Boosting Contextual Information in Content-Based Image Retrieval

Jaume Amores[1], Nicu Sebe[2], Petia Radeva[1], Theo Gevers[2], Arnold Smeulders[2]
[1] Computer Vision Center, UAB, Spain
{jaume, petia}@cvc.uab.es
[2] Univ. of Amsterdam, The Netherlands
{nicu, gevers, smeulders}@science.uva.nl

ABSTRACT

We present a new framework for characterizing and retrieving objects in cluttered scenes. This CBIR system is based on a new representation describing every object taking into account the local properties of its parts and their mutual spatial relations, without relying on accurate segmentation. For this purpose, a new multi-dimensional histogram is used that measures the joint distribution of local properties and relative spatial positions. Instead of using a single descriptor for all the image, we represent the image by a set of histograms covering the object from different perspectives. We integrate this representation in a whole framework which has two stages. The first one is to allow an efficient retrieval based on the geometric properties (shape) of objects in images with clutter. This is achieved by i) using a contextual descriptor that incorporates the distribution of local structures, and ii) taking a proper distance that disregards the clutter of the images. At a second stage, we introduce a more discriminative descriptor that characterizes the parts of the objects by their color and their local structure. By using relevant-feedback and boosting as a feature selection algorithm, the system is able to learn simultaneously the information that characterize each part of the object along with their mutual spatial relations. Results are reported on two known databases and are quantitatively compared to other successful approaches.

Categories and Subject Descriptors: H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval

General Terms: Algorithms.

Keywords: Content-Based Image Retrieval, Object Recognition, Contextual Information, Boosting

1. INTRODUCTION

Given the large amount of information available in the form of digital images, it becomes critical to develop systems that automatically organize and retrieve images based on their content. In this work, we present an object-based retrieval approach. An important difference with a general object class recognition approach is that achieving fast time responses is a major goal when the scope is to perform retrieval (i.e. interact with the user).

We regard an object as a collection of parts and their mutual spatial relations. In this sense, the representation of the image must take into account local information characterizing the parts, and contextual information characterizing what is the context of each part (how the rest of the parts are spatially related to it). In retrieval of objects, many authors rely on a segmentation of the image into blobs [3, 4, 18]. Local information is used by extracting a set of descriptors from each blob, which very often with the current segmentation techniques does not represent the whole object. A classical contextual representation is the "Attributed Relational Graph" (ARG) [9, 12]. This descriptor represents the parts as nodes and their spatial relationships as arcs. If we obtain the parts by segmenting the image into blobs, this descriptor is not appropriate for complex images. The reason is that with the state-of-the art segmentation, the number of blobs and their spatial distribution are not constant across different images of the same object.

Instead of using a discrete descriptor (e.g. ARG) many authors take into account local properties along with their spatial relations through multi-dimensional histograms [11, 13, 19, 7]. These authors use generalizations of the color histogram in order to take into account the color of the pixels and their spatial relations. The difference is that they use different spatial relations, restricting or not the relations to be between pixels of the same color, and using all the pixels or just pixels forming edges. The common feature of their approaches is that they use a single histogram for the whole image. The context around every part of the image is then mixed up and blurred by aggregating the relationships into one final spatial histogram. This does not allow to represent the different points of view of the object, i.e. how the context is represented around different parts. Moreover, by this approach the background is aggregated into the descriptor making it not robust in cluttered scenes. Furthermore, using color as the only local information restricts the search to objects that have the same color as the query. This prevents general object class recognition because, for example, black cars cannot be matched with red cars.

Belongie et al. [1] consider the use of several spatial histograms for describing the object. They developed a de-

MIR'04, October 15–16, 2004, New York, New York, USA.

scriptor called shape context for describing objects by their shape. This descriptor represents the spatial relations between points of the contour by using the distance and the angle between them. The authors do not consider any local information and their main scope is to describe binary shapes in images without clutter. Shape contexts from two images are matched by a computationally expensive registration algorithm, and the final distance is computed as the sum of the distances between matched descriptors. Again, the success of registration is very dependent on having images well segmented and without clutter.

In this work, we design a feature space that takes into account local and contextual information representing the parts of the object and their spatial relations. For this purpose, we consider several multi-dimensional histograms representing the joint distribution of local properties and spatial relationships. Each histogram represents the spatial distribution relative to a different origin or "point of view" of the object. Instead of aggregating the different descriptors or views, the image is represented by the whole set of views, which makes the representation robust against clutter and more discriminative.

There are two major contribution of this paper. First, we retrieve objects based on their shape in the presence of clutter. For this purpose, we use three components: i) a multi-dimensional histogram using the log-polar spatial quantization of Belongie [1] and local structure as local information; ii) a coarsely hand-segmented query image, and iii) a proper distance that avoids taking into account the clutter of the images. The tests show that a significant improvement is achieved by using our descriptors instead of the shape context for performing shape-based retrieval in cluttered images. Tests were performed also on images free of clutter, where the shape context alone performs very well, and still the multi-dimensional histogram outperforms the shape context.

Second, after a first retrieval, we learn what are the characteristic properties of each part of the object along with their spatial relationship. Now we use as local information not only the local structure but also the color of the regions. By building the descriptors properly and using boosting as feature selection, the subsequent retrieval is not restricted to objects with the same color but still can use the color information from parts with characteristic color. For example, if a car is presented, the system will learn that the bottom parts (tires and shadow beneath the car) have a characteristic black color, but the rest of the parts only have a characteristic local structure. If an elephant is presented, the system then learns that this animal has a characteristic color in every part. For testing the performance, we used the same database as the one used by Fergus et al. [5] for structured objects with clutter. The results are comparable or better to those of Fergus, while the computational time is much lower in the learning stage.

The rest of the paper is organized as follows: in section 2 we present the feature space used in our system, in section 3 we explain the similarity comparison when using retrieval and in section 4 we show in detail how learning is performed. Section 5 reports our results and we present our conclusions and future directions in section 6.

2. FEATURE SPACE: TAKING INTO ACCOUNT THE RELATIVE SPATIAL DISTRIBUTION OF LOCAL PROPERTIES

We present in this section a multi-dimensional histogram that takes into account local properties and their spatial relationships, and is suitable for describing complex objects in the presence of highly cluttered scenes. The spatial distribution is taken relative to several points of the image in order to obtain invariance against translations. Each of these points has then associated one contextual descriptor with a particular "view" of the spatial distribution of the parts, and the whole image is described by a set of different contextual descriptors. In the following, a general framework for the contextual descriptor is provided and then we explain in detail the particular implementation we use here.

Let $P = \{p_i\}_{i=1}^{N}$ be a dense set of points at interesting parts (edges) of the image (see Fig. 1(a)). Let $l_i = (l_{i1}, l_{i2}, \ldots, l_{id})$ be a local feature vector measuring the local properties around the point p_i. Let $X = \{x_j\}_{j=1}^{M}$ be a more sparse set of points ($M << N$) covering the different locations from which we measure the relative spatial distribution of local properties (see Fig. 1(b)-(d)). For each point x_j we extract a histogram of the joint distribution of relative spatial positions $(p_i - x_j)_{i=1}^{N}$ and local properties $\{l_i\}_{i=1}^{N}$. Let us express the spatial vector $(p_i - x_j)$ in polar coordinates: (α_{ij}, r_{ij}) and let the contextual descriptor associated with x_j be h_j. For representing this histogram we make a partition of the $d+2$ dimensional space and count the proportion of points that fall into each bin. As we do not perform linear quantization, the bins of our partition can be better expressed by the combination of uni-dimensional bins. Let $b_s(r)$ be the r bin in the s dimension, let this dimension be partitioned into n_s bins, and let $B(k_1, k_2, \ldots, k_{d+2})$ be one bin in the $d+2$ dimensional space, where $k_s \in \{1, \ldots, n_s\}$. Let $v_{ij} = (\alpha_{ij}, r_{ij}, l_{i1}, l_{i2}, \ldots, l_{id})$ be one vector in the $d+2$ dimensional space, and let v_{ij}^e be the e element of that vector. The histogram h_j is then a $n_1 \times n_2 \times \ldots \times n_{d+2}$ vector that can be expressed as:

$$h_j(k_1, k_2, \ldots, k_{d+2}) = \frac{1}{N} |\{v_{ij} \in B(k_1, k_2, \ldots, k_{d+2}), i = 1, \ldots, N\}|$$
$$B(k_1, k_2, \ldots, k_{d+2}) = \{v_{ij} \in \mathbb{R}^{d+2} : v_{ij}^e \in b_e(k_e) \forall e = 1 \ldots d+2\}$$
$$k_s \in 1, \ldots, n_s \quad \forall s = 1, \ldots, d+2.$$

For the spatial coordinates α_{ij}, r_{ij} we use the same log-polar quantization as Belongie et al. [1] (see Fig. 1(b)-(d)). The bins $b_1(k), \forall k = 1, \ldots, n_1$ result from a linear quantization of the interval $\alpha \in [0, 2\pi)$ into n_1 bins, and the bins $b_2(k), \forall k = 1, \ldots, n_2$ increase exponentially in size from $k = 1$ to $k = n_2$, $n_1 = 12, n_2 = 5$ (we refer to [1]). This makes the contextual descriptor more sensitive to the local context than to the far context. The rest of the dimensions regarding the local properties $l_{i1}, l_{i2}, \ldots, l_{id}$ are linearly quantized; we explain below each of them in turn.

As local information we use the local structure and color around a small neighborhood. The local structure allows the descriptor to distinguish between objects that have global similar spatial arrangements but parts with different structures (see Fig. 2). It also makes our descriptor focus on the structures of the query object, disregarding those from the background and those resulting from false detected contours. The edges themselves constitute a type of descriptor for local structure and are computed by first order derivatives of the intensity map. If we take higher order derivatives we add

(a) (b) (c) (d)

Figure 1: (a) Dense cloud of points covering interesting parts of the image (edges). (b)-(d) Three contextual descriptors using log-polar spatial quantization. Each descriptor represents a different "point of view" of the object's spatial arrangement.

information about the local geometry. We can measure the direction of the edges or the change of direction (curvature). In this work, we use a simple yet effective local structure descriptor extracting the angle of the edges as measured along the curve formed by these edges. For this purpose the edges must be contiguous to each other in order to form a curve. This is obtained by performing region segmentation and regarding the boundaries of the regions as the edges of our image. Note that although finally we are segmenting the image, the blobs do not play any other role than providing contiguous edges. We apply the segmentation algorithm of Chen et al. [4] clustering by k-means the pixels based on their local texture and color and then apply a postprocessing similar to the one by Carson et al. [3], which obtains contiguous blobs and more accurate contours. The whole process takes less than two seconds. The parameters of the segmentation algorithm are set to perform over-segmentation, as we are interested in recovering all the contours of the image. As the segmentation uses texture features, it also permits to avoid high concentrations of edges in textured regions, which is another advantage over gradient based edge detectors.

To compute the angle of the edges at a particular position, we describe the contour in parametric form, using the arc-length as intrinsic parameter. We then compute the $\dot{x}$ and $\dot{y}$ components of the tangent at every point by making a convolution with the derivative of the Gaussian with scale $\sigma = 8$. The angle is taken modulus π, and we make a quantization into 4 bins, so that $n_3 = 4$.

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. Let $n_c = 12$ be the resulting number of color bins. In the previous framework, the color information associated to the point $p_i \in P$ is represented by assigning p_i to one color bin. This would be the case if there were always a single dominant color in the local area around p_i. As in practice this is not the case, we have to perform a fuzzy assignment of point p_i to several color bins. Hence, p_i does not belong completely to only one color bin, but it belongs in some proportion to several color bins, the sum of these proportions adding up to 1. In the present work we use the local color histogram $h_i^c : \{1, 2, \ldots, n_c\} \rightarrow [0, 1]$ as the *color membership function* for p_i, so that the membership of p_i to color cluster $k \in \{1, 2, \ldots, n_c\}$ is $h_i^c(k)$. For making the descriptor more sparse and saving storage requirements, we actually take the 5 most frequent colors of h_i^c, set to 0 the rest of of the entries, and normalize so that the new color histogram adds up to 1.

Let $h_j^{tc}, j = 1, \ldots, M$ be a contextual descriptor using as local information both local structure and color. Let $h_j^t, j = 1, \ldots, M$ be a contextual descriptor using as local information only the local structure. h_j^t is the projection of h_j^{tc} onto the color dimension, removing its influence. h_j^t describes the object only by its (contextual) geometric properties, whereas h_j^{tc} describes the object by both the geometric and color properties. By h_j^{tc} we can distinguish a black car from a black horse, but a red car will be regarded as different from a black one.

In the query stage where there is no learning, the user presents an image and the system computes the similarity to the query using the descriptors $h_j^t, j = 1, \ldots, M$, thus not restricting the matching to be between objects of the same color.

Let h_j^f be the concatenation of both descriptors: $h_j^f = (h_j^t(1), \ldots, h_j^t(n^t), h_j^{tc}(1), \ldots, h_j^{tc}(n^{tc}))$, where $n^t = n_1 n_2 n_3$ is the dimension of h_j^t, and $n^{tc} = n_1 n_2 n_3 n_4$ is the dimension of h_j^{tc}. This descriptor will be used in the learning stage. In this stage, if one part has a constant color across the training set, its color is learnt along with its local structure and relative position. For the rest of the parts the classifier learns only their local geometry and relative position, disregarding their color.

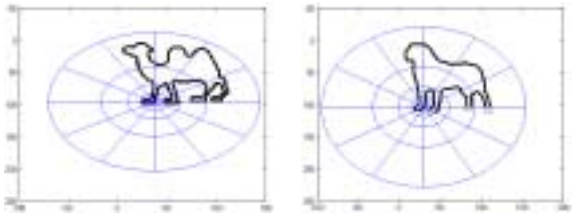


Figure 2: Two objects with very similar spatial configurations but different context of local structures. Extracted from Kimia's database

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 scale of our objects, we must compute the contextual descriptors for different scales fixed a priori. Let n_{scales} be the number of scales (we use currently $n_{scales} = 7$). The scales were computed from different instances of a small set of objects and used for the rest of the experiments without being changed. The final set of descriptors stored for each image is $\{\{h_{jk}^f\}_{j=1}^M\}_{k=1}^{n_{scales}}$. Note that the descriptors h^t used in retrieval are extracted from h^f simply by considering only the first n^t dimensions.

3. PLAIN RETRIEVAL: QUERYING BY EXAMPLE

At the beginning, the user has to provide some example of what he regards as relevant images in his search in the image database. The most common approaches are to query by single image example, without no additional user input, query by a group of images, or to allow the user to provide a drawn sketch of the type of image [16]. In the last years it has become clear that a retrieval system can not perform accurately if it does not provide tools for the user to interact with the system and state clearly in that manner what is he asking for [16, 8, 14]. Initially, we suppose that the user has a very small amount of examples, in the limit just one. We work with the supposition that the user only has one image which contains several objects. We allow him to segment coarsely the object he is interested in by drawing a polygon on its boundaries. The user presents this hand-segmented image and queries for similar images in the database. Using then relevant-feedback, he selects the relevant and non-relevant retrieved images which will constitute the training set for learning the properties of the object. In this section we explain how query by example works in our system leaving for the next section the learning approach.

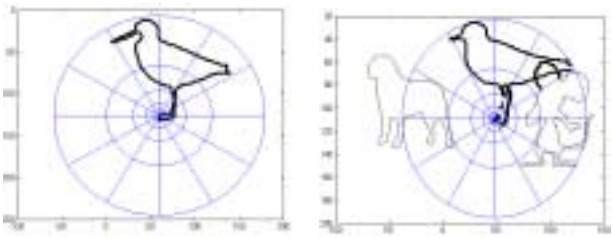
As explained in the previous section, we use only $\{h^t_{jk}\}$ in query by example. Let us call as v the descriptors h^t for the query image and as ω the descriptors for any target I of our database. First, the user presents an image and draws a polygon onto the boundaries of the interesting object. The boundary points of that hand-based segmentation constitute the set of points P for the query. The set of points X are sampled from P so that they cover the whole object. The implementation used is that of Malik's sampling: we start with $X = P$ and remove in each iteration the point whose geometric distance is the minimum with respect to the remaining points, until M points are left. The set of descriptors $\{v_j\}_{j=1}^{M_Q}$ are then extracted for the query. As we know exactly the size of the query object, the descriptors are scaled accordingly.

For each image I we read the pre-computed set $\{\{\omega_{jk}\}_{j=1}^{M_I}\}_{k=1}^{n_{scales}}$ for all the scales n_{scales}. Having the distance between any pair of vectors $d_{i,jk} = d(v_i, \omega_{jk})$, the distance between the query Q and the target I is taken as the minimum of the Chamfer distance over each scale:

$$d(Q, I) = \min_{k \in \{1,\ldots,n_{scales}\}} \sum_{i=1}^{M_Q} \min_{j \in \{1,\ldots,M_I\}} d_{i,jk}$$

This is based on the unidirectional Chamfer distance from the query to the target. The other direction (target to the query) includes the distance from points over all the target, including the background, which we want to avoid.

We choose the χ^2 as distance $d_{i,jk} = d(v_i, \omega_{jk})$. This is a typical distance for histograms that incorporates a normalization factor. We compute this distance considering only the non-empty dimensions of the query vector v. This avoids taking into account non-relevant bins that receive points from the background of the target. In Fig. 3 we show this idea, on the left being a query object (the bird) and on the right a target image with the object and clutter (e.g. the other two objects: the dog and the mouse). Thick points on the right represent points that fall in non-empty bins of the query, as we can see most of the background objects in the right are avoided by restricting the descriptor to non-empty bins of the left. For the sake of clarity in this image we are only representing non-empty bins in the *spatial* dimensions. The robustness against clutter is farther increased restricting the non-empty bins in the whole joint distribution (i.e. restricting that the local structure in some spatial bin be also the same as in the query). In this case, the points belonging to the ear of the mouse would not be taken into account, as their local structure is different in that relative spatial position than the local structure of the points of the query.



(a) (b)

Figure 3: Avoid non-object structures by considering only the non-empty bins. Query object (a), target image from which only thick points are taken into account (b)

4. BOOSTING: LEARNING THE CONTEXT

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 [6] and Support Vector Machines (SVM) [2]. 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. Furthermore, boosting can also be used as a feature selection algorithm. Given several weak classifiers whose error rate is slightly lower than 0.5, boosting provides a strong classifier by finding a suitable combination of the weak-classifiers (we refer to [6] for details). This combination provides a weight for each classifier according to its importance.

We use the AdaBoost version of boosting reported in [6] and implemented in [17] for retrieval. In this algorithm, every weak classifier is based on a single dimension of the feature space, and the final strong-classifier is based on the most discriminant dimensions of the space, weighted by their discriminant power. Our objective is to learn what are the characteristic properties of every single part of the object along with the characteristic spatial distribution of all the parts. This is achieved by learning the relevant dimensions of the vectors h^f defined in section 2. As explained, 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. 4).

We use as positive and negative images of our training set the relevant and non-relevant images provided by the user. For every image a set of descriptors is extracted, each of them representing the spatial distribution relative to a different origin x_j (constituting indeed a specific point of view of the object). Learning the spatial distribution of the object is done for every specific point of view x_j, and so we must first match the homologous x_j in the positive images. Then, we learn a model representing the object

(a) (b) (c) (d)

Figure 4: 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

for each different view j, let its parameters be Θ_j. Each model conveys a piece of evidence about the existence of the object in one image. First we explain the global classification process once we have these models, and then we explain how we build the training sets.

4.1 Classification of images: combining the classifiers

Let us recall that an image I is represented by descriptors h in different scales k. Let H^k be the set of descriptors h_k with scale k. We take as final representation H^k for the image I the most appropriate one according to our query (note that an image can have different objects with different scales, thus depending on the query we will take one scale or the other). Let us suppose for the moment that there is only one scale, and thus I is represented by one set of descriptors H. Every descriptor $h \in H$ represents the context of the image from a particular point of view. Let $l_j(h) \in [0,1]$ be the probability that the context descriptor h represents our context model Θ_j. Let $L_j^H(H) \in [0,1]$ be the probability that any of the descriptors in H represents Θ_j. For computing L_j^H we take as OR rule the maximum:

$$L_j^H(H) = \max_{h \in H} l_j(h). \quad (1)$$

Let $L_f^H \in [0,1]$ be the probability that H represents the object according to the evidence from all the models $\{\Theta_j\}_{j=1}^M$. As we want all the models to contribute to this classification score, we use as combination rule the sum of likelihoods, with equal weight for each model:

$$L_f^H(H) = \sum_{j=1}^{M} \frac{1}{M} L_j^H(H)$$

Note that L_f^H is a mixture of probabilities and can be regarded as the final probability for H given all the models. A more appropriate combination rule for this set of classifiers is provided by boosting, but we let this for a future work. Finally, the probability that any of the scaled representations H^k of I contains our object is computed again by using the maximum as OR rule:

$$L^I(I) = \max_k L_f^H(H^k)$$

This can be regarded as seeking the scale k with maximum global resemblance according to all the points of view Θ_j. As this will be useful later, we express this scale detection as:

$$S(I) = \arg\max_k L_f^H(H^k) = \arg\max_k \sum_{j=1}^{M} \frac{1}{M} L_j^H(H) \quad (2)$$

In the same way, the computation of $L_j^H(H)$ in Eq. 1 can be regarded as finding the descriptor h that matches Θ_j in the set H. This descriptor h is associated to one point of view x, which we express as $h(x)$. We say that Θ_j matches with the point of view x of I in scale $k = S(I)$ and express this as:

$$M_j(I,k) = x \in X : h(x) = \arg\max_{h \in H^k} l_j(h). \quad (3)$$

4.2 Building the training set

As explained, each model Θ_j represents the context of the object from a different point of view. For building it we must provide descriptors corresponding to homologous points, i.e. perform matching. We deal with that by using a small set of hand-segmented images, and performing registration with the algorithm of Belongie [1]. As we do not want the user to segment the whole training set, in the rest of the positive images we take as most probable matching point for each model the one which maximizes the likelihood according to a first boosting stage. The process can be decomposed in the following steps.

- Registration of a small set of hand-segmented positive images. Let N_p^1 be the number of hand-segmented positive images. The output of the registration is M sets, the j set $\{x_{ij}^+\}_{i=1}^{N_p^1}$ being points corresponding to the same point of view in different images.
- Compute descriptors associated with points x. Let H_j^+ be the set of descriptors associated with $\{x_{ij}^+\}_{i=1}^{N_p^1}$. We take only local structure as local information, leaving for later the whole context. As the image is segmented, the scale is just the size of the object. We take as negative set H^- all the descriptors from all the negative images with all the scales. These descriptors are pre-computed and we only have to read them. This negative set is the same for all the models j.
- With those descriptors, learn the classifier (by boosting) for each different point of view. The M trained classifier receives as parameters the set $\Theta_j, j = 1, \ldots, M$.
- Matching in the rest of positive images: for every learnt model Θ_j corresponding to a point of view of

the object, detect the matching point in every non hand-segmented image. Let $I_i^+, i = N_p^1 + 1 \dots N_p$ be a non hand-segmented positive image. The matching is performed in two stages:

1. detect the scale of the object in I_i^+. Let $s_i = S(I_i^+)$ be this scale, computed according to Eq. 2.
2. Detect the matching point for model j in image I_i^+: let $x_{ij}^+ = M_j(I_i^+, s_i)$ be this point, computed according to Eq. 3.

- We have matched the homologous points for all the views, and now we build the final training sets associated to each view. For that purpose we use the whole information h^f, corresponding to the joint distribution of local structure and spatial context, along with the joint distribution of local structure, local color and spatial context. Our new positive set consists of the descriptors h^f associated to the points $\{x_{ij}^+\}_{i=1}^{N_p}$. The scales are the size of the object for the first N_p^1 images, and the detected scales s_i for the rest of the images. The negative set again consists of all the descriptors with all the scales for all the negative images. The final model learnt with the training set for view j is written Θ_j^C.

5. RESULTS

Two groups of experiments are performed, the first testing the performance of the contextual descriptor in querying a database by example, and the second testing the accuracy of learning the context when classifying images of a database. We use two known databases: Kimia's data set [15] and Fergus' database [5]. The first one consists of 1069 binary images designed for testing object retrieval by shape/geometric characteristics. The second one consists of 3512 images and is used for general object recognition.

5.1 Querying by contextual descriptors

As explained in section 3, we query a database by presenting a coarsely hand-segmented image and using the contextual descriptors h^t that incorporate local structure as local information. We want to compare this to the performance achieved when using the shape context descriptor developed by Belongie et al. [1]. Shape context is well suited for shape retrieval, has the same spatial quantization as our descriptor and does not incorporate local information in the histogram. The comparison is performed in the Kimia's and Fergus' databases. The first one is divided into 64 categories, and we take those containing at least 3 different objects, thus we have only 43 categories to perform our tests. We use all the images of each category as query and measure the average precision-recall graph over each of them.

For each category we summarize its average precision-recall into one scalar by computing the area below the curve and normalizing among the maximum possible area, obtaining the accuracy measure used by Howe [10].

The results are presented in table 1. For 18 out of 43 categories, using the context descriptor h^t outperforms the shape context descriptor, the average difference being 10%, the maximum difference 17.5% and the minimum 4.5%. For 2 categories, the shape context performs better, the average difference being 9.81%. In these cases, the object had high variance in local structures among the different instances.

Category	Context Descriptor h^t	Shape Context
Misk	91.48%	80.90%
Arb	100.00%	88.35%
Bird	35.14%	23.12%
Bottle	87.37%	79.07%
Camel	98.18%	80.64%
Cat	52.42%	37.04%
Crown	94.43%	89.88%
Dino	35.34%	21.19%
Dog	78.77%	68.26%
Dude	98.71%	91.33%
Fgen	93.68%	84.08%
Flightbird	78.28%	65.55%
Fork	82.40%	74.04%
Hammer	14.08%	9.43%
Mgen	96.09%	85.79%
Ray	28.41%	19.54%
Tool	10.32%	4.92%
Turtle	21.30%	13.20%
Bone	79.82%	93.25%
Key	78.27 %	84.46%

Table 1: Results on Kimia's categories, except for the last 2, our contextual descriptor outperforms the shape context

These local variances among the instances can be detected by the learning stage, and thus the spatial bins that have this variance are avoided when classifying by boosting. For the rest of categories the performance is very similar, being the average score in those categories 74.22% for our context descriptor and 73.86% for the shape context.

We perform the same comparison using now 5 categories from the Fergus' database: car, plane, background (mixture of different background images), leaves, and motorbikes. In Fig. 5 we present three images per category from this database. The queries are performed on categories with objects: the car, the plane, the leaves, and the motorbikes. For comparing the performance with Fergus results we take in each case a subset of images consisting of images from the query's object category and images from the background category. When the images contain clutter, the plain shape context descriptor performs poorly. Instead, we extract here shape contexts from contours of isolated blobs, trying that each blob represents the biggest part of the object (i.e. not performing over-segmentation). The scale of the shape context is computed proportional to the size of the blob. These settings provide the best performance for the shape context in cluttered images. The contextual descriptor with local structure is computed based on contours from over-segmented images, as explained in section 2, and with different fixed scales (7 scales in this experiment). For each category, we perform 30 queries with both descriptors, Fig. 6 shows the average precision-recall graph for each of them: car category in Fig. 6(a), plane in Fig. 6(b), leave in Fig. 6(c), and motorbike in Fig. 6(d). The discontinuous line shows performance using the shape context and the continuous line using the presented contextual descriptor. The results are significantly better in all of the categories except for the leaves, which have a similar performance. The car category in this database has queries with very poor performance. As the car is seen from behind, the hand-based segmentation is just a rectangle (see Fig. 7(b)), a struc-

(a) (b) (c)

Figure 7: Car category in Fergus' database. Set of car images (a), their hand-based segmentation (b), and the real edges (c)

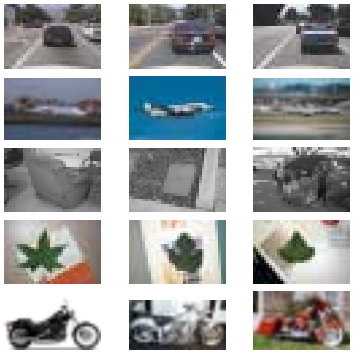
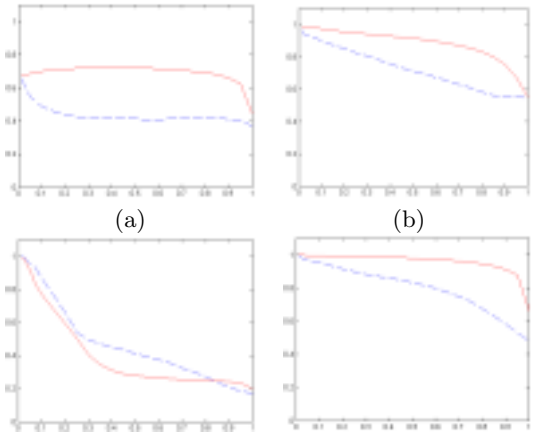


Figure 5: Fergus' database, three images per category are shown: from left to right, up to bottom: car, plane, background, leaves, motorbikes



(a) (b) (c) (d)

Figure 6: Precision-recall in Fergus database. The continuous line represents results obtained with our contextual descriptor and the discontinuous line with the shape context. See text.

ture that is repeated very often in other categories. This is greatly improved using the real contours of the images (see Fig. 7(c)), and the color information (see Fig. 7(a)). However, for doing so we must learn what are the good contours of the car and the characteristic color (such as road color and the shadows beneath the car). Boosting allows us to use all this information in an efficient manner.

5.2 Boosting retrieval

We compare our results with the one obtained by Fergus on the same database. In order 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 being extracted automatically. As the the background images are only provided in gray-level, we can only take gray-level as color information for performing a fair comparison with Fergus' results. For doing so, we use Matlab®rgb2gray function, which maps the R,G,B color space to L,u,v space and take the L band as gray-level. We quantize this band into 16 bins, and use this quantization for the color dimension of our contextual descriptor, as explained in section 2. The classification hit rate is measured using the receiver-operating characteristic (ROC) equal error rates: p(True Positive)=1-p(False positive). Table 2 presents results for the different methods. Fig. 8 shows a precision-recall curve in the car category when using h^f context (i.e. including local gray-level) and when using h^t context (i.e. only including local structure). The ROC equal error rate is 97.0% with the complete context and 92.4% with the other one. Without non-optimized code, the learning stage takes at most 3 hours, which is not much compared to the time spent by the E-M algorithm used by Fergus (36 hours). If we take advantage of the sparseness of our feature space, and not process the zeros of the vectors, the time spent by boosting can be much further reduced to the order of minutes. We also tested the performance when classifying one object against the rest of the objects (e.g. car against planes, leaves and motorbikes), using R,G,B color space in the contextual descriptor (see table 3). As we classify one category against all the rest, the number of images used is higher in this case. All the results are above 90%, and using R,G,B has an overall better performance than using gray-level (96.9% against 96.4%).

6. DISCUSSION

We have introduced an object-based retrieval 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. [5]. Our novel contributions are as follows.

- We proposed an efficient object-based approach that

Category	Fergus	Boosting Context	Shape Context	Geometric Context
Car	90.3%	97.0%	54.2%	79.0%
Plane	90.2%	92.7%	61.7%	80.2%
Leave	-	97.8%	65.5%	62.4%
Motorbike	92.5%	98.1%	73.9%	91.2%

Table 2: ROC equal error rates measures resulting from the method reported by Fergus, boosting the context, querying by example with shape context, and querying by a contextual descriptor with local geometric information

Category	boosting context
Car	97.33%
Plane	94.57%
Leaf	97.85%
Motorbike	97.82%

Table 3: ROC equal error rate measures resulting from classifying one object against the others using R,G,B color

uses boosting as a strong-classifier and feature selection, which makes the method suitable for retrieval.

- We performed a query by example methodology that is able to obtain shape-based retrieval in images with heavy clutter. This is done by a combining a proper distance with a discriminant contextual descriptor suitable for shape characterization.

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.

7. REFERENCES

[1] S. Belongie, J. Malik, and J.Puzicha. Shape matching and object recognition using shape contexts. Technical Report UCB//CSD-00-1128, UC, Berkeley, 2001.

[2] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[3] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik. Blobworld: A system for region-based image indexing and retrieval. In *Visual*, pages 509–516, 1999.

[4] Y. Chen and J. Wang. A region-based fuzzy feature matching approach to content-based image retrieval. *IEEE Trans. PAMI*, 24(9):1252–1267, 2002.

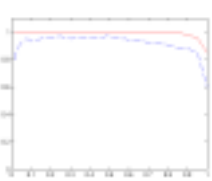


Figure 8: Results from boosting in the car category. The continuous line is the precision-recall curve resulting from the complete context, and the discontinuous line from using only the context of local structures

[5] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003.

[6] Y. Freund and R. E. Shapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.

[7] T. Gevers and A. Smeulders. Pictoseek: Combining color and shape invariant features for image retrieval. *IEEE Tran. IP*, 9(1):102–119, January 2000.

[8] J. D. H. D. Tagare, C. C. Jaffe. Medical image databases: A content-based retrieval approach. *Journal of the American Medical Informatics Association*, 4:184–198, 1997.

[9] P. Hong and T. S. Huang. Spatial pattern discovering by learning the isomorphic sub-graph from multiple attributed relation graphs. *Journal of Discrete Applied Mathematics*, 2003.

[10] N. Howe. A closer look at boosted image retrieval. In *CIVR*, pages 61–70, 2003.

[11] J. Huang, S. Kumar, M. Mitra, W. Zhu, and R. Zabih. Image indexing using color correlograms. In *Proc. CVPR.*, pages 762–768, 1997.

[12] E. G. M. Petrakis and C. Faloutsos. Similarity searching in medical image databases. *IEEE Trans. Knowledge Data Engineering*, 9(3), 1997.

[13] A. Rao, R. Srihari, and Z. Zhang. Geometric histogram: A distribution of geometric configurations of color subsets. In *SPIE: Internet Imaging*, volume 3964, pages 91–101, 2000.

[14] S. Santini, A. Gupta, and R. Jain. A user interface for emergent semantics in image databases. In *IFIP Working Conference on Database Semantics*, volume 8, 1999.

[15] K. Siddiki and B. Kimia. Parts of visual form: Computational aspects. *IEEE Trans. PAMI*, 17(3):239–251, March 1995.

[16] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. PAMI*, 22(12):1349–1380, December 2000.

[17] K. Tieu and P. Viola. Boosting image retrieval. *Int. J. Comput. Vision*, 56(1-2):17–36, 2004.

[18] J. Wang, J. Li, and G. Wiederhold. SIMPLIcity: Semantics-sensitive integrated matching for picture LIbraries. *IEEE Trans PAMI*, 23(9):947–963, 2001.

[19] R. Zhao and W. Grosky. Negotiating the semantic gap: From feature maps to semantic landscapes. *Pattern Recognition*, 35:593–600, 2002.