

Indexation et recherche d'images à très grande échelle avec une AFC incrémentale et parallèle sur GPU

Nguyen-Khang Pham, François Poulet, Annie Morin, Patrick Gros

IRISA - Texmex
Campus Universitaire de Beaulieu, 35042 Rennes Cedex
{prenom.nom}@irisa.fr
<http://www.irisa.fr/texmex>

Résumé. Nous présentons un nouvel algorithme incrémental et parallèle d'analyse factorielle des correspondances (AFC) pour la recherche d'images à grande échelle en utilisant le processeur de la carte graphique (GPU). L'AFC est adaptée à la recherche d'images par le contenu en utilisant des descripteurs locaux des images (SIFT). L'AFC permet de réduire le nombre de dimensions et de découvrir des thèmes qui permettent de diminuer le nombre d'images à parcourir et donc le temps de réponse d'une requête. Pour traiter de très grandes bases d'images, nous présentons une version incrémentale et parallèle d'AFC, puis nous utilisons ses indicateurs pour construire des fichiers inversés pour retrouver les images contenant les mêmes thèmes que l'image requête. Cette étape est elle aussi parallélisée sur GPU pour obtenir des réponses rapides. Les résultats numériques sur la base de données d'images Nistér-Stewénius plongée dans 1 million d'images de FlickrR montrent que notre algorithme incrémental et parallèle est très significativement plus rapide que sa version standard.

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 aisée à cause des possibles changements de point de vue, variation de luminosité ou occlusion. Récemment, l'utilisation de descripteurs locaux a permis d'obtenir de bons résultats pour l'analyse d'images. Contrairement aux descripteurs globaux qui sont calculés sur une image entière, les descripteurs locaux sont extraits en des points particuliers de l'image. Cela permet de trouver des images qui partagent un ou quelques éléments visuels seulement avec l'image requête. Initialement, les méthodes utilisaient des mécanismes de vote pour la recherche d'images en mettant en correspondance des points d'intérêts (Lowe, 1999). Puis des méthodes utilisées initialement pour des données texte comme la pondération $tf*idf$ (Salton et Buckley, 1988), LSA (Latent Semantic Analysis) (Deerwester et al., 1990), PLSA (Probabilistic Latent Semantic Analysis) (Hofmann, 1999) ou LDA (Latent Dirichlet Allocation) (Blei et al., 2003) ont été adaptées aux images (Bosch et al., 2006). Dans le traitement des données texte, ces méthodes utilisent un modèle de sac de mots : elles prennent en entrée une matrice de co-occurrence (appelée aussi tableau de contingence croisant les documents et les mots) et essayent de réduire la dimension. Dans le cas des

données images, les images jouent le rôle des documents et les "mots visuels" celui des mots. Nous allons utiliser l'Analyse Factorielle des Correspondances (AFC) (Benzécri, 1973) pour l'indexation et la recherche dans une grande base de données d'images.

La section 2 présente brièvement l'algorithme d'analyse factorielle des correspondances, puis sa version incrémentale et parallèle sur GPU (Graphics Processing Unit). La section 3 décrit la parallélisation de l'indexation et de la recherche d'images basés sur un indicateur de l'AFC. La section 4 présente quelques résultats avant la conclusion et les travaux futurs.

2 Analyse Factorielle des Correspondances

L'AFC est une méthode exploratoire classique pour l'analyse des tableaux de contingence, proposée dans le contexte de la l'analyse de données textuelles. Comme la plupart des méthodes factorielles, l'AFC utilise une décomposition en valeurs et vecteurs propres d'une matrice particulière et permet la visualisation des mots / documents dans un espace de dimension réduite. Cet espace a la particularité d'avoir un nuage de points projetés (mots et/ou documents) d'inertie maximale. De plus, l'AFC fournit des indicateurs pertinents pour l'interprétation des axes comme la qualité d'un mot et/ou d'un document sur un axe. L'AFC consiste donc à construire et à chercher les valeurs et vecteurs propres d'une matrice particulière : $A = F^T P^{-1} F Q^{-1}$ où F est la matrice de données, (le tableau de fréquences relatives), P et Q sont des matrices diagonales dont les éléments sont les totaux marginaux en ligne et colonne de F . Pour les grands tableaux de contingence, il est impossible de charger la matrice F en mémoire, nous proposons donc une méthode incrémentale et parallèle pour la construction de la matrice A . Nous avons étendu l'algorithme incrémental de (Pham et al, 09) pour en obtenir une version parallèle sur GPU. Nous avons donc développé une version parallèle de l'algorithme incrémental d'AFC sur GPU pour bénéficier de bonnes performances de calcul à faible coût. L'implémentation parallèle et incrémentale utilise la librairie CUBLAS pour les calculs matriciels. Le déroulement de l'algorithme est le suivant : pour chaque étape incrémentale i , on charge un bloc de données F_i en CPU, on le copie en GPU où on calcule les matrices Q et $F^T P^{-1} F$ en parallèle à l'aide de CUBLAS, il ne reste plus qu'à calculer la matrice A sur GPU avant de la recopier en CPU. L'algorithme calcule alors les K vecteurs propres de la matrice A sur CPU et les recopie en GPU. La fin de l'algorithme correspond à l'étape de projection : on parcourt l'ensemble des blocs de données, copie en CPU vers GPU et calcule la projection avant recopie du résultat en CPU.

3 Parallélisation de la recherche d'images par AFC sur GPU

L'application de l'AFC aux images permet notamment de réduire le nombre de dimensions de N à K . Si K est petit, les techniques d'indexation comme les fichiers inversés peuvent être utilisées pour accélérer la recherche. Dans le cas de l'AFC, le nombre d'axes conservés K est généralement assez élevé (souvent plus de cent) et la représentation réduite des images, issue de l'AFC, n'est plus creuse. L'utilisation directe des techniques de fichiers inversés est donc impossible. Nous cherchons donc une autre méthode pour accélérer la recherche d'images sans réduire la qualité des résultats. Notre approche se base sur un indicateur de l'AFC et un système de fichiers inversés construits à partir de cet indicateur pour accélérer la recherche.

3.1 Les indicateurs de l'AFC

L'AFC nous fournit deux indicateurs pertinents pour interpréter les résultats : la contribution d'un point (image ou mot visuel) à un axe et la qualité de représentation d'un point sur un axe. La qualité de représentation (ou cosinus carré) permet d'apprécier si un point est bien représenté sur un axe factoriel (facteur), on l'appelle aussi contribution relative du facteur à la position du point. Pour analyser la proximité entre les points, on s'intéresse aux points ayant un cosinus carré élevé. Les proximités entre points, observées dans le sous-espace factoriel donnent une bonne représentation de leurs proximités réelles. Habituellement, la représentation creuse des images (modèle de sac de mots visuels) est exploitée pour accélérer la recherche via un système de fichier inversé (en faisant l'hypothèse d'indépendance des mots). Mais dans le cas des images il existe une certaine dépendance entre les mots visuels. Les fichiers inversés basés sur les mots visuels seront donc moins efficaces.

Notre système de fichiers inversés n'est pas construit directement à partir des mots visuels mais se base sur les thèmes, issus des axes de l'AFC. L'hypothèse d'indépendance des thèmes est plus réaliste et le système ne souffre pas de la longueur des requêtes. Nous cherchons à associer les images aux thèmes auxquels elles appartiennent. Un fichier inversé est construit pour chaque thème et contient les images liées à ce thème. Soit une requête, la recherche consiste à déterminer les thèmes auxquels la requête appartient, puis nous fusionnons les fichiers inversés associés et filtrons les images non pertinentes (celles qui ont le moins de thèmes liés à la requête) pour former la liste d'images candidates. La recherche se fait ensuite séquentiellement dans la liste d'images candidates. Nous présentons ci-dessous le système de fichiers inversés basé sur la qualité de représentation de l'AFC. Les axes de l'AFC correspondant aux thèmes, nous construisons, pour un axe, deux fichiers inversés correspondant aux deux thèmes (pour les parties positive et négative). Etant donné un seuil $\varepsilon > 0$, les deux fichiers inversés basés sur la qualité de représentation associés à l'axe i , notés QF_i^+ et QF_i^- sont définis par : $QF_i^+ = \{j \mid \cos^2_i(j) > \varepsilon \text{ et } z_{ji} > 0\}$ et $QF_i^- = \{j \mid \cos^2_i(j) > \varepsilon \text{ et } z_{ji} < 0\}$, où z_{ji} est la coordonnée de l'image j sur l'axe i . Le seuil peut être choisi égal à la qualité de la représentation moyenne. Puisque la somme des cosinus carrés est égale à 1, le seuil ε est donc déterminé par $1/K$ avec K le nombre d'axes conservés.

3.2 Algorithme de recherche

L'association des images aux thèmes obtenus par l'AFC forme une représentation creuse des images. La représentation creuse est une version binaire de la dense (via l'association des images aux thèmes) qui ne contient que des 0 et des 1 (l'image appartient ou non au thème) et permet d'utiliser la technique des fichiers inversés. Le nombre de thèmes est beaucoup plus petit que le nombre de mots visuels et une image n'appartient qu'à un petit nombre de thèmes : la représentation creuse est plus compacte que celle basée sur les mots visuels. Mais les mesures de similarité ne peuvent pas se calculer directement à partir de la représentation basée sur les thèmes à cause de la perte d'information (la discrimination des images) de la représentation creuse. Nous proposons donc un algorithme de recherche en deux étapes qui exploite la représentation creuse (les thèmes) pour accélérer la recherche et le pouvoir

discriminant de la représentation dense des images pour améliorer les résultats. La première étape de la recherche filtre les images non pertinentes pour la requête et donne une liste d'images candidates en utilisant les fichiers inversés, la seconde affine la recherche par balayage séquentiel dans la liste de candidates. La similarité des images sera mesurée à partir de la représentation dense. Le tableau 1 décrit les étapes de l'algorithme de recherche.

- 1) projection de la requête r dans l'espace réduit
- 2) détermination des thèmes de la requête et choix des fichiers inversés correspondant
- 3) fusion des fichiers inversés et calcul de la fréquence des images
- 4) filtrage des images non pertinentes, construction de la liste des images candidates
- 5) chercher les plus proches voisins de la requête r dans la liste d'images candidates

TAB. 1 – *Algorithme de recherche basé sur les fichiers inversés*

Les thèmes associés à une requête sont déterminés de la même manière que la construction des fichiers inversés, en prenant les axes dont la qualité de représentation est supérieure à un seuil ϵ . Lorsqu'un axe est choisi, le thème correspondant à la partie négative ou positive sera associé à la requête. Soit F l'ensemble des n fichiers inversés associés à la requête r , obtenus par l'étape de détermination des thèmes. La fréquence d'une image i , notée $\text{freq}(i)$ est définie comme le nombre de fichiers inversés auxquels l'image i appartient, c'est également le nombre de thèmes que l'image partage avec la requête. L'ensemble des fichiers inversés est fusionné et on obtient alors une liste d'images avec leurs fréquences. Le filtrage des images non pertinentes se fait sous l'hypothèse de pertinence "plus la fréquence d'une image est élevée, plus l'image est pertinente pour la requête". Le filtrage consiste donc à déterminer un seuil θ tel que la liste d'images candidates contienne des images pertinentes pour la requête r . Pour une raison d'efficacité θ doit être le plus grand possible car la taille de la liste d'images candidates décroît lorsque θ augmente. Pour garantir que le système retourne toujours au moins n images, il faut que la liste d'images candidates contienne au moins n images. Partant de cette idée, nous proposons de choisir un seuil θ tel que la liste d'images candidates contienne au moins s images (par exemple 500).

3.3 Parallélisation de la recherche d'image par AFC sur GPU

La méthode de recherche d'images approximative que nous avons proposée permet d'accélérer de 4 à 20 fois la recherche par rapport à une recherche exhaustive sans diminuer la qualité des résultats. Pour une base d'un million d'images, 99% du temps de calcul est consacré à l'étape de filtrage, ce qui nous incite à paralléliser cette étape afin d'accélérer la recherche. La parallélisation est réalisée sur GPU en utilisant CUDA.

L'étape de filtrage dans la recherche consiste à calculer les fréquences de chaque image dans la base et à rejeter les images non pertinentes pour la requête en se basant sur leur fréquence. Pour une parallélisation efficace des calculs sur GPU, chaque processus ne doit manipuler et accéder que ses propres données. C'est la raison pour laquelle nous représentons les fichiers comme des vecteurs de bits. Un fichier inversé représente la présence ou l'absence des images par des bits 1 et 0 pour compresser les fichiers inversés et permettre également le calcul en parallèle de la fréquence des images avec indépendance entre processus. Chaque processus sur GPU calcule la fréquence de 8 images (un octet). Pour une

image requête, on détermine les fichiers inversés correspondants et on transfère les indices de ces fichiers vers le GPU pour le calcul parallèle de la fréquence des images en faisant la somme sur les bits correspondant des fichiers inversés. On copie le résultat en CPU et on détermine le seuil θ comme dans la version non parallèle (même liste d'images candidates).

Base d'images	GPU		CPU		exhaustive	
	temps(ms)	P@3	temps(ms)	P@3	temps(ms)	P@3
100 k	1.47	0.681	7.56	0.681	94.32	0.676
200 k	2.09	0.665	13.53	0.665	179.76	0.660
500 k	4.17	0.641	33.07	0.641	435	0.639
1M	7.53	0.625	79.99	0.625	860.88	0.623

TAB. 2 – Comparaison des méthodes de recherche

4 Expérimentations

Nous avons réalisé les expérimentations sur la base (Nistér et Stewénus, 2006). Pour évaluer le passage à l'échelle des méthodes proposées, nous avons fusionné la base Nistér-Stewénus avec 100 000, 200 000, 500 000 et un million d'images de FlickrR. Le nombre de mots visuels est fixé à 5000. Les algorithmes sont implémentés en C++ avec les bibliothèques LAPACK et ATLAS. Les versions parallèles de l'AFC et de filtrage sur GPU utilisent CUBLAS et CUDA sur une NVidia GeForce GTX-280 (PC Intel, 3.2Ghz, 8Go ram). Après avoir effectué l'AFC sur les images, nous avons conservé les 500 premiers axes, la mesure du cosinus est utilisée pour calculer la similarité entre les images et la méthode de recherche approximative utilise les fichiers inversés basés sur la qualité de représentation. Le seuil θ est choisi tel que la liste d'images candidates contienne au moins 500 images. La version parallèle sur GPU est environ 12 fois (pour la construction de la matrice A et la projection des images sur les axes factoriels) et 7 fois (au total) plus rapide que la version non parallèle sur des bases d'images avec différentes tailles. Le tableau 2 montre le temps de réponse en millisecondes et la précision aux trois premières images retournées (P@3) pour différentes méthodes de recherche : la méthode approximative avec le filtrage des images non pertinentes (sur GPU et sur CPU) et la méthode exhaustive. Sur une base d'un million d'images, la méthode avec filtrage des images non pertinentes effectué sur GPU est 10 fois plus rapide que la méthode non parallèle et 114 fois plus rapide qu'une recherche exhaustive. De plus, la recherche approximative donne de meilleurs résultats que la méthode exhaustive grâce à la qualité de représentation (indicateur de l'AFC) pour interpréter la proximité des images.

5 Conclusion et perspectives

Nous avons présenté un nouvel algorithme incrémental et parallèle d'ACP sur GPU. L'aspect incrémental de l'algorithme nous permet de traiter de très grands tableaux de contingence, dans un temps réduit, par une parallélisation sur GPU. Les gains sont significatifs, puisque la durée de calcul est globalement divisée par un facteur 10. Basé sur l'un des indicateurs de l'AFC, nous avons proposé une méthode de construction de fichiers inversés parallélisée elle aussi sur GPU, qui permet l'indexation de très grandes bases. Les gains sont là encore intéressants, puisque la vitesse est divisée par 10 par rapport à la même

version (CPU) et par plus de 100 par rapport à une recherche exhaustive, de plus, la précision est améliorée par rapport à la recherche exhaustive, par l'utilisation de l'un des indicateurs de l'AFC. Plusieurs améliorations sont à venir, une première est de paralléliser le calcul des vecteurs propres de la matrice (1/3 du temps de calcul actuel).

Références

- Benzécri J.P, (1973). L'analyse des correspondances, Paris, Dunod, 1973.
- Blei D.M, Ng A.Y, Jordan M, (2003). Latent dirichlet allocation in Journal of Machine Learning Research, 3:993-1022.
- Bosch A., Zisserman A, Munoz X, (2006). Scene classification via pLSA in proc. of the European Conference on Computer Vision, Graz, Autriche, 517-530.
- Deerwester S, Dumais S, Furnas G, Landauer T, Harsman R, (1990). Indexing by latent semantic analysis in Journal of the American Society for Information Science, 41(6):391-407.
- Hofmann T, (1999). Probabilistic latent semantic analysis, in proc. of the 15th Conference on Uncertainty in Artificial Intelligence, Stockholm, Suède, 289-296.
- Lowe D.G, (1999). Object recognition from local scale-invariant features, in proc. of the 7th International Conference on Computer Vision, Corfou, Grèce, 1150-1157.
- Nistér D. et Stewénus H, (2006). Scalable recognition with a vocabulary tree, in proc. of the IEEE Conference on Computer Vision and Pattern Recognition, vol.2, 2161-2168.
- Pham N-K, Morin A, Gros P. (2009), Utilisation de l'AFC pour la recherche d'images à grande échelle, in RNTI-E15, 283-294.
- Salton G. et Buckley C, (1988). Term weighting approaches in automatic text retrieval in Information Processing & Management, 24(5):513-523.

Summary

We present a new incremental and parallel Factorial Correspondence Analysis (FCA) algorithm on GPU for large scale content based image retrieval. The FCA is adapted to image data by the use of local image descriptors (SIFT). FCA allows reducing the number of dimensions and discovering topics. In image retrieval, its use can reduce the number of images to parse and decrease the time needed to answer a request. To deal with very large databases, we present an incremental and parallel FCA algorithm. The incremental part split the database into small blocks processed one by one, this allow us to deal with potentially very large databases. The parallel part is the use of massively parallel architecture (GPU) to run the algorithm in a tractable time at low cost. Once the FCA has been performed, the FCA indicators are used to build inverted files for image retrieval on GPU to get as fast as possible answer to a request. Some results with Nister-Stewenius Dataset plugged in 1 million images show our algorithm is significantly faster than its standard version.