

# Nouvelle approche pour la recherche d'images par le contenu

Nguyen-Khang PHAM\*\*\*, Annie MORIN\*

\* IRISA, Campus de Beaulieu, F - 35042, Rennes Cedex  
{pnguyenk,amorin}@irisa.fr  
<http://www.irisa.fr>

\*\* Université de Cantho, Campus III, 1 Ly Tu Trong, Ville de Cantho, Vietnam  
pnkhang@cit.ctu.edu.vn  
<http://www.cit.ctu.edu.vn>

**Résumé.** On utilise l'analyse factorielle des correspondances (AFC) pour la recherche d'images par le contenu en s'inspirant directement de son utilisation en analyse des données textuelles (ADT). L'AFC permet ici de réduire les dimensions du problème et de sélectionner des indicateurs pertinents pour la recherche par le contenu.

En ADT, l'AFC est appliquée à un tableau de contingence croisant mots et documents. La première étape consiste donc à définir des « mots visuels » dans les images (analogue des mots dans les textes). Ces mots sont construits à partir des descripteurs locaux (SIFT) des images. La méthode a été testée sur la base Caltech4 (Sivic et al., 2005) sur laquelle elle fournit de meilleurs résultats (qualité des résultats de recherche et temps d'exécution) que des méthodes plus classiques comme TF\*IDF/Rocchio (Rocchio, 1971) ou pLSA (Hofmann, 1999a, 1999b).

Enfin, pour passer à l'échelle et améliorer la qualité de recherche, nous proposons un nouveau prototype de recherche qui utilise des fichiers inversés basés sur la qualité de représentation des images sur les axes après avoir fait une AFC. Chaque fichier inversé est associé à une partie d'un axe (positive ou négative) et contient des images ayant une bonne qualité de représentation sur cet axe. Les tests réalisés montrent que ce nouveau prototype réduit le temps de recherche sans perte de qualité de résultat et dans certains cas, améliore le taux de précision par rapport à la méthode exhaustive.

## 1 Introduction

L'utilisation des descripteurs locaux permet d'obtenir de bons résultats pour la reconnaissance d'images, la classification d'images et la recherche d'images par le contenu. Ces descripteurs sont robustes aux changements de contenu. Cette méthode a été proposée en 1997 par C. Schmid dans (Schmid et Mohr, 1997). Récemment, les méthodes développées originellement pour l'analyse des données textuelles (ADT) comme pLSA (probabilistic Latent Semantic Analysis) (Hofmann, 1999a), LDA (Latent Dirichlet Allocation) (Blei, 2003) sont appliquées en analyse d'images, par exemple pour la classification des images (Willamowski, 2004), la découverte des thèmes dans l'image (Sivic et al., 2005), la classifications des scènes (Bosch et al., 2006), et la recherche d'images (Lienhart et Slaney (2007)).

Dans ce travail, nous utilisons l'analyse factorielle des correspondances (AFC) pour la recherche d'images. Etant donné une image requête, le système doit retourner les images

(dans la collection des images) les plus similaires à la requête. L'AFC permet de réduire l'espace pour représenter les images et calculer la similarité entre les images dans cet espace réduit. Les deux contributions principales de cet article sont l'utilisation de l'AFC pour la recherche d'images, et la proposition d'un prototype de recherche d'images utilisant des fichiers inversés basés sur la qualité de représentation des facteurs de l'AFC.

L'article est organisé de la façon suivante : nous décrivons brièvement les méthodes pLSA et l'AFC dans la section 2. La section 3 présente la recherche d'images par l'AFC. La section 4 est consacrée aux résultats expérimentaux. Dans la conclusion, nous présentons les perspectives de ce travail.

## 2 Méthodes

### 2.1 Représentation des images

Qu'il s'agisse des méthodes comme le pLSA (probabilistic semantic latent analysis) ou l'AFC, il faut d'abord représenter le corpus sous forme d'une matrice d'occurrences  $F$  (un tableau de contingence) de dimension  $M \times N$  où  $M$  désigne le nombre de documents et  $N$  indique le nombre de différents mots apparaissant dans le corpus. Chaque case de la matrice  $F_{ij}$  décrit le nombre de fois où le mot  $j$  (indice de colonne) est observé dans le document  $i$  (indice de ligne). Une telle représentation ignore l'ordre des mots dans un document et est appelée modèle sac-de-mots (bag-of-words).

Il n'y a pas de mots au sens littéral du terme dans les images. Il faut donc les construire.

#### 2.1.1 Construction des mots visuels

Les mots dans les images, appelés mots visuels, doivent être calculés pour constituer un vocabulaire de  $N$  mots. Chaque image sera donc représentée enfin par un histogramme de mots. La construction des mots visuels se fait en deux étapes : (i) calcul des descripteurs locaux pour un ensemble d'images, (ii) classification (clustering) des descripteurs obtenus. Chaque cluster correspondra à un mot visuel. Il y aura donc autant de mots que de clusters obtenus à l'issue de l'étape (ii).

Le calcul des descripteurs locaux dans une image se fait aussi en deux étapes : il faut d'abord détecter des points d'intérêt dans l'image. Ces points d'intérêt sont, soit des maximums du Laplacien de Gaussien (Lindeberg, 1998), soit des extremums locaux 3D de la différence de Gaussien (Lowe, 1999), soit des points extraits par un détecteur Hessian-affine (Mikolajczyk, 2004). Ensuite, le descripteur de ce point d'intérêt est calculé sur le gradient des niveaux de gris dans la région autour du point. On a sélectionné des descripteurs invariants à la rotation et au changement d'échelle, les descripteurs SIFT (Lowe, 2004). Chaque descripteur SIFT est un vecteur à 128 dimensions. La seconde étape consiste à former des mots visuels à partir des descripteurs locaux calculés à l'étape précédente. La plupart des travaux effectués ont utilisé un *k-means* sur les descripteurs locaux et prend les moyennes de chaque cluster comme mots visuels (Willamowski, 2004, Sivic, 2005, Bosch et al., 2006).

Après avoir construit le vocabulaire visuel, chaque descripteur est affecté au cluster le plus proche. Pour cela, on calcule dans  $\mathbf{R}^{128}$  les distances de chaque descripteur aux représentants des clusters définis précédemment. Une image est ensuite caractérisée par la fréquence

de ses descripteurs dans chaque cluster. On obtient ainsi un tableau de contingence croisant les images et les clusters.

Dans nos expérimentations, nous avons utilisé la méthode décrite dans (Mikolajczyk, 2004) pour détecter des points d'intérêt. Le vocabulaire est construit en appliquant un *k-means* sur environ 300000 descripteurs tirés aléatoirement (un tiers pour chaque catégorie : faces, motorbikes, airplanes, background et cars). Le vocabulaire obtenu est de 2224 mots pour 4090 images. Ce choix de 2224 mots a été effectué par Sivic (Sivic et al., 2005).

## 2.2 PLSA

Introduite par Thomas Hofmann (1999a), le pLSA est une technique statistique qui s'inspire du LSA (Latent Semantic Analysis) (Deerwester, 1990) pour l'analyse des tableaux de contingence. LSA est une méthode purement géométrique qui ressemble beaucoup aux méthodes factorielles ; dans pLSA, on introduit un modèle probabiliste : la distribution des mots dans une image est considérée comme multinomiale. La méthode se base sur une décomposition des mélanges dérivée d'un modèle de variables latentes.

pLSA introduit une variable latente  $z \in \mathbf{Z} = \{z_1, z_2, \dots, z_K\}$  et modélise la probabilité jointe  $P(d, w)$  par :

$$P(d, w) = P(d)P(w|d) \quad \text{avec} \quad P(w|d) = \sum_{z \in \mathbf{Z}} p(w|z)P(z|d)$$

Le logarithme de la vraisemblance du corpus est défini par :

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

Où :

$\mathbf{D} = \{d_1, d_2, \dots, d_M\}$  : l'ensemble des documents

$\mathbf{W} = \{w_1, w_2, \dots, w_N\}$  : l'ensemble des mots.

$F$  : le tableau de contingence.

On utilise l'algorithme EM pour maximiser  $L$ .

## 2.3 Analyse factorielle des correspondances

L'AFC est une méthode exploratoire classique pour l'analyse des tableaux de contingence. Elle a été proposée par J. P. Benzécri (1973) dans le contexte de la linguistique, c'est-à-dire pour l'analyse de données textuelles. La première étude a été réalisée sur les tragédies de Racine. L'AFC sur un tableau croisant des mots et des documents permet de répondre aux questions suivantes : y a-t-il des proximités entre certains mots ? Y a-t-il des proximités entre certains documents ? Y a-t-il des liens entre certains mots et certains documents ? L'AFC comme la plupart des méthodes factorielles utilise une décomposition en valeurs singulières d'une matrice particulière et permet la visualisation des mots, et des documents dans un espace de dimension réduit. Cet espace de dimension réduit a la particularité d'avoir un nuage de points projetés (mots et/ou documents) d'inertie maximale. Par ailleurs, l'AFC fournit des indicateurs pertinents pour l'interprétation des axes comme la contribution d'un mot ou d'un document à l'inertie de l'axe ou la qualité d'un mot et/ou d'un document sur un axe (Morin, 2004).

## Nouvelle approche pour la recherche d'images par le contenu

Soit un tableau de contingence  $F = \{f_{ij}\}_{M,N}$ , de taille  $M \times N (N < M)$  on normalise  $F$  en  $X = \{x_{ij}\}_{M,N}$  par :

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

et note :

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

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

Pour déterminer le meilleur sous-espace de projection des données, on calcule les valeurs propres et les vecteurs propres de la matrice  $V = X^T P^{-1} X Q^{-1}$  de taille  $N \times N$ , où  $X^T$  est la transposée de  $X$ . On obtient alors les valeurs propres  $\lambda$  et les vecteurs propres  $u$  :

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

On ne garde que les  $K (K < N)$  premières valeurs propres les plus grandes et les vecteurs propres associés. Ces  $K$  vecteurs propres constituent une base orthonormée de l'espace réduit (appelé aussi espace des facteurs). Le nombre de dimensions du problème passe de  $N$  à  $K$ . Les documents (images) sont projetés dans le nouvel espace réduit :

$$Z = P^{-1} X A \quad \text{où} \quad A = Q^{-1} u \quad [1]$$

Dans cette formule,  $P^{-1} X$  représente les profils lignes et  $A$  est la matrice de transition associée à l'AFC. La projection des mots dans le sous-espace de dimension  $K$  est fournie par la formule suivante:

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

Un nouveau document (i.e. la requête)  $r = [r_1 \quad r_2 \quad \dots \quad r_N]$  sera projeté dans l'espace des facteurs par la formule de transition [1] :

$$Z_r = \hat{r} A \quad \text{où} \quad \hat{r}_i = \frac{r_i}{\sum r_j} \quad [3]$$

## 2.4 Mesure de similarité des images

Après avoir calculé les coordonnées des images dans le nouvel espace (i.e. représentation probabiliste dans le cas de pLSA, sous-espace pour l'AFC), nous devons définir la similarité entre une requête et les images. Plusieurs métriques pour cette mesure de similarité sont disponibles, parmi les quelles les 4 mesures suivantes :

Distance Manhattan (norme 1):  $L_1(a, b) = \sum |a_i - b_i|$

Distance euclidienne (norme 2):  $L_2(a, b) = \sqrt{\sum (a_i - b_i)^2}$

Distance cosinus :  $L_{\cos}(a, b) = \cos^{-1}\left(\frac{a \cdot b}{\|a\| \|b\|}\right)$

Divergence de Jensen-Shannon :

$$JS(a, b) = \frac{1}{2} \left( KL\left(a, \frac{a+b}{2}\right) + KL\left(b, \frac{a+b}{2}\right) \right) \quad \text{où}$$

$$KL(a, b) = \sum a_i \log\left(\frac{a_i}{b_i}\right)$$

Dans les expérimentations, nous avons utilisé la distance euclidienne et celle de cosinus.

## 3 Recherche d'images par l'AFC

### 3.1 AFC pour réduction de dimension

Un des avantages de l'AFC est de réduire la dimension du problème. Pour tenir compte du nombre d'images, il est préférable d'utiliser une structure d'indexation sous forme d'arbre comme un arbre k-d (Bentley, 1975). Cependant, une telle structure deviendra inefficace quand le nombre de dimensions est supérieur à 16 à cause de la malédiction de la dimension (Bellman, 1961). Enfin, l'indexation par l'arbre utilise la distance euclidienne la plupart du temps car on rencontre des difficultés lorsqu'on travaille avec d'autres distances.

### 3.2 Passage à l'échelle

Il y a deux indicateurs importants pour l'interprétation et l'évaluation en AFC. Ce sont la contribution des images à l'inertie d'un axe (facteur) d'une part et la qualité de représentation des images sur un axe d'autre part. Dans les expérimentations, nous avons constaté que la distance cosinus donnait de meilleurs résultats que la distance euclidienne. La distance cosinus est directement liée à la qualité de représentation des images sur les axes. Pour cette raison, nous avons proposé un nouveau prototype de recherche d'images utilisant des fichiers inversés basés sur la qualité de représentation. Cela permettra de réduire le nombre d'images à considérer lors du calcul de leur similarité avec la requête.

### 3.2.1 Fichiers inversés basés sur la qualité de représentation

**Définition 1 (qualité de représentation) :** la qualité de représentation d'un point  $i$  (correspond à l'image  $i$ ) sur l'axe  $j$  est le cosinus carré de l'angle entre l'axe  $j$  et le vecteur joignant le centre du nuage au point  $i$  :

$$qualité(i, j) = \frac{Z_{ij}^2}{\sum_{k=1}^K Z_{ik}^2}$$

où  $Z_{ij}$  est la coordonnée de l'image  $i$  sur l'axe  $j$ . Plus le cosinus carré est proche de 1, plus la position du point observé dans la projection est proche de la position réelle du point dans l'espace.

**Définition 2 (fichier inversé):** étant donné un seuil  $\varepsilon > 0$ , un fichier inversé  $F_j^+$  ( $F_j^-$ ) associé à la partie positive (négative) de l'axe  $j$  est une liste d'images ayant une qualité de représentation supérieure à  $\varepsilon$  et se trouvant dans la partie positive (négative) de l'axe  $j$ .

$$F_j^+ = \{i \mid qualité(i, j) > \varepsilon \text{ et } Z_{ij} > 0\}$$

$$F_j^- = \{i \mid qualité(i, j) > \varepsilon \text{ et } Z_{ij} < 0\}$$

Après avoir réduit la dimension des données à  $K$ , on construit, pour chaque axe, 2 fichiers inversés (un pour la partie positive et un autre pour la partie négative)<sup>1</sup>. Chaque fichier contient des images ayant une bonne qualité de représentation sur la partie (positive, négative) de l'axe associé. Le seuil  $\varepsilon$  est un paramètre qui contrôle la qualité de résultat et le temps de recherche. Dans les expérimentations, nous avons choisi un seuil égal à la moyenne de la qualité de représentation, à la moitié, et à un quart de la moyenne. Plus le seuil est grand, plus le nombre d'images dans un fichier inversé est petit, moins il faut de temps de recherche mais la qualité de résultat diminue.

### 3.2.2 Prototype de recherche

Algorithme pour rechercher une image requête  $q$  est donné dans le tableau 1.

<i>Entrée : un vecteur <math>q</math> représentant l'image requête</i>
1. Projeter $q$ dans l'espace des facteurs suivant la formule [3] et trier les facteurs par leur qualité de représentation.
2. Prendre $NF$ fichiers inversés associés à $NF$ premiers facteurs.
3. Faire l'union des fichiers inversés et filtrer les images par ces fichiers inversés : une image sera conservée si elle apparaît dans au moins $NF/2$ fichiers.
4. Calculer la distance entre la requête et les images conservées dans l'étape 3.
<i>Sortie : la liste des images les plus similaires à l'image requête</i>

TAB. 1 – Algorithme de recherche utilisant des fichiers inversés.

<sup>1</sup>Les images qui se trouvent à l'origine ne sont pas intéressantes. Elles sont souvent des fonds (backgrounds).

Le nombre d'images retrouvées après le filtrage sera beaucoup plus petit que le nombre d'images dans la base. Donc, le temps de recherche sera considérablement réduit.

Le nombre  $NF$  est choisi empiriquement. Cependant, il est préférable que  $NF$  soit impair et inférieur à  $K/2$  (pour le vote majoritaire dans l'étape de filtrage). Nous proposons une heuristique qui peut être utilisée pour déterminer  $NF$  en fonction de l'image requête. Cette heuristique se base sur l'observation suivante : si un point est bien représenté sur quelques axes, le cosinus carré sur ces axes sera grand et le cosinus carré des autres axes sera petit car la somme des cosinus carrés est égale à 1. Donc, on peut prendre  $NF$  axes tels que la somme des cosinus carrés sur ces axes est supérieure à  $\alpha$  (ex :  $\alpha = 0.75$ ).

## 4 Résultats

Nous avons implémenté l'AFC en GNU Octave (Eaton, 1995). Sur un PC Pentium 4, 512MO avec système d'exploitation Linux, le temps d'exécution pour l'AFC sur un tableau de contingence de 4090 x 2224 est d'environ 3 minutes.

### 4.1 Ensemble de tests

Les tests ont été réalisés sur la base Caltech4 (Sivic et al., 2005) tirée de la base Caltech101 (Fergus et al., 2003). Cette base contient 4090 images réparties en 5 catégories pour un vocabulaire de 2224 mots.

catégories	nombre d'images
faces	435
motorbikes	800
airplanes	800
backgrounds	900
cars (rear)	1155

TAB. 2 – Description de la base Caltech4



FIG. 1 – Des images extraites de la base Caltech4.

La base est divisée en 10 sous-ensembles, et l'évaluation est faite par validation croisée.

## 4.2 AFC vs d'autres méthodes

Nous avons fait une AFC sur les données de la base Caltech4 et avons gardé les 7 premiers axes (pour la comparaison au pLSA, avec 7 modalités). Lorsqu'on augmente le nombre d'axes conservés, on constate que les premières images retournées (i.e. les 10 premières images) sont bonnes mais que la qualité se dégrade lorsqu'on retient un plus grand nombre d'images (voir le tableau 3). La distance euclidienne et la distance cosinus sont utilisées pour calculer les similarités dans les espaces de dimension réduite. Nous avons utilisé la courbe de précision – rappel pour comparer la performance des différentes méthodes.

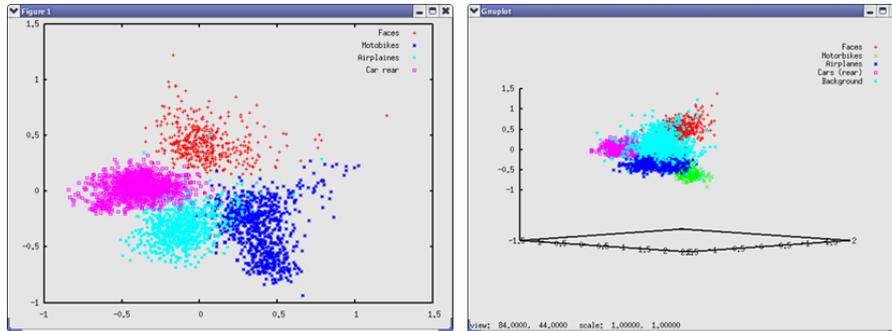


FIG. 2 – Projection de la base Caltech4 sur les axes : à gauche, projection des images (sans background) sur les axes 1 et 2 ; à droite, projection des images (avec catégorie background) sur les axes 1, 2 et 3.

### 4.2.1 TF\*IDF

Avec cette méthode, chaque élément  $F(i,j)$  dans le tableau de contingence est normalisé à  $TF(i,j)$  et pondéré par  $IDF(j)$  où  $TF(i,j)$  est le nombre de mots  $j$  qui apparaît dans l'images  $i$  divisé par le nombre de mots dans l'image  $i$  et  $IDF(j) = \ln(N/N_j)$  où  $N_j$  est le nombre d'images qui contiennent le mot  $j$  et  $N$  le nombre d'images dans la base. La distance euclidienne et la distance cosinus sont calculées sur les données pondérées.

### 4.2.2 pLSA

Nous avons appliqué un modèle pLSA avec 7 modalités (ce modèle donne les meilleurs résultats sur cette base (cf. Sivic et al., 2005)). Chaque image dans la base est représentée par sa distribution  $P(z/d)$ . La dimension du problème est réduite à 7. Les nouvelles coordonnées servent à calculer la similarité entre la requête et les images dans la base (Hofmann, 1999b).

### 4.2.3 Discussion

Le nombre de thèmes (modalités de la multinomiale) dans pLSA et le nombre d'axes conservés en AFC sont des paramètres à régler. Le nombre d'axes conservés en AFC est difficile à choisir car les valeurs propres en AFC décroissent très lentement. Nous avons pris 7 axes en AFC pour la comparaison avec pLSA. La figure 3 montre les résultats quand on fait des tests sur 4 catégories (on ne tient pas compte de la catégorie « background ») et sur 5

catégories (avec la catégorie « background »). Le meilleur résultat est obtenu avec l'AFC quelle que soit la distance, euclidienne ou cosinus. Nous avons également testé TF\*IDF avec la distance de Manhattan. Dans ce cas, on améliore le résultat mais ce n'est pas significativement supérieur aux résultats obtenus avec d'autres distances, ni avec pLSA et l'AFC.

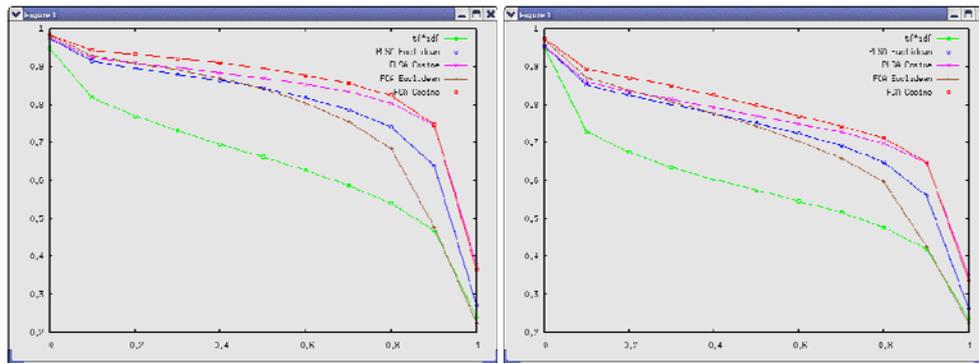


FIG. 3 – Courbe de précision – rappel : à gauche, résultat pour 4 catégories (sans background) et à droite, résultat pour 5 catégories (avec background).

### 4.3 AFC avec fichiers inversés

Pour comparer la performance de la nouvelle méthode de recherche avec la méthode exhaustive, nous avons calculé la précision sur les 5, 10, 50 et 100 premières images retournées. Le paramètre  $K$  est pris égal à 7, 15 et à 30 ; le seuil pour des fichiers inversés est mis à  $1/(4*K)$ ;  $NF$  est calé à 3. Les résultats figurent dans le tableau 3.

Méthodes	#imgs	5 imgs	10 imgs	50 imgs	100 imgs	temps
E : K = 7	-	94.87%	93.48%	90.54%	89.08%	4.24
E : K = 15	-	95.92%	94.61%	<b>91.35%</b>	<b>89.64%</b>	4.99
E : K = 30	-	<b>96.30%</b>	<b>95.03%</b>	91.22%	89.23%	6.56
N : K = 7	<b>867</b>	94.62%	93.30%	90.30%	88.70%	<b>1.10</b>
N : K = 15	<u>984</u>	95.97%	<u>94.63%</u>	<u>91.23%</u>	<u>89.43%</u>	<u>1.47</u>
N : K = 30	1137	<u>96.29%</u>	94.97%	91.05%	88.87%	2.12
TF*IDF	-	88.24%	84.81%	77.52%	73.72%	19.36

TAB. 3 – Taux de précision sur les premières images retournées,  $NF = 3$ , le seuil =  $1/(4*K)$ ; E : est la méthode exhaustive et N : est la nouvelle méthode ; #imgs : le nombre d'images restées après filtrage ; temps : le temps de réponse pour une image (millisecondes/image).

La nouvelle méthode limite le nombre d'images pour lesquelles on calcule la similarité avec la requête. Cela diminue le temps de réponse. En général, la nouvelle méthode est 4 fois plus rapide que la méthode exhaustive (on gagne 75% du temps) avec une perte inférieure à 1% du taux de précision. Par ailleurs, dans certains cas (ex.  $K = 15$ , et avec 5 premières images retournées), elle fait mieux que la méthode exhaustive. Enfin, dans tous les cas, la nou-

## Nouvelle approche pour la recherche d'images par le contenu

velle méthode est meilleure que TF\*IDF en qualité de résultat (plus de 10%) ainsi que le temps de recherche (13 fois plus rapide).

Les expérimentations suivantes montrent l'influence du seuil et celle du paramètre NF. Nous faisons varier ce paramètre avec des valeurs : 1/15, 1/30, 1/60 en fixant  $K = 15$ ,  $NF = 3$ . Les résultats sont figurés dans le tableau 4.

seuil	#imgs	5 imgs	10 imgs	50 imgs	100 imgs
1/15	<b>427</b>	95.43%	94.05%	89.71%	86.49%
1/30	<u>708</u>	<u>95.93%</u>	<u>94.55%</u>	<u>90.99%</u>	<u>88.90%</u>
1/60	984	<b>95.97%</b>	<b>94.63%</b>	<b>91.23%</b>	<b>89.43%</b>

TAB. 4 – Influence du seuil,  $NF = 3$ ,  $K = 15$ .

Evidemment quand le seuil augmente, le taux de précision augmente et le temps d'exécution augmente également.

Pour étudier l'influence du paramètre NF, nous avons pris  $K = 15$ , seuil = 1/60 et testé avec  $NF = 1$ ,  $NF = 3$ ,  $NF = 5$  et NF calculé automatiquement (cf. Section 3.1.2) pour chaque image requête.

NF	#imgs	5 imgs	10 imgs	50 imgs	100 imgs
1	1396	95.92%	<u>94.58%</u>	91.20%	89.25%
3	984	<b>95.97%</b>	<b>94.63%</b>	<u>91.23%</u>	<b>89.43%</b>
5	<u>694</u>	95.90%	94.52%	<b>91.26%</b>	<u>89.42%</u>
7	<b>502</b>	95.74%	94.45%	90.94%	88.71%
Auto (4.6)	783	<u>95.94%</u>	<u>94.58%</u>	<u>91.23%</u>	89.41%

TAB. 5 – Influence du paramètre NF,  $K = 15$ , seuil = 1/60.

Quand NF est petit (ex.  $NF = 1$ ), un seul axe ne suffit pas en terme d'information récupéré et donc le résultat n'est pas bon. Quand on prend plus d'axes (ex.  $NF = 3$ ) la qualité augmente et le filtre devient également trop contraignant (dans le cas  $NF = 7$ , une image sera gardée si elle apparaît au moins 4 fois dans les fichiers inversés). Par conséquent, cela dégrade la qualité du résultat. C'est pour cela qu'il faut choisir le paramètre NF en se basant sur la somme des cosinus carrés des NF premiers axes (cf. Section 3.1.2). Sur la base Caltech4, NF moyen est égale à 4.6 et notre proposition est justifiée a posteriori.

## 5 Conclusion et perspectives

Nous avons présenté dans cet article une nouvelle approche pour la recherche d'images par le contenu en utilisant l'AFC. Cette méthode est testée sur la base Caltech4 et comparée aux méthodes classiques comme : TF\*IDF et pLSA. Les expérimentations ont montré que dans tous les cas, l'AFC donne le meilleur résultat. Nous avons aussi proposé une nouvelle approche utilisant des fichiers inversés basés sur la qualité de représentation des images sur les axes. La nouvelle méthode réduit le temps d'exécution et dans certains cas, améliore le taux de précision par rapport à la méthode exhaustive. Comme la plupart des méthodes de réduction de dimensions, une question concerne le nombre de dimensions à conserver. Dans la littérature, on conserve souvent 100 dimensions pour LSA et 30 pour l'AFC. On peut suggérer l'utilisation de la méthode Bayésienne de (Teh et al., 2004) pour déterminer ce nombre.

Pour le passage à l'échelle, les méthodes basées sur la décomposition de la base d'images en sous-bases sont prometteuses. La base peut être décomposée en clusters et l'AFC est appliquée sur les clusters. Un algorithme heuristique sera utilisé pour sélectionner les clusters dans lesquels il faudra chercher la réponse à la requête. Une autre amélioration des résultats est de combiner l'AFC avec d'autres méthodes comme CDM (Contextual Dissimilarity Measure) de (Jegou et al., 2007) et/ou des méthodes d'apprentissage pour les rangs (Chu, 2005 ; Burges, 2005).

## Remerciements

Nous remercions A. Zisserman et ses collègues pour leurs données de Caltech4.

## Références

- Bellman, R. (1961). *Adaptive Control Processes: A Guided Tour*, Princeton University Press.
- Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517.
- Benzécri, J.-P. (1973). *L'analyse des correspondances*. Paris: Dunod
- Blei, D. M., A. Y. Ng, and M. I. Jordan (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*. 3:993–1022.
- Bosch, A., A. Zisserman, and X. Munoz (2006). Scene classification via pls. In *Proceedings of the European Conference on Computer Vision*.
- Burges, C. J. C. (2005). Ranking as Learning Structured Outputs. In *Proceedings of the NIPS 2005 Workshop on Learning to Rank*.
- Chu, W. and Z. Ghahramani (2005). Extensions of Gaussian Processes for Ranking: Semi-Supervised and Active Learning. In *Proceedings of the NIPS 2005 Workshop on Learning to Rank*.
- Deerwester, S., S. Dumais, G. Furnas, T. Landauer, and R. Harsman (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*.
- Eaton, J. W. (1995). Octave: A high-level interactive language for numerical computations. Department of Chemical Engineering, University of Wisconsin-Madison.
- Fergus, R., P. Perona, and A. Zisserman (2003). Object Class Recognition by Unsupervised Scale-Invariant Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 264 – 271.
- Hofmann, T. (1999a). Probabilistic latent semantic analysis. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI'99)*.
- Hofmann, T. (1999b). Probabilistic latent semantic indexing. In *Proceedings of the 22nd International Conference on Research and Development in Information Retrieval (SIGIR'99)*.

## Nouvelle approche pour la recherche d'images par le contenu

- Jegou, H., H. Harzallah and C. Schmid (2007). A contextual dissimilarity measure for accurate and efficient image search. In *Proc. CVPR'07*.
- Lienhart R. and M. Slaney (2007). pLSA on large scale image databases. In *IEEE International Conference on Acoustics, Speech and Signal Processing*
- Lindeberg, T. 1998. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116.
- Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the 7th International Conference on Computer Vision*, Kerkyra, Greece, 1150–1157.
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 91-110.
- Mikolajczyk, K. and C. Schmid (2004). Scale and affine invariant interest point detectors. In *Proc. IJCV*, 60(1) : 63–86.
- Morin, A. (2004). Intensive Use of Correspondence Analysis for Information Retrieval, In *Proceedings of the 26<sup>th</sup> International Conference on Information Technology Interfaces, ITI2004*, 255-258.
- Rocchio, J. (1971). Relevance feedback in information retrieval, In: G. Salton (ed.). *The Smart retrieval system: experiments in automatic document processing*, Prentice Hall, 313-323.
- Schmid, C. and R. Mohr (1997). Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:530–535.
- Sivic, J., B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman (2005). Discovering objects and their location in image collections. In *Proceedings of the International Conference on Computer Vision*.
- Teh, Y. W., M. I. Jordan, M. J. Beal, and D. M. Blei (2004). Hierarchical dirichlet processes. In *Proc. NIPS*.
- Willamowski, J., D. Arregui, G. Csurka, C. Dance, and L. Fan. Categorizing nine visual classes using local appearance descriptors (2004). *Workshop Learning for Adaptable Visual Systems (ICPR 2004) Cambridge, United Kingdom*.

## Summary

A new approach for content based image retrieval uses Factorial Correspondence Analysis (FCA), inspired from the use of FCA in textual analysis (TA). We get better results with FCA than with other classical methods such as: TF\*IDF or pLSA. We already propose a new prototype for image retrieval using inverted files based on the quality of representation of images on axes.