Kernels on bags for multi-object database retrieval

Philippe H. Gosselin
LIP6 CNRS UMR 7606
104 ave Kennedy
75006 Paris, France
philippe.gosselin@lip6.fr

Matthieu Cord
LIP6 CNRS UMR 7606
104 ave Kennedy
75006 Paris, France
matthieu.cord@lip6.fr

Sylvie Philipp-Foliguet
ETIS CNRS UMR 8051
6 ave du Ponceau
95014 Cergy, France
philipp@ensea.fr

ABSTRACT

In this paper, a kernel-based method for multi-object retrieval in large image database is presented.
First, our approach exploits a fuzzy region segmentation approach in order to get robust local feature extraction and characterization. All the region features are summarized in bags representing the image index. The main part of this work concerns the kernel functions to deal with sets of features. Based on the linear combination of minor kernels, a family of kernels on bags is introduced. Several weighting schemes and combinations are proposed. Their introduction are motivated in the specific context of dealing with multi-object recognition with heterogeneous background. Combined with SVMs classification and interactive online learning framework, the resulting algorithm satisfies the robustness requirements for representation and classification of objects. Experiments and comparisons demonstrate the good performances of our multi-object retrieval technique.

1. INTRODUCTION

Powerful strategies have been proposed for object categorization and retrieval in the last decade. Significant progress in object recognition including variation in poses, presence of clutter, occlusion and varying lighting conditions has been achieved. However, the problem remains very hard when considering multi object categorization and retrieval. Databases may be very large, with many objects embedded in large images with heterogeneous background.
In this context, object retrieval systems have to combine two main requirements:
• Effective data representation. Local-based image analysis are usually preferred to global ones in order to better grasp the relevant features characterizing the embedded objects. Many papers focus on these local feature detection, and very efficient techniques are now available, such as point of interest approaches (PoI) [9] or region-based techniques [1]. The main question is then: how to get an effective image signature from them ? Instead of computing a vector signature, working explicitly on a set of local features (a bag) constitutes an alternative way to represent an image that offers very interesting new possibilities for data comparison and classification. Kernel methods allow to develop new similarity between bags.
• Effective classification strategy. Combined with kernel-based data representations, SVMs are state-of-the-art large margin classifiers which have demonstrated remarkable performance in object recognition.

We propose in this paper to investigate the kernel framework to deal with multi-object retrieval in large databases. First, we propose to represent each image as a bag of fuzzy regions. Using fuzzy regions is a robust strategy to extract local information from images that match to objects or object parts. Next, local features computed on fuzzy regions are summarized in bags.

The main contribution of this paper is the introduction of specific classes of kernels on bags of features. We first motivate the approach of not building a feature map for bags, and then organize our kernel-based analysis around two points: the choice of the minor kernel, and the way to develop some kernels on bags that are Mercer kernels. The resulting algorithm satisfies the robustness requirements for representation and classification. Several experiments and comparisons have been carried out on a database of objects from the Columbia database with heterogeneous backgrounds. The RETIN active learning framework previously introduced in [3] is used to investigate the performance of our kernel-based learning algorithm for online object retrieval.

2. BAG-OF-FEATURES KERNEL FRAMEWORK

Before detailing the segmentation process in the next session, we develop here kernel framework that we are considering. Based on the region feature computation, each image is represented as a bag B_i of unordered vectors $\mathbf{b}_{ri}$. If heterogeneous backgrounds are considered, several feature vectors are relevant for object characterization, many are irrelevant or let's say noise. The next step is now to consider similarity functions between them. The major aim is to find the set of local descriptors that discriminates an object from other objects and from the background. In other words, we have to detect within the bag B_i which feature vectors are relevant.

An interesting and quite generic way to build functions

CIVR'07, July 9–11, 2007, Amsterdam, The Netherlands.

on bags is the one introduced in [11]:

$$K(B_i, B_j) \triangleq \sum_r \sum_s k(\mathbf{b}_{ri}, \mathbf{b}_{sj}) \quad (1)$$

When this function K on bags is based on a kernel function k on vectors (called *minor* kernel), it is easy to prove that K is also a kernel: let ϕ be the mapping into a Hilbert space $\mathcal{H}$ associated to the kernel function k:

$$k(\mathbf{b}_{ri}, \mathbf{b}_{sj}) = \langle \phi(\mathbf{b}_{ri}), \phi(\mathbf{b}_{sj}) \rangle$$

From this statement, we have:

$$\begin{aligned} K(B_i, B_j) &= \sum_r \sum_s k(\mathbf{b}_{ri}, \mathbf{b}_{sj}) \\ &= \sum_r \sum_s \langle \phi(\mathbf{b}_{ri}), \phi(\mathbf{b}_{sj}) \rangle \\ &= \langle \sum_r \phi(\mathbf{b}_{ri}), \sum_s \phi(\mathbf{b}_{sj}) \rangle \end{aligned}$$

It follows that the mapping function $\Phi(B_i)$ defined as:

$$\Phi(B_i) = \sum_r \phi(\mathbf{b}_{ri})$$

allows to write K as the corresponding dot product in the induced space:

$$K(B_i, B_j) = \langle \Phi(B_i), \Phi(B_j) \rangle$$

A problem when considering this class of kernels is the computational complexity with large bags. For PoI approaches, where 100 to 1000 vectors are usually required, several propositions have been made to reduce this complexity. A common strategy is to build a model of the bags, and next propose a kernel function for these models. For instance, in [6], Gaussian functions are used to represent bags, and a Bhattacharyya kernel to compare them. Another way is to represent a bag as a distribution of prototypes of points of interest [5, 4]. The main drawback is that the mapping is explicit in that case. To deal with the computational complexity, we have selected a different strategy, which is to work with regions : an image can be efficiently represented using around 10 times less regions than points of interest.

Furthermore, the definitive advantage is that the potential of the kernel framework is preserved. We focus on the design of the kernels. In this scope, we address the two major points of kernel design:

- The choice and the tuning of the minor kernel function;
- The combination of these minor kernels to build a bag function which satisfies Mercer's conditions.

In this framework, we present bags using fuzzy regions, existing propositions for kernels on bags in the literature, and in the next section our solutions to enhance them.

3. REGION-BASED FEATURE REPRESENTATION

In the same way as the human visual system perceives coarse zones with their approximate colours and sizes, we build regions which roughly correspond to the main colour parts of the image. Despite the fact that our visual system does not perform an accurate segmentation of the scene, the recognition of a landscape or a painting is instantaneous. Differently from the other systems which use regions [1], we use a segmentation into fuzzy regions. The main advantage is to be able to segment any image, even in difficult cases, when there is no clear limit between some parts of the objects. Fig. 1 shows an example of fuzzy segmentation. Also note that, compared to points of interest (PoI) [9, 7], the region-based representations are able to provide a very compact local representation : where hundreds to thousands of PoI are usually required, only a few dozen of regions are necessary for an image.

(a) (b) (c)

Figure 1: An example of fuzzy segmentation. (a) Original image. (b) Defuzzified regions (only used for display). (c) One of the fuzzy regions (the whiter, the higher the membership degree).

3.1 Fuzzy segmentation algorithm

Details of the algorithm of fuzzy segmentation are given in [10]. Extracted regions have the following properties : uniformity in terms of colour, contained expansion by high gradient norms, uncertainty where two (or more) regions encounter.

The algorithm first performs a watershed algorithm on the image of gradient norms, the uniform areas (of null colour gradient norm) constitute the cores of the fuzzy regions. The membership degrees of pixels to regions are then computed using the topographic distance to these cores, which is defined as the length of the shortest path connecting the pixel to the core, along the surface constituted by the gradient norm in the 3D space. The degrees slowly decrease according to the spatial distance to the core and strongly decrease when meeting an edge, zone of a large gradient norm. Impulse noise is bypassed, because a shorter path is found around it.

3.2 Region indexing

Each image i of the database is represented with a bag $B_i = \{\mathbf{b}_{ri}\}_r$ of regions $\mathbf{b}_{ri}$. Vectors $\mathbf{b}_{ri}$ are the concatenation of 4 histograms of 8 bins each, 1 for colors and 3 for textures. Histograms are obtained for each bin by adding the membership degrees of the pixels to the region. Thus pixels with small membership degrees belonging to transitions or outliers inside a region have little influence on the histogram shape.

4. PREVIOUS WORKS ON MATCHING KERNELS

The first approach for building kernels on bag is to find match-based similarity functions which satisfy Mercer's conditions. In this framework, a minor kernel function $k(\mathbf{b}_r, \mathbf{b}_s)$ is used as a matching function, i.e. a function that returns a high value if $\mathbf{b}_r$ and $\mathbf{b}_s$ are similar.

The simplest case is the function which returns the value of the best match between the vectors of two bags, thus the highest similarity between two feature vectors:

$$K(B_i, B_j) = \max_r \max_s k(\mathbf{b}_{ri}, \mathbf{b}_{sj}) \quad (2)$$

This function does not satisfy Mercer's conditions, altrough in pratice it achieves good performances.

In Multiple Instance Learning framework, it is known that a single match is not sufficient for a good discrimination [2]. The idea is then to build a similarity function that includes several matches, for instance the average value of the best matches for each region [14]:

$$K(B_i, B_j) = \frac{1}{|B_i|}\sum_r \max_s k(\mathbf{b}_{ri}, \mathbf{b}_{sj}) + \frac{1}{|B_j|}\sum_s \max_r k(\mathbf{b}_{ri}, \mathbf{b}_{sj}) \quad (3)$$

Again, this function is not a kernel function. An approximation as been proposed in [8] :

$$K(B_i, B_j) = \frac{1}{|B_i|}\frac{1}{|B_j|}\sum_r \sum_s k(\mathbf{b}_{ri}, \mathbf{b}_{sj})^q \quad (4)$$

However, when $q \to \infty$,this kernel is similar to function of Eq. 3, except that the minor kernel is raised to power q : $\frac{1}{|B_i|}\sum_r \max_s k(\mathbf{b}_{ri}, \mathbf{b}_{sj})^q + \frac{1}{|B_j|}\sum_s \max_r k(\mathbf{b}_{ri}, \mathbf{b}_{sj})^q$. With large values of q, this kernel tends to have no generalization capacity, just like a Gaussian with a very small σ.

5. PROPOSITIONS

In this section, we first propose two kernels on bag functions which aim at approximating the search of the best matches. Next, we propose to work on the minor kernel function, in order to take into account the dependencies of vectors in the same bag.

5.1 Kernel on Bag

The function of Eq. 2 returns the value of the best match between the feature vectors of two bags. This function is not a kernel function, but can be approximated using the following function, inspired from Minkowski distance, with high values of q:

$$K(B_i, B_j) = \left(\sum_r \sum_s k(\mathbf{b}_{ri}, \mathbf{b}_{sj})^q\right)^{\frac{1}{q}} \quad (5)$$

As Minkowski distance tends to be the L^∞ distance as $q \to \infty$, this kernel function tends to be the function of Eq. 2 as $q \to \infty$.

Let us note that another interest of this function is that it includes the parameter q, which allows a tuning between two extrema : the case $q = 1$, where the function returns the average value of the matches, and $q = +\infty$, where the function returns the highest value of the matches. Let us also note that, because of the power $\frac{1}{q}$, this kernel on bags is also usable on minor kernels that return positive values. But this is not really a problem in our context where histograms are generally used for the description of regions.

We also propose an approximation of the function of Eq. 3, which returns the average value of the best matches:

$$K(B_i, B_j) = \frac{1}{|B_i|}\sum_r \left(\sum_s k(\mathbf{b}_{ri}, \mathbf{b}_{sj})^q\right)^{\frac{1}{q}} + \frac{1}{|B_j|}\sum_s \left(\sum_r k(\mathbf{b}_{ri}, \mathbf{b}_{sj})^q\right)^{\frac{1}{q}} \quad (6)$$

We are not able to state if these functions satisfies or not the Mercer's conditions, altrough in pratice it achieves good performances.

5.2 Minor kernels

We propose to use a modified version of the Gaussian kernel with a χ^2 distance:

Gaussian χ^2: $k(\mathbf{b}_r, \mathbf{b}_s) = \exp\left(-\frac{1}{2\sigma^2}\sum_t \frac{(b_{tr}-b_{ts})^2}{b_{tr}+b_{ts}}\right)$

This minor kernel, that we call "Gaussian χ^1", is similar to the χ^2, except that we replace the squared difference with an absolute difference:

Gaussian χ^1: $k(\mathbf{b}_r, \mathbf{b}_s) = \exp\left(-\frac{1}{2\sigma^2}\sum_t \frac{|b_{tr}-b_{ts}|}{b_{tr}+b_{ts}}\right)$

The advantage of this kernel function is the independence to the bias : for any positive diagonal matrix Δ, $k(\Delta\mathbf{b}_r, \Delta\mathbf{b}_s) = k(\mathbf{b}_r, \mathbf{b}_s)$. Thus, no bias normalization process is required, which is very interesting for applications where the test data is not available.

5.3 Weighting feature vectors in bags

We propose in this section to include the fact that feature vectors within the same bag are not independent. To achieve this, we add a weight factor α_{ri} for each feature vector $\mathbf{b}_{ri}$:

$$\hat{\phi}(\mathbf{b}_{ri}) = \alpha_{ri}\phi(\mathbf{b}_{ri}) \quad (7)$$

Within this context, the first question is to find general rules for relevant weights. An important point is the noise generated by redundant feature vectors. In the same image, many similar regions can be present at the same time. The similarity of this image to another one is then highly based on these similar regions, and the matching of a singular one is lost within this context.

Hence, we propose to reduce these effects by increasing the importance of the singular feature vectors. The idea is to attribute a high value to a feature vector r which differs at most from the feature vectors in its own bag i:

$$\alpha_{ri} = \frac{1}{\sum_t k(\mathbf{b}_{ri}, \mathbf{b}_{ti})} \quad (8)$$

More generally, we can see this equation as :

$$\alpha_{ri} = \frac{1}{K(\{\mathbf{b}_{ri}\}, B_i)} \quad (9)$$

Where K is the basic kernel on bags (Eq. 1) applied on bag $\{\mathbf{b}_{ri}\}$ and bag B_i. Then, K returns how much the feature vector $\{\mathbf{b}_{ri}\}$ matches its own bag B_i. Let us note that we have only experimented this weighting approach with the basic kernel on bags, but we are planning to experiment other kernels on bag in the future.

These weights can be included into any kernel on bags, for instance the basic kernel on bags (Eq. 1) :

$$\begin{aligned}\hat{K}(B_i, B_j) &= \langle\hat{\Phi}(B_i), \hat{\Phi}(B_j)\rangle \\ &= \langle\textstyle\sum_r \hat{\phi}(\mathbf{b}_{ri}), \sum_s \hat{\phi}(\mathbf{b}_{sj})\rangle \\ &= \textstyle\sum_r \sum_s \langle\alpha_{ri}\phi(\mathbf{b}_{ri}), \alpha_{sj}\phi(\mathbf{b}_{sj})\rangle \\ &= \textstyle\sum_r \sum_s \alpha_{ri}\alpha_{sj}k(\mathbf{b}_{ri}, \mathbf{b}_{sj})\end{aligned}$$

In other words, this leads to the replacement of the minor kernel $k(\mathbf{b}_{ri}, \mathbf{b}_{sj})$ with $\hat{k}(\mathbf{b}_{ri}, \mathbf{b}_{sj}) = \alpha_{ri}\alpha_{sj}k(\mathbf{b}_{ri}, \mathbf{b}_{sj})$.

6. KERNELS EVALUATION

6.1 Database setup

Columbia database has been widely used to evaluate object recognition methods. With 100 objects, and 72 shots from different points of view for each object on a homogeneous black background, it turns out that most of the methods now achieve high performances with very few training data.

In order to retrieve objects in a much more realistic scenario, we built a synthetic database with objects from

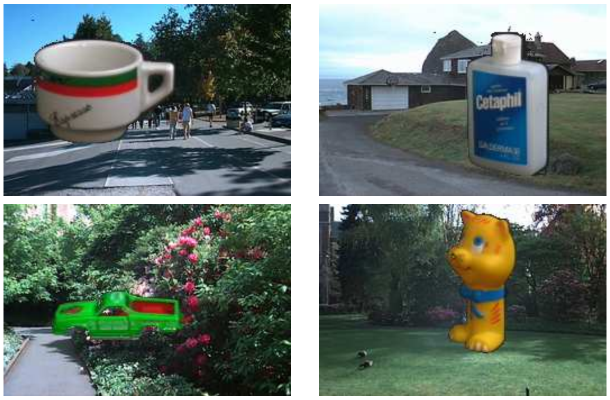


Figure 2: Objects of Columbia database on random backgrounds.

Columbia and background from the Washington database[1]. We randomly selected 12 views of 50 different objects from the Columbia database, removed the background, and embedded them on images of the Washington database (rescaled to a fixed size). The final database contains 600 images. All these backgrounds have very heterogeneous content. Examples are shown in Fig.2. In [8], Lyu also considers an extended database with objects. The one we built for our experiments is comparable to the Lyu's one, except that the objects are much smaller in our final database, making the problem much harder.

6.2 Experimental setup

The computation of the fuzzy regions requires a main parameter which is the number of regions the algorithm should return. As it depends on the level of detail expected by the user, it is specified by an interval. We tested several sizes and found that the interval $[5, 15]$ of fuzzy regions is robust for this kind of images. Each region is represented by 4 histograms, one of 8 chrominances values from $CIEL^{\star}a^{\star}b^{\star}$, and 3 of 8 textures from Gabor filters for 3 different scales.

Thanks to the kernel functions on features, we train SVM classifiers to discriminate images that contains or not the query object. This also allows us to use active learning, whose huge interest for interactive search is now well known [12]. In the following experiments, we use a precision-oriented active learning technique [3]. Each retrieval session is initialized with one image containing the object the user is looking for (*cf.* Fig. 3). Next, 5 images chosen by active learning are labelled with binary annotations (positive: the image contains the query object, negative: the image does not contain the query object). This process is repeated 5 times. At the end of retrieval session, the size of the training set is then 26 labels. Performances are evaluated with the Mean Average Precision, i.e. the sum of the Precision/Recall curve[2].

6.3 Maximum functions

We first experimented the Maximum function (Eq. 2) and the Average of Maximum function (Eq. 3) with different minor kernels. Results are shown on Fig. 4(a) and (b). Globally, one can see that the Average of Maximum is more

Figure 3: The RETIN graphic user interface. Top part: retrieved images; Bottom part: images selected by the active learner.

interesting than the Maximum function, certainly because a single region is not always sufficient to discriminate an object. Comparing minor kernels, the Gaussian-based kernels are the most interesting, especially the Gaussian kernels with χ^1 or χ^2 distances. This can be explained by the fact that Gaussian-based kernels are good matching functions, as soon as the image of the feature space is an hypersphere of $\mathcal{H}$. In other words, when two feature vectors match, the value returned by a Gaussian-based kernel is always close to 1, independently to their norm.

Next, we modified these two bag functions with the weighting approach proposed in section 5.3. Results are shown on Fig. 4 (a') and (b'). In many cases, this increases the performance by around 10%, especially with the Gaussian-based kernels. Again, this result is due to the capacity of these kernels to behave like a matching function.

6.4 Power kernels

We experimented the Power kernel (Eq. 5) which is the approximation of the Maximum function, and the Average Power function (Eq. 6) which is the approximation of the Average of Maximum function. We only tested for the most relevant minor kernel found during the previous experiments: Gaussian χ^1, χ^2, L2, and the Triangle kernel. We first confirmed that the power kernels are good approximations of maximum function, since their behavior tends to be the same with high values of q. Next, we carried out sereval runs to tune the parameter q in these functions. Results are shown on Fig. 4(c) and (d). Concerning the Power kernel,

[1]http://www.cs.washington.edu/research/imagedatabase/
[2]cf. TREC VIDEO conference:
http://www-nlpir.nist.gov/projects/trecvid/

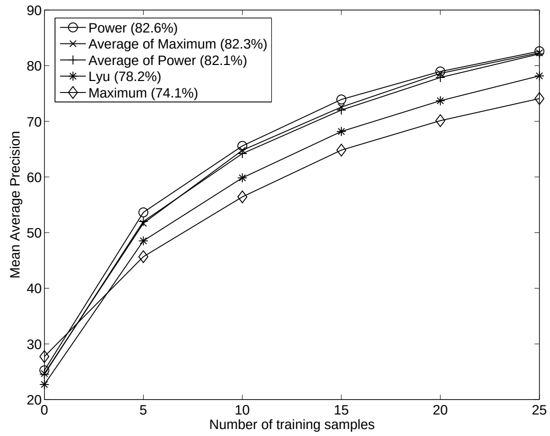


Figure 5: Comparison of kernels on bags with the best parameters tuning.

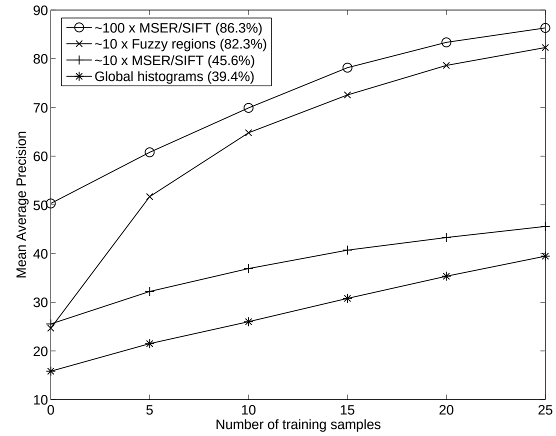


Figure 6: Performances with different features.

the tuning of parameter q is interesting, since the performances are enhanced. However, for the Average of Power function, the best q were the highest ones, which means that the Average of Maximum is the most efficient function.

We also tested these kernels with the weighting approach proposed in section 5.3. Results are shown on Fig. 4(c') and (d'). Again, we observe an increase of around 10%.

6.5 Comparison of kernels on bags

We summarized in Fig. 5 the results of each kernel on bags with the best parmater tuning :
• Maximum, weighting approach, Gaussian χ^2
• Average of Maximum, weighting approach, Gaussian χ^2
• Power, $q = 1.6$, weighting approach, Gaussian χ^2
• Average of Power, $q = 5$, weighting approach, Gaussian χ^2
• Lyu's kernel [8], $q = 1.4$, weighting approach, Gaussian χ^2

It turns out that Power, Average of Power and Average of Maximum have the same performances, followed by Lyu's kernel and the Maximum. This shows the interest of the Power kernel which, after a good tuning, provides excellent performances while being a true kernel function. However, efficiency speaking, the Average of Maximum is the most interesting, despite it is not a true kernel function, it does not require any paramater tuning, except for the choice of the minor kernel.

7. OTHER EXPERIMENTS

7.1 Feature comparison

We compare local features computed on regions to global features which represent an image with a single histogram of colors and textures [3]. We can see on Fig. 6 that local representations clearly outperform the global one. We also compare our fuzzy region indexing technique to the well-known approach combining MSER region detectors and SIFT descriptors[9]. We built two sets of MSER/SIFT descriptors, one with around 100 points per image, and another one with around 10 points per image. We also tested the different kernels for MSER/SIFT, and the same conclusions about the best kernel on bags and minor kernel were found (any kernel on bags except the Maximum, weighting approach, and Gaussian χ^2). With around 100 PoI per image, our kernel on bags technique performs better than when using region features. However, for equal numbers of features (around 10), the kernels on bags of regions performs significantly better. Thus, in terms of computational complexity, using bags of around 10 regions is more interesting for real applications since it provides performances a little lesser to 100 MSER/SIFT, but runs 100 times faster.

7.2 An example of retrieval session

We present in this section an example of interactive search of all images containing a specific object with our graphic user interface (*cf.* Fig. 3). The session is initialized with one image containing the query object, a white plastic cat. As one can see on Fig. 7(a), the first returned image is a red truck on exactly the same background - which is not surprising since this produces perfect matches[3]. Next, we labeled the images selected by the active learner, and then we asked for an update of the ranking (green squares on Fig. 7(b)). At this step, we get 9 out of the 12 images that contain our object. Notice that in the 3 last images, there is one with exactly the same background than one of the positive labeled images. The two other ones are images with a lot of sky, which is very present is every images we labeled as positive. Then, labeling these images as negative drops off the sky features, and brings two images with our object, and a last one with a green background (Fig. 7(c)). This shows the behavior of the system - in case of redundant features, it assumes that they are relevant for the object. Finally, we update the ranking with new labels, and get the 12 objects of the class (Fig. 7(d)).

8. CONCLUSION

A kernel framework to deal with database object retrieval is presented in this paper. The database objects we are dealing with are embedded in images with heterogeneous background. Our approach is based on local feature analysis:

[3]On this database, we have several images with exactly the same background from the Washington database.

first, a fuzzy segmentation is proposed to get fuzzy regions with overlapping. The process includes a region merging step to better control the final number of region, and is thus fully automatic. The segmentation is coarse, edges are inaccurate when there is no sharp gradient, regions overlap somewhat. But this inaccuracy is compensated by the feature computation which takes into account the membership degres of the pixels In comparison to point of interest approaches, the required number of local features to get an efficient representation of an image is quite smaller.
Region characterization next provides bags of feature vectors for each image. The resulting bags of features are matched thanks to a new class of kernels on sets to deal with images composed of objects with heterogeneous background. These kernels highlight the best local matches between features, so that they efficiently overcome the artifacts introduced by the background features. Results show that our approach is able to perform object retrieval in complex databases with heterogeneous backgrounds. The local region-based approach combined with powerful kernels designed on bags performs very well and provides a nice tradeoff between retrieval and computational efficiency.

9. ACKNOWLEDGMENTS

This work has been funded by the European network of excellence MUSCLE (*cf.* http://www.muscle-noe.org).

10. REFERENCES

[1] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(8):1026–1038, 2004.

[2] Y. Chen and J. Wang. Image categorization by learning and reasoning with regions. *International Journal on Machine Learning Research*, 5:913–939, 2004.

[3] P. Gosselin and M. Cord. Precision-oriented active selection for interactive image retrieval. In *IEEE International Conference on Image Processing*, Atlanta, GA, USA, October 2006.

[4] K. Grauman and T. Darell. Efficient image matching with distribution of local invariant features. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, San Diego, CA, June 2005.

[5] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *International Conference on Computer Vision and Pattern Recognition*, 2005.

[6] R. Kondor and T. Jebara. A kernel between sets of vectors. In *International Conference on Machine Learning (ICML)*, 2003.

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

[8] S. Lyu. Mercer kernels for object recognition with local features. In *IEEE International Conference on Computer Vision and Pattern Recognition*, San Diego, CA, 2005.

[9] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65:43–72, November 2005.

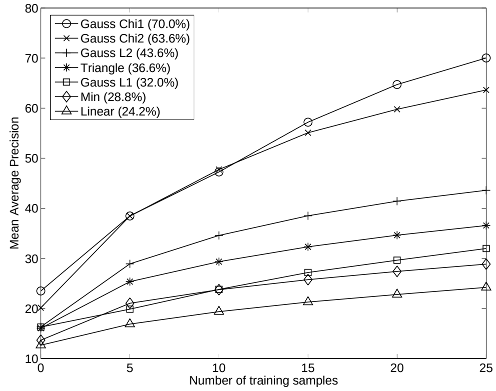
[10] S. Philipp-Foliguet and J. Gony. FReBIR : Fuzzy regions-based image retrieval. In *Image Processing and Multimedia Understanding (IPMU)*, Paris, France, July 2006.

[11] J. Shawe-Taylor and N. Cristianini. *Kernel methods for Pattern Analysis*. Cambridge University Press, ISBN 0-521-81397-2, 2004.

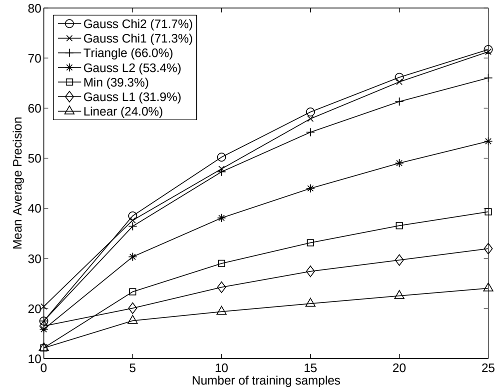
[12] S. Tong and D. Koller. Support vector machine active learning with application to text classification. *Journal of Machine Learning Research*, pages 2:45–66, November 2001.

[13] V. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, New York, 1999.

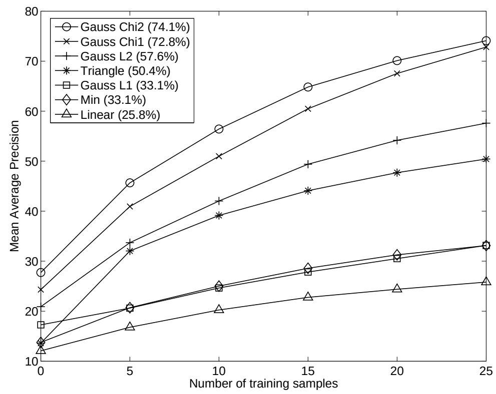
[14] C. Wallraven, B. Caputo, and A. Graf. Recognition with local features: the kernel recipe. In *International Conference on Computer Vision (ICCV)*, volume 2, pages 257–264, 2003.



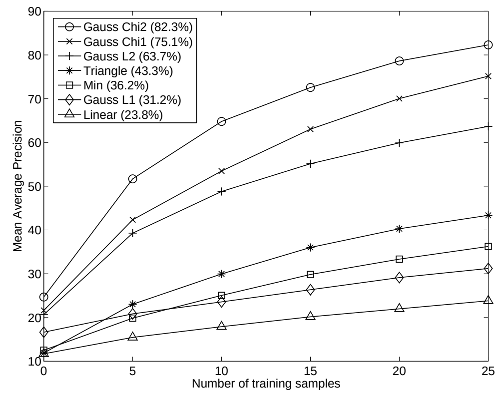
(a) Maximum



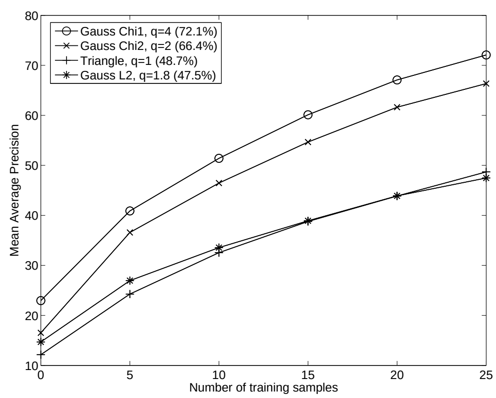
(b) Average of Maximum



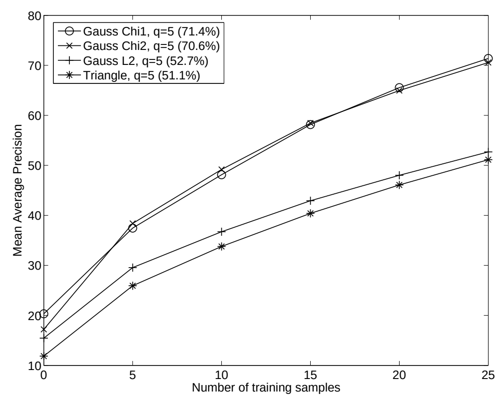
(a') Weighted Maximum



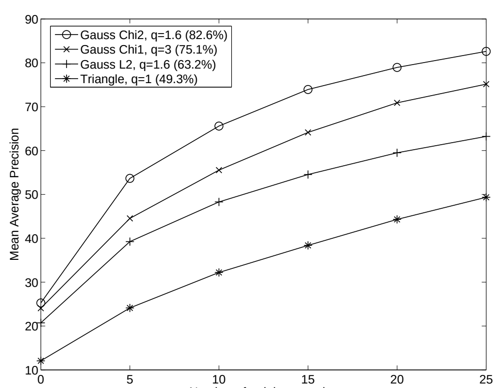
(b') Weighted Average of Maximum



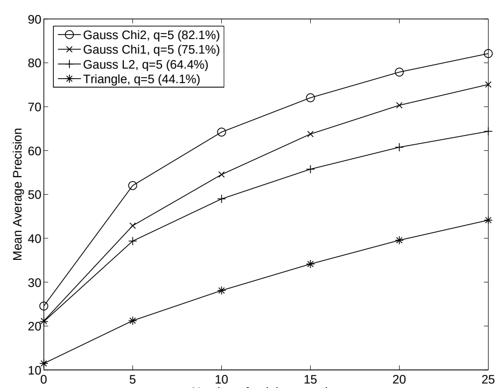
(c) Power



(d) Average of Power

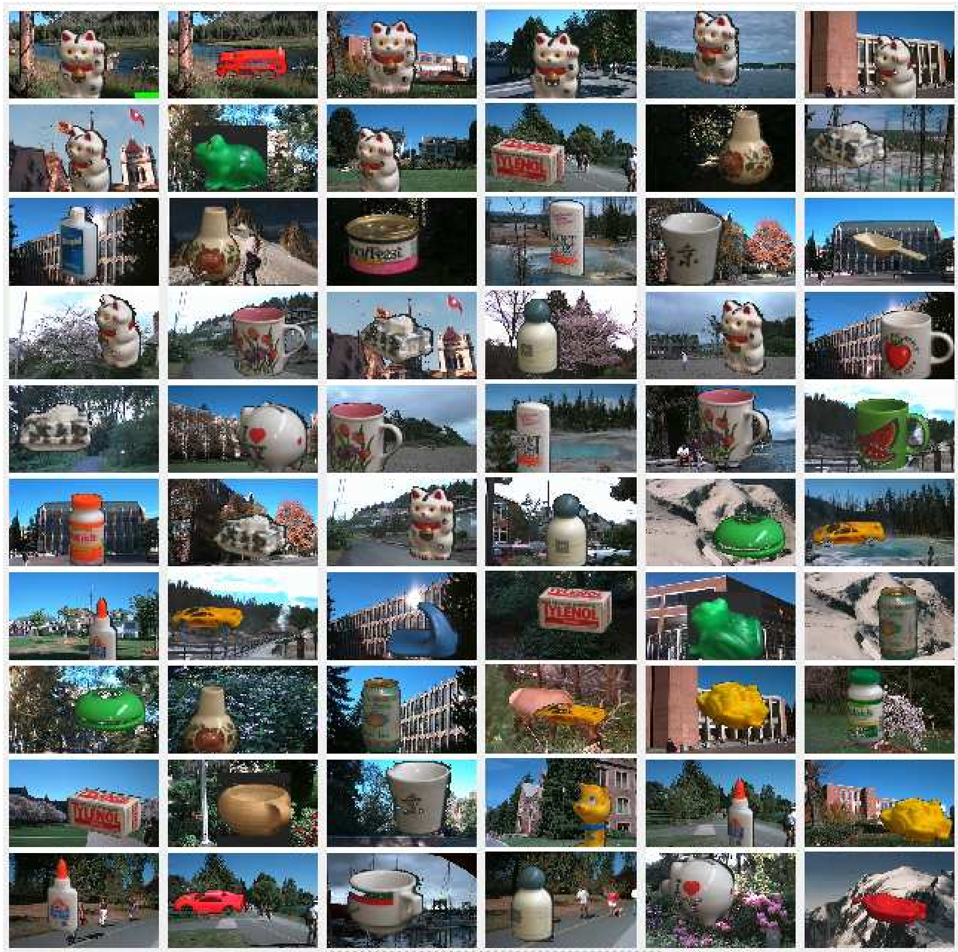


(c') Weighted Power

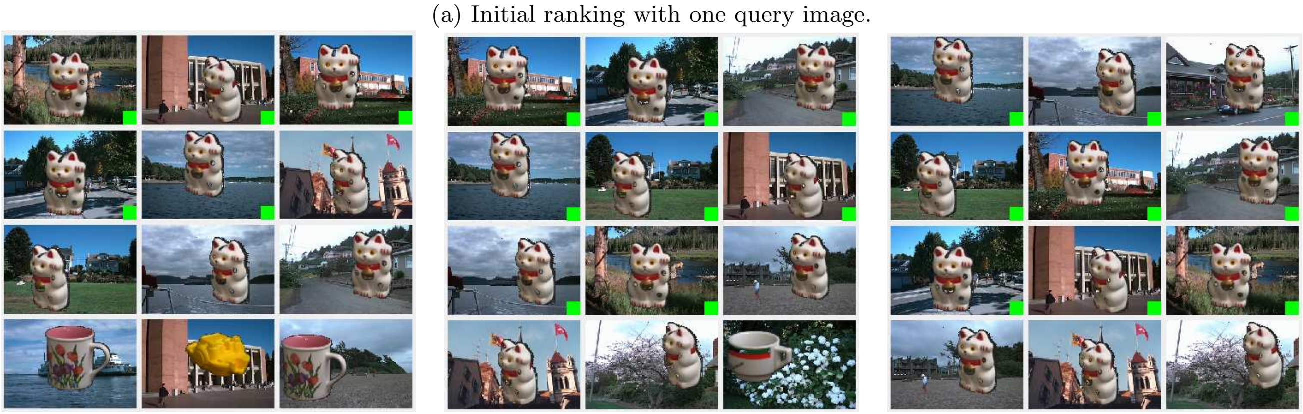


(d') Weighted Average of Power

Figure 4: MAP(%) with each kernel on bags, for different minor kernels.



(a) Initial ranking with one query image.



(b) Iteration 1 (c) Iteration 2 (d) Iteration 3

Figure 7: An example of retrieval session. Initialization with 1 image, 5 labels per iteration according to the active learner.