Architecture des bases d'images généralistes organisées en clusters

Z.Guellil[1] et L.Zaoui[2]

[1,2]Université des sciences et de la technologie d'Oran MB, Université Mohamed Boudiaf USTO -BP 1505 El Mnaouer -ORAN - Algérie
[1] g.zouaoui@gmail.com, [2] Zaoui_Lynda@yahoo.fr

Résumer. Les développements actuels, en matière de technologie liée à l'information numérique, ont permis l'acquisition et le stockage d'une quantité importante d'information, ce qui a engendré la nécessité d'élaborer des systèmes permettant la gestion de ces données (plus particulièrement les bases de données images). La recherche d'images par le contenu tente de répondre à ces besoins en se basant sur des caractéristiques de bas niveaux comme la couleur, la texture et la forme. C'est un domaine très actif dont les premières recherches s'étaient focalisées sur le stockage et l'étude des descripteurs pertinents pour la recherche. L'architecture retenue dans ces systèmes repose sur le stockage des images dans un seul emplacement et la recherche, se fait en balayant cet ensemble mais face à de grand volume de données cette architecture est devenue désuète. Actuellement, l'idée est d'utiliser une architecture à plusieurs clusters où chaque cluster contient un ensemble d'images similaires afin d'améliorer les performances en termes de qualité des résultats et du temps de réponse. Nous présentons dans cet article les défis à relever dans le développement d'un tel système d'indexation et de recherche d'images et les résultats de nos efforts de recherche.
Mot-clés : Bases d'images, indexation, Recherché d'image, Classification, clusters, images similaires.

1 Introduction

La gestion des bases de données images nécessite des systèmes spécifiques, les premiers systèmes étaient basés sur la recherche par mot-clés, ces systèmes ont montré quelques limites à cause de la subjectivité des mots-clés attribués. Ces limites ont conduit à la naissance des systèmes d'indexation et de recherche d'images par le contenu physique de l'image (CBIR, en anglais Content-Based Image Retrieval).

Le contenu d'une image possède des caractéristiques permettant de la résumer par des métriques mathématiques appelés descripteurs, ces descripteurs sont fondés sur des caractéristiques visuelles comme la couleur [6] [10], la texture [4] [8] et la forme [12].

Ces caractéristiques, dites de bas niveau, peuvent être calculées globalement sur l'image (descripteur global), comme ils peuvent être calculés au niveau local. Dans plusieurs travaux [1] [5] [6], l'arbre quaternaire a été employé pour le calcul des descripteurs local d'une image, il permet de décrire l'image à plusieurs niveaux, en la décomposant récursivement en quatre quadrants, plus l'arbre est profond plus la description est locale. La mesure de similarité entre deux images correspond à la distance entre leurs arbres quaternaires, on distingue trois distances principales T, Q, V [9].

Bien que cette technique de recherche (CBIR), soit puissante, elle aussi souffre d'un certain nombre de limites comme le fossé sémantique et le temps requis pour la recherche. La première limite est dû au fait que la recherche vise les images similaires à une image donnée [11], alors que le deuxième problème est la conséquence du volume élevé des données à explorer pour trouver des résultats.

Dans ce travail, nous proposons une architecture d'une base d'image en clusters, qui va permettre de réduire le volume des données à explorer en n'effectuant la recherche que dans le cluster jugé pertinent, les images du même cluster doivent être le plus similaires possible pour assurer la qualité des résultats.

2 Architecture d'un système d'indexation et de recherche d'images

Un Système d'indexation et de recherche d'images permet aux utilisateurs de retrouver les images qui satisfont leurs besoins dans des bases de données d'images. Son fonctionnement est décomposé en deux étapes : l'étape d'indexation dit, hors ligne (offline) ou le système va extraire les caractéristiques des images de la base et les stocker dans une base de données. La deuxième étape consiste à extraire les descripteurs de l'image requête et la comparer avec les descripteurs existant dans la base de données afin de trouver les images similaires à celle désirée.

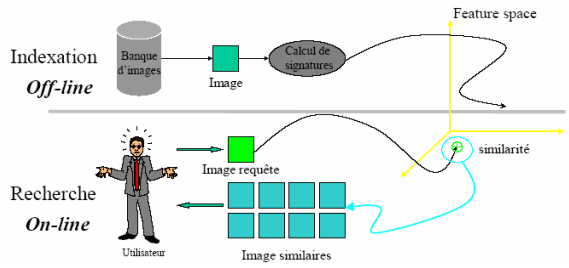


Figure. 1. Aperçu d'un système d'indexation et de recherche d'image.

La conception de notre système assure les fonctionnalités requises pour ces deux étapes à l'aide de quatre modules : module de représentation des images, module de classification, le module de stockage et le module de recherche.

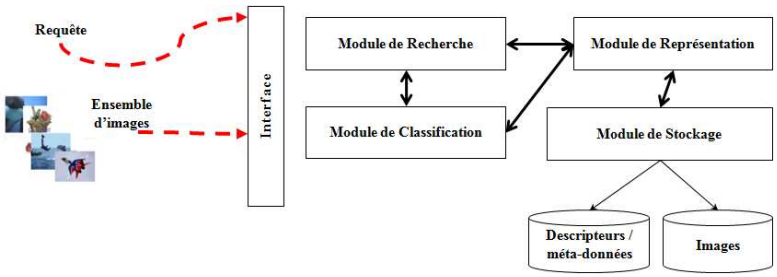


Figure. 2. Architecture en module du CBIR.

2.1 Module de représentation des images

Ce module est chargé de représenter les images dans un format unique et approprié, il permet aussi certaines opérations de base. Notre système exploite des descripteurs locaux, d'où la nécessité d'une représentation en arbre quaternaire.

Cette représentation rend le système indépendant du type (matriciel ou vectoriel) et du format d'image (BMP ou JPEG... etc.), de plus, cela permet la compression d'une image représentant une zone homogène par nœud du quadtree au lieu d'un grand nombre de pixels.

2.2 Module de Classification

Le module de classification est chargé de grouper l'ensemble d'images en N sous ensembles d'images similaires (phase hors ligne) et la détermination de la classe d'appartenance d'une image requête ou nouvellement insérée dans la base (phase en ligne). Pour cela, nous avant choisi deux méthodes de classification (k-means et PAM) afin de comparer l'usage de deux modes de représentations possible (centroide et medoide). La conception de ce module prend en charge les problèmes d'initialisation et le choix du nombre de classes. Les sous-sections suivantes exposeront les détails de cette conception.

Problème d'initialisation

La principale limite des méthodes par partitionnement est la dépendance des résultats des valeurs de départ (centres initiaux), à chaque initialisation correspond une solution différente (optimum local) qui peut dans certain cas être très loin de la solution optimale (optimum global).

La recherche des bonnes valeurs de départ s'avère difficile du fait que l'ensemble des valeurs possibles est très grand et qu'en général, les données occupent un très petit espace de cet ensemble sous forme de groupes, de ce fait nous constatons que les meilleurs points de départ sont ceux qui sont choisis parmi les données.

Par principe de regroupement des données, un objet est affecté à un groupe s'il lui est le plus proche, plus la distance entre eux diminue, plus la probabilité

d’appartenance à ce groupe augmente, dans le cas contraire, l’objet le plus loin de son groupe d’appartenance est considéré comme étant mal classé, il fera certainement un bon candidat afin de former le nouveau centre.

Nous proposons d’amorcer l’initialisation avec deux centres, afin d’assurer la séparabilité des données au cours de la classification, il est évidant de choisir au début les deux données les plus éloignées, à la suite le nouveau centre sera représenté avec la donnée la plus mal classée. Ce principe est illustré par l’algorithme suivant :

```
Algorithme 1 initialisation  par le  mal classé.
Début
   1) Création  d’une  matrice de distance
   2) Choisir  les  deux éléments les  plus  éloignés
      (ils  représentent les  deux premiers  noyaux) ;
   TANT QUE le nombre de classe souhaité  n’est  pas
      atteint  Faire
        3) Affecter  les  individus  aux noyaux
          disponibles ;
        4) Sélectionner  un élément mal classé (celui
          qui  possède  la  plus  grande  distance  de
          son centre le  plus  proche) ;
        5) Ajouter  cet individu  à l’ensemble  des
          noyaux ;
        6) Augmenter le  nombre  des  noyaux ;
   Fin Tant Que ;
Fin.
```

L’avantage de cette technique est qu’elle est indépendante de la méthode de regroupement et de la distance choisie. Le choix des valeurs trop éloignées permet de maximiser l’inertie inter classe, la minimisation de l’inertie intra classe est assurée par l’algorithme de regroupement.

Choix du nombre de groupes

Le choix du nombre de groupes est un facteur important qui influence sur les résultats en classification automatique, pour un nombre de groupes élevés[1] cela conduit à une partition non signifiante en divisant des groupes homogènes en plusieurs, au contraire un nombre plus faible conduit à une partition de groupe non homogène, ceci est dû au principe de création d’une partition, un élément doit appartenir à un seul et unique groupe, même si tous les groupes lui sont différents, il doit appartenir à l'un d’entre eux.

Rappelons que notre objectif vise à regrouper les images similaires dans un même cluster pour réduire l’espace de recherche et assurer une bonne qualité des résultats d’une recherche dans un cluster. Nous proposons de fixer une distance maximale entre une image et le représentant de son groupe, une image n’est affectée à un groupe que si la distance qui les sépare est inférieur à ce seuil, parmi les images non classées, nous choisissons celle qui possède la distance la plus élevée vers un groupe, cette image sera la représentante du groupe construit.

[1] Un nombre de groupes plus grands que le nombre de regroupements existants dans les données.

Remarquons que ce principe va augmenter le nombre de classes à chaque fois que cela sera nécessaire, en plus il peut être appliqué après l'organisation de la base d'images lors d'une mise à jour, si une image n'appartient à aucun des groupes existants, un nouveau groupe est créé. Enfin, il est recommandé d'amorcer le classifieur avec un nombre de groupes minimal.

k-means

Le k-means [2] exploite la représentation de classe en centroïde (moyenne des descripteurs de l'ensemble), l'application de cet algorithme au partitionnement d'un ensemble d'images représentées par des arbres quaternaires nécessite la définition du vecteur de descripteur et le représentant de classe.

Le vecteur de descripteur est construit à partir des arbres quaternaires, c'est un vecteur où chaque élément contient les descripteurs d'une feuille de sorte à avoir dans la même position dans le vecteur les feuilles homologue (du même identifiant). La Figure suivante illustre ce principe.

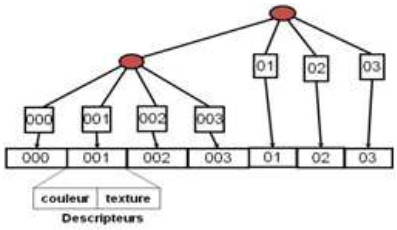


Figure. 3. Principe de construction des descripteurs.

Cependant, il est rare d'avoir la même structure d'arbre, cela nécessite de normaliser les arbres quaternaires pour avoir la même structure, cette procédure est réalisée soit :

- En complétant à la structure maximale (le plus grand arbre représentant une image de la base), ce qui va produire des vecteurs de descripteur de très grande taille.
- En fixant une profondeur maximale, les arbres n'ayant pas atteint cette profondeur seront complétés, ceux qui ont dépassé seront réduits, par conséquent, la taille de descripteur sera raisonnable avec une perte d'information.

D'après la nature de représentation des images (arbre quaternaire), la distance visuelle (V) est la mieux adaptée à ce type de situation, de plus, cette structure ne pose pas de problèmes lors du calcul de la distance v, puisque la structure est connue (arbre et descripteur), et donc les pondérations de chaque niveau sont connu.

Le représentant de classe est un vecteur de descripteur d'une image virtuelle, il sert à regrouper les images les plus similaires dans le même cluster, sa valeur est la moyenne des descripteurs appartenant au groupe lui-même.

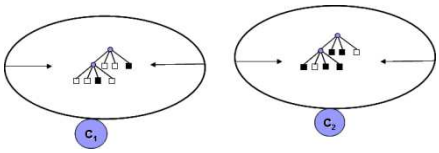


Figure. 4. Les images sont affectées au centre le plus proche (la position moyenne des modèles).

PAM

L'algorithme PAM [3] propose de représenter les classes par un objet de l'ensemble des données (medoide), c'est un objet dont la dissimilarité moyenne avec les autres objets du cluster est minimale. Cette représentation permet d'être moins sensible aux outliers ainsi que de classer n'importe quel type de données. Dans notre cas il est intéressant d'avoir un représentant d'une classe une image réelle.

2.3 Module de Stockage

Ce module utilise les deux modules décrit précédemment, pour accomplir sa tache, il est capable de gérer une base d'images organisée en plusieurs clusters en utilisant des méta-informations de l'ensemble de données.

Les méta-informations contiennent deux types d'informations, le premier concerne l'organisation de la base, il donne des informations sur le nombre des clusters utilisés et l'emplacement de chaque cluster, le deuxième type décrit les groupes, il diffère selon le type de la méthode de clustering utilisée.

Dans le cas du k-means nous avons besoin de connaître les représentants des clusters, ils sont stockés sous forme d'arbre quaternaire afin de pouvoir calculer la distance visuelle entre l'image[2] à classer et les différents représentants des groupes.

Le cas d'une méthode indépendante du type et de la structure de données tel que le PAM, les informations sont relativement plus simples, pour déterminer le groupe d'appartenance d'une image, il suffit de connaître l'identifiant de l'image medoïde de chaque groupe, et le type de distance utilisé pour le regroupement de la base (distance V, Q ou autre).

2.4 Module de recherche

Le module de recherche fournit une interface qui permet aux utilisateurs de spécifier leurs requêtes, et restitue les résultats correspondants aux critères de la requête (images similaires). Dans notre prototype, on a opté pour les requêtes par l'exemple [7], nous proposant deux types de recherches, la première baptisée « N plus proches images » dont le résultat est un ensemble de N images similaires à la requête,

[2] Cette image peut-être soit une image requête ou bien une image à insérer lors d'une mise à jour.

le deuxième permet de restituer les images qui sont dissimilaire à x % de l'image requête.

L'exécution d'une requête suit le processus suivant : Tout d'abord le module de présentation transforme l'image en arbre quaternaire il la transmet ensuite au module de classification pour la classer et déterminer le meilleur groupe qui fournit un bon résultat, enfin le module de recherche procède à un balayage de l'ensemble des descripteurs des images du cluster ciblé.

3 Évaluation et Expérimental

Notre expérimentation porte sur trois points principaux, évaluation de la structure de données utilisées par le module de classification où nous effectuons une comparaison entre l'usage d'un ensemble de vecteurs de descripteur et une ensemble de distances entre les objets à classer. Le deuxième point, évalue la qualité de la classification pour les deux algorithmes et l'apport de notre stratégie d'initialisation proposée. Enfin, le dernier point abordera le temps de réponse du système dans le cas d'une BDI organisée en mono ou multi clusters.

3.1 Évaluation du mode de représentation

Les images représenté par les arbres quaternaire peuvent se présenté a un classifieur sous deux formes : ensemble de vecteurs de descripteur, ceci nécessite que les arbres quaternaires de l'ensemble aient la même structure pour qu'on puisse les transformer en vecteurs de même dimension, ou un ensemble de distance, dans ce cas, on à besoin de calculer toutes les distances entre chaque pair d'image. Cette partie discute l'utilisation des deux modes (centroide pour représentation en vecteur ou medoide pour l'ensemble des distances).

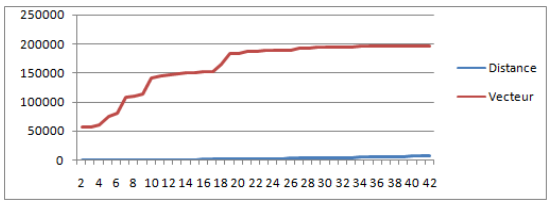


Figure. 5. Taille de la matrice de distance et un vecteur de descripteur en fonction du nombre d'images.

Dans ce graphe nous remarquons que la taille d'un seul vecteur représentant un arbre quaternaire est plus grande que celle d'une matrice de similarité contenant toutes les mesures de similarité.

Ajoutant à ce point, la complexité de la procédure d'affectation d'un nouvel élément, elle se fait suivant la même procédure dans les deux cas (présentation en centre ou medoïde) et ceci en suivant les mêmes étapes. Tout d'abord, les descripteurs

sont calculés par le module de présentation, ensuite le module de classification déterminera le groupe le plus approprié.

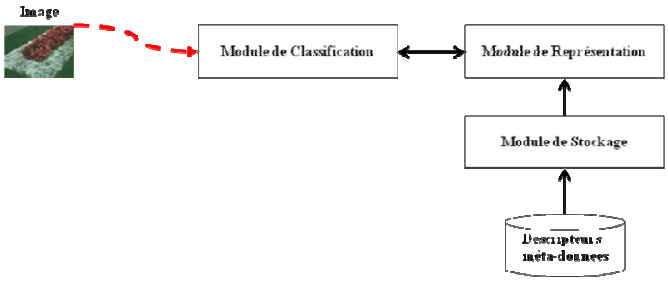


Figure. 6. Mécanisme d'affectation.

Ce qui fait la différence entre les deux modes de représentation est la structure requise pour effectuer cette opération. Lorsqu'on travaille avec des centres, on est obligé de transformer l'arbre quaternaire en vecteur de même dimension que celui qui représente les centres des groupes, ce qui nécessite de garder la structure de l'arbre obtenu dans la phase hors ligne. Par contre l'utilisation des medoïdes, seul l'identifient de l'image medoïde (représentant du groupe) est requis.

La structure de données légères et la flexibilité de classer une nouvelle image sont deux arguments qui nous motivent et qui nous boutent à utiliser des algorithmes basés sur les mesures de similarité pour ce type données (image représentée en arbre quaternaire).

3.2 Évaluation des classifications

Dans cette partie, nous validons l'apport de notre stratégie d'initialisation par l'application du k-means et PAM sur l'ensemble d'images. Rappelons qu'on ne peut utiliser que la distance visuelle V dans le cas des centres nous avons choisi appliqué le PAM en utilisant la distance visuelle afin de tester les deux algorithmes dans le même contexte.

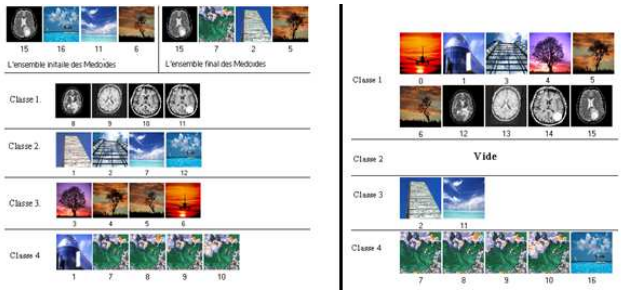


Figure. 7. Résultat de classification par la méthode PAM a gauche et k-means adroit (Distance V et initialisation aléatoire).

Dans cette Figure, on remarque dans l'ensemble des médoïdes initiaux la présence de deux images similaires (16, 11), en conséquence la classification a été perturbée d'où la présence d'une image mal classée dans la classe 4 (image « 1 »). Dans le cas du k-means les choses sont encore pires. L'algorithme guidé par une partition initiale (initialisation aléatoire) doit donner en sortie une solution optimale, l'initialisation aléatoire ne couvre pas l'espace occupé par les données ce qui produit des résultats inacceptables comme dans le cas de la Figure précédente, on remarque que la classe 3 est vides alors que le contenu de la classe 1 et 4 n'est pas homogène.

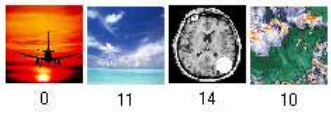


Figure. 8. Les images choisies par l'algorithme d'initialisation.

Nous avons utilisé l'algorithme proposé dans la section 2.2.1 pour sélectionner une partition initiale et d'améliorer les résultats du PAM et k-means, la Figure 8 présente les images qui représentent les groupes de la partition initiale. La Figure 9 affiche les résultats du PAM et k-means respectivement, on remarque que les résultats produits par les deux algorithmes sont bien améliorés grâce au bon choix de la partition initiale.

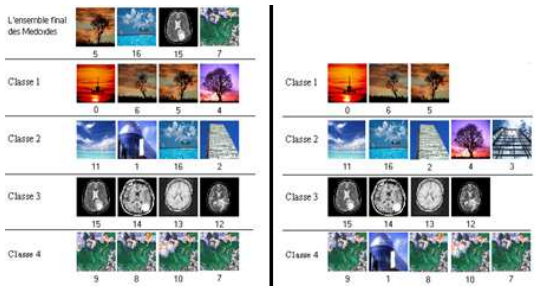


Figure. 9. Résultat de classification par le PAM et le k-means amélioré.

Bien que les deux algorithmes ont procédé à l'optimisation de la même partition leurs résultats sont différents, cette différence est expliquée par le fait que la nature des deux algorithmes tel que le PAM qui regroupe au sein du mêmes ensembles les images dont la distance entre eux est minimale, alors que le k-means utilise des centres, qui représentent la moyenne de chaque groupe, ces centres changent de position (valeur) au cours de la classification et de fausse moyenne conduit à de faux résultats.

3.3 Temps de recherche

Afin d'évaluer l'amélioration du temps de réponse apportée par l'architecture proposée nous avons mesuré le temps nécessaire pour exécuter une requête (temps global), temps d'affectation d'une image au groupe et le temps consommé pour la recherche dans ce groupe (dans le cas multi clusters), sur un ensemble de base d'image organisé en clusters et nous l'avons comparé avec celui consommé dans le cas d'une architecture mono cluster.

La base d'image choisie contienne 147 images, pour chaque test, le nombre de clusters varie de 2 à 40, on exécute trois requêtes différentes, appartienne à différentes classes et donc les descripteurs des images requêtes et l'ensemble des images des clusters de différentes tailles (arbre quaternaire), on calcule pour l'ensemble la moyenne du temps consommer.

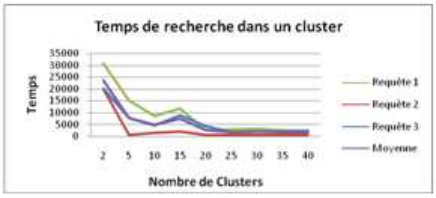
Figure.10. Les images requête.

Nous avons remarqué que le temps de recherche et celui de la classification d'une image requête n'est pas le même pour chaque image requête, cette différence est expliquée par le fait que la taille des arbres quaternaires de chaque image en nœud et en feuille est différente, dans l'image 1 :38113 Nœuds dont 28 585 feuilles, Image2 : 6137 Nœuds dont 4603 feuilles, Image 3 :2061 Nœuds dont 1546 feuilles. Ce qui influence sur le temps de calcul des distances et donc le temps de recherche et de classification. Un autre facteur qui entre en jeu c'est le nombre d'images que contienne le cluster de recherche.

Le temps de recherche sur la même base d'images organisé en mono cluster est :

- Requête 1 : 57650,85 ms.
- Requête 2 : 51002.24 ms.
- Requête 3 : 49215,29 ms.

Dans un cluster le temps de recherche diminue lorsque le nombre de ces derniers augmente. Il est évident que lorsque le nombre de clusters augmente leurs cardinalités réduit, en conséquence l'espace de recherche est limité sur un sous ensemble ce qui explique la réduction du temps de recherche.



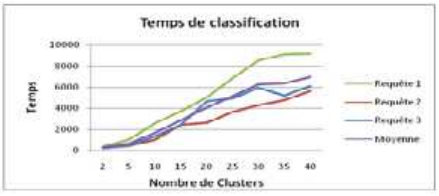


Figure. 11. L'influence du nombre de clusters sur temps de recherche et le temps de d'affectation.

Au contraire, lorsque le nombre de clusters augmente, l'image requête est comparée avec un nombre plus élevé de représentants des clusters est provoquera donc l'augmentation du temps requis pour l'affectation d'une image.

On augmentant le nombre de clusters, le temps de classification augmente et celui de la recherche diminue et influence le temps global de recherche, on observe sur le graphe de la Figure 12 que le temps global de recherche est décroissant sur la première partie du graphe et croissant dans la deuxième partie, ainsi que le taux de décroissance est supérieur à celui de croissance à cause des fortes changement des nombres d'images des clusters pour un nombre faible de clusters.

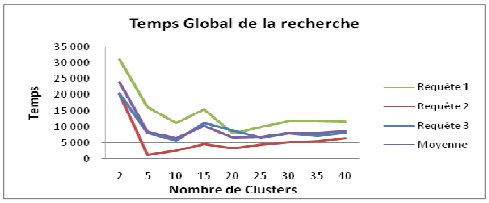


Figure. 12. Le temps global de la recherche.

Du point de vu temps de recherche, le nombre de clusters optimale est celui qui réduit au minimum le temps de recherche, malheureusement, il ne l'est pas forcément pour la classification de l'ensemble de la base d'image et peut produire une mauvaise classification entraînant une pauvre pertinence des résultats lors de la recherche.

Le temps global de recherche est moins prioritaire que la qualité des résultats pour cette raison notre solution au problème du nombre de classes a été : de fixer une dissimilarité maximale intra groupe, tel que tout groupe ayant un élément qui dépasse cette valeur doit construire son propre groupe, cette astuce permettra non seulement de préserver la qualité des résultats mais aussi d'équilibrer les clusters.

4 Conclusion

Nous avons présenté dans ce document une architecture d'un système d'indexation et de recherche d'images par le contenu basé sur une indexation en clusters, nous avons abordé certains problèmes et difficultés rencontrés lors de la mise en œuvre de cette architecture qui sont relié au module de classification (le cœur du système).

Notre système utilise des descripteurs locale basée sur les arbres quaternaires, pour la classification de cette structure de données, nous avons divisé les algorithmes de clustering en deux types : Méthodes basées sur représentation en centre et méthodes basées sur les distances entre les objets à classer, à des fins d'optimisation, l'utilisation des algorithmes basés sur les distances entre les objets est recommandé.

Le module de classification utilise deux mécanismes d'optimisation, le premier permet de déterminer le nombre optimal de clusters dans la phase indexation, et de

contrôler les résultats d'une recherche dans la phase online, ceci en fixant la dissimilarité maximal entre une image est le représentant de son cluster. Le deuxième mécanisme est un algorithme d'initialisation qui permet d'améliorer les résultats du clustering, il est applicable avec tout algorithme de clustering nécessitant une initialisation (EM, PAM, …).

Références

1. ALBUZ E., KOCALAR E., KHOKHAR A.: Quantized CIELab* Space and Encoded Spatial Structure for Scalable Indexing of Large Color Image Archives. In: IEEE Conf. on Acoustics, Speech, and Signal Processing (ICASSP), June 2000.
2. Boris Mirkin.: Clustering for Data Mining: A Data Recovery Approach, Chapman & Hall/CRC, 2005.
3. Douglas H. Fisher;: Knowledge acquisition via incremental conceptual clustering, Machine Learning, Volume: 2.pp139-172 1987.
4. R. M. Haralick. Statistical and structural approaches to texture. In: Proceedings of the IEEE, vol. 67, no. 5, pp. 786-804, 1979.
5. LIN S., TAMER ÖZSU M., ORIA V., NG R. An Extensible Hash for Multi-Precision Similarity Querying of Image Databases. In: Proc of the 27th Int. Conf. on Very Large DataBase (VLDB'2001), Roma (Italy), 2001.
6. LU H., OOI B.-C., TAN K.-L. Efficient Image Retrieval by Color Contents. In: Conf. on Applications of Database (ADB-94), Vadstena (Sweden), juin 1994
7. M. Flickner et al. Query by image and video content: The QBIC system. In M. T. Maybury, editor, Intelligent Multimedia Information Retrieval, chapter 1, pages 7–22. 1997.
8. Maria Petrou, Pedro Garcia Sevilla, Image processing : dealing with texture,Chichester: John Wiley & Sons, 2006.
9. Marta Rukoz, Maude Manouvrier, Geneviève Jomier. Distances de similarité d'images basées sur les arbres quaternaires, In Proceedings 18èmes Journées Bases de Données Avancées (BDA'02), pages 307-326, 2002.
10. Swain, M.J. & Ballard, D.H. Color Indexing. In: International Journal of Computer Vision, 7(1), pp. 11-32.1991
11. TAN K.-L., OOI B. C., YEE C. An Evaluation of Color-Spatial Retrieval Techniques for Large Image Databases.In: Multimedia Tools and Applications, vol. 14, p. 55–78, 2001.
12. Zhang and G. Lu, A Comparative Study of Curvature Scale Space and Fourier Descriptors. In: Journal of Visual Communication and Image Representation, 14(1):41-60, 2003