Int J Comput Vis
DOI 10.1007/s11263-008-0140-x

Multi-Class Segmentation with Relative Location Prior

Stephen Gould · Jim Rodgers · David Cohen · Gal Elidan · Daphne Koller

Received: 26 September 2007 / Accepted: 17 April 2008
© Springer Science+Business Media, LLC 2008

Abstract Multi-class image segmentation has made significant advances in recent years through the combination of local and global features. One important type of global feature is that of inter-class spatial relationships. For example, identifying "tree" pixels indicates that pixels above and to the sides are more likely to be "sky" whereas pixels below are more likely to be "grass." Incorporating such global information across the entire image and between all classes is a computational challenge as it is image-dependent, and hence, cannot be precomputed.

In this work we propose a method for capturing global information from inter-class spatial relationships and encoding it as a local feature. We employ a two-stage classification process to label all image pixels. First, we generate predictions which are used to compute a local relative location feature from learned relative location maps. In the second stage, we combine this with appearance-based features to provide a final segmentation. We compare our results to recent published results on several multi-class image segmentation databases and show that the incorporation of relative location information allows us to significantly outperform the current state-of-the-art.

Keywords Multi-class image segmentation · Segmentation · Relative location

1 Introduction

Partitioning or segmenting an entire image into distinct recognizable regions is a central challenge in computer vision which has received increasing attention in recent years. Unlike object recognition methods that aim to find a particular object (e.g., Winn and Shotton 2006; Opelt et al. 2006), multi-class image segmentation methods are aimed at concurrent multi-class object recognition and attempt to classify *all* pixels in an image (e.g., Schroff et al. 2006; Shotton et al. 2006; Yang et al. 2007).

Most multi-class segmentation methods achieve their goal by taking into account local (pixel or region) appearance signals along with a preference for smoothness, i.e., classifying visually-contiguous regions consistently. For multi-class image segmentation, this is often achieved by constructing a conditional Markov random field (CRF) (Lafferty et al. 2001; Pearl 1988) over the image that encodes local and pairwise probabilistic preferences. Optimizing the energy defined by this CRF is then equivalent to finding the most probable segmentation (e.g., Kumar et al. 2005; Shotton et al. 2006).

Some innovative works in recent years also employ global or contextual information for improving segmentation. Winn and Shotton (2006), for example, use close-to-medium range effects to impose a consistent layout of components recognized to be part of a known object; He et al. (2006) attempt to infer the environment (e.g., rural/suburban) and use an environment-specific class distribution prior to guide segmentation; Shotton et al. (2006) use an absolute location prior as a feature in their probabilistic construction.

In this paper, we improve on state-of-the-art multi-class image segmentation labeling techniques by using contextual information that captures spatial relationships between classes. For example, identifying which pixels in an image belong to the *tree* class provides strong evidence for pixels above and beside to be of class *sky* and ones below to be of class *grass*. One difficulty in using such global information

S. Gould (✉) · J. Rodgers · D. Cohen · G. Elidan · D. Koller
Department of Computer Science, Stanford University, Stanford, CA, USA
e-mail: sgould@stanford.edu

is that relative location preferences depend on pixel/region level predictions made at run-time and cannot be precomputed. On the other hand, incorporating complex global dependencies within the probabilistic segmentation model directly is computationally impractical. In this work, we propose a method for making use of relative location information while addressing this challenge.

We propose a two-stage prediction framework for multi-class image segmentation that leverages the relative location of the different object classes. We start by making a first-stage label prediction for each pixel using a boosted classifier trained on standard appearance-based features. We then combine this prediction with precomputed relative location maps—non-parametric probability representations of inter-class offset preferences—to form a relative location feature that is local. This feature is then incorporated into a unified model that combines appearance and relative location information to make a final prediction.

We show how our relative location feature can be incorporated within a general probabilistic framework for image segmentation that defines a joint distribution over contiguous image regions called *superpixels* (Ren and Malik 2003; Felzenszwalb and Huttenlocher 2004). Specifically, we incorporate this feature within two models: (i) a logistic-regression classifier that is applied independently to each superpixel and that facilitates efficient inference and learning; and (ii) a more expressive but more computationally expensive CRF that also includes pairwise affinity preferences between neighboring pixels/regions.

We demonstrate the effectiveness of our approach on the 9-class and 21-class MSRC image databases (Criminisi 2004) as well as the 7-class Corel and Sowerby databases (He et al. 2004), and show that our relative location feature allows us to improve on state-of-the-art results while using only standard baseline features. Importantly, we also show that using relative location allows even a simple logistic regression classifier to perform better than state-of-the-art methods and to compete with a computationally demanding CRF.

2 Related Work

There are many recent works on multi-class image segmentation that employ some kind of contextual information (e.g., He et al. 2004, 2006; Schroff et al. 2006; Shotton et al. 2006; Winn et al. 2005; Yang et al. 2007; Rabinovich et al. 2007; Shental et al. 2003; Kumar and Hebert 2005; Carbonetto et al. 2004). The simplest type of contextual information is in the form of a continuity preference for nearby pixels. This is commonly done via a conditional Markov random field that includes pairwise affinity potentials. Hierarchical models are less common, but also allow relationships between neighboring pixels/regions to be modeled (e.g., Adams and Williams 2003). Several authors also make use of additional contextual information, as discussed below.

He et al. (2006) encode global information based on an inferred scene context, which is used to condition local appearance based probability distributions for each class. Murphy et al. (2003) (along with several other works in the series) use a similar "gist" based approach in the context of object recognition, allowing the type of scene to focus attention to certain areas of the image when searching for a particular object. These methods infer a single scene context and do not allow, for example, the discovery of one class (or object) to influence the probability of finding others.

Other works extend this idea and do allow inter-class correlations to be exploited. Torralba et al. (2004) use a series of boosting rounds in which easier objects are found first, and local context used to help in the detection of harder objects during later rounds. The method is similar to Fink and Perona (2003) who propose a mutual boosting approach where appearance based predictions in one round become the weak learner for the next boosting round. Their approach allows, for example, 'nose' predictions to affect areas that appear to be an 'eye'. Here only the local neighborhood is considered at each round for incorporating contextual information. Furthermore, their method can only be applied in situations where correlation exists between detectable objects, and unlike our method does handle the relative location of background objects such as sky and grass.

An early example of incorporating spatial context for classifying image regions rather than detecting objects is the work of Carbonetto et al. (2004). They showed that spatial context helps classify regions when local appearance features are weak. Winn and Shotton (2006) also capture spatial context through asymmetric relationships between individual object parts, but only at the local level. Yang et al. (2007) propose a model that combines appearance over large contiguous regions with spatial information and a global shape prior. The shape prior provides local context for certain types of objects (e.g., cars and airplanes), but not for larger regions (e.g., sky and grass).

In contrast to these works (Torralba et al. 2004; Fink and Perona 2003; Carbonetto et al. 2004; Winn and Shotton 2006; Yang et al. 2007), we explicitly model spatial relationships between different classes, be they local or long range. That is, instead of inferring contextual information from image features, we directly model global context by learning the relative locations between classes from a set of labeled images. Notably, we model relationships between the class labels of the pixels themselves, rather than low-level appearance features. Furthermore, we use a two-stage inference process that allows us to encode this image dependent global information as a local feature, enabling us to use simpler probabilistic models.

He et al. (2004) encode longer range dependencies by applying pairwise relationships between parts of an image at multiple scales, allowing for additional coarse class-relationship modeling at the cost of a more complicated network. By capturing more than one class within a region, their multi-scale approach allows for prior geometric patterns to be encoded in much the same way that global location priors can be used over an entire image. The local, region and global features are combined multiplicatively into a single probabilistic model. Unlike their approach which models arrangements *of* classes, our method models relative location *between* classes. This allows us to generalize to geometric arrangements of classes not seen in the training data and better capture the interactions between multiple classes.

Recent work by Rabinovich et al. (2007) incorporates semantic context for object categorization. Their model constructs a conditional Markov random field over image regions that encodes co-occurrence preferences over pairwise classes. Our relative location maps can be thought of as extending this idea to include a spatial component.

Singhal et al. (2003) introduce a model for detecting background classes (materials) in an image. They also use a two-stage approach in which material detection algorithms first classify regions of the image independently. A second-stage Bayesian network then enforces a limited set of spatial constraints to improve classification accuracy of the material detectors. Their work does not consider any non-background inter-class relationships.

Most similar to us is the innovative work of Kumar and Hebert (2005) (and other works in the same series) who explicitly attempt to model inter-class relationships. Briefly, they suggest a first layer CRF with affinity potentials for initial pixel level segmentation, followed by a second layer for superpixel based segmentation that takes into account relations between classes. There are several important differences with respect to our work.

First, existing work considers only simple relative location relations (*above*, *beside*, or *enclosed*). Our method, on the other hand, relies on non-parametric relative location maps, allowing us to model complex spatial relationships, such as both *sky* and *car* are found above *road*, but *car* tends to be much closer than *sky*. This can be important when considering rich databases with many classes such as the 21-class MSRC database. Second, as mentioned above, our model encodes global information as local features enabling us to use much simpler probabilistic models, i.e., logistic regression, while still achieving state-of-the-art performance.

Finally, Kumar and Hebert (2005) propagate messages between all superpixels in the second stage of the model and thus rely on having a small number of superpixels (typically less than 20) to make inference tractable. This requires simple scenes with large contiguous regions of a single class

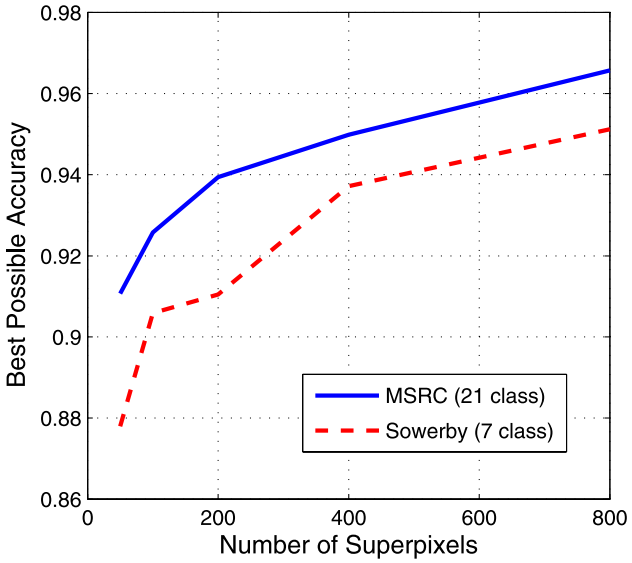
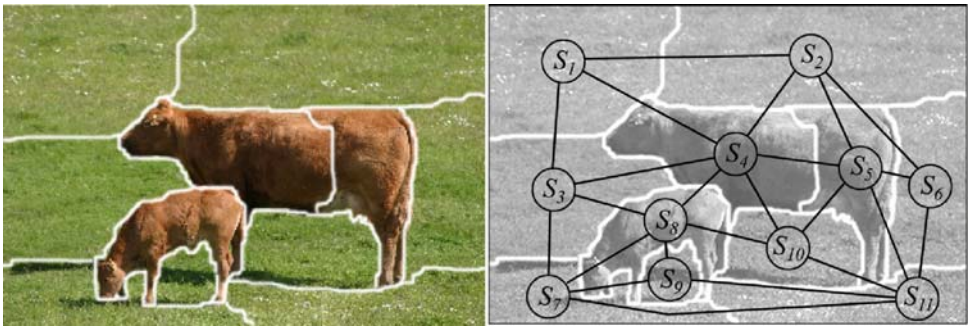


Fig. 1 Best possible accuracy (y-axis) given the constraint that all pixels in a superpixel are assigned the same label, as a function of the number of superpixels (x-axis). Results shown are an average over all images in the 21-class MSRC database (*solid*) and the 7-class Sowerby database (*dashed*)

or specialized segmentation methods. As scene complexity increases, a small number of superpixels cannot capture all the distinct regions. To quantitatively evaluate this, we segmented images from two different databases used the state-of-the-art over-segmentation method by Ren and Malik (2003). We then used ground-truth knowledge to assign the best possible labels to each superpixel. This constitutes an upper bound for any method that enforces all pixels in a superpixel to be assigned the same label. Unfortunately, as Fig. 1 shows, using less than 100 superpixels significantly degrades the best possible accuracy. Thus, to achieve good performance for complex images, a large number of superpixels is needed, making the approach of Kumar and Hebert (2005) computationally too expensive.

3 Probabilistic Segmentation

As a prelude to the construction of the relative location feature in the next section, we start by outlining the basic components that underlie probabilistic image segmentation based on a conditional Markov random field framework. Given a set $\mathcal{V}(\mathcal{I}) = \{S_1, \ldots, S_N\}$ of N regions (or individual pixels) in an image $\mathcal{I}$, multi-class image segmentation is the task of assigning a class label $c_i \in \{1, \ldots, K\}$ to each region S_i. In this paper, we pre-partition each image into a set of contiguous regions called superpixels using an over-segmentation algorithm (Ren and Malik 2003; Felzenszwalb and Huttenlocher 2004), and segment the image by labeling these superpixels.



(a) Over-segmentation (b) CRF Construction

Fig. 2 A toy example showing (**a**) a partitioning of the image to a small number of superpixels based on appearance characteristics, and (**b**) the corresponding conditional Markov random field structure over the superpixels

Toward probabilistic segmentation, we define a distribution over the labels of all superpixels in the image. Let $\mathcal{G}(\mathcal{I}) = \langle \mathcal{V}(\mathcal{I}), \mathcal{E}(\mathcal{I}) \rangle$ be the graph over superpixels where $\mathcal{E}(\mathcal{I})$ is the set of (undirected) edges between adjacent superpixels. Note that, unlike CRF-based segmentation approaches that rely directly on pixels (e.g., Shotton et al. 2006), this graph does not conform to a regular grid pattern, and, in general, each image will induce a different graph structure. Figure 2 shows a toy example of an image that has been pre-partitioned into regions together with the corresponding Markov random field structure.

The conditional distribution of a segmentation (class assignment c_i for each superpixel) for a given image has the general form

$$P(\boldsymbol{c} \mid \mathcal{I}; \boldsymbol{w}) \propto \exp\left\{ \sum_{S_i \in \mathcal{V}(\mathcal{I})} f(S_i, c_i, \mathcal{I}; \boldsymbol{w}) + \sum_{(S_i, S_j) \in \mathcal{E}(\mathcal{I})} g(S_i, S_j, c_i, c_j, \mathcal{I}; \boldsymbol{w}) \right\} \quad (1)$$

where f and g are the singleton and pairwise feature functions, respectively, and $\boldsymbol{w}$ are parameters that we estimate from training data.[1] A typical choice of features is local (singleton) appearance-based features and pairwise affinity features that encourage labels of neighboring regions to be similar.

Given this construction, finding the most likely segmentation amounts to inference on the distribution defined by (1) to find the most probable joint class assignment $\boldsymbol{c}$. For certain classes of models with restricted feature functions (so-called *regular potentials*) and over binary labels this inference task can be performed exactly and efficiently using min-cut-based algorithms (Boykov et al. 2001; Greig et al. 1989). Indeed, Szeliski et al. (2008) compared different energy minimization (max-assignment inference) methods and found that such methods are superior for the binary image segmentation task. However, in the multi-class case and for models with more general features such as the ones we rely on, such methods cannot be used and we have to resort to approximate inference approaches. Max-product loopy belief propagation is one such method. Here messages are passed between superpixels to iteratively update each superpixel's belief over its class label distribution (e.g., Shental et al. 2003). The method is simple to implement and its use is supported by the findings of Szeliski et al. (2008) who show that, even in the context of binary image segmentation, using loopy belief propagation incurs only 0.1% degradation in performance when compared to the exact solution.

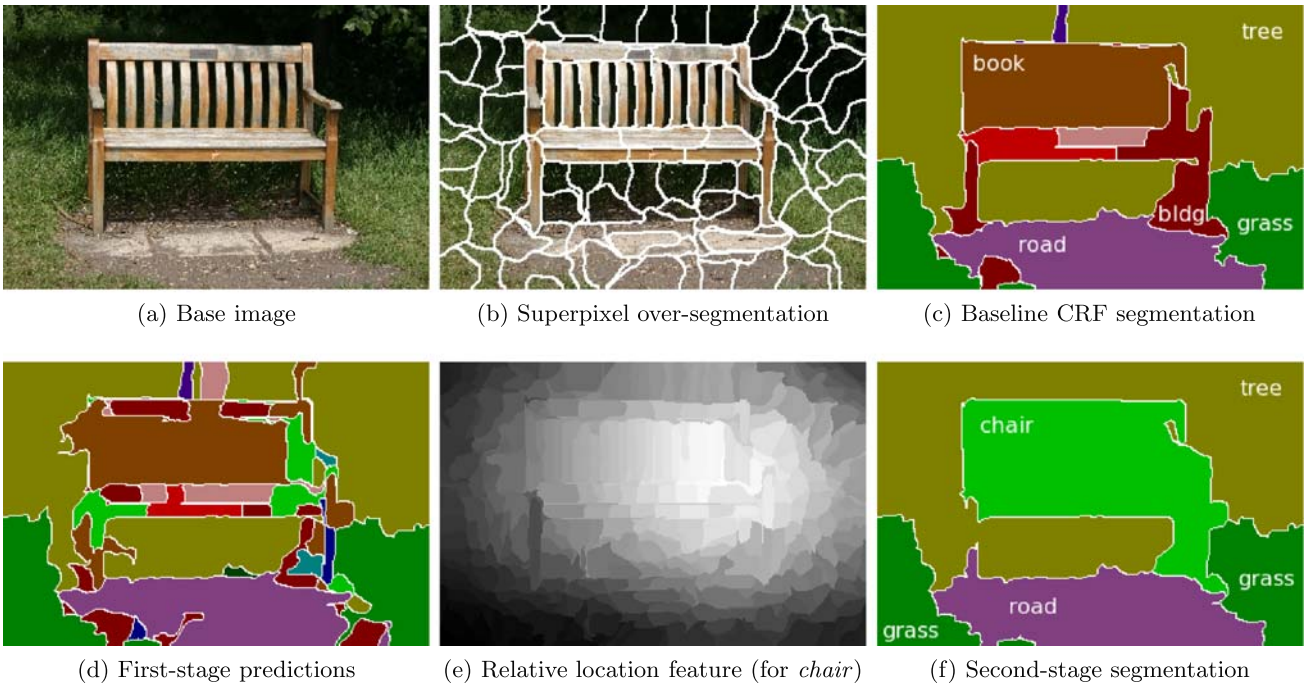
4 Encoding Relative Location

We now discuss the central contribution of this paper—the incorporation of global information as a local feature. Our global information comes from the relative location between two object classes. For example, we wish to make use of the fact that *sky* appears above *grass*. It is not clear how to encode such a global relation via neighboring superpixel preferences as different objects may appear between the *grass* and the *sky* (e.g., *buildings* and *cows*). Furthermore, encoding such preferences explicitly by constructing features that link every superpixel to every other superpixel will make even approximate inference intractable, unless the number of superpixels is very small, at which point important image details are lost.[2] In this section we propose an alternative approach that overcomes this difficulty via a two-stage prediction approach.

The idea of our method is straightforward. Based on training data, we construct relative location probability maps that encode offset preferences between classes. At testing time, we first predict the class $\hat{c}_i$ for each superpixel S_i independently using an appearance based boosted classifier. We then combine these predictions with the relative location probability maps to form an image-dependent relative

[1] In the following discussion we omit the parameters $\boldsymbol{w}$ from the arguments of the feature functions f and g for clarity and include them only when the parameterization of these functions is not obvious.

[2] We experimentally confirmed this claim, as was shown in Fig. 1 above.



(a) Base image (b) Superpixel over-segmentation (c) Baseline CRF segmentation

(d) First-stage predictions (e) Relative location feature (for *chair*) (f) Second-stage segmentation

Fig. 3 Example of our two-stage image segmentation approach for incorporating global information into local features. The first row shows the (**a**) image to be labeled, (b) the over-segmented image, and (**c**) baseline CRF predictions. The second row summarizes the main stages in our method: (**d**) shows the first-stage classification results using only local appearance features, (**e**) shows the relative location feature (normalized for visualization) computed by applying the relative location prior probability maps using the most probable class label for each superpixel from the first-stage classification results, and (**f**) shows second-stage classification results (which includes the relative-location feature). Results have been annotated with class labels

location feature that encodes label preferences for each superpixel based on *all* other superpixels. Finally, we use appearance and relative location features jointly to make a final prediction.

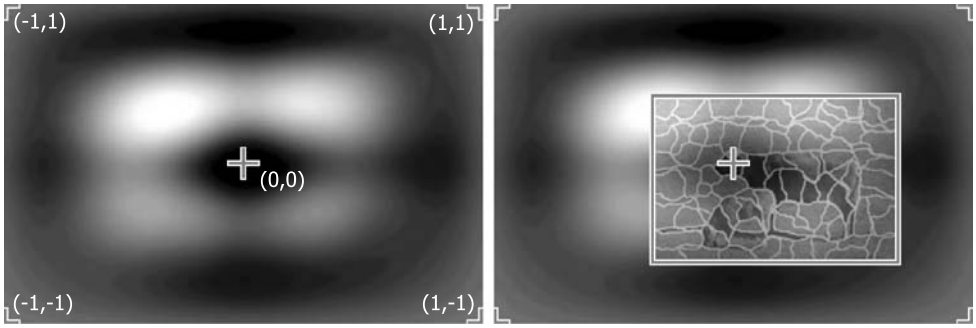
The entire process is exemplified in Fig. 3. The first stage prediction (bottom left) is combined with the precomputed relative location maps to form a relative location feature for the *chair* (light green) class (bottom center). Similarly, relative location features are computed for all other classes (not shown). As can be seen, this features encourages a labeling of a chair in the correct region, in part by making use of the strong road prediction (purple) at the bottom of the image. Combined with appearance features in our second-stage model, this results in a close to perfect prediction (bottom right) of the *chair* (light green), *grass* (dark green), *tree* (olive green) and *road* (purple) classes. Such long range dependencies cannot be captured by a standard CRF model (top right) that makes use of local smoothness preferences.

Below, we describe the construction of the relative location feature. In Sect. 5, we complete the details of the probabilistic segmentation model that makes use of our image-dependent relative location construction.

4.1 Relative Location Probability Maps

We now describe how to construct *relative location probability maps* that encode a-priori inter-class offset preference. Given pixel p' with a class label c', a map $\mathcal{M}_{c|c'}(\hat{u}, \hat{v})$ encodes the probability that a pixel p at offset $(\hat{u}, \hat{v})$ from p' has class label c. The map $\mathcal{M}_{c|c'}(\hat{u}, \hat{v})$ is maintained in normalized image coordinates $(\hat{u}, \hat{v}) \in [-1, 1] \times [-1, 1]$. We also have $\sum_{c=1}^{K} \mathcal{M}_{c|c'}(\hat{u}, \hat{v}) = 1$, so that $\mathcal{M}_{c|c'}$ represents a proper conditional probability distribution over labels, c. Note that a class can also define a relative location prior with respect to itself. An example of the learned relative location probability map $\mathcal{M}_{\text{grass|cow}}(\hat{u}, \hat{v})$ is shown in Fig. 4. The figure also demonstrates how the map, defined over the range $[-1, 1]$ in normalized image coordinates, allows communication of global information from any pixel to any other pixel.

The probability maps are learned from the labeled training data by counting the offset of pixels for each class from the centroid of each superpixel of a given class. The probability maps are quantized to 200×200 pixels and we normalize the offsets by the image width and height, and weight counts by the number of pixels in each superpixel. Because of the sparsity of training examples in the multi-class scenario, we apply a Dirichlet prior with parameter $\alpha = 5$ to



(a) Relative location prior (*grass* to *cow*) (b) Computing relative location offsets

Fig. 4 Example of a relative location non-parametric probability map. (**a**) Shows *grass* relative to *cow* (center pixel) and clearly defines a high probability (*white*) of finding *grass* surrounding *cow* (but more from above) and in close proximity. (**b**) Shows an example of how the map (over normalized range $[-1, 1]$) is used to align a *cow* prediction for one superpixel (corresponding to the head), and then provide a weighted vote for *grass* on any other superpixel in the image (over normalized range $[0, 1]$). To compute the relative location feature, we repeat the process for each superpixel and each pair of classes

the relative offset count. Finally, the relative location probability maps are blurred by a Gaussian filter with variance equal to 10% of the width and height of the image. This reduces bias from small pixel shifts in the training data. See Figs. 3e, 4a and 9 for examples of learned relative location probability maps.

4.2 Relative Location Features

We now describe how the first-stage local predictions are combined with the learned relative location probability maps to form a relative location feature that can be readily incorporated into a second stage segmentation model. Given a pre-partitioned image with superpixels $\mathcal{V}(\mathcal{I}) = \{S_1, \ldots, S_N\}$, we use the local appearance-based predictors (see Sect. 5.3) to find the most probable label $\hat{c}_j$ and its probability $P(\hat{c}_j \mid S_j)$ for each superpixel S_j. Then, based on the relative location probability maps, each superpixel casts a vote for where it would expect to find pixels of every class (including its own class) given its location (superpixel centroid) and predicted label. The votes are weighted by $\alpha_j = P(\hat{c}_j \mid S_j) \cdot |S_j|$, the probability of $\hat{c}_j$ multiplied by the number of pixels in the superpixel.

Thus, each superpixel S_i receives $N-1$ votes from all the other superpixels, S_j. We aggregate the votes from classes $\hat{c}_j \neq \hat{c}_i$ and $\hat{c}_j = \hat{c}_i$ separately to allow different parameters for "self" votes and "other" votes. We have

$$v_c^{\text{other}}(S_i) = \sum_{j \neq i : \hat{c}_j \neq \hat{c}_i} \alpha_j \cdot \mathcal{M}_{c|\hat{c}_j}(\hat{x}_i - \hat{x}_j, \hat{y}_i - \hat{y}_j), \tag{2}$$

$$v_c^{\text{self}}(S_i) = \sum_{j \neq i : \hat{c}_j = \hat{c}_i} \alpha_j \cdot \mathcal{M}_{c|\hat{c}_j}(\hat{x}_i - \hat{x}_j, \hat{y}_i - \hat{y}_j) \tag{3}$$

where $(\hat{x}_i, \hat{y}_i)$ and $(\hat{x}_j, \hat{y}_j)$ are the centroids for the i-th and j-the superpixels, respectively. Finally, the relative location feature is

$$f^{\text{relloc}}(S_i, c_i, \mathcal{I}) = w_{c_i}^{\text{other}} \cdot \log v_{c_i}^{\text{other}}(S_i) + w_{c_i}^{\text{self}} \cdot \log v_{c_i}^{\text{self}}(S_i) \tag{4}$$

where we take the log of the votes because our features are defined in log-space. The self-vote weight $w_{c_i}^{\text{self}}$ and aggregated-vote weight $w_{c_i}^{\text{other}}$ are learned parameters (see Sect. 5).

Our method is summarized in Algorithm 1: we first find the most probable label $\hat{c}_i$ for each superpixel based only on appearance. We then combine these predictions with the precomputed relative location probability maps to form, together with the learned weights, the local relative location feature. Finally, we incorporate this feature into the unified appearance and relative location model to produce a final high-quality image segmentation.

4.3 Complexity

An important benefit that we get from using superpixels is that the complexity of our method depends on the inherent complexity of the image rather than its resolution. Figure 5 shows, for a sample image, that superpixels computed using the method of Ren and Malik (2003) are essentially the same when computed for the same image at different resolutions. This is consistent with the observations of Mori et al. (2004), who note that initial over-segmentation into superpixels provides a higher-level representation of the image while maintaining virtually all structure in real images, i.e., cars, road, buildings, etc.

Computing the relative location feature is quadratic in the number of superpixels as the prediction for each superpixel affects all the others. However, as mentioned above, this can essentially be treated as a constant that is invariant to the image size: the higher the image resolution, the higher the saving when compared to a pixel based method. To make this

Fig. 5 An example demonstrating that approximately the same number of superpixels (100) can be used at different resolutions with negligible differences in terms of capturing the salient details in the image. Original image (*left*) is 640 × 480 pixels. Other images are the result of scaling down by a factor of 2

claim more concrete, in Sect. 6 we provide quantitative evidence that using more superpixels has essentially no effect on the performance of our method with performance saturating between 200 to 400 superpixels. Finally, we note that the actual computation of the relative location prior involves extremely simple operations so that this close-to-constant time is small: for the Sowerby images computing the relative location takes 0.20 seconds while inference takes 5.3 seconds when pre-partitioned into 200 superpixels, and 0.61 seconds and 31 seconds for computing relative location and running inference, respectively, when pre-partitioned into 400 superpixels. For comparison, the method of Kumar and Hebert (2005) takes 6 seconds for inference on the same dataset.

5 Probabilistic Image Segmentation with Relative Location Prior

We are now ready to describe how the relative location feature described in Sect. 4 is incorporated into a unified appearance and relative location model for producing our final multi-class image segmentation.

Recall that an image is pre-partitioned into a set of contiguous regions called superpixels. The set is denoted by $\mathcal{V}(\mathcal{I}) = \{S_1, \ldots, S_N\}$. Using this partitioning we perform a first-stage classification to compute $\hat{c}_i$ from a boosted appearance feature classifier,

$$\hat{c}_i = \operatorname*{argmax}_c P^{\text{app}}(c \mid S_i, \mathcal{I}) \tag{5}$$

(we defer discussion of these features until Sect. 5.3 below). As described above in Algorithm 1, using these predictions we construct our relative location feature and incorporate it into the full model, which is then used to predict a final labeled segmentation. We consider two variants of this final model: a simple logistic regression model that is applied independently to each superpixel; and a richer CRF model that also incorporates pairwise affinity potentials.

5.1 Simple (Logistic Regression) Model

Logistic regression is a simple yet effective classification algorithm which naturally fits within the probabilistic framework of (1). We define three feature functions, f^{relloc} ((4), Sect. 4.2), f^{app} ((10), Sect. 5.3), and a bias term $f^{\text{bias}}(c_i) = w^{\text{bias}}_{c_i}$, giving the probability for a single superpixel label as

$$P(c_i \mid \mathcal{I}; \boldsymbol{w}) \propto \exp\left\{ f^{\text{relloc}}(S_i, c_i, \mathcal{I}) + f^{\text{app}}(S_i, c_i, \mathcal{I}) + f^{\text{bias}}(c_i) \right\} \tag{6}$$

with joint probability $P(\boldsymbol{c} \mid \mathcal{I}; \boldsymbol{w}) = \prod_{S_i \in \mathcal{V}(\mathcal{I})} P(c_i \mid \mathcal{I}; \boldsymbol{w})$. Here the model for probabilistic segmentation defined by (1) is simplified by removing the second summation over pairwise terms. The bias feature $f^{\text{bias}}(c_i) = w^{\text{bias}}_{c_i}$ encodes the prevalence of each object class, and is not dependent on any properties of a particular image.

Since the logistic model (6) decomposes over individual superpixels, training and evaluation are both very efficient.

Algorithm 1 LabelImage: Multi-Class Segmentation with Relative Location Prior

Input : $\mathcal{I}$ // test image
$\mathcal{H}$ // learned models
$\{\mathcal{M}_{c|c'}\}$ // relative location maps
Output: $\mathcal{L}$ // pixel labels for $\mathcal{I}$

// Initial prediction: appearance features only
foreach *superpixel* $S_i \in \mathcal{I}$ **do**
  $\hat{c}_i \leftarrow \text{argmax}_c\, P^{\text{app}}(c \mid S_i, \mathcal{I})$
end
// Compute relative location votes (Eq. 2,3)
foreach *superpixel* $S_i \in \mathcal{I}$ **do**
  foreach *class* c **do**

$$v_c^{\text{other}}(S_i) = \sum_{j \neq i : \hat{c}_j \neq \hat{c}_i} \alpha_j \cdot \mathcal{M}_{c|\hat{c}_j}(\hat{x}_i - \hat{x}_j, \hat{y}_i - \hat{y}_j)$$

$$v_c^{\text{self}}(S_i) = \sum_{j \neq i : \hat{c}_j = \hat{c}_i} \alpha_j \cdot \mathcal{M}_{c|\hat{c}_j}(\hat{x}_i - \hat{x}_j, \hat{y}_i - \hat{y}_j)$$

  end
end
// Compute relative location feature (Eq. 4)
foreach *superpixel* $S_i \in \mathcal{I}$ **do**
  foreach *class* c_i **do**

$$f^{\text{relloc}}(S_i, c_i, \mathcal{I}) = w_{c_i}^{\text{other}} \cdot \log v_{c_i}^{\text{other}}(S_i) + w_{c_i}^{\text{self}} \cdot \log v_{c_i}^{\text{self}}(S_i)$$

  end
end
// Final prediction with relative location:
// Logistic regression (Eq. 6) or CRF (Eq. 8)
$\hat{\boldsymbol{c}} \leftarrow \text{argmax}_{\boldsymbol{c}}\, P(\boldsymbol{c} \mid \mathcal{I}; \boldsymbol{w})$
foreach *superpixel* $S_i \in \mathcal{I}$ **do**
  foreach *pixel* $p \in S_i$ **do**
    $\mathcal{L}[p] \leftarrow \hat{c}_i$
  end
end
return $\mathcal{L}$

In this model, our relative location feature f^{relloc} (which is computed based on our first-stage classifier's prediction of all labels in the image), is the only way in which inter-superpixel dependency is considered.

The weights for the logistic model

$$\boldsymbol{w} = \{w_{c_i}^{\text{other}}, w_{c_i}^{\text{self}}, w_{c_i}^{\text{app}}, w_{c_i}^{\text{bias}} \mid c_i = 1, \ldots, K\}$$

are learned to maximize the conditional likelihood score over our labeled training data using a standard conjugate-gradient algorithm (e.g., Minka 2003).

5.2 Conditional Random Field Model

Typically, a logistic regression model is not sufficiently expressive and some explicit pairwise dependence is required. Indeed, most recent works on probabilistic image segmentation use conditional Markov random field (CRF) models (He et al. 2006; Shotton et al. 2006; Winn and Shotton 2006) that, in addition to (6), also encode conditional dependencies between neighboring pixels/superpixels. The CRF formulation allows a smoothness preference to be incorporated into the model. Furthermore, pairwise features also encapsulate local relationships between regions. For example, given two adjacent superpixels, a pairwise feature might assign a greater value for a labeling of *cow* and *grass* than it would for *cow* and *airplane*, because cows are often next to grass and rarely next to airplanes. Note that this is different to the *global* information provided by the relative location feature.

Here in addition to the features defined for the logistic model, we define the (constant) pairwise feature between all adjacent superpixels

$$f^{\text{pair}}(c_i, c_j, \mathcal{I}) = \frac{w_{c_i,c_j}^{\text{pair}}}{0.5(d_i(\mathcal{I}) + d_j(\mathcal{I}))} \tag{7}$$

where $d_i(\mathcal{I})$ is the number of superpixels adjacent to S_i. This feature is scaled by d_i and d_j to compensate for the irregularity of the graph $\mathcal{G}(\mathcal{I})$ (as discussed in Sect. 1). Our full CRF model is then

$$P(\boldsymbol{c} \mid \mathcal{I}; \boldsymbol{w}) \propto \exp\left\{ \sum_{S_i \in \mathcal{V}(\mathcal{I})} \left(f^{\text{relloc}}(S_i, c_i, \mathcal{I}) + f^{\text{app}}(S_i, c_i, \mathcal{I}) + f^{\text{bias}}(c_i) \right) + \sum_{(S_i, S_j) \in \mathcal{E}(\mathcal{I})} f^{\text{pair}}(c_i, c_j, \mathcal{I}) \right\} \tag{8}$$

where the first summation is over individual superpixels and the second summation is over pairs of adjacent superpixels.

We use max-product propagation inference (Pearl 1988) to estimate the max-marginal over the labels for each superpixel given by (8), and assign each superpixel the label which maximizes the joint assignment to the image.

At training time, we start with weights already learned from the logistic regression model and hold them fixed while training the additional weights $w_{c,c'}^{\text{pair}}$ for the pairwise affinity features (with the constraint that $w_{c,c'}^{\text{pair}} = w_{c',c}^{\text{pair}}$). As it is computationally impractical to learn these parameters as part of a full CRF, we use piecewise training (Sutton and McCallum 2005; Shotton et al. 2006) in which parameters are optimized to maximize a lower bound of the full CRF likelihood function by splitting the model into disjoint node pairs and integrating statistics over all of these pairs.

5.3 Appearance Features

To complete the details of our method, we now describe how the appearance features are constructed from low-level descriptors. For each superpixel region S_i, we compute an 83-dimensional description vector $\phi(S_i)$ incorporating region

size, location, color, shape and texture features. Our features build on those of Barnard et al. (2003) and consist of mean, standard deviation, skewness and kurtosis statistics over the superpixel of:

- RGB color-space components (4×3)
- Lab color-space components (4×3)
- Texture features drawn from 13 filter responses, including oriented Gaussian, Laplacian-of-Gaussian, and pattern features such as corners and bars (4×13).

In addition, again following Barnard et al. (2003), we compute the size, location (x and y offsets and distance from image center), and shape of the superpixel region. Our shape features consist of the ratio of the region area to perimeter squared, the moment of inertia about the center of mass, and the ratio of area to bounding rectangle area. Pixels along the boundary of the superpixel are treated the same as the interior. We append to the description vector the weighted average of the appearance over the neighbors for each superpixel,

$$\frac{\sum_{S_j \in \mathcal{N}(S_i)} |S_j| \cdot \phi(S_j)}{\sum_{S_j \in \mathcal{N}(S_i)} |S_j|} \tag{9}$$

where $\mathcal{N}(S_i) = \{S_j \mid (S_i, S_j) \in \mathcal{E}(\mathcal{I})\}$ is the set of superpixels which are neighbors of S_i in the image and $|S_j|$ is the number of pixels in superpixel S_j.

We learn a series of one-vs-all AdaBoost classifiers (Schapire and Singer 1999) for each class label c'. Here, we take as positive examples the superpixels which are assigned to that class in the ground-truth labeling, and as negative examples all superpixels which are assigned to a different class in the ground-truth.

We apply the AdaBoost classifier that we have learned for each class c' to the vector of descriptors and normalize over all classes to get the probability

$$P^{\text{app}}(c_i = c' \mid S_i, \mathcal{I}) = \frac{\exp\{\sigma_{c'}\}}{\sum_c \exp\{\sigma_c\}}$$

where σ_c is the output of the AdaBoost classifier for class c. We then define the appearance feature as

$$f^{\text{app}}(S_i, c_i, \mathcal{I}) = w^{\text{app}}_{c_i} \cdot \log P^{\text{app}}(c_i \mid S_i, \mathcal{I}) \tag{10}$$

where the $w^{\text{app}}_{c_i}$ are learned as described in Sect. 5.1.

6 Experimental Results

We conduct experiments to evaluate the performance of the image-dependent relative location prior and compare the accuracy of our method both to a baseline without that prior and to recently published state-of-the-art results on several datasets: the 21-class and 9-class MSRC image segmentation databases (Criminisi 2004); and the 7-class Corel and Sowerby databases used in He et al. (2004).

In all experiments, when training, we take the ground-truth label of a superpixel to be the majority vote of the ground-truth pixel labels. At evaluation time, to ensure no bias in favor of our method, we compute our accuracy at the pixel level. For all datasets, we randomly divide the images into balanced training and test data sets (number of occurrences of each class approximately proportional to the overall distribution). The training set is used for training the boosted appearance classifier, constructing the image-dependent relative location priors, and training the parameters of the logistic and CRF models as described in Sect. 5. The remaining instances are only considered at testing time. In each experiment, we compare four different models: a baseline logistic regression classifier; a baseline CRF; a logistic regression model augmented with our image-dependent relative location feature; and a CRF model augmented with our relative location feature. To ensure robustness of the reported performance, we repeat *all* evaluations on five different random train/test partitionings for the large databases and ten different random partitionings for the small databases and report minimum, maximum and average performance. This is in contrast to all state-of-the-art methods we compare against, which were evaluated only on a single fold.

6.1 MSRC Databases

We start with the MSRC 21-class database which is the most comprehensive and complex dataset consisting of 591 images labeled with 21 classes: *building, grass, tree, cow, sheep, sky, airplane, water, face, car, bicycle, flower, sign, bird, book, chair, road, cat, dog, body, boat*. The ground-truth labeling is approximate (with foreground labels often overlapping background objects) and includes a *void* label to handle objects that do not fall into one of the 21 classes. Following the protocol of previous works on this database (Schroff et al. 2006; Shotton et al. 2006; Yang et al. 2007), we ignore void pixels during both training and evaluation. We over-segment our images using the method of Ren and Malik (2003), tuned to give approximately 400 superpixels per image. We randomly divide the images into a training set with 276 images and a testing set with 315 images, allowing us to be directly comparable to the results of Shotton et al. (2006).[3]

Table 1 compares the performance of the different models with the reported state-of-the-art performance of other

[3] We note that we could not verify the precise number of training images used by Yang et al. (2007) as they report 40% of 592 images, a non-integer quantity.

Table 1 Comparison of our results on the 21-class and 9-class MSRC databases (Criminisi 2004). Shown are the minimum, average, maximum, and standard deviation for pixel prediction accuracy over five separate random partitionings of the database into training and testing sets

Algorithm	21-class MSRC accuracy				9-class MSRC accuracy			
	Min.	Avg.	Max.	Std.	Min.	Avg.	Max.	Std.
Shotton et al. (2006)		72.2%*		n/a		–		–
Yang et al. (2007)		75.1%*		n/a		–		–
Schroff et al. (2006)		–		–		75.2%*		n/a
Baseline logistic	61.8%	63.6%	65.4%	1.67%	77.5%	78.9%	79.8%	1.05%
Baseline CRF	68.3%	70.1%	72.0%	1.81%	81.2%	83.0%	84.4%	1.28%
Logistic + Rel. Loc.	73.5%	75.7%	77.4%	1.74%	87.6%	88.1%	88.9%	0.67%
CRF + Rel. Loc.	74.0%	**76.5%**	78.1%	1.82%	87.8%	**88.5%**	89.5%	0.82%

*For the other works, results are only reported on a single fold

works on the 21-class MSRC database (left column). The advantage of the relative location prior is clear, improving our baseline CRF performance by over 6% on average. Importantly, combining this feature with standard appearance features (i.e., colors and textures), we are able to achieve an average improvement of 1.4% over state-of-the-art performance (the single fold experiment of Yang et al. 2007). A full confusion matrix summarizing our pixel-wise recall results over all 21 classes is given in Table 2, showing the performance of our method (top line of each cell) and that of the baseline CRF (bottom line of each cell in parentheses). Note that "object" classes (e.g., *cow*, *sheep*, *airplane*) obtain the greatest gain from relative location, whereas "background" classes (e.g., *grass*, *sky*) only benefit a little.

The relative location feature dramatically improves the performance of the simple logistic regression model (average +12.1% on 21-class MSRC). It performs superior to state-of-the-art and is only slightly inferior to the more complex CRF model. Thus, our efficient two-stage evaluation approach that globally "propagates" a relative location preference is able to compete with models that make use of pairwise affinities and require time-consuming inference.

For the 9-class MSRC database we follow the procedure of Schroff et al. (2006) by splitting the database evenly into 120 images for training and 120 images for testing. Results for the 9-class MSRC database are shown in the right hand columns of Table 1. Our method surpasses state-of-the-art performance by over 13% on average. Again there is little difference between the CRF model and simpler logistic regression model once relative location information is included.

6.2 Corel and Sowerby Databases

We now consider the somewhat simpler 7-class Corel and Sowerby databases consisting of 100 and 104 images, respectively. For the Corel and Sowerby datasets we follow the procedure of He et al. (2004) by training on 60 images and testing on the remainder. We repeat the evaluation on ten different random train/test partitionings and report the minimum, maximum and average for the test set. Following the pre-processing described in Shotton et al. (2006), we append appearance features computed on color and intensity normalized images to our base appearance feature vector for the Corel dataset.

A comparison of results is shown in Table 3. Most notable is the small range of performance of the different methods, particularly for the Sowerby database. Indeed, both the 2.5% superiority of our best result over the best Corel results and the 0.7% inferiority relative to the best Sowerby result have a magnitude that is less than one standard deviation and cannot be considered as statistically significant. We conjecture that on these somewhat simpler databases, performance of the state-of-the-art methods, as well as ours, are near saturation.

6.3 Relative vs. Absolute Location Prior

To visualize the kind of information that allows us to achieve performance gains over state-of-the-art, Fig. 9 shows the relative location priors learned between different classes. To quantitatively verify that these learned priors capture meaningful information *between* classes (e.g., grass around cow) rather than absolute location information (e.g., cow in center of image), we tested our model without the absolute location information that is used by our method as well as by the methods we compare against. That is, we removed the pixel location information from the appearance feature descriptor (Sect. 5.3). Without this information we achieved an average baseline CRF accuracy of 68.7% (cf. 70.1%) and relative location CRF accuracy of 74.9% (cf. 76.5%) on the 21-class MSRC database. This demonstrates that relative location information does indeed provide discriminative power above that of an absolute location prior. Furthermore, this shows that absolute location is only marginally helpful when relative location information is already incorporated in the model.

Table 2 Accuracy of our approach on the 21-class MSRC database (Criminisi 2004). The confusion matrix shows the pixel-wise recall accuracy (across all folds) for each class (rows) and is row-normalized to sum to 100%. Row labels indicate the true class, and column labels the predicted class. The second number in parentheses in each cell shows baseline CRF result

	Building	Grass	Tree	Cow	Sheep	Sky	Airplane	Water	Face	Car	Bicycle	Flower	Sign	Bird	Book	Chair	Road	Cat	Dog	Body	Boat
Building	**72.3**	0.8	3.2	1.1	0.6	2.4	0.5	3.2	1.8	2.6	0.4	–	1.5	0.4	1.6	0.5	5.4	–	0.6	0.5	0.5
	(72.1)	(1.2)	(3.5)	(1.1)	(0.3)	(3.4)	(1.1)	(2.5)	(1.3)	(2.7)	(0.3)	–	(1.4)	(0.1)	(2.4)	(0.2)	(5.4)	(0.2)	(0.3)	(0.3)	(0.1)
Grass	0.1	**94.8**	2.7	0.8	0.3	–	0.4	0.1	–	–	0.1	–	–	0.1	–	0.1	0.2	–	–	0.2	–
	(0.4)	(94.3)	(3.4)	(0.7)	(0.3)	–	(0.3)	(0.2)	–	–	–	–	–	–	–	–	(0.2)	–	–	(0.1)	–
Tree	4.6	5.0	**81.3**	0.1	–	2.2	0.6	1.9	0.2	0.4	0.7	–	0.5	0.6	–	1.1	0.1	–	0.4	0.2	0.1
	(6.7)	(6.6)	(79.1)	(0.3)	(0.1)	(2.3)	(0.5)	(1.0)	(0.1)	(1.2)	(0.8)	(0.1)	(0.1)	(0.3)	–	(0.2)	(0.3)	–	(0.2)	(0.1)	(0.1)
Cow	0.1	14.9	4.2	**66.3**	1.8	0.2	–	2.0	0.1	–	–	0.6	0.1	2.3	–	0.5	0.1	1.1	5.2	0.4	–
	(5.6)	(14.1)	(3.3)	(58.6)	(4.1)	(0.3)	(0.2)	(3.3)	(1.2)	(0.8)	(0.1)	(1.6)	(0.4)	(1.0)	(0.1)	(0.1)	(0.3)	(1.4)	(2.9)	(0.7)	–
Sheep	–	12.1	0.2	2.6	**71.0**	–	–	0.4	0.1	–	–	–	–	2.3	–	0.3	8.3	–	2.7	–	–
	(7.1)	(11.6)	(3.4)	(3.9)	(57.9)	(0.3)	(0.1)	(3.1)	(0.3)	(0.1)	(0.1)	(0.2)	–	(1.2)	–	–	(6.9)	(0.8)	(2.8)	(0.2)	–
Sky	2.2	–	1.0	0.1	–	**92.6**	0.5	3.3	–	0.1	–	–	0.2	–	–	–	0.1	–	–	–	–
	(2.5)	–	(0.6)	–	–	(91.2)	(0.4)	(4.4)	–	(0.1)	–	–	(0.2)	(0.1)	(0.1)	–	(0.4)	–	–	–	–
Airplane	20.2	1.8	1.0	–	–	2.3	**73.6**	0.3	–	0.4	–	–	–	–	–	–	0.3	–	–	–	–
	(30.6)	(2.7)	(4.1)	(0.1)	–	(1.7)	(53.2)	(0.4)	–	(5.6)	(0.2)	–	(0.7)	–	–	(0.1)	(0.4)	–	–	–	(0.3)
Water	3.6	4.4	3.0	0.2	0.3	4.4	0.1	**69.6**	–	2.3	1.4	0.1	–	0.2	–	0.3	9.2	–	0.2	0.3	0.4
	(5.5)	(5.3)	(4.6)	(0.2)	(0.3)	(4.8)	(0.1)	(65.5)	–	(2.0)	(0.7)	–	(0.1)	(0.3)	(0.2)	–	(9.2)	(0.1)	(0.2)	(0.1)	(0.9)
Face	4.2	0.3	1.5	0.9	–	0.1	–	0.1	**70.2**	0.1	–	1.1	–	0.1	7.6	0.3	0.1	2.0	0.3	11.3	–
	(11.9)	(0.5)	(2.8)	(3.5)	(0.1)	(0.1)	–	(0.1)	(66.2)	(0.5)	(0.2)	(0.5)	(0.1)	–	(1.4)	(0.1)	(0.3)	(1.8)	(1.2)	(8.7)	–
Car	12.5	–	3.7	–	–	1.6	–	5.9	–	**68.9**	–	1.7	0.8	0.4	–	–	3.4	–	–	–	1.1
	(17.3)	(0.1)	(4.5)	–	–	(2.4)	(1.8)	(9.7)	(0.2)	(53.6)	(0.8)	(2.3)	(0.9)	(0.1)	(0.7)	–	(3.6)	(0.2)	–	(0.4)	(1.2)
Bicycle	16.7	0.2	2.5	–	–	–	–	0.8	–	0.8	**71.7**	0.6	–	–	–	1.2	5.2	–	–	0.3	–
	(26.8)	(0.5)	(8.8)	–	–	(0.1)	(0.2)	(0.8)	(0.2)	(5.0)	(50.3)	(0.3)	(0.2)	–	–	(0.1)	(5.1)	(0.1)	–	(1.4)	(0.1)

Table 2 (*Continued*)

	Building	Grass	Tree	Cow	Sheep	Sky	Airplane	Water	Face	Car	Bicycle	Flower	Sign	Bird	Book	Chair	Road	Cat	Dog	Body	Boat
Flower	0.1	2.9	5.6	3.8	1.2	0.6	–	0.2	3.1	–	1.7	**67.6**	2.8	5.9	0.1	–	–	0.1	1.2	3.1	–
	(1.8)	(5.0)	(6.3)	(7.9)	(1.2)	(1.4)	(0.1)	(1.1)	(3.3)	(0.7)	(2.6)	(54.4)	(2.4)	(1.5)	(3.0)	(0.1)	(0.3)	(0.2)	(0.2)	(6.3)	–
Sign	21.0	–	1.3	–	–	1.1	–	0.7	0.2	0.1	–	1.6	**54.8**	0.5	13.2	2.3	1.9	0.9	–	0.5	–
	(26.8)	(0.2)	(3.1)	(0.5)	–	(2.2)	(0.7)	(1.0)	(0.6)	(4.1)	(0.6)	(1.7)	(44.6)	(0.2)	(9.9)	(0.1)	(1.9)	(0.3)	–	(1.2)	(0.4)
Bird	5.5	9.3	14.5	3.1	7.2	5.8	0.6	7.8	–	3.5	3.0	–	1.1	**23.0**	–	3.8	8.3	0.1	1.1	0.4	1.8
	(18.0)	(5.5)	(15.8)	(4.8)	(11.3)	(5.0)	(1.3)	(5.7)	(0.6)	(5.9)	(0.2)	–	(1.7)	(11.5)	(0.1)	(1.0)	(5.8)	(0.5)	(2.2)	(2.0)	(0.9)
Book	3.1	0.1	0.2	0.1	–	–	–	0.4	0.5	–	–	6.7	0.1	–	**82.5**	0.6	0.2	0.1	–	3.2	2.1
	(9.3)	(1.6)	(1.2)	(2.3)	(0.2)	(0.2)	(0.9)	(0.8)	(0.8)	(2.2)	(0.3)	(4.8)	(1.4)	(0.1)	(67.4)	(0.9)	(1.4)	(0.3)	(0.4)	(2.9)	(0.4)
Chair	28.0	6.0	4.8	6.8	–	0.1	0.1	0.3	–	1.0	0.3	2.1	–	1.6	2.0	**39.6**	5.6	0.9	–	0.2	0.6
	(39.2)	(7.5)	(8.5)	(8.1)	(0.1)	(0.1)	(2.9)	(1.1)	(0.6)	(2.1)	(1.4)	(2.0)	(0.4)	(0.4)	(2.6)	(16.5)	(4.2)	(0.4)	(0.2)	(0.9)	(0.7)
Road	5.1	0.6	0.3	–	0.3	1.5	0.5	9.1	0.4	2.2	0.6	0.1	–	0.1	–	0.3	**77.0**	0.6	0.5	1.0	–
	(8.2)	(0.9)	(0.4)	(0.1)	(0.1)	(1.7)	(0.2)	(9.9)	(0.3)	(1.7)	(0.3)	–	(0.1)	–	(0.1)	(0.2)	(74.5)	(0.2)	(0.3)	(0.7)	–
Cat	2.7	–	2.3	0.8	–	–	–	3.1	0.7	0.4	0.2	9.9	–	2.2	–	–	11.7	**60.4**	5.2	0.3	–
	(12.9)	(0.1)	(3.1)	(8.3)	–	(0.4)	(0.1)	(2.4)	(2.1)	(2.8)	(1.1)	(1.6)	–	(1.7)	(0.6)	–	(9.5)	(43.5)	(7.8)	(1.9)	(0.1)
Dog	2.9	2.3	4.8	3.7	1.5	2.9	–	0.2	7.8	0.1	–	–	–	2.6	–	0.4	5.7	11.7	**49.6**	4.0	–
	(9.4)	(2.7)	(2.7)	(7.3)	(4.8)	(5.7)	(0.1)	(2.5)	(8.1)	(0.2)	(0.4)	–	–	(2.5)	–	(0.2)	(7.6)	(7.9)	(34.8)	(2.9)	–
Body	5.1	3.3	3.3	7.6	0.2	0.1	–	2.1	8.1	0.5	0.9	8.6	0.6	0.4	1.8	3.4	2.8	0.1	1.6	**49.5**	0.2
	(9.7)	(3.1)	(1.8)	(9.6)	(1.3)	(0.4)	(0.1)	(2.3)	(6.7)	(4.5)	(1.8)	(3.6)	(1.2)	(0.4)	(4.5)	(0.4)	(3.5)	(0.1)	(0.9)	(43.9)	(0.3)
Boat	22.4	0.2	0.7	–	–	1.2	–	26.0	–	30.1	1.0	–	0.7	2.0	–	–	1.5	–	–	0.2	**14.0**
	(33.0)	(1.0)	(3.8)	(0.1)	–	(1.0)	(1.2)	(8.8)	(0.1)	(32.8)	(0.9)	–	(0.5)	(1.4)	(0.3)	(0.2)	(1.0)	(0.1)	–	(1.5)	(12.0)

Table 3 Comparison of our results on the 7-class Corel and Sowerby databases (He et al. 2004). Shown are the minimum, average, maximum, and standard deviation for pixel prediction accuracy over ten separate random partitionings of the database into training and testing sets

Algorithm	7-class Corel accuracy				7-class Sowerby accuracy			
	Min.	Avg.	Max.	Std.	Min.	Avg.	Max.	Std.
He et al. (2004)		**80.0%***		n/a		**89.5%***		n/a
Kumar et al. (2005)		–		–		89.3%*		n/a
Shotton et al. (2006)		74.6%*		n/a		88.6%*		n/a
Yang et al. (2007)		–		–		88.9%*		n/a
Baseline logistic	68.2%	72.7%	76.8%	2.68%	84.7%	86.4%	88.0%	0.92%
Baseline CRF	69.6%	74.9%	78.5%	2.80%	84.9%	87.2%	88.6%	1.01%
Logistic + Rel. Loc.	70.7%	76.4%	81.6%	3.07%	84.9%	87.2%	88.5%	0.98%
CRF + Rel. Loc.	71.1%	77.3%	82.5%	3.15%	85.2%	87.5%	88.8%	0.98%

*For the other works, results are only reported on a single fold

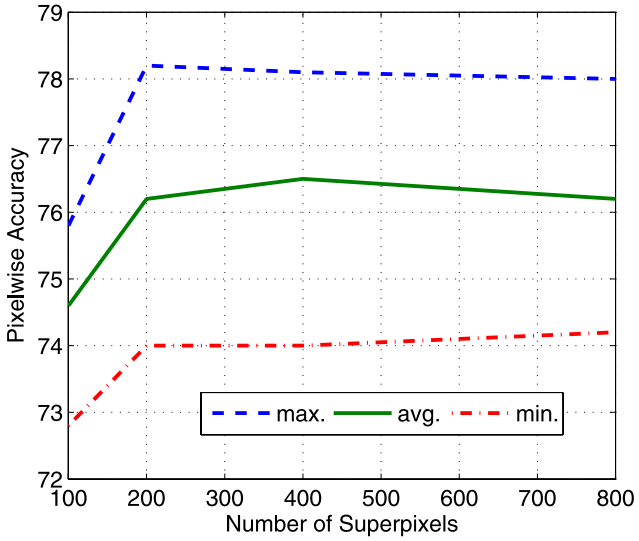


Fig. 6 Plot of accuracy versus number of superpixels for MSRC 21-class database. Shown are the results for 5 random test/train partitionings

6.4 Robustness to Over-Segmentation

To determine the effect of the parameters used by the initial over-segmentation algorithm, we repeated our experiments on the 21-class MSRC database using different numbers of superpixels. The results, shown in Fig. 6, indicate that after approximately 200 superpixels the accuracy of our algorithm is very insensitive to changes in the number of segments. This is because most of the spatial complexity of the image is captured by about 200 superpixels and any refinement to the over-segmentation does not capture any more information.

6.5 Qualitative Assessment

Finally, to gain a qualitative perspective of the performance of our method, Fig. 7 shows several representative images (first column), along with the baseline logistic regression and CRF predictions (second and third columns), as well as the predictions of those models when augmented with our image-dependent relative location feature (fourth and fifth columns). The segmentations exemplify the importance of relative location in maintaining spatial consistency between classes. The first example shows how relative location can correct misclassifications caused by similar local appearance. Here the similarity of the *chair* to a collection of *book*s is corrected through the context of *road*, *tree* and *grass*. In the second example, the baseline CRF labels part of the *sheep* as *road* since both *road*, *grass* and *sheep* are likely to appear together. Relative location can augment that prior with the information that road does not typically appear *above* sheep and results in a close to perfect prediction. The fourth row shows a similar result involving the *car*, *sign* and *sky* classes.

The third and fifth rows show interesting examples of how the relative location self-votes affect predictions. In the third row the local predictors vote for *dog*, *cow* and *sheep* (more clearly seen when the smoothness constraint is applied in the baseline CRF (b)). However we know from relative location information that these three classes do not occur in close proximity. The self-vote together with the fact the each superpixel weights its vote by the confidence in its first-stage prediction, $P(\hat{c}_j \mid S_j)$, allows the model to correctly label the entire dog. The fifth row illustrates the same affect in the more complicated street scene. Although *bicycles* (fruschia) are likely to appear below buildings and trees, a band of *people* is more likely.

In Fig. 8 we also show several cases for which our relative location model was not able to improve on the predictions of the baseline CRF. In the first row, a mixed *building/sign* prediction for a bird was changed to a *sign*. This is a result of the prevalence of signs surrounded by sky in the dataset and the rarity of flying birds. The second row shows ducks that are predicted as *tree* due to their bark-like appearance. As *tree* and *grass* are often found next to each other, the relative location feature is not able to correct this prediction. Finally, the third row demonstrates the common confusion

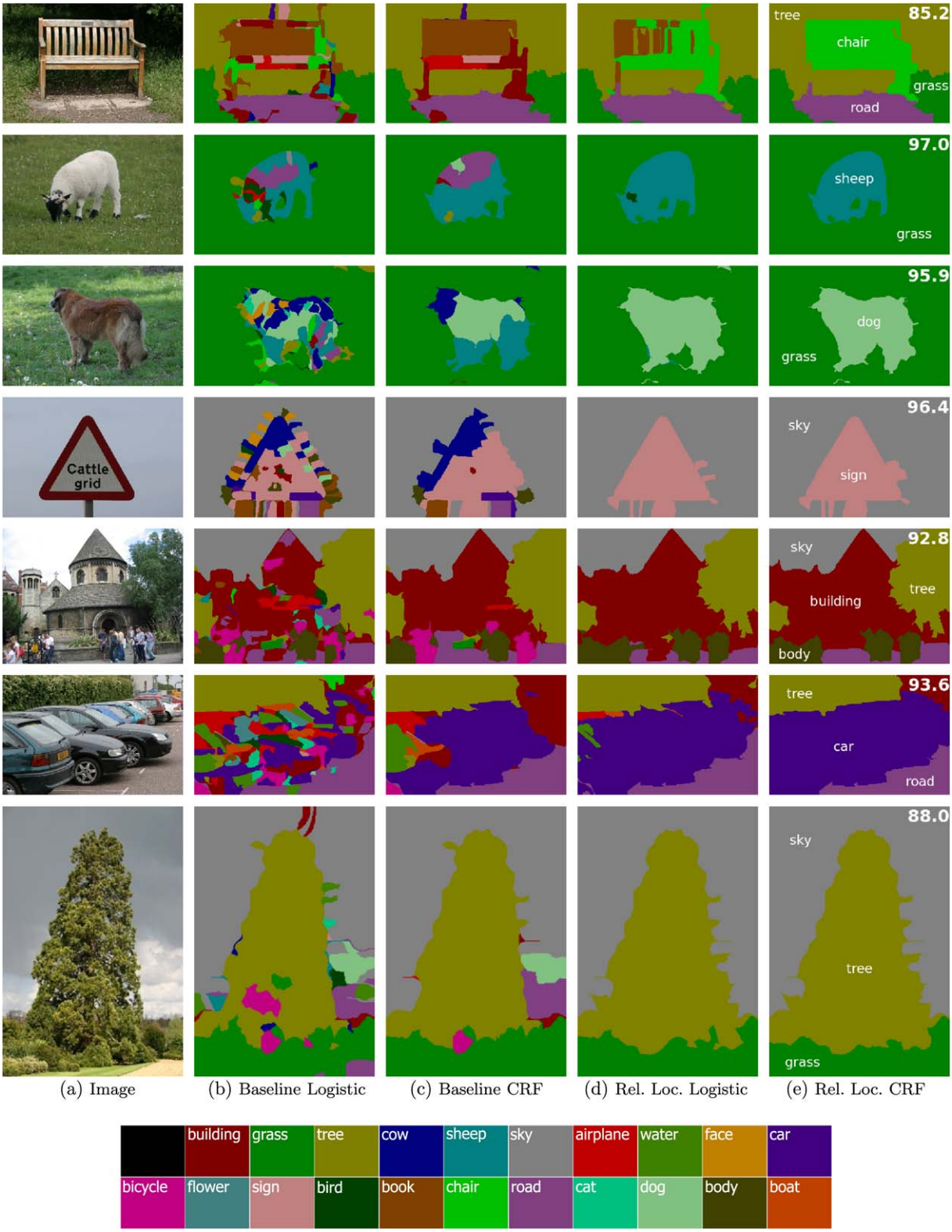


Fig. 7 Representative images where our relative location based method is able to correct prediction mistakes on the 21-class MSRC database (Criminisi 2004). Column (**a**) shows the original image to be labeled. Columns (**b**) and (**c**) show the prediction of the baseline logistic regression and CRF models, respectively. Columns (**d**) and (**e**) show the same result for these models when augmented with our relative location feature. Numbers in the upper-right corner indicate pixel-level accuracy on that image. Note that in many cases there is an effective upper limit on accuracy because ground truth is only approximate

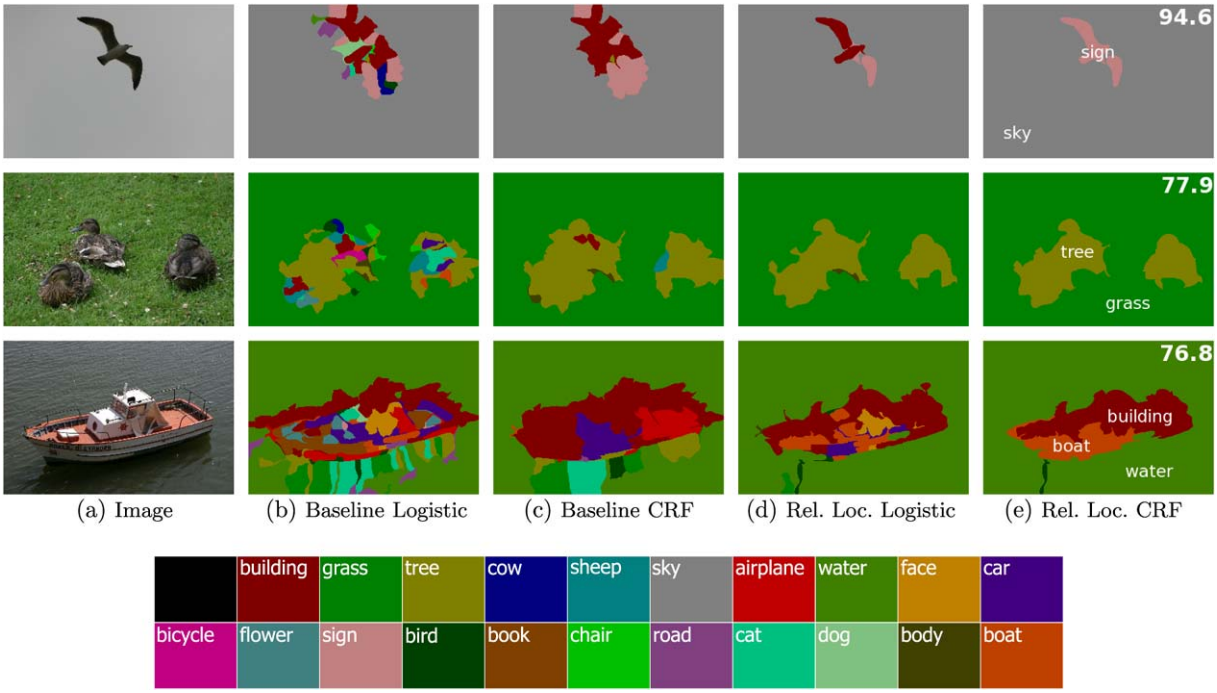


Fig. 8 Several example for which our relative location based method was not able to correct mistakes made by the baseline method on the 21-class MSRC database (Criminisi 2004). Column (**a**) shows the original image to be labeled. Columns (**d**) and (**e**) show the same result for these models when augmented with our relative location feature. Numbers in the upper-right corner indicate pixel-level accuracy on that image. Note that in many cases there is an effective upper limit on accuracy because ground truth is only approximate

between *boat* and *building* (see Table 2) despite the fact that boats are often surrounded by *water* and buildings are not. Although part of the *boat* is properly corrected, the other part is still labeled as *building* due to the strong appearance signal and that some of first-stage predictions for superpixels below the boat (*road*, *tree* and *car*) are actually supportive of the *building* hypothesis.

7 Discussion

In this paper we addressed the challenge of incorporating a global feature, i.e., relative location preference, into a probabilistic multi-class image segmentation framework. We proposed a method in which such a prior is transformed into a local feature via a two-step evaluation approach. By making use of relative location information, our method is not only able to improve on the baseline, but surpasses state-of-the-art results on the challenging MSRC image segmentation databases. Importantly, we demonstrated that, with our relative location feature, even a simple logistic regression approach to segmentation achieves results that are above state-of-the-art.

The main contribution of our paper is in presenting an approach that facilitates the incorporation of a global image prior via local features, thereby facilitating efficient learning and segmentation. In particular, unlike absolute location preferences, our approach applies to global information that *cannot* be computed by pre-processing, i.e., is image and model dependent.

We chose to represent inter-class spatial relationships using non-parametric relative location maps. This allowed our model to capture the complex (multi-modal) spatial distributions that exist between classes (see Fig. 9). An alternative approach could be to use semi-parametric models such as locally weighted kernels, which may be more efficient as the number of classes is increased and when considering scenes at different scales.

One of the main limitations of our approach is that it does not distinguish between objects at different scales. Indeed, our relative location probability maps are constructed as an average over all scales. Despite achieving state-of-the-art accuracy, we believe that even better results can be obtained by taking scale into account. A promising approach is to identify objects in the scene using standard object detectors and use these detections to index scale-aware relative location probability maps. Having such detections can also provide more information in the form of entire objects (rather than small object regions) which will allow multi-class image segmentation methods to better distinguish between classes, such as dogs and cats, where local appearance and relative location are similar. We intend to pursue this approach in future work.

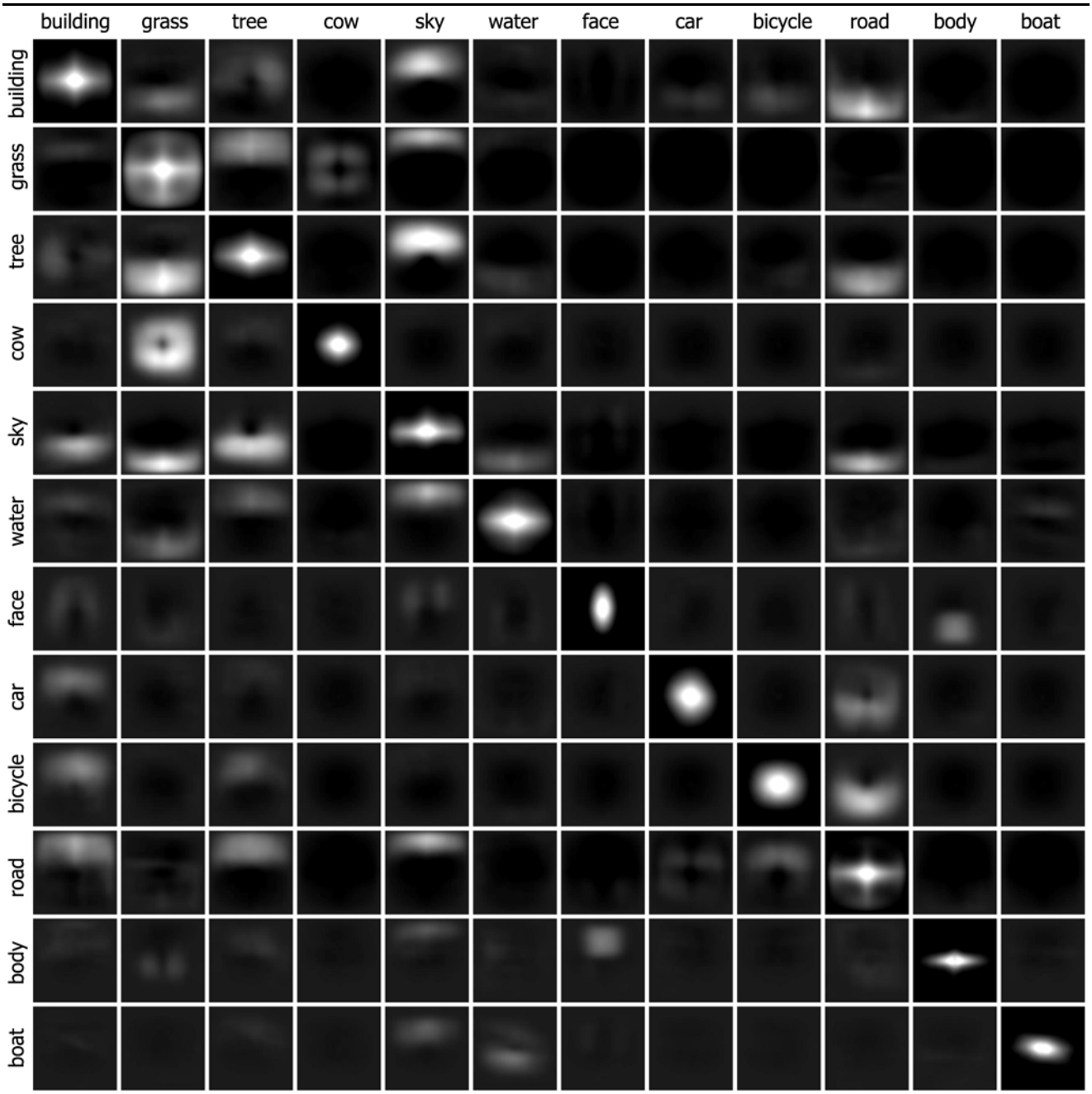


Fig. 9 Learned relative location between example classes. Table shows prediction of column class in relation to row class. For example, the 5th image in the top row shows that we learn *sky* occurs above *building*. *White* indicates a stronger preference

Another limitation of our approach is that mistakes in the first-stage classifiers can result in poor context features for the second-stage. At a high level the construction of the relative location feature can be viewed as a mixture-of-experts model, where each superpixel is seen to be an expert in predicting the label of *all* other superpixels in (4). We then perform a single belief propagation like step by multiplying the mixture-of-experts distribution with the current belief state (initial prediction) to make the final prediction. This is done as an alternative to the direct approach of combining multiple incoming messages into each superpixel, an approach which is computationally demanding. By allowing second-stage beliefs to propagate back to the first-stage, we may be able to correct errors made by the first-stage classifier. This view of our method may also explain why local pairwise effects (affinity functions) help only marginally after the relative location information has been propagated.

The above insight opens the door for exciting research into the balance between complexity of the model and the way in which it is used for inference. In particular, it suggests that many types of global information preferences can be efficiently incorporated into the segmentation process when applied with a limited horizon inference approach.

Acknowledgements The authors would like to thank Geremy Heitz for his helpful suggestions, and David Vickrey and Haidong Wang for discussions on efficient CRF inference and learning. This research was supported by the Defense Advanced Research Projects Agency (DARPA) under contract number SA4996-10929-3 and the Department of Navy MURI under contract number N00014-07-1-0747.

References

Adams, N. J., & Williams, C. K. (2003). Dynamic trees for image modelling. *Image and Vision Computing*, *21*, 865–877.

Barnard, K., Duygulu, P., Freitas, N. D., Forsyth, D., Blei, D., & Jordan, M. (2003). Matching words and pictures. *Journal of Machine Learning Research*, *3*, 1107–1135.

Boykov, Y., Veksler, O., & Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*, 1222–1239.

Carbonetto, P., de Freitas, N., & Barnard, K. (2004). A statistical model for general contextual object recognition. In *ECCV*.

Criminisi, A. (2004). Microsoft research Cambridge object recognition image database (version 1.0 and 2.0). http://research.microsoft.com/vision/cambridge/recognition.

Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision*, *59*(2), 167–181.

Fink, M., & Perona, P. (2003). Mutual boosting for contextual inference. In *NIPS*.

Greig, D. M., Porteous, B. T., & Seheult, A. H. (1989). Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society. Series B (Methodological)*, *51*(2), 271–279.

He, X., Zemel, R., & Carreira-Perpinan, M. (2004). Multiscale conditional random fields for image labelling. In *CVPR*.

He, X., Zemel, R. S., & Ray, D. (2006). *Learning and incorporating top-down cues in image segmentation*. Berlin: Springer.

Kumar, M. P., Torr, P. H. S., & Zisserman, A. (2005). OBJ CUT. In *CVPR*.

Kumar, S., & Hebert, M. (2005). A hierarchical field framework for unified context-based classification. In *ICCV*.

Lafferty, J. D., McCallum, A., & Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

Minka, T. P. (2003). *A comparison of numerical optimizers for logistic regression* (Technical Report 758). Carnegie Mellon University, Department of Statistics.

Mori, G., Ren, X., Efros, A. A., & Malik, J. (2004). Recovering human body configurations: combining segmentation and recognition. In *CVPR*.

Murphy, K., Torralba, A., & Freeman, W. (2003). Using the forest to see the tree: a graphical model relating features, objects and the scenes. In *NIPS*.

Opelt, A., Pinz, A., & Zisserman, A. (2006). Incremental learning of object detectors using a visual shape alphabet. In *CVPR*.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems*. San Mateo: Morgan Kaufmann.

Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., & Belongie, S. (2007). Objects in context. In *ICCV*.

Ren, X., & Malik, J. (2003). Learning a classification model for segmentation. In *ICCV*.

Schapire, R. E., & Singer, Y. (1999). Improved boosting using confidence-rated predictions. *Machine Learning*, *37*, 297–336.

Schroff, F., Criminisi, A., & Zisserman, A. (2006). Single-histogram class models for image segmentation. In *ICVGIP*.

Shental, N., Zomet, A., Hertz, T., & Weiss, Y. (2003). Learning and inferring image segmentations using the gbp typical cut. In *ICCV*.

Shotton, J., Winn, J., Rother, C., & Criminisi, A. (2006). TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV'06*.

Singhal, A., Luo, J., & Zhu, W. (2003). Probabilistic spatial context models for scene content understanding. In *CVPR*.

Sutton, C., & McCallum, A. (2005). Piecewise training of undirected models. In *UAI*.

Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., & Rother, C. (2008). A comparative study of energy minimization methods for Markov random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(6), 1068–1080.

Torralba, A. B., Murphy, K. P., & Freeman, W. T. (2004). Contextual models for object detection using boosted random fields. In *NIPS*.

Winn, J., Criminisi, A., & Minka, T. (2005). Object categorization by learned universal visual dictionary. In *ICCV*.

Winn, J., & Shotton, J. (2006). The layout consistent random field for recognizing and segmenting partially occluded objects. In *CVPR*.

Yang, L., Meer, P., & Foran, D. J. (2007). Multiple class segmentation using a unified framework over mean-shift patches. In *CVPR*.