Geodesic Object Proposals

Philipp Krähenbühl[1] and Vladlen Koltun[2]

[1] Stanford University, USA
[2] Adobe Research

Abstract. We present an approach for identifying a set of candidate objects in a given image. This set of candidates can be used for object recognition, segmentation, and other object-based image parsing tasks. To generate the proposals, we identify critical level sets in geodesic distance transforms computed for seeds placed in the image. The seeds are placed by specially trained classifiers that are optimized to discover objects. Experiments demonstrate that the presented approach achieves significantly higher accuracy than alternative approaches, at a fraction of the computational cost.

Keywords: perceptual organization, grouping.

1 Introduction

Many image parsing pipelines use sliding windows to extract densely overlapping bounding boxes that are then analyzed [7,11,22]. This approach has well-known disadvantages: the number of bounding boxes per image must be very large to achieve good recognition accuracy, most of the computational effort is wasted on futile bounding boxes, and the rectangular boxes aggregate visual information from multiple objects and background clutter. Both recognition accuracy and computational performance suffer as a result.

An alternative approach is to use segmentation to extract a set of proposed objects to be analyzed [5,9,12,15,21]. Ideal object proposals of this kind should encapsulate the visual signal from one object and have informative boundary shape cues that can assist subsequent tasks. Image analysis pipelines based on such segmentation-driven object proposals have recently achieved state-of-the-art performance on challenging benchmarks [3,4].

In this paper, we present an approach that produces highly accurate object proposals with minimal computational overhead per image. Our key idea is to identify critical level sets in geodesic distance transforms computed for judiciously placed seeds in the image. The seeds are placed by classifiers that are trained to discover objects. Since the geodesic distance transform can be computed in near-linear time and since each computed transform is used to generate proposals at different scales, the pipeline is extremely efficient.

Our experiments demonstrate that the presented approach achieves significantly higher accuracy than alternative approaches as measured by both bounding box overlap and detailed shape overlap with ground-truth objects. It is also substantially faster, producing a high-performing set of object proposals for a raw input image in less than a second using a single CPU thread.

D. Fleet et al. (Eds.): ECCV 2014, Part V, LNCS 8693, pp. 725–739, 2014.

Fig. 1. Object proposals (in red) produced by the presented approach for two images from the PASCAL VOC2012 dataset

2 Overview

Our overall proposal generation pipeline is illustrated in Figure 2. Given an image, we compute an oversegmentation into superpixels and a boundary probability map that associates a boundary probability with each superpixel edge. This step uses existing techniques. Next we identify a set of seed superpixels. The goal is to hit all objects in the image with a small set of automatically placed seeds. In Section 3 we describe a reasonable seed placement heuristic that outperforms other heuristic approaches, such as regular seed placement, random seed placement, or saliency-based placement. In Section 4 we develop a learning-based approach that uses trained classifiers to adaptively place seeds. As shown in Section 6, this approach outperforms all other approaches. For example, it hits 50% of objects in the VOC2012 dataset with just 4 seeds per image. With 20 seeds per image the approach discovers 80% of all objects, many of which are not much larger than a single superpixel. Figure 2b shows the output of the approach with a budget of 8 seeds.

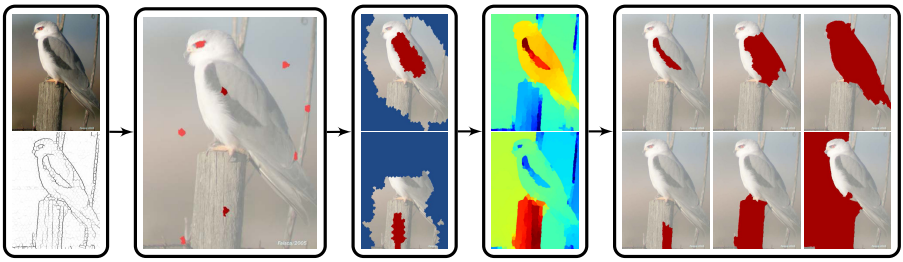
For each seed we generate foreground and background masks that will be used to compute the geodesic distance transform. As described in Section 3, a simple and effective approach is to use the seed itself as the foreground mask and the image boundary or the empty set as background. We can improve upon this by using a learning-based approach for computing the masks. This approach is developed in Section 4. Examples of such masks are shown in Figure 2c.

For each foreground-background mask we compute a signed geodesic distance transform (SGDT) over the image [2,6]. Each level set of the SGDT specifies an image region, but not all such regions form good proposals. As described in Section 3, we can extract a small set of high-quality object proposals by identifying certain critical level sets of the SGDT. Proposals formed by these critical level sets are shown in Figure 2e.

In the final step we sort all proposals produced for all seeds and masks to filter out near-duplicates. The overall pipeline yields state-of-the-art accuracy on standard datasets, as demonstrated in Section 6.

3 Proposal Generation

Preliminaries. Given an input image $\mathcal{I}$, we compute an oversegmentation into superpixels and a boundary probability map represented as a weighted graph



(a) Input (b) Seeds (c) Masks (d) SGDT (e) Proposals

Fig. 2. Overall proposal generation pipeline. (a) Input image with a computed superpixel segmentation and a boundary probability map. (b) Seeds placed by the presented approach. (c) Foreground and background masks generated by the presented approach for two of these seeds. (d) Signed geodesic distance transforms for these masks. (e) Object proposals, computed by identifying critical level sets in each SGDT.

$G_{\mathcal{I}} = (V_{\mathcal{I}}, E_{\mathcal{I}})$. This is done using existing techniques, as described in Section 5. Each node $x \in V_{\mathcal{I}}$ corresponds to a superpixel, each edge $(x, y) \in E_{\mathcal{I}}$ connects adjacent superpixels, and the edge weight $w(x, y)$ represents the likelihood of object boundary at the corresponding image edge.

The geodesic distance $d_{x,y}$ between two nodes $x, y \in V_{\mathcal{I}}$ is the length of the shortest path between the nodes in $G_{\mathcal{I}}$. The geodesic distance transform (GDT) measures the geodesic distance from a set of nodes $Y \subset V_{\mathcal{I}}$ to each node $x \in V_{\mathcal{I}}$:

$$D(x; Y) = \min_{y \in Y} d_{x,y}. \tag{1}$$

The GDT for all nodes in $V_{\mathcal{I}}$ can be computed exactly using Dijkstra's algorithm in total time $O(n \log n)$, where n is the number of superpixels. Linear-time approximations exist for regular grids [20,23], but our domain is not regular and we use the exact solution.

The geodesic distance transform can be generalized to consider a foreground set $F \subset V_{\mathcal{I}}$ and a background set $B \subset V_{\mathcal{I}}$ [2,6]. In this case, the signed geodesic distance transform (SGDT) is defined as

$$D(x; F, B) = D(x; F) - D(x; B). \tag{2}$$

Each level set λ of the SGDT encloses a unique image segment, which can be used as an object proposal:

$$\mathcal{P}_\lambda = \{x : D(x; F, B) < \lambda\}. \tag{3}$$

Our approach consists of computing promising foreground and background sets and identifying a small set of appropriate level sets λ for each foreground-background pair. The rest of this section describes the different stages of the approach. We begin by computing a set of foreground seeds: individual superpixels that are likely to be located inside objects. For each such seed, we construct foreground and background masks. For each pair of masks, we identify a small set of level sets. Each level set specifies an object proposal.

Seed Placement. Our first task is to identify a small set of seed nodes $S \subset V_{\mathcal{I}}$. The goal is to hit all the objects in the image with a small number of seeds, so as to minimize the overall number of object proposals that must be processed by the recognition pipeline. As shown in Section 6, naive seed selection strategies do not perform well. Both regular sampling and random sampling fail to discover small objects in images unless an exorbitant number of seeds is used. Saliency-based seed placement also performs poorly since it is not effective at identifying less prominent objects. We now describe a better seed selection heuristic, based on greedy minimization of geodesic distances.

The heuristic proceeds iteratively. The first seed is placed in the geodesic center of the image:

$$S \leftarrow \{\arg\min_{s} \max_{y \in V_{\mathcal{I}}} d_{s,y}\}. \tag{4}$$

The geodesic center is the superpixel for which the maximal geodesic distance to all other superpixels is minimized. It lies halfway on the longest geodesic path in the superpixel graph and can be found using three consecutive shortest path computations.

Each of the following seeds is placed so as to maximize its geodesic distance to previous seeds:

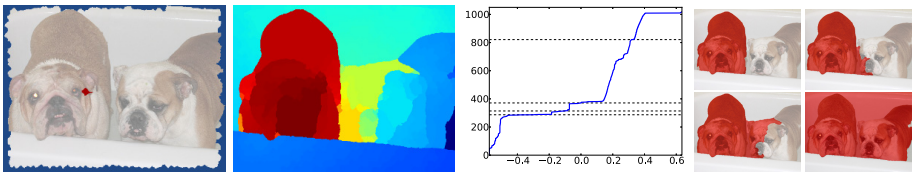
$$S \leftarrow S \cup \{\arg\max_{s} D(s; S)\}. \tag{5}$$

This is repeated until the desired number of seeds is reached. The $\arg\max$ in Equation 5 can be evaluated with one execution of Dijkstra's algorithm on $G_{\mathcal{I}}$, thus the total runtime of the algorithm is $O(N_S n \log n)$, where N_S is the number of seeds. The algorithm can be interpreted as greedy minimization of the maximal geodesic distance of all superpixels to the seed set.

This algorithm considerably outperforms the naive approaches. It will in turn be superseded in Section 4 by a learning-based approach, but it is a simple heuristic that performs well and may be sufficient for some applications.

Foreground and Background Masks. For each seed $s \in S$, we generate foreground and background masks $F_s, B_s \subset V_{\mathcal{I}}$ that are used as input to the SGDT. The goal here is to focus the SGDT on object boundaries by possibly expanding the foreground mask to include more of the interior of the object that contains it, as well as masking out parts of the image that are likely to be outside the object. This is a challenging task because at this stage we don't know what the object is: it may be as small as a single superpixel or so large as to span most of the image. We will tackle this problem systematically in Section 4, where a learning-based approach to generate foreground and background masks will be developed. As a baseline we will use the seed itself as the foreground mask. For the background we will use two masks: an empty one and the image boundary.

Critical Level Sets. Given a foreground-background mask, our goal is to compute a small set of intermediate level sets that delineate the boundaries of objects that include the foreground. Prior work on interactive geodesic segmentation considered a single segmentation specified by the zero level set of the SGDT [2,6,19].



(a) Image and masks (b) SGDT (c) Critical level sets (d) Proposals

Fig. 3. (a) An image with a foreground mask (red) and a background mask (blue). (b) The corresponding signed geodesic distance transform. (c) Critical level sets identified by our algorithm. (d) Corresponding object proposals.

However, the zero level set is sensitive to the detailed form of the masks and may not adhere to object boundaries [18]. We perform a more detailed analysis that yields a small number of level sets that capture object boundaries much better in the absence of interactive refinement by a human user.

Our analysis is based on the growth of the region $\mathcal{P}_\lambda$ as a function of λ. Specifically, let $A(\lambda) = |\mathcal{P}_\lambda|$ be the area enclosed by $\mathcal{P}_\lambda$. This function is illustrated in Figure 3c. Observe that when the λ level set reaches an object boundary, the evolution of the level set slows down. On the other hand, when the level set propagates through an object interior, it evolves rapidly. We can thus identify level sets that follow object boundaries by analyzing their evolution rate, given by the derivative $\frac{dA}{d\lambda}$. Specifically, to extract object proposals that adhere to object boundaries, we identify strong local minima of $\frac{dA}{d\lambda}$.

Selecting level sets purely by their evolution rate can lead to a lopsided selection, in which most proposals specify almost identical regions. To ensure diversity in the level set selection, we enforce the additional constraint that no two selected proposals can overlap by a factor of more than α. Overlap is defined as the Jaccard coefficient of two regions: $\mathcal{J}(\mathcal{P}_{\lambda_i}, \mathcal{P}_{\lambda_j}) = \frac{|\mathcal{P}_{\lambda_i}|}{|\mathcal{P}_{\lambda_j}|}$ for $\lambda_i < \lambda_j$. (Note that $\lambda_i < \lambda_j$ implies $\mathcal{P}_{\lambda_i} \subseteq \mathcal{P}_{\lambda_j}$.) We greedily select the critical level sets by iteratively choosing non-overlapping proposals with the lowest evolution rate. We stop when the desired number of proposals is reached or when no more non-overlapping level sets remain.

Once all proposals from all seeds are generated, we sort them by their evolution rate, which serves as a proxy for their quality. We then greedily select proposals that overlap with prior selections by at most α. To efficiently check the overlap between two proposals we use a hierarchical spatial data structure.

4 Learning Seed Placement and Mask Construction

The proposal generation pipeline described in Section 3 performs very well, as shown in Section 6. However, we can enhance its performance further by replacing two heuristic steps in the pipeline with learning-based approaches. These two steps are the seed placement algorithm and the construction of foreground and background masks.

Learning to Place Seeds. We now develop a learning-based approach for seed placement. The approach places seeds sequentially. We train a linear ranking classifier for the placement of each seed s_i, for $i = 1, \ldots, N_S$. This allows the placement strategy to adapt: the objective that is optimized by the placement of the first seeds need not be the same as the objective optimized by the placement of later seeds. For example, early seeds can prioritize hitting large and prominent objects in the image, while later seeds can optimize for discovering a variety of smaller objects that may require specialized objectives.

At each iteration i, we compute features $\mathbf{f}_x^{(i)}$ for each possible seed location $x \in V_{\mathcal{I}}$. These features include static features such as location within the image and adaptive features such as distance to previously placed seeds. In general, the feature values are a function of previously placed seeds: $\mathbf{f}_x^{(i)} \neq \mathbf{f}_x^{(j)}$ for $i \neq j$. The specific features we use are listed in Section 5.

The classifier for iteration i is trained after classifiers for iterations $j < i$. For iteration i, we train a linear ranking classifier that associates a score $\mathbf{w}_i^\top \mathbf{f}$ with any feature vector $\mathbf{f}$. During inference we place seed s_i in the top ranking location as determined by the trained classifier: $s_i = \arg\max_x \mathbf{w}_i^\top \mathbf{f}_x^{(i)}$. The training optimizes the weight vector $\mathbf{w}_i$. For the training, we partition each training image $\mathcal{I}$ into a positive region $P_{\mathcal{I}}$ and a negative region $N_{\mathcal{I}}$. The positive region consists of all superpixels contained in ground truth objects in the image that have not been hit by previously placed seeds. (The seeds are placed by classifiers previously trained for iterations $j < i$.) The negative region is simply the complement of the positive region: $N_{\mathcal{I}} = V_{\mathcal{I}} \setminus P_{\mathcal{I}}$. We will now formulate a learning objective that encourages the placement of seed s_i inside the positive region $P_{\mathcal{I}}$ in as many images $\mathcal{I}$ as possible.

Our learning objective differs substantially from standard ranking methods [13]. Standard algorithms aim to learn a ranking that fits a given complete or partial ordering on the data. In our setting, such a partial ordering can be obtained by ranking feature vectors associated with each positive region ($\mathbf{f}_x^{(i)}$ for $x \in P_{\mathcal{I}}$) above feature vectors associated with the corresponding negative region $N_{\mathcal{I}}$. While this standard objective works well for early seeds, it ceases to be effective in later iterations when no parameter setting $\mathbf{w}_i$ can reasonably separate the positive region from the negative.

Our key insight is that we do not need to rank all positive seed locations above all negative ones. Our setting only demands that the highest-ranking location be in the positive set, since we only place one seed s_i at iteration i. This objective can be formalized as finding a weight vector $\mathbf{w}_i$ that ranks the highest-ranking positive seed $\hat{x} \in P_{\mathcal{I}}$ above the highest-ranking negative seed $\hat{y} \in N_{\mathcal{I}}$. We use logistic regression on the difference between the two scores: $\mathbf{w}_i^\top \mathbf{f}_{\hat{x}}^{(i)} - \mathbf{w}_i^\top \mathbf{f}_{\hat{y}}^{(i)}$. The log-likelihood of the logistic regression is given by

$$\ell_{\mathcal{I}}(\mathbf{w}_i) = \log\left(1 + \exp\left(\max_{x \in N_{\mathcal{I}}} \mathbf{w}_i^\top \mathbf{f}_x^{(i)} - \max_{x \in P_{\mathcal{I}}} \mathbf{w}_i^\top \mathbf{f}_x^{(i)}\right)\right). \tag{6}$$

This objective is both non-convex and non-smooth, which makes it impossible to compute gradients or subgradients. However, we can replace each maximum

$\max_x \mathbf{w}_i^\top \mathbf{f}_x^{(i)}$ in Equation 6 with the softmax $\log \sum_x \exp(\mathbf{w}_i^\top \mathbf{f}_x^{(i)})$, which can be used to simplify the objective to

$$\ell_{\mathcal{I}}(\mathbf{w}_i) = \log \sum_{x \in V_{\mathcal{I}}} \exp\left(\mathbf{w}_i^\top \mathbf{f}_x^{(i)}\right) - \log \sum_{x \in P_{\mathcal{I}}} \exp\left(\mathbf{w}_i^\top \mathbf{f}_x^{(i)}\right). \tag{7}$$

This objective is smooth and any gradient-based optimization algorithm such as L-BFGS can be used to minimize it. While the second term in the objective is still non-convex, the optimization is very robust in practice. In our experiments, a wide variety of different initializations yield the same local minimum.

Learning to Construct Masks. Given a seed $s \in S$, we generate foreground and background masks $F_s, B_s \subset V_{\mathcal{I}}$. These masks give us a chance to further direct the geodesic segmentation to object boundaries by labeling some image regions as foreground or background. Given the formulation of the SGDT, these masks must be conservative: the foreground mask must be contained inside the sought object and the background mask must be outside.

To construct masks, we train one linear classifier for the foreground mask and one linear classifier for the background mask. Both classifiers operate on features $\mathbf{f}_x^{(s)}$, where s is the given seed and $x \in V_{\mathcal{I}}$ is a superpixel in the image. The training optimizes a weight vector $\mathbf{w}_F$ for the foreground classifier and a weight vector $\mathbf{w}_B$ for the background classifier.

We begin by considering the learning objective for the foreground classifier. This objective should reward the generation of the largest foreground mask $F_s \subseteq O_s$, where O_s is the ground-truth object that encloses seed s. The containment in O_s is a hard constraint: the foreground mask should not leak outside the object boundary. This can be formalized as follows:

$$\begin{aligned} \underset{\mathbf{w}_F}{\text{minimize}} \quad & \sum_s \sum_{x \in O_s} \rho\left(\mathbf{w}_F^\top \mathbf{f}_x^{(s)}\right) \\ \text{subject to} \quad & \forall s \in S \;\; \forall y \notin O_s \;\; \mathbf{w}_F^\top \mathbf{f}_y^{(s)} < 0. \end{aligned} \tag{8}$$

Here ρ is a penalty function that maximizes the number of true positives. We use the hinge loss, which allows us to minimize Equation 8 as a standard linear SVM with a high negative class weight.

The hard constraints in Equation 8 need to be satisfied for a large number of training objects O_s with hugely varying appearance and size. In our initial experiments, simply optimizing this objective led to trivial classifiers that simply produce the initial seed as the foreground mask and the empty set for the background mask. (The learning objective for the background mask is analogous to Equation 8.) To overcome this difficulty, we modify the formulation to train several classifiers. At inference time, we simply use each of the trained classifiers to generate object proposals. The basic idea is that one of the learned classifiers absorbs the challenging training examples that demand a highly conservative response (trivial foreground and background masks), while others can handle examples that allow larger masks.

(a) Image (b) Mask 1 (c) Mask 2 (d) Mask 3

Fig. 4. The output of learned mask classifiers. (a) Input image. (b-d) Foreground and background masks generated for a given seed by the learned classifiers. The first classifier is maximally conservative, the others are more risk-taking.

Specifically, we train K foreground classifiers, with weight vectors $\mathbf{w}_F^{(k)}$ for $k = 1, \ldots, K$. (We use $K = 3$.) In addition to the weight vectors, we also optimize a label k_s for each seed s. This is a latent variable $k_s \in \{1, \ldots, K\}$ that associates each training seed s with one of the classifiers. The classifiers and the associations are optimized in concert using the following objective:

$$\begin{aligned} \underset{\mathbf{w}_F^{(k)}, k_s}{\text{minimize}} \quad & \sum_s \sum_{x \in O_s} \rho\left(\mathbf{w}_F^{(k_s)} \cdot \mathbf{f}_x^{(s)}\right) \\ \text{subject to} \quad & \forall s \in S \;\; \forall y \notin O_s \;\; \mathbf{w}_F^{(k_s)} \cdot \mathbf{f}_y^{(s)} < 0. \end{aligned} \tag{9}$$

We use alternating optimization. The different classifiers $\mathbf{w}_F^{(k)}$ are initialized by picking K random seeds and optimizing the objective in Equation 8 for each of these seeds separately. We next optimize the associations k_s by evaluating each classifier on each seed s and associating each seed with the classifier that yields the lowest objective value on that seed. We then alternate between optimizing the classifier parameters given fixed associations and optimizing the associations given fixed classifiers. Note that each step decreases the compound objective in Equation 9.

The extension of the objective and the algorithm to incorporate background mask classifiers is straightforward. In the complete formulation, we train K pairs $(\mathbf{w}_F^{(k)}, \mathbf{w}_B^{(k)})$ of foreground and background classifiers. For each seed, the label k_s associates it with both the foreground classifier $\mathbf{w}_F^{(k_s)}$ and the background classifier $\mathbf{w}_B^{(k_s)}$.

Figure 4 demonstrates the output of the learned mask classifiers on an example test seed. As expected, one of the classifiers is conservative, using the input seed as the foreground and the empty set as the background. The other classifiers are more risk-taking. At test time we use all K masks for each seed to generate object proposals.

5 Implementation

We compute a boundary probability image using structured forests [8]. This boundary probability image is used to produce a superpixel segmentation. We use the geodesic k-means algorithm, which produces a regular oversegmentation

that adheres to strong boundaries [17]. Both algorithms are extremely efficient, with a combined runtime of 0.5 seconds for images of size 350×500.

Seed features used by the classifiers described in Section 4 include image coordinates x and y, normalized to the interval $[-1, 1]$, as well as absolute and squared normalized coordinates. We further use the minimal color and spatial distance to previously placed seeds, as well as the color covariance between the given superpixel pixels and all seed pixels. We also add geodesic distances to previously placed seeds, as well as to the image boundary. For computing these distances, we use both graphs with constant edge weights and with boundary probability weights.

Mask features used by the classifiers described in Section 4 include location relative to the seed, distance to each of the image boundary edges, and color similarity to the seed in both RGB and Lab color space. We also compute color histograms for each superpixel and use the χ^2 distance between the color histogram of the given superpixel and the seed superpixel. Finally we add an indicator feature for the seed itself, which ensures that there always exists a parameter setting satisfying Equation 9.

6 Evaluation

We evaluate the presented approach on the PASCAL VOC2012 dataset [10]. All segmentation experiments are performed on the 1449 validation images of the VOC2012 segmentation dataset. Bounding box experiments are performed on the larger detection dataset with 5823 annotated validation images. We train all classifiers on the 1464 segmentation training images. Training all seed and mask classifiers takes roughly 10 minutes in total. All experiments were performed on a 3.4 GHz Core i7 processor. Runtimes for all methods are reported for single-threaded execution and cover all operations, including boundary detection and oversegmentation.

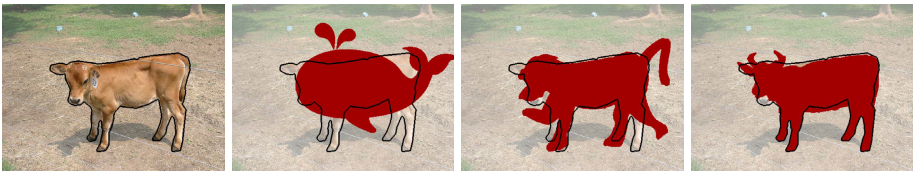
To evaluate the quality of our object proposals we use the Average Best Overlap (ABO), covering, and recall measures [5]. The ABO between a ground truth object set S and a set of proposals $\mathcal{P}$ is computed using the overlap between each ground truth region $R \in S$ and the closest object proposal $R' \in \mathcal{P}$:

$$\text{ABO} = \frac{1}{|S|} \sum_{R \in S} \max_{R' \in \mathcal{P}} \mathcal{J}(R, R').$$

Here the overlap of two image regions R and R' is defined as their Jaccard coefficient $\mathcal{J}(R, R') = \frac{|R \cap R'|}{|R \cup R'|}$. Figure 5 illustrates the relationship between the precision of fit of the two image regions and the corresponding Jaccard coefficient values.

Covering is an area-weighted measure:

$$\text{Covering} = \frac{1}{\sum_{R \in S} |R|} \sum_{R \in S} |R| \max_{R' \in \mathcal{P}} \mathcal{J}(R, R').$$



(a) Ground truth (b) $\mathcal{J} = 0.554$ (c) $\mathcal{J} = 0.703$ (d) $\mathcal{J} = 0.910$

Fig. 5. The relationship between region similarity and the Jaccard coefficient $\mathcal{J}$. A Jaccard coefficient of 0.5 admits rather significant departures from the ground truth shape. A Jaccard coefficient of 0.7 is more discriminative and a coefficient of 0.9 demands a very tight fit.

It discounts small and thin objects and assigns higher importance to larger objects.

The recall measure is defined as the fraction of ground truth segments with a maximum overlap larger than α [5,21]. It is also referred to as the detection rate [16]. A fairly lenient $\alpha = 50\%$ recall threshold has sometimes been used [21]. However, this threshold allows poorly fitting proposals to qualify, as shown in Figure 5b. A high recall at 50% can be achieved by covering the image evenly with generic proposals, rather than producing detailed object shapes. Our work focuses on generating object proposals with informative spatial support. In the best case, our pipeline can precisely delineate objects in the image, as shown in Figure 1. To evaluate the precision of object proposals produced by different approaches more stringently, we also report results for the tighter $\alpha = 70\%$ recall threshold.

Seed Placement. We first compare the geodesic seed placement heuristic described in Section 3, the learning-based seed placement approach described in Section 4, and four alternative seed placement strategies: regular sampling, random sampling, saliency-weighted random sampling, and sampling based on an oversegmentation of the image. The oversegmentation-based seed placement is modeled on the approach of Carreira et al. [5] and uses a hierarchical segmentation algorithm. For saliency-based seed placement we randomly sample superpixels weighted by their saliency as given by the algorithm of Perazzi et al. [17]. For each seed placement strategy we generate a single-seed foreground mask and use the image boundary as background.

Both saliency-based and regular seeds are able to discover a reasonable number of objects with up to 3 seeds, as shown in Figure 6a. However, both methods make less progress after the first few seeds. The saliency-based method biases the placement to prominent objects, missing less salient ones. Regular and random sampling both miss many smaller objects. Oversegmentation-based seeds generally perform better, but not as well as our geodesic or learned seeds.

Figures 6b shows the ABO of our pipeline for a fixed parameter setting and an increasing number of seeds. Random, saliency-weighted, and regular sampling perform equally well and about 5% and 7% worse than geodesic seed placement in ABO and recall respectively. Segmentation-based seeds perform better, but

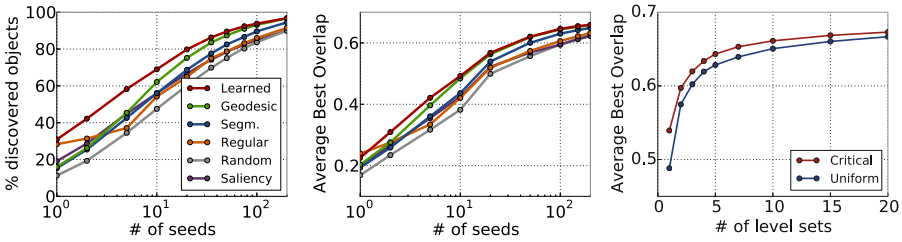


Fig. 6. (a,b) Comparison of our seed placement algorithms (heuristic and learned) to other seed placement algorithms: (a) percentage of objects discovered by placed seeds, (b) accuracy achieved by the proposal pipeline given seeds placed by different algorithms. (c) Comparison of our level set selection algorithm to uniform selection: the figure shows the accuracy achieved by the pipeline using each of these level set selection algorithms.

still 1-2% worse than geodesic seeds in both metrics. With a high seed budget, the geodesic and learned strategies perform similarly, however the learned strategy usually produces 5% fewer proposals, as seeds sometimes collide and produce duplicate proposals that are then filtered out.

Level Set Selection. Next, we compare the critical level set selection algorithm developed in Section 3 to simple uniform selection. For this experiment we use 100 geodesic seeds, with single-seed foreground masks and the image boundary as background. As shown in Figure 6c, our level set selection algorithm outperforms uniform selection, especially with a low budget of proposals per seed. For a single level set, our algorithm achieves a 5% higher ABO and 4% higher recall than the zero level set. Our algorithm consistently outperforms uniform selection. For 20 level sets our algorithm is within 0.5% of the maximal achievable ABO obtained with an oracle level set selector that uses the ground truth, while uniform selection requires twice as many level sets to achieve this level of accuracy.

Boundary Detection. In Figure 7, we evaluate the effect of the boundary detection procedure on the final proposal quality. We compare Sobel filtering, sketch tokens [14], and structured forests (single-scale and multi-scale) [8]. Sobel filtering yields poor accuracy since it produces a fairly inaccurate boundary map. Our pipeline performs well with all other boundary detectors. Multi-scale structured forests yield the best results and we use this procedure for all other experiments.

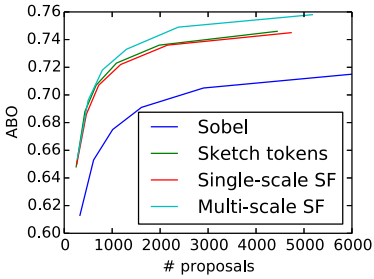


Fig. 7. Effect of boundary detection procedure

Object Proposals. We now use the VOC2012 segmentation dataset to evaluate the accuracy of object proposals produced by the baseline pipeline described in Section 3 and the enhanced pipeline that uses the seed placement and mask

Table 1. Accuracy and running time for three state-of-the-art object proposal methods compared to accuracy and running time for our approach. Results are provided for our baseline pipeline (Baseline GOP) and the enhanced pipeline that uses seed placement and mask construction classifiers (Learned GOP). Different budgets (N_S,N_A) for seed placement and level set selection control the number of generated proposals (# prop).

Method	# prop.	ABO	Covering	50%-recall	70%-recall	Time
CPMC [5]	646	0.703	0.850	0.784	0.609	252s
Cat-Ind OP [9]	1536	0.718	0.840	0.820	0.624	119s
Selective Search [21]	4374	0.735	0.786	0.891	0.597	2.6s
Baseline GOP (130,5)	653	0.712	0.812	0.833	0.622	0.6s
Baseline GOP (150,7)	1090	0.727	0.828	0.847	0.644	0.65s
Baseline GOP (200,10)	2089	0.744	0.843	0.867	0.673	0.9s
Baseline GOP (300,15)	3958	0.756	0.849	0.881	0.699	1.2s
Learned GOP (140,4)	652	0.720	0.815	0.844	0.632	1.0s
Learned GOP (160,6)	1199	0.741	0.835	0.865	0.673	1.1s
Learned GOP (180,9)	2286	0.756	0.852	0.877	0.699	1.4s
Learned GOP (200,15)	4186	0.766	0.858	0.889	0.715	1.7s

construction classifiers described in Section 4. Table 1 compares the accuracy of our pipeline (GOP) to three state-of-the-art object proposal methods, each of which produces a different number of segments. We set the number of seeds N_S and number of level sets N_A in our pipeline to different values to roughly match the number of proposals produced by the other approaches. Accuracy is evaluated using ABO, covering, and recall at $\mathcal{J} \geq 50\%$ and $\mathcal{J} \geq 70\%$.

Our baseline performs slightly better than CPMC [5] in ABO and 70%-recall and greatly outperforms it at 50%-recall. CPMC is better at proposing larger objects, which leads to higher covering results. Figure 8 provides a more detailed comparison. CPMC is based on graph cuts and is less sensitive to texture variations within large objects. However, CPMC is more than two orders of magnitude slower than GOP, making it impractical for larger datasets. Evaluating CPMC on 1464 images took two full days on an 8-core processor, while GOP processed the dataset in less than two minutes on the same machine. (0.6 seconds per image on a single core.)

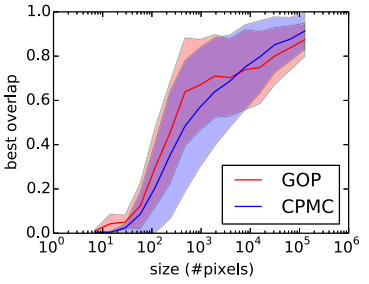


Fig. 8. Accuracy of CMPC and GOP as a function of segment size

Baseline GOP outperforms category-independent object proposals [9] using just two-thirds of the number of proposals. Again our approach is two orders of magnitude faster.

Selective search [21] performs extremely well at 50%-recall. However, when the recall threshold is increased to 70% our approach significantly outperforms selective search. At this threshold, Baseline GOP with 660 proposals outperforms the recall achieved by selective search with more than 4000 proposals. When the

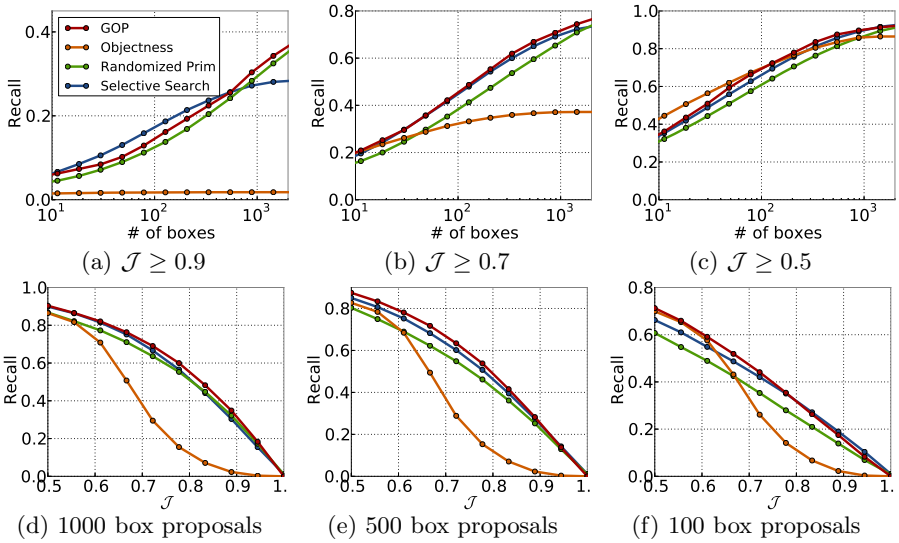


Fig. 9. Recall for bounding box proposals. (a-c) Recall at above a fixed threshold rate $\mathcal{J}$ for a varying number of generated proposals. (d-f) Recall at different thresholds for fixed proposal budgets.

proposal budget for GOP is increased to match the number of proposals produced by selective search, our 70%-recall is 10% higher.

The seed placement and mask construction classifiers yield a noticeable increase in proposal accuracy, as reflected in the ABO and 70%-recall measures. The classifiers increase the ABO by about 1% and the 70%-recall by up to 3%. The additional computational cost of evaluating the classifiers increases the running time by about half a second and is primarily due to the feature computation.

Bounding Box Proposals. We also evaluate the utility of the presented approach for generating bounding box proposals. We produce bounding box proposals simply by taking the bounding boxes of object proposals produced by GOP. In this mode, using mask construction classifiers does not confer an advantage over simple foreground-background masks since segmentation accuracy is less important. We thus use baseline foreground-background masks for this experiment. The seed placement classifiers still reduce the number of generated proposals by 5% and yield higher accuracy, especially for a small number of seeds.

To evaluate the accuracy of bounding box proposals we use the VOC2012 detection dataset and follow the evaluation methodology of Manén et al. [16]. The results are shown in Figure 9. Our approach is compared to three state-of-the-art methods: objectness [1], selective search [21], and the Randomized Prim algorithm [16]. We measure recall for different Jaccard coefficient thresholds and for different proposal budgets N. For objectness and selective search we select the N highest ranking proposals produced by these methods. For Randomized Prim and GOP we generate N proposals by varying the algorithms' parameters.

Table 2. Evaluation of bounding box proposals using the VUS measure

Method	VUS 10000 windows		VUS 2000 windows		Time
	Linear	Log	Linear	Log	
Objectness [1]	0.332	0.244	0.323	0.225	2.2s
Randomized Prim [16]	0.603	0.334	0.511	0.274	1.1s
Selective search [21]	0.573	0.350[1]	0.528	0.301	2.6s
GOP	0.624	0.363	0.546	0.310	0.9s

We further compute the volume under surface (VUS) measure as proposed by Manén et al. [16]. This measures the average recall by linearly varying the Jaccard coefficient threshold $\mathcal{J} \in [0.5, 1]$ and varying the number of proposals N on either linear or log scale. The results are shown in Table 2. Manén et al. [16] vary the proposal budget N from 0 to $10,000$. This unfairly favors our method and the Randomized Prim algorithm since the other approaches produce a lower average number of proposals. We therefore additionally compute a VUS for $2,000$ windows, for which each algorithm produces approximately the same number of proposals.

Objectness performs best at 50%-recall and a low proposal budget, since it is able to rank proposals very well. However, its performance degrades quickly when the recall threshold is increased.

Both selective search and GOP consistently outperform Randomized Prim. Selective search has the edge at high recall with a low proposal budget, while our approach performs better in all other regimes. This is also reflected in the results for the VUS measure (Table 2). GOP outperforms all other approaches in both linear and logarithmic VUS measure, for both 2000 and 10000 windows. The running time of our approach is again the lowest.

7 Discussion

We presented a computationally efficient approach for identifying candidate objects in an image. The presented approach outperforms the state of the art in both object shape accuracy and bounding box accuracy, while having the lowest running time. In the future it would be interesting to also learn the metric on which the geodesic distance transform is computed. In addition, joint learning of all parameters for all steps in the pipeline could exploit correlations between the different learned concepts and further increase the accuracy of the approach.

Acknowledgements. Philipp Krähenbühl was supported by the Stanford Graduate Fellowship.

[1] Note that our results for selective search differ significantly from the results reported by Manén et al. [16]. We use the highest-ranking bounding boxes in the evaluation instead of randomly subsampling them.

References

1. Alexe, B., Deselaers, T., Ferrari, V.: Measuring the objectness of image windows. PAMI 34(11) (2012)
2. Bai, X., Sapiro, G.: Geodesic matting: A framework for fast interactive image and video segmentation and matting. IJCV 82(2) (2009)
3. Carreira, J., Caseiro, R., Batista, J., Sminchisescu, C.: Semantic segmentation with second-order pooling. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VII. LNCS, vol. 7578, pp. 430–443. Springer, Heidelberg (2012)
4. Carreira, J., Li, F., Sminchisescu, C.: Object recognition by sequential figure-ground ranking. IJCV 98(3) (2012)
5. Carreira, J., Sminchisescu, C.: CPMC: Automatic object segmentation using constrained parametric min-cuts. PAMI 34(7) (2012)
6. Criminisi, A., Sharp, T., Rother, C., Pérez, P.: Geodesic image and video editing. ACM Trans. Graph. 29(5) (2010)
7. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
8. Dollár, P., Zitnick, C.L.: Structured forests for fast edge detection. In: ICCV (2013)
9. Endres, I., Hoiem, D.: Category-independent object proposals with diverse ranking. PAMI 36(2) (2014)
10. Everingham, M., Van Gool, L.J., Williams, C.K.I., Winn, J.M., Zisserman, A.: The Pascal Visual Object Classes (VOC) challenge. IJCV 88(2) (2010)
11. Felzenszwalb, P.F., Girshick, R.B., McAllester, D.A., Ramanan, D.: Object detection with discriminatively trained part-based models. PAMI 32(9) (2010)
12. Gu, C., Lim, J.J., Arbelaez, P., Malik, J.: Recognition using regions. In: CVPR (2009)
13. Joachims, T.: Optimizing search engines using clickthrough data. In: KDD (2002)
14. Lim, J.J., Zitnick, C.L., Dollár, P.: Sketch tokens: A learned mid-level representation for contour and object detection. In: CVPR (2013)
15. Malisiewicz, T., Efros, A.A.: Improving spatial support for objects via multiple segmentations. In: BMVC (2007)
16. Manén, S., Guillaumin, M., Gool, L.V.: Prime object proposals with randomized Prim's algorithm. In: ICCV (2013)
17. Perazzi, F., Krähenbühl, P., Pritch, Y., Hornung, A.: Saliency filters: Contrast based filtering for salient region detection. In: CVPR (2012)
18. Price, B.L., Morse, B.S., Cohen, S.: Geodesic graph cut for interactive image segmentation. In: CVPR (2010)
19. Sinop, A.K., Grady, L.: A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In: ICCV (2007)
20. Toivanen, P.J.: New geodesic distance transforms for gray-scale images. Pattern Recognition Letters 17(5) (1996)
21. Uijlings, J.R.R., van de Sande, K.E.A., Gevers, T., Smeulders, A.W.M.: Selective search for object recognition. IJCV 104(2) (2013)
22. Viola, P.A., Jones, M.J.: Rapid object detection using a boosted cascade of simple features. In: CVPR (2001)
23. Yatziv, L., Bartesaghi, A., Sapiro, G.: O(n) implementation of the fast marching algorithm. J. Comput. Physics 212(2) (2006)