

# Fast Spatial Pattern Discovery Integrating Boosting with Constellations of Contextual Descriptors

Jaume Amores

Universitat Autònoma de Barcelona  
Spain

Nicu Sebe

University of Amsterdam  
The Netherlands

Petia Radeva

Universitat Autònoma de Barcelona  
Spain

## Abstract

*We present a novel approach for fast object class recognition incorporating contextual information into boosting. The object is represented as a constellation of generalized correlograms that integrate both information of local parts and their spatial relations. Incorporating the spatial relations into our constellation of descriptors, we show that an exhaustive search for the best matching can be avoided. Combining the contextual descriptors with boosting, the system simultaneously learns the information that characterize each part of the object along with their characteristic mutual spatial relations. The proposed framework includes a matching step between homologous parts in the training set, and learning the spatial pattern after matching. In the matching part two approaches are provided: a supervised algorithm and an unsupervised one. Our results are favorably compared against state-of-the-art results.*

## 1 Introduction

Object class recognition has been a challenging area of pattern recognition and computer vision. Difficulties arise in the variability of object appearance, accidental conditions, and existence of clutter in the images. All this variability demands efficient learning techniques able to summarize key properties of the object under different scenarios. There has been several approaches recently to address object class recognition in cluttered scenes. Among them, characterizing the object as a collection of parts and their spatial arrangement has proved to be a promising direction [1, 15, 8, 10, 3, 12, 16]. In this work, we focus on efficient spatial pattern of local parts discovery, therefore we regard the object as a constellation of parts together with their mutual spatial relationships. Recently, Agarwal et al. [1] proposed the use of a dictionary of parts and a Winnow algorithm for learning active features of the object. Schneiderman [15] propose to use an efficient Bayesian network for learning the spatial arrangement.

Fergus et al. [8] used a principled unsupervised statistical learning of constellation of parts and spatial relations, and report results on several categories of objects with clutter. They use separate probabilistic models for the appearance of parts, and the spatial configuration with maximum-likelihood under expectation-maximization. In their work, every possible match between parts in the model and parts in the images is tested, which leads to an exponential cost. They propose to

compensate this cost by fast search methods such as  $A^*$ , and they finally report a maximum cost of 36 hours for the training stage. Regarding the representation of the parts, local properties are used such as the local appearance. The spatial relations are simply described by the difference in spatial position. Other authors [10] propose the use of Attribute Relational Graphs (ARGs) for object recognition and spatial pattern discovery. ARG is a common representation for describing an object as local properties of parts and spatial relations between them: parts of the object are represented by vertices of the graph, and relations between parts are represented by arcs between vertices. Vertices and arcs have associated feature vectors that describe local information and contextual (spatial) information respectively. Matching between features (parts in an image) and parts in the model is performed by relaxation. Relaxation has a cost of order  $O(KN^2M^2)$ , where  $N$  and  $M$  are the number of vertices in the sample and model ARG, respectively, and  $K$  is the number of iterations until convergence. This cost is much lower than the one obtained by combinatorial matching but is still prohibitive for a number of vertices of two orders of magnitude, which normally arises in complex images. An important contribution of the latter method is the theoretical derivation of expectation-maximization for modelling ARGs. An important difference between this method and other recent methods (e.g. [8]) is that the former estimates a probability for the joint distribution of spatial relations, while the latter uses separate PDFs for each spatial relation.

We aim at building a feature space in which we can gather both the local information describing the parts and the spatial relations among every possible pair of parts. Classical contextual representations such as ARGs and constellation of parts deal separately with these two forms of information: local information is represented by feature vectors associated to each part and contextual information is represented by a set of relative spatial vectors. The way to deal with the matching of parts from an instance object to the model is either testing every possible matching (exponential cost) or using the estimated parameters and spatial relations in a structural matching using probabilistic relaxation which also has a high cost. In this work, we propose to use a novel representation of the constellation of parts model, where now the feature vectors associated to each part describe not only the local properties of the part but, at the same time, the context of the part.

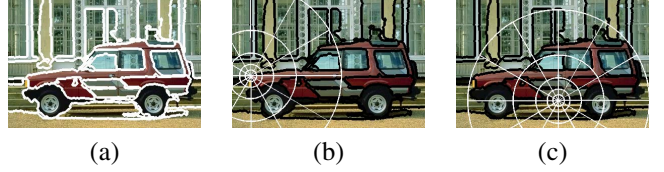
This representation has two important advantages over previous ones. First, the matching between parts is made much easier: when comparing parts from the model and the image, we are comparing simultaneously the local properties of both parts and the spatial relations from these parts to the rest of the object, which adds spatial coherence into the similarity. This makes suitable fast non-iterative matching techniques that simply look for the most similar part of the model, for example by using Chamfer distance. This already introduces spatial coherence (i.e. without needing to examine neighboring matchings as in relaxation techniques). Second, we can learn simultaneously both the spatial configuration and local appearance in an efficient way when combining this representation using an integrated feature selection and learning technique like AdaBoost. Note that we are gathering in the same feature vector local information and contextual information. The feature selection algorithm simply has to select the combination of elements that characterize the local appearance of an individual part and its context, just by examining a single vector and without needing to check what are the matchings between all the parts.

The main contribution of this paper is in integrating a discriminant representation of the image with a fast, efficient learning algorithm with feature selection. The image representation provides an integrated characterization of properties of each part of the object and their relationships using generalized correlograms into a constellation of parts framework. This information combined with boosting leads to an efficient object recognition scheme dealing with the spatial pattern of the object.

## 2 Image Representation

In this section we explain the generalized constellation of correlograms used for image representation, which was previously introduced in [2] using a different notation. Let an image  $I_k$  be represented by a constellation of  $U_k$  object parts, expressed as  $H_k = \{\langle o_i, \vec{h}_i, \vec{x}_i \rangle\}_{i=1}^{U_k}$ . The  $i$ -th detected part is represented by the tuple  $\langle o_i, \vec{h}_i, \vec{x}_i \rangle$ , where  $o_i$  is the label identifying the part,  $\vec{h}_i$  are the properties describing the part, and  $\vec{x}_i$  is its spatial position in the image. Due to clutter, parts in  $H_k$  might correspond to different objects. Let  $X_k = \{\vec{x}_i\}_{i=1}^{U_k}$  be the set of spatial positions of parts from  $H_k$ . One way to obtain potential parts of an image is by extraction of interest points, also called features or key points [18, 8], this is also our approach.

For our purpose, it is important not to miss any informative location, and to perform a fast interest point extraction. By interest point we mean any point located at an informative position, such as the edges, we do not mean necessarily corners. Two levels of interest points are extracted. First we obtain a dense set of interest points representing potential parts of objects. From this dense set, we extract local information around each point. Let  $H_k^L = \{\langle o_i^L, \vec{h}_i^L, \vec{x}_i^L \rangle\}_{i=1}^{U_k^L}$  denote this dense set (do not confuse with the final representation  $H_k$ ). We extract local information as properties  $\vec{h}_i^L$  of these parts.



**Figure 1.** (a) Dense cloud of points covering interesting parts of the image (edges). (b)-(c) Log-polar spatial quantization of our correlogram.

Let  $X_k^L = \{\vec{x}_i^L\}_{i=1}^{U_k^L}$  be the dense set of positions from  $H_k^L$ . In our implementation, these positions are located at extracted contours of the image (Fig. 1(a)). From  $X_k^L$  we sample a much more sparse set of interest points  $X_k \subset X_k^L$  covering the different locations from which we measure the relative spatial distribution of local properties in  $H^L$ .  $X_k$  contain the positions of our final constellation  $H_k$ . Each point  $\vec{x}_j \in X_k$  is the position of  $o_j$ . We associate as descriptor  $\vec{h}_j$  a correlogram that measures the joint distribution of spatial relations  $(\vec{x}_i^L - \vec{x}_j)$  and local properties  $\{\vec{h}_i^L\}_{i=1}^{U_k^L}$ . Let us express the spatial relation  $(\vec{x}_i^L - \vec{x}_j)$  in polar coordinates:  $(\alpha_{ij}, r_{ij})$ , and the  $d$  local properties as  $\vec{h}_i^L = (l_{i1}, l_{i2}, \dots, l_{id})$ . The joint distribution is measured by a histogram based on a partition of the  $d + 2$  dimensional space with vectors  $\vec{v}_{ij} = (\alpha_{ij}, r_{ij}, l_{i1}, l_{i2}, \dots, l_{id}), i = 1, \dots, U_k^L$ . The partition of this space is obtained by intersection of separate partitions made for each individual dimension. Let  $B_w$  be the  $w$ -th bin in the final  $d + 2$  space. The  $j$ -th correlogram is expressed as:  $\vec{h}_j(w) = \frac{1}{U_k^L} |\{v_{ij} \in B_w, i = 1, \dots, U_k^L\}|$ , i.e. the  $w$ -th bin of  $\vec{h}_j$  counts the number of vectors  $\vec{v}_{ij}$  falling into this bin. Note that this space contains vectors that express spatial relations and local properties, and thus the resulting descriptor  $\vec{h}_j$  is a correlogram of local properties in  $H_k^L$  considering their spatial distribution around the point of reference  $\vec{x}_j$ . As  $X_k \subset X_k^L$ , we are describing in the same vector  $\vec{h}_j$  attached to  $o_j$  the local properties of  $o_j$ , the local properties of the rest of parts in the dense set  $H^L$ , and the spatial distribution of these parts relative to  $o_j$ .

The dense set of interest points in  $X_k^L$  is obtained by extracting the contours from an over-segmentation with k-means and subsequent postprocessing that obtains spatially contiguous blobs. The sparse set of points in  $X_k$  is sampled from  $X_k^L$  keeping points with maximum spatial distance to each other, so that  $X_k$  covers points of view from different angles of the image (Fig. 1(b)-(c)). An important characteristic of our implementation is that it is fast, and the results show that allows accurate representation. For the spatial dimensions, we use the same log-polar spatial quantization as the shape-context correlogram of Belongie et al [4] (Fig. 1(b)-(c)). This makes the descriptor  $\vec{h}_j$  focus more on local properties around  $o_j$  (local context) than on far context. The dimensions regarding the local properties  $l_{i1}, l_{i2}, \dots, l_{id}$  are linearly quantized; we explain below each of them in turn.

As local information, local structure and color around a small neighborhood are used. As local structure, the local

direction of the edges is used. Specifically, the angle is measured along the curve formed by contours. After smoothing the contours, the angle is taken modulus  $\pi$ , and we make a quantization into 4 bins. The color is linearly quantized and mapped into one dimension. We perform a very coarse quantization of the R,G,B space into 3, 2, 2 bins to avoid large feature vectors in the final histogram. As there is not only one dominant color around the local part  $o_i^L$ , we take every color around a small neighborhood and consider the proportion of this color in this neighborhood, thus a local color histogram is taken. In this way, we are performing a fuzzy assignment of the part  $o_i^L$  to bins of the (local) color space, using the local color histogram  $h_i^c : \{1, 2, \dots, 12\} \rightarrow [0, 1]$  as the *color membership function* of  $o_i^L$ .

Different authors have used correlograms [11, 4]. The common feature is to use pixel-level properties, traditionally only color, considering every pixel in the image. High-level entities such parts of objects are not considered in their formulation. Authors do not consider constellations of their correlograms but aggregate all the descriptors into one single (spatial) histogram for the image. Belongie et al. [4] use constellations of shape contexts but do not use any local information, they describe binary contours by the presence of a spatial position. The definition presented here can be considered a generalization of correlograms into a constellation of parts framework. One drawback of the spatial quantization we use is that it must be scaled with the size of the object to provide scale invariance. This scaling is done by normalizing the distances  $r_{ij}$  by the size of the object. As we do not know a priori the size of our objects, we must compute the contextual descriptors for different scales fixed a priori. Let  $n_s$  be the number of scales (experimentally we chose  $n_s = 7$ ). The final representation of the image  $I_k$  is expressed in bold typeface as  $\mathbf{H}_k = \{H_k^s\}_{s=1}^{n_s}$ , where  $H_k^s$  is the set of parts of  $I_k$  with contextual descriptors  $h$  scaled according to scale  $s$ .

### 3 Learning Multiple Contextual Representations with Boosting

Recently, there has been a lot of research in classifiers that have good generalization performance by maximizing the margin. Examples of such classifiers are boosting [9] and SVMs [5]. Using boosting provides a good theoretical and practical convergence to a low error rate in few iterations, its speediness being one of the major advantages over other algorithms such as SVM. The explained representation is suitable for combination with a feature selection and learning method such as AdaBoost with weak classifiers based on single dimensions, that proved to be very efficient [14, 17]. By learning the relevant dimensions of vectors  $\vec{h}$  defined in section 2, we are simultaneously learning the properties characterizing every part of the object and their mutual spatial relations. Boosting will select the joint distribution of color and local structure for some (relative) spatial region if the color is characteristic in this region for most of the samples, otherwise using only the local structure will lead to a lower error (see

Fig. 2).

In our framework, the model of one object is expressed as  $\Omega = \{\langle \omega_i, \vec{\varphi}_i \rangle\}_{i=1}^M$ , where  $\omega_i$  is the label of one model part,  $\vec{\varphi}_i$  are the parameters learnt by the classifier for this model part, and  $M$  is the number of model parts. We denote as  $l_i^\omega(o_j | o_j \in H_k^s)$  the likelihood that part  $o_j \in H_k^s$  from image  $I_k$  with scale  $s$  represents the model part  $\omega_i$ . We denote as  $L_i^\omega(H_k^s)$  the likelihood that any  $o_i$  in  $I_k$  with scale  $s$  represents the model part  $\omega_i$ . As we are using contextual descriptors,  $\omega_i$  also represents the whole model object according to one particular point of view. Therefore,  $L_i^\omega$  conveys a piece of evidence of the existence of the model object according to the point of view  $\omega_i$ .  $L_i^\omega(H_k^s)$  is the likelihood that *any*  $o_j \in H_k^s$  represents  $\omega_i$ , we apply as OR rule the maximum so that  $L_i^\omega(H_k^s) = \max_{o_j \in H_k^s} l_i^\omega(o_j | o_j \in H_k^s)$ . This can also be regarded as matching  $\omega_i$  with some  $o_m \in H_k^s$ , which is expressed as  $M_i^\omega(H_k^s) = o_m = \arg \max_{o_j \in H_k^s} l_i^\omega(o_j | o_j \in H_k^s)$ .

Based on the individual likelihoods  $L_i^\omega$ , we denote as  $L^\Omega(H_k^s)$  the likelihood that the object exists in  $I_k$  with scale  $s$ , according to the whole model  $\Omega = \{\langle \omega_i, \vec{\varphi}_i \rangle\}_{i=1}^M$ . As we want all the model points of view  $\omega_i$  of the object to contribute to this likelihood, we use as combination rule the mixture  $L^\Omega(H_k^s) = \sum_{i=1}^M \frac{1}{M} L_i^\omega(H_k^s)$ .

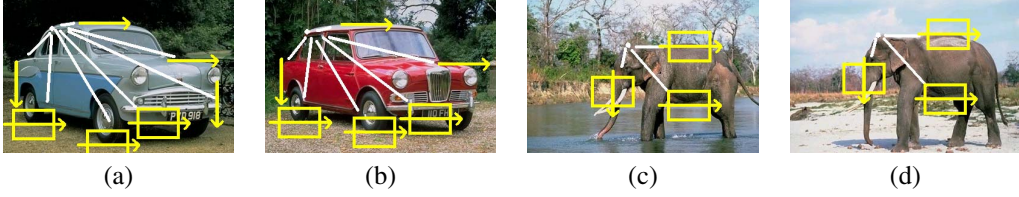
Recall that the image  $I_k$  is represented by different scales  $A_k = \{H_k^s\}_{s=1}^{n_s}$ . The likelihood that the object exists with *any* scale in the image representation  $A_k$  is expressed as  $L_f^\Omega(A_k) = \max_{H_k^s \in A_k} L^\Omega(H_k^s)$ , where we have applied again the maximum as OR rule. Again, this can be regarded as matching the model object with some *scaled* representation  $H_k^m$  in  $A_k$ , which we express as  $M_s(A_k) = H_k^m = \arg \max_{H_k^s \in A_k} L^\Omega(H_k^s)$ .

## 4 Matching of Homologous Parts

As explained, each model part  $\omega_i$  represents the context of the object around a different part of the object which represents a particular point of reference. For building it we must provide descriptors corresponding to homologous points, i.e. perform matching. In this work we present two different methods to deal with this: a supervised method and an unsupervised structural matching with expectation-maximization.

### 4.1 Supervised matching

To learn the model  $\Omega$ , AdaBoost is applied over each separate model point of view  $\omega_i$ . This requires a separate training set for each  $\omega_i$ . We denote as  $T_i$  this training set.  $T_i$  contains as positive samples the parts  $o_j$  matching  $\omega_i$  with the correct scale in every *positive* image (i.e. an image containing an object of the category we are learning). As negative samples  $T_i$  contains every part  $o_j$  with every scale in *negative* images. The problem of matching is solved in two stages. First, robust matchings are extracted from a small set of manually segmented images from the training set (we will see that very few images are enough). This is carried out by performing non-rigid registration [4] over these manually segmented



**Figure 2.** Learning the features characterizing parts together with the context. The rectangles with arrows symbolize parts from which both local structure and color are characteristics. The single arrows symbolize parts from which only local structure is characteristic. In (a)-(b), the instances of car have only three relative parts whose color is learnt, the rest are characterized by local structure. In (c)-(d) the instances of elephant have all the parts with characteristic color and local structure

images, which obtains an initial training set  $T'_i$  for each  $\omega_i$ .  $T'_i$  contains as positive instances only those from the manually segmented subset, but has many negative instances, as we use every part with every scale in every negative image. This allows to discard a lot of structures from clutter. We learn an initial model part  $\omega_i$  with  $T'_i$ . With the learnt model part, we can now match corresponding parts  $o_j$  with corresponding scales in the rest of images not segmented manually to construct the final big training set  $T_i$ . Registration is not robust in clutter, therefore we match  $\omega_i$  with those  $o_j$  that have high likelihood according to the previous learning. We apply in every positive image first the scale matching  $M_s(\mathbf{H}_k)$  and then we apply the part matching in the appropriate scale  $M_i^\omega(M_s(\mathbf{H}_k))$  (see the expressions above). Finally, we train again the model with the complete training set  $T_i$  and obtain the final classifier for the whole model object  $\Omega$ .

## 4.2 Unsupervised matching

One drawback of the previous approach is that we must ask the user to segment a certain number of images (although we will see in the results that it is enough to take quite a small number of images). We present here an alternative method for obtaining the matching. This method is based on the expectation-maximization developed by Hong and Huang [10] for probabilistic ARG modelling. An important difference is that we do not use relations between parts in the ARG because we are already taking into account these relations by using a contextual descriptor attached to each part. This reduces the computational cost from  $O(N^2M^2)$  to  $O(NM)$ , where  $N$  is the number of vertices (parts  $o_j$  in our case) in the image and  $M$  is the number of vertices in the model. Furthermore, we do not use relaxation as matching method but instead let a soft-assign matching between each vertex in the sample image and each one in the model. Relaxation is meant for introducing spatial coherence into neighboring matches when the vertices only have local properties, i.e. the individual matchings do not consider spatial relations. Finally, we extend the method to deal with several scaled representations of each image.

The original motivation of the method is to learn a spatial pattern  $Z$  that is governed by some probability distribution  $f(\mathbf{H}|Z)$ , where  $\mathbf{H}$  is some instance of  $Z$ . In order to consider multi-modal distributions, a mixture of parametric densities is utilized. We assume that  $Z$  consists of a set of parametric model components  $\{\Omega_t\}_{t=1}^T$ , where  $T$  is

the number of components. Assuming a mixture we obtain  $f(\mathbf{H}|Z) = \sum_{t=1}^T \alpha_t p(\mathbf{H}|\Omega_t)$ , where  $\alpha_t$  is the weight of the  $t$  component.

Using the same notation as in section 2, we have a sample constellation  $\mathbf{H}_k$  per each image  $I_k$ , where  $k = 1, \dots, K$ . As introduced in section 3, our model constellation is denoted  $\Omega$ , the difference now is that there is more than one model component,  $\Omega_t$ ,  $t = 1, \dots, T$ , in order to cope with the multimodality in the Gaussian Mixture distribution. Also, each model component has several scales,  $\Omega_t = \{\Omega_t^r\}_{r=1}^{n_s}$ , where  $n_s$  is the number of scales. Each scaled model component consists of model nodes (parts)  $\omega_i$  and associated parameters  $\vec{\varphi}_i$ ,  $\Omega_t^r = \{\omega_i, \vec{\varphi}_i\}_{i=1}^{N_t}$ . Now the parameters  $\vec{\varphi}_i$  describe a Gaussian distribution, i.e.  $\vec{\varphi}_i = \langle \vec{\mu}_i, \Sigma_i \rangle$ .

In this framework, the expectation-maximization proceeds in two stages. The expectation step estimates the a posteriori probability  $p(c|\vec{x})$  of the component  $c$  of the mixture given the sample  $\vec{x}$ . Here we call this probability a *matching* probability  $P_m$ , we say that class  $c$  matches sample  $\vec{x}$  with probability  $P_m(c|\vec{x})$ . As opposed to conventional expectation-maximization, here the matching is performed in three different levels. At the top level a whole sample  $\mathbf{H}_k$  matches a whole component model  $\Omega_t$  with probability  $P_m(\Omega_t|\mathbf{H}_k)$ . At the scale level,  $P_m(\Omega_t^r|\mathbf{H}_k^s)$  the matching is given between scales of  $\mathbf{H}_k$  and  $\Omega_t$ . The last matching is at the node level,  $P_m(\omega_i|o_j)$ . These matching probabilities are obtained as follows. The node matching  $P_m(\omega_i|o_j)$  is based on the probability  $p(o_j|\omega_i) \sim N(\vec{\mu}_i, \Sigma_i)$ , which follows a normal distribution. Here we simply use as *soft* matching  $p(o_j|\omega_i)$  normalized so that the sum for all the nodes  $\omega_i \in \Omega_t^r$  is 1. Hong et al. propose to use probabilistic relaxation [6]. This is necessary whenever the nodes are described by only local information, so that relaxation is used for obtaining spatial coherence (i.e. the matching between two nodes is spatially coherent between the matching of neighbors). In our approach those relations are directly considered in  $p(o_j|\omega_i)$ , because the node is described by a contextual descriptor. This avoids the high computational cost of relaxation,  $O(n^4)$ , if  $n$  is the number of nodes in sample and model constellation. The scale matching is obtained based on the probability

$$p(H_k^s|\Omega_t^r) = \sum_{o_j \in H_k^s} \sum_{\omega_i \in \Omega_t^r} P_m(\omega_i|o_j)p(o_j|\omega_i).$$

Based on  $p(H_k^s|\Omega_t^r)$  we apply relaxation over scales to obtain

the scale matching  $P_m(\Omega_t^r|H_k^s)$ , in order to obtain scale coherence in the matching based on relations bigger and smaller between scales. Finally, the top level matching is computed as:

$$P_m(\Omega_t|\mathbf{H}_k) = \frac{p(\mathbf{H}_k|\Omega_t)}{\sum_{t=1}^T p(\mathbf{H}_k|\Omega_t)}$$

$$p(\mathbf{H}_k|\Omega_t) = \sum_{s=1}^{n_s} \sum_{r=1}^{n_s} P_m(\Omega_t^r|H_k^s)p(H_k^s|\Omega_t^r)$$

Given the matching probabilities, the maximization step updates the parameters  $\vec{\mu}_i$  and  $\Sigma_i$  of the model components

$$\vec{\mu}_i = \frac{\sum_{k=1}^K \sum_{s=1}^{n_s} \sum_{o_j \in H_k^s} \vec{h}_j P_m(\omega_i|o_j) P_m(\Omega_t^r|H_k^s) P_m(\Omega_t|\mathbf{H}_k)}{\sum_{k=1}^K \sum_{s=1}^{n_s} \sum_{o_j \in H_k^s} P_m(\omega_i|o_j) P_m(\Omega_t^r|H_k^s) P_m(\Omega_t|\mathbf{H}_k)}$$

$$\Sigma_i = \frac{\sum_{k=1}^K \sum_{s=1}^{n_s} \sum_{o_j \in H_k^s} d_{ji} d_{ji}^T P_m(\omega_i|o_j) P_m(\Omega_t^r|H_k^s) P_m(\Omega_t|\mathbf{H}_k)}{\sum_{k=1}^K \sum_{s=1}^{n_s} \sum_{o_j \in H_k^s} P_m(\omega_i|o_j) P_m(\Omega_t^r|H_k^s) P_m(\Omega_t|\mathbf{H}_k)}$$

where  $\vec{h}_j$  is the descriptor associated to node  $o_j$ , and  $d_{ji} = (\vec{h}_j - \vec{\mu}_i)$ . PCA reduction was applied to the descriptors  $\vec{h}_j$  in order to make the estimation more robust.

Based on matching probabilities, a crisp matching is obtained by solving the hungarian assignment method [13]. A unified matching to only one model component is obtained by matching the nodes of the components to a centroid. As a result we obtain a training per each model node  $\omega_i$ . The method above follows the one derived in [10], where the difference is that relaxation at node level is avoided, and the relations are implicit in the description of the nodes. Hong and Huang use explicit relations that have associated distributions, which involves estimating  $O(n^4)$  parametric distributions if  $n$  is the number of nodes in the sample and model constellations. Both relaxation and explicit relations lead to a prohibitive cost in cluttered images, where the number  $n$  of landmarks must be large enough.

## 5 Results

We use the database collected by Fergus et al. [8] which consists of 7313 images and is used for general object recognition in cluttered scenes where most of the objects have a typical bi-dimensional arrangement. Illustrations of the categories can be found in [8] and [7]. The car side category has also been used as test by Agarwal et al. [1].

We compare our results with the one obtained in [8] on the same database. To have a fair comparison, we follow their approach of classifying images from one category versus the images from the background. The training set consists of half of the images in the positive category and half of the images in the negative category. The test is performed with the other half of both categories. In our case, we take 10 images from the positive set and perform a hand-segmentation on them; the contours from the rest (390 images) being extracted automatically. All the tests are made using as training set 400 positive images and 400 negative images. The negative category is a set of images collected using a web-search engine with the keyword “things”. As it has 522 images, only 122 images can be used as test. We perform a cross-validation procedure with rounds of 100 positive images and 100 negative images to obtain a comparison with 800 test images (400 positive and 400

negative), and provide an average result based on the resulting  $4 \times 4$  scores (4 to include 400 positive images in rounds of 100, and 4 to include 400 negative images). For the car (rear) category, we use as negative set images of roads without cars, reported in [8]. The negative set contains now 1370 images, and we perform one round picking randomly 400 images as training and 400 as test. The positive images in the training and test are the same as the one in [8]. Finally for the car (side) category, that are gray-level images, we use the same procedure using as negative images a set of gray-level backgrounds used in [8]. The gray-level is quantized with 16 levels to obtain the correlograms.

The classification hit rate is measured using the receiver-operating characteristic (ROC) equal error rates:  $p(\text{True Positive})=1-p(\text{False positive})$ . Table 1 presents results comparing our method against the constellation used by Fergus et al. in [8], they also report results with other approaches using the same data set (see reference). Without non-optimized code, the learning stage takes at most 4 hours, which is not much compared to the time spent by the E-M algorithm used by Fergus (36 hours). In all the categories except the spotted cat and face, our method outperforms the one reported by Fergus et al. The spotted cat has very different poses which makes the spatial quantization not so suitable. However, the inclusion of other local properties makes boosting focus more on local information than contextual information, so that not bad results are obtained. For the car (side) category the result is a recall-precision equal error.

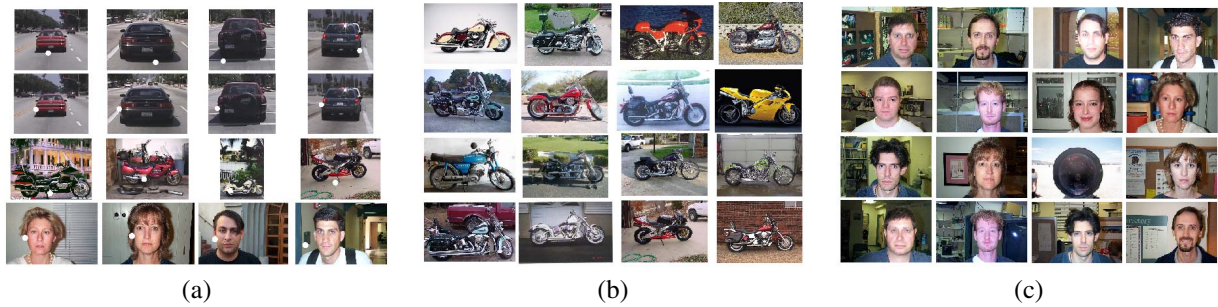
Table 2 compare the results using supervised and unsupervised matching. Using the E-M algorithm is more expensive, taking at most 15 hours. However, we not force the user to segment which is also important.

Category	Fergus	Boosting Context
Car(Rear)	90.3%	96.9%
Plane	90.2%	94.5%
Leaf	-	96.3%
Motorbike	92.5%	95.0%
Face	96.4%	89.5%
Cat	90.0%	86.5%
Car (Side)	88.5 %	90.0 %

**Table 1.** ROC equal error rates measures with supervised matching, except for the car (side) where it is a recall-precision equal error.

Category	Supervised Matching	Unsupervised Matching
Plane	94.5%	94.0%
Leaf	96.3%	87.5%
Motorbike	95.0%	91.42%
Face	89.5%	86.0%
Cat	86.5%	93.0%

**Table 2.** ROC equal error rates measures with supervised and unsupervised matching



**Figure 3.** Model part matching with part from instances (a), top ranked images classified as motorbikes (b), and faces (c)

Fig. 3(a) shows several matches from a part of the learnt model to a matched part across instances of the object, using as matching the two-stage boosting approach. In the car category (first and second rows) we can see that one of the parts is matched with the same shadow beneath the car in the instances. The same instances have another part matched always near the red light of the left. In the motorbike category we show a matching part across images with heavy cluttered. Despite the noise in the instances, the model is able to learn the relative position of the matching. Finally, a matching part across images of faces is shown, where usually the part is near the ear of the face.

In fig. 3(b)-(c) we show the top sorted images according to the classification score. Retrieved motorbikes at this similarity ranking show a heavy clutter and still there are no incorrect matches. Faces show the first incorrect match at position 60, the incorrect match being similar in shape.

## 6 Discussion

We have introduced an object class recognition system that is able to learn the characteristic parts of the object and their spatial relationship in the presence of clutter. We showed that incorporating contextual information and boosting we achieved very good results compared to the approach of Fergus et al. [8]. Our novel contribution is to propose an efficient object class recognition framework that incorporates a novel constellation of contextual descriptors into an efficient boosting algorithm used with feature selection.

For future research, we would like to enrich the feature space by combining the log-polar spatial quantization with other types of spatial quantization less sensitive to shape, in order to be able to recognize the same object under different spatial configurations (for example a dog with different poses). For example, if we only take into account the distances and avoid the angles the descriptor is more robust to different shapes. By boosting we can combine a descriptor sensitive to different shapes and a (contextual) descriptor robust against shape variations, and learn if the object is very structured (using then a finer spatial quantization) or not so structured (using a coarser quantization). It is also important to incorporate a method that speeds up the searching process. This can be done easily if we take advantage of the sparseness of the data, and use suitable approaches such as searching in inverted files.

## References

- [1] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *PAMI*, 26(11):1475–1490, 2004.
- [2] J. Amores, N. Sebe, P. Radeva, T. Gevers, and A. Smeulders. Boosting contextual information in content-based image retrieval. In *ACM Int'l Workshop MIR*, pages 31–38, 2004.
- [3] E. Bart and S. Ullman. View-invariant recognition using corresponding object fragments. In *Proc. ECCV*, 2004.
- [4] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 24(24):509–522, 2002.
- [5] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.
- [6] W. Christmas, J. Kittler, and M. Petrou. Structural matching in computer vision using probabilistic relaxation. *PAMI*, 17(8):749–764, 1995.
- [7] L. Fei-Fei, R. Fergus, and P. Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *ICCV*, volume 2, pages 1134–1142, 2003.
- [8] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003.
- [9] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *ICML*, page 148156, 1996.
- [10] P. Hong and T. S. Huang. Spatial pattern discovery by learning a probabilistic parametric model from multiple attributed relational graphs. *J. Discrete Applied Math.*, 139(1-3):113–135, 2003.
- [11] J. Huang, S. Kumar, M. Mitra, W. Zhu, and R. Zabih. Image indexing using color correlograms. In *CVPR.*, pages 762–768, 1997.
- [12] R. Nelson and A. Selinger. A cubist approach to object recognition. In *ICCV*, pages 614–621, 1998.
- [13] C. Papadimitriou and K. Stieglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice Hall, 1982.
- [14] R. E. Schapire and Y. Singer. Improved boosting using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.
- [15] H. Schneiderman. Learning a restricted bayesian network for object detection. In *CVPR*, pages 639–646, 2004.
- [16] A. Torralba, K. Murphy, W. Freeman, and M. Rubin. Context-based vision system for place and object recognition. In *ICCV*, pages 273 – 280, 2003.
- [17] P. Viola and M. J. Jones. Robust-real time face detection. *IJCV*, 57(2):137–154, 2004.
- [18] M. Weber, M. Welling, and P. Perona. Towards automatic discovery of object categories. In *CVPR*, pages 101–108, 2000.