Constrained Parametric Min-Cuts for Automatic Object Segmentation

Joao Carreira and Cristian Sminchisescu

Abstract—We present a novel framework for creating and ranking plausible object hypotheses in an image using bottom-up generation processes and mid-level selection cues. The object hypotheses are represented as figure-ground segmentations, and are extracted automatically, without prior knowledge about the properties of individual object classes, by solving a sequence of constrained parametric min-cut problems (CPMC) on a regular image grid. In a subsequent step, we learn to rank the corresponding segments by training a continuous model to predict their plausibility (putative overlap with ground truth) based on their mid-level region properties, then diversify the estimated overlap using maximum marginal relevance measures. We show that this algorithm significantly outperforms the state of the art for low-level segmentation in the VOC 2009 and 2010 datasets. It achieves the same average best segmentation covering on VOC2009 as the best performing technique to date [1], 0.61 when using just the top 7 ranked segments, instead of the full hierarchy in [1]. Our method achieves 0.78 average best covering using 154 segments. An extended version of the basic algorithm achieves 83% average per class object recall, using 200 segments per image on the VOC2010 segmentation dataset. In a related paper [2], it was shown that the algorithm achieves notable results when used in a segmentation-based recognition pipeline. This pipeline achieved the first place in the VOC2009 and VOC2010 image segmentation and labeling challenges.

Index Terms—Image Segmentation, figure-ground segmentation, learning

1 INTRODUCTION

Reliably identifying the spatial extent of objects in images is important for high-level vision tasks like object recognition. A region that covers an object fully provides a characteristic spatial scale for feature extraction, isolates the object from the potentially confusing background signal and allows for information to be propagated from parts of the object to the whole (a region covering a human fully makes it possible to propagate the person identity from the easier to identify face area to the rest of the body).

Given an image, the space of all its possible regions, or segments, is exponentially large. However, in our perceived visual world not all image regions are equally likely to arise from the projection of a three-dimensional object. Objects are usually compact and this results in their projection in the image being connected; it is also common for strong contrast edges to mark objects boundaries. Such properties reduce the number of plausible object regions greatly, but may not be sufficient to unambiguously define unique optimal regions for each object in an image.

In this paper, we follow a two step strategy by combining a figure-ground, bottom-up approach to segmentation with subsequent verification and ranking based on mid-level region properties. Key to an effective solution is the capability to leverage the statistics of real-world objects in the selection process. One possibility would be to learn the parameters of the segmentation algorithm directly, by training a machine learning model using large amounts of human annotated data. However, the local scope of dependencies and the intrinsically combinatorial nature of image segmentation diminishes the effectiveness of learning in such 'pixel spaces' as many interesting features such as the convexity and the smoothness of a region boundary are difficult to capture locally. On the other hand, once sufficient image support is available, learning to distinguish 'good' segments that represent plausible projections of real-world surfaces, from accidental image partitions becomes in principle feasible. This motivates our novel decomposition of the problem into two stages. In the first stage, we explore the space of regions that can inferred from local measurements, using cues such as good alignment with image edges. The process of enumerating regions with plausible alignment with the image contours is performed using exact combinatorial methods based on parametric max-flow. Then, in the restricted space of generated regions, we use a learned combination of sophisticated mid-level features to induce a more accurate global ranking of those regions in terms of their probability of being 'object-like'.

A key question, and one of our contributions, is how should image partitions be generated. Should region hypotheses be allowed to overlap with each other? Should one aim at multi-region image segmentations early? We argue that segmentation is already a sufficiently challenging problem without such constraints and global inter-region spatial consistency should be, perhaps, enforced at a later stage of processing, by higher-level routines that have better spatial scope for this calculation. We argue that attempts to enforce complex multi-region consistency constraints early may disallow the speculative behavior necessary for sampling regions effectively, given the inherently ambiguous nature of the low-level cues one typically operates on initially. Hence, differently from most of the existing approaches to segmentation, we derive methods to generate *several independent figure-ground partitions*, rather than a battery of splits of each image into multiple, non-

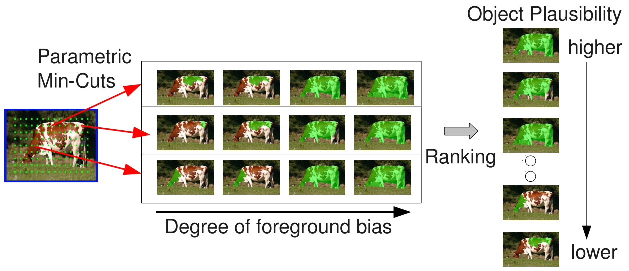

Fig. 1: Our object segmentation framework. Segments are extracted around regularly placed foreground seeds, with various background seeds corresponding to image boundary edges, for all levels of foreground bias, which has the effect of producing segments at different locations and spatial scales. The resulting set of segments is ranked according to their plausibility of being good object hypotheses, based on mid-level properties. Ranking involves first removing duplicates, then diversifying the segment overlap scores using maximum marginal relevance measures.

overlapping regions[1].

The overall framework we pursue is depicted in fig. 1. We first solve a large number of independent binary min-cut problems on an image grid, at multiple scales. These are designed as energy functions efficiently solvable with parametric min-cut/max-flow techniques. The resulting pool of segments is minimally filtered to remove trivial solutions and ranked using a regressor trained to predict to what extent the segments exhibit the regularities typical of real-world objects, based on their low and mid-level region properties. Since ranking tends to place similar inputs in similar ranks, we diversify the resulting segment ranking using Maximal Marginal Relevance measures, with the top ranked segments retained.

The quality of the list of object hypotheses returned by our algorithm is evaluated empirically by measuring how accurate they are with respect to pixel-level ground truth human annotations, in object recognition datasets. We also record performance as a function of the number of segments. Results are reported on several publicly available benchmarks: MSRC [4], the Weizmann Segmentation Database [5] and both VOC2009 and VOC2010 [6], [7] where the proposed method is shown to significantly outperform the state of the art, while at the same time using significantly fewer segments.

Several types of methods may benefit from outputs like the ones provided by our algorithm. Object detectors usually scan a large number of bounding boxes in sliding window schemes [8], [9] without considering the plausibility of pixel grouping within each. Semantic segmentation algorithms [10], [11], [12], [13] incorporate the outputs of these object detectors, and may need to mediate the transition between the rectangular regions produced by the detector and the desired free-form regions aligned with object boundaries. Unsupervised object discovery [14] also requires good class-independent object proposals. While the presentation focuses on the case of object segmentation, the proposed method is general and can rank lists of segments that follow the statistics of non-object, ‘stuff’ regions such as grass or sky, as long as appropriate ground truth training data statistics are provided.

An implementation of the proposed algorithm is made publicly available via our website [15].

Paper Organization: Section §2 reviews the related literature, §3 introduces the CPMC algorithm used to generate an initial pool of segments for an image and §4 presents the segment ranking procedure. Section §5 gives experimental results and shows comparisons with the state of the art. An extension of the basic algorithm to include bounding box constraints, and the corresponding results are described in §6. We conclude and discuss ideas for future work in §7.

2 Related Work

The first image segmentation approach, published more than 40 years ago by Muerle and Allen [16], was aimed to compute ‘object’ regions. Small patches having similar gray-level statistics were iteratively merged, starting at a seed patch. Region growing stopped when none of the neighboring candidate patches was sufficiently similar to the current region. The process was repeated until all pixels were assigned.

This method took advantage of the fundamental grouping heuristic that neighboring pixels with different color are more likely to belong to different objects. However it retrieved very local solutions and was not able to deal with textured regions, and even less, take advantage of more sophisticated object statistics. Later, more accurate techniques emerged—good surveys can be found in [17], [18], [19]. However, most methods still pursued a single optimal segmentation of an image into a set of non-overlapping regions that cover it fully (an image partitioning). But a sufficiently good partitioning is not easy to find given the ambiguity of low and mid level cues. There were also no quantitative benchmarks to gauge progress and most papers solely described qualitatively the merits of the output segmentations, and only on a reduced number of images.

As a result, in the nineties, part of the recognition community lost hope that a reliable segmentation procedure would be found and began investigating solutions that avoided bottom-up segmentation altogether [20]. This trend led to the current prevalence of bounding box detectors operating on sliding windows [8], [21]. These detectors rely on dense evaluation of classifiers over overlapping rectangular image regions, with consistency being usually enforced a posteriori by non-maxima suppression. This may have initially suggested that the original partitioning requirements assumed by most segmentation algorithms can be bypassed. Sliding window methods are indeed powerful for object localization for certain objects like faces or motorbikes, but do not obviously generalize to more complex objects and cannot be easily adapted for general 3d scene understanding: e.g. information predicted on rectangular image regions is not sufficient for operations such

[1] The algorithm proposed in this paper has been recently employed to generate multi-region segmentations by aggregating high-scoring sets of non-overlapping figure-ground segmentations, modeled as maximal cliques, with competitive results [3].

as manipulation of a cup by a robot, where precisely identify the cup handle in the image in order to grab it, is critical.

Such considerations made a revival of segmentation inevitable. The trend has gained momentum during the past ten years, propelled by the creation of annotated benchmarks [6], [22], together with new segmentation performance metrics [6], [23]. A second important factor was the adoption of machine learning techniques to optimize performance on benchmarks. A third factor was the relaxation of single partitioning requirements. A popular approach emerged by computing several fully independent segmentations, possibly using different algorithms. This idea was pursued by Hoiem *et al*. [24] for geometric labeling problems. Russel *et al*. computed normalized cuts for different number of segments and image sizes [14] for unsupervised object discovery problems. By generating tens to hundreds of thousands of segments per image, Malisiewicz and Efros [25] produce very good quality segments on the MSRC dataset. The segments were obtained by merging pairs and triplets of segments obtained using the Mean Shift [26], Normalized Cuts [27] and Felzenszwalb-Huttenlocher's (FH) [28] algorithms. Stein *et al*. [29] solved Normalized Cut problems for different number of segments, on a special affinity matrix derived from soft binary mattes, whereas Rabinovich *et al*. [30] shortlisted segmentations that tend to reoccur, hence are potentially more stable.

The computation of multiple segmentations can also be organized hierarchically. Shi and Malik [27] recursively solve relaxations of a Normalized Cut cost based on graphs constructed over pixel nodes. Sharon *et al*. [31] proposed algebraic multigrid techniques to efficiently solve normalized cuts problems at multiple levels of granularity, where graphs with increasingly more complex features are used at coarser levels. Arbeláez *et al*. [1] derive a segment hierarchy by iteratively merging superpixels produced by an oriented watershed transform. They use the output of the learned globalPb [32] boundary detector and can represent the full hierarchy elegantly in a single ultrametric contour map. The hierarchy is a natural representation for segmentation, as it lends itself to compositional representations. Inaccuracies in one level (due to the incorrect merging of two regions in the previous level), however, tend to propagate to all coarser levels. Therefore, given the same segmentation technique, generating a single hierarchy is likely to be less robust than generating independent segmentations.

Differently, our region sampling methodology generates multiple independent binary hierarchies constrained at different positions in the image. Each level of the hierarchy corresponds to a partitioning into figure and ground, where only the figure region is kept, and regions at finer levels are nested inside coarser levels regions (this is a result of our parametric max-flow methodology). In this way, we aim to better sample the space of plausible regions surrounding each pixel. We compute these partitionings using energies mostly related to the ones developed for interactive segmentation applications, where obtaining single figure-ground solutions is a common goal. In these applications, max-flow algorithms are quite popular because they can obtain exact optima for certain energy minimization problems involving region and boundary properties [33]. Generally the user assigns manually some pixels to foreground and background regions, these are encoded into an energy problem, which is solved using a global minimization algorithm. The two steps are repeated until the quality of the resulting binary segmentation is satisfactory.

Variants requiring less manual interaction have been developed, such as GrabCut [34], where a simple rectangular seed around the object of interest is manually initialized and an observation model is iteratively fitted through expectation maximization (EM). Bagon *et al*. [35] require a user to simply click a point inside the object of interest, and also use EM but to estimate a sophisticated self-similarity energy. These techniques can only optimize globally energies defined on local features such as contrast along the boundary and good pixel fit to a color or texture model. Interesting relaxation approaches exist for some energies whose minimization is NP-hard, such as curvature regularity of the boundary [36]. However many other more global properties may be more challenging to directly optimize, such as convexity or symmetry, motivating our ranking procedure. We differ from existing methods not only in our efficient parametric max-flow methodology to solve for multiple breakpoints of the cost, thus exploring a much large space of plausible segment hypotheses in polynomial time, but also in using regression-based ranking methods on generic mid-level features to score the generated segments and fully automate the process. No manual interaction is necessary in our method.

One of the big challenges in segmentation is to leverage the statistics of real world images in order to obtain more coherent spatial results. Methods that learn low-level statistics have been applied to distinguish real from apparent contours [37], [38], [39] and similar from dissimilar superpixels [24]. Pen and Veksler [40] proposed a learning procedure to select the best segment among a small set generated by varying the value of one parameter, in the context of interactive segmentation. Models based on mid-level properties have also been learned to distinguish good from bad regions [41]. High-level shape statistics can be incorporated into binary segmentation models, usually as non-parametric distributions of templates [42], [43], [44]. Expressive part-based appearance models have also been developed [45], [46], [47], [48]. It is likely that these methods may require bottom-up initialization, which an algorithm like ours can provide, as objects in real images exhibit large variability in pose, have high intra-class variation and are often occluded. Effectively leveraging shape priors in the initial steps of the visual processing pipeline may not always be feasible.

Our method aims to learn what distinguishes meaningful regions, covering full objects, from other accidental pixel groupings. Since our original publication [49], related ideas have been pursued also by Endres and Hoiem [50] who follow a processing pipeline related to ours, but employ a learned affinity measure between superpixels, rather than pixels, and a structured learning approach on a similar maximum marginal relevance measure to diversify ranking. To generate figure-ground segments, Levinshtein *et al*. [51] developed a procedure based on parametric max-flow principles similar to ours, but use a graph where similarity measures are constructed on superpixels. In parallel work, Alexe *et al.* [52] learn a naive Bayes model to distinguish bounding boxes enclosing objects from those containing amorphous background, without knowledge of the shape and appearance of particular object classes. They also show how to sample bounding boxes from this model efficiently. Salient object detection [53] approaches are also relevant to our work, but they focus on selection criteria inspired by attention mechanisms. We are instead interested in computing regions that cover well every object in an image, independently of whether they 'stand out' from the rest of the scene or not.

3 Constrained Parametric Min-Cuts (CPMC)

In order to generate a pool of segments with high probability of not missing object-quality regions, multiple constrained parametric min-cuts (CPMC) problems are solved with different seeds and unary terms. This leads to a large and diverse pool of segments of multiple spatial extent. The segments that correspond to implausible solutions are subsequently discarded using simple ratio cut criteria. The remaining segments are clustered so that all but representatives with low energy are retained, among the extremely similar ones. The final working set of segments is significantly reduced, with the most accurate segments preserved.

3.1 Setting up the Energy Functions

For each image, alternative sets of pixels are hypothesized to belong to the foreground—the foreground seeds. Then, for each set, we implicitly apply multiple levels of foreground bias, by assigning different costs on all remaining pixels but the ones assigned to background seeds. The foreground seeds are placed on a grid, while the background seeds are set along the image border. For each combination of foreground and background seeds we compute figure-ground segmentations, resulting from minimum cuts for multiple values of the foreground bias—searching over multiple foreground biases is intrinsic to our parametric max flow procedure. The optimization problem is formulated next.

Let $I(\mathcal{V}) \to R^3$ be an image defined on a set of pixels $\mathcal{V}$. As commonly done in graph-based segmentation algorithms, the similarity between neighboring pixels is encoded into edges of a weighted graph $G = (\mathcal{V}, \mathcal{E})$. Here, each pixel is a node in the set of nodes $\mathcal{V}$. The foreground and background partitions are represented by labels 1 and 0, respectively. Seed pixels $\mathcal{V}_f$ are constrained to the foreground and $\mathcal{V}_b$ to the background by setting infinity energy to any labeling where they receive the opposite label. Our overall objective is then to minimize an energy function over pixel labels $\{x_1, ..., x_k\}, x_i \in \{0, 1\}$, with k being the total number of pixels. In particular, we optimize the following energy function:

$$E^\lambda(X) = \sum_{u \in \mathcal{V}} D_\lambda(x_u) + \sum_{(u,v) \in \mathcal{E}} V_{uv}(x_u, x_v) \tag{1}$$

with $\lambda \in \mathbb{R}$, and unary potentials given by:

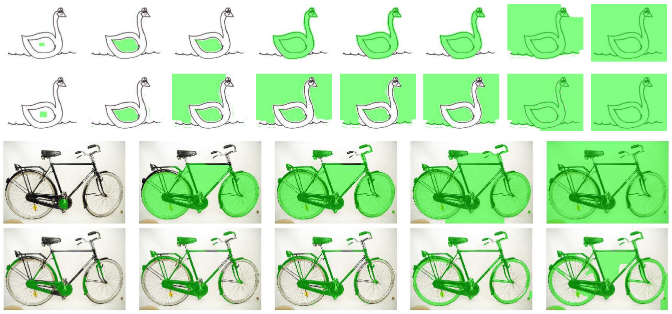

Fig. 2: Different effects of uniform and color-based unary terms. For illustration, a single foreground seed was placed manually at the same location for two energy problems, one with uniform and another with color unary terms. Shown are samples from the set of successive energy breakpoints (increasing λ values) from left to right, as computed by parametric max-flow. Uniform unary terms are used in rows 1 and 3. Color unary terms are used in even rows. Uniform unary terms are most effective in images where the background and foreground have similar color. Color unary terms are more appropriate for objects with elongated shapes.

$$D_\lambda(x_u) = \begin{cases} 0 & \text{if } x_u = 1, \ u \notin \mathcal{V}_b \\ \infty & \text{if } x_u = 1, \ u \in \mathcal{V}_b \\ \infty & \text{if } x_u = 0, \ u \in \mathcal{V}_f \\ f(x_u) + \lambda & \text{if } x_u = 0, \ u \notin \mathcal{V}_f \end{cases} \tag{2}$$

The foreground bias is implemented as a cost incurred by the assignment of non-seed pixels to background, and consists of a pixel-dependent value $f(x_u)$ and an uniform offset λ. Two different functions $f(x_u)$ are used alternatively. The first is constant and equal to 0, resulting in a uniform (variable) foreground bias. The second function uses color. Specifically, RGB color distributions $p_f(x_u)$ on seed $\mathcal{V}_f$ and $p_b(x_u)$ on seed $\mathcal{V}_b$ are estimated and derive $f(x_u) = \ln p_f(x_u) - \ln p_b(x_u)$. The probability distribution of pixel j belonging to the foreground is defined as $p_f(i) = \exp(-\gamma \cdot \min_j(||I(i) - I(j)||))$, with γ a scaling factor, and j indexes representative pixels in the seed region, selected as centers resulting from a k-means algorithm (k is set to 5 in our experiments). The background probability is defined similarly. This choice of function is motivated by efficiency, being much faster to estimate compared to a Gaussian mixture model.

Color-based unary terms are more effective when the color of the object is distinctive with respect to the background, as well as when the object has thin parts. Uniform unary terms are more useful in the opposite case. The complementary effects of these two types of unary energy terms are illustrated in fig. 2.

The pairwise term V_{uv} penalizes the assignment of different labels to similar neighboring pixels:

$$V_{uv}(x_u, x_v) = \begin{cases} 0 & \text{if } x_u = x_v \\ g(u,v) & \text{if } x_u \neq x_v \end{cases} \tag{3}$$

with similarity between adjacent pixels given by $g(u,v) = \exp\left[-\frac{\max(gPb(u), gPb(v))}{\sigma^2}\right]$. gPb returns the output of the multi-cue contour detector globalPb [32] at a pixel. The square distance is also an option that we experimented, with similar results, instead of the $\max$ operation. The *boundary sharpness* parameter σ controls the smoothness of the pairwise term.

The function defined by eq. 1 is submodular. Given a pair of foreground and background seeds and $f(x_u)$, the cost can be minimized exactly for all values of λ in the same complexity as a single max-flow problem, using a parametric solver [54]. In canonical form, parametric max-flow problems differ from max-flow problems in that capacities from the source node are allowed to be linear functions of a parameter, here λ. As λ (effectively our foreground bias) varies there are at most $(k-1)$ different cuts in the transformed graph, where k is the number of nodes, although for the graphs encountered in vision problems there are generally far fewer (see also our study in §3.3). The values of λ for which the cut values change are usually known as *breakpoints*. When the linear capacity functions from the source are either non-increasing or non-decreasing functions of λ, the problem is said to be monotonic. Our energy problems are monotonic because λ is multiplied by the same value, 1, in all unary terms. This important property implies that all cuts computed for a particular choice of source and sink seeds are nested.

In this work we use the *highest label pseudoflow* solver [55], which has complexity of $O(mN\log(N))$ for image graphs with N nodes and m edges. The complexity of the CPMC procedure is thus $O(kmN\log(N))$, as we solve multiple parametric max-flow problems, for each of the k combinations of foreground and background seeds, and for different choices of $f(x_u)$. The pseudoflow implementation requires a set of λ parameters for which to compute cuts. For the study in §3.3, we use additionally an implementation based on Gallo *et al*. [56] for the parametric analysis of a push-relabel max-flow solver which retrieves all breakpoints [57].

The graph construction that maps to the energy functions in (1) for each choice of foreground and background seed is similar to the one on [33], and requires to augment the graph G with two special nodes, source s and sink t that are required to be in separate partitions for any binary cut. The unary energy terms are encoded as edges between these special nodes and the nodes in V.

3.2 Effect of Grid Geometry

For the *foreground seeds*, we chose small solid squares. We have experimented with three different strategies to place them automatically: rectangular grid geometry, centroids of superpixels obtained with normalized cuts, and centroids of variable size regions, closest to each rectangular grid position, obtained using segments from the algorithm in [28]. As can be seen in table 1, the differences in results are not significant.

The *background seeds* are necessary in order to prevent trivial cuts that leave the background set empty. We used four different types: seeds covering the full image boundary, just the vertical edges, just the horizontal edges and all but the bottom image edge. This selection strategy allows us to extract objects that are only partially visible, due to clipping at different image boundaries.

In practice we solve around 180 instances of problem (1) for each image, for 30 λ values each (during processing, we skip duplicate breakpoints), defined on a logarithmic scale. The set of figure-ground segmentations is further enlarged by splitting the ones where the foreground has multiple connected components. The final pool has up to 10,000 segments.

As an alternative to multiple hard background seeds, it is possible to use a single soft background seed. This can be a frame one pixel wide covering the border of the image, with each pixel having a finite penalty associated to its assignment to the foreground. This construction is more efficient, as it decreases by 75% the number of energy problems to solve. We used this type of background seeds in an extension of the basic algorithm, presented in section §6.

Seed placement	MSRC score	Weizmann score
Grid	0.85 ± 0.1	0.93 ± 0.06
NCuts	0.86 ± 0.09	0.93 ± 0.07
FH	0.87 ± 0.08	0.93 ± 0.07

TABLE 1: Effect of spatial seed distribution. The use of superpixel segmentation algorithms (e.g. Normalized Cuts or FH [28]) to spatially distribute the foreground seeds does not significantly improve the average best segmentation covering score on the MSRC dataset, over regular seed geometries. On Weizmann, the average best F-measure is the same for all distributions, perhaps because the objects are large and any placement strategy eventually positions some seeds inside the object.

3.3 Effect of λ Schedule

We evaluated the effect of solving problem (1) for all λ values, instead of a preset logarithmic λ schedule, on the training set of the PASCAL VOC 2010 segmentation dataset (the typical distinction into training and testing is not relevant for the purpose of this experiment, where the goal is only to analyze the number of breakpoints obtained using different search strategies). We use a 6x6 regular grid of square seeds and solve using two procedures: (1) 20 values of λ sampled on a logarithmic scale (only the distinct energy optima are recorded) and, (2) all λ values, as computed as breakpoints of (1). We have computed the average computational time per seed, the ground truth covering score, and the number of breakpoints obtained under the two λ-search strategies. The results are shown in table 2. They suggest that a preset λ schedule is a sensible option. Using only 20 values produces almost the same covering as the one obtained using all values, it is 4 times faster and generates 10% of the total number of breakpoints, hence fewer segments. We also plot the distribution of the number of breakpoints per seed in figure 3, under the same experimental conditions. The frequency of breakpoints has a unimodal (bell) shape, with mean 110, but a slightly heavier tail in the direction of larger numbers of segments. There are never less than 15 breakpoints in this dataset.

# λ values	# breakpoints	Time (s)	Covering
20	12.3	1.8	0.713
all	114.6	7.5	0.720

# objects	1-2	3-4	5-6	7-13
# breakpoints all λ	112.19	124.60	125.29	142.83
# breakpoints 20 λ	12.27	12.64	13.08	13.45
# images	717	147	68	32

TABLE 2: Covering results obtained on the training set of VOC2010, based on a 6x6 grid of uniform seeds. The table compares the results of solving CPMC problems for 20 values of λ, sampled on a logarithmic scale, with the results obtained by solving for all possible values of λ. Shown are the average number of breakpoints per seed, and the average time required to compute the solutions for each seed. Computing all breakpoints for each seed provides modest ground truth covering improvements, at the cost of generating a larger number of segments and at increased computation time. The second table shows that images containing a larger number of ground truth objects tend to exhibit more breakpoints per seed.

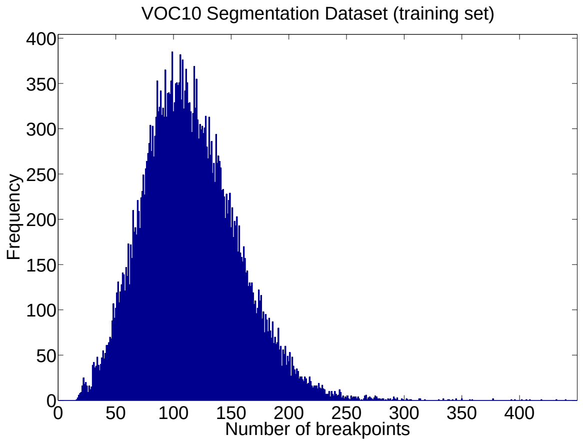

Fig. 3: Frequency of the parametric max flow breakpoints for each seed, on the training set of the VOC2010 segmentation dataset. These results were obtained using a 6x6 uniform grid of seeds. The number of breakpoints has mean 110, and a heavier tail towards a larger number of breakpoints.

3.4 Fast Segment Rejection

Generating a large set of segments increases the hit rate of the algorithm, but many segments are redundant or do not obey the statistics of real-world surfaces imaged by a camera. For images with large homogeneous regions, the original hypothesis generation step can produce many copies of the same segment because of the seeding strategy — every seed placed inside the region would tend to generate the same segment for the same λ. Moreover, sometimes visually arbitrary segments are created, as artifacts of the foreground bias strength and the seed constraints employed.

We deal with these problems using a fast rejection step. We first filter very small segments (up to 150 pixels in our implementation), then sort the segments using a simple criterion (we have used the ratio cut [58] as this is scale invariant and very selective) and retain up to 2,000 highest scoring segments. Then we hierarchically cluster the segments using overlap as a similarity measure, to form groups with all segments of at least 0.95 spatial overlap. For each cluster, we retain the segment with the lowest energy.

The number of segments that pass the fast rejection step is usually small, being indicative of how simple or cluttered an image is. In general, simple datasets have lower average number of segments. But even in the difficult PASCAL VOC 2009 dataset, the average was 154.

4 Mid-level Segment Ranking

Gestalt theorists [59], [60] argued that properties such as proximity, similarity, symmetry and good continuation are key to visual grouping. One approach would be to model such properties in the segmentation process [61], as long-range dependencies in a random field model. However, this poses significant modeling and computational challenges. With a segment set generated using weaker constraints, leveraging Gestalt properties becomes easier: rather than guide a complex inference procedure based on higher-order, long-range dependencies, we only need to check conformance with Gestalt regularities. It is therefore interesting to explore how the qualitative Gestalt theories can be implemented and what effects they produce in practice. An important question is whether Gestalt properties can be used to predict if segments have regularities typical of projections of real objects, in a manner that does not require prior knowledge about the class of the object in the image. This is a challenging problem, since the visual aspects of objects are extremely diverse. However, if object regularities can be identified, images could be represented by a handful of segments, which are easier to interpret and process by higher-level visual routines than a large set of pixels or superpixels.

In this work, we take an empirical approach: we compile a large set of features and annotated examples of segments of many objects from different categories, and use machine learning techniques to uncover their significance. Three sets of features (34 in total) are considered, representing graph, region and Gestalt properties. Graph properties, in particular variations of cut values, have long been used as cost functions in optimization methods for segmentation. Region properties encode mainly the statistics of where and at what scale objects tend to appear in images. Finally, Gestalt properties include mid-level cues like convexity and continuity, which can encode object regularities (e.g. objects background segments are usually non-convex and object boundaries are usually smoother than the boundaries of accidental, noisy, segments).

Graph partition properties (8 features) include the *cut* (sum of affinities along the segment boundary) [62], the *ratio cut* (sum of affinity along the boundary divided by their number) [58], the *normalized cut* (ratio of cut and affinity inside foreground, plus ratio of cut and affinity on background) [27], the *unbalanced normalized cut* (cut divided by affinity inside foreground) [31], and the *boundary fraction of low cut*, 4 binary variables signaling if the fraction of the cut is larger than a threshold, normalized by segment perimeter, for different thresholds.

Region properties (18 features) include area, perimeter, relative coordinates of the region centroid in the image, bounding box location and dimensions, major and minor axis lengths of the ellipse having the same normalized second central moments as the region, eccentricity, orientation, convex area, Euler number, diameter of a circle with the same area as the region, ratio of pixels in the region to pixels in the total bounding box, perimeter and absolute distance to the center of the image. Part of these features can be easily computed in Matlab with the *regionprops* function.

Gestalt properties (8 features) are implemented mainly as normalized histogram distances based on the χ^2 comparison metric: $\chi^2(x,y) = \sum_i \frac{(x_i - y_i)^2}{x_i + y_i}$ [63].

Let the texton histogram vector on the foreground region be t_f, and the one on the background be t_b. Then *inter-region texton similarity* is computed as the $\chi^2(t_f, t_b)$. *Intra-region texton similarity* is computed as $\sum_i \mathbf{1}(t_f(i) > k)$, with $\mathbf{1}$ the indicator function, and k a threshold, set to 0.3% the area of the foreground in our implementation. The textons are obtained through globalPb [1], which uses 65 nearest neighbor codewords.

Another two features we use are: *inter-region brightness similarity*, defined as $\chi^2(b_f, b_b)$, with b_f and b_b intensity histograms with 256 bins, and *Intra-region brightness similarity* defined as $\sum_i \mathbf{1}(b_f(i) > 0)$.

We also extract the *intra-region contour energy* as the sum of edge energy inside the foreground region, computed using globalPb, normalized by the length of the region perimeter. We also extract an *inter-region contour energy*, as the sum of edge energies along the boundary normalized by the perimeter.

The other Gestalt features we consider are *curvilinear continuity* and *convexity*. The first is the integral of the segment boundary curvature. We use an angle approximation to the curvature [64] on triplets of points sampled regularly every 15 pixels in our tests. Convexity is measured as the ratio of areas of the foreground region and its convex hull.

All features are normalized by subtracting their mean and dividing by their standard deviation.

4.1 Learning

We cast the problem of ranking the figure-ground hypotheses as regression on the largest **overlap** a segment has with a ground truth object, against its features. The definition of overlap is $O(S,G) = \frac{|S \cap G|}{|S \cup G|}$ [6]. This similarity function penalizes both under-segmentations and over-segmentations and has the advantage of being scale invariant. We experimented with both linear regression and random forests [65], a competitive non-linear model that predicts by averaging over multiple regression trees. Since the overlap induces a consistent ranking between all segments in the dataset, it is not necessary to use more specialized models that rank using pairwise preferences, such as the ranking SVM [66].

The *importance* of our features as learned by the random forests regressor [65], is shown in fig. 4. Some region properties appear to be quite informative, particularly features such as segment width and height and the location in the image. The ‘Minor Axis Length’ feature, which gets the highest importance works quite poorly in isolation, however, as shown in fig. 6, suggesting some cues are not informative in isolation, but correlate well to multiple other features. Convexity and the edge energy along the boundary, however, are assigned large importance, as expected.

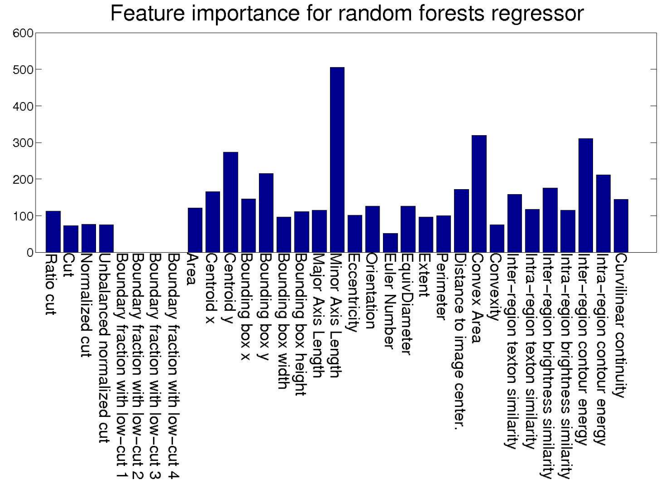

Fig. 4: Feature importance for the random forests regressor learned on the VOC2009 segmentation training set. The minor axis of the ellipse having the same normalized second central moments as the segment (here ‘Minor Axis Length’) is surprisingly the most important. This feature used in isolation results in relatively poor rankings however (see fig. 6). The Graph properties have small importance. The “Boundary fraction of low cut“ features, being binary, do not contribute at all. Gestalt features have above average importance, particularly the contour energies.

4.2 Maximum Marginal Relevance Diversification

The ranking results tend to place very similar segments in adjacent positions. An effective way to increase the quality of the first N segments is to **diversify** the ranking, which we do based on Maximal Marginal Relevance (MMR) measures [67]. To our knowledge this is the first application of such a technique to image segmentation. Starting with the originally top-scored segment, the MMR induces an ordering where the next selected segment (with maximum marginal relevance) is the one maximizing the original score minus a redundancy measure with respect to segments already selected. This procedure is iterated until all segments have been re-ranked. The redundancy measure we employ is the overlap with the set of previous segments selected based on the MMR measure.

Formally, let H be the full set of figure-ground segmentations and $H_p \subset H$ the hypotheses already selected. Let $s(H_i)$ be our predicted score for a given figure-ground segmentation and $o(H_i, H_j)$ the overlap between two figure-ground segmentations. The recursive definition for the next maximal marginal relevance selection is given as [67]:

$$MMR = \underset{H_i \in H \setminus H_p}{\operatorname{argmax}} \left[\theta \cdot s(H_i) - (1-\theta) \cdot \max_{H_j \in H_p} o(H_i, H_j))\right]$$

The first term is the score and the second is the redundancy. Parameter θ regulates the trade-off between the predicted score and the diversity measures in the first N selections. For example $\theta = 0$ will make the ranking ignore individual scores, and select the next set element with minimal overlap with any of the previously chosen elements. In contrast, $\theta = 1$ will always select the element with the highest score next. The best trade-off depends on the application. If high precision is desired then a higher weight should be given to the predicted score, whereas if recall is more important, then a higher weight should be given to diversity. If θ is very small, then ranking will be close to random. In our VOC experiments we have cross-validated at $\theta = 0.75$.

Ideas about selection of segments have been explored in the past, most notably by Ren and Malik [41]. They use a random search algorithm to iteratively hypothesize segmentations by combining different superpixels, and use a classifier to distinguish good segmentations from bad ones. For each segment, a feature vector is extracted, a classification score is computed, and the segmentation having the highest average score is selected. Images from the Berkeley Segmentation Dataset were used, with positive examples being matched with the corresponding human segmentation, and negative examples matched to a random human segmentation selected from a different image.

We differ from [41] in several important aspects: we use a superset of previously proposed features, including graph and region properties or convexity, we aim at obtaining independent object-level segments, and learn directly from object class recognition datasets. To learn how each segment obeys the statistical regularities of real-world objects, we train a regression model, not a classifier, hence we do not need to synthesize negative examples. Finally, we precompute an accurate set of figure-ground segmentations, making the process more efficient.

5 Experiments

We study both the quality of the pool of object hypotheses generated by CPMC, and the loss in quality incurred by selecting the topmost N object hypotheses, as opposed to working with a much larger pool. We use three publicly available datasets: Weizmann's Segmentation Evaluation Database [5], MSRC [4] and the VOC2009 train and validation sets for the object-class segmentation problem [6].

Weizmann consists of 100 gray-valued images having a single prominent foreground object. The goal is to generate coverage of the entire spatial support of the object in the image using a single segment, and as accurately as possible. We compare the performance of CPMC with published results from two state of the art segmentation algorithms. The results are reported using the best **F-measure** criterion, $F = \frac{2RP}{P+R}$, where P and R are the precision and recall of pixels in a segment relative to the ground truth [5]. Only the best F-measure in each image is relevant for the final score, because there is only one object in each image.

The MSRC dataset is quite different, featuring 23 different classes, including some *stuff* classes, such as water and grass. It has up to 11 objects present in each of its nearly 600 images. We use this dataset to evaluate the quality of the pool of segments generated, not individual rankings. The VOC 2009 dataset is challenging for segmentation, as it contains real-world images from Flickr, with 20 different classes of objects. The background regions are not annotated. MSRC and VOC2009 contain multiple ground-truth objects per image, therefore we use the **segmentation covering** [1] as an accuracy measure. The extent of covering a set of ground truth segments S by a set of machine segments S' is defined as:

$$C(S, S') = \frac{1}{N} \sum_{R \in S} |R| * \max_{R' \in S'} O(R, R') \qquad (4)$$

where N is the number of pixels in the image, $|R|$ is the number of pixels in the ground truth segment R, and O is the overlap.

5.1 Segment Pool Quality

The automatic results obtained using CPMC on the Weizmann dataset are shown in table 3 together with the previous best result, by Bagon et al [35], which requires the user to click a point inside the object. We also compare to the method of Alpert *et al*. [5], which is automatic. Results for CMPC were obtained using an average of 53 segments per image. Visibly, it generates an accurate pool of segments. Results on MSRC and VOC2009 are compared in table4 to Arbeláez *et al*. [1], which is arguably one of the state of the art methods for low-level segmentation. The methodology of the authors was followed, and we report the average best coverings. We use all the unique segments in the hierarchy returned by their algorithm [1] to compute the score. The pool of segments produced by CPMC is significantly more accurate and has an order of magnitude fewer segment hypotheses. A filtering procedure could be used for gPb-owt-ucm to reduce the number segments, but at a potential penalty in quality. The dependency between the quality of segments and the size of the ground truth objects is shown in fig. 5.

Weizmann	F-measure
CPMC	0.93 ± 0.009
Bagon *et al.*	0.87 ± 0.010
Alpert *et al.*	0.86 ± 0.012

TABLE 3: Average of best segment F-measure scores over the entire dataset. Bagon's algorithm is interactive. Alpert's results were obtained automatically. The table shows that for each image, among the pool of segment hypotheses produced by CPMC, there is usually one segment which is extremely accurate. The average number of segments that passed the fast rejection step was 53 in this dataset.

5.2 Ranking Object Hypotheses

We evaluate the quality of our ranking method on both the validation set of the VOC2009 segmentation dataset, and on hold-out sets from the Weizmann Segmentation Database. The training set of VOC2009 consists of 750 images, resulting in $114,000$ training examples, one for each segment passing the

MSRC	Covering	N Segments
CPMC	0.85 ± 0.1	57
gPb-owt-ucm	0.78 ± 0.15	670
VOC2009	**Covering**	**N Segments**
CPMC	0.78 ± 0.18	154
gPb-owt-ucm	0.61 ± 0.20	1286

TABLE 4: Average best image covering scores on MSRC and VOC2009 train+validation datasets, compared to Arbeláez *et al.* [1], here gPb-owt-ucm. Scores show the best covering of ground truth by segments produced using each algorithm. CPMC results before ranking are shown, to evaluate the quality of the pool of segments from various methods.

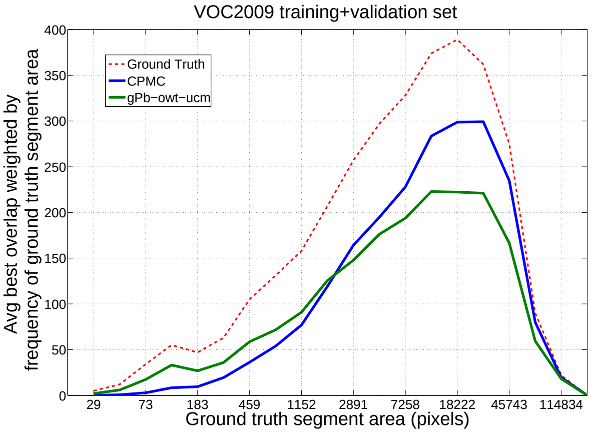

Fig. 5: Quality of the segments in VOC2009 joint train and validation sets for the segmentation problem, as a function of the area of the ground truth segments. Medium and large size objects, that are more frequent, are segmented significantly more accurately by CPMC than by gPb-owt-ucm [1], in this case.

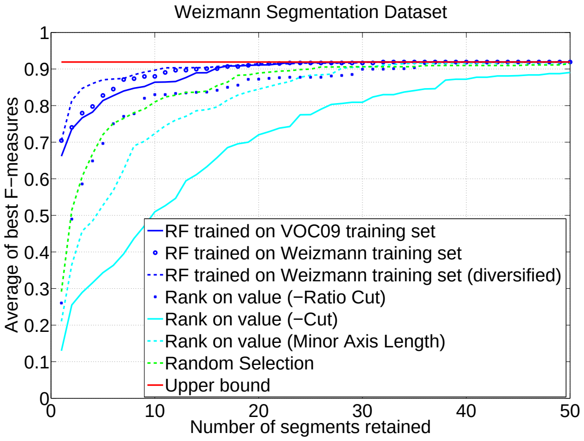

Fig. 6: Average best segment F-measure as we vary the number of retained segments given by our ranking procedure. Results were averaged over three different splits of 50 training and 50 testing images. Note that when working with our top-scored 5 segments per image, the results already equal the ones obtained by the interactive method of Bagon *et al.* [35]. Note also that using this learned ranking procedure, it is possible to compress the original pool of segments to a fifth (10 segments), at negligible loss of quality.

fast rejection step. On the Weizmann Segmentation Database we randomly select 50 images, resulting in $2,500$ training examples, and we test on the remaining 50 images. On Weizmann we compare a random forests regressor trained on the images in that dataset with a predictor trained on VOC2009. The results in fig. 6 are similar, showing that the model is not overfitting to the statistics of the individual datasets. This also shows that it is possible to learn to rank segments of arbitrary objects, using training regions from only 20 classes. The learned models are significantly better than ranking using the value of any single feature such as the cut or the ratio cut. On VOC2009 we have also run experiments where we have complemented the initial feature set with additional appearance and shape features — a bag of dense SIFT [68] features computed on the foreground mask, a bag of Local Shape Contexts [69] computed on its boundary, and a HOG pyramid [70] with 3 levels computed on the bounding box fitted on the boundary of the segment, for a total of 1,054 features. In this case, we trained a linear regressor for ranking (this is significantly faster than random forests, which takes about 8 hours to train on the basic 34 features). The results are shown in fig. 7. Clearly the new features help somewhat, producing results that are slightly better than the ones obtained by the linear regressor on the basic feature set. However, these are not better than a random forests model trained on the basic feature set. This shows that the set of basic features is already quite expressive in conjunction with nonlinear models.

Notice that by using this ranking procedure, followed by diversification, we can obtain object hypotheses of superior quality of those provided by the segmentation algorithm of [1]. In fact, by using the top 7 segments produced by our ranking procedure, we obtain the same accuracy, 0.61, as obtained using the full hierarchy of $1,286$ distinct segments in [1].

6 Subframe-CPMC

We have experimented with a different variant of the algorithm, the Subframe-CPMC, on the Pascal VOC2010 dataset. The goal was to achieve high object recall while at the same time preserving segmentation accuracy, with a mindset towards detection applications. To score a detection hypothesis as correct, benchmarks such as the Pascal VOC require a minimum overlap between correctly classified regions and the ground truth regions. In addition, benchmarks disregard the area of the ground truth regions (e.g. an object with 500 pixels is just as important as one occupying the full image), hence what matters is not so much achieving high *covering* scores (which explicitly take into account the size of the segments), but high *overlap*.

Subframe-CPMC uses an additional type of seed, and is configured to generate a larger number of segments. First we make the overall process faster by solving the energy problems at half of the image resolution. Quantitative results

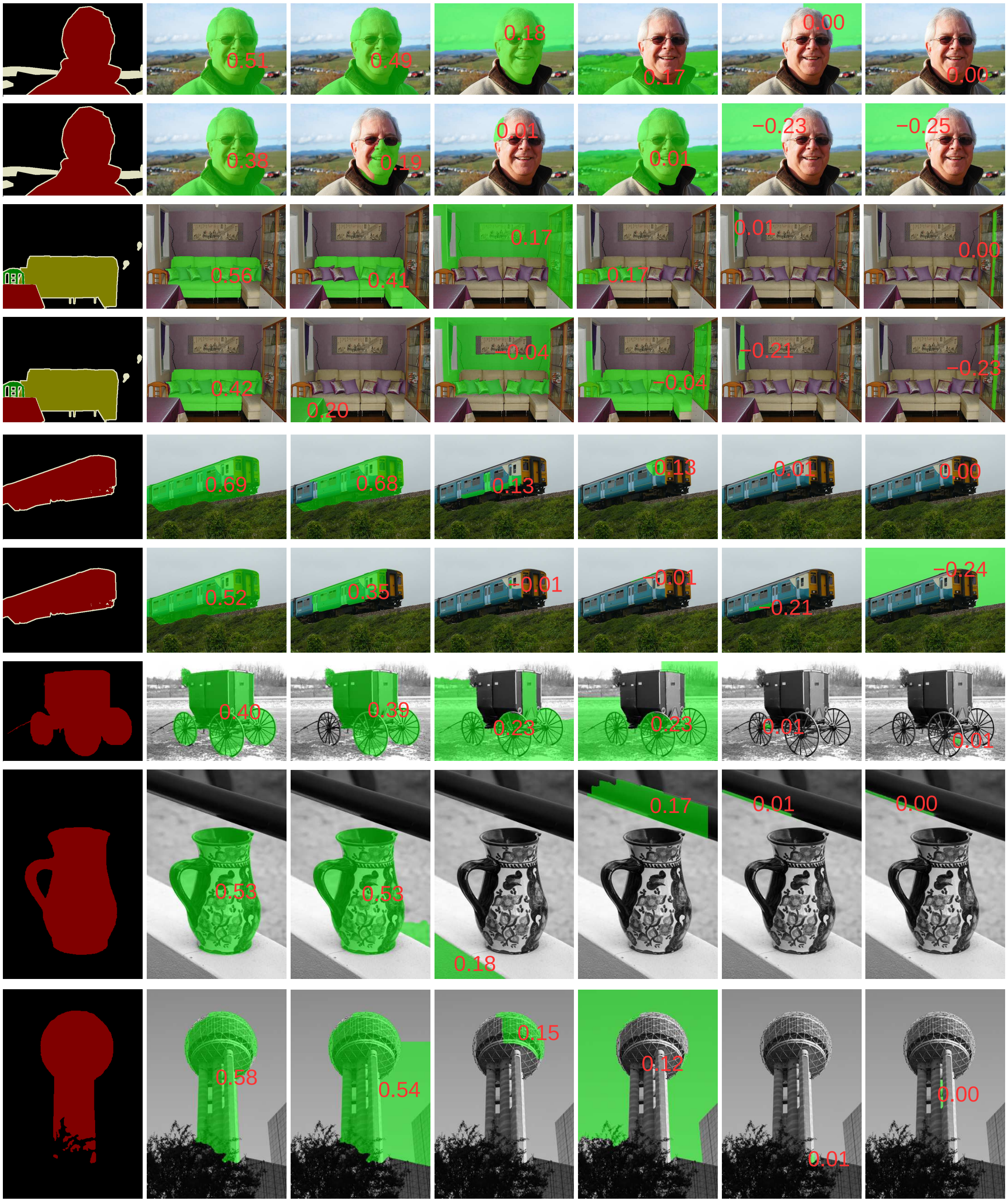

Fig. 8: Ranking results obtained using the random forests model learned on the VOC2009 training set. The green regions are the segment foreground hypotheses. The first image on each row shows the ground truth, the second and third images show the most plausible segments given by CPMC, the last two images show *the least* plausible segments, and the fourth and fifth images show segments *intermediately* placed by the ranking. The predicted segment scores are displayed in overlay. The first three images are from the VOC2009 validation set and rows 2, 4 and 6 show the diversified rankings, with $\theta = 0.75$. Note that in the diversified ranking, segments scored nearby tend to be more dissimilar. The last three rows show results from the Weizmann Segmentation Database. The algorithm has no prior knowledge of the object classes, but on this dataset, it still shows a remarkable preference for segments with large spatial overlap with the imaged objects. There are neither chariots nor vases in the training set, for example. The lowest ranked object hypotheses are usually quite small reflecting perhaps the image statistics in the VOC2009 training set.

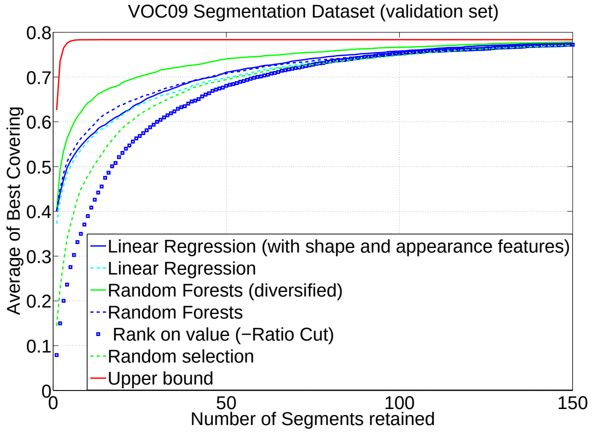

Fig. 7: Complementing the basic descriptor set with appearance and shape features improves the ranking slightly, but the basic set is still superior when used in conjunction with a more expressive random forests regressor. Further diversifying the ranking improves the average best covering given by the first top N segments significantly.

were equivalent. We also changed the seeding strategy to use a single soft background seed. We also increased the number of foreground seeds, by using a grid of 6x6 instead of the previously $5\text{x}5$, and reduced the value of the σ parameter by 30% in eq. 3, resulting in more segments due to sharper boundaries.

We have also complemented the existing seeds with *subframes*, background seeds composed of the outside of rectangles covering no more than 25% of the area in the image, with a single square foreground seed in the center. These seeds constrain segments to smaller regions in the image, as they force the possible contours to lie inside the rectangular region. This is especially helpful for segmenting small objects in cluttered regions. For this type of seed we also solve problems with and without a color unary term. Two alternative types of subframe seeds were tried: a $5\text{x}5$ regular grid of square subframes of fixed dimension, with width set to 40% of the image, and bounding boxes from a deformable parts detector [8], [71] with default parameters, set to the regime of high recall but low precision. For the detector, we discard class information and keep the 40 top-scored bounding boxes smaller than a threshold C, in this case 25% of the image area. Subframe energy problems are optimized efficiently by shrinking all nodes corresponding to pixels in background seeds into a single node, thereby reducing the size of the graph significantly.

The parameter σ, controlling the sharpness of the boundary, has an important influence on the number of generated segments. A value of 2.5 with the color-based seeds leads to 225 segments, average overlap of 0.61 and covering of 0.74, while for $\sigma = 1$ the method produces an average of 876 segments, average overlap of 0.69 and covering 0.76. We used $\sigma = 1$ for the uniform seeds, $\sigma = \sqrt{2}$ for the color seeds, and $\sigma = \sqrt{0.8}$ for the subframe seeds. This leads to a larger pool of segments, but also to higher quality segments, as noticeable in table 5.

Additional Features: Working with a larger pool of segments poses additional demands on the accuracy of ranking. An improvement we pursued was to enlarge the set of mid-level features with shape and texture features. The dimensionality of these features, together with the large number of training examples, makes linear regression the most practical learning procedure, as discussed in section 5.2. Histogram features, however, are known to be most effective when used with certain nonlinear similarities, such as a Laplacian-RBF kernel $k(\mathbf{x}, \mathbf{y}) = \exp(-\sum |x_i - y_i|)$ [63]. Some of these similarity functions can nevertheless be handled with linear regression, by first applying a randomized feature map that approximates the Laplacian-RBF kernel [72], [73].

As texture features we extracted two bags of words for each segment: one defined over gray-level SIFT features and the other over color SIFT features, both sampled every 4 pixels and at 4 different scales (16, 24, 36 and 54 pixels wide) to ensure a degree of scale invariance. Each one was quantized using a 300-dimensional codebook. As shape features we computed two pyramid HOGs, both with gradient orientation quantized into 20 bins, the first with the background segment gradients masked out on a pyramid composed of four levels, for a total of $1,700$ dimensions. The other PHOG was computed directly on the contour of the segment, with both foreground and background gradients masked out and a pyramid of three levels for a total of 420 dimensions. We map the joint vector of the two bags of words for texture features into a $2,000$-dimensional randomized feature map drawn from the Fourier transform of the Laplacian-RBF kernel [72], and process similarly the two PHOGs corresponding to shape features. We also append our original 34-dimensional feature set resulting in a total of $4,034$ features.

VOC2010 Results: The overlap measure is popular for distinguishing hits from misses in detection benchmarks. In the VOC2010 dataset, besides the overlap, we evaluate the recall under two different hit-metrics: 50% segment overlap and 50% bounding box overlap. Using the 50% segment overlap criterion the algorithm obtains, average per class, 87.73% and 83.10% recall, using 800 and 200 segments per image, respectively. Under a 50% bounding box overlap criterion, the algorithm achieves 91.90% using 800 segments and 87.65%, using 200 segments.

The top 200 ranked segments gave on average 0.82 covering and 0.71 overlap, which improves upon the results of the basic algorithm on the VOC2009 (0.78 and 0.66 with all segments). Details are shown in figs. 12 and 13; images are shown in fig. 10. The learned estimated weights of the linear regressor for all features are displayed in fig.9.

7 Conclusions

We have presented an algorithm that casts the automatic image segmentation problem as one of finding a set of plausible figure-ground object hypotheses. It does so by learning to rank figure-ground segmentations, using ground truth annotations

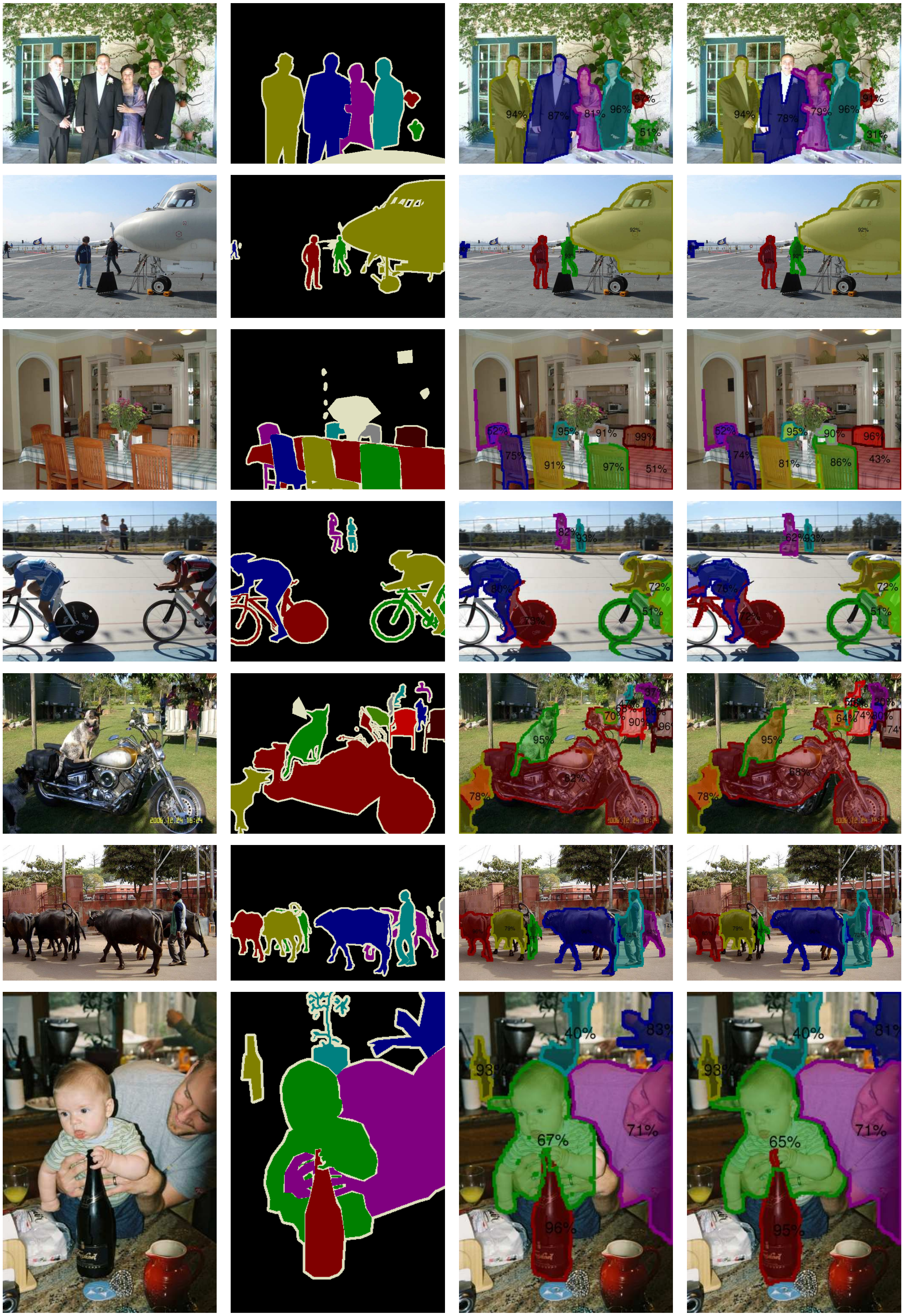

Fig. 10: Segmentation results on images from the validation set of the VOC2010 database. The **first** column contains the original images, the **second** gives the human ground truth annotations of multiple objects, the **third** shows the best segment in the Subframe-CPMC pool for each ground truth object, the **fourth** shows the best segment among the ones ranked in the top-200. The proposed algorithm obtains accurate segments for objects at multiple scales and locations, even when they are spatially adjacent. See fig. 11 for challenging cases.

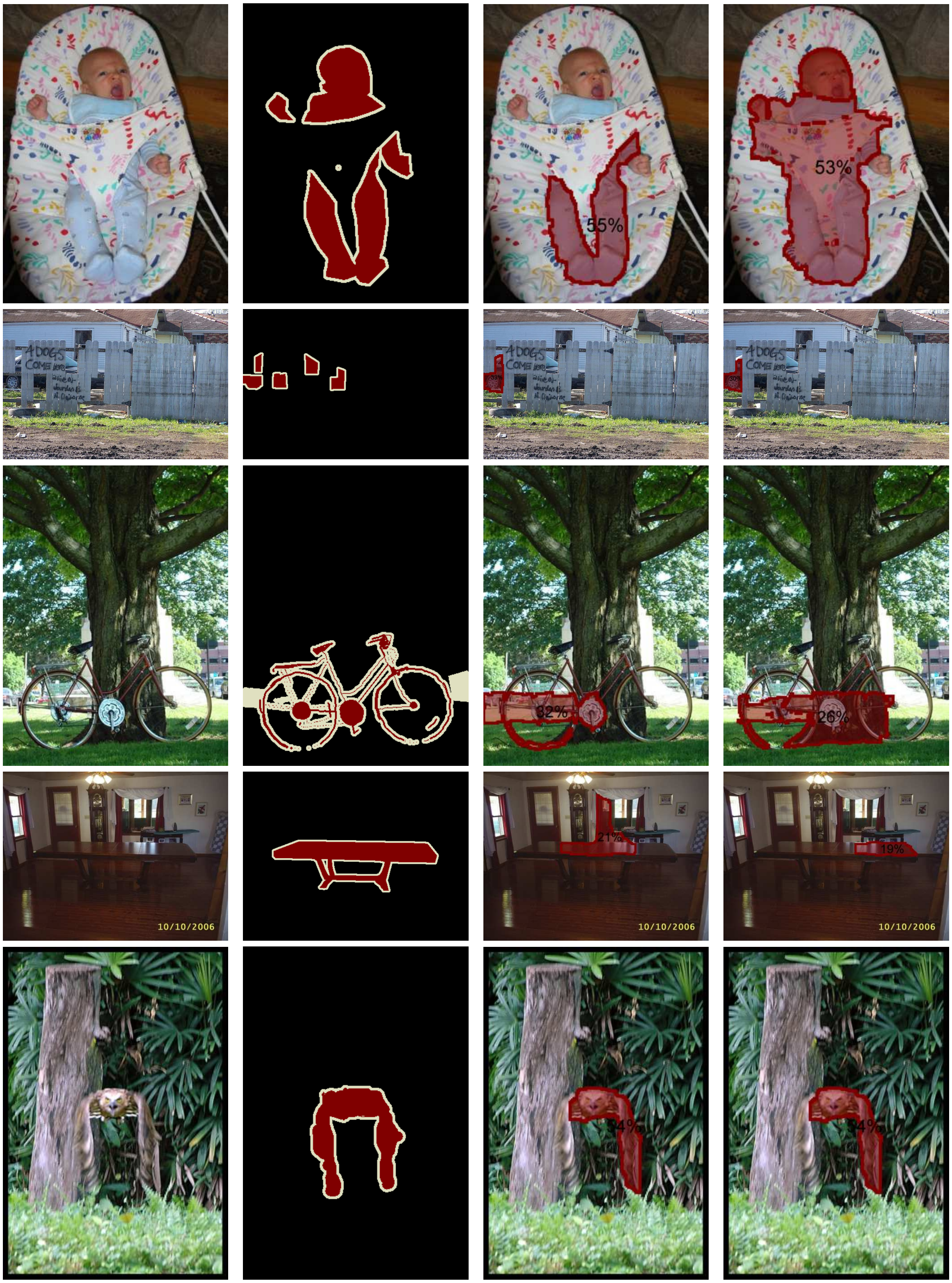

Fig. 11: Examples, taken from the validation set of VOC2010, where the CPMC algorithm encounters difficulties. The **first** column shows the images, the **second** the human ground truth annotations of multiple objects, the **third** shows the best segment in the entire Subframe-CPMC pool for each ground truth object, the **fourth** shows the best segment among the ones ranked in the top-200. Partially occluded objects (first two rows), wiry objects (third row) and objects with low background contrast (fourth and fifth row) can cause difficulties.

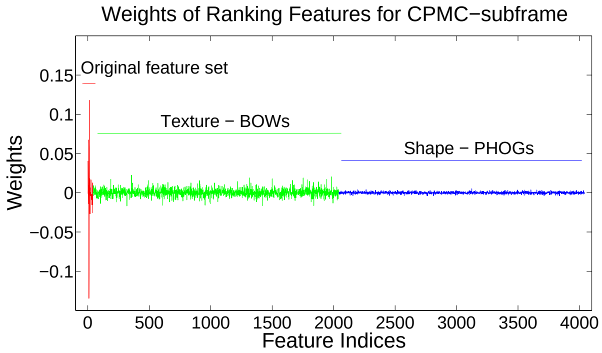

Fig. 9: Learned feature weights for the Subframe-CPMC model. The original set of mid-level features and region properties gets higher weights, texture features get intermediate weights and shape features get smaller weights. Texture features might help discard amorphous *stuff* regions such as grass, water and sky.

Quality Measure	Grid Subframes	BB Detector	No Subframes
Overlap	0.74	0.76	0.71
Covering	0.83	0.84	0.82
N segments	736	758	602

TABLE 5: Results on the training set of the VOC2010 segmentation dataset. Color and uniform seeds are complemented with subframe seeds, either placed on a regular grid or obtained from a bounding box detector. Using a regular grid gives only slightly inferior results compared to using detector responses. Both result in a large improvement in the recall of small objects, compared to models that do not use subframes. This is reflected in the overlap measure, which does not take into account the area of the segments.

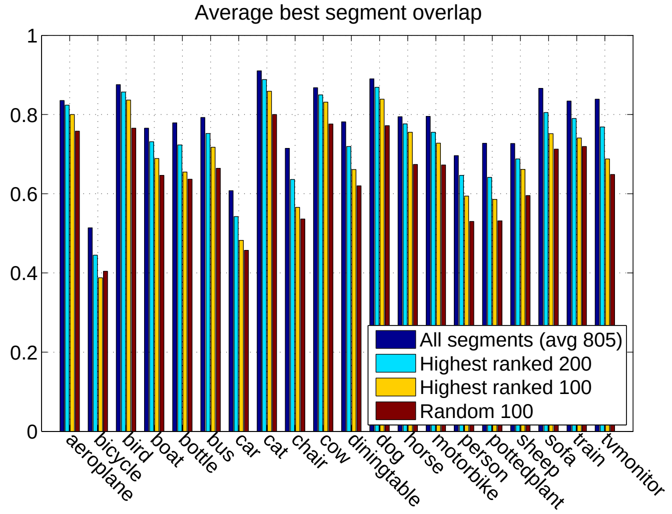

Fig. 12: Average overlap between ground truth objects and the best Subframe-CPMC segments, on the validation set of VOC2010. Certain classes are considerably harder to segment by the algorithm, such as bicycles, perhaps due to their wiry structure.

available in object class recognition datasets and based on a set of low and mid-level properties. The algorithm uses a very powerful new procedure to generate a pool of figure-ground segmentations–the Constrained Parametric Min-Cuts (CPMC). This uses parametric max-flow to efficiently compute figure-ground hypotheses at multiple scales on an image grid, followed by maximum relevance ranking and diversification. We have shown that the proposed framework is able to generate compact sets of segments that represent the objects in an image more accurately than existing state of the art segmentation methods. These sets of segments have been used successfully in a segmentation-based recognition framework [2], as well as, more recently, for multi-region image segmentation [3].

One difficulty of the current method is in handling of objects composed of disconnected regions that may arise from occlusion. While the energy minimization problems we solve sometimes generate such multiple regions, we chose to separate them into individual connected components, because they only rarely belong to the same object. In many such cases it may not be possible to segment the object correctly without top-down information (e.g. segmenting people embraced might require the knowledge of the number of arms a person has, and the configurations they can be in). It might be possible to handle the problem in a bottom-up fashion for simple cases, when cues like strong continuity may be exploited, but it appears more promising to do such analysis at a later stage of scene interpretation.

A somewhat suboptimal aspect of the proposed method is that energy minimization problems are solved independently, and the same number of problems is generated for all images, notwithstanding some having a single object and others having plenty. An interesting extension would make the process dynamic by making decisions on where and how to extract more segments conditioned on the solutions of previous problems. This would be conceivably more efficient and would make the transition to video smoother. It may also speed up processing and it should also be possible to stop early and degrade gracefully, when working on a temporal budget. A conditional sequential process could also make for a more biologically plausible control structure.

Acknowledgements

This work was supported, in part, by CNCSIS-UEFISCU, under project number PN II-RU-RC-2/2009, and by the European Commission, under MCEXT-025481.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "From contours to regions: An empirical evaluation," in *IEEE International Conference on Computer Vision and Pattern Recognition*, June 2009, pp. 2294–2301.

[2] F. Li, J. Carreira, and C. Sminchisescu, "Object recognition as ranking holistic figure-ground hypotheses," in *IEEE International Conference on Computer Vision and Pattern Recognition*, June 2010.

[3] J. Carreira, A. Ion, and C. Sminchisescu, "Image segmentation by discounted cumulative ranking on maximal cliques," Computer Vision and Machine Learning Group, Institute for Numerical Simulation, University of Bonn, Tech. Rep. 06-2010, June 2010.

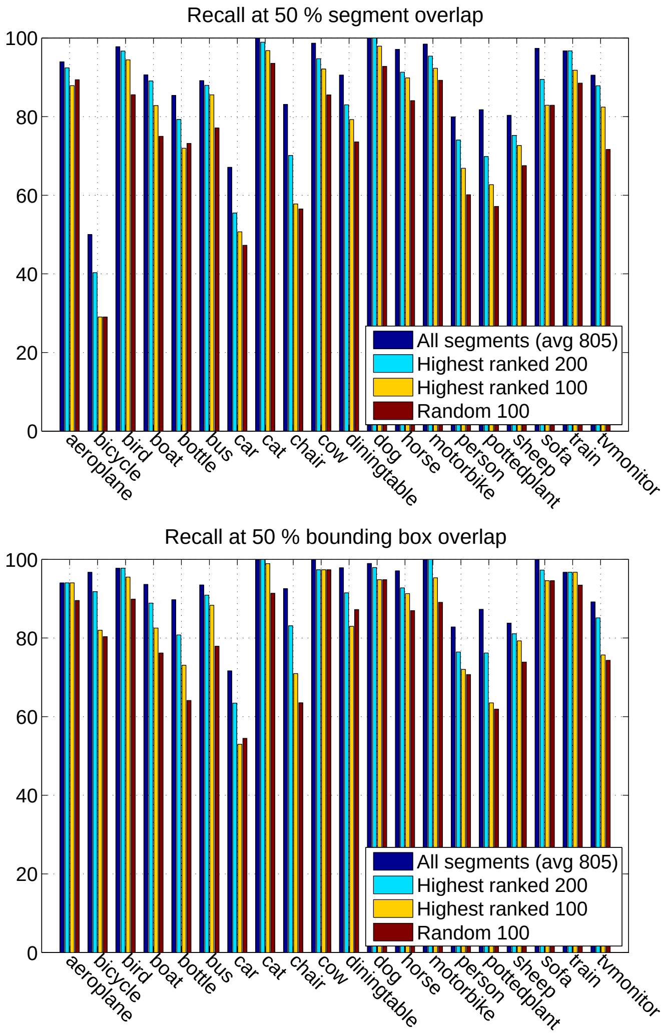

Fig. 13: Recall at 50% overlap between regions of ground truth objects and the best Subframe-CPMC segments (**top**) and between ground truth bounding boxes and best Subframe-CPMC segment bounding boxes (**bottom**). Note that bicycles are difficult to segment accurately due to their wiry structure, but there is usually some segment for each bicycle that has an accurate bounding box, such as the ones shown in the third row of fig. 2. These results are computed on the validation set of the VOC2010 segmentation dataset.

[4] J. Shotton, J. M. Winn, C. Rother, and A. Criminisi, "Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation." in *European Conference on Computer Vision*, May 2006, pp. 1–15.

[5] M. G. R. B. S. Alpert and A. Brandt, "Image segmentation by probabilistic bottom-up aggregation and cue integration," *IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 0, pp. 1–8, 2007.

[6] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results," http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html.

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results," http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html.

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 1627–1645, 2010.

[9] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman, "Multiple kernels for object detection," in *IEEE International Conference on Computer Vision*, September 2009.

[10] S. Gould, T. Gao, and D. Koller, "Region-based segmentation and object detection," in *Advances in Neural Information Processing Systems*, Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, Eds., December 2009, pp. 655–663.

[11] Y. Yang, S. Hallman, D. Ramanan, and C. Fowlkes, "Layered object detection for multi-class segmentation," in *IEEE International Conference on Computer Vision and Pattern Recognition*, June 2010.

[12] L. Ladicky, P. Sturgess, K. Alaharia, C. Russel, and P. H. Torr, "What, where & how many ? combining object detectors and crfs," in *European Conference on Computer Vision*, September 2010.

[13] J. M. Gonfaus, X. Boix, J. van de Weijer, A. D. Bagdanov, J. Serrat, and J. Gonzàlez, "Harmony potentials for joint classification and segmentation," in *IEEE International Conference on Computer Vision and Pattern Recognition*, San Francisco, California, USA, June 2010, pp. 1–8.

[14] B. Russell, W. Freeman, A. Efros, J. Sivic, and A. Zisserman, "Using multiple segmentations to discover objects and their extent in image collections," *IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 1605–1614, June 2006.

[15] J. Carreira and C. Sminchisescu, "Constrained parametric min-cuts for automatic object segmentation, release 1," http://sminchisescu.ins.uni-bonn.de/code/cpmc/.

[16] J. Muerle, , and D. Allen, "Experimental evaluation of techniques for automatic segmentation of objects in a complex scene." in *Pictorial Pattern Recognition*, 1968, pp. 3–13.

[17] R. M. Haralick and L. G. Shapiro, "Image segmentation techniques," *Computer Vision, Graphics, and Image Processing*, vol. 29, no. 1, pp. 100–132, 1985.

[18] S. Zhu, T. Lee, and A. Yuille, "Region competition: unifying snakes, region growing, energy/bayes/mdl for multi-band image segmentation," in *Computer Vision, 1995. Proceedings., Fifth International Conference on*, Jun. 1995, pp. 416 –423.

[19] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 99, no. PrePrints, 2010.

[20] J. L. Mundy, "Object recognition in the geometric era: A retrospective," in *Toward Category-Level Object Recognition*, 2006, pp. 3–28.

[21] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE International Conference on Computer Vision and Pattern Recognition*, December 2001.

[22] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *ICCV*, vol. 2, July 2001, pp. 416–423.

[23] R. Unnikrishnan, C. Pantofaru, and M. Hebert, *Toward Objective Evaluation of Image Segmentation Algorithms*, vol. 29, no. 1, pp. 929–944, June 2007.

[24] D. Hoiem, A. A. Efros, and M. Hebert, "Geometric context from a single image," *IEEE International Conference on Computer Vision*, vol. 1, pp. 654–661, October 2005.

[25] T. Malisiewicz and A. Efros, "Improving spatial support for objects via multiple segmentations," *British Machine Vision Conference*, September 2007.

[26] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[27] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

[28] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, September 2004.

[29] T. S. A. Stein and M. Hebert, "Towards unsupervised whole-object segmentation: Combining automated matting with boundary detection," in *IEEE International Conference on Computer Vision and Pattern Recognition*, June 2008.

[30] A. Rabinovich, S. Belongie, T. Lange, and J. M. Buhmann, "Model order selection and cue combination for image segmentation," *IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 1130–1137, June 2006.

[31] E. Sharon, M. Galun, D. Sharon, R. Basri, and A. Brandt, "Hierarchy and adaptivity in segmenting visual scenes," *Nature*, vol. 442, no. 7104, pp. 719–846, June 2006.

[32] M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik, “Using contours to detect and localize junctions in natural images,” *IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 0, pp. 1–8, June 2008.

[33] Y. Boykov and G. Funka-Lea, “Graph cuts and efficient n-d image segmentation,” *International Journal of Computer Vision*, vol. 70, no. 2, pp. 109–131, 2006.

[34] C. Rother, V. Kolmogorov, and A. Blake, “”grabcut”: interactive foreground extraction using iterated graph cuts,” *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.

[35] S. Bagon, O. Boiman, and M. Irani, “What is a good image segment? a unified approach to segment extraction,” *European Conference on Computer Vision*, pp. 30–44, October 2008.

[36] T. Schoenemann, F. Kahl, and D. Cremers, “Curvature regularity for region-based image segmentation and inpainting: A linear programming relaxation,” *IEEE International Conference on Computer Vision*, 2009.

[37] C. Fowlkes, D. Martin, and J. Malik, “Learning affinity functions for image segmentation: combining patch-based and gradient-based approaches,” in *IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 2, June 2003, pp. II–54–61 vol.2.

[38] P. Dollár, Z. Tu, and S. Belongie, “Supervised learning of edges and object boundaries,” in *IEEE International Conference on Computer Vision and Pattern Recognition*, June 2006.

[39] J. Kaufhold and A. Hoogs, “Learning to segment images using region-based perceptual features,” *IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 954–961, June 2004.

[40] B. Peng and O. Veksler, “Parameter Selection for Graph Cut Based Image Segmentation,” in *British Machine Vision Conference*, September 2008.

[41] X. Ren and J. Malik, “Learning a classification model for segmentation,” *IEEE International Conference on Computer Vision*, vol. 1, p. 10, October 2003.

[42] V. Lempitsky, A. Blake, and C. Rother, “Image segmentation by branch-and-mincut,” in *European Conference on Computer Vision*, October 2008, pp. IV: 15–29.

[43] D. Cremers, F. R. Schmidt, and F. Barthel, “Shape priors in variational image segmentation: Convexity, lipschitz continuity and globally optimal solutions,” *IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 0, pp. 1–6, June 2008.

[44] T. Schoenemann and D. Cremers, “Globally optimal image segmentation with an elastic shape prior,” *IEEE International Conference on Computer Vision*, vol. 0, pp. 1–6, October 2007.

[45] E. Borenstein, E. Sharon, and S. Ullman, “Combining top-down and bottom-up segmentation,” *Computer Vision and Pattern Recognition Workshop*, pp. 46–46, June 2004.

[46] B. Leibe, A. Leonardis, and B. Schiele, “Robust object detection with interleaved categorization and segmentation,” *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 259–289, 2008.

[47] A. Levin and Y. Weiss, “Learning to combine bottom-up and top-down segmentation,” *International Journal of Computer Vision*, vol. 81, no. 1, pp. 105–118, 2009.

[48] M. P. Kumar, P. Torr, and A. Zisserman, “Objcut: Efficient segmentation using top-down and bottom-up cues,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 530–545, 2010.

[49] J. Carreira and C. Sminchisescu, “Constrained Parametric Min-Cuts for Automatic Object Segmentation,” in *IEEE International Conference on Computer Vision and Pattern Recognition*, June 2010.

[50] I. Endres and A. Hoiem, “Category independent object proposals,” in *European Conference on Computer Vision*, September 2010.

[51] A. Levinshtein, C. Sminchisescu, and S. Dickinson, “Optimal contour closure by superpixel grouping,” in *European Conference on Computer Vision*, September 2010.

[52] B. Alexe, T. Deselaers, and V. Ferrari, “What is an object?” in *IEEE International Conference on Computer Vision and Pattern Recognition*, June 2010.

[53] T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H.-Y. Shum, “Learning to detect a salient object,” in *CVPR*, 2007, pp. 1 –8.

[54] V. Kolmogorov, Y. Boykov, and C. Rother, “Applications of parametric maxflow in computer vision,” *IEEE International Conference on Computer Vision*, pp. 1–8, October 2007.

[55] D. S. Hochbaum, “The pseudoflow algorithm: A new algorithm for the maximum-flow problem,” *Oper. Res.*, vol. 56, pp. 992–1009, July 2008.

[56] G. Gallo, M. D. Grigoriadis, and R. E. Tarjan, “A fast parametric maximum flow algorithm and applications,” *SIAM J. Comput.*, vol. 18, no. 1, pp. 30–55, 1989.

[57] M. A. Babenko, J. Derryberry, A. V. Goldberg, R. E. Tarjan, and Y. Zhou, “Experimental evaluation of parametric max-flow algorithms,” in *WEA*, 2007, pp. 256–269.

[58] S. Wang and J. M. Siskind, “Image segmentation with ratio cut,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 675–690, 2003.

[59] M. Wertheimer, “Laws of organization in perceptual forms (partial translation),” in *A sourcebook of Gestalt Psychology*, 1938, pp. 71–88.

[60] S. E. Palmer, *Vision Science: Photons to Phenomenology*. The MIT Press, 1999.

[61] S. C. Zhu and D. Mumford, “Learning Generic Prior Models for Visual Computation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 11, 1997.

[62] Z. Wu and R. Leahy, “An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1101–1113, 1993.

[63] O. Chapelle, P. Haffner, and V. Vapnik, “Support vector machines for histogram-based image classification,” *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1055–1064, 1999.

[64] A. M. Bruckstein, R. J. Holt, and A. N. Netravali, “Discrete elastica,” in *DCGA ’96: Proceedings of the 6th International Workshop on Discrete Geometry for Computer Imagery*. London, UK: Springer-Verlag, 1996, pp. 59–72.

[65] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[66] T. Joachims, “Optimizing search engines using clickthrough data,” in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2002, pp. 133–142.

[67] J. Carbonell and J. Goldstein, “The use of mmr, diversity-based reranking for reordering documents and producing summaries,” in *SIGIR ’98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA: ACM, August 1998, pp. 335–336.

[68] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[69] S. Belongie, J. Malik, and J. Puzicha, “Shape context: A new descriptor for shape matching and object recognition,” in *Advances in Neural Information Processing Systems*, November 2000, pp. 831–837.

[70] A. Bosch, A. Zisserman, and X. Munoz, “Representing shape with a spatial pyramid kernel,” *ACM International Conference on Image and Video Retrieval*, pp. 401–408, July 2007.

[71] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester, “Discriminatively trained deformable part models, release 4,” http://people.cs.uchicago.edu/ pff/latent-release4/.

[72] A. Rahimi and B. Recht, “Random features for large-scale kernel machines,” in *Advances in Neural Information Processing Systems*, December 2007.

[73] F. Li, C. Ionescu, and C. Sminchisescu, “Random Fourier approximations for skewed multiplicative histogram kernels,” in *Lecture Notes for Computer Science (DAGM)*, September 2010, dAGM paper prize.

[74] M. H. C. S. J. Ponce and A. Zisserman, *Toward category-level object recognition*. Springer, 2006, vol. 4170.

[75] B. S. S. Dickinson, A. Leonardis and M. Tarr, *Object Categorization: Computer and Human Vision Perspectives*. Cambridge University Press, 2009.

[76] A. Vedaldi and B. Fulkerson, “VLFeat: An open and portable library of computer vision algorithms,” http://www.vlfeat.org/, 2008.