BOOSTING OF FACTORIAL CORRESPONDENCE ANALYSIS FOR IMAGE RETRIEVAL

Nguyen-Khang Pham **, Annie Morin*, Patrick Gros**

* IRISA
Campus Universitaire de Beaulieu, F-35042 Rennes Cedex
{pnguyenk, amorin, pgros}@irisa.fr

** Cantho University
Campus III, 1 Ly Tu Trong Street, Cantho City, Vietnam
pnkhang@cit.ctu.edu.vn

ABSTRACT

We are concerned by the use of Factorial Correspondence Analysis (FCA) for image retrieval. FCA is designed for analysing contingency tables. In Textual Data Analysis (TDA), FCA analyses a contingency table crossing terms/words and documents. For adapting FCA on images, we first define "visual words" computed from Scalable Invariant Feature Transform (SIFT) descriptors in images and use them for image quantization. At this step, we can build a contingency table crossing "visual words" as terms/words and images as documents. In spite of its successful applications in information retrieval, FCA suffers from large dimension problem because of the diagonalization of a large matrix. We propose a new algorithm, CABoost, which overcomes this large dimension problem of FCA. The data are sampled by column (word) and a FCA is applied on the sample. After some samplings, we finally combine separated results by a weighting - Principle Component Analysis (PCA). The numerical experiments show that our algorithm performs more rapidly than the classical FCA without losing precision.

1. INTRODUCTION

Content based image retrieval (CBIR) aims to searching images which share some visual parts with a query in large databases. This is a very difficult task. It is due to occlusions, background clutters, and viewpoint or orientation changes. Recently, the use of local descriptors in images has shown to be a good choice for image analysis. Contrary to global descriptors which are computed from entire image, local descriptors are extracted at particular interest points in image. This allows finding images which share only one or some similar visual elements with the query. Initially vote-based methods were used for image retrieval by matching interest points [1, 2]. Later, the methods developed originally for Textual Data Analysis such as LSA (Latent Semantic Analysis) [3], pLSA (probabilistic Latent Semantic Analysis) [4, 5], LDA (Latent Dirichlet Allocation) [6] have been adapted on images [7, 8, 9, 10]. In Textual Data Analysis, these methods are based on a bag-of-words model. They take as input a co-occurrence matrix (called also contingency table which crosses documents and terms/words) and try to reduce dimensions. When adapting on images, we have images as documents and "visual words" as terms/words. Among the disadvantage of the methods above, we find the use of an ad hoc model and an EM algorithm to seek a local optimum and the difficulty to interpret the results. Most of the works use such methods as black boxes. Here, we focus on the use of Factorial Correspondence Analysis (FCA) for the retrieval of images. This work is motivated by the successful application of FCA on textual data [11]. FCA reduces the space representing images and defines the similarity among images in a smaller space. In early work, we proposed the use of FCA for image retrieval [12]. It was shown that FCA performed better than term frequency - inverse document frequency weight (TF*IDF) [13] and PLSA. Nevertheless, one of the main problems of FCA is matrix diagonalisation. This task is very time-consuming especially with high order matrices. To overcome this problem, we propose a new algorithm, called CABoost, which reduces the learning time when the vocabulary's size is large.

The article is organized as follows: we briefly describe the FCA method in the section 2. Section 3 presents word construction and image representation. The new algorithm, CABoost, is presented in the section 4. Section 5 shows some numerical results. In the last section, we present some perspectives for this work.

2. FACTORIAL CORRESPONDENCE ANALYSIS

FCA is a classical exploratory method for the analysis of contingency tables. It was proposed by J. P. Benzecri [14] in the linguistic context, i.e. textual data analysis. The first study was performed on the tragedies of Racine. FCA on a table

crossing words and documents allows answering the following questions: Is there any proximity among certain words? Is there any proximity among certain documents? Is there any link among certain words and certain documents? FCA like most factorial method uses a singular value decomposition of a particular matrix and allows viewing words and documents in a reduced space. This reduced space has a particular propriety where points are projected (words and/or documents) with a maximum inertia. In addition, FCA provides some relevant indicators for the interpretation of the axes as the contribution of a word or a document to the inertia of the axis or the representation quality of a word and/or document on an axis [15, 11]. We now briefly describe the method:

Given a contingency table, $F = \{f_{ij}\}_{M,N}, (N < M)$ we normalize F to X by:

$$\begin{aligned} s &= \sum_{i=1}^{M}\sum_{j=1}^{N} f_{ij} \\ x_{ij} &= \frac{f_{ij}}{s}\ ,\ \forall i = 1..M, j = 1..N \end{aligned}$$

and note:

$$p_i = \sum_{j=1}^{N} x_{ij}\ ,\ \forall i = 1..M \qquad q_j = \sum_{i=1}^{M} x_{ij}\ ,\ \forall j = 1..N$$

$$P = \begin{pmatrix} p_1 & & 0 \\ & \ddots & \\ 0 & & p_M \end{pmatrix} \qquad Q = \begin{pmatrix} q_1 & & 0 \\ & \ddots & \\ 0 & & q_N \end{pmatrix}$$

To determine the best sub-space for data projection, we calculate the eigenvalues and eigenvectors of the matrix $V = X^T P^{-1} X Q^{-1}$ with size $N \times N$ where X^T is transpose of X.

We then obtain the eigenvalues λ and eigenvectors μ of the matrix V:

$$\lambda = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \end{pmatrix} \qquad \mu = \begin{pmatrix} \mu_{11} & \mu_{12} & \dots & \mu_{1N} \\ \mu_{21} & \mu_{22} & \dots & \mu_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{N1} & \mu_{N2} & \dots & \mu_{NN} \end{pmatrix}$$

We keep only $K(K < N)$ first eigenvalues and their corresponding eigenvectors. These K eigenvectors constitute an orthonormal basis of the reduced space (also called, factor space). The number of dimensions passes from N to K. The documents (images) are projected in the reduced space by the following:

$$Z = P^{-1} X A \quad where \quad A = Q^{-1}\mu \tag{1}$$

In this formula, $P^{-1}X$ represents line profiles and A is the transition matrix associated to the FCA. The new coordinates of the terms/words are computed by:

$$W = Q^{-1} X^T Z \lambda^{-1/2} \tag{2}$$

An unseen document (i.e. query) $r = [r_1\ r_2\ \dots\ r_N]$ will be projected in the reduced space by the transition formula (1):

$$Z_r = \hat{r} A \quad where \quad \hat{r}_i = \frac{r_i}{\sum_{j=1}^{N} r_j} \tag{3}$$

3. IMAGE REPRESENTATION

In order to adapt FCA on images, we must represent the image corpus in the form of contingency table. Here images are treated as documents and "visual words" (to be defined) as terms/words.

Words in the images, called "visual words", must be calculated to form a vocabulary of N words. Each image will be represented by a word histogram. The construction of visual words is processed in two steps: (i) computation of local descriptors for an image set, (ii) classification (clustering) of obtained descriptors. Each cluster will correspond to a visual word. Local descriptors in an image are also computed in two stages: we first detect the interest points in the image. These points are either maximums of Laplace of Gaussian [16], or 3D local extremas of Difference of Gaussian [1], or the points detected by a Hessian-Affine detector [2]. Figure 1 shows some interest points detected by a Hessian-Affine detector. The descriptor of interest points is then computed on gray level gradient of the region around the point. The scalable invariant feature transform descriptor, SIFT [17] is often preferred. Each SIFT descriptor is a 128-dimensional vector. An example of SIFT is shown in figure 2. The second step is to form visual words from the local descriptors computed in the previous step. Most of the works perform a k-means on descriptors and take the averages of each cluster as visual word [7, 8, 9, 10, 18]. After building the visual vocabulary, each descriptor is assigned to the nearest cluster. For this, we compute, in $\mathbf{R}^{128}$, distances from each descriptor to the representatives of previously defined clusters. Thus an image is characterized by the frequency of its descriptors and the image corpora will be represented in the form of a contingency table crossing images and clusters (visual words).

In our experiments, we use the method described in [2] to detect interest points. The vocabulary is built using a k-means with about 300000 descriptors drawn randomly (one third for each category: faces, motorbikes, airplanes, cars and background). The obtained vocabulary consists of 2224 words from 4090 images. The number of words in the vocabulary was chosen by Sivic [9].

4. BOOSTING OF FCA

FCA invokes an "eigensolver" for eigenvalues and eigenvectors. This task is time and memory consumed for high order matrix. The main idea for overcoming this problem is sampling on the data and applying FCA on each sample. By this

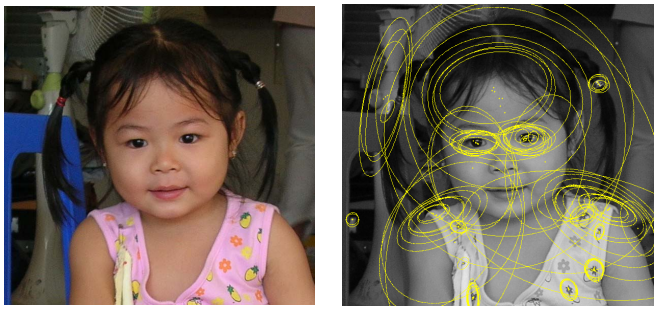

Fig. 1. Interest points detected by Hessian-Affine detector

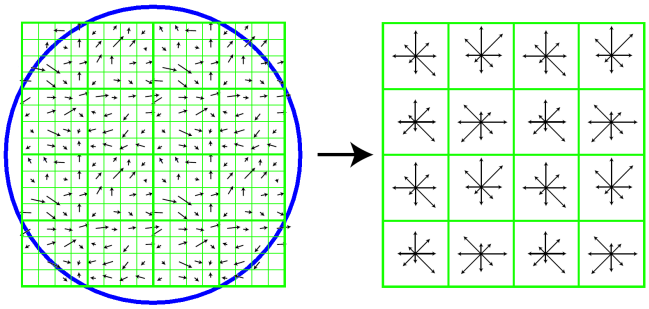

Fig. 2. A SIFT descriptor computed from the region around the interpret point (the circle): gradient of the image (left), descriptor of the interest point (right)

way we can benefit from the information of all words with a low cost because we diagonalise only small matrices.

4.1. Word sampling

We explore the partition of words into two ways: deterministic and random. Note that it is also possible to combine two approaches. In the deterministic approach, words are sorted in descending order by their frequency and by the number of images in which they appear. So we obtain one list of words for each criterion. Two lists are merged to give $Size$ ($Size$ is a fixed parameter) first words that appear at the same time in the both lists. We then apply FCA on only selected words. After the first FCA, words which contribute a lot to the inertia of some first axis, are removed from two lists. About two third selected words are removed. The fact that we remove some words with high contribution allows us to find new topics with remained words because a group of words with high contribution to the inertia of an axis defines a topic. We keep on choosing $Size$ next words for the second FCA in the same manner such as with the first FCA after removing some words and so on for next iterations. In random approach, the simplest solution is that words are uniformly sampled. The result could be slightly improved when the word distribution was updated by under weighting the words which were selected in previous iterations. This motivated a hybrid approach in which we updated the word distribution by taking into account the contribution of words to the inertia of axes as in the deterministic approach. The distribution of words can be initialized uniformly or by the inverse of their order in two sorted lists.

4.2. Result merging

The final distance between two images is computed from all results of FCA on samples:

$$d(a,b) \quad = \quad \sum_{i=1}^{T} \alpha_i d_i(a,b) \qquad (4)$$

Where $d_i(a,b)$ is the distance between two images a and b following the i^{th} model. The scalars α_i could be determined as in [19] where we performed a PCA in considering groups of words as points in $\mathbf{R}^{M*Size}$ with M is the number of images and $Size$ is the number of words in a group. α_i is the coordinate of i^{th} group on the first axis. The point clouds could be also stretched by Generalized Procrustes Analysis [20]. One of the simplest weighting described in Multiple Factor Analysis [19] normalises point clouds so that their inertia on the first axis is equal to one. By this way α_i is set to the inverse of the square root of the first eigenvalue of the i^{th} model.

However, the combination of the separated results (4) leads to increase the retrieval time because we have to compute the distance on all T models. To solve that, we propose to apply a PCA on weighted results in order to reduce the dimension. PCA allows to eliminate the redundancy and to look for common factors from all sub results [19]. The algorithm is finally given in the table 1.

5. NUMERICAL RESULTS

We test our algorithm on the Caltech4 dataset [9] drawn form Caltech101 [21]. The algorithm is implemented in C++ using CLAPACK library [22]. The precision - recall curve is used for performance comparison. We use TF*IDF weighting schema [13] with cosine distance as baseline method. In all of our experiments, FCA (classical or boosting of FCA) gives much better results than TF*IDF in time retrieval and result quality.

5.1. Dataset

The Caltech4 database contains 4090 images divided into 5 categories. Table 2 describes this database.

5.2. TF*IDF

In TF*IDF weighting schema, each element $F(i,j)$ in the contingency table is normalized to $tf(i,j)$ and weighted by $idf(j)$ where $tf(i,j)$ is the number of words j that appears in the image i divided by the number of words in the image i

CABoost Algorithm

Input:

F: contingency table crossing images and visual words

$Size$: sampling size

T: number of samplings

Output:

Z: new representation of images

Algorithm:

For $i = 1$ **to** T **do**

1 Sample $Size$ columns (words) from F:
S = Sample(F, $Size$)

2 Apply ***FCA*** on the sample S and obtain the new representation of images $Z^{(i)}$ by formula (1):
$Z^{(i)}$ = ***FCA***(S)

3 Normalize $Z^{(i)}$ by dividing by the square root of the first eigenvalue:
$Z^{(i)} = \frac{1}{\sqrt{\lambda_1^{(i)}}} Z^{(i)}$

End For

4 Stack $Z^{(i)}$ by column
$A = [Z^{(1)}\ Z^{(2)}\ \ldots\ Z^{(T)}]$

5 Apply ***PCA*** on A
Z = ***PCA***(A)

Return Z

Table 1. CABoost Algorithm

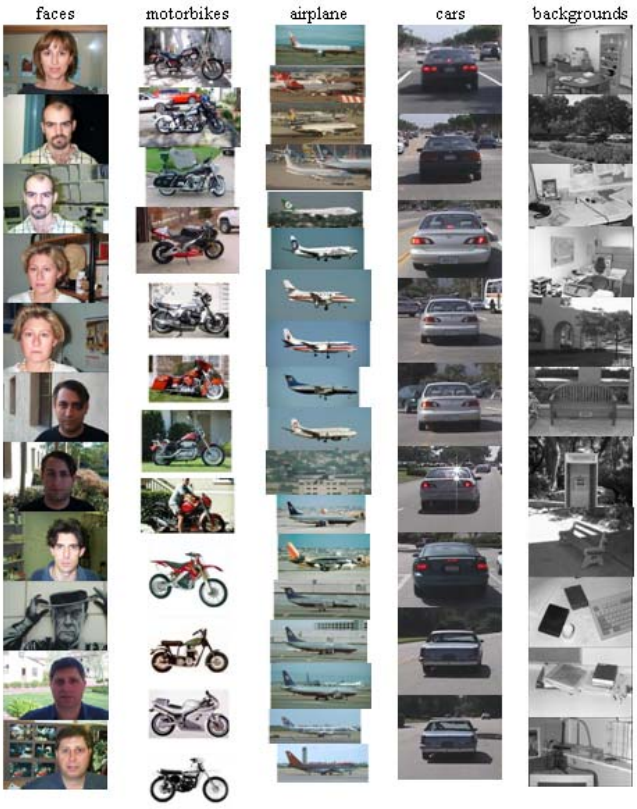

Fig. 3. Images drawn from the Caltech4 database

and $idf(j) = ln(M/M_j)$ where M_j is the number of images that contain the word j, and M is the number of images in the database. The cosine distance is usually used for similarity measure.

5.3. FCA for image retrieval

After applying FCA on the contingency table, we keep only K first axes and use them for similarity measure computation. We experiment with both distances: Euclidean and cosine and we find that cosine distance gives better result than Euclidean one. We take K equal to 20 for all of experiments.

Category	Number of images
faces	435
motorbikes	800
airplanes	800
backgrounds	900
cars (rear)	1155
Total	4090

Table 2. Description of the Caltech4 database

5.4. Boosting of FCA

There are some parameters that can influent the performance of the CABoost. Intuitively it can be considered that the larger sampling size is, the better result we obtain, and the more time it takes. We recommend to take the sampling size equal to $1/4$ the size of vocabulary and to iterate about 15 – 20 times. The number of iterations is also a factor that affects the result and the training time. The greater this number is, the better result we get. We experiment with sampling size equal to 500, and 20 iterations, the result is given in figure 4. In this test, CABoost is 3 times faster than the classical FCA and gives an equivalent result.

To study the impact of parameter T (number of iterations) we fix the sampling size equal to 500 and try with T equal to 1, 3, 10, and 30. Training time comparison is also shown in the figure 5 and precision - call curves are shown in the figure 6. It is clear that when the number of iterations increases, the performance is improved and training time increases too. In this experiment after 30 iterations, CABoost gives better result than classical FCA.

Table 3 shows the precision of two approaches: deterministic and random at 5, 10, 20, 50 and 100 first returned images. We find that the random approach takes more advantages than deterministic one. Because it is possible to combine any word

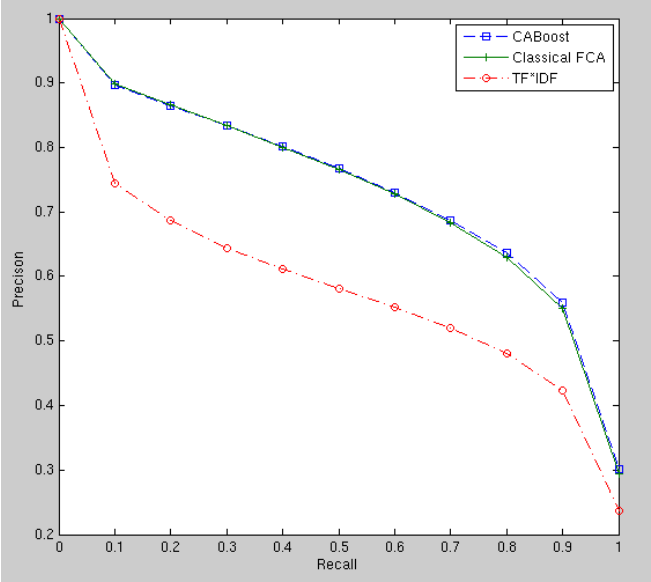

Fig. 4. Precision - recall curves for performance comparison: CABoost with sampling size = 500, T (number of iterations) = 20

#images	Deter.	Random 1	Random 2	Hybrid
5	0.951	0.958	0.963	**0.965**
10	0.937	0.943	0.949	**0.952**
20	0.924	0.928	0.936	**0.938**
50	0.903	0.906	0.915	**0.915**
100	0.885	0.886	**0.896**	**0.896**

Table 3. Comparison of two approaches: deterministic and random – #images: number of first returned images; Deter.: deterministic approach; Random 1: sampling uniformly without updating the word distribution; Random 2: under weighting all of words selected in previous steps; Hybrid: under weighting only words with high contribution in previous steps.

into group while in the deterministic way, words are grouped by their frequency. The hybrid approach slightly improves the result because of under weighting of these words with high contribution in previous steps. The remained words can contribute to form other topics.

6. CONCLUSION AND FUTURE WORKS

We have presented in this paper a new approach for image retrieval using SIFT descriptor and the adaptation of FCA on images. We propose also our new algorithm, CABoost, which can deal with large vocabulary. The experimentations have shown that CABoost performs faster and can give better results than classical FCA. In addition in all cases, FCA (classical and boosting of FCA) performs much better than TF*IDF.

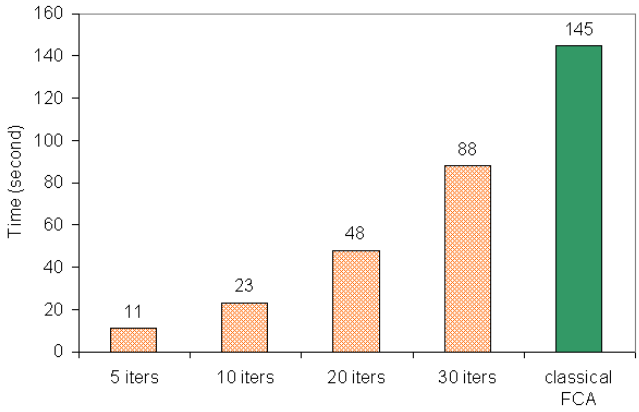

Fig. 5. Comparison of training time: CABoost (4 first columns) with sampling size set to 500 and number of iterations equal to 5, 10, 20 and 30 versus classical FCA

The sequences of FCA in CABoost algorithm are independent (in the case of uniform sampling), therefore it is possible to use parallel computations to accelerate the training rate. The merging of results in the section 4.2 could be extended to a system of multiple search engines where each search engine corresponds to a step of CABoost. For a query, we ask all the T engines to get the first images (e.g. 500 first images). And then these images are sorted by their distance to the query defined in the formula 4. Another possible improvement is to combine our method and other methods as CDM (Contextual Dissimilarity Measure) [18], and/or random forest [23] for dealing with massive data.

7. REFERENCES

[1] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the 7th International Conference on Computer Vision, Kerkyra, Greece*, 1999, pp. 1150–1157.

[2] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *Proceedings of IJC V*, vol. 60, no. 1, pp. 63–86, 2004.

[3] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harsman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[4] T. Hofmann, "Probabilistic latent semantic analysis," in *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI'99)*, 1999, pp. 289–296.

[5] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd International Conference on Research and Development in Information Retrieval (SIGIR'99)*, 1999, pp. 50–57.

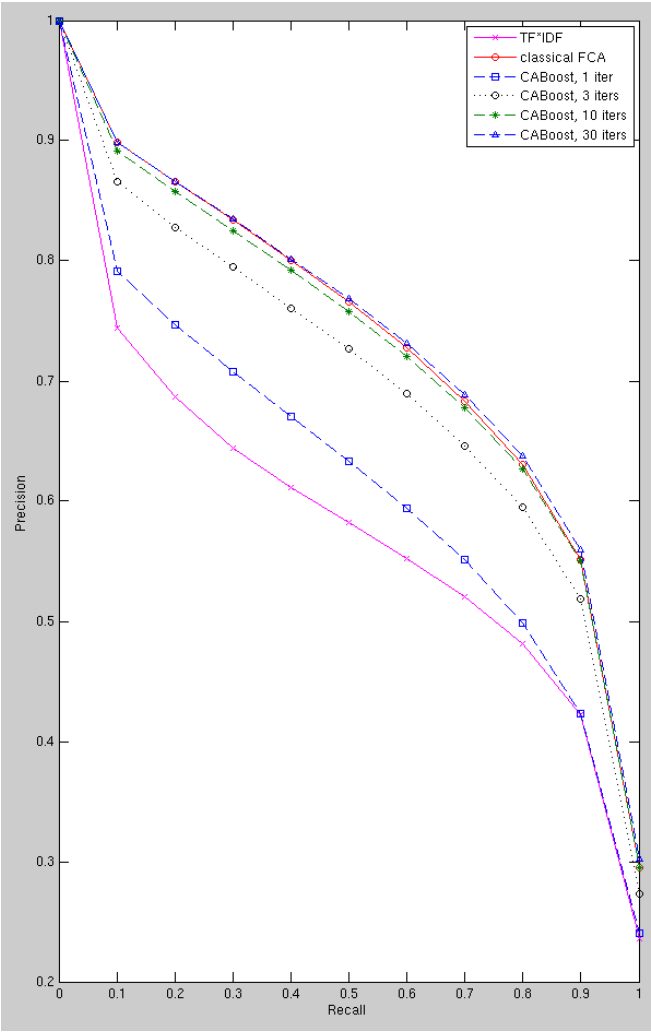

Fig. 6. Precision - recall curves: impact of parameter T (number of iterations): increasing the number of iterations causes an improvement the result

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[7] A. Bosch, A. Zisserman, and X. Munoz, "Scene classification via pLSA," in *Proceedings of the European Conference on Computer Vision*, 2006.

[8] R. Lienhart and M. Slaney, "plsa on large scale image databases," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2007, pp. 1217–1220.

[9] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman, "Discovering objects and their location in image collections," in *Proceedings of the International Conference on Computer Vision*, 2005, pp. 370–377.

[10] J. Willamowski, D. Arregui, G. Csurka, C. Dance, and L. Fan, "Categorizing nine visual classes using local appearance descriptors," in *Workshop Learning for Adaptable Visual Systems (ICPR 2004) Cambridge, United Kingdom*, 2004.

[11] A. Morin, "Intensive use of correspondence analysis for information retrieval," in *Proceedings of the 26th International Conference on Information Technology Interfaces, ITI2004*, 2004, pp. 255–258.

[12] N.-K. Pham and A. Morin, "Une nouvelle approche pour la recherche d'images par le contenu," in *Revue des Nouvelles Technologies de l'Information - Serie Extraction et gestion des connaissances*, 2008, vol. RNTI-E-11, pp. 475–486.

[13] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.

[14] J. P. Benzecri, *L'analyse des correspondances*, Paris: Dunod, 1973.

[15] M. J. Greenacre, *Correspondence analysis in practice, Second edition*, Chapman and Hall, 2007.

[16] T. Lindeberg, "Feature detection with automatic scale selection," *International Journal of Computer Vision*, vol. 30, no. 2, pp. 79–116, 1998.

[17] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," in *International Journal of Computer Vision*, 2004, pp. 91–110.

[18] H. Jegou, H. Harzallah, and C. Schmid, "A contextual dissimilarity measure for accurate and efficient image search," in *Proceedings of CVPR'07*, 2007, pp. 1–8.

[19] B. Escofier and J. Pages, *Analyse factorielles simples et multiples: objectifs, methods et interpretation*, Dunod, 1995.

[20] J. C. Gower, "Generalized procrustes analysis," *Psychometrika*, vol. 40, no. 1, pp. 33–51, 1975.

[21] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2003, vol. 2, pp. 264–271.

[22] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999.

[23] L. Breiman, "Random forest," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.