HAL open science

Category level object segmentation by combining bag-of-words models with Dirichlet processes and random fields

Diane Larlus, Jakob Verbeek, Frédéric Jurie

To cite this version:

Diane Larlus, Jakob Verbeek, Frédéric Jurie. Category level object segmentation by combining bag-of-words models with Dirichlet processes and random fields. International Journal of Computer Vision, 2010, 88 (2), pp.238-253. 10.1007/s11263-009-0245-x . inria-00439303v2

HAL Id: inria-00439303

https://inria.hal.science/inria-00439303v2

Submitted on 27 Apr 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Noname manuscript No.
(will be inserted by the editor)

Category level object segmentation by combining bag-of-words models with Dirichlet processes and random fields

Diane Larlus · Jakob Verbeek · Frédéric Jurie

Received: date / Accepted: date

Abstract This paper addresses the problem of accurately segmenting instances of object classes in images without any human interaction. Our model combines a bag-of-words recognition component, with spatial regularization based on a random field and a Dirichlet process mixture. Bag-of-words models successfully predict the presence of an object within an image; however, they can not accurately locate object boundaries. Random Fields take into account the spatial layout of images and provide local spatial regularization. Yet, as they use local coupling between image labels, they fail to capture larger scale structures needed for object recognition. These components are combined with a Dirichlet process mixture. It models images as a composition of regions, each representing a single object instance. Gibbs sampling is used for parameter estimations and object segmentation.

Our model successfully segments object category instances, despite cluttered backgrounds and large variations in appearance and viewpoints. The strengths and limitations of our model are shown through extensive experimental evaluations. First, we evaluate the result of two methods to build visual vocabularies. Second, we show how to combine strong labeling (segmented images) with weak labeling (images annotated with bounding boxes), in order to limit the labeling effort needed to learn the model. Third, we study the effect of different initializations. We present results on four image databases, including the challenging PASCAL VOC'07 data set on which we obtain state-of-the art results.

Diane Larlus
Darmstadt University of Technology, Germany
E-mail: larlus@mis.informatik.tu-darmstadt.de

Jakob Verbeek
INRIA Rhône-Alpes, France
E-mail: jakob.verbeek@inria.fr

Frédéric Jurie
University of Caen, France
E-mail: frederic.jurie@unicaen.fr

Keywords Object recognition · Segmentation · Random Fields

1 Introduction

After several decades of research, image segmentation still 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 learn 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 accurately segmenting instances 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'. Note that this differs from 'image segmentation' or 'scene segmentation', which correspond to the situation where everything in the image has to be segmented. 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 built from these images allow object recognition, which we couple with the segmentation process. In particular, we are interested in segmenting object categories that demonstrate large intra-class appearance variations. In Figure 1 we

Fig. 1 Examples of object category segmentation obtained by our method without user interaction. Input images (columns 1 and 4), object category masks (columns 2 and 5) and object category segmentation (columns 3 and 6).

show several typical images with corresponding segmentation masks produced by our method. 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 manual effort. The major contribution of our approach is an instance based modeling of the scene. More precisely, the object recognition is enhanced by a mechanism which allows to distinguish and model the different instances belonging to a particular class. The number of instances is automatically estimated and controls the number of regions produced by our segmentation.

The model presented in this paper combines three complementary components: (a) a random field (RF) component which ensures short-range spatial contiguity of the segmentation by aligning segment boundaries with low-level image boundaries, (b) a Dirichlet process component that ensures mid-range spatial contiguity by modeling the image as a composition of blobs, each of which corresponds to a single object, and (c) a bag-of-words object recognition component which allows for strong intra-class appearance and imaging variations. Although the combination of RFs with a recognition component based on visual words has been explored before, the main contribution of the model presented in this paper is the addition of a Dirichlet process to achieve higher quality segmentation and instance-level segmentation. This paper extends [19] with additional experiments and an evaluation of vocabulary construction methods.

In the remainder we first review related work in Section 2. Then, in Section 3 we present our model and the estimation of its parameters. Visual vocabulary construction for bag-of-words methods based on decision trees is described in Section 4. We present our experimental results in Section 5, and conclude with a discussion in Section 6.

2 Discussion of related work

Segmentation can be seen as a 'chicken-egg' problem, where object detection and recognition is required for accurate segmentation, and vice versa. We will first discuss generative bag-of-words object recognition methods, and then turn to methods which are primarily designed for segmentation.

Bag-of-words methods have proven to be very effective for the recognition of object classes. The 'visual words' in the image representations are obtained by quantization of low-level image descriptors. The quantization can be computed in different ways. Often, visual vocabularies are produced by a standard unsupervised clustering techniques [6, 14, 20]. In our model, the visual vocabulary is used to discriminate between classes at the level of patches. Methods have been designed to produce more discriminative vocabularies when labels are available at the image or at the patch level [18, 25]. Among such techniques, the ones based on trees are of particular interest because of their efficiency and the fact that they directly pursue class-discriminative quantization using patch labels. In Section 4 we describe quantization using decision trees in detail, and we compare such quantization to those obtained by k-means in our experiments.

Topic models, such as probabilistic Latent Semantic Analysis (pLSA) and Latent Dirichlet Allocation (LDA) [1, 13], have recently been introduced as an alternative over the simple Naive Bayes model for bag-of-words image representations. Topic models consider the bag-of-words as a mixture of several 'topics' which can be thought of scene elements in images, *e.g.* the visual words in an image of a beach scene are modeled as a mixture of words belonging to sea, sky, people, trees, etc. Each image has its own distribution over topics, and each topic is represented as a distribution over visual words. Several authors have extended the standard topic models from the text analysis community to include modeling of some spatial aspects of the image [5, 10, 33]. Such models are not only useful for image classification, where the images of each class are modeled using a gener-

ative topic model over the images of that class, but are also useful for object localization. The main limitation of these methods is that they either use a very rigid and coarse model of the object shape, are overly flexible without any shape prior, or use an initial over segmentation of the image and assign each segment as a whole to a topic which breaks if the initial segmentation contains errors. In all cases a precise object segmentation is not obtained in general.

Various forms of Random Fields (RFs) have been proposed for image segmentation [11,16,17,30,36]. They define a probability distribution over the labels of sites (pixels or image patches) which encodes correlations between neighboring sites. RFs incorporate evidence terms acting on individual sites; *e.g.* the visual word associated with a patch will increase the likelihood of the patch having a certain label. Ambiguities that arise when considering the local evidence for patches in isolation can be resolved by propagating evidence for labels spatially over the image.

Some models combine topic models and RFs [27,35]. However, these models do not include a component to ensure mid-range spatial contiguity of the segmentation: they only use the local regularization of the RF and the topic model that enforces a regularization at an image-wide scale. As compared to a standard topic model such models generate a crisper segmentation, while compared to a standard RF small regions with a label that does not appear elsewhere in the image are suppressed. In contrast, our model tries to capture object instances using blobs, which will result in mid-range regularization. In a similar spirit, in [32] a tree structure is learned dynamically to locate the position of the objects in an image, and the relative location of their parts. The modeling of object parts can improve the ability to differentiate instances, but the model does not include a fine random field type spatial regularization.

A number of approaches combining local regularization using RFs with more geometric object category models have been proposed [2,15,20,21,37,38]. These approaches model the shape of objects and their deformations, sometimes also taking occlusions and viewpoint changes explicitly into account. Although they are robust to small local shape variations, the strong geometric constraints embedded into the models are not suitable to model the complex appearances of weakly structured object classes. Examples of these complex appearances can be found in Figure 4, for the classes cats and people. Such classes require more flexible models.

Finally, we mention work on interactive segmentation tools [3,22,29] where a user roughly indicates the object of interest using a bounding box or using a brush tool. Models of the foreground and background are estimated, and these models are used in combination with a RF to spatially propagate the user-provided labels. After labels propagation the models are re-estimated and the procedure is repeated. Using such an interactive approach, remarkably accurate segmentation results can be obtained. The next step is to reduce the user interaction to only specifying the object category, *e.g.* a user could ask to segment all cats in an image.

3 The proposed segmentation model

In our model we represent images as a collection of overlapping square patches $\mathcal{P}_i, i \in \{1, \ldots, n\}$ 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 position, size, and shape are unknown. We associate a blob label with each patch, and define a Random Field (RF) 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 object model parameters have been estimated from labeled training images, we can use a Gibbs sampler to estimate the category labels of patches in a new unlabeled image.

Below, we first describe our feature extraction procedure in Section 3.1, then we continue in Section 3.2 with the Dirichlet process mixture model over the features, and then come to the RF component of the model in Section 3.3. We describe the Gibbs sampler for parameter estimation in Section 3.4. Finally, in Section 3.5 we discuss how we map the category labels obtained at the patch level to a smooth segmentation on the pixel level.

3.1 Visual feature extraction

For image patch i the feature set $\mathcal{P}_i$ contains

1. the SIFT descriptor [23], coded by the visual word w_i^{sift},
2. the hue descriptor [34], coded by the color word w_i^{color},
3. the average RGB value in the patch center, denoted rgb_i,
4. the image coordinates of the patch center $X_i = (x_i, y_i)$.

In Section 4 we will discuss the quantization of the SIFT and hue descriptor in detail which allows to compute w_i^{sift} and w_i^{color} for all patches.

In addition we extract a boundary map $\mathcal{G}$ that gives for each pixel an estimate of the probability of being part of a boundary between image segments. The map is based on characteristic changes in several local cues associated with natural boundaries, see Figure 2 for an example. Many methods exist to extract natural boundaries, striking different balances between accuracy and computational complexity. Here, our choice was purely based on accuracy, and we used one of the current state-of-the-art methods [24].

Fig. 2 Example image from the Graz database and its boundary map.

3.2 A Dirichlet process over patch characteristics

In this section we present a generative model for rough object/background segmentation. We use a model inspired by [33] with explicit spatial structure information: we consider that an image is made of regions that we call 'blobs'. Each blob generates the features of the patches associated with that blob, where the distribution over features 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 blob for the background. Given the blobs and their parameters, the patches in an image are assumed to be independent. The generative process for a patch is as follows: (i) sample a blob, and (ii) sample the features using the distribution of the blob. The remainder of this section details this generative process.

The Dirichlet process (DP) [26] can be seen as the limit as $K \to \infty$ of a finite K-component mixture model. The mixing weights of the components are controlled by a 'concentration parameter' $\alpha > 0$; smaller values implement a prior to use fewer mixture components. Note that even for a mixture with an infinite number of components, only finitely many mixture components can be associated with a finite sample. In our case the blobs will take the role of mixture components. This means that a newly sampled patch, can be either sampled from one of the blobs that have been used before, with probability $N_k/(n-1+\alpha)$ where N_k is the number of samples associated with blob k, and n is the number of samples including the current one. Alternatively, the patch can be sampled from a new blob with a probability $\alpha/(n-1+\alpha)$. DPs exhibit a 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 clustering property is desirable as it will reduces the likelihood to assign patches to classes that are rare in the image: if a patch observation leaves ambiguity on the corresponding category the most frequent class throughout the image is preferred. Below, we use p_{Dir} to denote the probability of the patch-to-blob assignment.

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 distribution $p(X_i|\Theta_k) = \mathcal{N}(X_i; \mu_k, \Sigma_k)$. 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} and its spatial model is defined as uniform over the image area.

In addition to the features $\mathcal{P}_i = \{w_i^{sift}, w_i^{color}, rgb_i, X_i\}$ 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 features are assumed to be independent, and we have

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

The color models, as in [29], capture color distributions of specific object instances and the background. This helps us to achieve coherent object instance 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 and have some degree of invariance to lighting conditions.

The probability of visual words associated with color and SIFT descriptors 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. The category models are the only source of information which is shared across images, and they 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 training images.

3.3 A random field over patch-to-blob assignments

Given the categories associated with the blobs, the patch-to-blob assignment $b = \{b_1, \ldots, b_n\}$ determines the segmentation of an image. To enforce local spatial contiguity in the above model we add an RF prior over blob assignments. By using the image boundary map to define the RF potentials, label changes will be aligned with low-level image boundaries. The RF is defined over the rectangular grid of patches using an 8-neighbor connectivity.

Above we defined a model over the patch-to-blob assignments and patch features $p(\mathcal{P}, b|\Theta) = p(b)p(\mathcal{P}|b, \Theta)$, where $p(b)$ was modeled using a Dirichlet process prior. We include the RF in the model $p(b)$ by defining our new model as the product of a RF model and the Dirichlet process:

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

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

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

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 encompasses the Dirichlet process:

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

The second term in Equation (3) represents p_{RF}, 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 $\mathcal{G}$, similar to the approach in [29,30,36]. 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.

We note that the definition of Equation (2) may seem problematic due to the fact that p_{Dir} distributes over an infinite state space, whereas p_{RF} will be defined over a finite state space. However, in practice we can clip p_{Dir} to assign zero probability to using more blobs than image patches, and re-normalizing the distribution over the remaining configurations. Since we will use a Gibbs sampler for inference we do not actually need to include the normalization term, and we omitted the clipping term for p_{Dir} in Equation (2).

3.4 Approximate inference using Gibbs sampling

In this section we consider how to use the model to infer the patch-to-blob assignment b for an image, together with the blob-to-category assignments l_k. Exact inference in our model is intractable, and we thus have to resort to approximate inference techniques. We have chosen to use a Gibbs sampler, motivated by its conceptual and practical simplicity, and not aiming to use the most efficient possible technique for approximate inference in our model. The Gibbs sampler samples in turn the blob parameters Θ_k, and the patch level variables b_i and c_i.

Given a fixed patch-to-blob assignment b, the blob parameters $\Theta_k = \{\mu_k, \Sigma_k, C_k, l_k\}$ 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:

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

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

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:

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

Given the patch-to-blob assignments, the c_i variables that denote the component of the color MoG used for each patch, are straightforwardly sampled from the posterior over mixture components in the corresponding MoG.

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:

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

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.

3.5 Towards a pixel-level segmentation

The model presented above works at the patch level, but our goal is to produce a precise pixel level segmentation. By using overlapping patches we can ensure precision of the segmentation using a simple post-processing method. The Gibbs sampler gives us 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

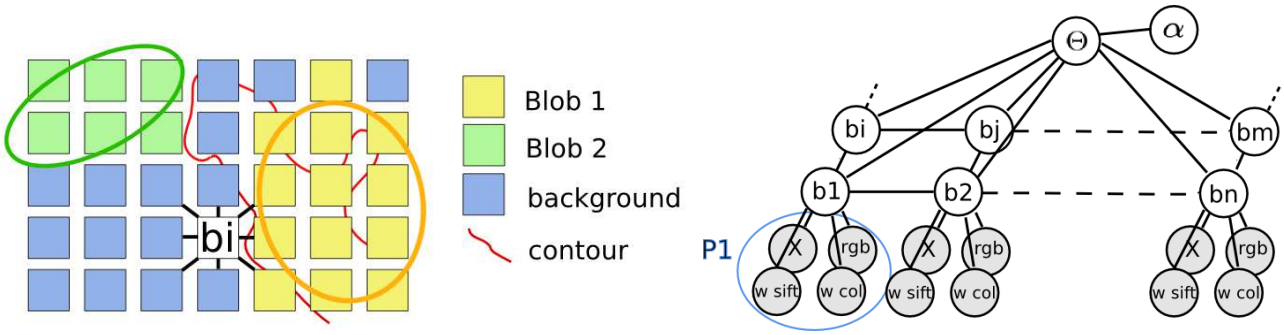


Fig. 3 The model captures spatial regularity by (i) a contrast sensitive pairwise potential, and (ii) the Gaussian and uniform spatial models associated with the object blobs and background, left panel. The right panel shows the graphical representation of the model.

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.

4 Decision trees as discriminant vocabularies

Our segmentation model relies on a visual vocabulary to represent image patches. It has recently been shown [25], in the context of image categorization, that decision trees are an efficient alternative to clustering for vocabulary construction that lead to more discriminative vocabularies. Motivated by this success, we consider them here in the context of segmentation. Decision trees have been used by others as a quantization method for segmentation [31], but a direct comparison to using clustering was not presented.

Note that the reason for quantizing the descriptor space is to facilitate the modeling of highly multi-modal class conditional distributions in the form of multinomials over a discrete vocabulary. The usual manner 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 appearance of object classes.

Decision trees are binary trees with a test embedded in each non-leaf node. They are constructed for optimal prediction of an output, here category label, given an input, the patch descriptor here. As in [25], we use binary tests that compare one of the descriptor components with a threshold. Depending on the result of this test, the patch descends to the left or right child node. Note that decision trees partition the descriptor space, just like clustering methods.

Using multiple randomly constructed decision trees concurrently is important for two reasons [4, 12]. First, the optimal decision trees have a high variance as a function of the training data, *i.e.* maximum likelihood estimation is not robust. Second, for most practical problems it is intractable to find the best decision trees for a given training set. In practice very good results are obtained by randomly constructing near-optimal trees and averaging over their predictions, similar to Bayesian model averaging. The randomized construction starts at the root node, and adds nodes one-by-one, for each node the best among several randomly generated candidate splits is used. The ensemble of multiple trees is often referred to as a 'forest'. The forest is characterized by (i) the number of trees, (ii) the number of leaves in the trees, and (iii) the number of candidate splits used during construction. We study the effect of these parameters in our experiments.

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 for multiple descriptors we proceed in a similar way: each patch having as many visual words as we have trees. Recall that each patch $\mathcal{P}_i$ is represented using a RGB value rgb_i, and its 2d image coordinate X_i, and multiple visual words which we now denote $w_i^j, j \in \{1, \ldots, J\}$. Equation (1) which gives the probability of the patch characteristics given the blob assignment now becomes

$$p(\mathcal{P}_i|b_i = k) = p(rgb_i|\Theta_k)p(X_i|\Theta_k) \prod_{j=1}^{J} p(w_i^j|\Theta_k). \quad (10)$$

The Gibbs sampler of blob parameters changes only for l_k, which are now sampled from

$$p(l_k|b) \propto \prod_{i \in \mathcal{B}_k} \prod_{j=1}^{J} p(w_i^j|l_k). \quad (11)$$

This formulation with multiple vocabularies can be used for any type of vocabulary (clustering or tree based).

5 Experimental results

In this section we present our experimental results. First we describe the data sets in Section 5.1. Then, in Section 5.2 we

Fig. 4 Example images from PASCAL VOC 2006 for categories *cat* (top) and *people* (bottom).

study the influence of the features used in our model, and show that all contribute to the final segmentation. We also compare vocabulary construction methods, and demonstrate the effectiveness of tree-based vocabularies.

In Section 5.3 we present qualitative segmentation results; quantitative results follow in Section 5.4. First, we assess performance in comparison to the state-of-the-art results, and obtain comparable results. Then we show how we successfully combine a small set of annotated images with a larger set of weakly labeled images. We also study the influence of the initialization of our algorithm, and show that this has a big impact on results.

Finally, in Section 5.5 we consider how the modeling of individual instances of an object class can help the segmentation at the category level. We show images where it helps as well as typical failures; in particular we present cases where the number of modeled instances does not correspond to the real number of instances in the image.

5.1 Object category data sets

In our experiments, we consider four challenging data sets for object segmentation: the TU Graz-02 data set, the PASCAL VOC 2006 and 2007 data sets, and the MSRC data set.[1] All four contain large intra-class appearance variations including scale, illumination, and viewpoint changes, as well as occlusions and complex backgrounds. In Figure 4 we illustrate two categories of the PASCAL VOC'06 data set.

The TU Graz-02 set contains images of the categories *bicycle*, *car*, and *person*. 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'06 data set contains examples of ten categories: *bicycles, buses, cats, cars, cows, dogs, horses, motorbikes, people*, and *sheep*. The 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'07 data set contains ten categories in addition to those of PASCAL VOC'06: *birds, boats, bottles, chairs, planes, potted plants, sofa, tables, trains,* and *monitors*. The data set contains 2501 training images, 2510 validation images, and 4952 test images. Within the training and validation sets, for a subset of 422 images, object instances are segmented at pixel level, in the other images object instances are marked by bounding boxes.

We also present results on the MSRC 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 segmentation does not distinguish different object instances. Furthermore, several non-object categories are included, such as *sky, grass,* and *road*.

For all data sets the same settings have been used to extract patches. Between 2000 and 4000 patches of 25×25 pixels are extracted per image, and the 5×5 pixel center is used to compute the RGB patch value.

5.2 Evaluation of features and vocabulary construction

Here 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.

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

For each class, we use half of the 300 images to learn the model, while the second half is used for testing.

We found that the effect of low-level features is independent of the vocabulary construction method. Therefore, we evaluate them only for k-means vocabularies.

5.2.1 Effect of different feature sets

For each patch we compute a SIFT descriptor, a hue descriptor, the average RGB values, and the 2d image coordinates. Here we 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 random field (RF) 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 clustering the descriptors of training images with k-means.

The results of this parametric study are reported in Figure 5. We observe that the two visual vocabularies w^{sift}, w^{color} are essential. If one of them is missing the performance decreases significantly, however the SIFT descriptor is more critical than the hue descriptor. These results show that we need indeed strong category level recognition cues to guide the segmentation process. Spatial regularization using the RF and the blob model, improves the results considerably, as the comparison of the red (all features) and magenta (without spatial information) curves shows.

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, non class-discriminative patches can be assigned to object or background depending on their color, as shown in Figure 6. In the same figure, we also illustrate the role of the different components of our model by showing the segmentation of an 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 RF.

5.2.2 Comparison between k-means and trees

Next, we compare the quality of the segmentation when using k-means vocabularies and ones obtained using decision trees. For simplicity, we consider here only the SIFT descriptor to code the category level information. To achieve a fair comparison with the forest, we run several times the k-means algorithm and combined statistics obtained by these multiple-vocabularies in the same way trees are combined. The left part of Figure 7 shows the comparison of the two vocabulary types for the bike category of the Graz data set, for combination of 5 vocabularies. The models include in both cases: SIFT descriptors converted into visual words, RGB components and patch positions. Each k-means vocabulary has 5000 visual words, while the tree based vocabulary has 5000 leaves per tree (for these experiments we tried 50 tests per node). The results show that for this setting, tree-based vocabularies outperform those obtained using k-means clustering. The right part of Figure 7 shows that varying the number of k-means clusterings considered does not significantly change the segmentation results.

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. The results in the left panel of Figure 8 show that the average precision improves when increasing the number of leaves, at least 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 previous findings [25].

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 9 shows precision-recall curves obtained for different values of this parameter, between one (fully random tree) and 100 trials per node, while keeping the number of trees fixed to 3. 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 9 shows the corresponding processing times. Note that the training time, even with 100 trials, is much lower than running k-means once. The gain in 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 high dimensional space, while in the second case we only compare a few descriptor dimensions to thresholds. Nevertheless, converting patch descriptors into visual words is only a small part of the total processing time; computing the features takes about 30 seconds for a dense extraction of an image (3000 descriptors), and parameter estimation with the Gibbs sampler takes about 1 minute.

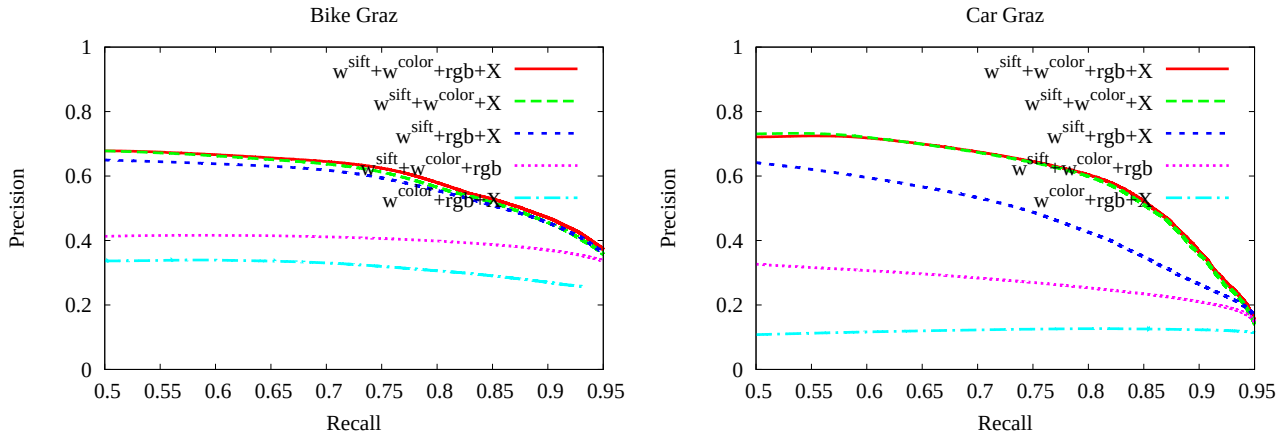


Fig. 5 Performance using different feature subsets from: SIFT vocabulary (w^{sift}), color vocabulary (w^{color}), color components (rgb) and spatial coordinates (X). The MRF component is used in experiments when the image coordinates X are used.

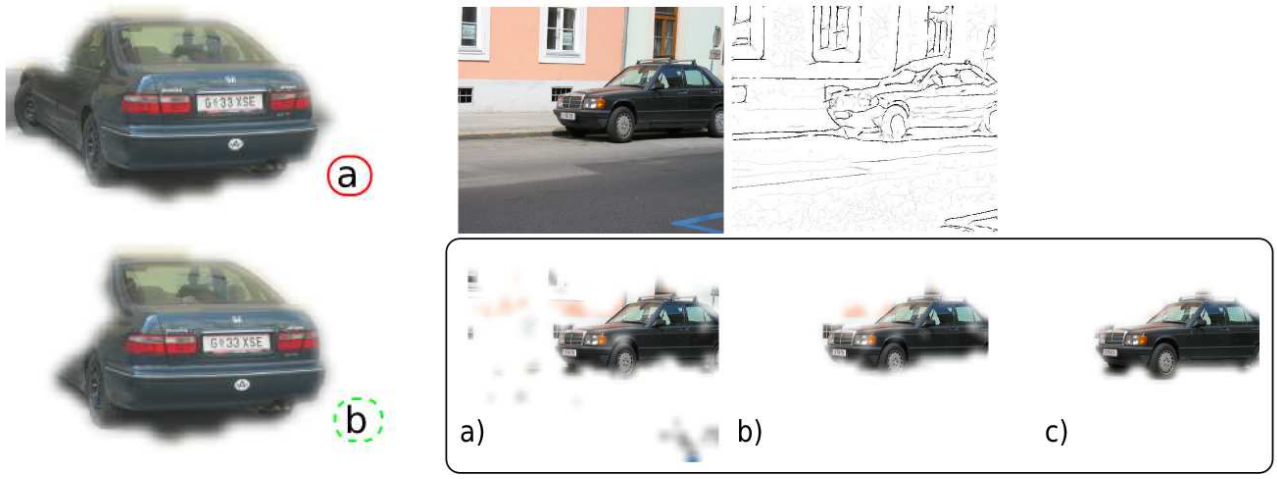


Fig. 6 Left: our model, with (a) and without (b) the instance specific RGB color model. Right: image, boundary map, and segmentation produced using a) simple patch based classifier ($w^{SIFT} + w^{hue}$), b) using the Dirichlet process mixture model, and c) the full model.

5.3 Qualitative results

In this section, we discuss some segmentation masks computed on Graz02, MSRC and PASCAL VOC 2006 databases, presented Figure 10. For each class, images are segmented into objects of interest and background regions. For the Graz 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 in Section 5.4 that on the PASCAL 2007 data set, the 20 object classes competing at the same time makes the problem much harder.

More segmentation results are shown Figure 1 and Figure 13. Our algorithm automatically detects and segments objects accurately despite large intra-class appearance variations, even with weak supervision (training with bounding boxes only). Even in a multi-class framework, MSRC images are accurately segmented, however, the variation of object appearance is less significant that for the PASCAL 2006 data set. Indeed, we observed that the simple pure patch-based classification already performs well for these images.

5.4 Quantitative results

Here, we first briefly present results on the MSRC data set, before turning to the PASCAL VOC 2007 data set.

Due to its popularity we compared our method with results recently published on the MSRC data set. Note that the task is here different because the background is divided into

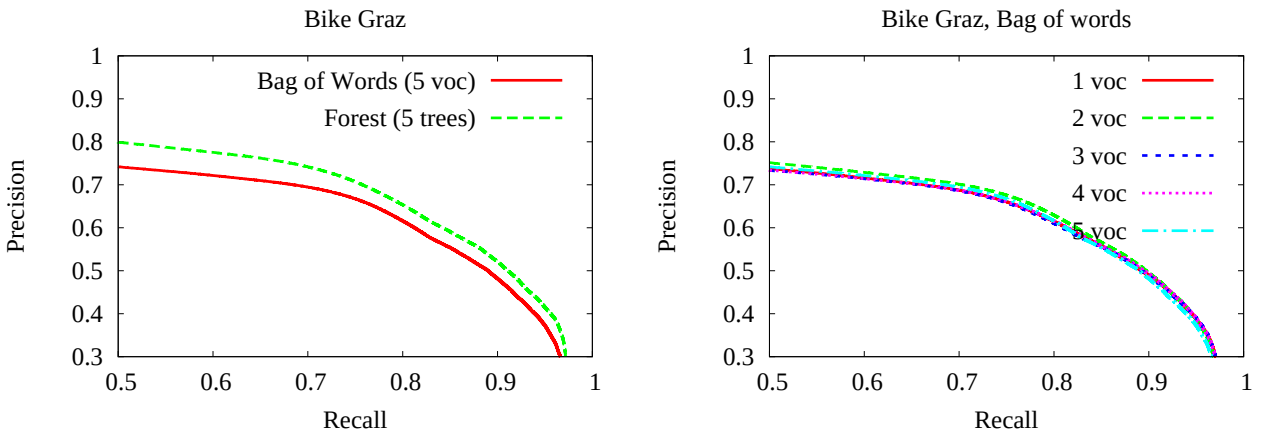


Fig. 7 Left: comparison between 5 k-means vocabularies and 5 trees based vocabulary. Right: influence on the number of vocabularies for k-means.

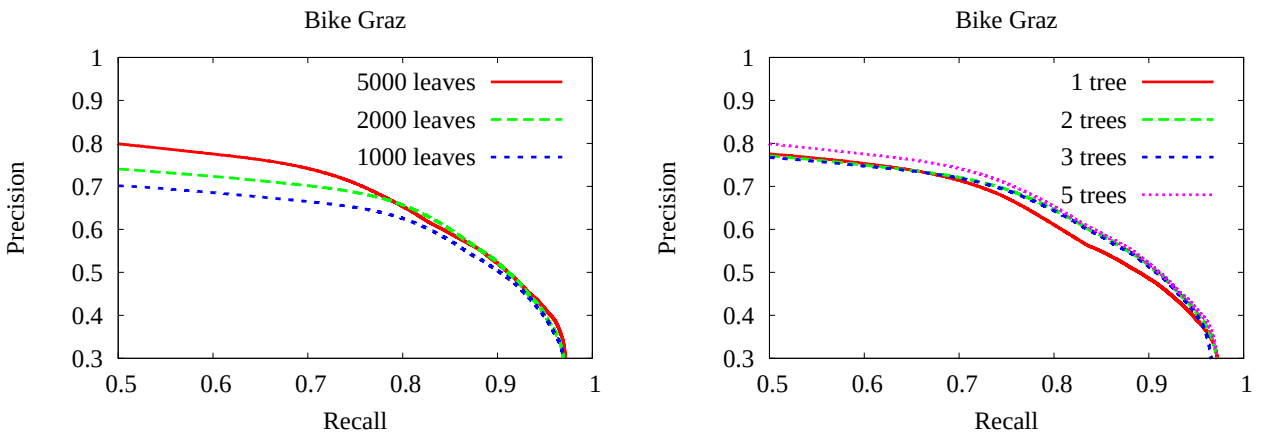
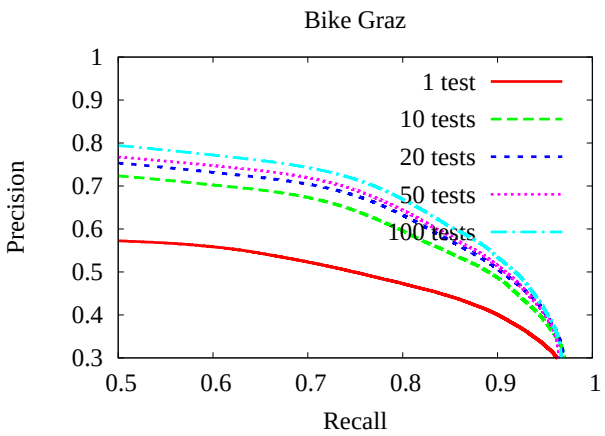


Fig. 8 Influence of the number of leaves per tree (left) and of the number of trees (right), on the accuracy of the final segmentation.



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

Fig. 9 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.

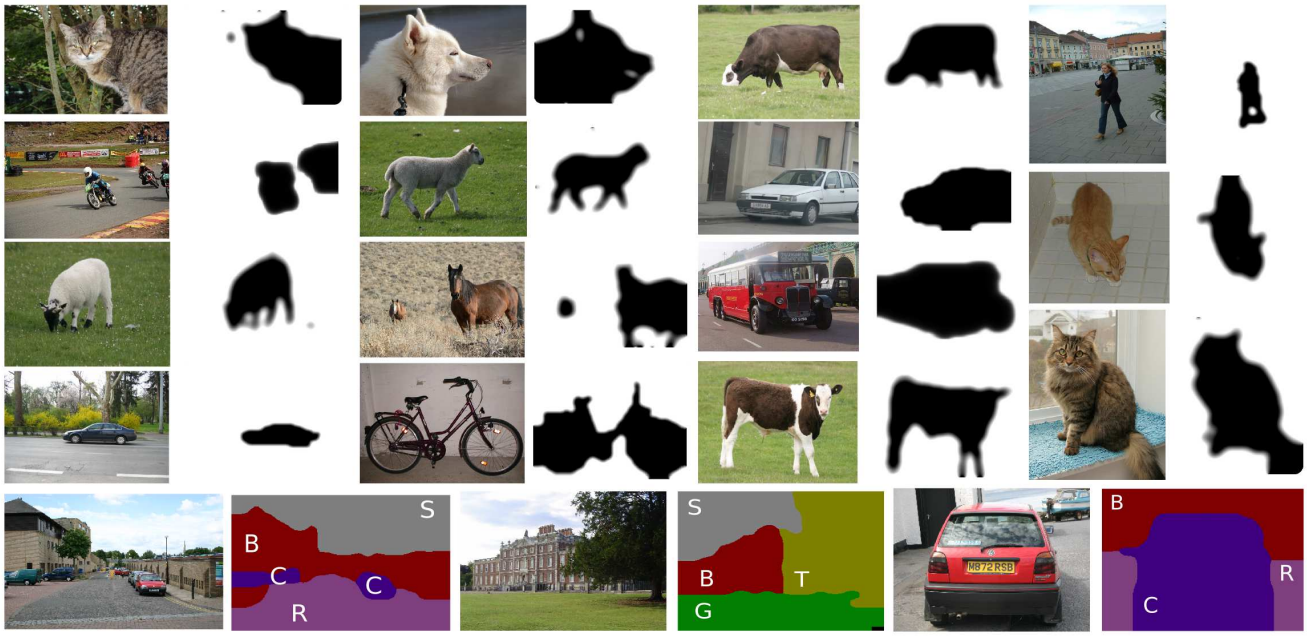


Fig. 10 Examples of segmentation obtained by our method on the Graz-02, PASCAL VOC 2006, and MSRC data sets (best viewed in color). For the last a color coding is used for the classes: building (B), car (C), grass (G), road (R), sky (S), and tree (T). For some classes (*e.g.* cats) category models are learned from bounding boxes only. We observe that even with complex backgrounds, the amount of confusion is limited.

	Cow	Sheep	Aeroplane	Face	Car	Bicycle	Sign	Bird	Chair	Cat	Dog	Body	Boat
TextonBoost [30]	58	50	60	74	63	**75**	35	19	15	**54**	19	**62**	7
MFAM [35]	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 Pixel-level classification accuracies for the 13 object categories of the MSRC data set, *i.e.* percentage of pixels correctly recognized.

several classes (grass, building, trees, etc.) so the goal is not figure/ground segmentation but full segmentation of images. As our method models instances as geometrical clusters of patches, it is not designed to deal with large background regions (stuff) surrounding these objects. That is why here we consider only objects (things) themselves in Table 1. It gives the classification accuracies of our algorithm on the 13 object categories of the data set. More precisely, for each class is computed the number of pixels correctly labeled for this class divided by the total number of pixels belonging to this class. We compared with the TextonBoost results [30], and with the Markov Field Aspect Model (MFAM) [35]. Our method gives comparable results, although it is not designed explicitly for this kind of task.

In its past three editions the PASCAL Visual Object Classes (VOC) challenge has evolved to be a major platform for comparison of current state-of-the-art methods for image categorization, object detection, and segmentation. 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 consider in this paper. 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 [9].

To estimate the model parameters we use all annotations; both segmentation masks and 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 rep-

resenting 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 produced segmentation masks are illustrated in Figure 11.

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. 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 [9] 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 [8]. 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'.

In addition to these two types of initialization, we 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 result obtained using detection algorithms, in which case the segmentation is simply given by the predicted object bounding box. Finally we report results obtained by three methods proposed since the challenge [7,28,31].

From these results, we can draw several conclusions. First, we see that for nearly all classes including training images with bounding box annotations (FT) brings a clear improvement. Second, the results demonstrate the importance of good initializations using the detector results (DI). Using the detector gives an overall improvement of about 20% 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. This behavior was also observed in [31], where the use of a detector also improved the accuracy of their method by 20%. They used the TKK detector [9] which outperformed our INRIA_PlusClass detector and thus obtained slightly better segmentation results. Third, we clearly outperform the best methods that entered in the challenge and have comparable or better results than the one proposed after the challenge [7,28,31]. We note that [7] uses a global image classifier which also guides the segmentation algorithm, this follows the same intuition than using the object detector.

In order to better understand the role of the detector, Figure 12 illustrates the behavior of the model on some images. Starting from the initial detections (first column), the segmentation method validates the object hypotheses and refines the object boundaries in most of the cases (segmentation results shown in the third column). This can be compared to the results obtained using local patch predictor initialization (last column). For the third image, we can clearly see both a *bicycle* and a *motorbike* detection. 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.

5.5 Instance based segmentation and limits of the method

Most evaluation campaigns only consider category level segmentation; instance based segmentation is usually not considered. The strength of our model lies in its ability to identify single object instances. Modeling different instances of the same category individually is of particular interest because it allows to fit an appearance model to each instance and make its description even more precise. This is illustrated on the first column of Figure 13, where two different car instances, with different colors, can each benefit from an accurate color appearance model. In this example, the Dirichlet process prior favors the creation of a second blob.

However, when objects are too close, the estimation of the number of instances fails, and multiple objects which are close to each other, or which are too similar to each other are considered as a single instance. See for example the two cows presented Figure 1 which are grouped in a single object. This behavior of our model can be explained by the fact that the Dirichlet process prior tends to limit the number of regions per image; unless there is enough evidence due to different appearances that are spatially coherent. In these situations an external object detector can be valuable to initialize our model with good estimate of the number of instances per category. See for example the second columns of Figure 13, in this image the detector returned multiple instances allowing the segmentation of the correct number of people. As a comparison, we considered for each class of the PASCAL VOC 2007 data set, images containing at least an instance of this category and computed the average number of instances per image, within this subset. Our model pro-

Fig. 11 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, with color coding shown in Figure 13).

	backgrd	plane	bicycle	bird	boat	bottle	bus	car	plant	sheep	sofa
FT+DI	49.4	20.5	70.4	23.5	16.5	28.7	22.7	58.4	22.0	23.7	27.9
ST+DI	57.2	13.6	35.1	19.6	10.6	23.8	16.8	56.8	14.4	17.8	24.1
FT+NI	15.0	17.7	9.4	1.6	15.9	4.8	10.2	25.1	38.0	8.9	4.2
ST+NI	21.0	11.7	10.0	3.6	15.5	8.7	10.7	17.4	3.4	8.5	8.7
Brookes	77.7	5.5	0.0	0.4	0.4	0.0	8.6	5.2	2.3	2.3	0.3
TKK	22.9	18.8	20.7	5.2	16.1	3.1	1.2	78.3	64.7	30.2	34.6
Texton Forests [31] DI	22	77	45	45	19	14	45	48	40	42	10
Texton Forests [31] NI	33	46	5	14	11	14	34	8	19	19	8
Multiple Segmentation [28]	59	27	1	8	2	1	32	14	11	26	1

	cat	chair	cow	table	dog	horse	moto	person	train	monitor	mean
FT+DI	65.5	28.2	10.4	0.9	3.7	65.4	51.8	60.1	65.2	65.5	37.2
ST+DI	63.1	25.0	10.6	0.6	4.0	41.2	55.3	64.1	46.2	59.7	31.4
FT+NI	15.2	23.8	7.5	10.6	20.7	15.7	21.9	27.6	4.9	17.5	15.1
ST+NI	7.4	21.2	7.8	5.8	15.7	14.3	11.3	40.5	3.9	18.1	12.6
Brookes	9.6	1.4	1.7	10.6	0.3	5.9	6.1	28.8	10.6	0.7	8.5
TKK	1.1	2.5	0.8	23.4	69.4	44.4	42.1	0.0	89.3	70.6	30.4
Texton Forests [31] DI	29	26	20	59	45	54	63	37	68	72	42
Texton Forests [31] NI	6	3	10	39	40	28	23	32	24	9	20
Multiple Segmentation[28]	14	4	8	32	9	24	15	81	28	17	20

Table 2 Segmentation accuracy (*i.e.* pixel-level classification accuracy) on the PASCAL VOC 2007 data set. The first four rows give the results obtained with our method using the full training set (FT), the small training set (ST), detector based initialization (DI), and naive initialization (NI). The two following rows, give best results among the submitted segmentation and detection methods respectively. The remaining rows correspond to methods proposed since the challenge. In [7] only the performance averaged over all classes is reported at 39.8.

duces an average of 1.73 while the ground truth shows an average of 1.63 instances per image.

We have seen in Figure 12 that mistakes made by the detector can be recovered by the segmentation algorithm. This is not always possible, c.f. the second column of Figure 13, where a person in the crowd was detected as a cat, or on the second line of Figure 12 where a dog was confused with a cat. As a last example, the third column of Figure 13 shows a sofa which is detected as car, this hypothesis being more consistent with its context. Another problem for most detection methods, is the detection of multiple instances where in reality there is only one. In particular it happens for unusually big objects, and the segmentation method does not always recover from such initializations. This is the case in the last row of Figure 13.

6 Discussion and conclusions

We conclude this paper with a discussion of our model, and indicate extensions to overcome some of its limitations.

Segmentation is commonly considered as an isolated problem: a given image has to be segmented in some 'meaningful' manner, without any supplementary information. Where 'meaningful' is often understood as segmenting at the level of objects, or their constituent parts. Much early work on segmentation tried to solve the task at a local level; clearly such methods can not resolve ambiguities in the local image features. Semantic grouping is required within the segmentation process, and category-level recognition can provide the necessary cues for this. Similarly, recognition requires accurate segmentation to avoid distraction from background

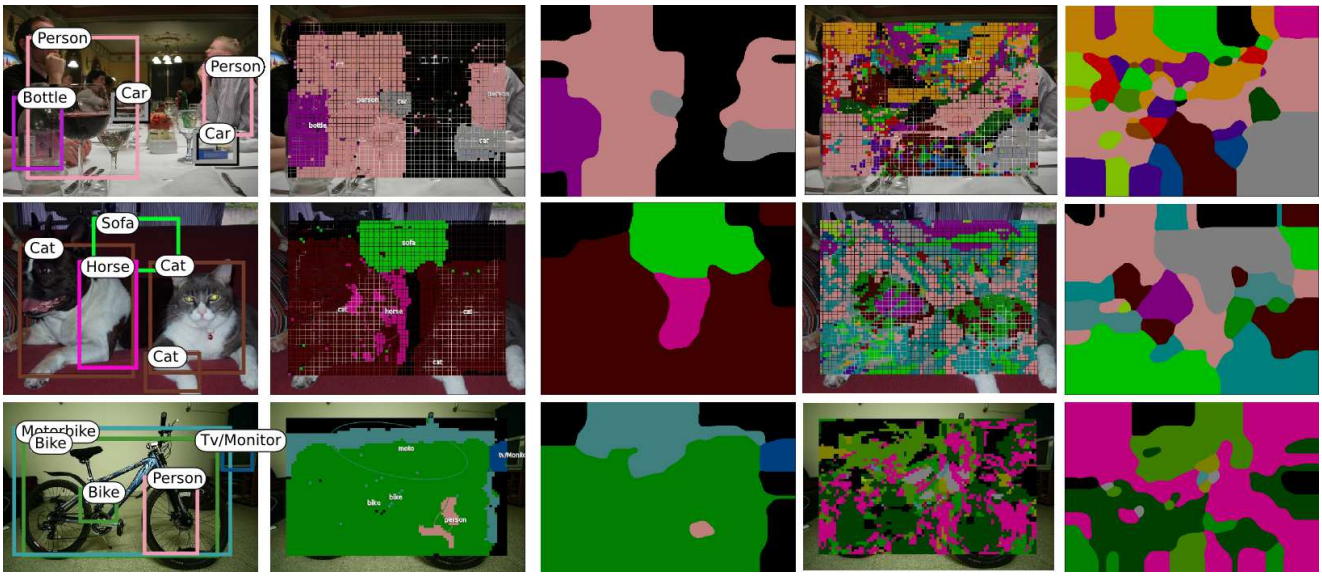


Fig. 12 Three example images from PASCAL VOC 2007. From left to right: (i) the original image with the detector results superimposed, (ii) category assignments after a few iterations, (iii) the final segmentation result produced from this initialization, (iv) class labels from patch-level initialization, and (v) the final result obtained using this initialization (best viewed in color, with color coding shown in Figure 13).

Fig. 13 Illustration of instance based segmentation. The role of the Dirichlet process prior and the detector, together with explanation of failures are described in the text. Ellipses represent blobs, and the color coding used in three of the images is shown on the right.

clutter and occluding objects. Our model couples these two processes, and the parameters of its category appearance models are estimated from manually segmented images. The estimate category models can then be applied to segment new instances of the same categories in other images.

Robust category-level recognition requires dealing with intra-class variations and imaging conditions such as occlusions, illumination changes, view point and scale variations. Our choice of patch descriptors ensures some level of invariance to illumination changes. Where some methods rely on rigid shape models for recognition, ours relies on a bag-of-words representations which are intrinsically robust to occlusions and non-rigid deformations. Our blob model does not impose hard constraints between object parts, but does implement accumulation of evidence on the object position and size to guide the assignment from patches to objects. The Dirichlet process over patch-to-blob assignments in our model is interesting because it introduces dependencies at an automatically adapted scale, which is determined by the size and number of the blobs. We can imagine using multi-scale patches which would probably improve the recognition ability of the model but increase its complexity.

Our experiments show the benefit of using a supplementary object category detector, which operates at a level of bounding boxes, to improve results when segmenting many object categories simultaneously. 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, as our model separates different object instances and handles multiple categories per image. Note that even if the final goal is to predict a class label per pixel, and not to identify all different instances of each

category, it can be beneficial to separately model the different instances. This is because it allows the modeling of instance specific appearance models, for color in our case, which can improve the segmentation accuracy. Furthermore, we experimentally find that we can successfully combine annotations in the form of pixel-level segmentation and bounding boxes, the latter being much easier to produce. Adding images annotated with bounding boxes leads to improved segmentation results.

In future work we want to further study the interplay between the instance specific and category level appearance models. In the current work low-level image cues are used by either the instance level or category level model, whereas in principle the features used by both models do not need to be disjoint. In particular it is interesting whether we could learn which features are useful at the instance level and which are useful at the category level. Furthermore, it is worthwhile to improve the capacity of the model to distinguish multiple instances of the same category which are very close to each other. Some form of geometric information should be included in the category level appearance models to resolve such ambiguities and improve segmentation results.

Acknowledgements We would like to thank E. Nowak for his help, and H. Harzallah for his category detection results.

References

1. D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
2. E. Borenstein and J. Malik. Shape guided object segmentation. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2006.
3. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *IEEE International Conference on Computer Vision*, 2001.
4. L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, 1984.
5. 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.
6. G. Csurka, C. Dance, L. Fan, J. Williamowski, and C. Bray. Visual categorizationwith bags of keypoints. In *ECCV workshop on Statistical Learning in ComputerVision*, 2004.
7. G. Csurka and F. Perronnin. A simple high performance approach to semantic segmentation. In *British Machine Vision Conference*, 2008.
8. N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *European Conference on Computer Vision*, 2006.
9. M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.
10. 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*, 2005.
11. 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.
12. P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, 2006.
13. T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1-2):177–196, 2001.
14. F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *IEEE International Conference on Computer Vision*, 2005.
15. M. Kumar, P. Torr, and A. Zisserman. OBJ CUT. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2005.
16. S. Kumar and M. Hebert. Discriminative random fields. *International Journal of Computer Vision*, 68(2):179–201, June 2006.
17. J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*, 2001.
18. D. Larlus and F. Jurie. Latent mixture vocabularies for object categorization. In *British Machine Vision Conference*, 2006.
19. 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.
20. B. Leibe and B. Schiele. Interleaved object categorization and segmentation. In *British Machine Vision Conference*, 2003.
21. A. Levin and Y. Weiss. Learning to combine bottom-up and top-down segmentation. In *European Conference on Computer Vision*, 2006.
22. Y. Li, J. Sun, C. Tang, and H. Shum. Lazy snapping. *ACM Transactions on Graphics*, 23(3):303–308, 2004.
23. D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
24. D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 26(5):530–549, 2004.
25. F. Moosmann, E. Nowak, and F. Jurie. Randomized clustering forests for image classification. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 30(9):1632–1646, 2008.
26. R. Neal. Markov chain sampling methods for dirichlet process mixture models. Technical Report 9815, Dept. of Statistics, University of Toronto, Sep 1998.
27. P. Orbanz and J. Buhmann. Smooth image segmentation by nonparametric Bayesian inference. In *European Conference on Computer Vision*, 2006.
28. C. Pantofaru, C. Schmid, and M. Hebert. Object recognition by integrating multiple image segmentations. In *European Conference on Computer Vision*, 2008.
29. C. Rother, V. Kolmogorov, and A. Blake. GrabCut: interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, 2004.
30. 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*, 2006.
31. S. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2008.
32. A. Storkey and C. Williams. Image modeling with position-encoding dynamic trees. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 25:859–871, 2003.
33. 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.
34. J. van de Weijer and C. Schmid. Coloring local feature extraction. In *European Conference on Computer Vision*, 2006.
35. J. Verbeek and B. Triggs. Region classification with markov field aspect models. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2007.

36. J. Verbeek and B. Triggs. Scene segmentation with CRFs learned from partially labeled images. In *Advances in Neural Information Processing Systems*, 2008.
37. J. Winn and N. Jojic. Locus: Learning object classes with unsupervised segmentation. In *IEEE International Conference on Computer Vision*, 2005.
38. 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*, 2006.