

# Efficient Object-Class Recognition by Boosting Contextual Information

Jaume Amores<sup>1,\*</sup>, Nicu Sebe<sup>2</sup>, and Petia Radeva<sup>1</sup>

<sup>1</sup> Computer Vision Center, Universitat Autònoma de Barcelona

<sup>2</sup> University of Amsterdam

**Abstract.** Object-class recognition is one of the most challenging fields of pattern recognition and computer vision. Currently, most authors represent an object as a collection of parts and their mutual spatial relations. Therefore, two types of information are extracted: local information describing each part, and contextual information describing the (spatial) context of the part, i.e. the spatial relations between the rest of the parts and the current one. We define a generalized correlogram descriptor and represent the object as a constellation of such generalized correlograms. Using this representation, both local and contextual information are gathered into the same feature space. We take advantage of this representation in the learning stage, by using a feature selection with boosting that learns both types of information simultaneously and very efficiently. Simultaneously learning both types of information proves to be a faster approach than dealing with them separately. Our method is compared with state-of-the-art object-class recognition systems by evaluating both the accuracy and the cost of the methods.

## 1 Introduction

In this work we deal with the problem of detecting the presence or absence of one object category in an image. In contrast to simple object recognition, object-class recognition is not restricted to images of the same physical object (e.g. different images of the same car), but deals with different instances of the object, e.g. images of different cars. This introduces a high variability of appearance across objects of our category. The difficulty is increased by the presence of clutter in the images, partial occlusion and accidental conditions in the imaging process. Among recent approaches, characterizing the object as a collection of parts and their spatial arrangement has proved to be a promising direction [1–4].

Classical contextual representations such as Attribute Relational Graph (ARG) [4] and constellation of parts [3, 5] 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, i.e. differences in spatial position.

In this work we define a constellation of generalized correlograms for object representation. Correlograms were used to measure the joint distribution of pixel-level color information along with the spatial distribution [6]. A generalized correlogram is introduced here to deal with higher level properties related to parts of an object. The image

---

\* Work supported by CICYT TIC2000-1635-C04-04, Spain.

is represented by constellations of such generalized correlograms, instead of using a unique descriptor as done originally [6]. Belongie et al. [7] introduced constellations of shape contexts (another type of correlogram) that only deal with spatial arrangements, i.e. do not consider local information. In our representation, every feature vector in our feature space gathers local and contextual information. A great advantage of this feature space is that it can be integrated with an efficient feature selection and learning algorithm such as AdaBoost [8, 9] with weak classifiers based on single dimensions. This leads to simultaneously learning those spatial relations and local properties of parts that are characteristic of the object category.

Summarizing, the main contribution of this work is in integrating a new constellation of generalized correlogram representation into AdaBoost with feature selection. AdaBoost used with weak classifiers based on single dimensions together with our object representation lead to an efficient object recognition scheme dealing with the spatial pattern of the object. We first explain the image representation in Section 2, followed by the description of the spatial pattern classifier with boosting in Section 3. In Section 4 we report results and conclude in Section 5.

## 2 Image Representation

In this section we introduce a new representation of the object by using a constellation of generalized correlograms. Let an image  $I_k$  be represented by a constellation of  $U_k$  object parts, expressed as  $H_k = \{\langle o_i, \mathbf{h}_i, \mathbf{x}_i \rangle\}_{i=1}^{U_k}$ . The  $i$ -th detected part is represented by the tuple  $\langle o_i, \mathbf{h}_i, \mathbf{x}_i \rangle$ , where  $o_i$  is the label identifying the part,  $\mathbf{h}_i$  are the properties describing the part, and  $\mathbf{x}_i$  is its spatial position in the image. Due to clutter, parts in  $H_k$  might correspond to different objects. Let  $X_k = \{\mathbf{x}_i\}_{i=1}^{U_k}$  be the set of spatial positions of parts from  $H_k$ . One way to obtain potential parts of an image is by extraction of interesting points, also called features or key points [3, 5], this is also our approach.

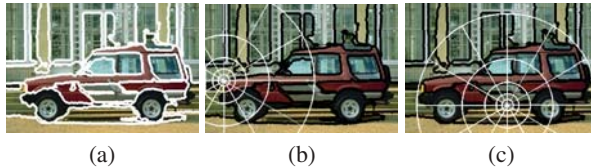
For our purpose, it is important to not miss any informative location, and to perform a fast interesting point extraction. By interesting point we mean any point located at an informative position, such as the edges, we do not mean necessarily corners. Two levels of interesting points are extracted. First we obtain a dense set of interesting points representing potential parts of objects. From this dense set, we extract local information around each point. Let  $H_k^L = \{\langle o_i^L, \mathbf{h}_i^L, \mathbf{x}_i^L \rangle\}_{i=1}^{U_k^L}, \mathbf{x}_i^L$  denote this dense set (do not confuse with the final representation  $H_k$ ). We extract local information as properties  $\mathbf{h}_i^L$  of these parts. Let  $X_k^L = \{\mathbf{x}_i^L\}_{i=1}^{U_k^L}$  be the dense set of positions from  $H_k^L$ . In our implementation, these positions are located at extracted contours of the image, see fig. 1(a). From  $X_k^L$  we sample a much more sparse set of interesting 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  $\mathbf{x}_j \in X_k$  is the position of  $o_j$ . We associate as descriptor  $\mathbf{h}_j$  a correlogram that measures the joint distribution of spatial relations  $(\mathbf{x}_i^L - \mathbf{x}_j)$  and local properties  $\{\mathbf{h}_i^L\}_{i=1}^{U_k^L}$ . Let us express the spatial relation  $(\mathbf{x}_i^L - \mathbf{x}_j)$  in polar coordinates:  $(\alpha_{ij}, r_{ij})$ , and the  $d$  local properties as  $\mathbf{h}_i^L = (l_{i1}, l_{i2}, \dots, l_{id})$ . The joint distribution is measured by a histogram based on a partition of the  $d + 2$  dimensional space with vectors  $\mathbf{v}_{ij} =$

$(\alpha_{ij}, r_{ij}, l_{i1}, l_{i2}, \dots, l_{id}), i = 1, \dots, U_k^L$ . 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 :  $\mathbf{h}_j(w) = \frac{1}{U_k^L} |\{v_{ij} \in B_w, i = 1, \dots, U_k^L\}|$ , i.e. the  $w$ -th bin of  $\mathbf{h}_j$  counts the number of vectors  $v_{ij}$  falling into this bin.

Note that this space contains vectors that express spatial relations and local properties, and thus the resulting descriptor  $\mathbf{h}_j$  is a correlogram of local properties in  $H_k^L$  considering their spatial distribution around the point of reference  $x_j$ . As  $X_k \subset X_k^L$ , we are describing in the same vector  $\mathbf{h}_j$  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 interesting 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 (see 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 [7] (see fig. 1(b)-(d)). This makes the descriptor  $\mathbf{h}_j$  focus more on local properties around  $o_j$  (local context) than to the 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.



**Fig. 1.** (a) Dense cloud of points covering interesting parts of the image (edges). (b)-(c) Log-polar spatial quantization of our correlogram. Each descriptor in (b) and (c) represents a different “point of view” of the object’s spatial arrangement

As local information, local structure and color around a small neighborhood are utilized. 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 [6, 7]. 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. [7] 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 as  $A_k = \{H_k^s\}_{s=1}^{n_s}$ , 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

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 [8, 9]. By learning the relevant dimensions of vectors  $\mathbf{h}$  defined in section 2, we are simultaneously learning the properties characterizing every part of the object and their mutual spatial relations.

In our framework, the model of one object is expressed as  $\Omega = \{\langle \omega_i, \varphi_i \rangle\}_{i=1}^M$ , where  $\omega_i$  is the label of one model part,  $\varphi_i$  are the parameters for this model part learnt by the classifier, 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  $\Omega = \{\langle \omega_i, \varphi_i \rangle\}_{i=1}^M$ . 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  $M_s(A_k) = H_k^m = \arg \max_{H_k^s \in A_k} L^\Omega(H_k^s)$ .

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 [7] over these manually segmented images, which obtains an initial training set  $T'_i$  for each  $\omega_i$ .  $T'_i$  contains as positive instances only those from the 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  $M_s(A_k)$  and then we apply the part matching in the appropriate scale  $M_i^\omega(M_s(A_k))$  (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  $\Omega$ .

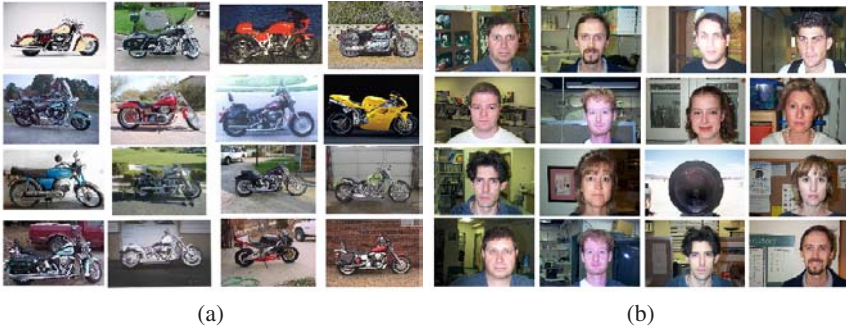
## 4 Results

We used the CALTECH database (<http://www.robots.ox.ac.uk/vgg/data3.html>) collected by Fergus et al. [3, 10], which consists of 7313 images with clutter for object recognition. Part of this database was also used by other authors such as Agarwal et al. in [1]. This database contains 7 different categories, which is a big step forward compared to many databases used in other works that are based on one or two categories. A full description of the database along with examples can be found in [3, 10], we do not show them here due to lack of space. The object categories can be found in table 1(a). Most of the object categories have instances under the same bi-dimensional arrangement, except for the spotted-cat category taken from Corel<sup>®</sup> database. Each category has roughly 800 images of different objects of this category. From the positive training set, 10 images are manually segmented. The negative set of images were taken by Fergus et al. from Google<sup>®</sup> by searching with the keyword “things”. This consists of 520 images, 400 were taken as training and 120 as test. Each time the training consisted of 400 positive images, 400 negative, and the test of 100 positive and 100 negative.

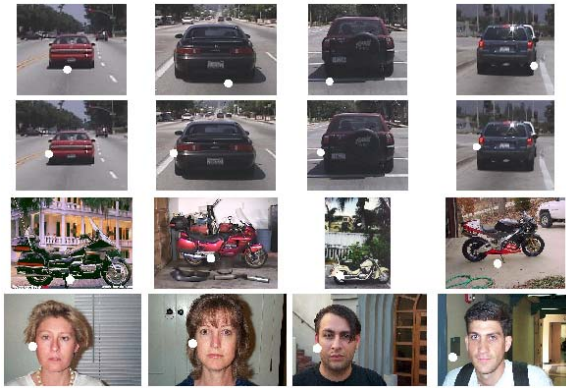
A cross-validation procedure was followed to test a total of 400 negative images and 400 positive images, average results are shown. Each time, the images included in the sets were picked randomly, always using disjoint sets for training and test.

Fig. 2(a)-(b) shows a example of results for 2 of the 7 categories, motorbikes in (a) and faces in (b). Fig. 2(a) shows every image correctly classified as motorbike. Some images show a heavy clutter and still there are no incorrect matches. Fig. 2(b) shows images classified as faces. Faces show an incorrect match, that can be seen to be similar in shape.

In fig. 3 each row shows the matching from a part  $\omega_i$  of the learnt model, to a matched part  $o_j$  in different instances of the object. In the first row we show matching of one model part of the car(rear) category. We can see that the model part is consistently



**Fig. 2.** (a) Example of images correctly classified as motorbikes, (b) images classified as faces



**Fig. 3.** Some matchings obtained with in several classes

matched with the same shadow beneath the car in the images. In the second row, another model part is matched consistently near the left red light in the images. In the motorbike category (third row), one model part matches with parts in a similar relative position of the instance motorbikes, despite the clutter. Finally, a model part of the face category (forth row) matches with parts near the ear of the face instances.

Given a test set with positive and negative images, the goal is to detect what images contain some instance of the object category and what do not contain any instance. 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(a), presents results comparing our method against the constellation used by Fergus et al. in [3], they also report results with other approaches using the same data set (see reference). 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 that we use not so suitable. However, the inclusion of local properties such as color makes boosting focus more on local information than contextual information in this category, so that not bad results are obtained. The face category is probably better

**Table 1.** (a) ROC equal error rates measures with the method in [3], and our method (b) Computational cost for different stages, see text for explanation, the second row is the inverted file arrangement cost, only in our system

(a)			(b)		
Category	[3]	Ours Context	Step	[3]	Ours
Car(Rear)	90.3%	96.9%	Description per image	15 sec.	6 sec.
Plane	90.2%	94.5%	I.F. per image	-	2 sec.
Leaf	-	96.3%	Training	36 hours	4 hours
Motorbike	92.5%	95.0%	Classification per image	3 sec.	0.23 sec.
Face	96.4%	89.5%			
Spotted Cat	90.0%	86.5%			
Car (Side)	88.5 %	90.0%			

represented using local appearance and PCA as Fergus does, we can include this in a future work. For the car (side) category the result is a recall-precision equal error. The negative set in this category contain images of roads without cars [3], so that a more realistic experiment can be made.

To speed up the algorithm, we take the non-empty bins of correlograms describing the current object category, and only use these bins in AdaBoost. That is, bins that are empty in our positive training set  $T_i$  (see section 3) are not used by the classifier. This also makes the algorithm more robust against clutter because we disregard structures not found in our object category. A similar idea was used in [11] for shape contexts. We also make use of the high sparseness of our generalized correlograms both in the training and in the recognition stage. Note that a correlogram is a special type of histogram. We only process the non-zero elements of our descriptors, by structuring the data as inverted files, a technique used in information retrieval [12]. For each dimension we keep the index of the descriptors that contain a non-zero value for this dimension, along with the value for this descriptor. Then the descriptors are sorted by the value of this dimension. This allows to use binary search in the recognition stage when we are looking for values in one dimension exceeding the threshold obtained by AdaBoost [9]. As our images contain a large constellation of descriptors with different scales (typically 100 descriptors with 7 scales) this technique speed ups the algorithm by obtaining a logarithmic cost in the number of scanned descriptors. Although sorting has a cost a bit higher than linear, it is done only once, saving later a lot of cost in the search for each model part  $\omega_i$ . Furthermore, sorting has linear cost on the number of descriptors that have a value greater than zero, only 20% of the descriptors due to the sparseness. This technique is also suitable for retrieving in large databases if we have pre-computed the descriptors. The time cost for every stage is shown in table 1(b), compared to the method in [3]. The second row denotes the cost of arranging the description of  $I_k$  as inverted file and sorting. We used a computer at 2.4 GHz for the experiments, while experiments in [3] were made with a computer at 2.0 GHz. We used Matlab<sup>®</sup> with some subroutines in C. Some parts such as the feature extraction and training stage could be made more efficient.

## 5 Discussion

In this work an object class recognition system has been proposed that is able to learn the characteristic parts of the object and their spatial relationship in the presence of clutter. The image is represented as a constellation of very sparse contextual descriptors and this representation is integrated with an efficient feature selection and learning algorithm such as boosting. We achieved very accurate classifier compared to the approach of Fergus et al. [3]. Furthermore, making use of the sparseness we showed that an efficient method can be achieved, suitable for scanning large databases. Summarizing, 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). By boosting we can combine a descriptor sensitive to different shapes and a (contextual) descriptor robust against shape variations, and let the classifier learn if the object is very structured.

## References

1. Agarwal, S., Awan, A., Roth, D.: Learning to detect objects in images via a sparse, part-based representation. *IEEE TPAMI* **26** (2004) 1475–1490
2. Schneiderman, H.: Learning a restricted bayesian network for object detection. In: *IEEE Proc. CVPR*. (2004) 639–646
3. R., F., P., P., A., Z.: Object class recognition by unsupervised scale-invariant learning. In: *IEEE Proc. CVPR*. (2003)
4. Hong, P., Huang, T.S.: Spatial pattern discovery by learning a probabilistic parametric model from multiple attributed relational graphs. *Journal of Discrete Applied Mathematics* **139** (2003) 113–135
5. Weber, M., Welling, M., Perona, P.: Towards automatic discovery of object categories. In: *IEEE Proc. CVPR*. (2000) 101–108
6. Huang, J., Kumar, S., Mitra, M., Zhu, W., Zabih, R.: Image indexing using color correlograms. In: *IEEE Proc. CVPR*. (1997) 762–768
7. Belongie, S., Malik, J., J.Puzicha: Shape matching and object recognition using shape contexts. *IEEE Trans. PAMI* **24** (2002) 509–522
8. Schapire, R.E., Singer, Y.: Improved boosting using confidence-rated predictions. *Machine Learning* **37** (1999) 297–336
9. Viola, P., Jones, M.J.: Robust-real time face detection. *Int'l J. of Computer Vision* **57** (2004) 137–154
10. Fei-Fei, L., Fergus, R., Perona, P.: A bayesian approach to unsupervised one-shot learning of object categories. In: *IEEE Proc. ICCV*. Volume 2. (2003) 1134–1142
11. Thayananthan, A., Stenger, B., Torr, P., Cipolla, R.: Shape context and chamfer matching in cluttered scenes. In: *IEEE Proc. CVPR*. (2003)
12. Squire, M.D., Muller, H., Muller, W.: Improving response time by search pruning in a content-based image retrieval system, using inverted file techniques. In: *IEEE Workshop CBAIVL*. (1999)