Utilisation de l'analyse factorielle des correspondances pour la recherche d'images à grande échelle

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

[*]IRISA, Campus de Beaulieu, F - 35042, Rennes Cedex
{pnguyenk,amorin,pgros}@irisa.fr,
http://www.irisa.fr
[**]Université de Cantho, Campus III, 1 Ly Tu Trong, Cantho, Vietnam
lqthang@cit.ctu.edu.vn
http://www.cit.ctu.edu.vn

Résumé. Nous nous intéressons à l'utilisation de l'Analyse Factorielle des Correspondances (AFC) pour la recherche d'images par le contenu dans une base de données d'images volumineuse. Nous adaptons l'AFC, méthode originellement développée pour l'Analyse des Données Textuelles (ADT), aux images en utilisant des descripteurs locaux SIFT. En ADT, l'AFC permet de réduire le nombre de dimensions et de trouver des thèmes. Ici, l'AFC nous permettra de limiter le nombre d'images à examiner au cours de la recherche afin d'accélérer le temps de réponse pour une requête. Pour traiter de grandes bases d'images, nous proposons une version incrémentale de l'algorithme AFC. Ce nouvel algorithme découpe une base d'images en blocs et les charge dans la mémoire l'un après l'autre. Nous présentons aussi l'intégration des informations contextuelles (e.g. la Mesure de Dissimilarité Contextuelle (Jegou et al., 2007)) dans notre structure de recherche d'images. Cela améliore considérablement la précision. Nous exploitons cette intégration dans deux axes: (i) hors ligne (la structure de voisinage est corrigée hors ligne) et (ii) à la volée (la structure de voisinage des images est corrigée au cours de la recherche sur un petit ensemble d'images).

1 Introduction

La recherche d'images par le contenu a pour but de trouver, dans une base d'images, les images les plus similaires à celle de la requête en utilisant des informations visuelles. Cette tâche n'est pas facile à cause de changement de vue, variation de luminosité, occlusion. Récemment, l'utilisation des descripteurs locaux a apporté de bonnes améliorations à l'analyse d'images. Contrairement aux descripteurs globaux qui sont calculés sur une image entière, les descripteurs locaux sont extraits en des points particuliers d'une image. Cela permet de trouver des images qui partagent un ou plusieurs éléments visuels seulement avec l'image requête. Initialement, les méthodes basées sur un mécanisme de vote ont été utilisées pour la recherche d'images en mettant en correspondance des points d'intérêt (Lowe, 1999; Mikolajczyk et Schmid, 2004). Les méthodes originellement développées pour l'Analyse des

Données Textuelles comme la pondération *tf*idf* (Salton et Buckley, 1988), LSA (Latent Semantic Analysis) (Deerwester et al., 1990), PLSA (Probabilistic Latent Semantic Analysis) (Hofmann, 1999), LDA (Latent Dirichlet Allocation) (Blei et al., 2003) ont été adaptées aux images (Bosch et al., 2006; Lienhart et Slaney, 2007; Sivic et al., 2005). Dans l'Analyse des Données Textuelles, ces méthodes utilisent un modèle sac-de-mots. Elles prennent en entrée une matrice de co-occurrences (appelé aussi un tableau de contingence qui croise des documents et des termes/mots) et essayent de réduire la dimension. Dans l'adaptation aux images, nous avons les images comme documents et les « mots visuels » comme termes/mots. Dans le contexte de grandes bases d'images, les méthodes comme LSA, PLSA et LDA deviennent très coûteuses en terme de temps et de mémoire utilisée. Ici, nous nous intéressons à l'utilisation de l'Analyse Factorielle des Correspondances (AFC) pour la recherche l'images à grande échelle sous trois aspects : (i) l'utilisation de l'AFC permet d'améliorer la distance entre les images par rapport à *tf*idf* et PLSA (Pham et Morin, 2008). (ii) Dans le cas d'un grand nombre d'images, nous proposons une version incrémentale de l'algorithme AFC. Pour éviter les problèmes de mémoire, on découpe la base en blocs chargés les uns après les autres en mémoire. (iii) Nous introduisons une structure de recherche très efficace pour des images en utilisant un indicateur de l'AFC : *la qualité de représentation*. Cette structure se base sur un système de fichier inversé (Sivic et Zisserman, 2003; Nistér et Stewénius, 2006) qui évite la comparaison entre la requête et toutes les images de la base. Néanmoins nos fichiers inversés ne sont pas construits directement à partir des « mots visuels » mais à partir des thèmes trouvés par l'AFC. Nous présentons aussi l'intégration de la Mesure de Dissimilarité Contextuelle (Jegou et al., 2007) dans notre structure de recherche.

Le reste de l'article est organisé comme suit : la construction des mots visuels et son utilisation pour la représentation des images sont présentées dans la section 2. Nous décrivons brièvement l'Analyse Factorielle des Correspondances dans la section 3. La section 4 est dédiée à présenter la version incrémentale de l'algorithme d'AFC. La recherche d'images à grande échelle en utilisant l'AFC est présentée dans la section 5. Nous montrons quelques résultats expérimentaux dans la section 6 avant la conclusion et les perspectives.

2 Représentation des images

Les mots dans les images, appelés mots visuels, doivent être calculés pour constituer un vocabulaire de N mots. Chaque image sera donc représentée ensuite 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 du Gaussien (Lindeberg, 1998), soit des extremums locaux 3D de la différence de Gaussien (Lowe, 1999), soit des points extraits par un détecteur Hessien-affine (Mikolajczyk et Schmid, 2004). La figure 1 montre quelques points d'intérêt déctectés par un détecteur Hessien-Affine. Ensuite, le descripteur de ce point d'intérêt est calculé sur le gradient des niveaux de gris dans la région autour du point. La plupart des applications (Sivic et al., 2005; Bosch et al., 2006) utilisent des descripteurs invariants à la rotation et au changement d'échelle, les descripteurs SIFT (Lowe, 2004). Chaque descripteur SIFT est un vecteur

FIG. 1 – *Points d'intérêt détectés par un détecteur Hessian-Affine*

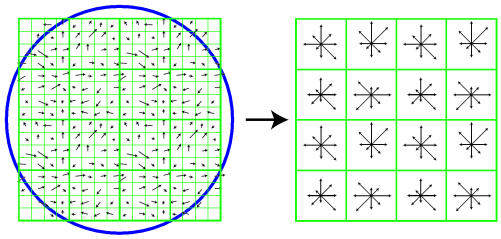

FIG. 2 – *Un descripteur SIFT calculé à partir de la région autour d'un point d'intérêt (le cercle) : gradient de l'image (à gauche), et le descripteur du point d'intérêt (à droite)*

à 128 dimensions. Le calcul des descripteurs SIFT est illustré dans la figure 2. La seconde étape consiste à former des mots visuels à partir des descripteurs locaux calculés à l'étape précédente. La plupart des travaux effectue un k-means sur les descripteurs locaux et prend les moyennes de chaque cluster comme mots visuels (Sivic et al., 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 R^{128} les distances de chaque descripteur aux représentants des clusters définis précédemment. Une image est ensuite caractérisée par la distribution des mots visuels. On obtient ainsi un tableau de contingence croisant les images et les clusters.

3 Analyse Factorielle des Correspondances

L'AFC est une méthode exploratoire classique pour l'analyse des tableaux de contingence. Elle a été proposée par (Benzecri, 1973) dans le contexte de la linguistique, c'est-à-dire pour l'analyse de données textuelles. 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; Greenacre, 2007).

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 :

$$s = \sum_{i=1}^{M}\sum_{j=1}^{N} f_{ij} \tag{1}$$

$$x_{ij} = \frac{f_{ij}}{s}\ ,\ \forall i = 1..M, j = 1..N \tag{2}$$

et note :

$$p_i = \sum_{j=1}^{N} x_{ij}\ ,\ \forall i = 1..M \qquad q_j = \sum_{i=1}^{M} x_{ij}\ ,\ \forall j = 1..N \tag{3}$$

$$P = \begin{pmatrix} p_1 & & & 0 \\ & p_2 & & \\ & & ... & \\ 0 & & & p_M \end{pmatrix} \qquad Q = \begin{pmatrix} q_1 & & & 0 \\ & q_2 & & \\ & & ... & \\ 0 & & & q_N \end{pmatrix} \tag{4}$$

Pour déterminer le meilleur sous-espace de projection des données, on calcule les valeurs propres et les vecteurs propres de la matrice V de taille $N \times N$:

$$V = X^{'} P^{-1} X Q^{-1} \tag{5}$$

où $X^{'}$ est la transposée de X.
On obtient alors les valeurs propres λ et les vecteurs propres μ :

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

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 sont projetés dans le nouvel espace réduit :

$$Z = P^{-1} X A \tag{6}$$

Dans la formule 6, $P^{-1}X$ représente les profils lignes et $A = Q^{-1}\mu$ 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 :

$$W = Q^{-1} X^{'} Z \lambda^{-\frac{1}{2}} \tag{7}$$

Un nouveau document (i.e. la requête) $r = [r_1\ r_2\ \cdots\ r_N]$ sera projeté dans l'espace des facteurs par la formule de transition (6) :

$$\hat{r}_i = \frac{r_i}{\sum_{j=1}^{N} r_j},\ \forall i = 1..N \tag{8}$$

$$Z_r = \hat{r} A \tag{9}$$

4 Algorithme incrémental pour l'AFC

Comme présenté dans la section 3, le problème de l'AFC vise à construire la matrice V (formule 5) et à chercher ses valeurs propres et vecteurs propres. Dans le cas où il y a beaucoup d'images, la matrice X est très grande. Il est donc impossible de la charger dans la mémoire. L'idée est de chercher une procédure incrémentale pour construire la matrice V.
D'abord, réécrivons la formule 5 :

$$V_0 = X^{'} P^{-1} X \tag{10}$$
$$V = V_0 Q^{-1} \tag{11}$$

Puis, la matrice X est découpée en blocs par lignes (supposons que nous avons B blocs notés par X_1, X_2, ..., X_B) :

$$X = \begin{pmatrix} X_1 \\ \vdots \\ X_B \end{pmatrix} \tag{12}$$

Ensuite, nous calculons $P_1, P_2, \ldots, P_B$ et $Q_1, Q_2, \ldots, Q_B$ de la même manière que nous calculons Q et P en replaçant X par $X_i, i = 1..B$. Il est clair que :

$$P = \begin{pmatrix} (P_1) & & & 0 \\ & (P_2) & & \\ & & \ddots & \\ 0 & & & (P_B) \end{pmatrix} et \tag{13}$$

$$Q = \sum_{i=1}^{B} Q_i \tag{14}$$

Si on note :

$$V_i = X_i^{'} P_i^{-1} X_i \tag{15}$$

alors

$$V_0 = \sum_{i=1}^{B} V_i \tag{16}$$

Les deux formules 14 et 16 nous aident à construire un algorithme incrémental pour calculer V. La projection des images Z dans la formule 6 peut être réalisée de la même manière. La version incrémentale de l'algorithme d'AFC est décrite dans le tableau 1.

5 Recherche d'images à grande d'échelle

5.1 Structure de recherche

Afin d'accélérer la recherche, nous utilisons la méthode décrite dans (Pham et Morin, 2008) pour rejeter les images non pertinentes. L'idée est que l'application de l'AFC sur des images

AFC incrémental	
1.	$Q = 0$
2.	$V_0 = 0$
3.	**pour** i = 1 **à** B
4.	charger le bloc X_i en mémoire
5.	calculer P_i, Q_i à partir de X_i (voir formules (3) et (4))
6.	Q += Q_i
7.	$V_i = X_i^{'} P_i^{-1} X_i$
8.	V_0 += V_i
9.	**fin pour**
10.	$V = V_0 Q^{-1}$
11.	calculer les valeurs propres λ et vecteurs propres μ de V
12.	garder K premiers vecteurs propres de V
13.	calculer matrice de transition $A = \mu Q^{-1}$
14.	**pour** i = 1 **à** B
15.	charger le bloc X_i en mémoire
16.	calculer P_i à partir de X_i
17.	$Z_i = P_i^{-1} X_i A$
18.	**fin pour**

TAB. 1 – *Algorithme incrémental pour AFC.*

nous permet de trouver des axes correspondant à un thème. Une image qui est bien représentée sur un axe appartient au thème associé à cet axe. Une image ne partageant aucun thème avec la requête sera considérée comme non pertinente. Le filtrage des images non pertinentes est effectué en utilisant des fichiers inversés basés sur la qualité de représentation des images selon les axes.

Définition 1 (qualité de représentation) : la qualité de représentation d'un point i (correspondant 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 :

$$Cos_j^2(i) \quad = \quad \frac{Z_{ij}^2}{\sum_{k=1}^{K} Z_{ik}^2} \tag{17}$$

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 $\epsilon > 0$, 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 ϵ et se trouvant dans la partie positive (négative) de l'axe j.

$$F_j^+ \quad = \quad \{i | Cos_j^2(i) > \epsilon \text{ and } Z_{ij} > 0\} \tag{18}$$

$$F_j^- \quad = \quad \{i | Cos_j^2(i) > \epsilon \text{ and } Z_{ij} < 0\} \tag{19}$$

Après avoir réduit la dimension des données à K, nous construisons, pour chaque axe, 2 fichiers inversés (un pour la partie positive et un autre pour la partie négative). Chaque fichier contient des images ayant une bonne qualité de représentation sur la partie (positive, négative) de l'axe associé.

Étant donnée une image requête q, la recherche commence par filtrer les images non pertinentes. Pour ce faire, d'abord nous projetons q dans l'espace réduit par la formule 8. Ensuite, nous choisissons les axes qui représentent bien la requête en triant la qualité de représentation de la requête sur ces axes et prenons les fichiers inversés associés. Puis, les fichiers inversés sont fusionnés. Enfin, nous ne gardons que les images qui partagent le plus de thèmes avec la requête. La liste contentant ces images-là est appelée la *liste de candidates*. La recherche des images les plus similaires à la requête est effectuée dans la liste des candidates dont la taille est beaucoup plus petite que celle de la base d'images. Cela diminue considérablement le temps de réponse.

Le nombre de thèmes qu'une image partage avec la requête joue un rôle important pour sa présence dans la liste de candidates. Le choix d'un seuil adapté est déterminé empiriquement. L'utilisation du vote majoritaire pour déterminer ce seuil est une solution possible (Pham et Morin, 2008). Dans ce cas, une image sera retenue dans la liste de candidates si elle partage, avec la requête, au moins une moitié du nombre de thèmes auxquels la requête appartient. Dans la recherche de type k plus proches voisins (comme utilisé dans la section 5.2 pour calculer des termes de régularisation), on souhaiterait retourner les k images les plus proches à l'image requête. Il est nécessaire pour une liste de candidates de contenir au moins k images. C'est la raison pour laquelle nous utilisons *la taille minimum de la liste de candidates*, **nb_min**, comme un critère pour rejeter les images non pertinentes.

5.2 Mesure de dissimilarité contextuelle

La mesure de dissimilarité contextuelle, (MDC ou CDM : abréviation de Contextual Dissimilarity Measure) (Jegou et al., 2007) est une intégration des informations contextuelles à la recherche par la similarité. Cette mesure prend en compte la structure de voisinage des points pour corriger la symétrie du schéma de recherche de type k plus proches voisins (k-PPV). Nous associons à chaque point un poids proportionnel à la densité de la région contenant ce point. Cette pondération favorise les points isolés et pénalise les points se trouvant dans les regions denses.

Considérons le voisinage $\mathcal{N}(i)$ d'un point i défini par $\#\mathcal{N}(i) = n_{\mathcal{N}}$ plus proches voisins de i. Nous définissons la distance de voisinage $r(i)$ comme la moyenne des distance du point i à tous les points dans son voisinage $\mathcal{N}(i)$:

$$r(i) \quad = \quad \frac{1}{n_{\mathcal{N}}} \sum_{j \in \mathcal{N}(i)} d(i,j) \tag{20}$$

où $d(i,j)$ est la distance (ou une mesure de dissimilarité quelconque) entre les points i et j. La quantité $r(i)$ est calculée pour tous les points de la base. La mesure de dissimilarité contextuelle $d_{MDC}(i,j)$ entre 2 points i et j est définie par :

$$d_{MDC}(i,j) \quad = \quad d(i,j) \left(\frac{\bar{r}^2}{r(i)r(j)} \right)^{\alpha} \tag{21}$$

où $0 < \alpha < 1$ est le facteur de lissage et $\bar{r}$ est la moyenne géométrique des distances de voisinage $r(i)$ obtenue par :

$$\bar{r} = \prod_{i=1}^{M} r(i)^{\frac{1}{M}} \tag{22}$$

La formule 21 peut se réécrire :

$$d_{MDC}(i,j) = d(i,j)\left(\frac{\bar{r}^2}{r(i)r(j)}\right)^{\alpha} \tag{23}$$

$$= d(i,j)\left(\frac{\bar{r}}{r(i)}\right)^{\alpha}\left(\frac{\bar{r}}{r(j)}\right)^{\alpha} \tag{24}$$

$$= d(i,j)\delta(i)\delta(j) \tag{25}$$

où

$$\delta(i) = \left(\frac{\bar{r}}{r(i)}\right)^{\alpha} \tag{26}$$

Les k plus proches voisins d'une requête q sont obtenus par :

$$\text{k-}PPV(q) = \text{k-}argmin_j\{d(j,q)\delta(j)\} \tag{27}$$

Les $\delta(i)$ sont appelés termes de régularisation et stockés avec la base.

5.3 Intégration de la mesure de dissimilarité contextuelle dans la structure de recherche

Quand la taille de la base d'images est grande, un inconvénient de la MDC est que le calcul des termes de régularisation implique le calcul des distances entre les points de la base. La complexité de calcul est quadratique avec le nombre d'images. Pour réduire la complexité, (Jegou et al., 2007) ont proposé une structure de recherche en groupant les images en clusters. Le voisinage d'un point i est cherché dans un nombre limite (e.g. 3, soit 1% de la base) de clusters les plus proches de i. Le nombre de clusters voisins d'un cluster donné augmente de manière exponentielle avec le nombre de dimensions. Donc, si le nombre de clusters les plus proches est trop petit, le risque devient grand de ne pas obtenir un bon voisinage et la précision est dégradée.

C'est pour cela que nous nous proposons d'utiliser notre structure de recherche se basant sur les fichiers inversés décrite dans la section 5.1 pour calculer les termes de régularisation. Notre structure de recherche permet de trouver de bons voisins d'une requête en examinant seulement un tout petit ensemble d'images (e.g. 0.05% de la taille de la base). Ainsi la complexité de calcul est diminuée et la précision est améliorée. Nous exploitons cette mesure de dissimilarité contextuelle de deux manière :

- hors ligne : les termes de régularisation $\delta(i)$ sont calculés une seule fois avant la recherche. Cette solution est appropriée aux bases d'images statiques où la mise à jour des bases est rare.
- à la volée : au cours de la recherche, nous calculons les termes de régularisation $\delta(i)$ seulement pour des images de la liste de candidates. De cette manière, le problème de mise à jour de la base n'existe plus.

FIG. 3 – *Images tirées de la base Nistér Stewénius*

6 Résultats expérimentaux

6.1 Base d'images et critère d'évaluation

Les expérimentations sont effectuées sur la base Nistér Stewénius (Nistér et Stewénius, 2006). Cette base contient 2550 groupes de 4 images (prises sur une scène avec différentes positions de l'appareil photo) et fait au total 10200 images. La figure 3 montre certaines images tirées de la base Nistér Stewénius. Pour pouvoir évaluer notre approche à grande échelle, cette base est fusionnée avec 1 million d'images téléchargées de l'internet. Les mots visuels sont calculés à partir d'une autre base pour produire un vocabulaire de 5000 mots. Nous utilisons la même mesure que celle utilisée dans (Nistér et Stewénius, 2006) pour évaluer la performance de notre méthode. C'est à dire que nous calculons la précision sur les 4 premières images retournées.

6.2 AFC sans MDC

Le tableau 2 montre les résultats quand nous testons sur la base Nistér Stewénius avec différentes tailles (10200 images « plongées » dans 100000, 200000, 500000 et 1 million d'images internet). Dans cette expérimentation, la distance utilisée est celle du cosinus. La colonne *nb_min* décrit la taille minimum des listes de candidates (voir section 5.1) ; la colonne *#list* montre la taille réelle des listes de candidates ; la colonne *#pert.* décrit le nombre d'images pertinentes dans la liste de candidates ; la précision de l'algorithme est décrite dans la colonne *préc.* et la colonne *tf*idf* présente la précision de la méthode *tf*idf* avec la structure de recherche décrite dans (Jegou et al., 2007) en examinant 5% et 10% de la base. Il est clair que notre structure de recherche fait beaucoup mieux que *tf*idf* en terme de précision ainsi qu'en terme de temps de réponse. Avec une base d'un million d'images, une liste de candidates de 600 images (0.06% de la taille de la base) contient 86.7% images pertinentes. Cela explique pourquoi notre structure de recherche réduit le temps de recherche et à la fois préserve la précision.

taille	nb_min	#liste	pert.	préc.	tf*idf (5% - 10%)
100K + 10200	200	242	0.885	0.757	
	300	366	0.904	0.761	0.678 - 0.707
	500	609	0.922	0.763	
200K + 10200	200	244	0.881	0.745	
	300	364	0.897	0.747	0.700 - 0.697
	500	608	0.914	0.748	
500K + 10200	200	244	0.858	0.728	
	300	367	0.874	0.730	0.656 - 0.682
	500	611	0.892	0.731	
1M + 10200	200	248	0.834	0.717	
	300	370	0.850	0.719	0.644 - 0.669
	500	614	0.867	0.720	

TAB. 2 – *AFC sans Mesure de Dissimilarité Contextuelle, comparaison à la méthode tf*idf avec clustering.*

6.3 AFC avec MDC

Dans le tableau 2, il est clair que le taux des images pertinentes dans la liste de candidates (colonne *pert.*) est assez élevé. Cela nous montre que la précision peut être améliorée si une mesure appropriée est utilisée. C'est aussi la raison principale pour laquelle nous essayons d'intégrer la Mesure de Dissimilarité Contextuelle dans notre structure de recherche. Le tableau 3 montre les résultats avec l'intégration de la Mesure de Dissimilarité Contextuelle à la structure de recherche à l'aide de l'AFC. La colonne *AFC(1)* montre la précision de l'intégration de l'AFC avec MDC hors ligne ; *AFC(2)* désigne la combinaison de l'AFC avec MDC calculée au cours de la recherche et la dernière colonne, *tf*idf + MDC*, décrit la précision de la méthode décrite dans (Jegou et al., 2007) qui combine MDC et clustering. Avec la méthode AFC(1), nous avons calé *la taille minimum de la liste des candidates* à 300 pour calculer les termes de régularisation, tandis qu'avec la méthode AFC(2), les termes de régularisation sont calculés pour toutes les images dans la liste de candidates dont la taille minimum, *nb_min*, est fixée à 200, 300 et 500 comme dans la colonne *nb_min* du tableau. Pour des raisons de comparaison, nous avons fixé $n_{\mathcal{N}}$ égal à 10 et α égal à 0.6 comme dans (Jegou et al., 2007). Les résultats montrent que notre structure de recherche (hors ligne et en ligne) fait beaucoup mieux que la structure de recherche basée sur le clustering. Une explication possible est que la liste de candidates dans notre structure de recherche contient de vrais plus proches voisins. Par conséquent, (i) les termes de régularisation approximatifs sont très proches des idéaux (qui sont calculés de manière exhaustive) tandis que la méthode basée sur le clustering ne donne pas de bons termes de régularisation et (ii) un nombre limite de clusters ne contient pas suffisamment d'images pertinentes pour une requête et cela dégrade le résultat.

7 Conclusion et perspectives

Nous avons présenté dans cet article l'utilisation de l'Analyse Factorielle des Correspondances pour la recherche d'images par le contenu dans une très grande base d'images. Dans un premier temps l'AFC est utilisée à la place de *tf*idf* pour améliorer le calcul des distances

taille	nb_min	#liste	AFC(1)	AFC(2)	tf*idf + MDC (5% - 10%)
100K + 10200	200	242	0.801	0.801	
	300	366	0.803	0.808	0.736 - 0.772
	500	609	0.806	0.812	
200K + 10200	200	244	0.791	0.794	
	300	364	0.793	0.798	0.727 - 0.762
	500	608	0.794	0.801	
500K + 10200	200	244	0.774	0.779	
	300	367	0.776	0.783	0.712 - 0.745
	500	611	0.779	0.786	
1M + 10200	200	248	0.758	0.766	
	300	370	0.763	0.772	0.701 - 0.732
	500	614	0.766	0.776	

TAB. 3 – *AFC avec MDC - **AFC(1)** : combinaison de AFC avec MDC hors ligne ; **AFC(2)** : AFC avec MDC à la volée et **tf*idf + MDC** : combinaison de MDC et clustering.*

entre les images de la base de données. Ensuite, pour pouvoir traiter des bases avec un très grand nombre d'images, nous avons proposé une version incrémentale de l'algorithme d'AFC, qui ne charge à un instant t qu'un petit bloc d'images en mémoire. Puis nous avons utilisé un indicateur de l'AFC (la qualité de représentation) pour définir une structure de recherche pour les images basée sur un système de fichier inversé ce qui évite la comparaison entre l'image requête et toutes les images de la base de données. Enfin nous avons proposé une amélioration de cette structure de recherche en lui intégrant la Mesure de Dissimilarité Contextuelle de (Jegou et al., 2007). Les résultats expérimentaux montrent que :

- l'AFC est bien plus performante que *tf*idf*,
- les résultats sont encore améliorés (meilleure précision) lorsque que l'on intègre la Mesure de Dissimilarité Contextuelle dans notre structure de recherche, que ce soit dans la version de calcul hors-ligne ou à la volée.

Avec la méthode proposée seulement 0.06% de la base de données est exploré (en moins d'un huitième de seconde) et ces 0.06% contiennent 86.7% d'images pertinentes. Une amélioration possible de ce travail est la parallélisation de l'algorithme d'AFC.

Références

Benzecri, J. P. (1973). *L'analyse des correspondances*. Paris : Dunod.

Blei, D. M., A. Y. Ng, et M. I. Jordan (2003). Latent dirichlet allocation. *Journal of Machine Learning Research 3*, 993–1022.

Bosch, A., A. Zisserman, et X. Munoz (2006). Scene classification via pLSA. In *Proceedings of the European Conference on Computer Vision*.

Deerwester, S., S. Dumais, G. Furnas, T. Landauer, et R. Harsman (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science 41*(6), 391–407.

Greenacre, M. J. (2007). *Correspondence analysis in practice, Second edition*. Chapman and Hall.

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

Jegou, H., H. Harzallah, et C. Schmid (2007). A contextual dissimilarity measure for accurate and efficient image search. In *Proceedings of CVPR'07*, pp. 1–8.

Lienhart, R. et M. Slaney (2007). plsa on large scale image databases. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1217–1220.

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*, pp. 1150–1157.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, pp. 91–110.

Mikolajczyk, K. et C. Schmid (2004). Scale and affine invariant interest point detectors. *Proceedings of IJC V 60*(1), 63–86.

Morin, A. (2004). Intensive use of correspondence analysis for information retrieval. In *Proceedings of the 26th International Conference on Information Technology Interfaces, ITI2004*, pp. 255–258.

Nistér, D. et H. Stewénius (2006). Scalable recognition with a vocabulary tree. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Volume 2, pp. 2161–2168.

Pham, N.-K. et A. Morin (2008). 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*, Volume RNTI-E-11, pp. 475–486.

Salton, G. et C. Buckley (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management 24*(5), 513–523.

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

Sivic, J. et A. Zisserman (2003). Video Google : A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*, Volume 2, pp. 1470–1477.

Summary

We are interested in the intensive use of Factorial Correspondance Analysis (FCA) for Large-scale Content-Based Image Retrieval. We adapt FCA, method for analyzing textual data, on images using local descriptors SIFT. Here, FCA is used to reduce dimensions and limite the number of images to be considered during the search. An incremental algorithm for FCA is proposed to deal with large databases. Finally we integrate the Contextual Dissimilarity Measure in our search structure in order to improve the response time and the accuracy.