To appear, Proceedings of the Seventh IEEE Computer Society Workshop on Perceptual Organization in Computer Vision (POCV), June 2010.

Top-Down Pairwise Potentials for Piecing Together Multi-Class Segmentation Puzzles

Sudheendra Vijayanarasimhan and Kristen Grauman
University of Texas at Austin
{svnaras,grauman}@cs.utexas.edu

Abstract

Top-down class-specific knowledge is crucial for accurate image segmentation, as low-level color and texture cues alone are insufficient to identify true object boundaries. However, existing methods such as conditional random field models (CRFs) generally impose the class-specific knowledge only at the "node" level, evaluating class membership probabilities at the (super)pixels that define the random field graph. We introduce a strategy for pairwise *potential functions that capture top-down information, where we prefer to assign the same label to adjacent regions when the entropy reduction that would result from their merging is high. By measuring how the certainty of the object-level classifiers changes when considering the appearance description extracted from adjacent regions, we can "piece together" objects whose heterogenous texture would prevent both the too-local node potentials and conventional bottom-up smoothness terms from recognizing the object. We show how this idea can be used as either an affinity function for agglomerative clustering, or a pairwise potential for a CRF model. Experiments with two datasets show that the proposed entropy-guided affinity function has a clear positive impact on multi-class segmentation.*

1. Introduction

Segmentation and recognition are fundamental vision problems, and recent work shows that treating them in a unified way can significantly improve performance over methods that tackle either component alone. Whereas bottom-up techniques that rely solely on low-level cues such as texture or color are limited to finding regions with fairly homogeneous appearance, methods that also leverage "top-down" knowledge from category models can guide the grouping towards regions that best support object detection. In particular, techniques are available to compute either a class-specific foreground-background segmentation [1, 2, 3, 4, 5, 6, 7], or to label all pixels according to a set of multiple known categories [8, 9, 10, 11, 12, 13, 14]. The key insight of such approaches is that achieving object-level boundaries requires expressing an objective for segment quality that includes terms favoring category-specific attributes, not just pixel similarity.

However, existing methods generally incorporate category-specific attributes by classifying a single base "unit" (pixel, superpixel, region)[1], whereas interactions between neighboring units are restricted to capturing bottom-up cues. For example, a standard conditional random field model (CRF) specifies *node* potentials that prefer label assignments that agree with the object classifiers' posterior probabilities as computed for the pixels in that unit, whereas the *pairwise* potentials a priori prefer smooth label assignments for neighbors if their texture/color are similar, or if they lack a strong intervening edge. Similarly, techniques based on multiple-segmentations evaluate class models only within the spatial extent of whatever regions exist in the pool of candidates.

There are several drawbacks in restricting the top-down cues' influence to individual units. First, a unit may give too local of a view for a classifier to realize actual agreement with its class-specific model, particularly for objects with heterogenous texture patterns: imagine looking at a small superpixel patch on the side of a boat, and trying to decide if it is more likely a boat, or car, or some other man-made structure. Secondly, a pairwise potential function that only uses bottom-up cues will resist giving neighboring units the same label the more distinct their textures are (see Figure 1). Thirdly, limiting class-specific terms to whatever pool of units a bottom-up process provides means there is a risk of missing the combination of units at which some object's classifier would strongly respond. Widening the pool of candidate region scales can help (e.g. [15, 16, 17, 18]), but one pays a computational price.

To address these shortcomings, we propose a pairwise potential function that assesses top-down information within the full spatial extent of any neighboring units. The

[1] We use "units" as a generic term to refer to the basic tokens with which a segmentation method operates. In general, a unit is a region of pixels that some bottom-up process provides for further processing. In a CRF, these are the "nodes", which correspond to pixels or superpixels. We also use the terms "affinity" and "pairwise potential" interchangeably.

main idea is to prefer assigning the same label to adjacent regions that, once merged into a single region, would produce a reduction in entropy according to the learned top-down model classifiers. By measuring how the certainty of the object-level classifiers changes when considering the appearance description extracted from adjacent regions, we can "piece together" objects whose heterogenous texture thwarts conventional measures.

We designate both the feature types and classifiers that are appropriate for the proposed use of entropy reduction, meaning that even for merges between local pieces of an object we will be able to detect greater confidence. Essentially, this requires that larger parts of an object be closer in feature space to the whole than the smaller parts. Our method first builds classifiers for any known categories of interest using the ground truth (full object) spatial extent. Given a novel image, we score the initial individual regions based on their entropy under the classifiers, and also consider the reduction in entropy that would result once adjacent pairs are merged together (assigned the same label). Importantly, a candidate merging adjusts the feature by extracting it from the larger region, thereby giving a new view of the pixels, and affecting the classifier response.

We provide two different frameworks where the affinity is of use, and analyze the tradeoffs. In the first, we use the entropy-guided affinities to agglomeratively group segments, updating the affinity and features each time two segments are merged. This can be seen as a greedy procedure for decreasing the overall uncertainty in the segmentation of the image. To avoid getting stuck in local optima, we use multiple initialization points (multiple segmentations) and combine the results. In the second, we use the affinity as a pairwise potential in a CRF, encouraging neighboring segments that produce high reductions in entropy to be given the same label. In either case, including our pairwise affinity offers some robustness to what are initially myopic appearance features within the over-segmented regions. Furthermore, since the adjacency and scale of regions adjust dynamically as the agglomerative framework proceeds, it can capture regions with a larger spatial extent—a potential advantage, since the regions might support a particular object's presence only once joined together.

Our main contribution is a novel affinity function between segments that encourages grouping those which can be classified with more certainty once they are joined. We validate the idea within the agglomerative and CRF frameworks, and demonstrate its impact relative to traditional models with two benchmark datasets.

2. Related Work

Bottom-up segmentation methods (e.g. [19, 20]) group low-level cues such as texture, color, or contour continuity, without using any external knowledge about the objects occurring in the image. Some work shows how to learn optimal combinations of cues based on human perceptual judgments to provide robust boundary detection [21].

A number of top-down class-specific segmentation approaches have been proposed in recent years, where the low-level cues are balanced concurrently with insight from trained object models. One set of methods deals with foreground segmentation, where the best figure/ground assignment will have boundaries that agree with a previously learned shape model or other class-based cue [1, 6, 2, 22, 3, 4, 5, 7]. Such two-class techniques are intended for single-object images with a given target class, and generally work best when it is possible to construct a consistent shape prior for that category (e.g., side views of horses). Some explore the space of groupings using merge operations [22, 2], which we also consider here, though with a distinct objective function. The approach of [22] does not use any class-specific knowledge but instead learns to differentiate between "good" and "bad" moves among merges, shifts, and splits between superpixels using low-level cues and hand-drawn segmentations.

A second set of methods handle multi-class segmentation (also referred to as "image labeling"), where images containing objects from different categories are concurrently recognized and segmented [8, 9, 10, 23, 11, 12, 13]. Many successful approaches are based on conditional random field (CRF) models, which yield a pixel-level output that maximizes the probability of the joint label assignment. Such methods usually use appearance smoothness and intervening contour information as the pairwise terms, to smooth the segmentation over local regions. However, this approach only captures smoothness interactions between the base units at a single scale.

Several extensions to the random field approach capture interactions at multiple scales by using hierarchical representations [8, 18, 24], or by extracting features from the neighborhoods of the base superpixels [25]. Our affinity function also attempts to consider wider spatial extents for labeling, though in contrast to existing methods it does so with a novel entropy-guided measure; furthermore, within an iterative agglomerative grouping framework, our method considers region combinations that cannot be considered by CRF models without possibly exorbitant node connections between hierarchy levels (i.e., not just parent-child connections). This is in contrast to [25], where node features are always extracted from a predetermined fixed neighborhood of the base superpixel. Recent models have also shown how to incorporate information about co-occurring objects and their spatial layouts [12, 11].

The authors of [26] design a discriminative pairwise potential for the CRF model, where the features from two regions are concatenated and then classified as indicating "same" or "different" labels. However, the intent in that approach is to moderate the smoothness constraint when two regions are different in appearance, so while the potential

does incorporate a trained function, its meaning is closer to other pairwise functions that promote smoothness.

An alternative approach falling somewhere in between the bottom-up and concurrent methods is to compute multiple low-level segmentations with varying parameters, and then essentially look for good segments that are most consistent based on extracted model parameters [15, 10, 17, 16]. In a sense, our approach focuses this search by leveraging the class-based cues with candidate merged neighboring segments. The idea of expanding the region pool specifically to adjacent regions is explored in [16]; in contrast, we are proposing to automatically select among adjacent merges according to learned class-based information.

The proposed top-down pairwise potential function provides a simple and efficient way to integrate fine-to-coarse spatial support for objects of interest. We show that even when starting from an initial over-segmentation that alone would yield weak classification, our entropy-guided merges can reliably piece together same-class regions.

3. Approach

Given a test image, we start by preprocessing the image into a large number of superpixels, which are coherent local regions that preserve most of the object boundaries [22]. A set of classifiers provides top-down cues about the content of the current regions. We would like to prioritize *merges* (in the case of the agglomerative variant) or *same-label assignments* (in the case of the CRF variant) for the segments based on how great of a reduction in uncertainty they yield when together, as well as their consistency under a collection of Gestalt-inspired low-level cues.

3.1. Pairwise Affinities for Image Regions

In the following, we first describe classifier construction for the known categories of interest, then outline how pairs of regions are compared based on entropy reduction, and then explain the other low-level cues we incorporate.

3.1.1 Top-down Entropy-Guided Region Affinities

The basic idea behind our top-down affinity is that we would like to encourage the final segmentation of the objects to be as close to the examples in the training images as possible. When starting with units that were produced by a bottom-up segmentation algorithm, we do not expect the initial regions to capture the complete spatial extent of the objects. Therefore, we derive a pairwise potential that encourages segments to be merged if they look more like the true objects as seen in the correctly segmented training images. This idea is the key contribution of the paper.

In order for this to work, the features need to be such that "the whole is greater (closer to a trained model) than the sum of the individual parts". In other words, we require the feature representations of the larger merged region of segments, say S_i, S_j, to be closer in feature space to the complete object than when they are separate. Intuitively, histogram-based features have this property; measuring the histogram from a joined larger region of an object (i.e., summing the component histograms) makes it move closer to the full object's histogram, if the two component regions belong to that class.

Therefore, for whatever categories of interest are specified during training, we learn classifiers based on histogram representations to provide the top-down cue for segmentation. The training data consists of labeled, segmented exemplars for each of N classes. We extract histogram-based features to represent each in-class region—a texton histogram, color histogram, pyramid of histograms of oriented gradients [27], and a context descriptor based on the texton histograms of all regions other than the object. (See Sec. 4.)

We compute a combined kernel by averaging individual kernels that compare each feature type. Given the combined kernel, we learn a multi-class kernel-based classifier using the *probabilistic K-nearest neighbor* (PKNN) classifier developed in [28]. Importantly, we chose a nearest neighbor classifier because the changes in kernel values/distances in feature space directly propagate into the final classification probabilities, unlike margin-based approaches. Hence, the entropy score of the classification probabilities will reflect whether the merge brings the region closer to the feature representation of the whole object.

Given a new image and its current region segmentation (initially simply the oversegmented superpixels), we apply the classifier to each region, and obtain the list of posterior probabilities for each class. Thus for a given segmentation $\mathcal{S} = \{S_1, \ldots, S_M\}$ consisting of M regions, for each region S_j, we compute N probabilities: $P(1|S_j), \ldots, P(N|S_j)$, where $P(l|S_j)$ denotes the probability that the region S_j belongs to class l. These values compactly capture the top-down uncertainty about the given region's content, which we show how to exploit via a novel region-region affinity function.

Note that although the novel test images will initially be oversegmented, we train the classifiers using true object segmentations. The idea is that during grouping, we want the most confident classifier responses to occur once a region covers nearly the full spatial extent of an object.

Next we describe how to encourage merges or same-label assignments between adjacent regions that will reduce entropy once joined together. Our approach uses the classifier confidence probabilities outlined above, and is motivated by the following objective. Given an image, we ideally want to select a disjoint set of regions, one per object, so as to minimize the overall uncertainty in all regions. The uncertainty is measured by the entropy computed using each region's posterior probabilities for the N known classifiers, weighted by the region size.

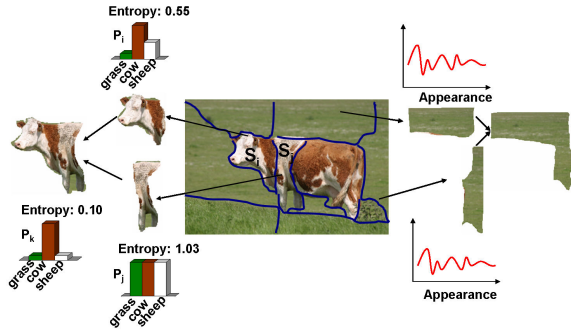

Figure 1. Our entropy-guided affinity function favors merges between adjacent regions that will reduce classifier uncertainty once joined. Here the two 'cow' regions, S_i, S_j, are not confidently classified by any category model, nor would their appearance features strongly agree. However, if merged, the more complete region would more confidently respond to the cow classifier. Our method uses top-down knowledge to piece together objects from the bottom-up, as it identifies adjacent regions that when together reduce uncertainty or agree in appearance.

Thus the optimal segmentation $\mathcal{S}^*$ is defined as:

$$\mathcal{S}^* = \arg\min_{\mathcal{S}} U(\mathcal{S}),$$
$$\text{where } U(\mathcal{S}) = \sum_{j=1}^{|\mathcal{S}|}\sum_{l=1}^{N} -P(l|S_j)\log P(l|S_j)\,|S_j|. \quad (1)$$

Here, $U(\mathcal{S})$ denotes the *pixel-level* entropy under the segmentation $\mathcal{S}$, $|S_j|$ denotes the number of pixels in S_j, and $|\mathcal{S}|$ denotes the number of regions in the segmentation.

To obtain the segmentation that is least uncertain, we need to identify the best pixel grouping among all possible segmentations. The optimization is intractable due to the exponential number of possible segmentations. Instead, we define a local affinity function between neighboring regions such that the affinity is high for segments that produce reductions in the objective in Equation 1.

The reduction in entropy to segmentation $\mathcal{S}$ due to a candidate merge between two regions S_i and S_j is given by:

$$\begin{aligned} R(S_i,S_j) &= U(\mathcal{S}) - U(\mathcal{S}\cup\{S_k\}\setminus\{S_i,S_j\}) \\ &= \left(\sum_{l=1}^{N} P(l|S_k)\log P(l|S_k)\right)(|S_i|+|S_j|) \\ &\quad - \left(\sum_{l=1}^{N} P(l|S_i)\log P(l|S_i)\right)|S_i| \\ &\quad - \left(\sum_{l=1}^{N} P(l|S_j)\log P(l|S_j)\right)|S_j|, \end{aligned}$$

where S_k denotes the merging of the inputs: $S_k = \{S_i \cup S_j\}$. In other words, since the segments are disjoint in space, the impact of merging the two inputs on total uncertainty is simply computed by measuring the change in the entropy after the proposed merge, with terms weighted according to segment sizes. Using this we can define the entropy-guided affinity function $A_u(S_i,S_j)$ as follows:

$$A_u(S_i,S_j) = \begin{cases} R(S_i,S_j), & \text{if } (S_i,S_j)\in\mathcal{A} \\ 0, & \text{otherwise} \end{cases}, \quad (2)$$

where $\mathcal{A}$ denotes the set of adjacent pairs of segments. This value is high only if by merging the regions we see something that "looks more like" a known object according to our classifiers (see Figure 1). The pairwise term we have defined can be loosely viewed as if we are putting together a jigsaw puzzle: pieces that once joined seem to indicate (at least a part of) some familiar object look promising. The affinity A_u can be used to greedily minimize the objective in (1) when we consider an agglomerative grouping strategy (Section 3.2.1). In the case of a CRF, it encourages neighboring segments that are better together to be given the same label (Section 3.2.2).

3.1.2 Bottom-Up Appearance Cues

Alongside the classifier confidence, we also include usual appearance cues in the pairwise affinities, motivated by three factors. First, images we segment may also contain objects unfamiliar to the classifiers. In this case, using only the change in entropy could lead to unwanted merges between the segments of unseen objects. At the same time, since some objects have rather homogenous appearance (e.g., grass, sky, water), a local patch alone could satisfy the classifier; the entropy-based scoring would resist merging such regions for familiar objects, given that their entropy would remain level once merged. Third, we could benefit from additional information that is not relevant to the region-based classifier (e.g., cues from boundaries cutting *between* regions). Therefore, using the features described in [22] as inspiration, we also consider grouping cues based on classic Gestalt principles:

Similarity. We measure the similarity between two regions based on their color features. The color affinity is defined in terms of two segments' color histograms: $A_s(S_i,S_j) = \exp(-\frac{1}{2\sigma_c}\|C_i - C_j\|^2)$, where C_i and C_j are the two histogram vectors, and σ_c is set to the average L_2 distance between the color histograms of the segments from training images. Similar functions could of course be added based on additional features, such as texture, brightness, etc.

Contour Energy. We measure the energy along the boundary of a segment pair using the boundary detector of [21] and convert it into an affinity as $A_e(S_i,S_j) = 1 - \sum_{p\in S_i\cap S_j}\frac{P_b(p)}{|S_i\cap S_j|}$ where $P_b(p)$ is the "probability of boundary" at pixel p, as defined in [21].

3.2. Grouping Strategy

The affinities defined in the previous section can be used alone within any clustering algorithm to group regions, or can be taken as a pairwise potential between the nodes of a CRF defined on superpixels. We explore both approaches, as each offers different advantages.

An agglomerative procedure that sequentially merges adjacent regions specifically accounts for the fact that as more regions are combined, their uncertainty and adjacency neighborhood continue to change. The procedure converges to a local optimum of the objective in Equation 1; we run the approach with multiple initializations (multiple segmentations obtained with different parameters) and combine the outputs of each run to avoid local maxima and produce better segmentations. On the other hand, if the affinity is used as a pairwise term in a conditional random field, it can help favor the correct labeling of non-homogenous segments of an object in cases where bottom-up smoothness cues would be misleading. In the following sections we explain each variant in detail.

3.2.1 Agglomerative Grouping Model

For the agglomerative procedure, we train a logistic regression classifier based on the top-down uncertainty-guided cue and the two low-level cues. In this way we learn the weights to associate with each component affinity, and can compute the probability that two regions should be merged:

$$P(\text{merge}|S_i, S_j) = \frac{1}{1 + e^{-\left(\sum_f w_f A_f(S_i,S_j)+b\right)}}, \quad (3)$$

where each A_f is an affinity function defined above (A_u,A_s,A_e). We learn the parameters w_f, b by maximizing the likelihood on the training examples using iterated reweighted least squares.

At each iteration of the agglomerative grouping, we compute the probability of merges between every pair of neighboring segments, and choose the pair with the largest value. Once we merge a pair of segments, we recompute the neighborhood graphs and the cues, and repeat the process until entropy stops decreasing. At that point, we return the final segmentation, together with the uncertainty associated with each region. Note that each merge is a greedy step towards minimizing the objective in Equation 1, and that as the regions expand we have candidate merges that potentially extend across much farther distances within the image than the initial superpixels. Thus, whereas the CRF variant below will need to select a single granularity of "sites" to be considered for grouping (e.g., a fixed number of superpixels), the agglomerative variant iteratively identifies segments that can be merged in order to form new (larger) sites with improved classifier confidence. This is a potential advantage, as our experiments confirm.

The method is straightforward to implement and efficient; at each step we need to compute affinities only for adjacent pairs, and following a merge operation we need only to update features and scores corresponding to one row of the affinity matrix.

The agglomerative merging process can loosely be viewed as if the algorithm is putting together a jigsaw puzzle for which it has not seen the box cover: pieces that seem consistent just based on appearance look promising, but so do pieces that once joined seem to indicate some familiar object. As with piecing together a puzzle, the information is quite local at first, but then as we attach larger and larger regions, the confidence about parts of the scene stabilizes.

3.2.2 Conditional Random Field Model

In the second variant, we use our affinity inside the pairwise potential of a conditional random field model to encourage similar labels on segments that are better classified when considered together. Let c_i denote a class label for segment S_i. We define the conditional probability of the labeling L given an image I and a set of segments $\{S_i\}$ as

$$\begin{aligned} \log\ P(L|I;\theta) &= \sum_i \log\ P(c_i|S_i,\theta_p) \\ &+ \sum_{(S_i,S_j)\in\mathcal{A}} \phi(c_i,c_j;\theta_\phi) - \log\ Z(\theta,I), \end{aligned}$$

where $P(c_i|S_i,\theta_p)$ refers to the probability of classifying segment S_i as c_i, as output by the PKNN classifier, and $Z(\theta, I)$ is the partition function.

The pairwise edge potentials are defined based on our pairwise affinities as follows:

$$\phi(c_i,c_j;\theta_\phi) = \delta(c_i \neq c_j) \sum_f \theta_\phi^f A_f(S_i,S_j), \quad (4)$$

where each A_f is an affinity function defined in the previous sections (A_u,A_s,A_e), and δ is the δ-function. We design the edge potential $\phi(\cdot)$ to incur a high cost on a labeling where the affinities are high, but the neighbors are given different labels. Therefore, similar to the agglomerative grouping strategy, A_u's contribution would encourage neighboring segments that are classified better together to be given the same label. Furthermore, since the reduction in the entropy can also be negative, it encourages distinct labels for segments that do *not* fit together.

Unlike the agglomerative procedures, the CRF works at a single scale (fixed base units); however, the effects can still propagate in a neighborhood through the pairwise interactions. Additionally, there are several efficient algorithms for optimizing the above energy function.

4. Results

The main goal of our experiments is to analyze the usefulness of the proposed entropy-based pairwise affinity in both the grouping strategies explained above.

	Per-pixel	Per-class	building	grass	tree	cow	sheep	sky	aeroplane	water	face	car	bicycle	flower	sign	bird	book	chair	road	cat	dog	body	boat
Without A_u	70.6	55.0	62.9	97.2	79.3	51.5	48.9	87.9	**69.7**	**68.1**	15.2	42.7	**81.5**	53.5	**57.6**	22.5	85.8	33.7	69.1	49.6	34.4	36.4	8.4
With A_u (ours)	**76.2**	**62.9**	**69.9**	**98.3**	**85.9**	**61.6**	**64.5**	**91.3**	52.7	67.2	**56.1**	**64.7**	74.6	**56.8**	56.6	22.7	**86.4**	**53.7**	**78.4**	**66.6**	**46.3**	**53.9**	**12.5**
Upper bound	85.1	75.4	82.5	98.9	91.9	73.9	68.9	97.0	66.9	75.0	88.5	91.5	79.6	67.9	69.5	46.3	84.8	70.4	94.5	74.0	58.9	75.8	26.7

Table 1. Results on the MSRC v2 dataset using the agglomerative grouping strategy.

	Per-pixel	Per-class	building	grass	tree	cow	sheep	sky	car	bicycle	flower	sign	road	leaf	chimney	door	window
Without A_u	69.6	60.4	80.8	79.2	64.0	68.3	78.5	91.3	65.9	64.9	49.7	49.8	30.5	**66.4**	33.0	32.9	50.2
With A_u (ours)	**76.8**	**68.2**	**87.1**	**80.3**	**71.0**	**74.1**	**81.7**	**93.0**	**84.3**	**72.8**	**60.0**	**55.1**	**61.7**	62.5	**44.6**	**36.6**	**57.9**

Table 2. Results on the MSRC v0 dataset using the agglomerative grouping strategy.

Datasets: We evaluate our approach with the **MSRC v2** and **MSRV v0** datasets. We chose these two because they have ground truth pixel labels (needed for training and evaluation) and multi-label images. The MSRC contains 21 classes in 591 images, and is commonly used as a benchmark by multi-class segmentation methods. We follow the standard test/train breakdown as given in [9].

We obtained ground truth object outlines for a set of 15 categories in the MSRC v0 dataset. The dataset contains 3,259 images and the following categories: {building, grass, tree, cow, sheep, sky, car, bicycle, flower, sign, road, leaf, chimney, door, window}. The categories were chosen based on the criterion that each have at least 150 image examples. We use a four-to-one train/test split.

Implementation details: We initialize our methods with two different oversegmentations; about 50-60 superpixels computed with Normalized Cuts [20], which forms small to medium regions of about equal size, and a Mean-Shift [29] segmentation with kernel parameters (12,15), which forms more variable-sized regions. We run the agglomerative strategy on both segmentations and obtain the final result by taking the pixel-wise product of the probabilities output on each segmentation. We do this to mitigate the influence of local maxima, and we found that this does produce some gain in the overall segmentation accuracy.

We use texton histograms with 18 filters and 120-d color histograms as features for the PKNN classifier. In addition, we compute pHOG histograms [27] for every training object by masking out the rest of the objects in order to capture both the overall shape of an object and the scene layout. Finally, we obtain a context descriptor for every object/segment by masking out the object/segment and accumulating a texton histogram of the rest of the image. We found that the pHOG and context descriptors provide a boost for structured classes such as "chair" and other categories which have very few training images. We use χ^2 RBF kernels. We set the weights on the pairwise affinities for the CRF (θ_ϕ) equally, although one could potentially learn them from training data.

In the following subsections we evaluate the impact of the proposed entropy-guided affinity in each of the two possible grouping strategies.

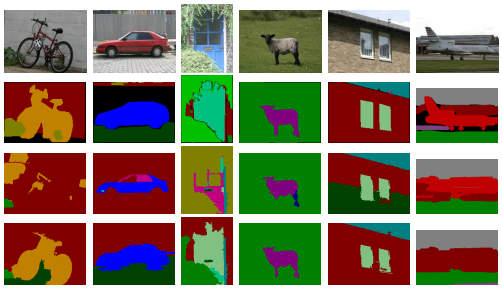

Figure 2. Representative outputs when using the agglomerative grouping variant. **First row:** input image; **Second row:** ground truth segmentation (colors denote labels); **Third row:** result without our affinity; **Fourth row** result when including our top-down affinity to score merges. Our entropy-guided merging correctly encourages merges between non-homogenous segments in cases where the purely bottom-up affinity fails. (Best viewed in color).

4.1. Agglomerative Grouping Results

We run our agglomerative grouping strategy in two modes; (1) with the logistic regression classifier trained on all three pairwise affinities (denoted "With A_u" in the figures), and (2) with only the bottom-up cues (denoted "Without A_u"). All other settings (including the classifier) are exactly the same. Note that the "Without A_u" baseline *does* include the bottom-up pairwise terms A_e and A_s, and so we are directly isolating the impact of our approach.

Table 1 compares the classification accuracy in either setting for the MSRC v2 dataset. The baseline performs quite well on many of the categories, with a per-pixel average accuracy of 70.6% and category-wise mean of 55.0%. This result suggests that for a number of categories such as "grass", "tree", "water", and "bicycle", low-level cues are quite sufficient for obtaining good segmentations. However, adding our entropy-based affinity improves the per-pixel average to 76.2% and the category-wise mean to 62.9%. This shows the dramatic impact of our entropy-guided merging and illustrates our method's ability to continually find wider-scale regions that better support the known classes, even starting with very local appearance. Table 2 shows the classification accuracy on the larger MSRC v0 dataset. We see a similar trend as with the v2.

Figure 2 shows some representative example segmentations where our top-down pairwise potential improves over

	Per-pixel			Per-class		
	Mean shift	Ncuts	Combined	Mean shift	Ncuts	Combined
Without A_u	59.8	69.6	70.6	44.0	54.9	55.0
With A_u (ours)	71.9	72.8	76.2	57.4	60.5	62.9

Table 3. Impact of using multiple initializations.

the same model using only the bottom-up affinities. Note that although the last two rows use different pairwise potentials for defining the merges, they use the *same* classifier (PKNN) to classify the image. As seen in several examples in the figure, our entropy-guided merging is able to encourage merges between non-homogenous segments on which bottom-up cues fail. For example, the wheels and windows of the car in the second column have different colors than the body of the car, and hence a bottom-up affinity does not merge the segments correctly. Similarly, bottom-up cues fail to merge the legs of the sheep with its body in the fourth column, and the shaded part of the building in the fifth column. The last column shows a failure case by our method, where the top-down affinity provides the wrong signal, merging airplane regions with buildings.

In comparison to the state-of-the-art on the MSRC v2, our approach produces very good results compared to existing *region*-based approaches. The authors of [11] report a per-pixel average of 76.5% using a region-based CRF, [10] report 75.1%, and [30] report 76.4%, whereas we obtain 76.2% with the agglomerative grouping and entropy-guided merges. Using *pixel*-level classifiers and hierarchical random fields, [18] and [24] report accuracies of 86% and 81%, respectively. The bottom row of Table 1 shows the accuracy attainable (85.1%) were we to use our classifiers to label the ground-truth segmentations on the test images, indicating the approximate upper bound on performance we could achieve if computing perfect merges. This upper bound helps separate the effects of our grouping procedure from the classifier/feature choices.

We found that the accuracy of the agglomerative procedure is quite insensitive to the choice of the number of superpixels in the initial segmenatation. When varying the number of superpixels between (50, 100, 200) the per-pixel accuracy on the MSRC v2 ranged from 75.2 to 76.2 when using A_u, and from 68.5 to 70.6 when not using A_u.

Table 3 shows the impact of using multiple initializations. The table reports the accuracies obtained by each of the two multiple segmentations used and the final combined segmentation. Using our top-down affinity improves all three results. Combining the two segmentations produces a large improvement when using the top-down cue, but is less noticeable for low-level cues alone. This mitigates the effect of the local maxima, as described above.

4.2. CRF Labeling Results

For the CRF variant we use the probabilities output by our PKNN classifier directly as unary potentials, and the pairwise affinities as defined in Equation 4. We again run

Figure 3. Representative outputs when using the CRF variant. **First row:** input image; **Second row:** ground truth segmentation and labeling (colors denote labels); **Third row:** result when using only bottom-up paired potentials with the top-down unary potentials; **Fourth row:** result when adding our top-down pairwise potentials. (Best viewed in color.)

the CRF model in two modes to see the impact of our top-down pairwise affinity; (1) with all three pairwise affinities (With A_u), and (2) with only the bottom-up pairwise potentials (Without A_u). All other settings including the classifier are exactly the same. We stress that while the baseline runs inference on the CRF without *pairwise* top-down potentials, both our method and the baseline are using the usual *unary* top-down potentials (i.e., the classifier responses evaluated per individual node). What our method adds is the proposed pairwise entropy-guided potential. Thus the baseline is meant to represent the way current CRF-based approaches use top-down information—restricting the pairwise potentials to measure bottom-up cues, like smoothness of the label assignment and intervening contours.

Table 4 shows the classification accuracy on the MSRC v2 dataset. Including our top-down pairwise potential again produces better accuracies for both the pixel-level and the category-wise averages. Looking more closely at the individual categories (see third row), we see that our method has the best impact for categories such as building, car, or chair—which are less homogenous in appearance than "stuff" classes such as grass, sky, or water, and therefore more difficult to classify if viewed too locally. For the latter, our affinity usually neither helps nor hurts the performance significantly, which is intuitive, since these objects have rather homogenous appearance, and a local patch alone could satisfy the classifier via the unary potential. The entropy-based scoring might resist merging such regions for familiar objects, given that their entropy would remain level once merged. An interesting future extension would be to learn to weight the impact of each pairwise potential (θ_ϕ) for different classes (or at least "stuff" vs. "things") in order to mitigate this effect.

The lower absolute performance when using our affinity within the CRF (as compared to others' reported results) may be due to the particular feature representation used in this experiment, which we chose purposefully to keep things parallel with the above agglomerative test. In fact, when we run the experiment using the state-of-the-art

	Per-pixel	Per-class	building	grass	tree	cow	sheep	sky	aeroplane	water	face	car	bicycle	flower	sign	bird	book	chair	road	cat	dog	body	boat
Without A_u	58.8	43.0	21.5	**96.5**	**81.4**	12.8	**41.3**	**94.1**	7.0	18.3	**89.9**	30.3	47.3	64.9	**36.1**	**22.9**	35.4	19.0	**89.1**	21.8	**4.9**	**49.2**	**19.1**
With A_u (ours)	**60.4**	**47.1**	**33.5**	94.3	79.1	**25.5**	34.0	87.1	**21.8**	**20.5**	88.7	**53.6**	**59.8**	**77.7**	28.7	21.5	**35.8**	**37.6**	80.9	**42.2**	4.8	42.9	18.1
% improvement			**+56%**	-2%	-2%	**+99%**	-18%	-7%	**+211%**	**+12%**	-1%	**+77%**	**+26%**	**+20%**	-20%	-6%	**+1%**	**+98%**	-9%	**+94%**	-2%	-13%	-5%

Table 4. Results on the MSRC v2 dataset using a CRF model.

boosted features provided by [18], we find that using our top-down potential produces a slight increase in the overall accuracy (81.3% vs. 80.7%).

Figure 3 shows some representative example segmentations where our top-down pairwise potential improves over the purely bottom-up pairwise potential in the CRF variant. In this case our affinity seems to also prevent segments of *different* classes from being given the same label. For example, in the example in the leftmost column, bottom-up pairwise potentials prefer allowing the sign and the building in the background to have the same label, since they have very similar appearance. However, our top-down affinity discourages such labelings (possibly, with the help of the pHOG features) and provides a better result. A similar effect can be seen in columns two, three, and four, where (grass, cow regions), (chair, ground regions), (road, and building regions), respectively, are assigned the same label by the strictly bottom-up paired potential.

When varying the number of superpixels between (50, 100, 200) the per-pixel accuracy on the MSRC v2 ranged from 57.1 to 60.4 when using our method, and from 56.7 to 58.8 when not using A_u. The improvement in the accuracy over $without\ A_u$ dropped by 25% for 200 superpixels over 50 superpixels. This is understandable given that merging very small-scale regions might not produce significant reductions in the entropy, preventing our method from having as much impact.

5. Conclusions

We introduced a novel affinity function between segments that encourages grouping those which can be classified with more certainty once they are joined. We demonstrated the impact of the proposed top-down pairwise affinity within both an agglomerative procedure and a CRF. Results on two datasets show that our method notably improves traditional paired terms that strictly capture bottom-up information.

Acknowledgements. Many thanks to Lubor Ladicky for providing MSRC features and the Stanford group for the STAIR vision library. This research is supported in part by Microsoft Research and the Henry Luce Foundation.

References

[1] E. Borenstein and S. Ullman. Class-Specific, Top-Down Segmentation. In *ECCV*, 2002.

[2] L. Liu and S. Sclaroff. Region Segmentation via Deformable Model-Guided Split and Merge. In *ICCV*, 2001.

[3] B. Leibe, A. Leonardis, and B. Schiele. Combined Object Categorization and Segmentation with an Implicit Shape Model. In *Workshop on Statistical Learning in Computer Vision*, 2004.

[4] J. Winn and N. Jojic. LOCUS: Learning Object Classes with Unsupervised Segmentation. In *ICCV*, 2005.

[5] M. Kumar, P. Torr, and A. Zisserman. OBJ CUT. In *CVPR*, 2005.

[6] S. Yu, R. Gross, and J. Shi. Concurrent Object Recognition and Segmentation by Graph Partitioning. In *NIPS*, 2002.

[7] T. Cour and J. Shi. Recognizing Objects by Piecing Together the Segmentation Puzzle. In *CVPR*, 2007.

[8] X. He, R. Zemel, and M. Carreira-Perpinan. Multi-scale Conditional Random Fields for Image Labeling. In *CVPR*, 2004.

[9] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint Appearance, Shape and Context Modeling for Multi-class Object Recognition and Segmentation. In *ECCV*, 2006.

[10] L. Yang, P. Meer, and D. Foran. Multiple Class Segmentation Using a Unified Framework over Mean-Shift Patches. In *CVPR*, 2007.

[11] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller. Multi-Class Segmentation with Relative Location Prior. *IJCV*, 2008.

[12] C. Galleguillos, A. Rabinovich, and S. Belongie. Object Categorization Using Co-occurrence, Location, and Appearance. In *CVPR*, 2008.

[13] P. Kohli, L. Ladicky, and P. Torr. Robust Higher Order Potentials for Enforcing Label Consistency. In *CVPR*, 2008.

[14] L. Zhu, Y. Chen, Y. Lin, C. Lin, and A. Yuille. Recursive Segmentation and Recognition Templates for 2d Parsing. In *NIPS*, 2008.

[15] B. Russell, A. Efros, J. Sivic, W. Freeman, and A. Zisserman. Using Multiple Segmentations to Discover Objects and their Extent in Image Collections. In *CVPR*, 2006.

[16] T. Malisiewicz and A. A. Efros. Improving Spatial Support for Objects via Multiple Segmentations. In *BMVC*, 2007.

[17] C. Pantofaru, C. Schmid, and M. Hebert. Object Recognition by Integrating Multiple Image Segmentations. In *ECCV*, 2008.

[18] L. Ladicky, C. Russell, and P. Kohli. Associative Hierarchical CRFs for Object Class Image Segmentation. In *ICCV*, 2009.

[19] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and Texture Analysis for Image Segmentation. *IJCV*, 43:7–27, 2001.

[20] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *TPAMI*, 2000.

[21] D. Martin, C. Fowlkes, and J. Malik. Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues. *TPAMI*, 2004.

[22] X. Ren and J. Malik. Learning a Classification Model for Segmentation. In *ICCV*, 2003.

[23] D. Batra, R. Sukthankar, and T. Chen. Learning Class-Specific Affinities for Image Labeling. In *CVPR*, 2008.

[24] L. Zhu, Y. Chen, Y. Lin, C. Lin, and A. Yuille. Recursive Segmentation and Recognition Templates for 2d Parsing. In *NIPS*, 2008.

[25] B. Fulkerson, A. Vedaldi, and S. Soatto. Class Segmentation and Object Localization with Superpixel Neighborhoods. In *ICCV*, 2009.

[26] S. Kumar and M. Hebert. Discriminative Random Fields: A Discriminative Framework for Contextual Interaction in Classification. In *ICCV*, 2003.

[27] A.Bosch, A.Zisserman, and X.Munoz. Representing Shape with a Spatial Pyramid Kernel. In *ACM CIVR*, 2007.

[28] P. Jain and A. Kapoor. Active Learning for Large Multi-class Problems. In *CVPR*, 2009.

[29] D. Comaniciu and P. Meer. Mean Shift: A Robust Approach Toward Feature Space Analysis. *PAMI*, 24:603–619, 2002.

[30] S. Gould, R. Fulton, and Daphne Koller. Decomposing a Scene into Geometric and Semantically Consistent Regions. In *ICCV*, 2009.