Modelling Objects using Distribution and Topology of Multiscale Region Pairs

Himanshu Arora, Narendra Ahuja
Beckman Institute, University of Illinois at Urbana Champaign
Urbana, IL 61801, USA
{harora1,n-ahuja}@uiuc.edu

Abstract

We propose a method for simultaneous detection, localization and segmentation of objects of a known category. We show that this is possible by using segments as features. To this end, we propose an object model in which the image is represented as a tree, that captures containment relationships among the segments. Using segments as features has the advantage that object detection and segmentation is done simultaneously, forgoing the need for a separate sophisticated model for object segmentation. A generative model of an object category is estimated in a supervised mode, in terms of the characteristics of its constituent regions, their relative locations, and their mutual containment. The novel aspect of this work lies in simplifying the description of the hierarchy in terms of constraints that apply to only pairs of nodes, instead of all nodes in the tree. We show that this indeed improves the speed of learning algorithm. Inference is done using graph cuts. We report the performance of the model on standard datasets.

1. Introduction

This paper is concerned with the problem of automatic detection and segmentation of objects belonging to categories specified through examples contained in training images. It uses a generative object model, defined in terms of object regions occurring in a low level hierarchical segmentation of the image, or segmentation tree. The model contains information about relative locations of the regions contained in the object, their geometrical properties (size, shape, etc.), their topological properties (how many other regions are contained within a region), and their appearances. Instead of incorporating the large number of constraints that would be needed to capture the hierarchical structure of the large number of nodes in the segmentation tree, the modeling process is simplified by restricting the description to pairs of node. The model specifies joint properties of pairs of nodes related by different links (parent-child, siblings, etc.) These properties are estimated from the segmentation trees derived from prespecified example images of objects.

Segmentation tree contains the collection of all salient image regions along with their associated gray level contrasts, arranged in a geometric tree that captures their mutual containment information. The pairwise node constraints are captured in a topology feature vector which is extracted from the tree. This representation transforms the complex structure of the tree into one in the simpler vector space, which is easier to model. We show that fast learning of object model can be performed in a supervised setting. Recognition in new images is done via alternating optimization between estimating a segmentation using graphcuts, and estimating object configuration using fast Hough transform voting. The objects are thus represented using region hierarchies in a probabilistic framework. In this sense, our approach combines statistical and structural object representation methodologies.

The rest of the paper is organized as follows. In Sec. 2 we present a brief review of the existing literature on object category modelling. In Sec. 3, we describe the low level image representation we use. Sec. 4 describes the object model along with our learning and inference algorithms. In Sec. 5 we present experimental results, and finally conclude the paper in Sec. 6.

2. Related Work

Models of object categories proposed in literature can be categorized according to the types of low level image primitives, and the relationships among them, they use. Much recent work has focused on the use of locations of feature points, along with subimages in certain, e.g., elliptical, neighborhoods around the points, called intensity or texture patches, or simply patches. The relationships among these patches used range from modelling simple histograms of patch properties, while ignoring their locations [5] (bag-of-words model), to modelling the locations of these patches under different degrees of independence assumptions [8, 12, 19, 1] (constellation model, implicit shape model *etc*.). Performance of these approaches with

respect to object localization, namely, detecting the vicinity of the objects in new images varies according to the types of spatial relationships they capture. For instance, the bag-of-words model cannot localize objects, whereas the implicit shape model reports excellent localization performance. However, since there is no explicit modelling of the extent of the object, these models cannot delineate object boundaries, *i.e*. segment the object, for which they need more information, e.g., edges [11] and region homogeneity cues [13]. In addition to texture patches, local edge/contour fragments have also been used for representation [14], which again suffer from the same problems as mentioned above for patches.

Another class of models represents object categories as a 2D region (shape and intensities) [22, 11] which minimizes the cost of deformable matching of all available instances of the object. These methods return probabilities for each pixel of being inside the object instead of a hard segmentation. But, due to the large size of the model, these methods are computationally expensive. This restricts their application to relatively friendly environments where there is not enough change in scale or position of the object.

Relatively less attention has been directed towards using image segments for building an object category model. Using salient segments offers several advantages. First, since there significantly fewer segments than there are pixels, this reduces the number of model primitives, and therefore, the models complexity. Second, defining relationships among segments instead of pixels helps capture richer and more global image characteristics. Third, the usage of segments makes it possible to simultaneously detect and segment an object. There is no need for a separate sophisticated model for object segmentation, as is required by the local texture patch based methods. We use the hierarchical segmentation, or segmentation tree, based on the approach described in [2] as our basic image representation. This algorithm returns the collection of all salient image regions along with their associated gray level contrasts, arranged in a geometric tree that captures their mutual containment information. In our model, we use the relative locations of the segments, their mutual containment relationships (configuration in which they appear in the tree), and their gray level characteristics to define the object model.

The work on tree matching in computer vision can be broadly classified into two types. The first type establishes correspondences between similar nodes of the trees that also occur in the same topological environment [15, 21]. These approaches match the complete, global topological structure and are thus computationally expensive, e.g. the Max-Clique approach of [15] is NP-Hard. The second type uses a representation of the tree obtained by embedding individual nodes as points in a vector space, and analyzes local clusters of these points which capture local topology [6, 17]. Our approach is closer to the embedding approach, since we abstract the tree as a collection of local pairwise relations. While the global approaches yield fewer falsepositives because they consider the entire topology, local approaches accept partial matches and are therefore useful when, e.g, an object is partially occluded.

None of the above works use both image regions and their hierarchies for representing object categories. The closest to ours is the work in [20], which represents an object using segmentation trees. This algorithm attempts matches of the entire topological structure of segmentation trees across a set of training images, each of which may contain one or more examples of the object category of interest, to develop a canonical tree representation of the category. For detection, this canonical tree is matched to the segmentation subtrees of the new image. The tree matching algorithm used for learning has a very high complexity, $\mathcal{O}(M^2|V|^4)$, where M is the number of training images, and V is the number of regions in each image. In contrast, as we will see later, our algorithm has a lower complexity of $\mathcal{O}(M|V|^2)$ with no performance loss.

The contributions of our paper can thus be summarized as follows. 1) We match local pairwise relations between node pairs within a segmentation tree instead the entire trees, due to which fast learning and inference is possible. 2) We propose a novel probabilistic model for the process of generation of this embedding, and in this sense ours is a generative model. It is thus easily generalizable to include information other than the segmentation tree that we use in this paper. For instance, we can include local texture patches in our image representation, and use an existing generative model that describes the likelihood of these patches along with our model, to compute the joint likelihood of the combined observation of the segmentation tree and texture patches.

3. Image Representation

An image is represented as a tree of segments obtained by the algorithm proposed in [2]. The algorithm assigns to each image region a contrast value which is the minimum along its entire boundary. This results in a strict merger of regions as the contrast is increased, making a tree representation feasible. As one descends this segmentation tree, the region size decreases.

To represent the image, we include in the segmentation tree the individual properties of its regions. Each region $\boldsymbol{r}_v$ in the tree is described in terms of its position in the image $\boldsymbol{r}_v^l$, and an attribute vector $\boldsymbol{r}_v^a$. The position of the region is computed as the location of its center of mass $\boldsymbol{r}_v^l = (x_v, y_v)$. The attribute vector captures the shape and photometric properties of the region. The shape is described by the area a_v of the region, and a descriptor of the shape of the boundary of $\boldsymbol{r}_v$, used in [20]. This descriptor divides

the region into K pie slices around its center of mass, and computes the the region area in each slice to compute the shape histogram $h_v(k), k = 1, \ldots, K$. For rotation invariance, $k = 1$ is chosen to be aligned with the axis of the slice having the largest histogram value. We also use the entropy H_v of the shape descriptor h_v as an attribute. To capture the appearance of the region, we use the mean μ_v, and variance σ_v^2 of the gray level population in the region. We also compute moments of the spatial distribution of the gray levels around the center of mass, as additional attributes of spatial variation of intensities inside the region. We use the set of seven scale, translation and rotation invariant normalized central moments proposed in [18], denoted by the vector $\boldsymbol{\nu}_v$.

The relative location of a region with respect to its siblings in the tree is captured using a context vector $\boldsymbol{\phi}_v$ used in [20], which is obtained by computing a force of attraction on the region by its neighbors in spatial and attribute domains. We further introduce an attribute dispersion index which captures the the variability of the attributes of the children of a node. For this purpose, we focus on the mean gray level value μ_v and area a_v of the children, and compute the index as

$$\begin{aligned} d\mu_v &= \left[\tfrac{1}{N_v}\textstyle\sum_{u\in C(v)}(1-\mu_v/\mu_u)^2\right]^{1/2} \\ da_v &= \left[\tfrac{1}{N_v}\textstyle\sum_{u\in C(v)}(1-a_v/a_u)^2\right]^{1/2} \end{aligned}$$

where, $C(v)$ denotes the set containing children of node v, and $N_v = |C(v)|$, *i.e.* the number of children of node v. Note that the dispersion index here actually represents the root mean square deviation of the attribute of a node w.r.t. its children, normalized w.r.t. node attribute.

The complete attribute vector of a region can then be written as

$$r_v^a = (a_v, h_v(1), \ldots, h_v(K), H_v, \mu_v, \sigma_v^2, \boldsymbol{\nu}_v, \boldsymbol{\phi}_v, da_v, d\mu_v)$$

This attribute vector captures properties of individual regions. We now describe how we represent the relative arrangement of these regions in the segmentation tree. Instead of considering the relationships among all regions simultaneously, e.g., in [20], we do so for only pairs of regions. The topological structure of the segmentation tree is compressed into pairwise relations between its constituent regions. This reduces the complexity of representation from that corresponding to subtree matching to matching only pairs of nodes. Different types of relations can be used to select the node pairs, *e.g.* parent-child, siblings *etc*. We index each type of relation by an integer p. Corresponding to each relation p, a feature vector $\boldsymbol{t}_p$ is extracted containing ordered pairs of regions indices which follow this relation. For instance, let us suppose that $p = 0$ denotes the parent child relationship, and in the observed segmentation tree, region r_1 occurs as a parent of region r_2. Then the topology feature vector $\boldsymbol{t}_0$ contains the ordered pair $(1, 2)$. We can write this topology vector as

$$\boldsymbol{t}_p = \{(u_1^p, v_1^p), \ldots, (u_n^p, v_n^p)\} \tag{1}$$

where u_i^p and v_i^p are region indices in the segmentation tree. By representing the segmentation tree as a set of topology vectors above, we map the structure present in the tree into a simpler, vector space. These vectors can be viewed as features extracted from the tree.

There has been a lot of work towards representing tree structures in the literature on Natural Language Processing [3], where trees are used to depict the dependencies among the different parts-of-speech in a sentence [4] which are represented by nodes in the tree. A method for representing these trees found to be robust in speech processing is by counting the number of occurrences of different types of subtrees, each depicting a type of relation [4]. Since the set of values that a node can take (*i.e.* parts-of-speech) is finite, the types of subtrees that can exist is countable, and since the only property of interest for a subtree is its presence or absence, counting the number of occurrences makes sense. Our representation above is similar to this representation, since we store all occurrences in the tree of a particular relation (analogous to speech subtrees). Note however that the set of attributes that a region can take in the segmentation tree is real valued and therefore infinite. Hence just counting will not suffice. Because of this reason, in our representation, we separately store each individual occurrence in the topology vector. Topology need not be limited to pairwise relationships, as used in this paper. Other relationships can be explored.

4. Model

As discussed in Sec. 3, an image I is represented in terms of its regions $\boldsymbol{r}_v = (\boldsymbol{r}_v^l, \boldsymbol{r}_v^a)$, and topology vectors $\{\boldsymbol{t}_p\}_{p=1}^P$. In this section, we explain our model for the case when there is only one object instance in the image; the extension to multiple instances is straightforward. We assume that the object edges are captured in at least one of the regions in the segmentation tree. This assumption is reasonable since object boundaries usually have considerable contrast with the background. With each region, we associate a label l_v denoting whether region $\boldsymbol{r}_v$ is inside the object or not. Here, $l_v = 1$ if $\boldsymbol{r}_v$ is fully contained inside the object and $l_v = 0$ otherwise. The collection of regions with labels equal to 1 thus gives us an object-background segmentation. Also, with each object instance, we associate an object configuration vector $oc = (o_l, o_s)$, where o_l is the location of the object and o_s is its scale. As our generative model, we can write the likelihood of regions $R = \{\boldsymbol{r}_v\}$ and topology $T = \{\boldsymbol{t}_p\}$ of an observed image I given a labelling $L = \{l_v\}$ and object configuration oc as follows

$$P(T, R|L, oc) = P(T|R, L)P(R|L, oc)$$

The first term on the right of the above expression denotes the likelihood of observing a certain topology of regions (arrangement in the tree) given their positions in the image and their attributes. We assume that this is independent of the object configuration. The second term denotes the likelihood of individual region attributes and locations given a labelling and object configuration. Next, we describe how we model the above two terms.

For concise representation and noise reduction, we first make a dictionary of region attributes. We do this by clustering region attributes into K clusters using a gaussian mixture model. Each region attribute $\boldsymbol{r}_v^a$ thus has a probability of being generated by k-th cluster, given by $P(\boldsymbol{r}_v^a|c_v = k)$. Here, c_v is the random variable representing the cluster from which $\boldsymbol{r}_v^a$ is generated. Towards generating a region v given its label and the object configuration, first an attribute cluster is chosen according to the probability $\pi^{l_v}(k) = P(c_v = k|l_v)$. Given this attribute cluster, the region attribute $\boldsymbol{r}_v^a$ and region location $\boldsymbol{r}_v^l$ are assumed to be generated independently. Region attribute is generated according to the known density $P(\boldsymbol{r}_v^a|c_v = k)$ which does not depend on the region label l_v . Region location is generated according to the density $f_{loc}^{l_v}(\boldsymbol{r}_v^l|k, oc) = P(\boldsymbol{r}_v^l|c_v = k, l_v, oc)$. We assume that all regions are generated independently given their individual labels, and are independent of the labels of other pixels. We thus have

$$P(R|L, oc) = \prod_v \sum_k f_{loc}^{l_v}(\boldsymbol{r}_v^l|k,oc) P(\boldsymbol{r}_v^a|c_v=k)\pi^{l_v}(k) \qquad (2)$$

Modelling the joint likelihood of the entire topology vector would have high complexity. Hence we assume that each ordered pair of regions in a topology vector $(u, v) \in \boldsymbol{t}_p$ is generated independently. Note however that since the topology vector depends on pairs of regions, this independence assumption is not very restrictive. For instance, consider two regions $\boldsymbol{r}_u$ and $\boldsymbol{r}_v$ such that the ordered pair (u, v) does not appear in any topology vector, *i.e.* $(u, v) \notin \boldsymbol{t}_p \forall p$. This does not mean that $\boldsymbol{r}_u$ and $\boldsymbol{r}_v$ are independent, since there might exist ordered pairs $(u, w) \in \boldsymbol{t}_p$ and $(v, w) \in \boldsymbol{t}_q$ for some p and q which will induce a relation between $\boldsymbol{r}_u$ and $\boldsymbol{r}_v$ through $\boldsymbol{r}_w$. To write the likelihood of topology vectors, we further assume that each ordered pair depends only on the region attributes and labels of the regions in that pair. We thus have

$$P(T|R, L) = \prod_p \prod_{(u,v)\in \boldsymbol{t}_p} P((u,v)\in \boldsymbol{t}_p|\boldsymbol{r}_u^a, \boldsymbol{r}_v^a, l_u, l_v)$$

Further marginalizing over the cluster center associated with region attributes, we get

$$P((u,v)\in \boldsymbol{t}_p|\boldsymbol{r}_u^a, \boldsymbol{r}_v^a, l_u, l_v) = \sum_{k_1,k_2} f_{top}^{(l_u,l_v)}(k_1,k_2,p) \cdot P(c_u=k_1|\boldsymbol{r}_u^a) P(c_v=k_2|\boldsymbol{r}_v^a), \text{ where,} \qquad (3)$$

$$f_{top}^{(l_u,l_v)}(k_1,k_2,p) = P((u,v)\in \boldsymbol{t}_p|c_u=k_1, c_v=k_2, l_u, l_v)$$

4.1. Learning

During training, we want to learn the following parameters of the above model, given their ground truth labels:

1. the attribute cluster prior probabilities, $\pi^{l_v}(k)$, where $\sum_k \pi^{l_v}(k)=1$
2. the per attribute location distributions, $f_{loc}^{l_v}(\boldsymbol{r}_v^l|k,oc)$, where, $\sum_{\boldsymbol{r}_v^l} f_{loc}^{l_v}(\boldsymbol{r}_v^l|k,oc)=1$ and
3. the per attribute pair topology distributions, $f_{top}^{(l_v,l_u)}(k_1,k_2,p)$.

Let us suppose that we have N training images with extracted set of regions R^n, topology vectors T^n, and ground truth labels L^n, where $n = 1, \ldots, N$. From the ground truth labels, we can compute the object configurations oc^n for each image. The above parameters can thus be learnt by maximizing the log-likelihood $\sum_n \log(P(T^n|R^n, L^n)) + \log(P(R^n|L^n, oc^n))$.

The maximum likelihood solution of the above problem does not have a closed form. Iterative algorithms such as Expectation Maximization can be used to obtain a local maximum. These algorithms however involve computation of posterior densities and have very high complexity. We thus adopt an approximation strategy, wherein we replace the region attributes with the values of the cluster center closest to them in the region attribute dictionary. For each region attribute $\boldsymbol{r}_v^n$, we thus know the associated cluster c_v^n. The summations in Eq. (2) and (3) then reduce to single terms and maximization is easy. Empirically, this modification is a good approximation since we have observed that the posteriors of region attributes over the dictionary are peaky, *i.e.* the regions are usually associated with only one cluster center in the dictionary. Using this approximation, the parameters can be computed as

$$\begin{aligned}
\pi^l(k) &= \frac{\#\{c_v^n:c_v^n=k, l_v^n=l\}}{\#\{c_v^n:l_v^n=l\}} \\
f_{loc}^l(\boldsymbol{r}|k,oc) &= \frac{\#\{\boldsymbol{r}_v^{ln}:(\boldsymbol{r}_v^{ln}-o_l^n)/o_s^n=\boldsymbol{r}, c_v^n=k, l_v^n=l\}}{\#\{\boldsymbol{r}_v^{ln}:c_v^n=k, l_v^n=l\}} \\
f_{top}^{(l,m)}(k_1,k_2,p) &= \frac{\#\{(c_u^n,c_v^n):(c_u^n,c_v^n)=(k_1,k_2),(l_u^n,l_v^n)=(l,m)\}}{\#\{(c_u^n,c_v^n):(l_u^n,l_v^n)=(l,m)\}}
\end{aligned}$$

The location distribution estimated above is discrete. To estimate the continuous distribution over region location $\boldsymbol{r}$ given the cluster center k and object configuration oc, we use a parzen windows estimate obtained using training regions with label l and cluster center k.

The complexity of the learning algorithm can be computed as follows. Let us suppose that we have M training images, each having V regions. Then clearly, π^l and f_{loc}^l can be computed in $\mathcal{O}(MV)$, *i.e.* the total number of regions present in the dataset. The topology parameter $f_{top}^{(l,m)}$ can be computed in $\mathcal{O}(MV^2)$, which is the dominant term in complexity. Hence, the complexity of our learning algorithm is $\mathcal{O}(MV^2)$. Note that this is much lower when compared to $\mathcal{O}(M^2V^4)$ complexity of [20].

4.2. Inference

In this section we describe our algorithm to infer a Maximum-a-Posteriori (MAP) labelling and object configuration in a new image with an object instance in it. As our MAP estimate, we want to maximize the posterior $P(L, oc|T, R) \propto P(T, R|L, oc)P(L, oc)$. Here, the likelihood $P(T, R|L, oc)$ can be computed using the learnt parameters of the model. We assume that in the prior distribution $P(L, oc)$, labels and object configuration are independent. For object configuration, we use a uniform prior. For the labels, we use a Markov Random Field (MRF) prior on the tree structure, which means that the regions connected in the segmentation tree are apriori likely to be labelled same. The MAP estimate can thus be written as the minimizer of negative log-likelihood as follows:

$$(L^*, oc^*) = \underset{L,oc}{\operatorname{argmin}} -\mathcal{L}(L, oc) = \underset{L,oc}{\operatorname{argmin}} \textstyle\sum_u \phi(\boldsymbol{r}_v|l_v, oc) + \sum_{(u,v)\in T} \psi_1(\boldsymbol{r}_u, \boldsymbol{r}_v, l_u, l_v) + \sum_{(u,n)\in\mathcal{N}} \psi_2(l_u, l_v) \quad (4)$$

Here, the unary term ϕ is computed using individual region likelihoods, which describes whether a region is more likely to be labelled as object or background. The interaction term ψ_1 is computed using the topology distributions. This describes whether a region pair (u, v) observed in a particular topology is likely for a corresponding labelling (l_u, l_v). The interaction term ψ_2 represents the MRF prior on region labels, indicating that the regions connected together in the observed segmentation tree are more likely to have same labels. We use $\psi_2(l_u, l_v) = \delta(l_u \neq l_v) \exp(c|l_u - l_v|)$.

Alternating minimization. To minimize the negative log-likelihood in Eq. (4), we use an iterative algorithm, which alternates between minimizing over L and oc. This can be concisely written as

$$oc^{(t)} = \operatorname{argmin}_{oc} -\mathcal{L}(L^{(t-1)}, oc) \quad (5)$$

$$L^{(t+1)} = \operatorname{argmin}_{L} -\mathcal{L}(L, oc^{(t)}) \quad (6)$$

Since oc only appears in the unary term in Eq. (4), the minimization in Eq. (5) can be easily performed by voting the log-likelihoods due to regions labelled as objects onto oc-space. To perform the minimization in Eq. (6) over L, following the formulation in [10], we use graph cuts. This is under the assumption that the interaction potentials ψ_1 and ψ_2 are regular. Furthermore, since the alternating algorithm always seeks a global minimum in each iteration, following [9], it converges.

Initialization. To initialize the algorithm above, we use the local minima of the unary term ϕ. Given an object configuration oc, a region is labelled as either object or background on the basis of its likelihood due to either label. The likelihood for oc is computed by adding these log likelihoods. The first L (=10) maxima in the oc space are selected as the initial hypotheses $oc^{(0)}$.

Multiple Instances. The alternating minimization procedure on each of the multiple initial hypotheses results in multiple segmentations. These segmentations are pruned in a greedy fashion according to their areas of overlap and likelihoods. First the segmentation with the highest likelihood is selected. Then the segmentation with next best likelihood, having less than 25% overlap with the existing segment is selected. This process is repeated till the all hypotheses or the entire image area is exhausted.

Invariances. Since we return object location and scale, our algorithm is invariant to translation and scaling. Note that the region attribute vector and the topology vector described in Sec. 3 are rotation invariant. The region location however changes for different rotations of the object. The algorithm can thus be made rotation invariant by maximizing the likelihoods over different hypothesized rotations. Since the distribution of the locations of all regions within the object is used, which changes with object's rearticulation, our algorithm is not articulation invariant. Due to the usage of local pairwise relations, our algorithm is robust to occlusions.

5. Results

We evaluate our algorithm on carefully chosen datasets with respect to a) the impact of using topology information, b) localization performance, c) segmentation accuracy, and d) performance relative to existing techniques.

5.1. Experimental Setup

Datasets. To test our algorithm w.r.t. detection and segmentation accuracy, we require datasets with given groundtruth segmentations. We test our algorithm on Faces, Cars (sideviews) and Motorbikes. For faces (435 images) and motorbikes (798 images), we use the Caltech-101 dataset [7] which has ground truth segmentations. We use a randomly selected set of 200 images as our training set to learn the model parameters as described in Sec. 4.1. Testing is done on all remaining images. For training the model on cars, we use the Caltech-101 dataset (123 images). We do not use this set for testing since it is not very challenging because of the following reasons. First, each image contains only a single car, and second, the car lies at the center of the image, occupying most of its area. The test images for cars are obtained by the online image annotation tool *LabelMe* [16]. For this, an initial set of images is selected by querying side views of cars on the *LabelMe* website. Of these, the images which contain at least one car occupying more than 10% of image area are selected as test images. We choose this threshold on area because very small objects do not allow many tree levels, and therefore, significant topological information. The test set thus obtained contains 97 images with a total of 107 instances of a car. The dataset has scale

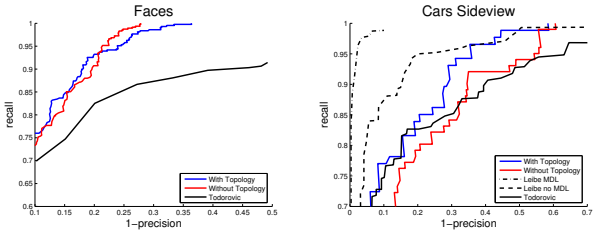
variation and images with multiple cars (see Fig. 2). We do not use the UIUC-Cars dataset [1] which has been extensively used in literature, since, 1) it does not have ground truth segmentation masks which we need for evaluation, and 2) the images in this dataset are very low resolution for the segmentation algorithm to capture the topology. All the images are resized so that the largest dimension is 400 pixels.

Evaluating Localization Accuracy. To test the localization accuracy of our algorithm, we follow the evaluation strategy described in [20], which in our case is executed as follows. The algorithm explained in Sec. 4 outputs the area occupied by the detected objects along with their associated likelihoods. For normalization across different images, we divide the likelihoods of detections in an image by the total number of regions in that image. The detection algorithm compares the normalized likelihood $\mathcal{N}_i$ for the i-th detection against a detection threshold τ and returns a positive detection if $\mathcal{N}_i > \tau$ and a negative detection otherwise. Thus, as the threshold is increased, the number of positive detections returned by the algorithm decrease. The ground truth labels for the i-th detection is obtained by comparing the area occupied by this detection in the image, against the available ground truth segmentation. We say that a detection is a ground truth negative if either a) the area of intersection between ground truth segmentation and that estimated by the algorithm is less than 75% of the ground truth segmentation area, or b) more than 25% of the detection area lies outside the ground truth segmentation. All other detections are labelled as ground truth positive. Note that this criteria are stricter than the overlap between bounding rectangles used in [12, 1], since we account for the actual shape of the object. We measure the performance in terms of the standard precision-recall plots obtained as the detection threshold τ is changed. Here, precision is measured as the percentage of true-positives out of the total number of positives returned by the algorithm, and recall is measured as the percentage of true-positives out of the total number of ground truth positives. An algorithm is deemed to be better than another if its recall vs. 1-precision plot lies strictly above that of the other.

Evaluating Segmentation Accuracy To test the segmentation accuracy of our algorithm, we use the standard pixel-label mismatch percentage used in the literature [22, 13]. From the ground truth segmentation, we have labels for whether a pixel belongs to an object or background. The detection algorithm also returns a similar label for each pixel in the image. We report the percentage of mislabelled pixels per image, averaged over the test set.

5.2. Experiments

Significance of Topology. To evaluate the importance of topology information, we first compare the performance of our algorithm *with* and *without* using topology vector $\boldsymbol{t}_p$ (Eq. (1)) in our representation. For the model *with* topology, we use the inference mechanism described in Sec. 4.2. To perform inference on the model *without topology*, we remove the interaction potential ψ_1 in Eq. (4) and proceed with the inference algorithm. Fig. 1 shows the precision recall curves for the faces and cars. The curve for the model *with* topology is shown in blue and for *without* topology is shown in red. As expected, including topology information improves the localization performance. Note however that this increase is more significant for cars, than for faces. This is because cars usually have a lot of variation in color and shape as compared to faces, leading to deeper segmentation trees and hence increased benefits of topology information.



(a) **Faces** (b) **Cars - side view**

Figure 1. **Precision-Recall** curves for **(a)** Faces, and **(b)** Cars. **Red plots** - performance when only individual region attributes are used. **Blue plots** - performance when both the individual region attributes and topology information is used. We perform better than the unsupervised approach of [20]. For cars, our performance is close to the state-of-the-art performance reported in [12].

We compare our results with those reported in literature. As can be seen in Fig. 2, we perform better than the the technique in [20], which uses the same segmentation tree representation as ours, for both cars and faces. Note however that [20] is an unsupervised approach whereas ours is supervised, *i.e*. we have a friendlier learning environment. We obtain an equal error rate performance of 91.7% for cars and 81.1% for faces as compared to 87.5% and 78.3% respectively, reported in [20]. Our algorithm has an inferior performance when compared against the best reported performance in literature, that of the local texture patch based algorithm in [12]. This algorithm however uses a model complexity cost (MDL) to prune detections for handling multiple instances. The same algorithm without MDL produces an equal error rate performance of 91%, which is comparable to our performance. Also note that our evaluation criteria are stricter than [12], which reports the bounding box detection results and not the segmentation area overlap.

Segmentation Accuracy. Table 1 shows the comparison of pixel based segmentation error for different test sets. First note that topology improves results. When compared against the results reported in literature, for cars, we perform better than the state-of-the-art performance of the pixel based algorithm presented in [22]. For faces, our performance is worse as compared to the approach in [20].

	Without Topology	With Topology	Todorovic [20]	LOCUS [22]
Cars	4.5	4.1	9.3	6.0
Faces	17.9	11.5	6.8	-
Motorbikes	40.8	28.6	-	-

Table 1. **Segmentation error** comparison with other techniques. The inclusion of topology improves results. Also, the results are better than those reported in the literature.

The reason is that as compared to cars and motorbikes, the changes in region hierarchy across images are small, which preserves the tree structures and helps the approach of [20] to obtain better results compared to our approach which matches only pairwise relations. Fig. 2 shows some example segmentations on our test sets. These results are obtained for the value of threshold τ at the equal error rate. For each pair of rows, at the top is the original image, and at the bottom is the estimated segmentation. First note the complexity of cars datasets which has huge scale variations, multiple instances and partial occlusions. Fig. 2(a) and (b) show detection under huge scale variations of approximately 8x area change. Fig 2(c) and (d) show detections of multiple instances and under partial occlusions of approximately 10% of the area of the car. Note that the boundaries are not correctly detected for many dark colored cars. This is because these cars do not have enough contrast w.r.t. the background, and hence the low level segmentation fails to detect its boundaries. Figs 2(e) and (f) show results for motorbikes and faces.

6. Conclusions

We have presented a statistical model to describe objects using a tree of image segments. The novel aspect lies in modelling the relationships between segments that occur in the tree. We show in the results that the rich topology information present in the segmentation is indeed useful for detection. In addition, detection performance of the algorithm compares well with the state-of-the-art methods, indicating that image segments based representation is useful for modelling objects. To model the topology in the tree, currently we only use pairwise relations. In the future, we plan to investigate how the performance changes as we include more and higher-order relations. We also plan to enhance the model to incorporate other image based information, such as image gradients, which aids in object segmentation.

Acknowledgment

The support of the Office of Naval Research under grant N00014-06-1-0101 is gratefully acknowledged.

References

[1] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Trans. PAMI*, 26(11):1475–1490, November 2004.

[2] H. Arora and N. Ahuja. Analysis of ramp discontinuity model for multiscale image segmentation. In *Proc. Intl. Conf. on Patt. Recog.*, volume 26, pages 99–103, 2006.

[3] R. Bod. *Beyond Grammar: An Experience-Based Theory of Language*. CSLI publications/Cambridge University Press, 1998.

[4] M. Collins and N. Duffy. Convolution kernels for natural language. In *Proc. NIPS*, Cambridge, MA, 2002. MIT Press.

[5] G. Csurka, C. Dance, L. Fan, and C. Bray. Visual Categorization with Bags of Keypoints. In *Workshop on Stat. Learning in Comp. Vision, European Conf. on Comp Vision*, pages 1–22, 2004.

[6] M. F. Demirci, A. Shokoufandeh, Y. Keselman, L. Bretzner, and S. Dickinson. Object recognition as many-to-many feature matching. *Intl. Journal of Comp. Vision*, 69(2):203–222, 2006.

[7] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *IEEE. CVPR 2004, Workshop on Generative-Model Based Vision.*, 2004.

[8] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. IEEE Conf. on Comp. Vision and Patt. Recog.*, volume 2, pages 264–271, 2003.

[9] A. Gunawardana and W. Byrne. Convergence theorems for generalized alternating minimization procedures. *Journal of Machine Learning Research*, 6:2049–2073, 2005.

[10] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Trans.PAMI*, 26(2):147–159, 2004.

[11] M. P. Kumar, P. Torr, and A. Zisserman. Objcut. In *Proc. IEEE Conf. on Comp. Vision and Patt. Recog.*, pages 18–25, 2005.

[12] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Proc. Workshop on Stat. Learning in Comp. Vision*, 2004.

[13] A. Levin and Y. Weiss. Learning to combine bottom-up and top-down segmentation. In *proc. ECCV (4)*, pages 581–594, 2006.

[14] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. In *Proc. European Conf. on Comp. Vision*, 2006.

[15] M. Pelillo, K. Siddiqi, and S. W. Zucker. Many-to-many matching of attributed trees using association graphs and game dynamics. pages 583–593, 2001.

[16] B. Russell, A. Torralba, K. Murphy, and W. Freeman. LabelMe: a database and web-based tool for image annotation. AI Lab Memo AIM-2005-025, MIT, Murray Hill, New Jersey, 2005.

[17] A. Shokoufandeh, L. Bretzner, D. Macrini, M. F. Demirci, C. Jonsson, and S. Dickinson. The representation and matching of categorical shape. *Comp. Vision and Image Understanding*, 103(2):139–154, 2006.

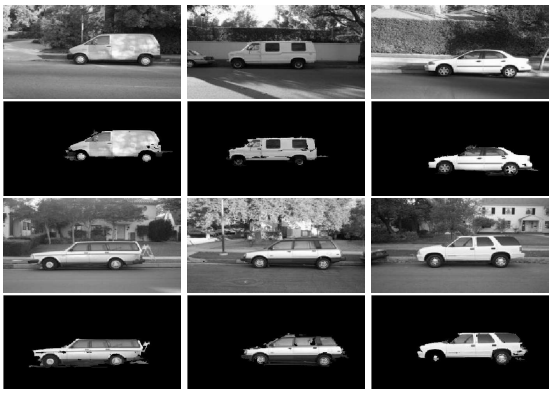
[18] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis and Machine Vision*. Chapman and Hall, 1993.

[19] E. Sudderth, B. Torralba, W. Freeman, and A. Willsky. Learning Hierarchical Models of Scenes, Objects, and Parts. In *in Proc. IEEE Intl. Conf. on Comp. Vision (ICCV)*, pages 1331–1338, 2005.

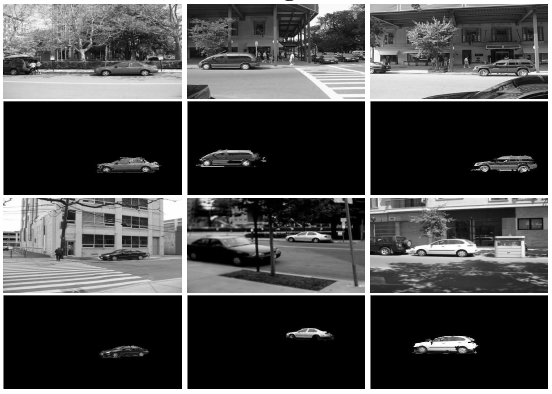
[20] S. Todorovic and N. Ahuja. Extracting subimages of an unknown category from a set of images. In *Proc. IEEE Conf. on Comp. Vision and Patt. Recog.*, pages 927–934, 2006.

[21] A. Torsello and E. R. Hancock. Computing approximate tree edit distance using relaxation labeling. *Patt. Recog. Letters*, 24(8):1089–1097, 2003.

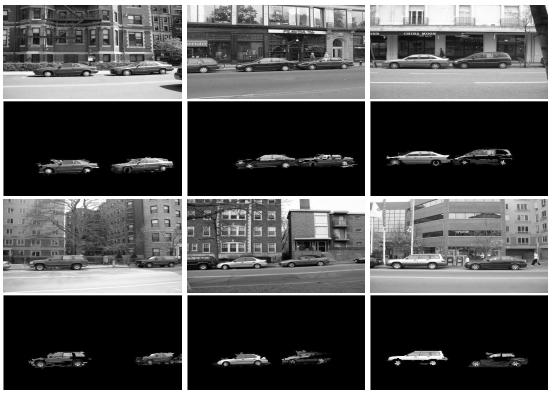
[22] J. Winn and N. Jojic. LOCUS: Learning Object Classes with Unsupervised Segmentation. In *in Proc. IEEE Intl. Conf. on Comp. Vision (ICCV)*, pages 756–763, 2005.



(a) **Cars - Higher scales**



(b) **Cars - Smaller scales**



(c) **Cars - Multiple instances**

(d) **Cars - Partial Occlusions**

(e) **Faces**

(f) **Motorbikes**

Figure 2. **Segmentation results.** Each pair of rows contain the original image at the top, and segmentation at the bottom. Figures **(a)-(d)** show results for cars. Images in **(a)** show results where the car occupies a large portion of image area (higher scale). Note the correct detections despite large scale variation with respect to images in **(b)**, where cars occupy very small image area. Figures **(c)** and **(d)** show results on images with multiple cars, and partially occluded cars. Figures **(e)** and **(f)** show results for faces and motorbikes datasets. Note that we get good pixel accuracy in segmentation along with detection.