Closing the Loop for Edge Detection and Object Proposals

Yao Lu and Linda Shapiro
University of Washington
{luyao, shapiro}@cs.washington.edu

Abstract

Edge grouping and object perception are unified procedures in perceptual organization. However the computer vision literature classifies them as independent tasks. In this paper, we argue that edge detection and object proposals should benefit one another. To achieve this, we go beyond bounding boxes and extract closed contours that represent potential objects within. A novel objectness metric is proposed to score and rank the proposal boxes by considering the sizes and edge intensities of the closed contours. To improve the edge detector given the top-down object proposals, we group local closed contours and construct global object hierarchies and segmentations. The edge detector is retrained and enhanced using these hierarchical segmentations as additional feature channels. In the experiments we show that by closing the loop for edge detection and object proposals, we observe improvements for both tasks. Unifying edges and object proposals is valid and useful.

Introduction

Edge grouping and object perception are unified procedures in human perceptual organization (Palmer 1999) but are classified as independent tasks by the current computer vision literature. Take object proposal algorithms for instance. They are designed to generate very few object candidates so as to accelerate the overall object detection pipeline. Image segmentation and edge detection results are often used as their input (Arbelaez et al. 2014; Krähenbühl and Koltun 2014; Uijlings et al. 2013; Wang et al. 2015; Wang and Zhao 2015; Xiao et al. 2015; Zitnick and Dollár 2014). To improve object proposals, many of them rely on better segmentation or edges. This dependency enables and simplifies the object perception but blocks any feedback or backward signal, from object proposals to edges, that can be useful.

In this paper, we argue that edge detection and object proposals should benefit one another. To link these two major computer vision tasks, we leverage the hierarchical structure between boundaries and regions to enhance the edge detector. To achieve this, for given bounding boxes, we extract *closed contours* that represent potential objects within, using a simple search algorithm. The closed contours are the

key to both routines discussed in this paper: the *boundary-to-proposal routine*, and the *proposal-to-boundary routine*.

To improve object proposals using boundaries, EdgeBox (Zitnick and Dollár 2014) claims that a good proposal box should contain more boundaries inside, and less boundaries on the border. This direct translation is sub-optimal due to the gap between the bounding boxes and objects of interest, as shown in Figure 1. Instead, we extract closed contours that are atomic representation of objects. A novel objectness metric is proposed that measures the sizes and intensities of the closed contours, instead of the entire bounding boxes. We show that object proposals rely on observation scales. Closed contours that contribute to good proposal boxes should have the following characteristics: (1) they should dominate the proposal boxes in area, and (2) they should have strong intensities on, and weak intensities inside, their outline. Closed contours enable us to generate region proposals without extra effort; region proposals can be extracted directly from the proposal boxes.

To improve boundaries using object proposals, our strategy is to enhance the edge detector using the object hierarchies in the scene. This top-down pipeline is in contrast to most state-of-the-art edge detectors that depend on bottom-up cues (Dollár and Zitnick 2013; Arbelaez et al. 2011; Bertasius, Shi, and Torresani 2015). However it remains an open problem to construct object hierarchies using local class-agnostic object proposals. We provide a novel solution in this paper by again leveraging the closed contours, obtained from different local proposal boxes, that already possess hierarchical structures. Based on these local contour hierarchies, global object hierarchies and segmentations for the entire image can be constructed using a grouping algorithm described later in this paper. Finally we consider global hierarchical segmentations as additional feature channels to enhance the edge detector.

Given the above context, we are able to close the loop for edge detection and object proposals. We demonstrate experiments on RGB and RGB-D images to show that, by refining edges and object proposals in an iterative manner, our unified framework provides improvements on both tasks. This indicates that unifying edges and object proposals is valid and useful; these two computer vision tasks should be treated as a whole.
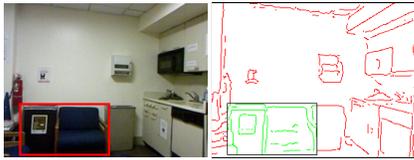
Figure 1: An example object proposal by EdgeBox (Zitnick and Dollár 2014). **Left:** Original image with one highly ranked proposal box in red. **Right:** Boundaries considered to be object (green) and background (red). Directly transferring boxes to objectness scores is suboptimal. For this example, the red bounding box has multiple objects, making it a poor candidate.

Related Work

Object proposals and edge detection. Most object proposal algorithms use boundaries or regions as their input (Arbelaez et al. 2014; Krähenbühl and Koltun 2014; Uijlings et al. 2013; Wang et al. 2015; Wang and Zhao 2015; Xiao et al. 2015; Zitnick and Dollár 2014). Applying better image segmentation or edge detection algorithms is an obvious pathway to improve object proposals. MCG (Arbelaez et al. 2014) enhances the normalized-cut image segmentation for later object proposal generation. (Wang et al. 2015) demonstrate that multiple segmentation helps object proposal generation. EdgeBox (Zitnick and Dollár 2014) utilizes a state-of-the-art Structured Edge detector (Dollár and Zitnick 2013) and provides promising objectness results. (Krähenbühl and Koltun 2014) propose to use geodesic segmentation as input to improve object proposals. Another category of work utilizes deep neural networks to generate more effective object segmentations and achieves good proposal performance, such as (Kuo, Hariharan, and Malik 2015) and (Erhan et al. 2014). On the other hand, the pathway from class-agnostic object proposals to boundaries is still unclear. In this paper, a novel method is proposed to connect object proposals back to edge detection. This enables us to close the loop for the two tasks, and we show improvements on both sides.

Objectness metrics. The objectness metric proposed in this paper is inspired by the edge intensity and object closure cues for human object perception. Edge intensity, or contour strength, is effective in various object perception tasks. (Lu et al. 2016) fit segment boundaries to true object boundaries according to edge intensities. EdgeBox (Zitnick and Dollár 2014) to our best knowledge is the only work that utilizes edge intensities for object proposals. They use the intensities implicitly to cluster the edges and assign weights to different edge groups. In this paper we explicitly use the edge intensity cue to measure the objectness of a given closed contour. Object closure, on the other hand, is widely applied in various computer vision tasks (Karpathy, Miller, and Fei-Fei 2013; Levinshtein, Sminchisescu, and Dickinson 2010; Cheng et al. 2014; Lu et al. 2011; 2012), among which BING (Cheng et al. 2014) utilizes the gradient closure to indicate objects. In this paper, we further aim at a contour level of object closure that is more challenging due to false and missing edges.

Hierarchical image segmentation. State-of-the-art image segmentation and edge detection frameworks aim to produce hierarchical representation of the objects in the scene,

such as the probabilistic boundaries (Arbelaez et al. 2011), sparse-coding-based approach (Xiaofeng and Bo 2012), and Structured Edge detector (Dollár and Zitnick 2013). These methods rely on bottom-up information of low-level edge intensities. In this paper, a different aspect of this problem is studied; the hierarchical structure of object proposals is recovered initially, while hierarchical segmentations are generated using such top-down objectness cues. This high-level hierarchical segmentation is fed to improve the edge detector afterwards.

Closed contour extraction. To represent potential objects given continuous edge intensities, previously closed contours are extracted using the ultrametric contour map (UCM) (Arbelaez et al. 2011), or the structured edge (Chen, Yang, and Yang 2015) algorithms. In this paper, we propose a new closed contour extraction algorithm based on a simple but effective search strategy. (Pinheiro, Collobert, and Dollar 2015) propose an algorithm to extract soft object masks within candidate boxes using a deep neural network. MCG (Arbelaez et al. 2014) also provides region proposals using a combination of segments. Our system is different from these; the proposed closed contour extraction algorithm is fully unsupervised, while the above approaches (Pinheiro, Collobert, and Dollar 2015; Arbelaez et al. 2014) require multiple training images.
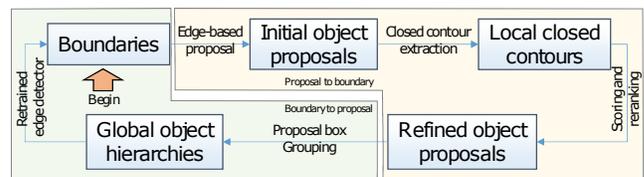

Figure 2: Closing the loop for object proposals and edge detection. Our pipeline starts with an edge detector (Begin). The boundary-to-proposal and the proposal-to-boundary routines are displayed in yellow and green regions respectively.

Framework

We aim to close the loop for object proposals and edge detection. Figure 2 summarizes the pipeline of our closing-the-loop framework. Given the initial object proposal boxes produced by an edge-based objectness approach such as Edge-Box (Zitnick and Dollár 2014), we extract the *closed contours*, which we define as groups of edges that constitute the outlines of potential objects, from each box. We then evaluate the closed contours using a novel metric to produce refined object proposals. Next we construct global object hierarchies and segmentations by grouping the local proposal boxes and closed contours. The hierarchical segmentations are treated as additional feature channels for a retrained edge detector to produce better boundaries. Finally the refined boundaries are fed into the edge-based object proposer for a new round of the process. The loops here are infinite but in our experiments we observe that they converge quickly after three or four loops. In the following sections we describe each separate module in detail.
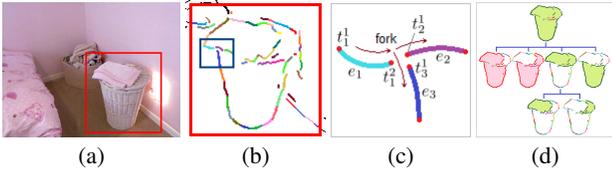
Figure 3: **(a)** Given a bounding box, we group boundary pixels into edge groups. **(b)** Different colors represent different edge groups. **(c)** Zooming in on the rectangular region in (b). Our search algorithm finds loops of edge groups represented by a sequence of terminal points in red. The algorithm forks in this example. **(d)** Resulting closed contours form a local hierarchy; larger contours contain smaller ones. Principal closed contours are shown in green, non-principal closed contours in red.

Figure 4: **Left:** Size is a critical factor for object proposals. A good proposal box should contain an object that occupies the majority of the area. Red and blue contours correspond to the principal closed contours of the red and the blue bounding boxes. Objects contribute to the bounding boxes of their scale. **Right:** po-NMS is applied on proposal boxes that share a common principal object in red. Overlapping boxes are shrunk to fit the common principal object. The green box is the result after merging and shrinking.

Closed contour extraction

Proposal boxes generated by objectness approaches are noisy and may contain no object, part of an object, or multiple objects. Neither background models (Rother, Kolmogorov, and Blake 2004) nor Hough-like techniques (Gall and Lempitsky 2013; Maji and Malik 2009) are feasible to extract single and complete objects, since they have arbitrary shapes. We leverage *closed contours* that are atomic representation of objects within the proposal boxes. We will discuss later in this section how single and complete objects can be extracted using the closed contours.

Given the edge map E and the orientation map O produced by an edge detector (e.g., the Structured Edge detector (Dollár and Zitnick 2013)), we first cluster the boundary pixels into edge groups according to their similar intensities, orientations, and connectivities, as is done in EdgeBox (Zitnick and Dollár 2014). Each edge group has two or more terminal points. The closed contour is then *a set of edge groups*, represented by a sequence of terminal points, that (1) form a closed loop, and (2) lie fully within the bounding box. The algorithm to extract candidate closed contours within a bounding box b is described below and illustrated in Figure 3.

1. For a given terminal point t_n^1 belonging to edge group e_n, find a terminal point t_n^2 that belongs to the same edge group. Add t_n^1 and t_n^2 to the current path. If more than two terminal points exist for e_n, fork.

2. For terminal point t_n^2, search within a radius r inside the bounding box b for its neighboring terminal point t_{n+1}^1 that belongs to another edge group. Add t_{n+1}^1 to the current path. If two or more terminal points are found, fork.

3. Go to step 1, recursively find the next terminal point until all the reachable edge groups in b are visited. If a loop is found during the search, add its path to the result $\mathscr{C}_b$.

We set the search radius $r > 1$, so that disconnected edges can be found during the search. To initialize the algorithm, we randomly pick a starting terminal point near the bounding box boundary. Each bounding box ultimately yields one or more closed contours if there is an object inside.

Local hierarchy of closed contours. Since we recursively search for every possible closed loop within a given bounding box, the closed contours already possess a local hierarchical structure; larger closed contours contain smaller ones, as shown in Figures 3(d). This important property is leveraged in the proposal box grouping algorithm described later
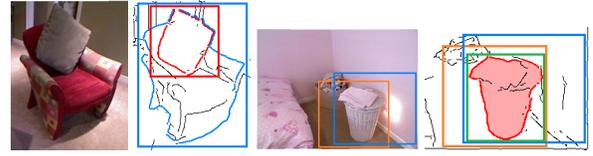
in this paper.

The boundary-to-proposal routine

What is a good proposal box? We study the good criteria for good object proposal boxes. Once we obtain the closed contours within a proposal box, the objectness criteria are two-fold:

(1) Size. Object perception highly depends on the observation scale. A good proposal box should contain one single object; only that object can be used to measure the objectness of the box. In other words, the closed contour should occupy a majority of the area in that proposal box.

For objects with hierarchical parts, size remains a critical factor. Since object parts are also objects, a good proposal box for the part should still contain only one object that is the largest in area. Consequently, object parts contribute to the proposal boxes of their scale, instead of to the bounding boxes containing the whole objects. Taking Figure 4 (left) for an example, the pillow contributes to the red bounding box instead of to the blue one. Meanwhile, for the blue bounding box, only the chair can be recognized as the object of interest due to its size.

(2) Intensity. Edge strength is effective during human object perception (Kellman and Shipley 1991; Palmer 1999), and it reflects true object boundaries on a high level. That is, inter-object edges are more likely to have strong intensities. This is the goal for most state-of-the-art edge detection frameworks (Dollár and Zitnick 2013; Arbelaez et al. 2011; Bertasius, Shi, and Torresani 2015). In our objectness metric, another criterion is the ratio of edge intensities on the object boundary versus those inside the object.

Proposal boxes containing multiple objects should get low objectness scores either due to: (1) the size for the object of interest is relatively small, since there are other objects in the box, or (2) there are edges with strong intensities within the object of interest, and those edges are more likely to be inter-object boundaries.

Scoring proposal boxes by the principal objects. Given a candidate closed contour cc within a bounding box b, the metric to measure the objectness for cc is:

$$s(cc) = \frac{area(cc)^\gamma}{area(b)} \cdot \left(\frac{avg(I_{cc})}{avg(I_{ic})}\right)^\lambda \qquad (1)$$

where ic represents the inner edges within cc, and $avg(I_{cc})$ is the average intensity on cc (intensities between disconnected terminal points are zero). λ weights between the size
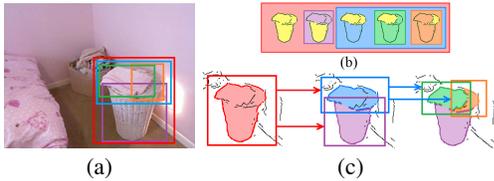
Figure 5: **(a)** A set of input proposal boxes. **(b)** The set of input principal closed contours, each of which corresponds to an input proposal box in (a). Rectangles with different colors contain the closed contours (in yellow) found in the proposal boxes of those colors. For example, the red proposal box contains all the closed contours, while the blue proposal box contains the last three closed contours. Our grouping algorithm finds global hierarchies upon local closed contour hierarchies. **(c)** Resulting global object hierarchy.

and intensity factors, and γ penalizes small loops caused by noisy edges.

The objectness for bounding box b is then recognized as the maximum score for all the candidate closed contours:

$$S(b) = \max_{cc \in \mathscr{C}_b} s(cc) \qquad (2)$$

We denote the principal object, or the principal closed contour, as the object with the maximum objectness in its corresponding proposal box.

Non-maximum suppression on the principal objects (po-NMS). Given that nearby bounding boxes often belong to the same object, non-maximum suppression (NMS) is used in current object detection systems, e.g., (Felzenszwalb et al. 2010), to remove false-positive detections. We simply extend this idea to the principal objects. Given a set of bounding boxes, false-positive detections can be eliminated if they share a common principal object. Further, the final bounding box is shrunk to fit the principal object for refinement. Figure 4 (right) illustrates an example.

Generating refined box/region proposals. After evaluating the objectness for all initial proposal boxes, we apply po-NMS and then re-rank the boxes to obtain refined object proposals in box representation. To generate region proposals, we simply take the principal closed contour as one region proposal for each refined proposal box.

The proposal-to-boundary routine

Grouping the proposal boxes. Recall that for given proposal boxes we extract closed contours that already possess local hierarchical structures. Constructing the global object hierarchies can be achieved by grouping the refined proposal boxes and closed contours at different scales. We describe the grouping algorithm in Alg. 1.

The algorithm takes as input $\mathscr{C}_P$, the set of principal closed contours, each of which corresponds to a proposal box. Then a non-overlapping subset of closed contours are found, which represent different objects or parts. Larger closed contours in the subset are linked to the smaller closed contours (children) they contain, according to the local hierarchies already extracted. This process is done recursively until reaching the leaf closed contours that contain no inner loop. $c.children$ denotes the children closed contours of c that are principal closed contours corresponding to other smaller proposal boxes. Non-principal closed contours are

---

Algorithm 1: Grouping proposal boxes

Name: $clusterCCs(\mathscr{C}_P)$
Steps:
> $h \leftarrow \varnothing, C \leftarrow \varnothing$; sort $\mathscr{C}_P$ by area in descending order.
> **for** pcc in $\mathscr{C}_P$ **do**
> > **if** $pcc \notin C$ and $\forall c \in C, pcc \notin c.children$,
> > > **then** $C = C \cup pcc$.
> **for** $c \in C$ **do**
> > **if** $|c.children| > 1$,
> > > **then** $h \leftarrow h \cup clusterCCs(\mathscr{C}_P - C)$.
> > **else** $h = h \cup \{c\}$.
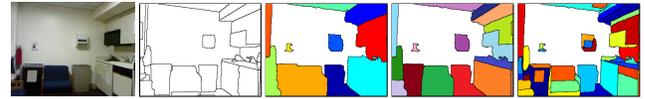> **return** $\{h\}$ in a tree structure.

Figure 6: An example of the global hierarchical segmentations. Left to right: original image, boundary groundtruth, seg_4, seg_3, seg_2.

not considered here due to their weak objectness. The algorithm ultimately yields the global hierarchical structure of the principal closed contours and proposal boxes. Note that the final result may consist of multiple trees, generated from non-overlapping object proposal boxes. Figure 5 shows an example of the grouping algorithm.

Constructing hierarchical segmentations. It is straightforward to construct hierarchical segmentations for the objects in the scene. Traversing from the leaf nodes to the roots in the object hierarchies, the closed contours are drawn and used to generate the segmentation for each layer in the hierarchy. Straight lines are placed between disconnected terminal points that belong to different edge groups, so as to make contours fully closed. Each closed contour is filled and labeled with its depth in the hierarchy. We denote $seg_1, ..., seg_t$ as the t hierarchical segmentations obtained. seg_1 is the finest segmentation and seg_t is the coarsest. The finest four segmentations $seg_1, .., seg_4$ are used in practice, since in our experiments we find they are more informative. Figure 6 shows examples of the hierarchical segmentations.

Training the enhanced edge detector. (Dollár and Zitnick 2013) take the intensities from different color channels as input to train the Structured Edge detector. (Gupta et al. 2014) extend their framework and propose a RGB-D boundary detector by taking the depth values as an additional feature channel to the Structured Edge detector. Inspired by this simple but effective approach, we similarly treat the hierarchical segmentations as additional feature channels to the input. Before training the enhanced edge detector, we run one pass of the object proposal and closed contour grouping algorithm on the training set to generate the hierarchical segmentations. We leverage $seg_1, .., seg_4$ as four additional feature channels for training. The detailed training procedures are described in (Dollár and Zitnick 2013). Finally at test time, we utilize the retrained edge detector to produce new boundaries for each proposal-to-boundary routine in our closing-the-loop framework.

Implementation details

We note several details in our implementation. To extract closed contours, the search space is large. Hence to reduce the computational complexity is critical. For terminal points with direct connection to other terminal points, the search radius r can be set low; we use $r = 3px$. For terminal points without direct connection, r is set to be 15% of the bounding box size to ensure all the loops can be found. We perform a pre-pass on the entire image, aiming to eliminate small loops and cache the paths for valid loops. Further, due to the hierarchical structure of closed contours, we store results for small candidate boxes and reuse them for larger ones.

To close the loop for edges and object proposals, we set a threshold on the objectness score before grouping the closed contours. This is to reduce noise from pool-ranked candidate boxes. Meanwhile we set $\lambda = 0.8$ to put more weight on the size factor.

Experiments

We connect and bootstrap the edge detection and object proposal routines described in the above sections. We show in our experiments how they can improve one another, i.e., how they perform on state-of-the-art datasets and benchmarks after several algorithmic iterations.

Quantitative Evaluations

To follow the object proposal evaluation routine, we leverage the Pascal VOC 2012 (Everingham et al. 2010) dataset with RGB images, and the NYU-D (Silberman et al. 2012) dataset with RGB-D images. For the later dataset we apply the training-testing split from (Gupta et al. 2014), which contains 795 training images. The detection rate (recall) versus the number of candidate boxes, and the area under curve (AUC) are reported. The intersection over union (IOU) metric over bounding boxes is used in comparing results with ground truth, as is applied in most current object proposal methods (Zitnick and Dollár 2014; Cheng et al. 2014; Uijlings et al. 2013; Wang et al. 2015). A good object proposal algorithm should propose as few candidate boxes as possible while maintaining a high recall rate.

The quantitative evaluations of our boundary-to-proposal routine is demonstrated in Figure 7 and Table 2 with IOU=0.7 and number of candidate boxes=1000. We compare with EB (Zitnick and Dollár 2014), MCG (Arbeláez et al. 2012), SS (Uijlings et al. 2013), and BING (Cheng et al. 2014). We observe a clear improvement when boundaries and object proposals are connected. On the NYU-D dataset, our framework improves 11.6% AUC for object proposals in iteration 1 and 2.7% AUC in iteration 2. Meanwhile on the Pascal VOC 2012 dataset, we obtain about 3% improvement in iteration 1 and 1% in iteration 2. Note that in iteration 0 we leverage the edges from (Dollár and Zitnick 2013) and the object proposals from (Zitnick and Dollár 2014) as initial inputs. After only three iterations, our method achieves comparable performance to the state-of-the-art MCG and does even better with fewer candidate boxes. On the NYU-D dataset, MCG outperforms our approach with high number of candidate boxes as they train to rank the proposed regions,
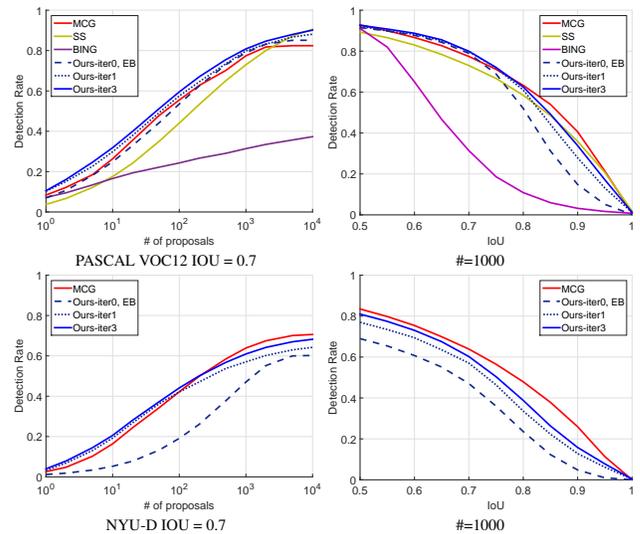

Figure 7: Quantitative evaluation of object proposals on the Pascal VOC2012 validation (RGB) and the NYU-D test (RGB-D) datasets. Our results are shown in dotted to solid lines in blue, representing iteration 0,1,3 respectively.

Method	BSD500			NYUD		
	ODS	OIS	AP	ODS	OIS	AP
gPb	0.71	0.74	0.65	0.63	0.66	0.56
SC	0.74	0.76	0.77	0.64	0.67	0.57
SE+	-	-	-	0.70	0.72	0.69
N^4-fields	0.75	0.77	0.78	-	-	-
DeepEdge	**0.75**	**0.77**	**0.81**	-	-	-
Ours-iter0, SE	0.74	0.76	0.78	0.69	0.69	0.68
Ours-iter1	**0.75**	0.76	0.79	0.71	0.72	0.72
Ours-iter2	**0.75**	**0.77**	0.80	0.71	0.74	0.73
Ours-iter3	**0.75**	**0.77**	0.80	**0.72**	**0.74**	**0.74**

Table 1: Quantitative evaluations for edges on the BSD500 (RGB) and NYU-D (RGB-D) datasets at iteration 0 to 3.

while our objectness metric is unsupervised. Our framework converges quickly after iteration 4 on both datasets.

To evaluate our proposal-to-boundary routine, we leverage the BSD500 (Arbelaez et al. 2011) and the NYU-D datasets. We apply the evaluation metrics in (Arbelaez et al. 2011; Dollár and Zitnick 2013), measuring fixed contour threshold (ODS), per-image best threshold (OIS) and average precision (AP). Table 1 reports the evaluation results. We compare with gPb (Arbelaez et al. 2011), SC (Xiaofeng and Bo 2012), SE (Dollár and Zitnick 2013), SE+ (Gupta, Arbelaez, and Malik 2013), N^4-fields (Ganin and Lempitsky 2014), and DeepEdge (Bertasius, Shi, and Torresani 2015). Again, we observe improvements to the edges when object proposals are connected. Note that the BSD500 dataset is not designed for object detection; each image has only one major object. Connecting edge detection and object proposals yields less improvement than on the NYU-D dataset.

Qualitative Evaluations

Figure 8 demonstrates the boundaries and hierarchical segmentations produced by our algorithm. We further show in Figure 9 the intermediate boundaries at iterations 0,1,2 on the NYU-D dataset. The resulting boundaries gradually fit
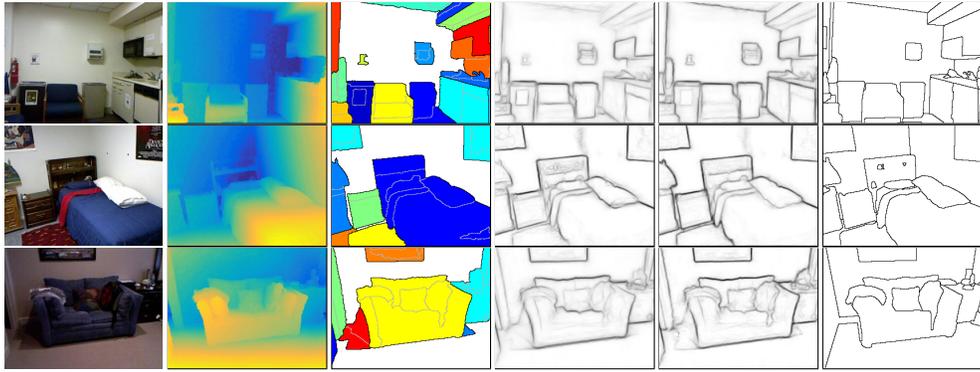
Figure 8: Closing the loop on the NYU-D dataset. Left to right: original image, depth map, hierarchical segmentation assembled using Seg_4 (color) and Seg_2 (boundaries), initial edges, edges after iteration 4, and ground truth. Edge detections are before NMS. Our refined edge detector produces edges that better fit the object hierarchies: intensities are stronger for inter-object edges and weaker for intra-object edges.
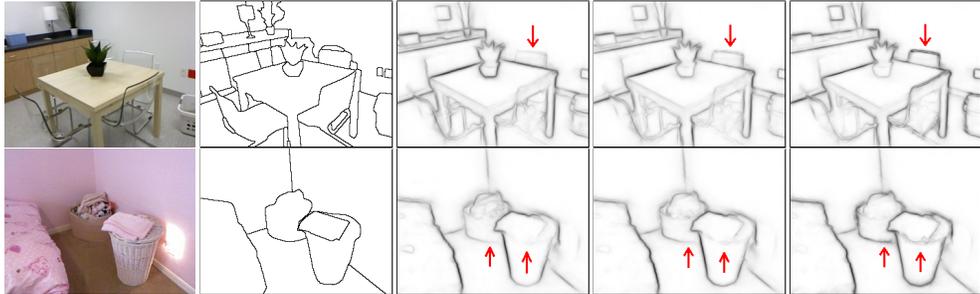

Figure 9: Intermediate boundaries when the loop runs. Left to right: original image, ground truth, boundaries for iteration 0, 1, and 2. We show that the boundaries gradually fit the object hierarchies, e.g., the edges pointed to by the red arrows.

Figure 10: Demonstration of our box and region proposals on the Pascal VOC 2012 validation set. Region proposals are recognized as the principal closed contours for each candidate box. The top ranked proposals are shown.

Method	Iteration	0	1	2	3	4
SE&	Proposal AUC	0.40	0.52	0.55	0.56	0.56
EdgeBox	Edge AP	0.68	0.74	0.75	0.76	0.76
SC&	Proposal AUC	0.39	0.46	0.47	0.47	0.47
EdgeBox	Edge AP	0.63	0.67	0.69	0.69	0.69
SE&	Proposal AUC	0.42	0.54	0.57	0.57	0.58
MCG	Edge AP	0.68	0.75	0.76	0.77	0.77

Table 2: Closing the loop on the NYU-D (RGB-D) dataset with different edge detection and object proposal combinations. For object proposals, IOU = 0.7. Proposal AUC is measured with # of boxes=1000.

the object hierarchies in the scene; inter-object edges become stronger and inner-object edges become weaker, e.g., the sofa and the basket in Figure 8 and 9.

Figure 10 shows examples of the region proposals generated by our closed contour algorithm. As described previously, extracting object shapes within proposals boxes can be achieved without additional cost. In summary, all the above quantitative and qualitative results indicate that object proposals provide useful information to help edge detection, in strengthening important edges as well as in removing noisy edges. Better boundaries, on the other hand, help object proposals, as is already achieved by many object proposal algorithms. These two tasks are able to help one another in an iterative manner.

Finally, we are interested in how our closing-the-loop framework can be applied on generic edge detection and ob-

ject proposal algorithms. As shown in Table 2, we replace SE with SC (Xiaofeng and Bo 2012), while EdgeBox is replaced with MCG (Arbelaez et al. 2014). SC is trained similarly as SE with additional feature channels from the hierarchical segmentations. All experimental settings yield satisfying improvements, indicating that our closing-the-loop framework is generic and valid to many current edge detection and object proposal algorithms. These two major computer vision tasks should be treated as a whole.

Conclusions

We present a unified framework for bootstrapping object proposals and edge detection. This closing-the-loop strategy combines top-down objectness cues with bottom-up boundary cues, leading to meaningful improvement that facilitates future research in other computer vision tasks related to object perception, such as semantic segmentation and object detection.

References

Arbelaez, P.; Maire, M.; Fowlkes, C.; and Malik, J. 2011. Contour detection and hierarchical image segmentation. *TPAMI* 33(5):898–916.

Arbeláez, P.; Hariharan, B.; Gu, C.; Gupta, S.; Bourdev, L.; and Malik, J. 2012. Semantic segmentation using regions and parts. In *CVPR*.

Arbelaez, P.; Pont-Tuset, J.; Barron, J.; Marques, F.; and Malik, J. 2014. Multiscale combinatorial grouping. In *CVPR*.

Bertasius, G.; Shi, J.; and Torresani, L. 2015. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In *CVPR*.

Chen, Y.-T.; Yang, J.; and Yang, M.-H. 2015. Extracting image regions by structured edge prediction. In *WACV*.

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

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

Erhan, D.; Szegedy, C.; Toshev, A.; and Anguelov, D. 2014. Scalable object detection using deep neural networks. In *CVPR*.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *IJCV* 88(2):303–338.

Felzenszwalb, P. F.; Girshick, R. B.; McAllester, D.; and Ramanan, D. 2010. Object detection with discriminatively trained part-based models. *TPAMI* 32(9):1627–1645.

Gall, J., and Lempitsky, V. 2013. Class-specific hough forests for object detection. In *Decision forests for computer vision and medical image analysis*. Springer. 143–157.

Ganin, Y., and Lempitsky, V. 2014. N^ 4-fields: Neural network nearest neighbor fields for image transforms. In *ACCV*. 536–551.

Gupta, S.; Arbelaez, P.; and Malik, J. 2013. Perceptual organization and recognition of indoor scenes from rgb-d images. In *CVPR*.

Gupta, S.; Girshick, R.; Arbeláez, P.; and Malik, J. 2014. Learning rich features from rgb-d images for object detection and segmentation. In *ECCV*.

Karpathy, A.; Miller, S.; and Fei-Fei, L. 2013. Object discovery in 3d scenes via shape analysis. In *ICRA*.

Kellman, P. J., and Shipley, T. F. 1991. A theory of visual interpolation in object perception. *Cognitive psychology* 23(2):141–221.

Krähenbühl, P., and Koltun, V. 2014. Geodesic object proposals. In *ECCV*.

Kuo, W.; Hariharan, B.; and Malik, J. 2015. Deepbox: Learning objectness with convolutional networks. *arXiv preprint arXiv:1505.02146*.

Levinshtein, A.; Sminchisescu, C.; and Dickinson, S. 2010. Optimal contour closure by superpixel grouping. In *ECCV*.

Lu, Y.; Zhang, W.; Lu, H.; and Xue, X. 2011. Salient object detection using concavity context. In *ICCV*.

Lu, Y.; Zhang, W.; Jin, C.; and Xue, X. 2012. Learning attention map from images. In *CVPR*.

Lu, Y.; Bai, X.; Shapiro, L.; and Wang, J. 2016. Coherent parametric contours for interactive video object segmentation. In *CVPR*.

Maji, S., and Malik, J. 2009. Object detection using a max-margin hough transform. In *CVPR*.

Palmer, S. E. 1999. *Vision science: Photons to phenomenology*, volume 1. MIT press Cambridge, MA.

Pinheiro, P. O.; Collobert, R.; and Dollar, P. 2015. Learning to segment object candidates. *arXiv preprint arXiv:1506.06204*.

Rother, C.; Kolmogorov, V.; and Blake, A. 2004. Grabcut: Interactive foreground extraction using iterated graph cuts. *TOG* 23(3):309–314.

Silberman, N.; Hoiem, D.; Kohli, P.; and Fergus, R. 2012. Indoor segmentation and support inference from rgbd images. In *ECCV*.

Uijlings, J. R.; van de Sande, K. E.; Gevers, T.; and Smeulders, A. W. 2013. Selective search for object recognition. volume 104, 154–171.

Wang, X. C. H. M. X., and Zhao, Z. 2015. Improving object proposals with multi-thresholding straddling expansion. *CVPR*.

Wang, C.; Zhao, L.; Liang, S.; Zhang, L.; Jia, J.; and Wei, Y. 2015. Object proposal by multi-branch hierarchical segmentation. *CVPR*.

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

Xiaofeng, R., and Bo, L. 2012. Discriminatively trained sparse code gradients for contour detection. In *NIPS*.

Zitnick, C. L., and Dollár, P. 2014. Edge boxes: Locating object proposals from edges. In *ECCV*.