \; & \max_{\mathbf{a}} \{ \sum_{i=1}^{m} g(a[i], i) \} \\ s.t. \;\; & |a[i] - a[i-1]| \leq \delta, \;\; i = 2, \ldots, m \quad (8) \\ & a[1] = k, \;\; |a[m] - k| \leq \delta, \quad (9) \end{aligned}$$

where $\mathbf{a}$ denotes a path and every $a[i]$ is the radial coordinate with angle i. Eq. (8) enforces the constraint that the optimized curve should be continuous in polar coordinates. We set $\delta = 3$ in our experiments. The constraint (9) prescribes that $\mathbf{a}$ connects the starting point to the ending one, since $\mathbf{a}$ is closed.

This problem is a standard shortest path search one, which seeks an optimal path from the source point $(1, k)$ to $\{(m, k - \delta), \ldots, (m, k + \delta)\}$. It can be efficiently solved by dynamic programming where the global optimum exists. The computation amounts to checking n possible starting points, and then selecting the best path

$$\max_{k} E(k, g). \tag{10}$$

By optimizing Eq. (10), we obtain the required path and its energy – the latter is used to reject non-object proposals.

Discussion on the Two Coordinate Systems In our discrete solution, a curve is composed of exactly m segments, represented by m elements in the polar matrix $\mathbf{M}$. Every element in the polar matrix records the mean intensity of the corresponding segment in the Euclidean coordinate system. All curves are normalized in terms of length.

The computation in the polar coordinates is simple. Checking n starting points reduces the time complexity to the order of the matrix resolution. Also, all continuous paths from top to bottom have the *same* length, thus naturally canceling out the length normalization factor $\oint_c 1 d\mathbf{x}$ in the original objective function, which explains why dynamic programming can now be applied. The time complexity of our method is $O(mn^2)$, which is constant w.r.t. the window size.

Notwithstanding, there exist closed curves in the original coordinate system that cannot be mapped to a continuous path in the polar coordinate system. These exceptions mainly arise when centroid of the image window is not enclosed by the optimal curve. But we found this case is rare. In the VOC 2007 training data, 99.3% of the object boundaries enclose centroid of the image window. In our experiments, the difference of solutions respectively produced in the two coordinate systems is insignificant.

3.3. Non-object Indication

As discussed above, our contour score is used as a non-object indicator. We determine the non-object threshold T and label those proposals whose contour scores are lower than T as "non-objects". To this end, we manually label 40,000 proposals as "with contour" and "without closed object contour" respectively on pascal VOC 2007 dataset. Then, we seek the best λ for Eq. (7) and T together in these two categories. Finally, we label proposals whose contour scores are smaller than T as "non-objects".

4. Experiments

Following previous work [5, 15, 26, 11], we evaluate our method on the VOC2007 dataset [9]. We compare our results with those state-of-the-arts [2, 5, 19, 26, 21, 15, 1]. The simple method of computing closed curves using level-set [14] is also compared with, since it is a famous contour generation strategy.

We use the author-released implementations with recommended parameter settings. To evaluate the recall of bounding boxes, we use the Intersection-over-Union (IoU) metric, which computes the intersection of a candidate box and the ground-truth divided by the area of their union. We report the performance at the challenging IoU threshold 0.8.

4.1. Difference in the Two Coordinate Systems

We solve the problem in Eq. (5) in the polar coordinate system. We carry out the following experiments to veri-

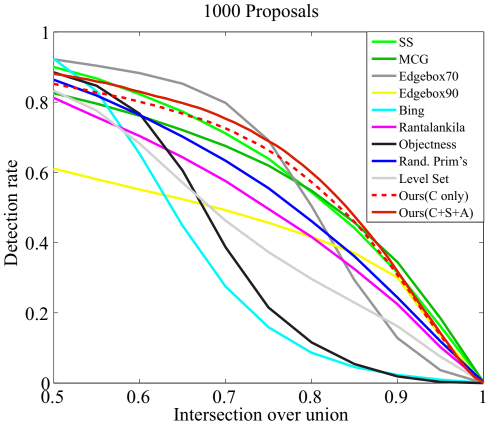


Figure 5. Recall with different IoUs using 1,000 proposals. "ours (C only)" means contour score only. "ours (C+S+A)" is the linear combination of contour score, size and aspect ratio.

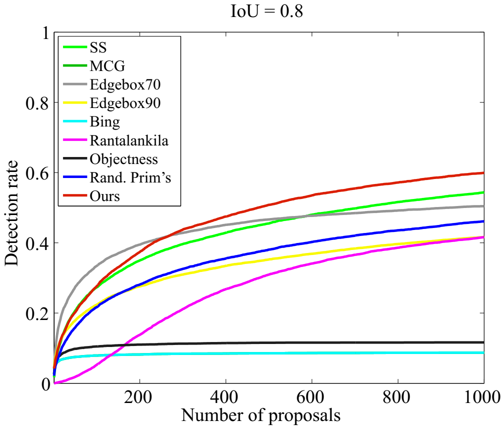


Figure 6. Recall with different proposal numbers at IoU 0.8.

fy that our solution is a good approximation of the original one. We randomly select 20,000 proposals and compute their contour scores in both the Euclidian and polar coordinate systems. Then, we solve the original Eq. (5) by exhaustively searching all possible closed curves. This is very time-consuming and takes about 2 days to finish. We evaluate a ranking distance metric to evaluate the two results, since we are concerned more with the relative score for rejecting non-object proposals. The ranking distance we use is Kendall-tau rank distance, which is defined as

$$\frac{\#\{\text{concordant pair}\} - \#\{\text{discordant pair}\}}{\frac{1}{2}n(n-1)}. \qquad (11)$$

n is the total number of scores (20,000 here). The Kendall-tau rank distance lies in the interval $[-1, 1]$. Normally, a distance with 0.8 indicates that two score lists are very close in terms of element ranking. The distance of the results computed in the two coordinate systems is **0.971**, which indicates that these two sets of results are almost identical.

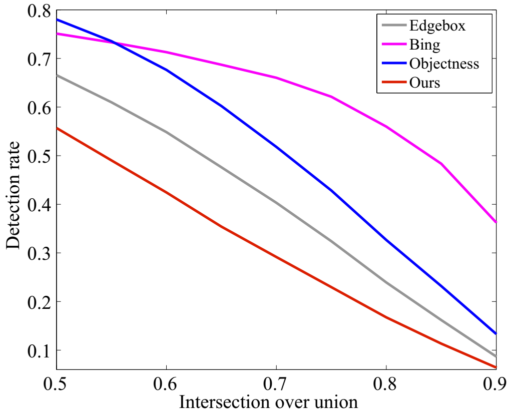


Figure 7. Recall of non-object proposals labeled by different methods (false positive rate).

4.2. Proposal Quality Comparisons

In this section, we empirically show that our Contour Box, which adopts non-object proposal rejection using contour scores, improves object recall and achieves state-of-the-art performance. Following [5, 15, 26, 11], we evaluate our results on the VOC2007 test set, which consists of 4,952 images with bounding box annotations for objects in 20 different categories.

We evaluate our performance using the standard recall-over-IoU performance over 1,000 proposals: every method reports 1,000 bounding boxes and its recall is tested over different IoUs. To produce 1,000 proposals, we first perform the selective search [21] to generate around 10,000 proposals for every image. Then our contour score is used to discard proposals labeled as non-object. The average number of proposals we remove is 6783.3 per image.

Finally, we report 1,000 proposals by randomly sampling from the remaining proposals. To explore the statistics of object boxes, we also combine aspect ratio and size of object boxes by weighting them with the contour score. The 2D weight is learned using linear SVM. Here, object aspect ratio and size are respectively $\frac{h}{w}$, $\frac{\sqrt{hw}}{\sqrt{HW}}$, where h and w are respectively height and width of the bounding box, H and W are respectively height and width of the image.

Figure 5 shows the statistical comparison between our method and SS [21], MCG [2], BING [5], EdgeBox [26], Rantalankila [19], Objectness [1], Prime proposal [15], Rahtu [17], and Blas [3]. We tune the code of Rahtu [17] and Blas [3] and report curves with the best recall at IoU= 0.8.

In [26], a high IoU (0.7) is recommended for object detection. Our method works well with an even higher IoU. We observe about 6% improvement at the challenging overlap threshold 0.8. When evaluated within the IoU range $[0.9, 1]$, our method is still comparable to state-of-the-art

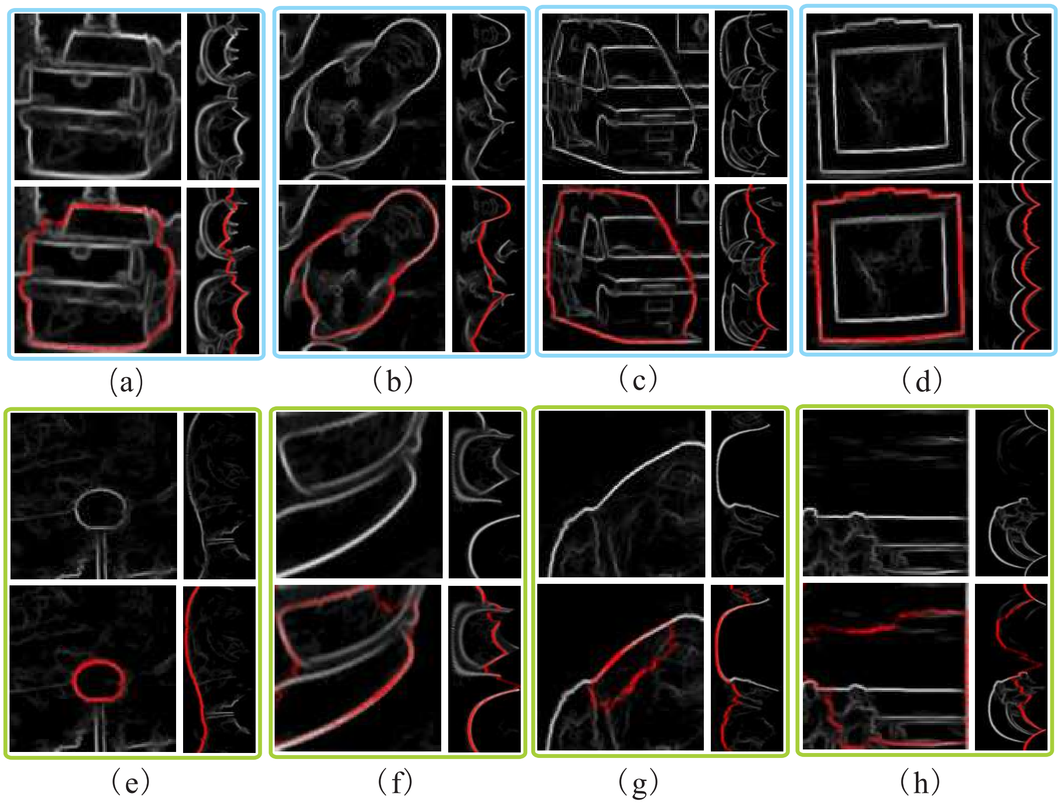


Figure 8. Result examples with optimized closed paths. In every example, the left and right maps are respectively the results in the image and polar coordinate systems, while the top and bottom are the original images and proposals with the optimal paths highlighted in red. (a)–(d) are four examples with closed contours where scores are 102.5, 90.4, 89.2 and 110.9 respectively. (c)–(d) are two examples without closed contours and the scores are 34.9 , 25.6, 27.2 and 17.9 respectively.

methods	[5]	[21]	[2]	[26]	ours
recall (in %)	27	44.5	42	44	49.5

Table 1. We randomly pick 200 occluded objects from the VOC 2007 and 2012 datasets. The recall at IoU 0.8 for different methods given 1000 proposals are reported.

methods. Note our method applies the new contour score to pick 1,000 proposals from selective search [21]. They are significantly better than the 1,000 proposals reported by selective search at almost all IoUs.

To show more about the performance, Figure 6 plots the detection rates under varying numbers (from 1 to 1000) of object proposals for overlap 0.8. Our method runs best starting at number 348.

4.2.1 Occluded Objects

Partially occluded objects are common in images. Our method also works to an extent for these objects since the non-occluded part still forms a complete and tight closed path. Empirically, we randomly pick 200 occluded objects from the VOC 2007 and 2012 datasets. Table 1 reports the recall at IoU 0.8.

IoU	0.5	0.6	0.7	0.8	0.9
[13] (in %)	89.4	83.1	72.3	58.2	36.4
Ours (in %)	90.3	85.7	75.6	62.6	38.1

Table 2. Recall on applying our contour rejection scheme to the proposals generated by [13], where the number of proposals is 1000.

4.2.2 Scoring on [13]

To demonstrate that our contour scoring is general, we apply our method to another proposal generator [13] besides selective search. For every image in VOC 2012, our contour rejection method works on the boxes generated by the method of [13]. We report the recall performance on different IoUs in Table 2. The result shows that our contour scoring achieves about 4.4% recall improvement at IoU 0.8.

4.3. Non-object Identification Evaluation

A number of methods, such as [1, 26, 5] provide objectness scores, which can also be used to label non-object proposals. In our experiments, we sort proposals in a descend-

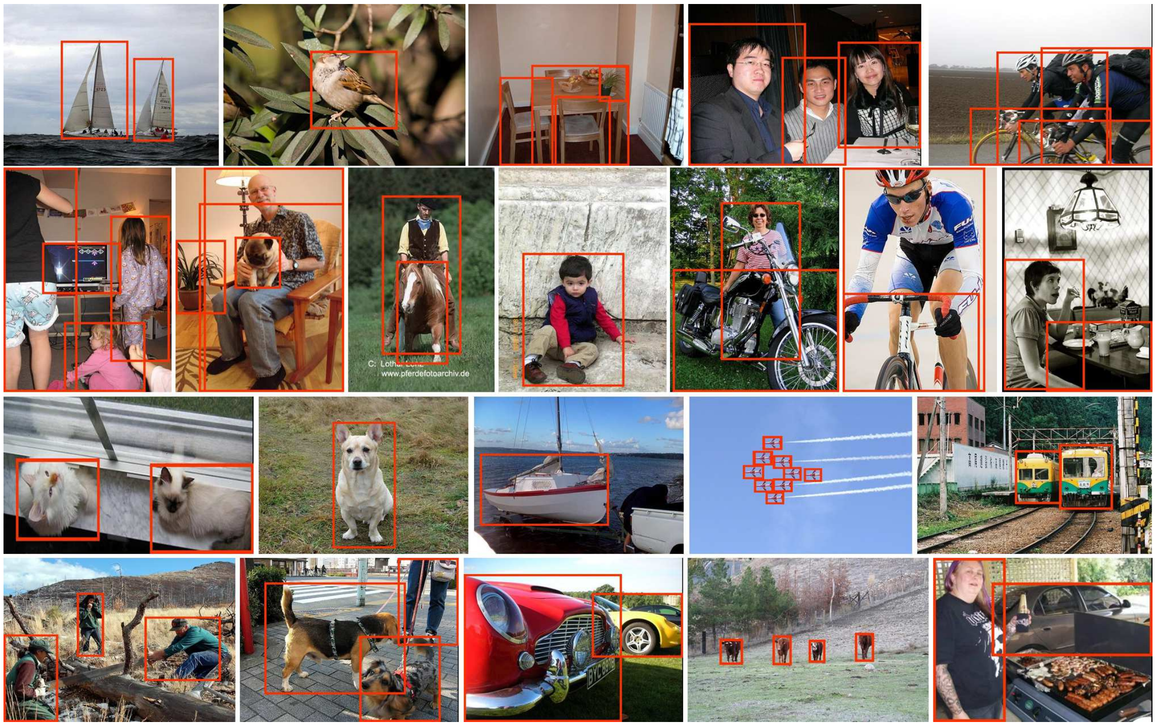


Figure 10. Examples of object proposals. Our generated proposals are shown in red.

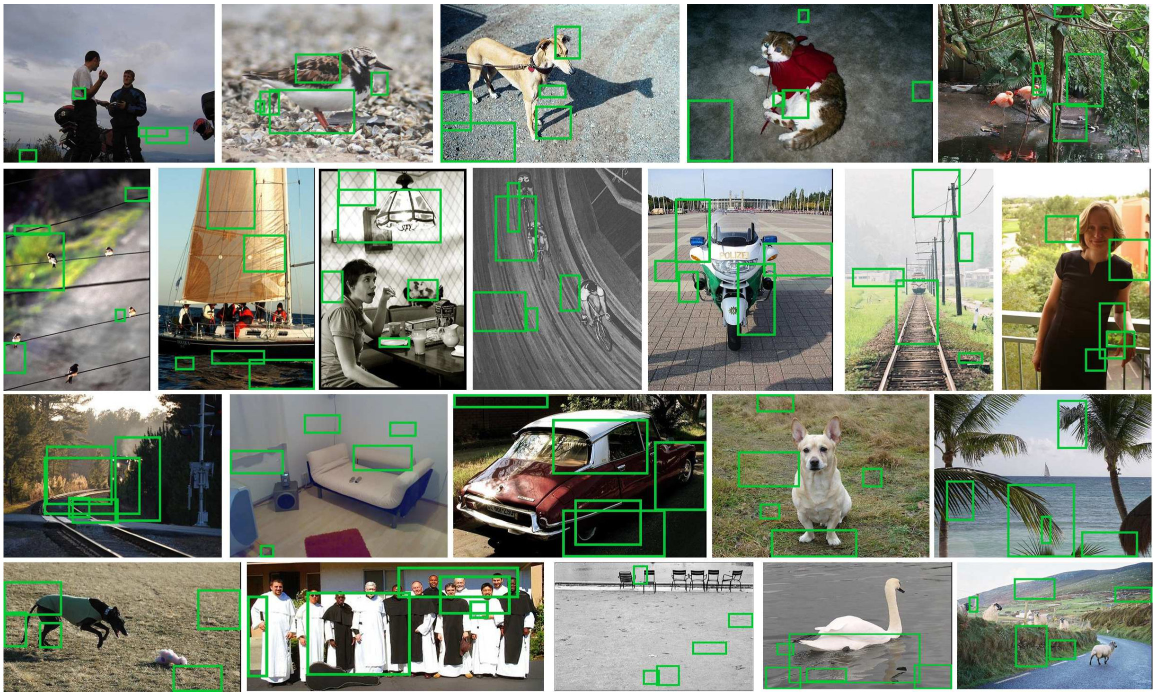
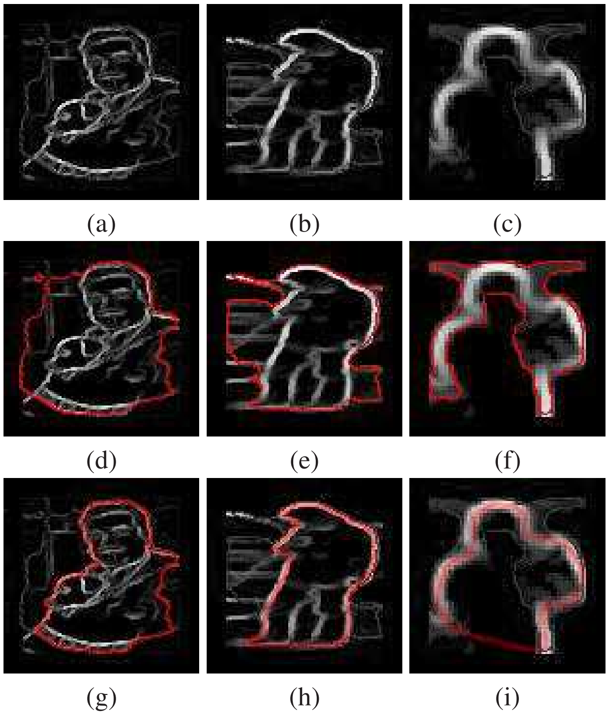


Figure 11. Examples of labeled non-object proposals are shown as green bounding boxes.

ing order regarding the objectness scores, and label the last k proposals as non-objects. To make comparison fair, we set k as the number of non-object proposals we have labeled for every image using our method. The proposals used for scoring are provided by the selective search [22], which produces about 10,000 boxes for every image.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 9. (a)–(c) input edge maps; (d)–(f) level set results; (g)–(i) our results. The level set method [14] falls short to compute the desired object without considering the completeness and tightness constraints. Note that we can reject case (c) easily due to its low completeness score.

We report the recall values of object proposals that are mistakenly labeled as non-object proposals. A small recall means that a small number of object proposals are mistakenly labeled as non-objects. Figure 7 shows results. Our method is better than others using respectively calculated object scores for non-object identification.

Figure 8 shows examples with optimized closed paths and their corresponding contour scores in the two coordinate systems. Some objects in those examples have complicated structures.

The level-set method is a standard technique for computing closed paths. It is less optimal in our case as shown in Figure 5 without exploiting the completeness and tightness constraints. A few failure cases are shown in Figure 9.

Figure 10 shows high-scored proposals. While proposals rejected by our method are shown in Figure 11. In the low-scored non-object proposals, the optimized closed curves do not have salient edge evidences and/or are not tightly bounded by the window. The running time of our method is about 2.3×10^{-2} seconds on a PC using a single thread.

5. Conclusion and Future Work

We have presented a new method to explicitly compute object contours subject to the completeness and tightness conditions. We compute these contours in the polar coordinate system where a complex search problem becomes an easily tractable dynamic programming problem. As every generated contour has a score, it can be used to select good proposals. Our new scheme is not based on the best scores but instead to sample proposals after removing the low-score proposals because they are mostly non-objects.

We have validated our non-object proposal rejection scheme. Our performance is reasonable. Our future work will be to find other reliable features for identifying objects and combine them with our contour cues. The executable will be publicly available in our project website.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2012.

[2] P. A. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marqués, and J. Malik. Multiscale combinatorial grouping. In *CVPR*, 2014.

[3] M. B. Blaschko, J. Kannala, and E. Rahtu. Non maximal suppression in cascaded ranking models. *Image Analysis*, pages 408–419, 2013.

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

[5] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In *CVPR*, 2014.

[6] M. de La Gorce and N. Paragios. Fast dichotomic multiple search algorithm for shortest cirular path. In *ICPR*, 2013.

[7] P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *ICCV*, 2013.

[8] I. Endres and D. Hoiem. Category-independent object proposals with diverse ranking. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2014.

[9] M. Everingham, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision (IJCV)*, 2010.

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

[11] J. Hosang, R. Benenson, and B. Schiele. How good are detection proposals, really? In *BMVC*, 2014.

[12] A. Humayun, F. Li, and J. M. Rehg. RIGOR: reusing inference in graph cuts for generating object regions. In *CVPR*, 2014.

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

[14] C. Li, C. Xu, C. Gui, and M. D. Fox. Distance regularized level set evolution and its application to image segmentation. *IEEE Transactions on Image Processing (TIP)*, 2010.

[15] S. Manen, M. Guillaumin, and L. J. V. Gool. Prime object proposals with randomized prim's algorithm. In *ICCV*, 2013.

[16] A. K. Mishra, Y. Aloimonos, L.-F. Cheong, and A. Kassim. Active visual segmentation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 34(2):639–653, 2012.

[17] E. Rahtu, J. Kannala, and M. Blaschko. Learning a category independent object detection cascade. In *ICCV*, 2011.

[18] E. Rahtu, J. Kannala, and M. B. Blaschko. Learning a category independent object detection cascade. In *ICCV*, 2011.

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

[20] C. Sun and B. Appleton. Multiple paths extraction in images using a constrained expanded trellis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(12):1923–1933, 2005.

[21] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision (IJCV)*, 104(2):154–171, 2013.

[22] K. E. Van de Sande, J. R. Uijlings, T. Gevers, and A. W. S-meulders. Segmentation as selective search for object recognition. In *ICCV*, 2011.

[23] X. Wang, M. Yang, S. Zhu, and Y. Lin. Regionlets for generic object detection. In *ICCV*, 2013.

[24] Y. Xiao, C. Lu, E. Tsougenis, Y. Lu, and C.-K. Tang. Complexity-adaptive distance metric for object proposals generation. In *CVPR*, 2015.

[25] V. Yanulevskaya, J. R. R. Uijlings, and N. Sebe. Learning to group objects. In *CVPR*, 2014.

[26] L. Zitnick and P. Dollar. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014.