Category level object segmentation by combining bag-of-words models and Markov Random Fields

Diane Larlus, Jakob Verbeek, Frédéric Jurie

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Category level object segmentation by combining bag-of-words models and Markov Random Fields

Diane Larlus — Jakob Verbeek — Frédéric Jurie

N° 6668

Octobre 2008

Thème COG

ISSN 0249-6399 ISRN INRIA/RR--6668--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
GRENOBLE - RHÔNE-ALPES

Category level object segmentation by combining bag-of-words models and Markov Random Fields

Diane Larlus , Jakob Verbeek , Frédéric Jurie*

Thème COG — Systèmes cognitifs
Équipes-Projets LEAR

Rapport de recherche n° 6668 — Octobre 2008 — 28 pages

Abstract: This paper presents an approach to segment unseen objects of known categories. At the heart of the approach lies a probabilistic model of images which captures local appearance of objects through a bag-of-words representation. Bag-of-words models have been very successful for image categorization; however, as they model objects as loose collections of small image patches, they can not accurately predict object boundaries. On the other hand, Markov Random Fields (MRFs), which are often used in many low-level application for general purpose image segmentation, do incorporate the spatial layout of images. Yet, as they are usually based on very local image evidence they fail to capture larger scale structures needed to recognize object categories under large appearance variations. The main contribution of this article is to combine the advantages of both approaches into a single probabilistic model. First, a mechanism based on a bag-of-words representation produces object recognition and localization at a rough spatial resolution. Second, a MRF component enforces precise object boundaries, guided by local image cues (color, texture, and edges) and by long-distance dependencies. Gibbs sampling is used to infer the model parameters and the object segmentation. The proposed method successfully segments object categories, despite highly varying appearances, cluttered backgrounds and large viewpoint changes. Through a series of experiments, we emphasize the strength as well as the limitation of our model. First, we evaluate the results of several strategies for building the visual vocabulary. Second, we show how it is possible to combine strong labeling (segmented images) with weak labeling (images annotated with bounding boxes), in order to limit the amount of training data needed to learn the model. Third, we study the influence of the initialization on the model estimation. Last, we present extensive experiments on four different image databases, including the challenging Pascal VOC 2007 dataset on which we obtain state-of-the art results.

Key-words: Image Segmentation, Visual Recognition, Markov Random Fields

* Université de Caen

Centre de recherche INRIA Grenoble – Rhône-Alpes
655, avenue de l'Europe, 38334 Montbonnot Saint Ismier
Téléphone : +33 4 76 61 52 00 — Télécopie +33 4 76 61 52 52

Combinaison d'un modèle sac-de-mots et d'un champ de Markov, pour la segmentation de catégories d'objets

Résumé : Ce rapport présente une approche pour la segmentation d'objets de catégories connues. Le cœur de l'approche réside dans un modèle probabiliste des images qui capture les apparences locales à travers une représentation par sac-de-mots. Les modèles par sac-de-mots se sont montrés très performants pour la catégorisation d'images ; cependant, comme ils considèrent les objets comme des collections non ordonnées de petites vignettes d'images, ils ne peuvent prédire avec précision la frontière des objets. Les champs de Markov, souvent utilisés pour différentes applications bas-niveau dans le cadre général de la segmentation d'images, utilisent la structure spatiale de l'image. Cependant, comme ils sont basés sur des évidences locales, ils ne peuvent capturer les structures à plus grande échelle qui sont nécessaires pour reconnaître des catégories dont l'apparence varie beaucoup. La principale contribution de ce rapport est la combinaison des avantages des deux approches préalablement citées en un seul modèle probabiliste. Tout d'abord, un méchanisme basé sur la représentation par sac-de-mots reconnaît l'objet et le localise à une résolution grossière. Ensuite, un composant de champ de Markov force la précision des frontières d'objet, guidé par des indices d'images locaux (comme la couleur, la texture, les contours) et par des dépendances à plus grande échelle. Un échantillonneur de Gibbs est utilisé pour l'inférence des paramètres du modèle et la segmentation des objets. La méthode proposée segmente avec succès les catégories d'objet, malgré de fortes variations d'apparence, un fond encombré et de larges changements de points de vues. À travers une série déxperiences, nous démontrons les avantages ainsi que les limitations de notre modèle. En premier lieu, nous évaluons les résultats de différentes stratégies pour la construction d'un vocabulaire visuel. Deuxièmement, nous montrons comment il est possible de combiner des annotations fortement supervisées (images segmentées) avec des annotations moins précises (images annotées avec des boites englobantes), de façon à limiter le nombre d'images d'apprentissage dont le modèle a besoin pour l'apprentissage. Troisièmement, nous étudions l'influence de l'initialisation sur l'estimation du modèle. Enfin, nous proposons des expériences complètes sur quatre bases d'images différentes, y compris la difficile base Pascal VOC 2007 sur laquelle nous obtenons des résultats comparables aux meilleurs méthodes.

Mots-clés : Segmentation d'images, reconnaissance visuelle, champ de Markov

Contents

Figure 1: Examples of object category segmentation produced with our method, obtained without any user interaction. Input images (cols 1,4), object category masks (cols 2,5), object category segmentation (cols 3,6).

1 Introduction

Still after several decades of research, image segmentation remains an open problem. Many different approaches have been investigated, combining various image properties such as color, texture, edges, motion, etc. Initially, these methods worked in an unsupervised way: without exploiting a database of manually segmented images to automatically tune parameters for optimal performance. Also, many of the methods operate in a 'bottom-up' way, generating the image segmentation by a process of aggregating local image information, and usually failing to capture high level image information. However, image segmentation is deeply related to image understanding, requiring long-range dependencies to resolve ambiguities that arise at a small scale.

The problem we address in this paper is that of generating accurate segmentations of object classes in images, without giving any prior information on object identities, orientations, positions and scales. This is also known as 'figure-ground segmentation': the task is to identify an object in an image and separate it from the background. Note that this differs from 'image segmentation'. Image segmentation, or scene segmentation, corresponds to the situation where everything in the image has to be segmented, whereas in object segmentation only several objects of interest have to be segmented.

We assume the objects to belong to known categories, and these categories are defined by sets of training images which are used to learn object appearance models. These training images play a fundamental role because object models build from these images allow object recognition. The recognition process drives the image segmentation process. In particular, we are interested in segmenting object categories that demonstrate large intra-class appearance variations.

Figure 1 shows several typical images, and corresponding segmentation results produced using our model. Starting from cluttered images including objects of interest, the method is able to recognize and localize objects, and to automatically produce segmentation masks that can be used to extract objects without any manual effort.

The main contribution of the model presented in this paper is that it takes advantage of two complementary components:

(a) an MRF component which ensures short-range spatially contiguity of the segmentation by aligning segment boundaries with low-level image boundaries,

(b) a bag-of-words object recognition component which performs localization of objects despite strong intra-class and imaging variations and which allows for longer range consistency in segmentation at a more semantic level.

In this paper we extend our preliminary work [15], with additional experiments, and an experimental evaluation of visual vocabulary construction methods for the proposed segmentation model.

In the rest of this article, we first review related works in Section 2. Then, in Section 3 we describe our model, and how we estimate its parameters. An alternative method to construct the visual vocabularies for the bag-of-words methods, based on decision trees, is presented in Section 4. We present our experimental results in Section 5, and conclude with a discussion in Section 6.

2 Related work

The development of image segmentation methods over time has shown the importance of integrating high level information into the segmentation process. Segmentation can be seen as a 'chicken-egg' problem, where object detection and recognition is required for accurate segmentation, and conversely accurate segmentation assists object detection and recognition. For this reason, we first present detection methods which have been applied to a wide variety of object categories. Secondly, we present methods which are primarily designed for segmentation and how they deal with long range dependencies.

The probabilistic Latent Semantic Analysis (pLSA) model [10] is a popular 'topic model' build on top of the bag-of-words representation that is efficiently fitted to data using a simple EM algorithm. Topic models consider the bag-of-words as a mixture of several 'topics', *i.e.* the visual words obtained from a visual scene can be modeled as a mixture of words belonging to background, and several objects.

Visual words are obtained by quantization of low level image descriptors. The quantization can be obtained in different ways, leading to different vocabulary types. Often, visual vocabularies are produced by a simple clustering algorithm such as k-means algorithm [4, 27], although hierarchical clustering [16] or mean-shift methods [11] are also used.

In our model, the visual vocabulary is used to discriminate between classes at the level of patches. If manually labeled patches are available, these can be used to design more discriminative vocabularies using *e.g.* the methods in [23, 14].

Alternatively, one can use decision trees to quantize the descriptor space [20]. The main attraction of this approach is that the assignment of image patches to visual words is very efficient due to the hierarchical structure of the decision tree. In Section 4 we describe vocabulary construction using decision trees in more detail, and in our experiments we evaluate vocabulary construction using clustering and decision trees.

Recently it has been shown [7] that the pLSA model, besides from being useful for image classification, can also be extended so that it can be used for object localization. This extended model can cope with fairly large variations in object appearance, as it is robust with respect to occlusions and to orientation

and scale changes, the latter being mainly due to the coarseness of the geometric structure of the model. The drawback of this model for object segmentation is that segmentation is obtained only up to a bounding box.

Variations of topic models defined using Dirichlet processes, and including Gaussian distributions over the spatial locations of the visual words have also been proposed [29]. Each Gaussian may be interpreted as a cluster of visual words associated with a single object. Dirichlet processes allow automatic selection of model complexity, *i.e.* the number of clusters or objects in the scene. Although modeling the object shapes with Gaussian distributions does allow for reasonable object localization, the resulting segmentation is very rough as the model does not enforce crisp boundaries between objects.

Cao *et al* [3] tried to overcome this limitation with a spatially coherent latent topic model. Their representation of images makes use of the association between segmented homogeneous regions and visual words in those regions. The regions are generated using a generic over-segmentation method, that will cut objects in many regions, but yields regions that are often not crossing any object boundaries. The class of each region is computed from visual words found within the region. The advantage is that regions provide a good starting point for object segmentation and reduce the number of elements to deal with in the segmentation process. However, there is a trade-off between the size and the quality of the over-segmentation. Less regions leads to faster processing and less risk to introduce erroneous object when labeling the regions. On the other hand, having only few regions increases the risk that some of the regions will have object boundaries inside them, which are errors which cannot be recovered at a later stage. Our approach is not subject to this trade-off, as we do not rely on preprocessed low-level image segmentation.

Markov Random Fields (MRFs) [8], and variants like Conditional Random Fields (CRFs) [13, 26, 32], and Discriminative Random Field [25], have a long history in image segmentation. Typically, they define probability distributions over the labels of pixels, or more coarsely over image patches. These distributions follow an intuitive conditional independence relation: the label of one pixel (or patch) is independent of all other labels given the labels of it neighboring pixels (or patches) in the image. The neighborhood relation is often defined as the regular 4 or 8 neighborhood system over a rectangular grid of pixels or patches over the image. In image labeling applications, the parameters of the random field often implement a strong positive correlation between the labels of neighboring sites. The random field distribution is then combined with local evidence from the image; *e.g.* the visual words associated with a patch will increase the likelihood of having a certain class at that location in the image. The positive correlations of random field model can resolve ambiguities that arise in bag-of-words models, by propagating evidence for certain labels spatially over the image.

Shotton *et al* [26] propose to use CRFs to learn a discriminative model of object classes, incorporating appearance, shape, and context information. Our model is quite similar to theirs, even though they do not consider a generic background category but rather have different background sub-classes, like 'grass', 'sky', 'road', etc. The main difference is that we model explicitly each separate object instance, allowing us to incorporate instance specific appearance models, in addition to the class level appearance.

Dealing with the same task of scene interpretation, Verbeek and Triggs [31] proposed to combine a MRF, for local dependencies, with a topic model at the image level, for global interaction. As compared to a standard topic model like pLSA, their model generated much crisper object category segmentation. As compared to a standard MRF, the topic model suppresses small regions that are labeled with a category that does not appear elsewhere in the image. However, their model produces relatively coarse segmentation results, as it does not use local gradient and color structure in the images to guide the category boundaries.

In a similar spirit, a combination of MRFs and Dirichlet process mixture models was proposed in [22]. Their model is an unsupervised Dirichlet process mixture model, that allows automatic selection of the number of mixture components, and uses the MRF structure to enforce spatially contiguous assignment of image pixels to mixture components. The model was applied to unsupervised segmentation of SAR, RADAR, and MR images. In our model we use the same principle, but we use the mixture components to represent instances of known object categories that we want to segment.

Winn and Shotton [34] proposed the 'layout consistent random field' that, like our model, models individual object instances, but also explicitly models object occlusions. The occlusion reasoning is achieved by modeling the internal layout of objects, rather than modeling the objects internally as a bag-of-words. However, in its current form the model allows only for a limited variability in scale, and a single object category (but multiple instances) per image.

Simultaneously, in another line of research aiming at user-interactive tools for graphics applications, remarkable object segmentation algorithms based on MRFs have been proposed, *e.g.* [24, 17]. These methods require a user to give a rough indication of the object of interest and the background positions, by giving a bounding box or using a brush-like tool in the image. In [24], the key idea is to model image foreground and background color distributions using a mixture of Gaussians (MoG). These distributions are iteratively re-estimated, and after each iteration a graph-cut energy minimization is performed to separate the image pixels between foreground and background. The MRF energy function value for a given foreground/background label image depends on (a) the similarity of nearby pixels that have different labels, and (b) the likelihood of pixel colors under the mixture models over foreground/background colors. With these interactive algorithms quite accurate segmentation results can be obtained, and the next step is now to eliminate the user interaction. Our goal is to segment objects from images by only specifying the object category (*e.g.*segment out all individual sheep in an image).

We end our discussion of related work by discussing several papers on the use of shape models for object segmentation. Kumar *et al* [12] proposed a methodology to combine CRFs and pictorial structure (PS) models. The CRF part provides figure/ground segmentation, whereas the PS part encourages the CRF to follow the object shape.

Leibe and Schiele [16] use hand segmented images to learn the relations between segmentation masks and visual codebook entries. Their 'implicit shape model' allows to localize objects and to segment images combining the local segmentation masks corresponding to visual words. A voting process in a Hough-space of the object location, rotation, and scale is used to obtain a consistent

set of local features that agree on the object segmentation, and filter out noisy erroneous local features.

In [1], the authors propose a method to do figure/ground segmentation that is shape specific and texture invariant. A multi-scale bottom-up segmentation is combined to shape templates for producing the final segmentation. Again this method heavily relies on the initial segmentation.

In [33], object category shape and appearance is learned from a set of training images, and new objects are segmented by fitting a deformed version of this model.

Although they are robust to small local shape variations, the strong geometric constraints embedded into all these shape models are not well adapted to model the complex appearances of weakly structured object classes. Examples of these complex appearances can be found in Figure 5, for the classes cats and people. Such classes require more flexible models.

3 A combined MRF and bag-of-words segmentation model

In our model we represent images as a collection of patches of a fixed size extracted on the nodes of a regular grid. We suppose the image patches are generated by a number of objects and a background; we use simple Gaussian and uniform models for their spatial extent, and refer to both objects and background as 'blobs'. In each image both the number of blobs and their characteristics (position, size, and shape) are unknown. We associate a blob label with each patch, and define a Markov Random Field structured energy function over them to encode the short-range correlations among them. Through the category labels of blobs, we also associate category labels with the patches. Once the model parameters have been estimated from labeled training images, we can use a Gibbs sampler to estimate the category labels of patches in new, unlabeled, images.

In the remainder of this section we first describe the visual feature extraction procedure in more detail. Then, we present our model, describing its two components in turn in Section 3.2 and Section 3.3. We conclude the section by discussing the per-image model estimation procedure in Section 3.4.

3.1 Visual feature extraction

From each image we extract two types features: a set of n overlapping patches, and a 'boundary map'. Each image patch $\mathcal{P}_i, i \in \{1, \ldots, n\}$, is defined as a square image region of fixed size, and we compute the following four characteristics:

1. the SIFT descriptor [18], coded by the corresponding visual word w_i^{sift},
2. the hue descriptor of [30], coded using the corresponding color word w_i^{color},
3. the average RGB value of pixels in the center of the patch, denoted by rgb_i,
4. the coordinates of the patch center $X_i = (x_i, y_i)$ in the image.

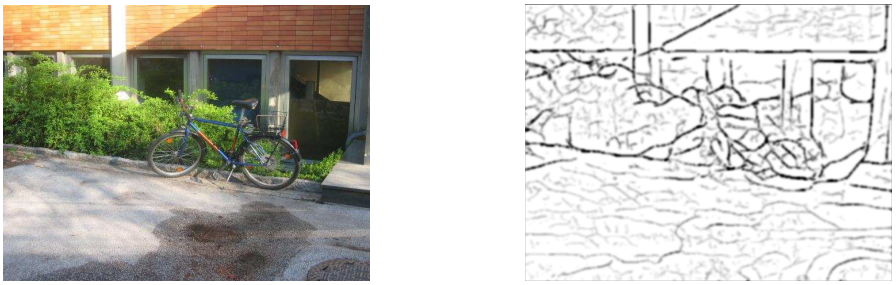


Figure 2: An example image from the Graz database and its boundary map.

In addition to the patch based characteristics, we also extract a boundary map $\mathcal{G}$ that gives an estimate of the probability of finding a boundary between image segments at each pixel location (x, y). The map is based on characteristic changes in several local cues associated with natural boundaries [19]. See Figure 2 for an example of an image and its boundary map.

3.2 A Dirichlet process mixture model over patch characteristics

In this section we present a generative model for rough object/background segmentation. We use a model inspired by [29] with explicit spatial structure information: we consider that an image is made of regions that we call 'blobs'. Each blob generates the characteristics of the patches associated with that blob, where the distribution over patch characteristics depends on the parameters associated with the blob. Intuitively, if an image contains three objects, say a car, a pedestrian and a bike, we may have four blobs: one corresponding to each object, plus an additional for the background.

Given the blobs and their parameters, the patches $\mathcal{P}$ in an image are assumed to be independent. The generative process for a patch is as follows: (i) select a blob, and (ii) draw the patch characteristics from the distribution associated with the blob. The remainder of this section details this generative process.

Dirichlet processes [21] exhibit the so-called clustering property: the more often a given value has been sampled in the past, the more likely it is to be sampled again. The Dirichlet process can be seen as the limit as K goes to infinity of a finite K-component mixture model. Note that even for a mixture with an infinite number of components, with any finite sample from the mixture we can only associate finitely many of the mixture components. In our case, the blobs will take the role of mixture components, and note that we do not know or fix their number in an image in advance. This means that for each newly sampled patch, it can be either sampled from one of the finitely many blobs that have been used before, with probability $\frac{N_k}{n-1+\alpha}$ where N_k is the number of samples already drawn from the particular blob, and n is the number of samples including the current one. Alternatively, the patch can be sampled from a new blob with a probability $\frac{\alpha}{n-1+\alpha}$, where α is the concentration parameter of the Dirichlet process. These probabilities will be called p_{dir} in the next section.

With each blob B_k we associate a set of parameters: $\Theta_k = \{\mu_k, \Sigma_k, C_k, l_k\}$. The density over the spatial positions X_i of associated patches is given by a Gaussian $p(X_i|\Theta_k) = \mathcal{N}(X_i, \mu_k, \Sigma_k)$. The category associated with the blob is denoted l_k, and C_k denotes the parameters of a mixture of Gaussian (MoG) model over the color vectors rgb_i of the associated patches. The background is defined by a color distribution C_{bg}; its spatial model is uniform over the image area.

In addition to the observed characteristics $\mathcal{P}_i = \{w_i^{sift}, w_i^{color}, rgb_i, X_i\}$, we associate two random variables, b_i and c_i, with each patch. The index of the blob that generated the patch is denoted by b_i, and c_i denotes the generating component in the corresponding MoG over RGB values.

Given the index of the blob that generated a patch $\mathcal{P}_i$, the characteristics are assumed to be independently distributed, *i.e.* we have:

$$p(\mathcal{P}_i|b_i = k) = p(w_i^{sift}|\Theta_k)p(w_i^{color}|\Theta_k)p(rgb_i|\Theta_k)p(X_i|\Theta_k). \quad (1)$$

The MoG color model of each blob capture object-instance and image-background specific color distributions, as in [24]. This helps us to achieve coherent object level segmentation, even if locally recognition is ambiguous. Note that this color model plays a different role than the model over the color words w_i^{color}, which model category-level color information. The probabilities of the visual words associated with color and SIFT descriptors, words are modeled by multinomials associated with the category of the blob, *i.e.* $p(w_i^{sift}|\Theta_k) = p(w_i^{sift}|l_k)$ and $p(w_i^{color}|\Theta_k) = p(w_i^{color}|l_k)$. These distributions encode category-level appearance information, and form the recognition component of our model. These category models are the only information shared between images, and are learned from annotated training images. The maximum likelihood estimates of these distributions are found by simply normalizing the counts of how often visual words appear in each class and in the background, for all images.

3.3 A Markov Random Field over patch-to-blob assignments

Given the categories associated with the blobs, the assignments $b = \{b_1, \ldots, b_n\}$ of patches to blobs determine the segmentation of the image. The segmentation quality is enhanced with the second component of our model: the MRF over blob assignments. The MRF models the expected correlations in the assignments of neighboring patches, and aligns label changes with probable boundary locations according to the boundary map. The MRF is defined over the rectangular grid of patches using an eight-neighbor connectivity.

Above we defined a generative model over the patches $p(\mathcal{P}, b|\Theta) = p(b)p(\mathcal{P}|b, \Theta)$, where $p(b)$ was modeled using a Dirichlet process prior. Here, we will include an MRF component in the prior $p(b)$ by defining our new prior as the product of a regular MRF prior and the Dirichlet process prior, *i.e.*

$$p(\mathcal{P}, b|\Theta) \propto p_{dir}(b)p_{mrf}(b|\Theta)p(\mathcal{P}|b, \Theta). \quad (2)$$

To simplify the formulation of the MRF, we drop Θ from the notation, and rewrite the joint probability as $p(\mathcal{P}, b|\Theta) \propto \exp(-E(\mathcal{P}, b))$ using the energy

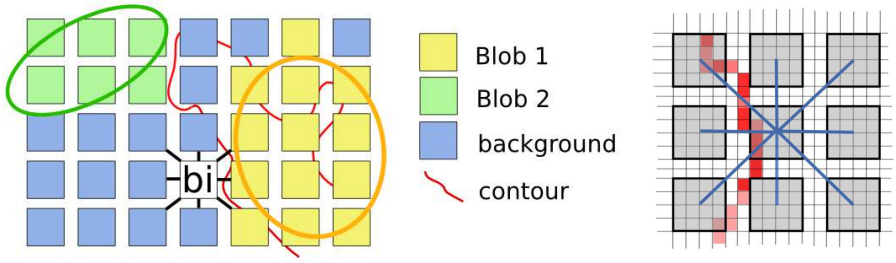


Figure 3: The model captures spatial regularity by (i) an MRF style pairwise potential, and (ii) the Gaussian and uniform spatial models associated with the object blobs and background. The MRF potential is based on the boundary map, to align label transitions with natural image boundaries. The right panel gives a close-up for a patch and its eight neighbors.

function

$$E(\mathcal{P}, b) = U(\mathcal{P}, b) + \gamma \sum_{i,j \in \mathcal{C}} V_{i,j}(b_i, b_j), \tag{3}$$

where $\mathcal{C}$ represents the set of neighbors (or cliques) in the eight-connected patch grid, γ is a parameter that balances the two terms, and

$$U(\mathcal{P}, b) = -\log(p(\mathcal{P}|b, \Theta)p_{dir}(b)). \tag{4}$$

The model p_{mrf} is represented by the second term in Equation (3), and its pair-wise potentials are defined as

$$V_{i,j}(b_i, b_j) = [l_{b_i} \neq l_{b_j}] \exp(-\beta \Phi_{i,j}), \tag{5}$$

where [.] is the indicator function. This potential enforces local coherence of the patch labels b_i, and encourages label changes to be located with high values in the boundary map $\mathcal{G}$. The maximum value in the boundary map between the centers of patches $\mathcal{P}_i$ and $\mathcal{P}_j$ is denoted $\Phi_{i,j}$, and β is the inverse of the average of the $\Phi_{i,j}$ over the image. Thus, $V_{i,j} = 0$ for neighboring patches that are assigned to the same blob, otherwise a penalty is incurred that decreases when the probability of having a boundary between the patches increases, according to $\mathcal{G}$. See Figure 3 for an illustration.

3.4 Inferring patch-to-blob and blob-to-category assignments

Above we have defined our combined Dirichlet process and MRF model. In this section we consider how to use the model to infer the patch-to-blob assignment b for a new image, together with the blob-to-category assignments l_k. In order to do this we use a Gibbs sampler that in turn samples the blob parameters Θ_k, and the patch level variables b_i and c_i. In the remainder of this section we consider the conditional distributions that are used by the Gibbs sampler.

Given a fixed patch-to-blob assignment b, the parameters of all blobs in the image $\Theta_k = \{\mu_k, \Sigma_k, C_k, l_k\}$ are distributed independently. We assume

uninformative priors over Θ_k, and we use the shorthand $\mathcal{B}_k = \{i \ : \ b_i = k\}$ to compactly write the posteriors over the parameters. For the parameters governing the spatial extent of the blob, μ_k and Σ_k, we find:

$$\mu_k \sim \mathcal{N}(Mean(\{X_i \ : \ i \in \mathcal{B}_k\}), \frac{1}{N_k} Cov(\{X_i \ : \ i \in \mathcal{B}_k\})), \quad (6)$$

$$\Sigma_k \sim \mathcal{W}(Cov(\{X_i \ : \ i \in \mathcal{B}_k\}), N_k - 1), \quad (7)$$

where we use $\mathcal{N}$ to denote a normal distribution and $\mathcal{W}$ to denote a Wishart distribution. The parameters C_k of the blob-specific color MoG are estimated using stochastic EM, using samples rather than expectations in the E-step. Finally, the multinomial from which we sample the category labels l_k are given by:

$$p(l_k|b) \propto \prod_{i \in \mathcal{B}_k} p(w_i^{sift}|l_k) p(w_i^{color}|l_k). \quad (8)$$

The variables c_i, which denote the component of the color MoG used for each patch, are straightforwardly obtained by sampling (in parallel, if desired) from the posterior over mixture components in the corresponding MoG given the patch-to-blob assignments. The patch-to-blob assignments b_i are sampled sequentially, given the blob parameters Θ_k and all other patch-to-blob assignments $b_{-i} = b \setminus \{b_i\}$. We distinguish two cases: sampling an assignment to a blob also assigned to other patches, and assigning the patch to a new blob:

$$p(b_i|b_{-i}, \Theta, \mathcal{P}) \propto \begin{cases} p(\mathcal{P}_i|b_i) \frac{N_{b_i}}{n-1+\alpha} \exp(-\gamma \sum_{i,j \in \mathcal{C}} V_{i,j}) & : \text{ existing blob} \\ p(\mathcal{P}_i|b_i) \frac{\alpha}{n-1+\alpha} \exp(-\gamma \sum_{i,j \in \mathcal{C}} V_{i,j}) & : \text{ new blob} \end{cases} \quad (9)$$

To calculate Equation (9) for a new blob, we sample parameters for the blob as follows. The category label l_k is sampled uniformly among the available categories, the blob center μ_k is sampled uniformly over the image area, and Σ_k is taken isotropic with standard deviation corresponding to half the smallest side of the image. The parameters of the color MoG, C_k, are set to the mean and covariance of all pixels in the image.

4 Using decision trees to obtain discriminant vocabularies

The segmentation model presented in the previous section relies on the notion of a visual vocabulary to represent image patches. The main reason for quantizing the patch descriptors is to make it easier to model highly multi-modal class conditional distributions over the space of low-level descriptors in the form of multinomials over the discrete vocabulary.

The conventional strategy to create visual vocabularies using simple clustering algorithms like k-means, is computationally expensive; both to create the visual vocabulary, and to assign descriptors to words. Furthermore, there is no guarantee that a vocabulary obtained by clustering is good at discriminating the visual appearance of different object classes. In fact, often the most frequent patches are not class specific, but belong to a generic background.

It has been shown recently [20] that random forest classifiers are an attractive alternative to standard clustering techniques for vocabulary construction: it

is more efficient, and yields to more discriminative vocabularies. Below, we describe how we create visual vocabularies using decision trees in Section 4.1, and then describe in Section 4.2 how we use these in our model.

4.1 Creating vocabularies using randomized decision trees

As with standard clustering techniques, the vocabulary construction using decision trees uses a large number of patches extracted from training images, described using a descriptor such as SIFT. The decision trees are constructed for optimal prediction of the category of the patch using the descriptor.

Decision trees are hierarchical structures of binary weak classifiers embedded in the tree nodes. Here, as in [20], the binary classifiers compare one of the descriptor components with a threshold. Depending on the result of this test, the patch will continue its path through the left child node or the right child node. One test is defined by two attributes, the coordinate of the descriptor component, and the threshold.

In randomized decision trees, each classifier (component / threshold pair) is chosen among a small set of randomly generated alternatives; whereas in standard decision trees the optimal classifier is chosen for each node, if computationally feasible. The justification for randomized trees can be found in [2]. The amount of randomness can be controlled by the number of possible decisions evaluated for building each node.

The quality of a test is given by the mutual information $I_{S,C}$, where S is a random variable encoding the test outcome, and C is a random variable representing the category of a patch. This criterion, directly inspired from [9], is written as:

$$I_{S,C} = H(S) + H(C) - H(S,C), \tag{10}$$

where $H(S)$ is the entropy of the test in terms of population, $H(C)$ in terms of classes, and $H(S,C)$ the joint entropy. This criterion favors well balanced trees, with leafs that contain patches of only few categories. As the tree is built, we store at each leaf the distribution over classes among the patches reaching the leaf.

Due to the randomized construction, trees constructed in such a way have a high variance. This variance can be reduced in two ways. First, the tree can be pruned, and second, results obtained by several trees constructed on the same way can be combined. We apply both methods. The pruning stage deletes nodes whose mutual information $I_{S,C}$ is low. In our experiments we control the pruning by specifying the maximum total number of leaves per tree. We then combine several pruned trees, yielding a 'forest', by combining the category probabilities obtained using the different trees.

Parameters we have to set for the tree construction are the number of tests considered for each node, the number of trees and the number of leaves per tree after pruning. We study the effect of these parameters in our experiments in the next section.

Note that the decision tree partitions the descriptor space, just as in a clustering based vocabulary. When using a forest of the decision trees, each tree gives a different quantization of the descriptor space. As the elementary tree decisions are based on the mutual information between descriptors and cate-

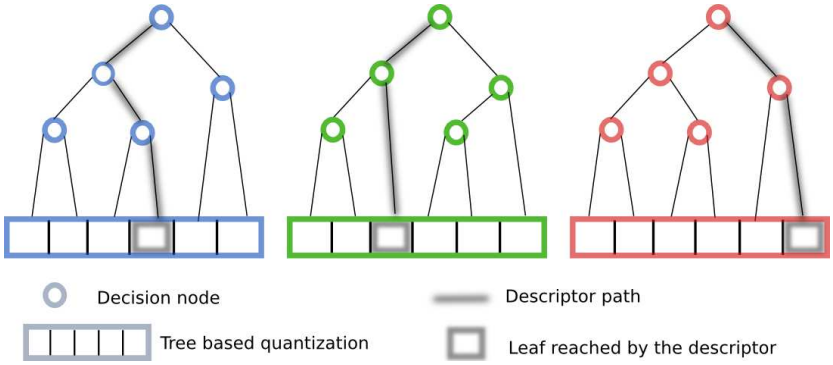


Figure 4: Decision trees are used to quantize the descriptor space.

gories, the resulting vocabularies are well-adapted to discriminate a given set of categories. See Figure 4 for an illustration.

4.2 Putting the random trees into our model

Recall that in our original model we used two visual vocabularies, one for the SIFT descriptors and one for the color descriptors. When using a forest of decision trees to obtain a collection of quantizations, we proceed in a similar way to incorporate them in our model. As before, each patch $\mathcal{P}_i$ is represented using its average RGB values rgb_i, and its 2d image coordinate X_i. To accommodate more general visual vocabularies, we replace the visual words w_i^{sift} and w_i^{color} by a collection of visual words $w_i^j, j \in \{1, \ldots, J\}$ which represent indices into J different visual vocabularies. These J visual vocabularies can be either constructed using standard clustering techniques or using decision trees, and they can be based on one or more low-level descriptor such as SIFT or color values.

To reflect this in our model, we replace Equation (1), which gives the probability of a patch given the blob assignment, and blob parameters, by

$$p(\mathcal{P}_i|b_i = k) = p(rgb_i|\Theta_k)p(X_i|\Theta_k)\prod_{j=1}^{J} p(w_i^j|\Theta_k). \tag{11}$$

Again, the probabilities of visual words given blobs only depend on the category label of the blob, $p(w_i^j|\Theta_k) = p(w_i^j|l_k)$. The latter are trivially obtained by counting and normalizing how often the visual words occur within each category, regardless of whether the visual words are obtained using clustering or decision trees. The Gibbs sampler of blob parameters changes only for l_k, which are now sampled from:

$$p(l_k|b) \propto \prod_{i \in \mathcal{B}_k} \prod_{j=1}^{J} p(w_i^j|l_k). \tag{12}$$

5 Experimental results

In this section we present our experimental results. First, in Section 5.1 we describe the data sets used in our experiments, and in Section 5.2 we discuss how

Figure 5: Example images from PASCAL VOC 2006 for categories *cat* and *people*.

we map the category labels obtained at the patch level to smooth segmentations on the pixel level. Then, in Section 5.3 we present a first series of experiments in which we investigate the effectiveness of different vocabulary construction methods. In Section 5.4 and Section 5.5 we present qualitative and quantitative experimental results in which we assess performance in comparison to existing state-of-the-art results.

5.1 Object category data sets

In our experiments, we consider three challenging data sets for object/background segmentation: the TU Graz-02 data set, the PASCAL VOC data sets of 2006 and 2007, and the MSRC data set.[1] All three contain object classes with large intra-class appearance variations, together with generic and cluttered backgrounds. Furthermore, objects have scale and illumination variations, viewpoint changes, as well as occlusions. In Figure 5 we illustrate some of the variations in two of the categories on the PASCAL VOC 2006 data set. Below, we discuss these three data sets in more detail.

The TU Graz-02 set contains images of three object categories: *bicycles*, *cars*, and *persons*. The availability of ground-truth segmentation masks makes this database interesting for quantitative evaluation of segmentation methods, and for parametric studies. This set is composed of 404 bicycle images, 420 car images, 311 images with people, and 380 background images. There are 300 images of each object class with a precise ground truth segmentation mask, and we only consider this subset in our experiments.

The PASCAL VOC 2006 data set includes a wide variety of examples of ten categories: *bicycles, buses, cats, cars, cows, dogs, horses, motorbikes, people*, and *sheep*. The full data set is composed of 5304 images which are divided in 1277 images for training, 1341 images for validation, and 2686 images for testing. As segmentation masks are not available for these images, they only interest us for qualitative experiments.

The PASCAL VOC 2007 data set contains ten categories in addition to those of PASCAL VOC 2006: *birds, boats, bottles, chairs, planes, potted plants, sofa, tables, trains,* and *TV/monitors*. The data set contains 2501 training images, 2510 validation images, and 4952 test images. Segmenting images, many of

[1]These data sets are freely available at `http://www.emt.tugraz.at/~pinz/data`, `http://www.pascal-network.org/challenges/VOC`, and `http://research.microsoft.com/vision/cambridge/recognition`.

which contain multiple objects and multiple categories, is a challenging task for current state-of-the-art methods.

We also present results on the Microsoft Research Cambridge data set, which consists of 591 images which are manually segmented in 21 categories. Each image typically contains two to five categories, but the manual segmentations do not distinguish different object instances. Furthermore, several non-object categories are included, such as *sky, grass,* and *road*.

5.2 From labeled patches to pixels

The models we present in this paper work at the patch level, but our goal is to produce precise pixel level segmentations. By using widely overlapping patches we can ensure precision of the segmentations using a simple post-processing method.

Using the Gibbs sampling procedure described in Section 3 we can obtain estimates of the posterior probabilities of the blob assignment of each patch, and a probability of the category label of each blob. From those, we can estimate the class label probability for a patch by summing the blob-class probabilities, weighted by the probability that the patch belongs to each blob. The probability for pixel p_x to belong to a category or to the background is computed by accumulating the probabilities of all patches containing this pixel. We do this with a weighted sum of the patch-level probabilities, where the weights depend on the distance between the pixel and the center of a patch. A crisp segmentation mask can then be obtained by assigning each pixel to the most probable class.

5.3 Evaluation of vocabularies: features and construction methods

In this section we evaluate different feature sets and vocabulary construction methods for our method using the TU-Graz02 data set. Images in this set contain only one object category, so the segmentation task can be seen as a binary classification problem. Thus the accuracy can be measured by precision-recall curves that show how many pixels from the object category (all images of a class merged) are correctly classified. For each class, we use half of the 300 annotated images to learn the model, while the second half is used for testing.

5.3.1 Effect of different feature sets using k-means vocabularies

Our method relies on the use of a visual vocabulary, which is a quantization of patch descriptors. We have proposed two different ways for building this vocabulary: the most common way is to use a standard clustering algorithm, as suggested Section 3, while in Section 4 we discussed a more efficient method based on random trees. However, we have observed that the impact of using different low-level features is independent of the vocabulary construction method. This first part of the parametric study thus only considers vocabularies built using k-means clustering. In the next section we will compare the vocabulary construction methods.

Several features are computed for each patch: a SIFT descriptor, a hue descriptor, the average RGB values, and the 2d image coordinates. Here we

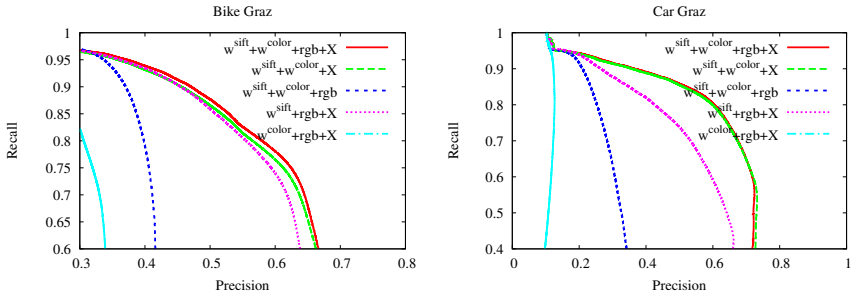


Figure 6: Performance using different feature subsets from: SIFT vocabulary (w^{sift}), color vocabulary (w^{color}), color components (rgb) and spatial coordinates (X). The MRF component is used in experiments when the image coordinates X are used.

evaluate the relative importance of these features for the segmentation result. We compare the full model, denoted $w^{sift} + w^{color} + rgb + X$, which is the one using all the features, with different models using only a subset of these features. We used the MRF component of our model in experiments that use the spatial image coordinates X, in other experiments we did not. Visual vocabularies of 5000 words are created for the SIFT descriptors, and of 100 words for the hue descriptors. They are obtained by quantizing the descriptors of training images with k-means.

The results of this parametric study are reported in Figure 6. We observe that the two visual vocabularies w^{sift}, w^{color} are essential. If one of them is missing the performance decreases significantly, but the SIFT descriptor is more critical than the hue descriptor. These results show that we need indeed strong recognition cues to guide the segmentation process.

Spatial regularization using the MRF and the blob model, improves the results considerably, as the comparison of the red (all features) and blue (without spatial information) curves shows. This regularization also considerably visually improves the segmentations qualitatively.

The rgb color feature, used at the instance level, gives an improvement for two categories out of three. When an object is correctly localized, we observed that this color component improves considerably the segmentation accuracy. In this case, some non discriminative patches can be assigned to object or background depending on their color. This phenomena is illustrated in Figure 7. It shows how the RGB color component can help segmenting a part of an object which is not initially assign to the object but whose color is consistent with the RGB color model of the object. However, when objects are not localized correctly, the color component deteriorates the results in some cases.

We can qualitatively understand the role of the different components of our model from the illustration in Figure 8. The left panel of Figure 8 shows three different segmentations of the same image obtained using a) a simple patch classifier (each visual word predicts its category), b) the Dirichlet process mixture model, and c) the full model including the MRF component. The right part of the same figure emphasizes the importance of the Dirichlet process. This image is best described with two blobs of the same category, allowing each

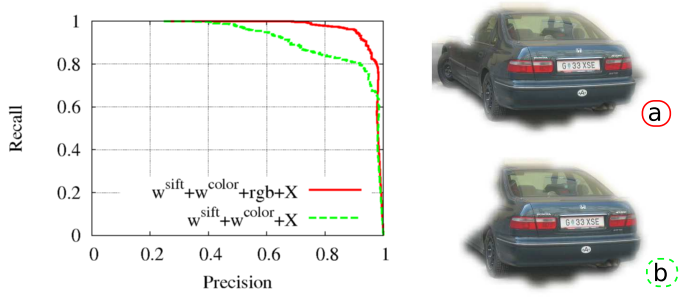


Figure 7: Our model, with and without the object-instance specific RGB color model. Precision-Recall curve in left panel, and the corresponding images in the right panel.

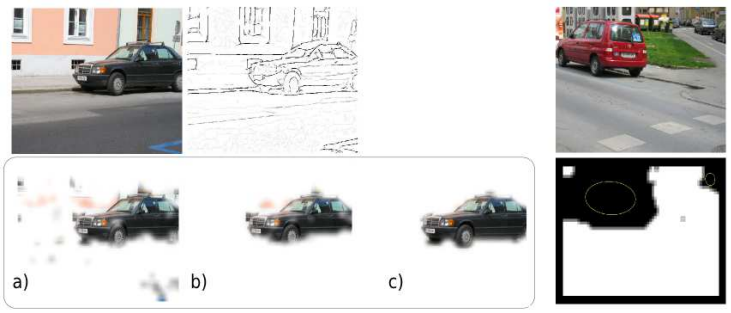


Figure 8: Left: image and its boundary map, together with segmentations produced using a) simple patch based classifier ($w^{SIFT}+w^{hue}$), b) using the Dirichlet process mixture model, and c) the full model. Right: the Dirichlet process predicts two blobs to explain the appearance difference between instances.

instance to have its own spatial and color distributions. This description is more precise, and produces a more accurate segmentation.

5.3.2 Comparison between k-means and tree based vocabularies

In this section we evaluate the quality of the segmentation when using k-means vocabularies and vocabularies obtained using decision trees. For simplicity, we consider here only the SIFT descriptor to code the category level information. Figure 9 shows the comparison of the two vocabulary types for two different classes of the Graz data set. The models include in both cases: SIFT descriptors, RGB components and patch positions. The k-means vocabulary has 5000 visual words, while the tree based vocabulary has 5000 leaves per tree (for these experiments we used three trees and 50 tests per nodes). The results show that in this setting tree-based vocabularies outperform those obtained using k-means clustering.

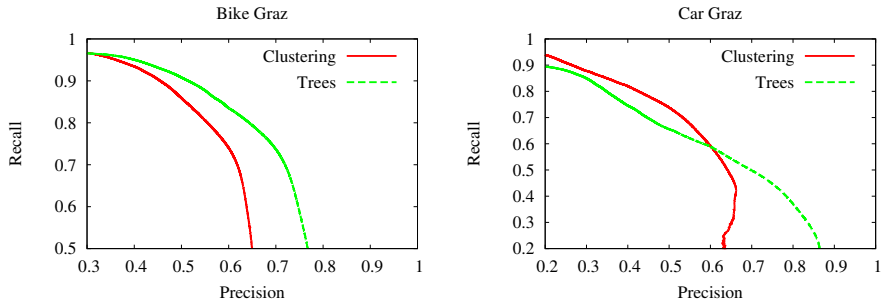


Figure 9: Comparison between the k-means vocabulary and the tree based vocabulary for two classes of the Graz data set.

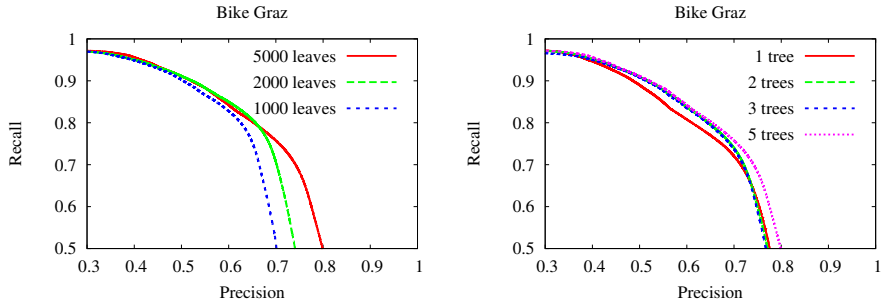
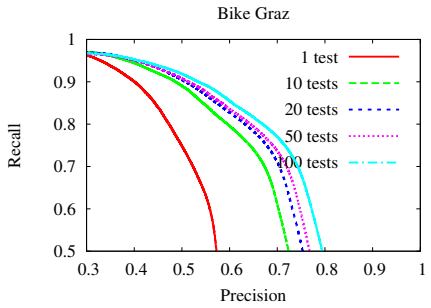


Figure 10: Influence of the number of leaves per tree (left) and of the number of trees (right), on the accuracy of the segmentation.

The random trees approach is relying on different parameters. It is therefore interesting to evaluate their influence on the segmentation results. First, the number of leaves per tree is an important parameter. It sets up the coarseness of the segmentation. We were expecting that too many clusters could lead to overfitting, but this is something we did not observe. In the left panel of Figure 10 results show that the average precision is always improved when increasing the number of leaves (up to 5000) while keeping the number of trees fixed to 3. The right part of the same figure shows the influence of the number of trees (for 5000 leaves); having more trees slightly improves the average precision, but the results are less dependent on the number of trees than on the number of leaves, which is coherent with results reported in [20].

Another key parameter is the number of split conditions evaluated for choosing the best split for each node. This parameter controls the amount of randomness while also having an impact on the time needed to build the trees. The left panel of Figure 11 shows precision-recall curves obtained for different values of this parameter, between one (fully random tree) and 100 trials per node. The improvement is significant from fully random to 10 tests per nodes; larger values (above 100) do not lead to significant improvements in accuracy. The time needed to build the trees increases with the number of trials. The right panel of Figure 11 shows the corresponding processing times. Note that the training time, even with 100 trials, is much lower than running k-means. The gain in



Method	Nodes	Time (s)
k-means	-	2507
Trees	100 tests	1342
Trees	50 tests	706
Trees	20 tests	301
Trees	10 tests	174
Trees	1 test	34

Figure 11: Left: influence of the number of tests for each node on the quality of the final segmentation. Right: the associated computation time compared to k-means clustering.

efficiency is also visible during the test stage, where patch descriptors have to be assign to visual words: assigning a descriptor to a k-means word takes 1030 μs, while assigning this descriptor to a leaf takes 4.53 μs. In the first case one Euclidean distance per visual word has to be computed in a large dimensional space, while in the second case we only compare few attributes to a threshold. Nevertheless, converting patch descriptors into visual words is only a small part of the total processing time; most of the time is spent on the model's parameter estimation (several minutes).

5.4 Qualitative results

In this section, we discuss some segmentation masks computed on Graz02, MSRC and PASCAL VOC 2006 databases, presented Figure 12. For each class, images are segmented into objects of interest and background regions. For Graz (Bike, Car and Person) and MSRC data sets, the object model is trained using the available segmentation masks. On the PASCAL 2006 data set object category models are trained from bounding box annotations only. It should be noted that this data set is used in a binary classification framework, object vs background, which reduces the complexity of the task. Accurate segmentation are produced despite the very strong appearance variations of these categories. We will see section 5.5.2 that on the PASCAL 2007 data set, the 20 object classes competing at the same time makes the problem much harder.

More typical segmentation results are shown Figure 1 and Figure 8. Our algorithm automatically detects and segments objects accurately despite large intra-class variations and scale/orientation changes, even with weak supervision (training with bounding boxes only).

Even in a multiclass framework, MSRC images are accurately segmented, however, the variation of object appearance is less significant that for the VOC 2006 data set. Indeed, we observed that the simple pure patch-based classification already performs well for these images.

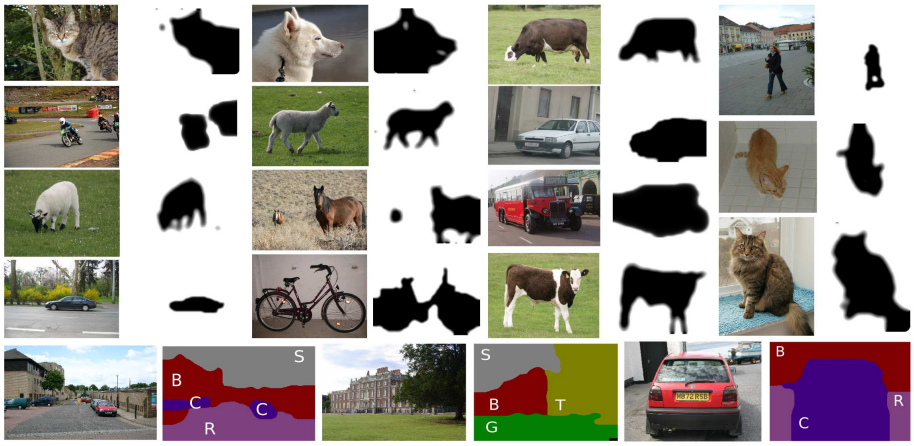


Figure 12: Examples of segmentations obtained by our method on the Graz-02, PASCAL VOC 2006, and MSRC (best viewed in color) data sets. For the last a color coding is used, and we use G for grass, Sh for sheep, S for sky, B for building, T for tree, and C for car.

	Cow	Sheep	Aeroplane	Face	Car	Bicycle	Sign	Bird	Chair	Cat	Dog	Body	Boat
Textonboost	58	50	60	74	63	**75**	35	19	15	**54**	19	**62**	7
MFAM	73	**84**	**88**	70	**68**	74	33	19	**34**	46	49	54	**31**
Our Method	**84**	81	66	**78**	50	62	**36**	**22**	16	43	**52**	30	9

Table 1: Results on the 13 object categories of the MSRC data set.

5.5 Quantitative results

In this section we present more quantitative results. First we briefly present some results for the MSRC data set in Section 5.5.1, and then present more extensive results on PASCAL VOC 2007 in Section 5.5.2.

5.5.1 Microsoft data set

Due to its popularity we compared our method with results recently published on the MSRC. Note that the task is here different because the background is divided into several classes (grass, building, trees, etc.) so the goal is not figure/ground segmentation but full segmentation of images. Table 1 gives the performance of our algorithm on the 13 object categories of the data set. We compared with the Textonboost results [26], and with the Markov Field Aspect Model (MFAM) [31]. Our method gives comparable results, although it is not designed explicitly for this kind of task.

Figure 13: Examples of additional annotations (segmentation masks) automatically produced for the unsegmented training images, obtained by applying our algorithm on the provided bounding boxes (best viewed in color).

5.5.2 Experiments on the PASCAL VOC 2007 data set

The PASCAL Visual Object Classes (VOC) challenge is an international competition for image categorization, object detection, and segmentation. In its past three editions it has evolved as a major platform for comparison of current state-of-the-art methods. We use this data set to evaluate our category level segmentation algorithm and compare it to state-of-the-art results. The segmentation challenge considers generating pixel-wise segmentation *i.e.* the label of each pixel has to be predicted as being an object class or the background, which is exactly the task we are tackling in this paper. As said earlier, the data set is made of 20 object classes and one background class. It includes more than 5000 images for training (including validation images), 422 of them are accurately annotated with segmentation masks. For the other training images, only the bounding boxes of object instances are given.

The experiments have been done according to the Pascal VOC 2007 protocol. We compute the average segmentation accuracy across the twenty classes and the background class. The segmentation accuracy, for each class, is the number of correctly labeled pixels of that class divided by the true total number of pixels of that class [6].

To estimate the model parameters we use all annotations; both the segmentation masks and the bounding boxes. The training is done in two steps. First a rough initial model of object categories is learned from the segmented training images only. We then use the remaining training images to refine the initial model. To this end we use our initial model to segment the images for which only the bounding box is given. This is done by running our segmentation algorithm, while representing each object bounding box by a single blob in our model; fixing the blob's spatial model and category label to values given by the bounding box. We only estimate the patch labels and color models given these constraints. This gives us new series of more accurate annotations, which we use to re-estimate the category level appearance models. We experimentally confirmed that these automatically produced annotations are reliable; examples of segmentation masks produced in this way are illustrated in Figure 13.

When processing test images, the number and classes of objects present in an image is not known. With the relatively large number of possible classes, we observed (results are given below) that initializing the algorithm with local patch predictor, as we have done before, is not enough to obtain good results.

	backgrd	plane	bicycle	bird	boat	bottle	bus	car
FT+DI	49.36	20.5	70.36	23.50	16.53	28.72	22.69	58.38
ST+DI	57.23	13.63	35.10	19.60	10.60	23.75	16.78	56.82
FT+NI	14.97	17.68	9.42	1.56	15.85	4.76	10.2	25.10
ST+NI	20.97	11.67	10.02	3.57	15.45	8.65	10.67	17.39
Brookes	77.7	5.5	0	0.4	0.4	0	8.6	5.2
TKK	22.9	18.8	20.7	5.2	16.1	3.1	1.2	78.3

	cat	chair	cow	table	dog	horse	moto	person
FT+DI	65.5	28.17	10.41	0.92	3.7	65.4	51.75	60.1
ST+DI	63.08	24.98	10.58	0.64	4.04	41.15	55.34	64.08
FT+NI	15.19	23.79	7.46	10.61	20.69	15.72	21.89	27.59
ST+NI	7.35	21.18	7.81	5.82	15.71	14.29	11.33	40.54
Brookes	9.6	1.4	1.7	10.6	0.3	5.9	6.1	28.8
TKK	1.1	2.5	0.8	23.4	69.4	44.4	42.1	0

	plant	sheep	sofa	train	monitor	**mean**
FT+DI	22.02	23.71	27.93	65.20	65.46	**37.16**
ST+DI	14.37	17.83	24.13	46.21	59.72	**31.41**
FT+NI	38.01	8.88	4.24	4.94	17.46	**15.05**
ST+NI	3.42	8.52	8.66	3.93	18.09	**12.62**
Brooks	2.3	2.3	0.3	10.6	0.7	**8.5**
TKK	64.7	30.2	34.6	89.3	70.6	**30.4**

Table 2: Results on the PASCAL VOC 2007 data set. The first four rows give the results obtained with our method using naive initialization (NI), detector based initialization (DI), the small training set (ST), and the full training set (FT). The two last rows, give best results among the submitted segmentation and detection methods respectively.

We then tried to use a template matching based detector, and noticed that this significantly improved the segmentation accuracy. More precisely, we used the INRIA_PlusClass detector [6] to initialize the blob positions and labels. This is a detector based on a sliding window approach including a linear SVM classifier and image descriptors based on histograms of oriented gradients [5]. When reporting our results, we use 'DI' to denote the use of this Detector for the Initialization. The naive initialization, based on patch predictions is denoted 'NI'.

Additionally to these two types of initialization, we have also evaluated how much the segmentation of unsegmented training images helps to segment test images. We compare our method trained with only the 422 segmented training images, denoted 'ST', and trained with the full training set of more than 5000 images including additional segmentation masks generated by our algorithm, denoted 'FT'.

Thus, we have four possible combinations, that have been evaluated; results obtained on the 20 classes of the VOC 2007 are given Table 2. We also report the best segmentation result submitted to the VOC 2007 competition, as well as the best possible results that has been obtained using detection algorithms, in which case the segmentation is simply given by the predicted object bounding box.

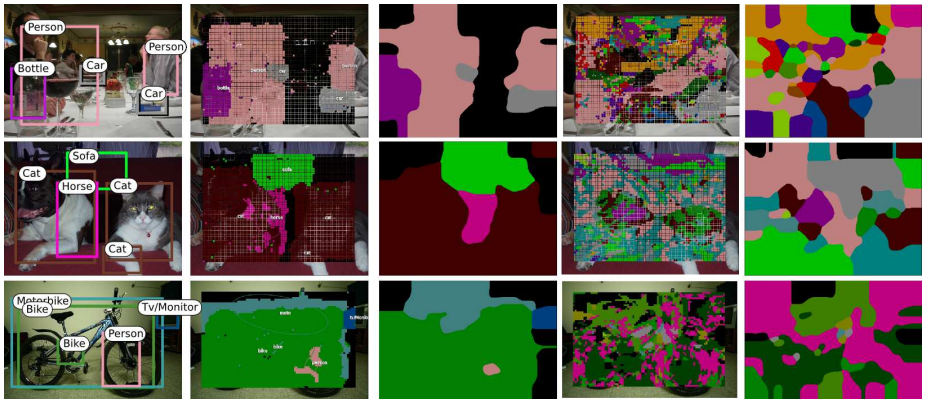


Figure 14: Three example images from PASCAL VOC 2007. From left to right: (i) the original image with the detector results superimposed, (ii) category assignments after a few iterations, (iii) the final segmentation result produced from this initialization, (iv) class labels from patch-level initialization, and (v) the final result obtained using this initialization.

From this table, we can draw three main conclusions. First, we see for nearly all classes a clear improvement in results when also using training images with bounding box annotations. Second, the results demonstrate the importance of good initializations using the detector results. Using the detector gives an overall improvement of about 15% mean accuracy. This can be explained by the large number of classes involved in the segmentation task. The detection algorithm proposes relevant candidates, which are then validated and refined by the segmentation algorithm. For some classes (like *table* or *dog*) the results are better with the naive initialization; for these classes the detector often fails. Third, we clearly outperform the best methods that entered in the challenge.

In order to better understand the role of the detector, Figure 14 illustrates the behavior of the model on some images. Starting from the initial detections, the segmentation method validates the object hypotheses and refines the object boundaries in most of the cases. For the third image, we can clearly see that the detector algorithm fired for both the *bicycle* and *motorbike* category. From these competing hypotheses the segmentation selects the bicycle. We can also see that some obvious false detections, like the person in the third image, are mostly discarded.

6 Discussion

Segmentation is commonly considered as an isolated problem: an image is given, and without any exterior knowledge, this image has to be segmented in some meaningful manner. Clearly, many interpretations of 'meaningful' are possible, but most often it is understood as segmenting at the level of objects, or their constituent parts. Many early segmentation works tried to solve this task at a local level. Although many advances were made on local descriptors, the results of local methods will always be bounded by their limited spatial horizon.

Longer-range semantic grouping is required within the segmentation process; and category-level recognition can provide the necessary cues for this. In this paper we have considered relatively simple category-level appearance models which can be efficiently estimated from a set of manually segmented training images. Hence, our method applies in cases where we want to segment instances of categories in new images. Similarly, recognition requires accurate segmentation to avoid distraction from background clutter and occluding objects. Therefore, we have designed our model to couple these two processes. We conclude this paper with a discussion of our model in the context of related work, and possible extensions to overcome some of its limitations.

Robust category-level recognition requires robustness to intra-class variations and imaging conditions such as occlusions, illumination changes, view point and scale variations. The recognition part of our model relies on a bag-of-words representation, which makes it intrinsically robust to occlusions. For the same reasons [31, 26] share the same property. However, this is not the case for methods based on rigid shape models [12, 16], which cannot deal with large occlusions. Invariance to illumination is insured by the patch descriptors we use [18, 30], and scale and position changes are handled using our 'blob' based modeling of object instances. We believe this is more realistic than using category specific location priors as in [26, 31], which can sometimes be useful except when dealing with images with unusual spatial layouts of objects. Appearance variations resulting from strong viewpoint changes and intra-class variations of non-rigid objects are also inherently handled using our flexible blob-based object instance model. There are no rigid constraints between patch pairs of an object, but there is some accumulation of hypothesis on the object position and size which guide the assignment from patches to objects. This is why we can deal with challenging non rigid classes, such as persons and cats. This contrasts with the philosophy of shape methods [12, 16] which learn accurate models of objects, but allow only small viewpoint and shape variations.

In our experiments we showed that we can use a supplementary object category detector, which operates at a level of bounding boxes, to improve results when segmenting many object categories simultaneously. However, note that the segmentation that our model returns is richer than what could be obtained using a simple combination between a detector and a color based segmentation method like Grab-Cut [24]. Using our model we can separate different object instances, and deal with multiple categories per image.

Regardless of the complexity of the category models, all segmentation methods rely on some regularization constraints; typically assuming neighboring parts of an image to belong to the same category or segment. The image resolutions of the image parts differs, ranging from pixels [22], to patches [31, 26, 15], or regions [1]. In any case, some scale has to be fixed for the neighborhoods, which require to strike a balance between over and under smoothing. Some method tried to overcome this difficulty, *e.g.* in [31, 32] semantic information at the image-wide level showed improvements on segmenting images with several, but few, object categories per image. It is unclear how it would behave for more complex scenes with many object categories. In contrast, in this paper we use the Dirichlet process over our blobs to incorporate semantic information over semi-global ranges, as in [28], which automatically adapts the range of the dependencies.

Our model has the ability to segment different instances of the same object category in different blobs. This could appear useless when the final goal is to predict a class label per pixel, and not to identify different instances of the same category. However, still in this case it can be beneficial to separately model the different instances as it allows us to fit more precise instance specific appearance models (color model in our case). In this manner each object instance can be more accurately segmented, leading to a better overall result.

Many segmentation methods require manually segmented training images; albeit with different levels of detail. When only bounding boxes of objects are available, particular care has to be taken not to include the object context in the object model. Roughly speaking, this can be done in two ways. Either, by using image contours, or discontinuities between homogeneous regions, we can refine each bounding box by looking for a large consistent region in it. Or, alternatively, one can hope to have sufficiently similar object instances against sufficiently different backgrounds to determine which part is the object and which part is background [12]. Like *e.g.*[33], we use both strategies to obtain satisfying segmentation results on the PASCAL VOC 2006 data using only bounding boxes as training data.

For more complex problems involving more classes we will need to proceed in a different way. We showed in our experiments how we can combine annotations with different levels of detail: pixel-level segmentations and bounding boxes. We achieve this by learning an initial model from the pixel-level segmentations, and then use this to apply our model where we use the bounding boxes to give fixed parameterizations of the blobs. The resulting category-level segmentations are then used to re-estimate the category appearance models. In our experiments we show that training mask produced on the basis of bounding boxes in this manner can be really accurate, and allows us to use much more training images and improve segmentations of new images.

The way we used 'weak' annotation here in the form of bounding boxes, differs from the type of weak annotations considered in [32]. There, CRF segmentation models were learned from 'partial' pixel-level segmentations. In the partial segmentations, some pixels can be either completely unlabeled, or marked as belonging to one of several possible classes, but not to any other classes. However, the method is not straightforwardly applied to learning from bounding boxes, as the pixels in bounding boxes would be marked as potentially belonging to the object or the background. Thus, there are no explicit constraints to prevent the trained model from labeling everything as background: this would not violate any constraints in the partially labeled training data.

In extensions of the current model we can further develop the interplay between the instance specific and category level appearance models. For now, the low-level image cues are used by either the instance-level or category level model. Note that we use two different color descriptors, one for each model. The fact that these descriptors are necessarily highly correlated suggests that we may actually use all descriptors in both models: enforcing that all appearance facets are compatible with both the instance model, and the category model. Of course, other descriptors can be included, *e.g.*based on scale and shape.

In its current form, the model has difficulty to distinguish two objects of the same category which are very close to each other. For instance, a car occluding another car would be grouped in a single object blob. Some form

of geometric information should be included in the category-level appearance models, to resolve such ambiguities and improve results further.

Acknowledgments

The authors would like to thank Eric Nowak for his help, and Hedi Harzallah for providing his category detection results.

References

[1] E. Borenstein and J. Malik. Shape guided object segmentation. In *IEEE Conference on Computer Vision & Pattern Recognition*, pages 969–976, 2006.

[2] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.

[3] L. Cao and L. Fei-Fei. Spatially coherent latent topic model for concurrent object segmentation and classification. In *IEEE International Conference on Computer Vision*, 2007.

[4] G. Csurka, C. Dance, L. Fan, J. Williamowski, and C. Bray. Visual categorizationwith bags of keypoints. In *ECCV workshop on Statistical Learning in ComputerVision*, pages 59–74, 2004.

[5] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *European Conference on Computer Vision*, 2006.

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html, 2007.

[7] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *IEEE International Conference on Computer Vision*, volume 101, pages 5228–5235, 2005.

[8] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[9] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Mach. Learn.*, 63(1):3–42, 2006.

[10] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Mach. Learn.*, 42(1-2):177–196, 2001.

[11] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *IEEE International Conference on Computer Vision*, pages 604–610, 2005.

[12] M. Kumar, P. Torr, and A. Zisserman. OBJ CUT. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2005.

[13] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*, volume 18, pages 282–289, 2001.

[14] D. Larlus and F. Jurie. Latent mixture vocabularies for object categorization. In *British Machine Vision Conference*, 2006.

[15] D. Larlus and F. Jurie. Combining appearance models and markov random fields for category level object segmentation. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2008.

[16] B. Leibe and B. Schiele. Interleaved object categorization and segmentation. In *British Machine Vision Conference*, 2003.

[17] Y. Li, J. Sun, C.K. Tang, and H.Y. Shum. Lazy snapping. *ACM Trans. Graph.*, 23(3):303–308, 2004.

[18] D Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[19] D Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(5):530–549, 2004.

[20] F. Moosmann, E. Nowak, and F. Jurie. Randomized clustering forests for image classification. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2008. to appear.

[21] R. Neal. Markov chain sampling methods for dirichlet process mixture models. Technical Report 9815, Dept. of Statistics, University of Toronto, Sep 1998.

[22] P. Orbanz and J. M. Buhmann. Smooth image segmentation by nonparametric bayesian inference. In *European Conference on Computer Vision*, 2006.

[23] F. Perronnin, C. Dance, G. Csurka, and M. Bressan. Adapted vocabularies for generic visual categorization. In *European Conference on Computer Vision*, 2006.

[24] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.

[25] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *IEEE International Conference on Computer Vision*, pages I:503–510, 2005.

[26] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European Conference on Computer Vision*, pages I: 1–15, 2006.

[27] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *IEEE International Conference on Computer Vision*, 2003.

[28] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky. Describing visual scenes using transformed dirichlet processes. In *Advances in Neural Information Processing Systems*, 2005.

[29] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky. Describing visual scenes using transformed objects and parts. *International Journal of Computer Vision*, 77(1-3):291–330, 2008.

[30] J. van de Weijer and C. Schmid. Coloring local feature extraction. In *European Conference on Computer Vision*, pages 334–348, 2006.

[31] J. Verbeek and B. Triggs. Region classification with markov field aspect models. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2007.

[32] J. Verbeek and B. Triggs. Scene segmentation with crfs learned from partially labeled images. In *Advances in Neural Information Processing Systems*, 2008.

[33] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *IEEE International Conference on Computer Vision*, 2005.

[34] J. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *IEEE Conference on Computer Vision & Pattern Recognition*, pages 37–44, 2006.

Centre de recherche INRIA Grenoble – Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399