Complexity-Adaptive Distance Metric for Object Proposals Generation

Yao Xiao Cewu Lu Efstratios Tsougenis Yongyi Lu Chi-Keung Tang

The Hong Kong University of Science and Technology
{yxiaoab,lucewu,tsougenis,yluaw,cktang}@cse.ust.hk

Abstract

Distance metric plays a key role in grouping superpixels to produce object proposals for object detection. We observe that existing distance metrics work primarily for low complexity cases. In this paper, we develop a novel distance metric for grouping two superpixels in high-complexity scenarios. Combining them, a complexity-adaptive distance measure is produced that achieves improved grouping in different levels of complexity. Our extensive experimentation shows that our method can achieve good results in the PASCAL VOC 2012 dataset surpassing the latest state-of-the-art methods.

1. Introduction

Object Proposals has become indispensable for object detection, providing the latter with a set of image regions where objects are likely to occur. A proposal generator should produce an ideal number of candidate bounding boxes to include most observable objects (high recall) in a considerably short time (high efficiency).

Currently, the mainstream methods [19, 2] partition the image into hundreds of superpixels, and then group them under certain criteria to form object proposals. This strategy is reasonable because an image object is composed of a set of superpixels (or in fact, pixels). Much research has therefore been dedicated to seeking a better superpixels grouping strategy.

A powerful distance metric is conducive to acceptable grouping of superpixels regardless the grouping strategies (e.g., greedy grouping, cluster-based grouping [17]). In conventional methods [19], the distance metric computes the difference between two superpixels in terms of an aggregate measure. That is, all the elements (superpixels) participate in calculating the measure. Take color cue as an example [19]. Color histograms are computed to represent a superpixels set which involves *all* superpixels in the set.

While existing distance metrics are well suited to low-complexity superpixels grouping, they become less effective in high-complexity grouping scenarios. Figure 1(a) shows that a typical low-complexity case. Here, the aggregate measure, mean color, can adequately capture the distance between the two superpixel sets. In a typical high-complexity case, however, as shown in Figure 1(b), while the mean color of the two superpixel sets are significantly different, it is still reasonable to group them into a single proposal, since the two superpixel sets can be well connected by superpixel A and B. Typical aggregate measures fail to capture the inherent complexity when the superpixel set contains highly diversified elements.

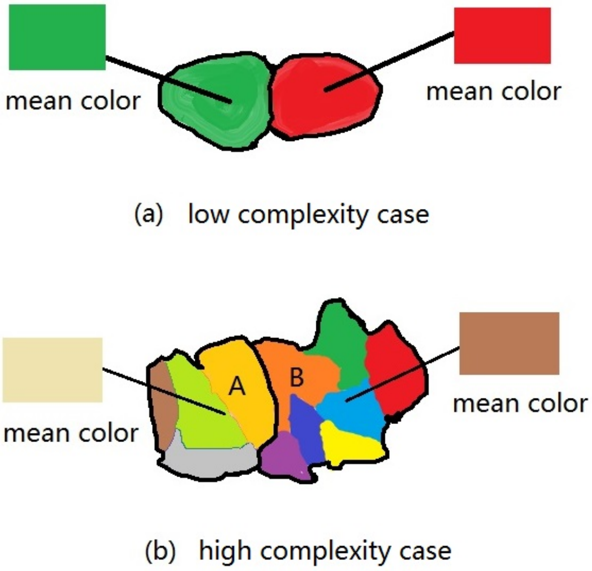


Figure 1. Color distance metrics. (a) The mean color is well behaved to delineate low-complexity superpixels, but (b) such aggregate measure fails to reflect the distance between two high-complexity superpixel sets.

The lesson learned from the above example is twofold: (1) the distance metric should be adaptive to the complexity of superpixel sets, and (2) a new distance metric is required for high-complexity superpixel sets. For example, while exceptions exist if we consider the minimum distance between two elements from the respective adjacent superpixels, which are A and B in this example and the distance is small here, the two superpixel sets will be merged to form a single proposal.

In this paper, we contribute a novel distance metric for grouping two superpixel sets which is adaptive to their complexity. Specifically, our distance metric combines a "low-complexity distance" and a "high-complexity distance" to make it adaptive to different complexity levels. Extensive experiments have been carried out on the Pascal VOC 2012 dataset. The results show our method can achieve a better recall performance compared with state-of-the-art methods.

2. Related Work

Efficient segmentation [14] [12] and superpixel composition [11] algorithms have given rise to the design of novel detection proposal methods that recently achieved state-of-the-art performance [3] [21]. Sliding window, as the initial solution for generating candidates location, [7] [10], performs an exhaustive search in all image locations for identifying the object. Although high detection performance is reported, they fall short of the efficiency requirement since the number of examined windows can reach 10^6. In practice, an object proposal can be represented as either a bounding box or a segmented region. One can refer to [15] for an extended review and performance evaluation of object proposal methods.

Bounded Boxes Initially, the bounded box process has been evaluated in [13] for incorporating possible detected objects synthesized from grouping pixels. The term "objectness" has been introduced in [1] quantifying the existence of an object within the tested window / box. Following [1], in [2], the authors infer objectness based on a score output from a bayesian framework that combines multiple image cues such as multiscale saliency, color contrast, edge density, and superpixels straddling. Selective search [19] is an efficient algorithm widely applied to achieve good time performance while maintaining high recall. Inspired by selective search, [16] proposed randomized low-level merging based also on learning weights. Bing [6] applied a simple linear classifier over edge features trained in a sliding window manner and achieve state-of-the-art time performance in high efficiency. Recently, Edge Boxes [21] justified that their a score based on edges wholly enclosed within a bounding box can lead to high recall and fast results.

Segmented Regions Apart from object proposals represented as bounding boxes, a number of works introduced the segmentation-wise proposals. CPMC, introduced in [5], a process that generates overlapping segments iteratively initialized by diverse seeds, manage to reduce to 10.4 object proposals per image. CPMC [5] uses 34 features (related to graph, region, Gestalt properties) and complements with SIFT, HOG, etc, and the random Forest regressor based on largest overlap with ground truth, followed by reranking using MMR to suppress redundant and similar segments to report high-saliency object segments. Following the spirit of CPMC, [9] applies an hierarchical segmentation along with a learned regressor that detects boundaries between surfaces with different orientations, in contrast to [5] that sets the image border to the background. In [17], selective search is applied for merging superpixels and then CPMC for further merging resulting to the final segments. Rigor algorithm, also based on CPMC, manages to achieve high quality segments by reducing at the same time the computational cost minimizing heavy computations. [3] achieved high recall and efficiency by speeding up the eigenvector estimation for contour globalization that made it possible to extend also the searching space combined from multiscale regions. [20] applied category-dependent shape prior to accurately localize objects. [18] proposed training a cascade boundary classifiers to adopt to the segmentation scale. Recently, [4] took advantage of the PageRank algorithm for hierarchical merging; a set of candidate regions (up to three) is generated followed by a selection stage for removing most of the segments.

Although the majority of segmentation-wise object proposal methods output high quality segmentation masks, their computation cost is high compared to bounding box methods. In addition, Markov Random Field based segmentation methods [5] [17] generate segmented region proposals based on initial seeds, which makes it difficult to capture the characteristics of the whole object. Although our method has similar philosophy to [19] by considering the hierarchical structure to form proposals, we clearly differentiate from [19] in the following: (1) our grouping process is not limited to local neighboring superpixels; instead we consider a graph distance for regularization thus allowing disjoint segments to merge; (2) the iterative propagation of low level features applied in [19] is avoided during superpixels merging; (3) we use a more efficient approach where feature distance between each initial element is initially computed and updated cheaply throughout the clustering procedure.

3. Our Approach

In this section, we introduce our complexity-adaptive distance. We adopt the grouping scheme of [19]. Initially, a number of superpixels are generated by applying the efficient segmentation method [11]. In each iteration, two superpixel sets with the smallest distance are merged. However, different from [19], low-level features (histogram) are not propagated during superpixel merging. Our main contribution is the introduction of a novel approach that efficiently computes superpixel distance. Thus, we focus on our design in the following: first, the basic distances are presented. Then, our new low-complexity and high-complexity distances will be detailed. Finally, we discuss our complexity-adaptive distance based on low and high-complexity distance.

3.1. Basic Distance

In this section we introduce a number of basic distance metrics which will be a part of our low-complexity and high-complexity distance functions.

Color and texture feature distance Color and texture are important low-level features for describing regions; objects or individual parts with uniform appearance share similar colors and textures. For color feature, we obtain a normalized one-dimensional color histogram h_c, with 20 bins per color channel. For texture, we take Gaussian derivatives in eight orientations forming a normalized texture histogram h_t with 10 bins for each orientation per color channel. The distance of color and texture histograms d_{ct} between superpixel i, j is measured using L1 distance and then summed together:

$$d_{ct}(i,j) = \|h_c(i) - h_c(j)\| + \|h_t(i) - h_t(j)\| \tag{1}$$

Graph Distance Although our algorithm does not restrict grouping exclusively local neighboring superpixels, incorporating spatial affinity is still useful for preserving the intrinsic image hierarchy. Therefore, we use graph distance to regularize the grouping process to prefer spatially close superpixels. We construct a connectivity graph G based on our initially segmented superpixels from [11]. The graph distance for each pair of nodes $d_g(i,j)$ is obtained via the Floyd-Warshall algorithm with the edge weight set to 1 and each node corresponding to a single superpixel. The graph distance D_g between each set of superpixels S_m, S_n can be defined as

$$D_g(m,n) = \min\{d_g(i,j) | i \in S_m, j \in S_n\} \tag{2}$$

While enforcing spatial affinity is significant for grouping homogeneous segments at low-complexity level, the weight of the graph distance should decrease accordingly with increasing complexity to achieve non-local affinity. By searching in a larger area beyond a local region, highly-complex object segments with disjoint but similar parts are more likely to be grouped together. Moreover, non-local search makes it more robust against occlusion. Thus relaxing the spatial constraints upon grouping larger superpixel set is beneficial. As will be shown, in practice, we execute a cascade of spatial constraint strengths to encourage different diversity during hierarchical grouping.

Edge cost Edge cost measures the edge responses along the common border of the segments. The initial edge map E is generated using the Structured Edge algorithm [8]. The edge cost for each neighboring segments is calculated by summing up the edge responses within the common border pixels and then normalized by the length of the common border. Disjoint segments are set to have zero edge response. Denote the common border pixels set as $l_{i,j}$, then

$$d_e(i,j) = \begin{cases} \frac{1}{|l_{i,j}|} \sum_{p \in l_{i,j}} E(p) & \text{if } |l_{i,j}| \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The edge cost between two sets of superpixels is

$$D_e(m,n) = \begin{cases} \frac{\sum_{i \in S_m, j \in S_n} |l_{i,j}| d_e(i,j)}{\sum_{i \in S_m, j \in S_n} |l_{i,j}|} & \text{if } \sum_{i \in S_m, j \in} l_{i,j} \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Edge response is typically weak within homogeneous regions while relatively strong between incoherent segments or superpixels. As will be detailed in the following, we set a large weight for edge cost for low-complexity segments, and adaptively adjust the weight according to the superpixel set complexity.

3.2. Low and High-Complexity Distance

Given a superpixel, we should consider its distance differently depending on its complexity. We first detail the design and then discuss the rationale behind it.

Consider two super-pixel set S_m and S_n, we define the nearest and farthest pairwise distance as,

$$D_{\min}(m,n) = \min\{d_{ct}(i,j) | i \in S_m, j \in S_n\} \tag{5}$$

$$D_{\max}(m,n) = \max\{d_{ct}(i,j) | i \in S_m, j \in S_n\} \tag{6}$$

$D_{\max}$ and $D_{\min}$ can be used to indicate respectively the low and high-complexity distance of two given superpixel sets.

A small $D_{\max}$ indicates that all the elements in the two sets are similar, meaning that they are of low complexity. Thus $D_{\max}$ suits for low-complexity region merging. In contrast, a small $D_{\min}$ means that the two sets are connected by at least two elements from the respective two superpixel sets. Therefore, a small $D_{\min}$ is a reasonable indicator for merging in high-complexity scenarios.

Figure 2 gives an example how these two distances interact and contribute in different complexity levels. Initially, the three superpixel sets A, B, and C are of high complexity level. The background wall is more similar to the baby's head than his clothes in terms of color and texture. Using the minimum $D_{\max}$ will result in merging the head with background wall. On the contrary, D_{min} will link part of head to the hand allowing a more semantic grouping.

By combining $D_g(m,n)$ and $D_e(m,n)$, our low-complexity distance D_L and high-complexity distance D_H

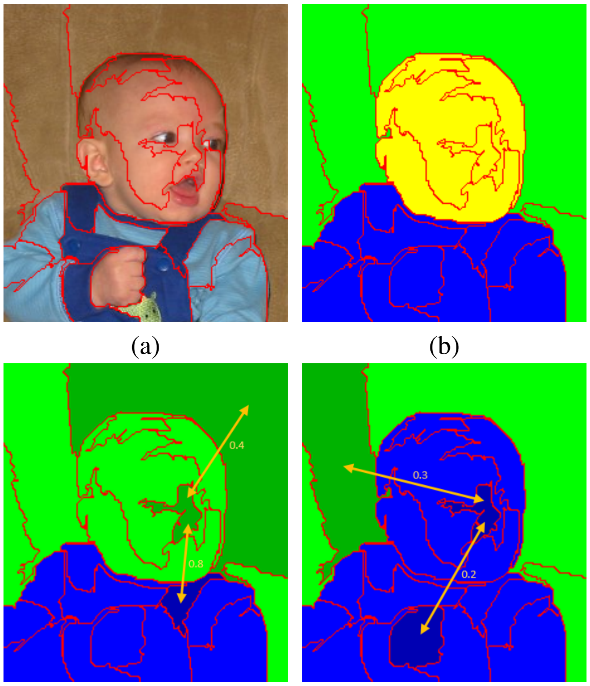

(a) (b) (c) (d)

Figure 2. (a) Superpixels. (b) three sets A, B, and C are indicated by green, yellow, blue superpixels. (c) $D_{\max}$ (d) $D_{\min}$. Set B will be merged through searching for minimum $D_{\max}$. However B is closer to C in terms of the minimum $D_{\min}$ which produces an accurate semantic grouping.

are respectively given by

$$D_L(m,n) = D_{\max}(m,n) + D_e(m,n) + D_g(m,n) \quad (7)$$

and

$$D_H(m,n) = D_{\min}(m,n) + bD_g(m,n) \quad (8)$$

Here, the graph distance $D_g(m,n)$ serves as the spatial constraint to prefer merging spatially close segments at the beginning. Then, we will reduce the influence of $D_g(m,n)$ in the high-complexity distance by multiplying a factor $b, 0 < b < 1$, which serves as a lower bound of the spatial constraint.

In practice edge response is mostly weak inside homogeneous regions but strong between dissimilar segments. Thus, it is more useful to group coherent segments in low-complexity level than in high-complexity level, which explains that $D_e(m,n)$ is used only in D_L.

3.3. Complexity-Adaptive Distance

By combining the low and high complexity distance and complexity level factor $\rho_{n,m}$, we define the complexity-adaptive distance as

$$D_{total} = \rho_{m,n} D_L + (1-\rho_{m,n}) D_H + \eta D_s \quad (9)$$

Here D_s is defined as

$$D_s(m,n) = r_m + r_n \quad (10)$$

where r_m and r_n are the respective sizes of super-pixels m and n. In the bottom-up hierarchical grouping methods [19, 16], this size term plays an important role in ensuring that small-sized superpixels will be merged first. In practice η is set to 2.

The function $\rho(m,n)$ indicates the complexity level of two sets m and n; we denote the element number of two sets respectively as T_m and T_n, and the total number of superpixels as T. Due to the reason that superpixels are mostly homogeneous regions, we can use the superpixel number to indicate the complexity level. We define

$$\alpha = -\log_2 \frac{T_m + T_n}{T} \quad (11)$$

$$\rho_{m,n} = (1 + \exp(-\frac{\alpha - \lambda}{\sigma}))^{-1} \quad (12)$$

where α represents the complexity level and λ controls the boundary of different complexity levels. Since the superpixel number of each set grows exponentially, we use logarithm representation here. The σ parameter affects the transition smoothness. In our experiments σ is set to 0.1. We found that changing σ does not significantly affect the experimental results. When $\alpha = 0$ the merged output is the whole image, which corresponds to the highest complexity level. The upper bound of α is determined by the total number of superpixels T so it may vary depending on images.

3.4. Proposals ranking

The proposals produced above may contain a large number of redundant regions, such as single superpixel within the background. To prune the proposals which are unlikely object candidates, we compute a score for each output bounding box. The score is calculated based on edge evidence, which is powerful for indicating the existence of an object. Many bounding box proposal methods [8] [1] [6] also use edges as the main cue for searching object contours.

We make use of the edge cost of neighboring segments. Suppose proposal p is formed by set S_p, then the score is computed as the summation of the edge cost on outer boundary, and then normalized by its length:

$$score(p) = \frac{\sum_{i \in S_p, j \notin S_p} |l_{i,j}| d_e(i,j)}{(\sum_{i \in S_p, j \notin S_p} |l_{i,j}|)^{\kappa}} \quad (13)$$

where κ is set to be less than 1 to favor larger windows. The output proposals are ranked based on their scores in descending order.

	HSV	Lab	Combination
MABO	0.798	0.788	0.814
AUC	0.590	0.572	0.616

Table 1. Comparison of performance in different color spaces in terms of mean average best overlap (MABO) and area under curve (AUC).

4. Experiments

In this section we demonstrate the performance of our method when compared to the-state-of-the-art algorithms. Similar to [16] [17] [3], we evaluate our algorithm's performance on the PASCAL VOC 2012 dataset. The dataset contains 11540 images and 27450 objects in 20 different categories.

The accuracy of bounding box proposals is typically measured by the Intersection over Union (IoU) measure. The IoU is defined as the intersection area of the box proposal and the ground-truth box divided by the area of their union. In PASCAL VOC and ILSVRC object detection, an IoU threshold of 0.5 is used to indicate successful detection. In our experiments, we compute the recall as the fraction of ground-truth objects with the best IoU above a predefined threshold. While the recall is largely related to the number of proposals, we rank the candidate boxes by the scores, and discard low-ranked boxes to better handle box numbers.

Two measurement criteria are used for overall performance evaluation: the Mean Average Best Overlap (MABO), introduced in [19], corresponds to the mean value of average best overlap considering all object categories, while the Area Under Curve (AUC) is the total area under the "recall versus IoU threshold" curve [21] [15]. Both measures are adequate to reflect the overall quality of the results for a given number of candidates.

4.1. Color space

According to [19] [16], multiple color spaces are helpful since each color space has a distinct invariance property for lighting conditions. A combination of complementary color spaces tends to produce higher quality results. In this paper, we examine the HSV and Lab space and their combination on an average number of about 2200 proposals. As shown in Figure 3 and Table 1, operating in the HSV space produces better performance than in the Lab space; however, their combination achieves the best performance than each individual color space. In our experiments we thus utilize color spaces combination to improve proposal quality.

4.2. Low and High Complexity Distance

We examine the algorithm's performance on using individual low-complexity distance, a low-to-high transition strategy, and their combination as well. According to Eq. 12 λ handles the transition. We can adjust λ to control the transition from low and high complexity; if λ is set to 0, then $\rho_{m,n}$ is close to 1 for most complexity levels resulting in major contribution by the low level complexity distance D_L; if λ is set to a large value, e.g., $\lambda = 10$, then D_H will become dominant.

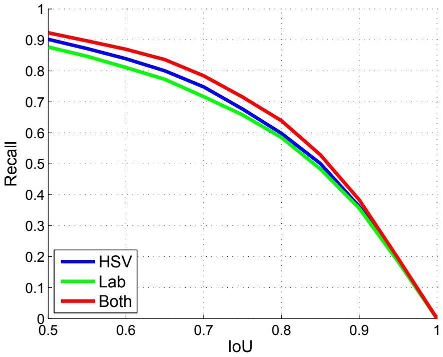


Figure 3. Evaluation of performance in different color space using 2200 candidates.

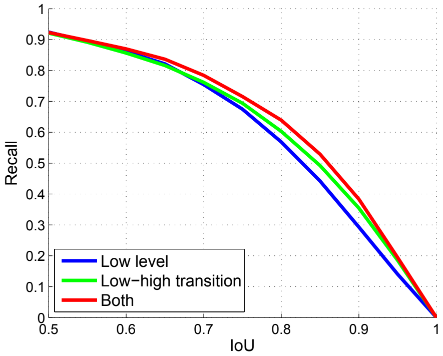


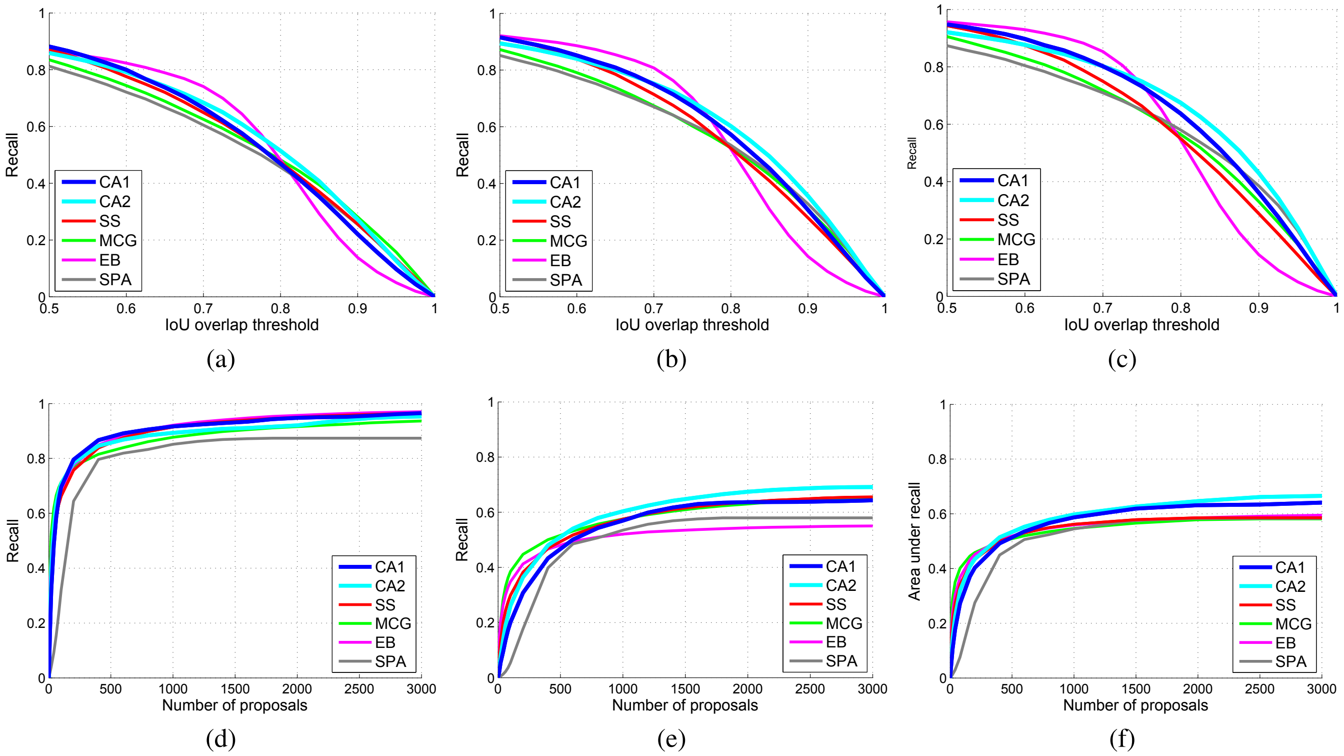
Figure 4. Evaluation of performance on applying different distance metrics using 2000 candidates.

	Complete	Transition	Combination
MABO	0.804	0.793	0.813
AUC	0.597	0.579	0.615

Table 2. Comparison of performance using different distance metrics in terms of MABO and AUC.

In this experiment we compare the results of $\lambda = 0$, $\lambda = 6$ and the combination of both. The reason we choose $\lambda = 6$ is that the typical range of the total number of initial superpixels is 300–500. The "threshold" at which a set of superpixels switches from low to high complexity level is about 5–10. It is typically larger than the superpixel number in a homogeneous region. For combination, we concatenate both results and select highly-ranked candidates.

Figure 4 and Table 2 show the results. The average number of candidates tested is 2000. We can see that the



(a) (b) (c)

(d) (e) (f)

Figure 5. Comparison of our method with various state-of-the-art methods, Selective Search [19], MCG [3], EdgeBox [21] and Rantalankila [17]. CA1 and CA2 respectively denote the two settings of our parameters. (a)–(c) are the recall vs intersection over union (IoU) threshold curves using different number of proposals, i.e., 500 proposals for (a), 1000 proposals for (b) and 2000 proposals for (c). (d) and (e) are the recall vs candidate number curves under IoU threshold of 0.5 and 0.8, respectively. (e) is the area under curve (AUC) vs candidate number.

transition strategy works better than using individual low-complexity distance, while their combination can surpass both. The basic reason for combining both individual and the transition strategy is to make our system robust to cases where the nearest pairwise distance D_{min} may result in false grouping, i.e., a scene with multiple objects with similar appearance that are close to each other; the greedy searching strategy for partially similar segments may first group similar parts of different objects which will lead to an irreversible error. This undesirable situation justifies that the two strategies are complementary to each other and should work in synergy to achieve high-quality results.

4.3. Comparison with State-of-the-Art Methods

In this section, we compare our complexity-adaptive distance algorithm to the following state-of-the-art methods [19] [17] [3] [21]. Two groups of evaluation are set: The top row in Figure 5 shows the recall versus IoU threshold curve for different number of proposals. The bottom row of Figure 5 shows the recall under varying number of proposals at a given IoU threshold. To control the candidates number to a specific value for the examined methods [19] [17] [3] [21], we refer to [15] and apply similar approaches:

1. Selective Search [19] returns a randomized priority. The bounding boxes are ranked by priority and highly ranked ones are selected as the evaluation set.
2. MCG [3] returns sorted proposals. The first n candidates are selected as the evaluation set.
3. Rantalankila [17] has no parameters to control the output proposal number. We choose the first n proposals as evaluation set.
4. Edgebox [21] returns the scores of candidates. High scoring proposals are selected as the evaluation set.

For comparison, we use two different settings in our method. Recall that b is the lower bound of spatial constraint. In the first setting CA1, we set $\lambda = 0, 6$ and $b = 1, 0.4$ to diversify the spatial constraints. This setting results in a number of 4 branches producing on average a total of 3000 candidates. In the second setting CA2, we set $\lambda = 0, 3, 6, 9$ and $b = 1, 0.7, 0.4$ to expand the diversity. As a result it produces an increased number of candidates totaling 3760. Notice that despite its number of branches is increased to 12, most of the output bounding boxes are duplicates and thus filtered out. To obtain a larger proposal

	500 Candidates		1000 Candidates		2000 Candidates		
Methods	MABO	AUC	MABO	AUC	MABO	AUC	time
Selective Search [19]	0.771	0.517	0.799	0.562	0.812	0.585	5.4
MCG [3]	0.757	0.510	0.782	0.547	0.802	0.578	33.4
EdgeBox [21]	0.755	0.520	0.782	0.559	0.798	0.585	0.3
SPA [17]	0.736	0.487	0.776	0.545	0.800	0.583	16.7
CA1	0.768	0.517	0.809	0.585	0.836	0.631	6.3
CA2	**0.775**	**0.536**	**0.812**	**0.597**	**0.840**	**0.647**	22.6

Table 3. Comparison results of MABO and AUC using 500, 1000, and 2000 candidates; CA1 and CA2 respectively are the two parameter settings used in our method. Running time of all the methods are also shown.

number, the selection of color spaces and parameter settings should be extended.

From Figure 5, we can see both CA1 and CA2 settings improve significantly for higher IoU values and larger number of boxes. The high values of MABO and AUC indicate that our method generates more accurate candidate boxes compared to the other tested algorithms, which is a desirable property in object detection. For the case of small number of boxes, our performance slightly drops at low IoU with small candidate number; the reason is that inaccurate bounding boxes have weaker edge response along the segment boundary, which leads to low ranking score thus making them unlikely to be chosen. On the contrary, the high IoU performance is greatly boosted by the combination of multiple strategies. Selective Search [19] and Edgebox [21] perform quite well for low IoU candidates, but they degenerate rapidly with the increase of IoU threshold. MCG[3] and Rantalankila's method [17] achieve competitive accuracy at high IoU threshold owing to their segmentation-based property. However at low IoU they are not able to achieve high recall even with considerable large number of proposals.

Table 3 tabulates the MABO and AUC results of all the tested methods using 500, 1000 and 2000 candidates where our algorithm's efficiency is also compared to the-state-of-the-art methods. Although not equally efficient as [21], our method still produces high-quality results with acceptable execution time comparing to [17] [3]. However, we note that different branches may share parts of the process when grouping low complexity level, thus it is possible to design a more efficient scheme to manage the grouping process to achieve a significant reduction in running time, which is our future work to pursue.

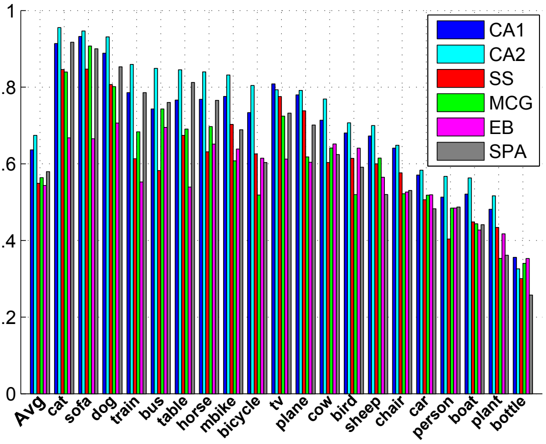


Figure 6. Evaluation of performance on applying different distance metrics using 2000 candidates.

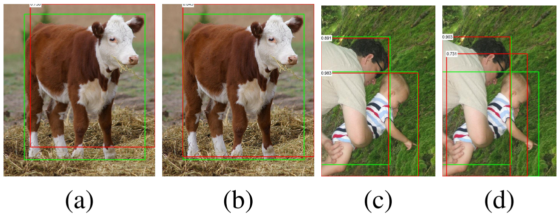


Figure 7. Example of success and failure cases of applying our complexity-adaptive distance. (a,c) show results with only low level distance and (b,d) are produced through combining low and high level distance. In (a) the hoof is not included, but is successfully enclosed in (b). In (c) local searching produces accurate bounding box. However in (d) the daddy's face mistakenly appears in the baby's bounding box, due to the fact that multiple complex objects co-exist and interfere with each other.

We further visualize the recall values for each of the 20 categories in Figure 6. For fair of comparison, the average number of output proposals are set to be 2000 for all the methods we tested earlier. The overlap threshold is set to be 0.8. It can be observed that our method reaches the highest recall for almost all of the categories, with either or both parameter settings. This result shows the robustness and generality of our complexity-adaptive distance measurement for different objects. The leftmost bars show the overall recalls. Note that the overlap at 0.8 is challenging, and so we believe our method can benefit object detection task with better localization. We further test our algorithm on BSDS dataset and also achieve state-of-the art results. All the figures are included in supplementary material.

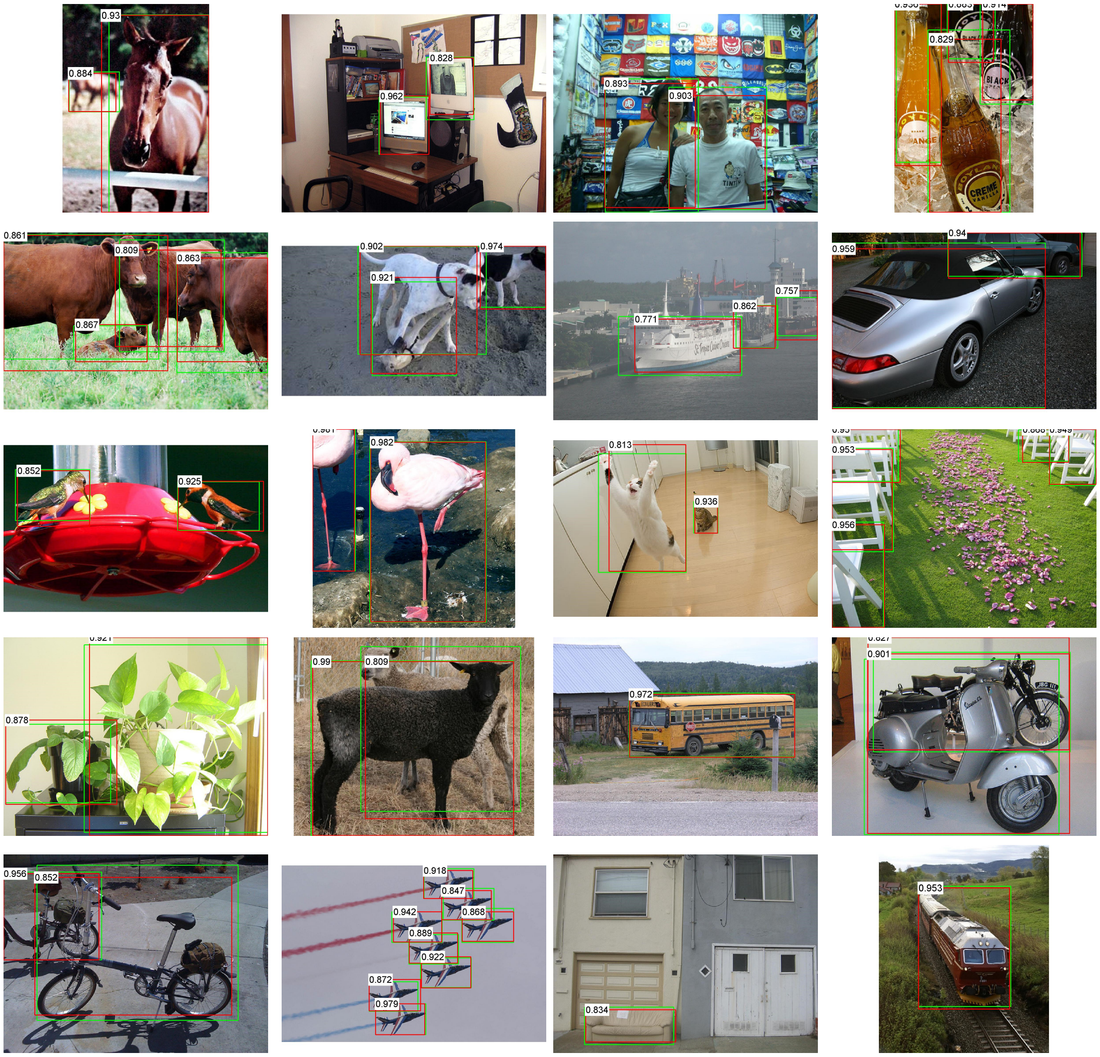


Figure 8. Sample candidates produced by our method. Green bounding boxes are ground truth and red ones are proposals with the highest IoU scores, marked at the top-left corner of each box.

5. Conclusions and Future Work

In this paper, we introduced a novel method for computing complexity-adaptive distance between superpixel sets. Based on the hierarchical grouping structure, superpixels are hierarchically merged based on a new distance metric. To adapt to objects with distinct complexity, low and high complexity level distance are computed. A complexity level factor is measured to adjust the combination ratio of low and high level distance. Through complexity-adaptivity we can achieve highly accurate bounding box candidates for a great variety of objects. Our algorithm has been extensively tested on the VOC PASCAL 2012 dataset and reported state-of-the-art performance, surpassing the latest state-of-the-art methods in the examined cases. In addition, our simple and novel distance measurement manages to keep the running time practical for real applications.

In the future we will seek ways to accelerate our methods. Currently, the candidate boxes produced by different branches (settings) contain a large number of redundant and duplicate proposals. We believe there is a great potential in reducing the time cost by letting different branches share part of the processes while merging low level segments. Another concern is to apply our method in object detection task to testify its effectiveness. In the recent ILSVRC 2014, most of proposal-based detection systems, especially those based on RCNN, require high-quality object proposal candidates. Since our approach provides more accurate bounding box than the widely-adopted [19], we believe this work can further improve the performance of these detection systems. The executables of the new distance metric will be available in the project page http://www.cse.ust.hk/∼yxiaoab/cvpr2014/CADM.html.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 73–80. IEEE, 2010. 2, 4

[2] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2189–2202, 2012. 1, 2

[3] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. CVPR, 2014. 2, 5, 6, 7

[4] B. Bonev and A. L. Yuille. A fast and simple algorithm for producing candidate regions. In *Computer Vision–ECCV 2014*, pages 535–549. Springer, 2014. 2

[5] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3241–3248. IEEE, 2010. 2

[6] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In *IEEE CVPR*, 2014. 2, 4

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005. 2

[8] P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1841–1848. IEEE, 2013. 3, 4

[9] I. Endres and D. Hoiem. Category independent object proposals. In *Computer Vision–ECCV 2010*, pages 575–588. Springer, 2010. 2

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010. 2

[11] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004. 2, 3

[12] M. Galun, E. Sharon, R. Basri, and A. Brandt. Texture segmentation by multiscale aggregation of filter responses and shape elements. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 716–723. IEEE, 2003. 2

[13] C. Gu, J. J. Lim, P. Arbeláez, and J. Malik. Recognition using regions. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1030–1037. IEEE, 2009. 2

[14] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 654–661. IEEE, 2005. 2

[15] J. Hosang, R. Benenson, and B. Schiele. How good are detection proposals, really? *arXiv preprint arXiv:1406.6962*, 2014. 2, 5, 6

[16] S. Manen, M. Guillaumin, and L. V. Gool. Prime object proposals with randomized prim's algorithm. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2536–2543. IEEE, 2013. 2, 4, 5

[17] P. Rantalankila, J. Kannala, and E. Rahtu. Generating object segmentation proposals using global and local search. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 1, 2, 5, 6, 7

[18] Z. Ren and G. Shakhnarovich. Image segmentation by cascaded region agglomeration. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2011–2018. IEEE, 2013. 2

[19] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013. 1, 2, 4, 5, 6, 7, 8

[20] D. Weiss and B. Taskar. Scalpel: Segmentation cascades with localized priors and efficient learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013. 2

[21] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *Computer Vision–ECCV 2014*, pages 391–405. Springer, 2014. 2, 5, 6, 7