Extracting Image Regions by Structured Edge Prediction

Yi-Ting Chen Jimei Yang Ming-Hsuan Yang
University of California at Merced
{ychen86, jyang44, mhyang}@ucmerced.edu

Abstract

*We present two approaches to extract regions from structured edge detection. While the state-of-the-art algorithm based on globalized probability of boundary (gPb) generates a hierarchical region tree, it entails significant computational load. In this work, we exploit an efficient algorithm for structured edge prediction to extract regions. To generate high quality regions, we develop a novel algorithm to link the structured edge and gPb hierarchical image segmentation framework with steerable filters. The extracted regions are grouped by the proposed hierarchical grouping method to generate object proposals for effective detection and recognition problems. We demonstrate the effectiveness of our region generation for image segmentation on the BSDS*500 *database, and region generation for object proposals on the PASCAL VOC* 2007 *benchmark database. Experimental results show that the proposed algorithm achieves the comparable or superior quality to the state-of-the-art methods.*

1. Introduction

Image regions play an important role between high level semantics and low level pixel-wise information. Effective region information is useful for image segmentation and object detection. However, it is challenging to extract high quality regions due to occlusion, heterogeneous texture, color within one object and obscured boundary with respect to other image parts.

Recent years have witnessed significant progress in image segmentation with numerous applications [24, 11, 14, 4, 1, 6]. A class of image segmentation algorithms aim to extract image segments or regions from edge or contour signals. Arbelaez *et al*. [4] propose a hierarchical image segmentation algorithm from the state-of-the-art gPb contour detector. However, the gPb contour detector has a heavy computational load to generate contours. In this work, we propose algorithms to efficiently extract high quality regions for image segmentation.

In contrast to existing image segmentation algorithms that focus on partitioning an image into segments precisely, object proposal algorithms [12, 26, 2, 19, 7] have been developed in recent years to generate a smaller set of windows for detecting object instances in the image efficiently. The objectness measure [7, 2] predicts whether a proposed window is covering a certain object. On the other hand, mid-level features obtained from initial segmentation algorithms are merged to generate object proposals in [19, 12, 26]. While the graph-based algorithm [14] generates image segments efficiently, the extracted regions are of irregular size and shape [1], thereby making them difficult to merge for rendering object proposals. In contrast, we tackle this problem with an algorithm that generates high quality regions.

In this paper, we propose two novel region generation algorithms to address the issues of image segmentation and object detection. Based on the high quality results generated by the efficient structured edge (SE) algorithm [10], we integrate it with steerable filters (SF) [15], oriented watershed transform (OWT) [4] and ultrametric contour map (UCM) [3] to generate image segments (refer it as SE-SF-OWT-UCM). For object detection, we incorporate the SE-SF-OWT-UCM algorithm with a hierarchical grouping method [26] to generate object proposals. We demonstrate the effectiveness of the proposed region generation algorithm for image segmentation on the BSDS500 database [4], and the region generation algorithm for object detection on the PASCAL VOC 2007 [13] benchmark database. The proposed SE-SF-OWT-UCM algorithm achieves comparable performance to the state-of-the-art gPb framework [4] among the region benchmarks on the BSDS500 database, and outperforms the other state-of-the-art methods. Furthermore, the proposed SE-SF-OWT-UCM algorithm followed by a hierarchical grouping process generates higher quality object proposals than the state-of-the-art selective search method [26].

2. Related Work and Problem Context

In this section, we discuss the most relevant work on edge detection and region generation.

Edge Detection. Edge detection is a fundamental problem that can be applied to numerous vision tasks. It captures important information such as the object shape, surface boundary and local structure in an image. In addition, it serves as an effective representation of an image in a lower dimensionality manner while preserving structural properties.

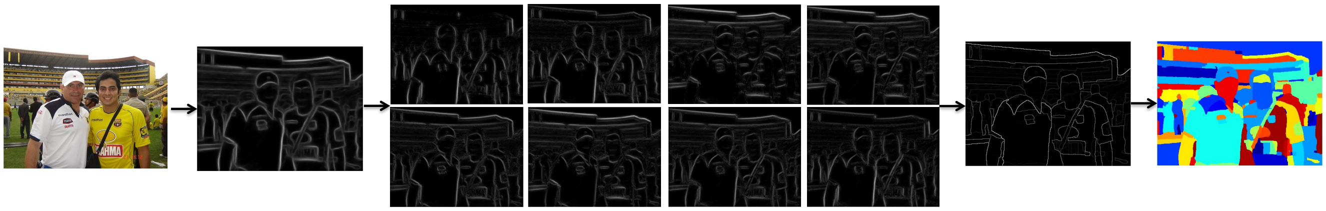
Methods that exploit color, texture and other sophisticated descriptors have been developed for edge detection [22, 15, 4, 20]. In addition, numerous learning methods [20, 9, 23, 18] have been developed for edge detection. Martin *et al.* [20] combine local image features including brightness, color and texture in an optimal way, and features are used to train a logistic regression classifier for predicting the posterior probability of a boundary at every location of a given image. Dollár *et al.* [9] propose a Boosted Edge Learning (BEL) method where features are combined in a probabilistic boosting-tree-based classifier [25]. Ren and Bo [23] tackle the edge detection task by computing sparse code gradients (SCG) which learns rich representations from image patches. Most recently, Lim *et al.* [18] learn effective structural features from mid-level information obtained from hand-drawn sketch tokens using the BSDS500 database [4] for edge detection.

Structured learning [21] has drawn much attention and been applied to semantic labeling [17] and edge detection [10] in recent years. Since edges in a local patch exhibit similar characteristics, a structured learning approach is considered. Dollár and Zitnick [10] propose a structured learning approach based on random decision forests for edge detection. That is, the edge detection task is formulated as a local-edge-mask prediction problem from image patches. A more detailed review of [10] is discussed in Section 3. In this paper, we exploit structured edges to generate regions for image segmentation and object detection.

Region Generation. Region generation is one of the core research topics in computer vision. Much progress on region generation has been made in the past decade [24, 11, 14, 4, 1, 6, 12, 26, 2, 19, 7]. We categorize the board family of region generation algorithms into four types. First, superpixel algorithms [1] aim to efficiently generate regular regions that adhere to boundaries [1]. Achanta *et al.* [1] propose the simple linear iterative clustering (SLIC) method which performs more efficiently than conventional k-means methods by limiting the search space in the assignment step. Second, object segmentation methods [6, 28] are developed to generate either generic or category specific object hypotheses. The constrained parametric min-cut (CPMC) method [6] tackles object segmentation by solving a sequence of graph cut problems to generate hypotheses. Subsequently, a ranking procedure is applied to generate object hypotheses. On the other hand, Yang *et al.* [28] propose a category specific object hypotheses algorithm by combining a parametric segmentation model and a nonparametric matching mechanism.

Third, bottom-up image segmentation algorithms [24, 14, 4] play an important role in numerous vision tasks. Shi and Malik [24] propose a graph-based normalized cut algorithm. Instead of focusing on local patches, the normalized-cut method utilizes global information and partitions the graph based on the dissimilarity measure between the different groups and the similarity measure within the groups [24]. Another graph-based segmentation proposed by Felzenszwalb and Huttenlocher [14] aims to efficiently cluster pixels into groups where the resulting group is the minimum spanning tree. Arbelaez *et al.* [4] develop a unified approach to contour detection and image segmentation. They propose the oriented watershed transformation (OWT) to form an initial segmentation from the gPb contour detector [4]. The boundaries of the initial segmentation are used to form the ultrametric contour map (UCM) [3], which is a representation of a hierarchical region tree. The gPb hierarchical segmentation framework provides high quality hierarchical regions. However, the gPb contour detector requires heavy computational loads. In this paper, we present an efficient and high quality hierarchical region generation algorithm based on structured edge detection [10].

Fourth, we regard object proposal methods as a type of region generation algorithms. A object proposal algorithm aims to find a small number of windows where objects are likely to appear. Instead of using sliding windows for object detection, object proposal algorithms [12, 26, 2, 19, 7] have been demonstrated an effective approach for reducing the search space. Alexe *et al.* [2] integrate a set of image cues in the Bayesian framework and compute an objectness score to measure the likelihood of a window containing an object. Another type of objectness work proposed by Cheng *et al.* [7] is based on the observation that each object has a well-defined closed contour from which the likelihood measure is computed using a binarized normed gradients. Endres *et al.* [12] develop a structured learning method to rank regions that are coarsely extracted based on a number of cues. Two object proposal algorithms [26, 19] apply a graph-based segmentation [14] to generate initial segments, which area followed by different grouping criteria. Manen *et al.* [19] propose a randomized Prim's algorithm for grouping the initial segments and demonstrate favorable performance under a range of evaluation scenarios. On the other hand, Uijlings [26] use a greedy data-driven grouping, which aims to diversify the search space to obtain more information. Most recently, the object proposal algorithm proposed by [26] is combined with convolutional neural networks (CNNs) for the generic object detection task [16]. As evident in recent work [26, 19, 16], the initial segmentation results [14] play a key factor in learning effective object proposals. Although the graph-based segmentation method [14] generates regions efficiently, the re-



(a) (b) (c) (d) (e)

Figure 1. **Outline of the region generation architecture.** The structured edge detector [10] is applied to an input image in Fig. 1(a) to obtain the structured edge map shown in (b). Then, the structured edge map is transformed to 8 oriented structured edge maps shown in (c) by steerable filters. Subsequently, the OWT and UCM steps are conducted to obtain the ultrametric contour map (UCM), a weighted contour map, shown in (d). Finally, Fig. 1(e) is the segmentation result obtained by thresholding the UCM at different scales.

sulting segments are usually fragmented and less coherent to object boundaries. In this work, we first generate high quality regions efficiently and and then group them hierarchically to generate object proposals effectively.

3. Structured Edge Detection

In this section, we review the structured edge detection [10] which is used for extracting regions in the proposed algorithm. Dollár *et al.* formulate the edge detection task in the structured learning framework where a random decision forest is exploited. A decision tree $f_t(x)$ classifies an input $x \in \mathcal{X}$ by splitting the data between the left or right sub-tree according to a binary split function $h(x, \theta_j)$ with parameter θ_j at each node j. Given a node j and a training set $\mathcal{S} \subset \mathcal{X} \times \mathcal{Y}$, the training goal of decision tree is to find parameters θ_j that maximize the information gain criterion I_j defined by

$$I_j = I(\mathcal{S}_j, \mathcal{S}_j^L, \mathcal{S}_j^R)\,, \tag{1}$$

where $\mathcal{S}_j^L = (x, y) \in \mathcal{S}_j | h(x, \theta_j) = 0, \mathcal{S}_j^R = \mathcal{S}_j \setminus \mathcal{S}_j^L$. When a binary split function $h(x, \theta_j) = 0$, the input x is classified to the left, and otherwise right sub-tree. This process terminates at a leaf (terminal) node (beige path) [8]. The output $y \in \mathcal{Y}$ is the prediction for the input x and stored at the leaf node. A decision forest is an ensemble of independent decision trees f_t. An ensemble model is used to combine the prediction of each decision $f_t(x)$ given the input x. The choice of the ensemble model depends on the output label space $\mathcal{Y}$ [10]. However, the major drawback of the decision tree classification is data overfitting. Thus, using random decision forests are considered to ameliorate the overfitting.

Dollár *et al.* extend random forests to generate structured outputs $\mathcal{Y}$ [10]. Given an image patch $x \in \mathcal{X}$, the output $y \in \mathcal{Y}$ stores the corresponding segmentation mask or binary edge map. A segmentation mask is denoted by $y \in \mathcal{Y} = \mathbb{Z}^{d \times d}$ and a binary edge map is represented by $y' \in \mathcal{Y}' = \{0, 1\}^{d \times d}$, where d is the patch width. Training a random forest classifier with structured outputs is challenging because of the high dimensionality of the output space and the function used to define the information gain criterion.

The main goal of the structured random forests is to map all structured labels to a discrete set $c \in \mathcal{C}$. Dollár *et al.* solve this problem by first mapping the structured output space $\mathcal{Y}$ to an intermediate space $\mathcal{Z}$. The problem with high dimensionality of the structured output spaces $\mathcal{Y}$ is alleviated by sampling m dimensions of $\mathcal{Z}$ followed by the principal component analysis (PCA). Most importantly, an information gain criterion, a similarity measurement over $\mathcal{Y}$, is obtained by computing the approximated distance in $\mathcal{Z}$. Finally, the proposed intermediate space $\mathcal{Z}$ is mapped to the discrete labels space $\mathcal{C}$.

4. Region Generation

In this section, we describe how we extract regions from structured edge for image segmentation and object detection.

4.1. Region Generation for Image Segmentation

The overall region generation architecture is shown in Fig. 1. The structured edge detector is applied to an input image in Fig. 1(a) to obtain the structured edge map shown in (b). Then, the structured edge map is transformed to 8 oriented structured edge maps shown in (c) by steerable filters. Subsequently, the OWT and UCM steps are conducted to obtain the ultrametric contour map (UCM), a weighted contour map, shown in (d). Finally, Fig. 1(e) is the segmentation result obtained by thresholding the UCM at different scales.

Edges can be obtained from segmentation masks easily when they are available and adherent to object boundaries. However, it is not straightforward to retrieve segmentation masks from edges. Arbelaez *et al.* [4] develop a unified approach to generate a hierarchical region tree from the gPb contour detector. Given an input image I, the gPb contour detector predicts the globalized posterior probability of boundary at location (x, y) with respect to θ in 8 different

Table 1. **Region benchmarks on the BSDS500 database [4].** The proposed algorithm generates comparable region quality as gPb-OWT-UCM, but it is much faster according to Table 2. On the other hand, it achieves superior performance than other state-of-the-art algorithms.

	Covering			PRI		VI	
	ODS	OIS	AP	ODS	OIS	ODS	OIS
Human	0.72	0.72	-	0.88	0.88	1.17	1.17
gPb-OWT-UCM [4]	**0.59**	**0.65**	**0.74**	**0.83**	**0.86**	**1.69**	**1.48**
Mean-Shift [11]	0.54	0.58	0.66	0.79	0.81	1.85	1.64
FH [14]	0.52	0.57	0.69	0.80	0.82	2.21	1.87
Normalized Cut [24]	0.45	0.53	0.67	0.78	0.80	2.23	1.89
Ours	0.58	0.64	**0.74**	0.82	**0.86**	1.71	1.52

orientations, where $\theta \in [0, \pi)$. Due to its high computational load, we adopt structured edge [10] in this work. In the following, we denote structured edge (SE) as $e(x, y)$.

We integrate the structured edge $e(x, y)$ with the oriented watershed transformation (OWT) in a way similar to the gPb algorithm in which we present an effective way to generate oriented structured edge $e(x, y, \theta)$ from the $e(x, y)$ using a steerable filter [15]. First, we treat the structured edge $e(x, y)$ as a gradient magnitude map. In addition to structured edge $e(x, y)$, a coarse orientation map $\Theta(x, y)$ is also computed [10]. Based on the orientation map $\Theta(x, y)$, we project the structured edge $e(x, y)$ to the x and y directions. The gradient in the x direction is defined as $e_x(x, y) = e(x, y)\cos(\Theta(x, y))$ and the gradient in the y direction is defined as $e_y(x, y) = e(x, y)\sin(\Theta(x, y))$. We then obtain the oriented structured edge $e(x, y, \theta)$ as follows:

$$e(x, y, \theta) = |e_x(x, y)\cos(\theta) + e_y(x, y)\sin(\theta)|\,, \qquad (2)$$

where orientations θ in this paper are $[\pi/8\,, \pi/4\ 3\pi/8\,, \pi/2\,, 5\pi/8\,, 3\pi/4\,, 7\pi/8\,, \pi]$, and they are selected for performance improvement.

Subsequently, the OWT step is conducted because the output of the structured edge is not closed contours. We use an approach similar to that in the gPb method [4]. Finally, the hierarchal regions are obtained by a greedy graph-based merging algorithm and represented by the UCM scheme [3] as a weighted contour map. To obtain a different number of segments from the hierarchical region tree, we threshold the contour map at different scales $k \in [0, 1]$, where $k = 0$ corresponds to the finest partition and $k = 1$ corresponds to the original image. We refer the proposed region generation algorithm SE-SF-OWT-UCM for short.

4.2. Region Generation for Object Detection

The graph-based segmentation algorithm [14] has been used in several state-of-the-art object proposal algorithms [26, 19] and other vision tasks [16, 27]. While it generates initial segments efficiently, the generated regions do not usually adhere to object boundaries as shown in the second column of Figure 2, Motivated by this, we use the proposed SE-SF-OWT-UCM algorithm to generate high quality regions effectively.

Table 2. **Segmentation speed comparison.** We report the average segmentation speed for two different algorithms on the BSDS500 testing set, which contains 200 images. The proposed algorithm generates comparable region quality as gPb-OWT-UCM [4] with higher efficiency.

Method	Computational time (s)
gPb-OWT-UCM [4]	53.6
Ours	4.6

Recently, Uijlings [26] propose the selective search method for object proposals based on the graph-based segmentation [14] and hierarchical grouping. Instead, we use the proposed SE-SF-OWT-UCM algorithm to obtain initial regions and group them hierarchically using a number of similarity measures. In [26], four different similarity measures are computed based on color, texture, size, and fill (which measures how well two regions fit into each other). In this work, we use the color and size similarity measures as they can be computed efficiently. For further performance gain, we use a diversification scheme by sampling information from the HSV and Lab color spaces in a way similar to [26].

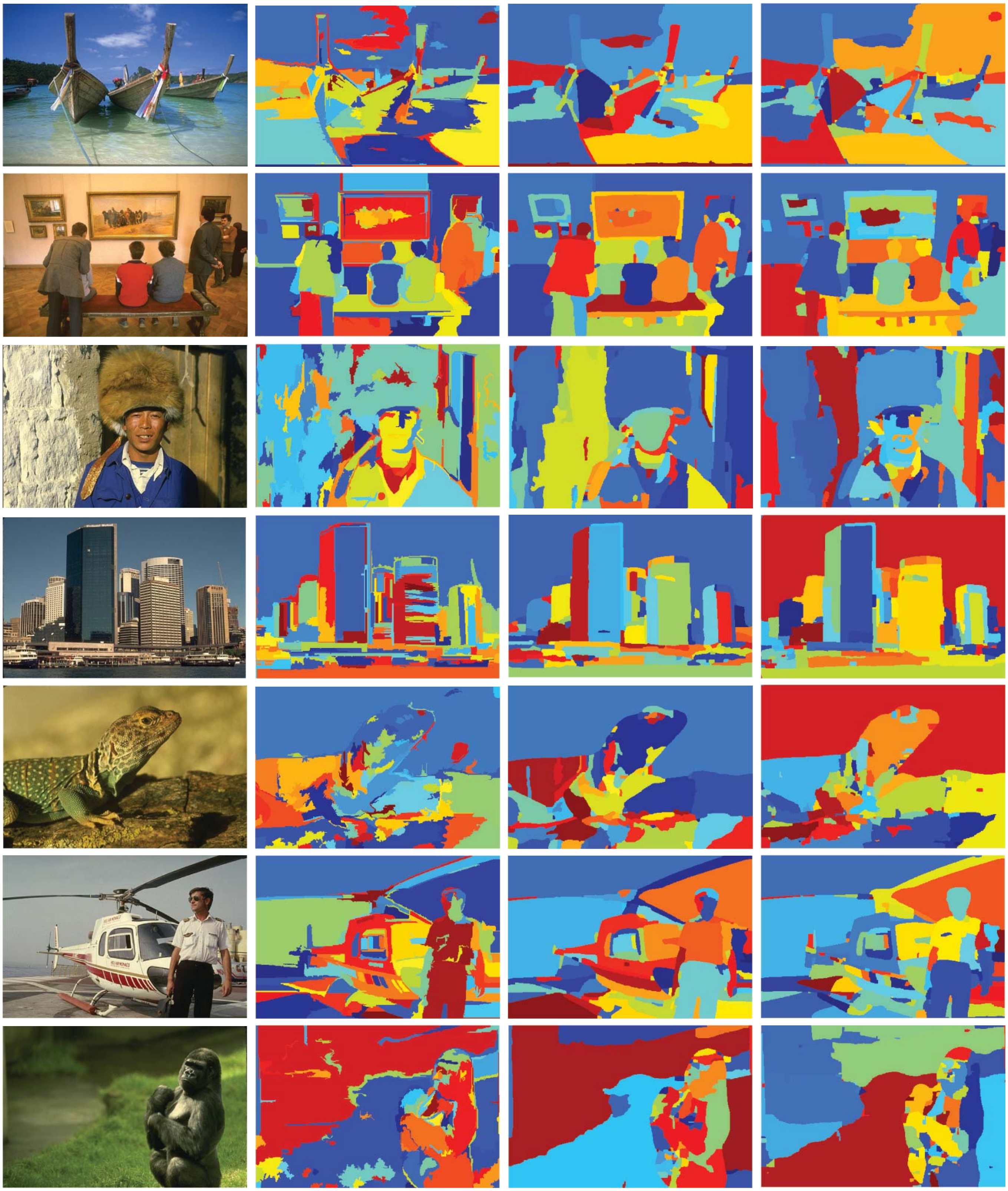
5. Experiments

We present experimental results for image segmentation on the BSDS500 database [4] with evaluations against several state-of-the-art image segmentation algorithms, i.e., the gPb-OWT-UCM [4], mean-shift [11], FH [14] and normalized cut [24] methods. In addition, we evaluate the boundary quality of the proposed algorithm against several state-of-the-art edge detection algorithms, i.e., the gPb-OWT-UCM [4], FH [14], Canny [5], SCG [23], Sketch tokens [18] and SE-MS [10] approaches.

For object detection, we present experimental results on the PASCAL VOC 2007 [13] benchmark database with comparisons to the selective search [26] method.

5.1. Evaluation on Image Segmentation

Experimental Setting. We first evaluate region quality



(a) Original image (b) FH [14] (c) gPb-OWT-UCM [4] (d) Ours

Figure 2. **Segmentation results.** First column: input images. Second column: results by the efficient graph-based segmentation method [14]. The generated segments are usually fragmented and less coherent to object boundaries. Third column: results by the gPb-OWT-UCM method [4]. Fourth column: results by the proposed SE-SF-OWT-UCM algorithm. Each segmentation result shown above has roughly 100 segments for comparisons.

on the BSDS500 database [4] which includes 200 training images, 100 validation images and 200 test images. Region quality is evaluated on the test set of the BSDS500 database according to segment covering, probabilistic rand index (PRI) and variation of information (VI). For the covering score, the segmented regions are evaluated by using three different quantities: optimal dataset scale (ODS), optimal image scale (OIS) and best covering criteria [4]. For PRI and VI measures, they are computed based on ODS and OIS. For the boundary benchmarks, the optimal scale for the whole dataset (ODS), the optimal value for each image (OIS) and average precision (AP) are used. We compute the SE at multiple scales (MS) in the experiments. In addition, we use fine grained gradient maps with 8 directions for structured edge and 8 decision trees.

Experimental Results. In Table 1, we present the region benchmark results compared to several state-of-the-art approaches [4, 11, 14, 24] on the BSDS500 database. The results show that the proposed algorithm generates comparable region in terms of quality as [4], but with 50 times speed-up on average according to Table 2 (average over 200 images in the test set of the BSDS500 database). In addition, our algorithm outperforms other state-of-the-art approaches as shown in Table 1. We also provide the boundary benchmark results with comparisons to the state-of-the-art approaches [4, 14, 5, 23, 18, 10] in Table 3. Experimental results show that the proposed algorithm performs efficiently and effectively in all metrics against the gPb-OWT-UCM method [4].

Table 3. **Boundary benchmarks on the BSDS500 database [4].** Results of seven state-of-the-art algorithms are presented. The proposed algorithm is based on structured edge (SE), steerable filters (SF), oriented watershed transformation (OWT) and ultrametric contour map (UCM). From the work of [10], SE achieves the best results using multiscale (MS) detection with 4 evaluated trees *T*. Our algorithm enjoys the merit of structured edge [10] and obtains better performance under OIS measurement. In addition, our algorithm outperforms the gPb-OWT-UCM method [4] in all evaluation metrics with 11 times speed-up as shown in Table 2.

	ODS	OIS	AP
Human	0.8	0.8	-
gPb-OWT-UCM [4]	0.73	0.76	0.73
FH [14]	0.61	0.64	0.56
Canny [5]	0.60	0.64	0.58
SCG [23]	**0.74**	0.76	0.77
Sketch tokens [18]	0.73	0.75	**0.78**
SE-MS, *T*=4 [10]	**0.74**	0.76	**0.78**
Ours	**0.74**	**0.77**	0.77

5.2. Evaluation on Object Detection

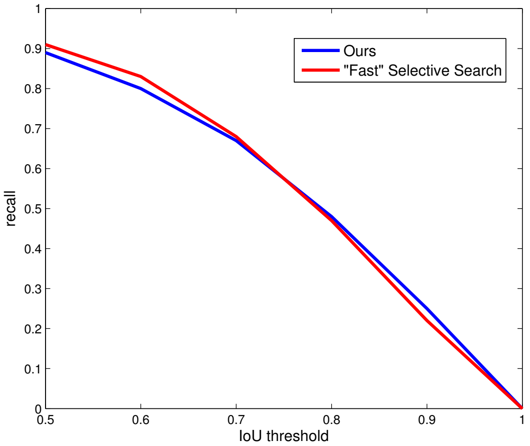


Figure 3. **Recall of object proposals.** We compare the recall rate between the proposed approach and the “fast” selective search algorithm [26]. The “fast” selective search method has better performance when the IoU ratio is below 0.75 as it generates more proposals according to Table 4. When the IoU ratio is larger than 0.75, the quality of proposals becomes a dominant factor and the proposed object proposal algorithm generates higher quality proposals than the “fast” selective search method.

Experimental Setting. We evaluate the proposed object proposal algorithm on the PASCAL VOC 2007 dataset [13], which contains 2501 training images and 4952 test images. Object instances in this database are all from 20 different categories, and each object is annotated by a bounding box.

We compare the proposed algorithm with the state-of-the-art selective search method [26] on the the PASCAL VOC 2007 test set. The “fast” selective search uses two different color spaces, HSV and Lab, two similarity measures, color as well as size similarity, and the parameter $K = [50, 100]$, which controls the number of segments generated by the graph-based segmentation algorithm [14]. We use the same criteria to generate object proposals in all the experiments.

Two metrics are used to evaluate the proposed object proposal algorithm against the state-of-the-art methods. First, we present the result of the recall based on the intersection-over-union (IoU) [13], $IoU(p, g) = \frac{|p \cap g|}{|p \cup g|}$, where p is the object proposal and g is the ground truth window. The object is detected if the IoU is greater and equal to a threshold. In our experiments, we evaluate the recall under different thresholds between [0.5, 1]. The proposed SE-SF-OWT-UCM algorithm generates different numbers of segments based on the parameter k. In addition, we present the results based on the mean average best overlap (MABO) metric [26]. To obtain the MABO, we first calculate the average best overlap (ABO) defined by

$$ABO = \frac{1}{|G^c|} \sum_{g_i^c \in G^c} \max_{l_j \in L} IoU(g_i^c, l_j)\,, \tag{3}$$

where G^c is the ground truth windows set for the class c, g^c is a ground truth window for the class c; L are the object proposals in a given image and l_j is one of the object proposals. The MABO is obtained by averaging ABO over all classes.

Experimental Results. Fig. 3 shows the recall rates of the proposed algorithm and the "fast" selective search method [26] on the PASCAL VOC 2007 test set. The "fast" selective search method [26] performs well when the IoU is smaller than 0.75, and the proposed approach achieves slightly better recall rates when the IoU is above 0.75. With the same MABO rate, Table 4 shows that the "fast" selective search method and the proposed algorithm generates roughly 2000 and 1000 proposals, respectively. We note that the "fast" selective search achieves better results for smaller IoU due to larger number of proposals with comparisons to the proposed method. As the threshold is increased, high quality proposals become a dominant factor, which indicates that the proposed algorithm generates higher quality proposals than the "fast" selective search method. In addition, the results of the MABO in Table 4 support that the proposed algorithm generates high quality object proposals. We note the runtime performance of the proposed algorithm is comparable to the "fast" selective search method as shown in Table 5.

Fig. 4 shows high quality object proposals generated by the proposed algorithm. The red boxes are the object proposals generated by the proposed algorithm and the yellow boxes are the ground truth windows. Overall, the proposals generated by the proposed algorithm have the highest intersection-over-union score that matches the ground truth annotations well.

Table 4. **Mean Average Best Overlap (MABO) and number of proposals on the PASCAL VOC 2007 testing set.**

Method	MABO	# windows
"Fast" selective search [26]	0.81	2007
Ours	0.81	1191

Table 5. **Runtime performance.** Average speed for generating object proposals on the PASCAL VOC 2007 test set.

Method	Computational time (s)
"Fast" selective search [26]	2.21
Ours	5.97

6. Conclusions

We have presented two novel region generation algorithms from the structured edge prediction for image segmentation and object detection. Motivated by the high quality and high efficiency of the structured edge, we link it with the state-of-the-art gPb hierarchical segmentation algorithm with steerable filters. We exploit an efficient algorithm for structured edge prediction to extract regions. In addition, it generates comparable region quality to the gPb hierarchical segmentation method. To generate high quality object proposals for object detection task, we apply the proposed SE-SF-OWT-UCM algorithm to extract regions and group them hierarchically. Experimental results show that the proposed algorithms perform favorably for image segmentation on the BSDS500 database and for the object detection on the PASCAL VOC 2007 database.

Acknowledgment. This work is supported in part by the NSF CAREER Grant #1149783 and NSF IIS Grant #1152576.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Ssstrunk. Normalized cuts and image segmentation. *PAMI*, 34(11):2274–2282, 2012.

[2] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *PAMI*, 34(11):2189–2202, 2012.

[3] P. Arbelaez. Boundary extraction in natural images using ultrametric contour maps. In *POCV*, 2006.

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

[5] J. Canny. A computational approach to edge detection. *PAMI*, 8(6):679–698, 1986.

[6] J. Carreira and C. Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *PAMI*, 34(7):1312–1328, 2012.

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

[8] A. Criminisi, J. Shotton, and E. Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, 7(2–3):81–227, 2012.

[9] P. Dollár, Z. Tu, and S. Belongie. Supervised learning of edges and object boundaries. In *CVPR*, 2006.

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

[11] C. Dorin and P. Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 24(5):603–619, 2002.

[12] I. Endres and D. Hoiem. Category-independent object proposals with diverse ranking. *PAMI*, 36(2):222–234, 2014.

[13] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[14] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, 2004.

[15] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *PAMI*, 13(9):891–906, 1991.

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

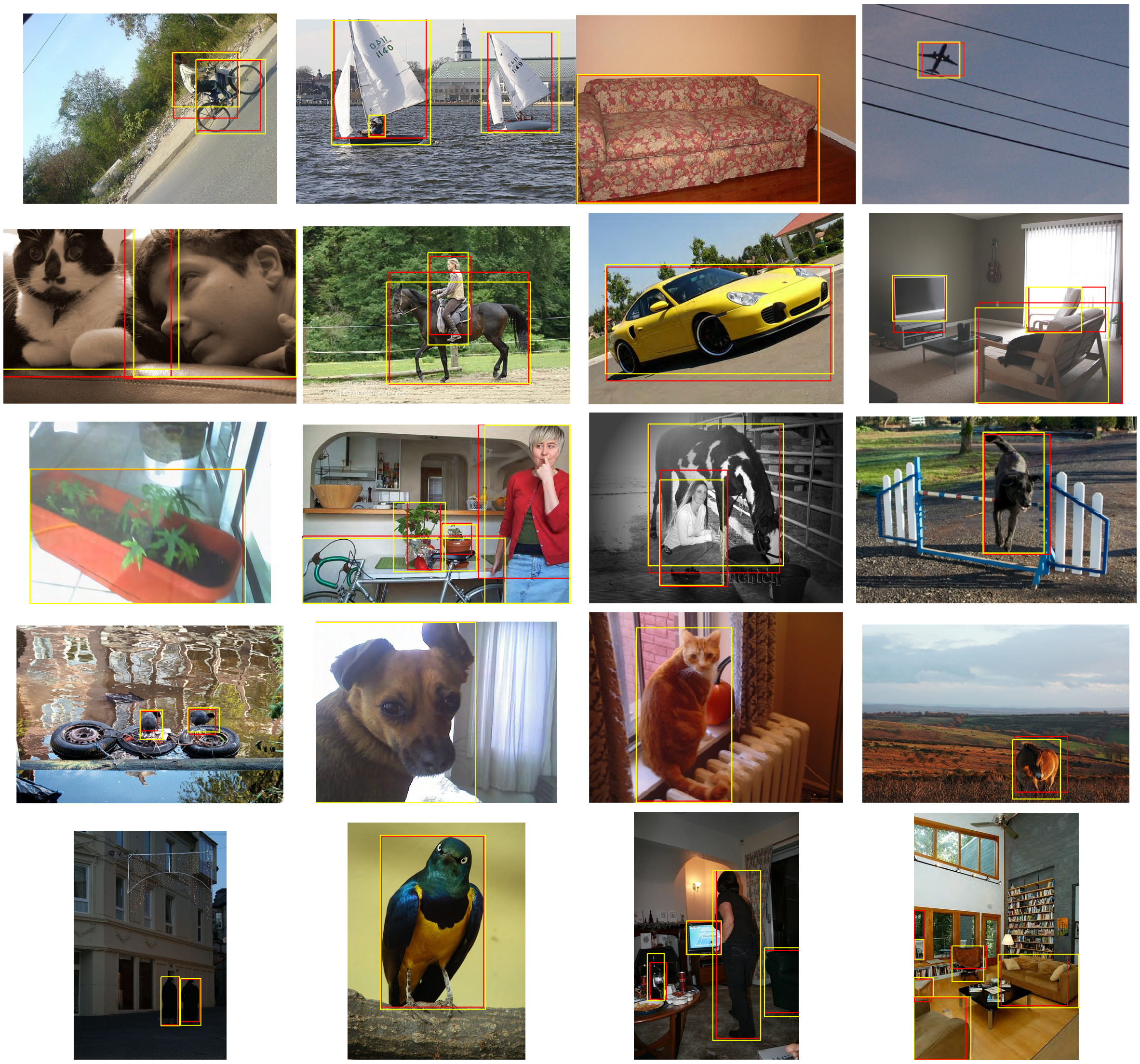


Figure 4. **Examples of object proposals by the proposed algorithm.** Red boxes: object proposals generated by the proposed algorithm. Yellow boxes: ground truth windows. The proposals in the red boxes have the highest intersection-over-union score. Overall, the proposed algorithm generates high quality object proposals.

[17] P. Kontschieder, S. R. Bulo, H. Bischof, and M. Pelillo. Structured class-labels in random forests for semantic image labelling. In *ICCV*, 2011.

[18] J. Lim, C. L. Zitnick, and P. Dollár. Sketch tokens: A learned mid-level representation for contour and object detection. In *CVPR*, 2013.

[19] S. Manén, M. Guillaumin, and L. Van Gool. Prime object proposals with randomized prim's algorithm. In *ICCV*, 2013.

[20] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color and texture cues. *PAMI*, 26(5):530–549, 2004.

[21] S. Nowozin and C. H. Lampert. Structured prediction and learning in computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6:185–365, 2011.

[22] P. Perona and J. Malik. Detecting and localizing edges composed of steps, peaks and roofs. In *ICCV*, 1990.

[23] X. Ren and L. Bo. Discriminatively trained sparse code gradients for contour detection. In *NIPS*, 2012.

[24] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 22(8):888–905, 2000.

[25] Z. Tu. Probabilistic boosting-tree: learning discriminative models for classification, recognition, and clustering. In *ICCV*, 2005.

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

[27] J. Yang, B. Price, S. Cohen, , and M.-H. Yang. Context driven scene parsing with attention to rare classes. In *CVPR*, 2014.

[28] J. Yang, Y.-H. Tsai, and M.-H. Yang. Exemplar cut. In *ICCV*, 2013.