Multimedia indexing and fast retrieval based on a vote system

S. Philipp-Foliguet[(1)], G. Logerot[(1,2)], P. Constant[(2)], PH. Gosselin[(1)], Christian Lahanier [(3)]

(1) ETIS, ENSEA/UCP/CNRS, 6 avenue du Ponceau, 95014 Cergy , France
(2) PERTIMM, 44 rue Pierre Brossolette, 92600 Asnières, France
(3) C2RMF, Palais du Louvre, Porte des Lions, 14 quai F. Mitterrand, 75001 Paris

ABSTRACT

We present a new system, called Retimm, for searching databases made of documents containing images and text. Images are indexed by colour and texture distributions.. Colour and texture classes are obtained by a quantization adapted to the whole database. Signatures are ranked *m* times, once for each dimension, but values are not stored. The search engine works as a vote system : the score for each document is the total of the votes of all coordinates, these last votes depending on a *k*-nn search on each dimension. Retimm is able to retrieve very quickly images from large databases from any request composed of one or several images and/or one or several words. The system is interactive, since the query can be modified at any moment by adding or removing images or words.

1. INTRODUCTION

Among the huge number of existing image databases, some of them contain textual information. For example medical images are part of a medical file, which includes a lot of information, concerning the patient, his illness, and so on. On websites, pages usually contain text and some images illustrating the text. Image and text are complementary and users would like to use both to browse through Internet or to search databases. Many works deal on the one hand with Content-Based Image Retrieval (CBIR) and on the other hand with the text retrieval, but few manage both information at the same time. CBIR systems sometimes take inspiration from text retrieval for the image representation. For example in Viper system [6] images are indexed by a huge number of visual features which can either be present or absent in each image, as words in a text. In [8] a vocabulary of "keyblocks" aiming at representing the image content are treated as words in a textual document. Some works aim at automatically annotating images like [1], where a set of keywords is assigned to each image after segmentation. Actually few systems combine textual and visual information. Usually, as noticed by Westerveld [7] both modalities are searched separately and merged in an ad hoc fashion.

Our application domain is the artwork databases, and more precisely painting databases [5]. Together with the image, which is a digital representation of the painting, there are texts. These texts usually include the title of the artwork, the painter's name, the date, and a lot of other information concerning the history of the painting (restoration for example), or the content of the painting (style, school, etc.).

The aim is to retrieve either a precise image or a set of images. In both cases the query is made of one or several images and one or several words. We will see that the query can be updated during the process.

An image can be described by visual features and by keywords. Concerning the visual features, we used global signatures based on colour and texture histograms. They are presented in section 2. Despite the size of the database, the user wants to access rapidly to the images he is looking for. So the way the signatures are stored are of the highest importance (see section 2).

For the keywords, we only used words linked to the image semantics : words of the title and of the comments.

For the research, we used a search engine developed by Pertimm, originally conceived for text research, and we used it with both visual and textual features.

Our aim is double ; first we propose an image representation by visual features and an indexing scheme of both visual features and words. Secondly, we present our retrieval system called Retimm, able to retrieve very quickly images from a large database from any request composed of images and/or text.

2. IMAGE INDEXING

Image representation through visual features must have two complementary but contradictory properties : compacity and efficiency. The problem for colour coding consists in finding a palette allowing a similarity computation, that is why a fixed palette is the most often chosen. If this palette has to be common for all images of the database, it is better if it is adapted to the database content. So we have chosen to built signatures based on a *C*-means classifier.

Each image is represented by a global signature aiming at coding the colour and the texture distributions. HSV space is used for color, and twelve Gabor filters in 3 different scales and 4 orientations are used for texture analysis. Both spaces are quantified using an enhanced

version of LBG algorithm [4]. From previous tests made with our CBIR system [3] we know that 256 is an upper bound of the class number enough for a good quantization. So each image is first quantified in 256 classes for colour and 256 for texture. Then these 256 classes for each image are quantified in a given number of classes, which can be different for colour and texture. The advantage of such a classification in two steps is that the addition of new images in the database does not require the re-computation of all the classes (as far as new images are not too many of them or too different from the rest of the database). We will show in section 4 that an accurate choice of the class number is not crucial. So the visual signature of each image is composed of one vector of *m* features representing the colour and texture distributions.

Most of the CBIR systems just store the signatures sequentially. But the search can be drastically accelerated by using more judicious indexing of the data. For example inverted files are used in [6]. In order to be very fast in the on-line phase of image retrieval, we used the following indexing scheme : signatures are ranked m times, once for each dimension, but values are not stored. We will see in the following section, that values themselves are not used in our similarity measure, but only the rank in each coordinate. So instead of storing a m-dimension vector for each image, we store m times sorted references to each image (once for each coordinate).

This off-line processing will considerably accelerate the on-line searches.

3. VOTE-SYSTEM

There are many ways to measure similarities between vectors. Retimm engine uses a vote system. The aim is to be as fast as possible, in order to be able to mine huge databases. The method is also optimised to require as less memory access as possible.

Let us first explain the method with only one request image, represented by a vector of dimension *n*. Of course it exists algorithms to approximate the *k*-nearest neighbours (*k*-nn) research in O(*k.n.* log *N*) if *N* is the number of images in the database [1]. We propose a method which is even faster, though it does not approximate exactly the Euclidean distance. It consists in giving a score for each image of the database, allowing a ranking of images towards the request image.

For a feature space of dimension *n*, the system performs *n* votes, one for each dimension of the feature vector. The score for each image is the total of the votes of all coordinates. The vote is based on a *k*-nn search on each dimension : for each coordinate, a score of 1 is attributed for each image whose coordinate is amongst the *k*-nn. The scores are added for all coordinates leading to the final score.

This vote system allows image ranking ,but it has none of the properties of a distance. The maximum score can be obtained by several images (including the request). The vote is not symmetric : if an image belongs to the *k*-nns of a request for a given feature, the request does not necessary belong to the *k*-nns of this image for the same feature. Moreover, images having the same score can have feature vectors completely different, one for example being close to the request by the texture features and the other one by the colour features.

In order to perform quantitative evaluation of our system, we first used an image database, for which we have the ground-truth. It is a general database of 1200 photographs of various kinds (see Fig. 4). They are classified into classes, so it is possible to make statistics and comparisons.

We first compared various sizes of visual signatures and various values of *k*. To perform the comparison we used the Mean Average Precision (MAP) used in TREC Video conference. One can see from the curves (Fig. 1) that increasing the texture dimension is of no use, whereas increasing the colour dimension slightly improves results, but gains are weak (4 % for 90 instead of 50 colours) compared to the increasing of computation time. The most important parameter is the number of neighbors *k* used for the vote. Fig. 1 clearly shows that, whatever the signature size, this number must be larger than 100 but there is no advantage of increasing it over 300.

Another important factor is the time used to retrieve images (cf. Fig. 2). Of course it linearly increases with *k* and with the number of query images, but it remains weak (less than half a second) for $k < N/2$. Finally depending on the constraints, precision versus rapidity, *k* can be chosen between $N/10$ and $N/2$, where *N* is the database size.

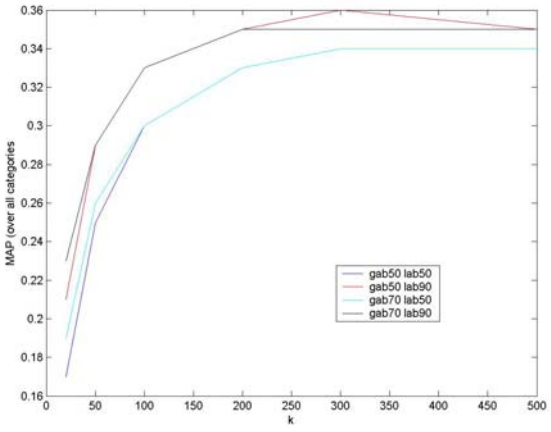

Fig. 1. Mean Average Precision on all categories of the general database of 1200 photographs, according to *k*, for 4 visual signature sizes

The other test we performed aims at validating the choice of the vote system towards the Euclidean distance. The query consists in a single image randomly drawn in each of the tested categories (airplane, lion/tiger and portrait). With an Euclidean distance, one can only retrieve images which are close for all features. This penalizes multimodal categories. For example, for the car category, if a red car is given as query, there is little chance to retrieve yellow or black cars, which are not close in the feature space, although their texture features are close. For each of the three categories, the vote system retrieves more images of the category than the Euclidean distance (cf. Fig. 3).

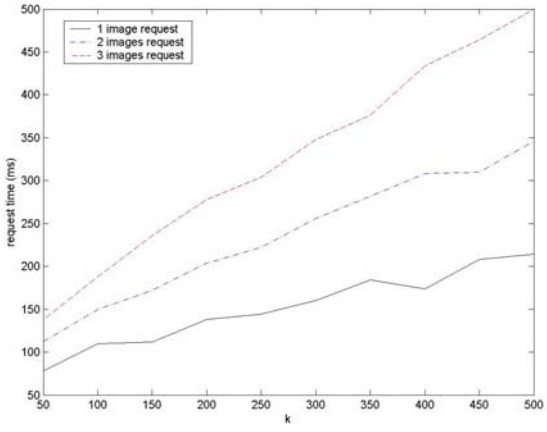

Fig. 2. Search time according to *k*, for queries made of one, two or three images

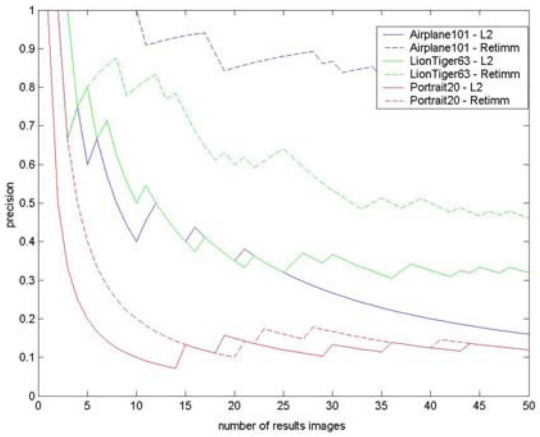

Fig. 3. Precision curves comparing Retimm votes and L_2 (Euclidean) distances for three categories.

The vote-system is able to manage multimodal categories, which is one of the problems in image retrieval. A semantic category is often split into several modes in the feature space, and the user does not always wish to choose himself the sets of discriminating features. The vote system can indeed work with several query images instead of one : the score is the sum of the scores for each of the query images. If the query images are representative of the various modes of the category, images of all modes will receive high scores. For example Fig. 4 displays the 32 best results of a car search starting from a query made of three cars (of different colours). Retrieval times is 450 ms.

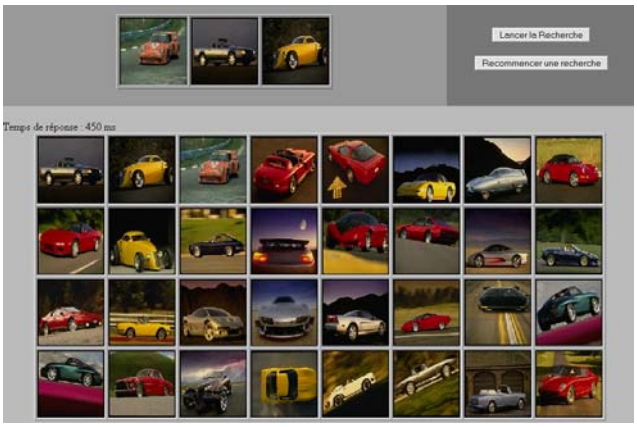

Fig. 4. Result of a research in the general database with a query made of 3 images (at the top)

4. TEXT AND IMAGE SEARCH

With this vote system, there is a great flexibility in the way of searching the database. First the user can choose the features : he can indifferently use visual features such as texture or colour, but also words. Secondly, he can choose the type and the number of queries (either images or words) and he can add or suppress images or words as required. The similarity is simply updated by adding (respectively subtracting) votes of each added (respectively removed) image or word.

Moreover images or words can be presented as counter-examples, the system only has to subtract the scores of the corresponding features.

For the multimedia search, we use a database composed of 18,755 images from the C2RMF database representing 3,168 different paintings. The most frequent words of the titles are “saint” (2,256 occurrences), “portrait” (1,997),”virgin” (913), etc. amongst a total of 12,087 words.

We have compared results of the vote system with k=1000 for a portrait search (if possible of the face only) starting from various queries. For a fast search, we used a signature with 50 colour features and 50 texture features. In Fig. 5 the query consists in one image, and only visual features are used. The 24 first results include many other images than portraits. With a second example of portrait image, the results are improved (Fig. 6). And with the adding of the word “portrait”, the results are even better. The word “portrait” appears in the title or comment of 1,997 images. A query with the only “portrait” word returns very

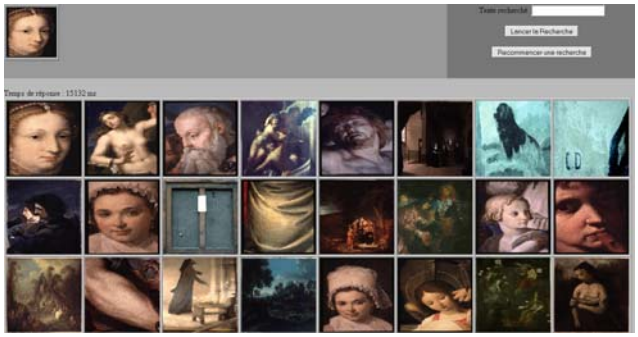

Fig. 5. 24 first retrieved images with visual features only and a query made of one image (top left)

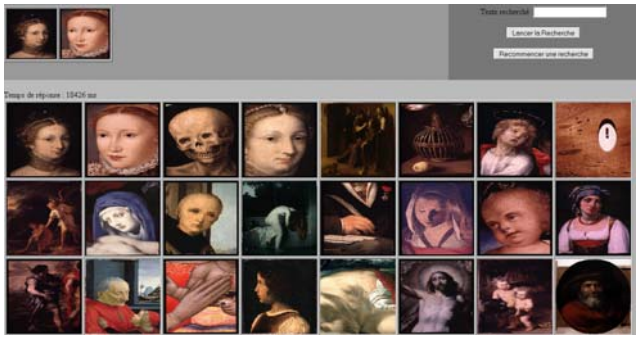

Fig. 6. 24 first retrieved images with visual features only and a query made of two images

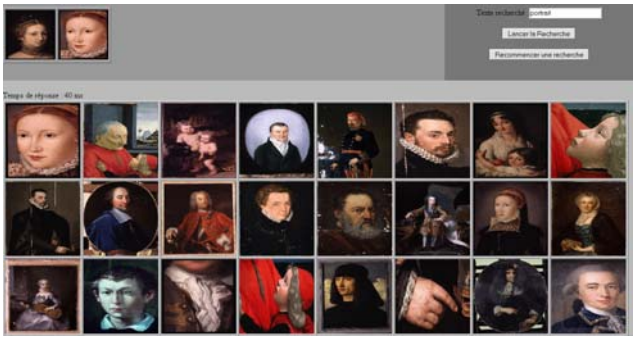

Fig. 7. 24 first retrieved images with one keyword and the visual features of two images

various images, including backs of painting, or details, etc. (see Fig. 8)

5. CONCLUSION

System Retimm is able to manage documents containing images and texts. With the vote system independent for each coordinate of the feature space, it allows to manage queries made of one or several images and one or several words. It is both efficient (better than the Euclidean distance) and fast. It is very flexible, since the user can add or remove words and images during the search. At last we have given a way for building visual signatures adapted to the database content, and a efficient way to store them.

Another advantage of Retimm system is interactivity. With the vote system, the user can start the search with any image or word and add or remove images and words as required.

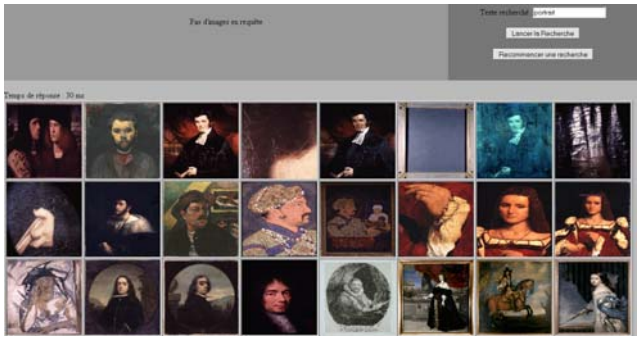

Fig. 8. 24 first retrieved images with the keyword "portrait"

6. REFERENCES

[1] S. Arya, D. M. Mount, N. S. Netanyahu, Ruth Silverman and A. Y. Wu, "An optimal algorithm for approximate nearest neighbor searching fixed dimensions", *Journal of the ACM*, 45(6), 891-923, 1998

[2] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, and M.I.Jordan, Matching Words and Pictures, *J. of Machine Learning Research*, 3, 1107-1135, 2003

[3] J. Fournier, M. Cord, S. Philipp-Foliguet, "RETIN: A Content-Based Image Indexing and Retrieval System", *Pattern Analysis and Applications Journal*, Special issue on image indexation, 4,(2/3), 153-173, 2001

[4] P.H. Gosselin, "Méthodes d'apprentissage pour la recherche de catégories dans des bases d'images" , PhD thesis, Cergy-Pontoise, France, dec. 2005

[5] R. Pillay, D. Pitzalis, C. Lahanier and G. Aitken, EROS : New Development in Digital Archiving for Research in Conservation : EVA2005, Florence, Pitagora, p.192-197, March 2005

[6] D. M. Squire, W. Muller, H. Muller, and J. Raki, "Content-based query of image databases, inspirations from text retrieval: inverted files, frequency-based weights and relevance feedback", 10th Scandinavian Conference on Image Analysis (SCIA'99), Kangerlussuaq, Denmark, june 1999

[7] T. Westerveld, "Using generative probabilistic models for multimedia retrieval", PhD thesis, Twente, Nederland, sept. 2004

[8] L Zhu, A Rao, A Zhang, "Theory of keyblock-based image retrieval", *ACM Trans. on Information Systems*, 20(2), 24-257, 2002