Factorial Correspondence Analysis for Image Retrieval

Nguyen-Khang Pham* †, Annie Morin†, Patrick Gros†, Quyet-Thang Le*

*Cantho University
Cantho, Vietnam
{pnkhang, lqthang}@cit.ctu.edu.vn

†IRISA
Rennes, France
{amorin, pgros}@irisa.fr

***Abstract*—We are concerned by the use of Factorial Correspondence Analysis (FCA) for image retrieval. FCA is designed for analyzing 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. The method was tested on the Caltech4 and Stewénius and Nistér datasets on which it provides better results (quality of results and execution time) than classical methods as tf*idf [20] or Probabilistic Latent Semantic Analysis (PLSA). To scale up and improve the quality of research, we propose a new retrieval schema using inverted files based on the relevant indicators of Correspondence Analysis (the quality of representation and contribution to inertia). The numerical experiments show that our algorithm performs more rapidly than the exhaustive method without losing precision.**

Bag of words, Content based Image Retrieval, Factorial Correspondence Analysis, Inverted file, SIFT

I. INTRODUCTION

The use of local descriptor in images has shown to be a good choice for image analysis. There are many successful applications using local descriptors such as: image recognition, image classification and image retrieval. Recently, the methods developed initially for Textual Data Analysis such as LSI/LSA (Latent Semantic Analysis) [7], PLSA (probabilistic Latent Semantic Analysis) [10, 11], LDA (Latent Dirichlet Allocation) [5] have been used for image analysis, e.g. image classification [22], image topic discovery [21], scene classification [6] and image retrieval [13]. These methods try to model the corpus and to reduce dimensions. Among the disadvantage of these methods, we find the use of an ad hoc model and an EM algorithm to find a local optimum and the difficulty to interpret the results. Most of the works use such methods as black box.

Here, we focus on the use of Factorial Correspondence Analysis (FCA) for the retrieval of images. Given an image query, the system must return the most similar images (in the image collection) to the query. FCA reduces the space representing images and defines the similarity among images in a smaller space.

To deal with large databases, many techniques have been developed. Most of them suffer from high dimension problem due to the curse of dimensionality. For overcome this problem we propose a new retrieval schema using inverted files constructed from relevant indicators of Correspondence Analysis.

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

II. METHODES

A. *Probabilistic Latent Semantic Analysis*

Proposed by Thomas Hofmann, PLSA is a statistical technique for the analysis of contingency tables. PLSA is evolved from Latent Semantic Analysis (LSA) which is a purely geometric method mapping documents to a reduced vector space, so-called latent semantic space. The mapping is restricted to be linear and is based on Singular Value Decomposition of the co-occurrence table. In contrast of LSA PLSA is based on a decomposition of mixtures derived from a latent variable model for co-occurrence data which associates an unobservable variable $z \in \boldsymbol{Z} = \{z_1, z_2, \ldots, z_K\}$ with each observation. The joint probability $P(d,w)$ over documents and words is defined by the mixture:

$$P(d,w) = P(d)P(w|d), P(w|d) = \sum_z P(w|z)P(z|d)$$

The log of likelihood of the corpus is computed by the formula:

$$L = \sum_{d \in D} \sum_{w \in W} F(d,w) \log(P(d,w))$$

where

$\boldsymbol{D} = \{d_1, d_2, \ldots, d_M\}$: set of documents;

$\boldsymbol{W} = \{w_1, w_2, \ldots, w_N\}$: vocabulary;

$\boldsymbol{F}$: contingency table.

The model is fitted by an Expectation Maximization (EM) algorithm. EM alternates two steps:

(i) Step E:

$$P(z|d,w) = \frac{P(z)P(d|z)P(w|z)}{\sum_{z'} P(z')P(d|z')P(w|z')}$$

(ii) Step M:

$$P(d|z) = \frac{\sum_{w} F(d,w)P(z|d,w)}{\sum_{d',w} F(d',w)P(z|d',w)}$$

$$P(w|z) = \frac{\sum_{d} F(d,w)P(z|d,w)}{\sum_{d,w'} F(d,w')P(z|d,w')}$$

$$P(z) = \frac{1}{R}\sum_{d,w} F(d,w)P(z|d,w), R \equiv \sum_{d,w} F(d,w)$$

B. *Factorial Correspondence Analysis*

FCA is a classical exploratory method for analysis of contingency tables. It was proposed by J. P. Benzécri [4] 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. FCA produces a visual representation of the relationships between the row categories and the column categories in the same 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 [9, 18]. We now briefly describe the method:

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

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

and note

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

$$P = \begin{pmatrix} p_1 & & & 0 \\ & p_2 & & \\ & & \ddots & \\ 0 & & & p_M \end{pmatrix} \quad Q = \begin{pmatrix} q_1 & & & 0 \\ & q_2 & & \\ & & \ddots & \\ 0 & & & q_N \end{pmatrix}$$

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

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

$$\lambda = \begin{pmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_N \end{pmatrix} \quad \mu = \begin{pmatrix} \mu_{11} & \mu_{12} & \cdots & \mu_{1N} \\ \mu_{21} & \mu_{22} & \cdots & \mu_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{N1} & \mu_{N2} & \cdots & \mu_{NN} \end{pmatrix}$$

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:

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

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

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

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

$$Z_r = \hat{r}A \quad where \quad \hat{r}_i = \frac{r_i}{\sum r_j} \tag{3}$$

III. IMAGE REPRESENTATION

In order to adapt textual methods (e.g. tf*idf, PLSA, FCA) on images, we must first represent the image corpora in the form of contingency table. Here images are treated as documents and the "visual words" (to be defined) as terms/words.

The 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. The 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 [14], or 3D local extremas of Difference of Gaussian [15], or the points detected by a Hessian-Affine detector [17]. Then, the descriptor of the interest points is computed on the gray level gradient of the region around the point. The scalable invariant feature transform descriptor, SIFT [16] is usually preferred. Each SIFT descriptor is a 128-dimensions vector. The second step is to form visual words from the local descriptors computed in the previous step. Most of works perform a k-means on descriptors and take the averages of each cluster as visual word [6, 21, 22]. After building the visual vocabulary, each descriptor is assigned to the nearest cluster. For this ends, we compute, in $\boldsymbol{R}^{128}$, distances from each descriptor to the representatives of previously defined clusters. An image is then characterized by the frequency of its descriptors. The image corpus will be represented in the form of a contingency table crossing images and clusters.

In our experiments, we used the method described in [17] to detect interest points. The vocabulary is built using a k-means algorithm. For Caltech4 database, about 300000 descriptors drawn randomly (one third for each category: faces, motorbikes, airplanes, cars and background) are used for word construction. The vocabulary obtained consists of 2224 words from 4090 images. The number of words in the vocabulary was chosen by Sivic [21].

Figure 1. Interest points detected by Hessian-Affine detector

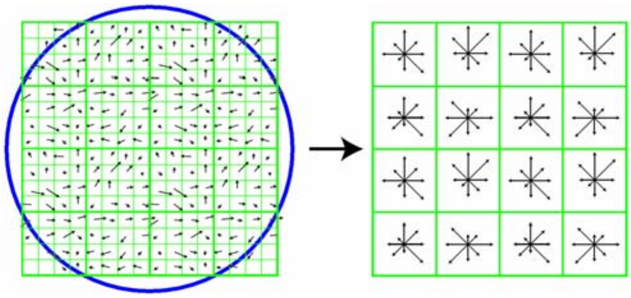

Figure 2. A SIFT descriptor computed from the region around the interpret point (the circle): gradient of the image (left) and it descripor (right)

IV. FCA FOR IMAGE RETRIEVAL

A. *FCA for reduction of dimension*

One of advantages of FCA is to reduce the dimensionality of the problem. A tree structure based indexing as a kd-tree [2] is desirable after dimension reduction by the AFC. However, such a structure would become inefficient when the number of dimensions is greater than 16 because of the curse of dimensionality [3]. In addition tree based indexing often uses the Euclidean distance for nearest neighbors search. So we encounter some difficulties when working with other distances.

B. *Advanced search with FCA*

In the context of ***k*** nearest neighbors query, many approaches have been developed to overcome the curse of dimensionality. They are usually classified in five categories: (1) tree based indexing; (2) space-filling curves; (3) dimension reduction; (4) approximate algorithms and (5) filtering-based (i.e. approximation) approaches. Our approach is based on the last category. The filtering-based approaches filter the points so that only a small fraction of the database is scanned during a search. The main idea of our approach is that two similar images share certain common proprieties. Given a query the images which share nothing with the query will be filtered.

There are two important indicators for the interpretation and evaluation of FCA. These are contribution to the inertia of an axis (factor) of images on the one hand and the quality of representation of images on an axis on the other. We will use these indicators as relevant properties for filtering. This will reduce the number of images to be considered when calculating their similarity with the query and sometimes improves the quality of results. In fact, we associate two inverted files with each axis (one for positive part and another for negative part). Each part of an axis corresponds to a relevant propriety that allows distinguishing images. The definition of inverted files is given in the following.

1) *Definition 1 (contribution):* the contribution of the image *i* to the inertia of the axis *j* is defined by:

$$Cr_j(i) = p_i \frac{Z_{ij}^2}{\lambda_j}$$

where

p_i: mass of the image *i*;

λ_i: i^{th} eigenvalue;

Z_{ij}: coordinate of the image i on the axis j.

2) *Definition 2 (representation quality):* the representation quality of the image *i* on the axis *j* is the square of cosine of the angle between the vector which joints the gravity center G to the point *i* and the axis *j*:

$$Cos_j^2(i) = \frac{d_j^2(i,G)}{d^2(i,G)} = \frac{Z_{ij}^2}{\sum_j Z_{ij}^2}$$

3) *Definition 3 (contribution based inverted file):* given a threshold ε > 0, the two contribution based inverted files associated to the axis *j* are defined by:

$$CF_j^+ = \{i | Cr_j(i) > \varepsilon \quad and \quad Z_{ij} > 0\}$$
$$CF_j^- = \{i | Cr_j(i) > \varepsilon \quad and \quad Z_{ij} < 0\}$$

4) Definition 4 (representation quality based inverted file): given a threshold ε > 0, the two representation quality based inverted files associated to the axis j are defined by:

$$QF_j^+ = \left\{ i \middle| Cos_j^2(i) > \varepsilon \quad and \quad Z_{ij} > 0 \right\}$$
$$QF_j^- = \left\{ i \middle| Cr_j^2(i) > \varepsilon \quad and \quad Z_{ij} < 0 \right\}$$

5) Image retrieval algorthim using inverted files:
The query is folded-in using formula (3). We then choose some relevant proprieties and take their associated inverted file. The inverted files are merged to form a list of candidate images. Finally, the k nearest neighbors of the query are searched in the list of candidates. The algorithm is given in table I.

In this algorithm, there are two parameters to tune: the number of inverted files to take n, in step 3 and *n_thres* in step 5. A naïf solution is that we take any odd number for n and set *n_thres* to $\lceil n/2 \rceil$ (majority vote in filtering step).

We propose here a heuristic which can be used to automatically determine n and *n_thres* depending on the image query. This heuristic is based on the following observation: "if a point is well represented on some axes, the representation quality on these axes will be great and the representation quality on the other axes will be small because the sum of representation quality is equal to one". So, we can take n first axes such that their representation quality greater than the threshold ε in the phase of construction of inverted files and/or the sum of their representation quality is greater than a threshold α (e.g. α = 0.75). To determine *n_thres*, we base on the fact that if *n_thres* is too great the constraint is too restricted and the list of candidate images can be empty. If *n_thres* is too small the number of images in the list will be great. The respond time will augment. So *n_thres* should be the greatest integer such that the number of images in the list of candidate images is greater than a given number (e.g. 500).

V. NUMERICAL RESULTS

We implemented the FCA algorithm in C++ using CLAPACK library [1] for matrix manipulation and eigen problem solver. The tests were realized on two datasets: Caltech4 [21] and Stewénius and Nistér [19].

A. Datasets

1) Caltech4 dataset
The Caltech4 image database contains 4090 images extracted from the Caltech11 database [8] distributed into 5 categories. The size of vocabulary is 2224. Table II describes the database.

2) Stewénius and Nistér dataset
The set consists of 2550 groups of 4 images each which give 10200 images in total. All the images are 640x480. Fig. 5 draws some images in this database. We experimented with a 5000 visual words constructed from a subsample of SIFT descriptors extracted from the Corel image database.

TABLE I. RETRIEVAL ALGORITHM USING INVERTED FILES

Input:
q: query
k: number of nearest neighbors

Output:
k nearest neighbors of q

Algorithm:
1. Project q in factor space (fold in query), formula (3) with $r \equiv q$
2. Sort the axes by contribution (or representation quality)
3. Choose n first axes and take their associated inverted file
4. Merge the inverted files ➔ L
5. Filter images to form a list of candidate images C
   C = ∅
   For each image i in L do
   If i appears at least *n_thres* times in L
   C = C ∪ {i}
6. Find k nearest neighbors of q in C
7. Return k nearest neighbors of q

TABLE II. CALTECH4 DATABASE

Category	Number of images
Face	435
Airplane	800
Background	900
Car(rear)	1155
Total	4090

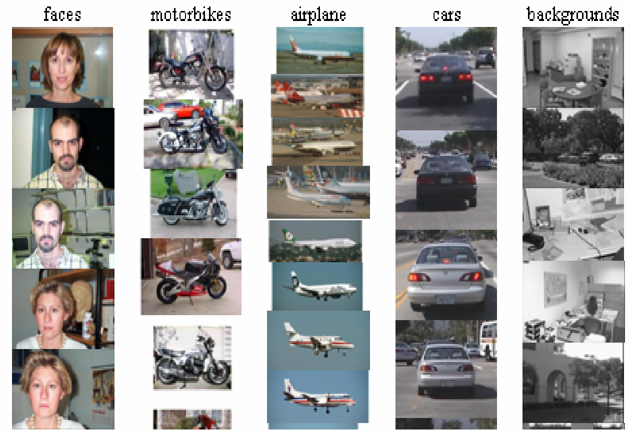

Figure 3. Images drawn from the Caltech4 dataset

B. FCA versus other methods

*1) tf*idf:* this method performs a matrix transformation. Each element of the co-ocurrence table is normalized to $tf(i,j)$ and weighted by $idf(j)$ where $tf(i,j)$ is number of times the word j appears in the document j divised by the length of the document j; and $idf(j) = \log(N/N_j)$ with Nj is the number of documents containing the word j and N is the number of documents in the corpora. The *tf*idf* weighting scheme is often used in the vector space model together with cosine similarity to determine the similarity between two documents. In our experiments, we use cosine similariy to compute the similarity of query and the images in databas The technique of inverted file in [19] is also used for accelerating the search.

2) *PLSA:* We used a PLSA model with 7 modalities which gives the best results on the base Caltech4 [19]. Each image in the database is represented by its distribution $P(z|d)$. The dimension of the problem is reduced to 7. The class specific distribution $P(z|d)$ are used to compute the similarity between the query and images in the database.

3) *FCA:* to compare with PLSA we kept only 7 first axis after having applied FCA on images. The projection of images on some first axes is drawn in Fig. 5.

4) *Discussion:* the number of topics (PLSA), number of axes kept (FCA), are parameters to tune. It is difficult to choose because the eigenvalues of FCA decrease slowly. The number "7" is chosen for comparison to PLSA. The performance augments when we keep more axes (e.g. 15, 30 axes). Fig. 6 shows the performance (precision – recall curves) of different methods. It is clear that FCA performs better other methods with cosine similarity measure. In all of the cases, PLSA and FCA give better result than *tf*idf*.

Figure 4. Images drawn from Stewénius and Nistér image database

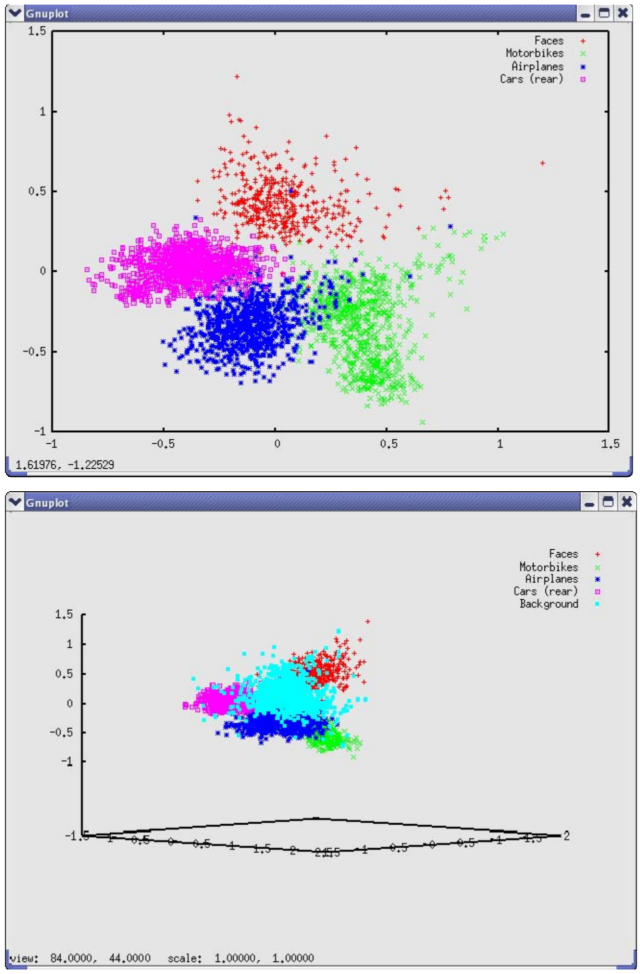

Figure 5. Projection of image in the Caltech4 database on some first axes after FCA: 4 categories (top), 5 categories (bottom).

C. *FCA with inverted file*

1) *Caltech4 dataset*

To compare the performance of the new retrieval method with the exhaustive method (scan entirely database), we computed the precision on 5, 10, 50 and 100 top images returned. The number of axis kept, parameter K, is set to 7, 15 and 30. Cosine similarity is used to measure similarity between two images. The threshold for inverted file is set to ¼ mean of contribution in the case of contribution based inverted files and is set to ¼ mean of representation quality in the case of representation quality based inverted file; *n* is set to 3 and is computed automatically according to the heuristic described in the section B.5 above; an image will be added in the list of candidates if it appears at least in $n\_thres = \lceil n/2 \rceil$ inverted file (majority vote). The results are shown in table III. The "#imgs" column describes the average size of the candidate list. In this experiment, it is shown that the new algorithm performed about 4 times more rapidly than the exhaustive one losing less than 1% in term of precision.

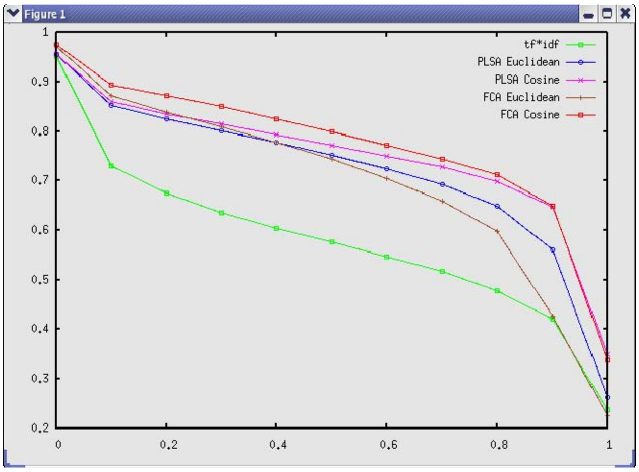

Figure 6. Precision – Recall curve, performance comparison of tf*idf, pLSA and FCA with euclidean distance and cosine similarity.

TABLE III. PERFORMANCE COMPARISION ON 5, 10, 50 AND 100 TOP RETURNED IMAGES (CALTECH4 DATABASE). TIME IS THE RESPOND TIME (MILLISECOND) PER IMAGE

Methods	#imgs	5	10	50	100	Time
(1) K=7	-	94.87	93.48	90.54	89.08	0.42
(1) K=15	-	95.92	94.61	**91.35**	**89.64**	0.50
(1) K=30	-	**96.30**	**95.03**	91.22	89.23	0.66
(2) K=7	836	94.62	93.30	90.30	88.70	**0.11**
(2) K=15	984	95.97	94.63	91.23	89.43	0.15
(2) K=30	1137	96.29	94.97	91.05	88.87	0.22
(3) K=7	888	94.55	93.23	90.24	88.76	0.12
(3) K=15	791	95.85	94.47	91.01	88.93	0.13
(3) K=30	**741**	96.18	94.88	90.38	87.05	0.15
(4) auto 1	783	95.94	94.58	91.23	89.41	0.13
(5) auto 2	922	95.82	94.52	91.12	89.23	0.14
(6) tf*idf	-	88.24	84.81	77.52	73.72	2.94

(1). Exhaustive method, scan entirely database

(2). Representation quality based inverted file method, n is fixed to 3

(3). Contribution based inverted file method, n is fixed to 3

(4). Representation quality based inverted file method, n is computed automatically (n ≈ 4.6) and K = 15

(5). Contribution based inverted file method, n is computed automatically (n ≈ 2.8) and K = 15

(6). tf*idf method

2) *Stewénius and Nistér database*

For this database, we compute the precision on 4 top returned images and use it for performance comparison. The number of axes kept is set to 200. Table IV shows the results of FCA, FCA with inverted files and tf*idf methods. We accelerated about 10 times (compare to exhaustive method and 30 times compare to tf*idf) without losing precision and even improved the result.

TABLE IV. PERFORMANCE COMPARISION ON STEWÉNIUS AND NISTÉR DATABASE

Methods	#imgs	Perf. (%)	Time (millisecond)
FCA	10200	79.82	12.01
Advance FCA *n* = 21 *n_thres* = $\lceil n/2 \rceil$ = 11	650	79.75	1.04
Advance FCA *n: auto* (~36) *n_thres* = $\lceil n/2 \rceil$	**332**	79.62	**0.99**
Advance FCA *n*: auto (~36), (sum > 0.75)[a] *n_thres*: auto	377	79.72	1.03
Advance FCA *n*: auto (~55) (sum > 0.85) *n_thres*: auto	397	79.96	1.33
Advance FCA *n*: auto (~70), (sum > 0.9) *n_thres*: auto	407	**80.01**	1.55
tf*idf	10200	73.04	36.45

a. *n* is computed by the heuristic in section IV.B.5 with sum of representation quality of axes > 0.75

VI. CONCLUSION

We have presented in this paper the use of Factorial Correspondence Analysis for content based image retrieval. We also proposed a new algorithm for improving the search quality (respond time and precision). The numerical experiments have shown that FCA gave much better results than tf*idf and slightly better PLSA. The new retrieval algorithm using inverted files improves considerable the respond time without losing precision of result.

While studying the impact of parameter *n_thres* we found that with a size of 1/100 database the list of candidates contains about 90% relevant images. It means that we can achieve a precision of 90% by scanning only 1/100 database if an appropriate similarity measure is used. This motivates us to plan to combine our indexing technique and a relevant similarity measure as CDM (Contextual Dissimilarity Measure) [12] in future works.

ACKNOWLEDGMENT

We thank A. Zisserman for data of Caltech4 and H. Jegou for preparing data of Stewénius and Nistér database.

REFERENCES

[1] E. Anderson, Bai Z., C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, LAPACK Users' Guide, Third Edition, Society for Industrial and Applied Mathematics, 2005.

[2] R. Bellman, Adaptive Control Processes: A Guided Tour, Princeton University Press, 1961.

[3] J. L. Bentley, Multidimensional binary search trees used for associative searching, Communications of the ACM, vol. 18, n° 9, pp. 509-517, 1975.

[4] J. P. Benzécri, L'analyse des correspondances, Paris: Dunod, 1973.

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

[6] A. Bosch, A. Zisserman, and X. Munoz, Scene classification via pLSA, In Proceedings of the European Conference on Computer Vision, 2006.

[7] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harsman, Indexing by latent semantic analysis, Journal of the American Society for Information Science, 1990.

[8] 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, pp.264 – 271, 2003.

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

[10] T. Hofmann, Probabilistic latent semantic analysis, In Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI'99), 1999.

[11] T. Hofmann, Probabilistic latent semantic indexing, In Proceedings of the 22nd International Conference on Research and Development in Information Retrieval (SIGIR'99), 1999.

[12] H. Jegou, H. Harzallah and C. Schmid, A contextual dissimilarity measure for accurate and efficient image search, In Proceeding of the CVPR'07, 2007.

[13] R. Lienhart and M. Slaney. pLSA on large scale image databases, In IEEE International Conference on Acoustics, Speech and Signal Processing, 2007.

[14] T. Lindeberg, Feature detection with automatic scale selection. International Journal of Computer Vision, vol. 30, pp.79–116, 1998.

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

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

[17] K. Mikolajczyk and C. Schmid, Scale and affine invariant interest point detectors, In Proceedings of the IJC V, vol. 60, pp. 63–86, 2004.

[18] A. Morin, Intensive Use of Correspondence Analysis for Information Retrieval, In Proceedings of the 26th International Conference on Information Technology Interfaces, ITI2004, pp. 255-258, 2004.

[19] D. Nistér and H. Stewénius, Scalable recognition with a vocabulary tree, In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 2161-2168, June 2006.

[20] J. Rocchio, Relevance feedback in information retrieval, In: G. Salton (ed.). The Smart retrieval system: experiments in automatic document processing, Prentice Hall, 1971, pp. 313-323.

[21] 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.

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