

Segmentation d'une Base d'Images Hétérogène en des Groupes d'Images Homogènes par une Méthode Issue du Domaine du Data Mining

Souhila DJERROUD* et Lynda ZAOUÏ*

* B.P. 1505 El-Mnaouer, Oran, 31036, Algeria.

Dsij_umd@yahoo.fr

Zaoui_Lynda@yahoo.fr

Résumé : Les méthodes de recherche d'images par le contenu procèdent dans un premier temps à une indexation dans une base d'images. Il s'agit d'extraire des descripteurs de bas niveaux (couleur, texture, forme.) Afin de pouvoir établir un critère de similarité entre elles, ensuite, pour chaque requête de l'utilisateur, une distance est calculée en fonction de ces descripteurs et les images les plus proches de la requête sont retournées. Le principal objectif de cet article est de présenter une méthode de segmentation issue du domaine du Data Mining permettant de regrouper les images d'une base généraliste en groupe d'images similaires qui seront stockées par une structure arborescente particulière : « Arbre Quaternaire générique » qui permet la minimisation de l'espace de stockage par partage d'informations entre arbre quaternaire représentant les images. Cette méthode est une adaptation de la méthode de segmentation hiérarchique CHAMELEON aux images, elle nécessite la définition d'une mesure de dissimilarité (ou similarité) entre les images représentées en arbre quaternaire.

Mots clefs : Base d'images ; arbre quaternaire générique ; segmentation hiérarchique; CHAMELEON.

INTRODUCTION

La nature des documents numériques a profondément évolué au cours des dernières décennies. Quelque soit leur type (texte, image, son,...) les moyens de création, de duplication et de transmission se sont rapidement développés, conduisant leurs nombres à s'accroître considérablement. En conséquence, la gestion, la recherche et l'exploration de ces documents nécessitent des moyens plus performants afin de décrire ces derniers en fonction de leur contenu multimédia notamment visuel.

La navigation a pour but de répondre au problème de la recherche d'un document précis ou d'un type de document en catégorisant et structurant la base à laquelle il appartient. Les méthodes de recherche d'images par le contenu procèdent, dans un premier temps, à une indexation de la collection, autrement dit, à l'extraction des descripteurs de bas niveaux des images (couleur, texture, forme,...) afin de pouvoir établir un critère de similarité entre elles. Pour chaque requête de l'utilisateur, une distance de similarité est calculée en fonction de ces descripteurs et les images à priori inconnues. Deux contraintes principales sont imposées par le problème de l'indexation et la

recherche d'images par le contenu: la première concerne le temps de calcul, du fait que le calcul de similarité se fait en parcourant toute la base d'images. La deuxième concerne la taille de stockage de la base d'images.

Plusieurs méthodes de classification de base d'images existent, et apportent une aide précieuse dans la résolution de ce type de problème en découpant la base en groupes d'images similaires. Une fois que la base d'images est partitionnée, les systèmes de visualisation choisissent de définir une image représentative pour chaque groupe, plutôt que de représenter la totalité de la base. Ainsi, l'utilisateur peut naviguer rapidement et efficacement dans la base d'images. De manière générale, les problèmes de classification s'attachent à déterminer des procédures permettant d'associer une image à une classe. Ces problèmes se déclinent essentiellement en deux variantes : la classification dite « supervisée » et la classification dite « non supervisée ». Dans l'approche de la classification supervisée, les classes existent a priori. Par opposition, dans l'approche de classification « non supervisée », on dispose au départ d'un ensemble d'objets non étiquetés. A partir de ces données, l'idée est de parvenir à détecter des objets similaires afin de les regrouper. Un clustering (segmentation) sera jugé satisfaisant si on obtient en

sortie de la méthode des groupes d'images similaires [NAC 98a] [DEL 02] [GIL 00].

Le terme de Data Mining signifie littéralement: forage de données. Comme dans tout forage, son but est de pouvoir extraire des connaissances à partir des données qui peuvent être stockées dans des entrepôts, dans des bases de données distribuées ou sur Internet (Web Mining). Ces concepts s'appuient sur le constat qui existe au sein de chaque entreprise des informations cachées dans le gisement de données; ils permettent grâce à un certain nombre de techniques spécifiques, de faire apparaître des connaissances [BRA 03] [NAC 98b] [ABD 02]. Le Data Mining offre des moyens pour aborder les corpus en langage naturel (Text Mining), les images (Images Mining), le son (Sound Mining) ou la vidéo et dans ce cas on parle plus généralement de Multimédia Mining.

Dans cet article, nous nous intéressons à l'une des méthodes de segmentation du data Mining [KAR 99] appelée CHAMELEON. Cette dernière est réalisée sur des images stockées en arbre quaternaire Générique. Cette structure minimise le stockage des images similaires organisées en arbre quaternaires par partage de partie communes entre elles.

1. Méthodes de segmentation

La segmentation est la division des données dans des groupes d'objets similaires. Chaque groupe, appelé segment (cluster), est composé d'objets similaires entre eux et dissimilaires des autres groupes d'objets. Elle consiste à former des groupes homogènes à partir d'un ensemble de données hétérogène ainsi, il n'y a pas de classe à expliquer ou de valeur à prédire définis a priori, Il appartient ensuite à un expert du domaine de déterminer l'intérêt et la signification des groupes ainsi constitués.

1.1. Principe de la segmentation

Le principe général de tout système de clustering est de maximiser la similarité intra-classe (à l'intérieur d'un cluster) et de minimiser la similarité inter-classe (entre cluster). Traditionnellement, la segmentation est généralement divisée en hiérarchique et partitionnement. Les algorithmes de partitionnement démarrent avec un nombre fixe de segments. Leur principe est de découvrir des segments par transfert d'itération de points entre groupes, ou d'identifier les segments qui ont une haute densité des données. Pour les algorithmes de la première classe on cite (PAM, CLARA et CLARANS) [RAY 94]. Par opposition aux algorithmes de partitionnement, Les algorithmes hiérarchiques construisent les segments graduellement. La segmentation hiérarchique est aussi subdivisée en agglomération et division. BIRCH, CURE et CHAMELEON sont des algorithmes bien adaptés au data mining.

1.2. La Segmentation Hiérarchique

En principe, il existe deux classes de méthodes de segmentation hiérarchique:

La segmentation descendante: Qui démarre avec un cluster réunissant tous les objets, puis va diviser les clusters jusqu'à ce qu'un critère d'arrêt soit satisfait,

La segmentation ascendante: Dont le but est de former une hiérarchie de clusters, telle que plus on descend dans la hiérarchie, plus les clusters sont spécifiques à un certain nombre d'objets considérés comme similaires; Afin d'apporter des améliorations pour le clustering hiérarchique ascendant plusieurs algorithmes ont été définis dans le domaine du data mining. Tels que: BIRCH (Balanced Iterative Reduced and Clustering using Hierarchises) [ZHA 98], CURE (Clustering Using Representatives) [GUH 98] et CHAMELEON (A Hierarchical Clustering Algorithm Using Dynamic Modelling) [KAR 99].

L'algorithme de BIRCH n'est pas complexe mais il ne permet pas de détecter les clusters de formes ou de tailles différentes. D'un autre côté CURE ne considère pas l'inter-connectivité des deux clusters, et par conséquent il peut y avoir une pénalisation de deux groupes homogènes qui ont des points non condensés. Pour remédier à ces inconvénients on doit utiliser une distance Inter-connectivité et proximité relative pour prendre en compte le rapport de dispersion des points à l'intérieur d'un groupe. Pour cela, la méthode CHAMELEON proposée consiste en la maximisation de ses deux mesures de similarité entre deux clusters.

Dans ce travail, nous intégrons la méthode CHAMELEON dans un système d'indexation et de recherche d'image par le contenu qui a été développé en 2005, appelé REQUIT [BEL 05]. Ce dernier permet de représenter chaque image par un arbre quaternaire [VAN 93] dont le critère de découpage est l'homogénéité de la couleur, ce qui permet de stocker l'ensemble des images en arbre quaternaire générique. Cette structure [MAN 00] minimise l'espace de stockage, par partage de parties communes entre images. Différentes opérations sur l'image où entre images sont proposées par ce système (la recherche globale, par région, par niveaux).

2. L'Arbre quaternaire

C'est une structure hiérarchique, dite aussi quad-tree, construite par division récursive de l'espace en quatre quadrants disjoints de même taille [VAN 93], en fonction d'un critère de découpage (exemple: homogénéité de la couleur) de telle sorte que chaque nœud de l'arbre quaternaire représente un quadrant dans l'image. L'image entière est le nœud racine, si une image n'est pas homogène par rapport au critère de découpage, le nœud racine de l'arbre quaternaire représentant l'image a quatre fils représentant les quatre premiers quadrants de l'image, un nœud d'arbre quaternaire est dit feuille, si le critère d'homogénéité est vérifié sinon le nœud est interne. La similarité entre images est calculée en fonction des trois distances T, Q, V [BEL 05].

2.1. Définition de la distance entre images

La distance Δ est une distance entre images représentées par des arbres quaternaires. La distance Δ entre deux images i et j est définie par une somme de distance $\delta_k(i,j)$ entre les nœuds des arbres quaternaires représentant les images i et j , pondérées par des coefficients $C_k > 0$:

$$\Delta(i,j) = \sum C_k \delta_k(i,j) / \sum C_k \quad (1)$$

- $\delta_k(i,j)$ est une distance normalisée entre les nœuds homologues k des arbres quaternaire i et j définie précédemment.

- k est l'identificateur d'un nœud pris parmi l'union des identificateurs de nœuds apparaissant dans les arbres quaternaires des images i et j .

- C_k est un coefficient positif représentant le poids du nœud k dans le calcul de la distance C_k . Chaque poids C_k est choisi selon l'importance qu'on souhaite donner à certains quadrants d'image par rapport à d'autres dans le calcul de la distance Δ (on peut même donner à l'utilisateur la possibilité de choisir lui-même les poids de certains quadrants). Par exemple, si certains quadrants ne doivent pas apparaître dans le calcul de Δ (le cas de la distance par région ou par niveau), alors ils peuvent être associés à un poids nul. Si la surface des quadrants doit entrer en jeu dans le calcul de Δ , alors chaque coefficient C_k doit être proportionnel à la surface représentée par le quadrant par rapport à l'image entière.

2.1.1. Cas particuliers de la distance Δ

En fonction des différents poids C_k associés aux nœuds et de la distance choisie entre les nœuds, plusieurs familles de distances peuvent être définies à partir de la distance Δ . Nous définissons deux familles de distances basées sur la structure des arbres quaternaires, appelées **T-distance** (**T** pour Tree) et **Q-distance** (**Q** pour quadrant), et une famille de distances de similarité visuelle entre images, appelée **V-distance** (**V** pour visuel). Les deux premières familles de distances comparent les arbres quaternaires représentant les images. La dernière famille compare visuellement les images en utilisant leur représentation en arbre quaternaire [MAN 02].

3. Méthode CHAMELEON

C'est un algorithme de classification hiérarchique agglomérative qui utilise une modélisation dynamique pour l'agrégation de classes. A travers cet aspect dynamique, le but est de pouvoir retrouver des classes avec des formes irrégulières et des tailles différentes, Il s'agit de créer un ensemble initial important de clusters, puis les fusionner selon une nouvelle mesure. L'algorithme possède deux étapes (phases).

Phase 01. Une matrice de similarité est calculée. Celle-ci regroupe toutes les similarités entre les images prises deux à deux. A partir de cette matrice, un graphe des k plus proches voisins (k-ppv) sera, dans un premier temps, construit. Il existe un lien entre l'image i et l'image j si et si seulement si

l'image i fait partie des k-ppv de l'image j et l'image j fait partie des k-ppv de l'image i . La valeur de k étant un paramètre de l'algorithme. Le graphe des k-ppv permet de diminuer le temps de calcul pour la suite de l'algorithme, car seule une partie des données initiales est utilisée. Une fois le graphe des k-ppv créé, les auteurs appliquent dans un second temps un algorithme de partitionnement de graphe appelé Hmétis pour créer un ensemble de classes initiales.

Phase 02. Consiste en une combinaison itérative de l'ensemble de classes initiales en utilisant un nouvel algorithme de classification hiérarchique. Cet algorithme repose sur deux mesures pour regrouper deux classes: l'inter connectivité relative (Relative Interconnectivity ou **RI**) et la proximité relative (Relative Closeness ou **RC**).

Inter-connectivité Relative RI : L'inter-connectivité relative entre deux classes est la valeur absolue de l'inter-connectivité entre ces deux classes normalisées ($EC_{\{c_i, c_j\}}$) par la moyenne arithmétique de l'inter-connectivité de chaque classe (EC_{c_i}). Ainsi l'inter-connectivité entre deux classes C_i et C_j est définie de la façon suivante :

$$RI(C_i, C_j) = \frac{|EC_{\{c_i, c_j\}}|}{\frac{1}{2}(|EC_{c_i}| + |EC_{c_j}|)} \quad (2)$$

$EC_{\{c_i, c_j\}}$: Somme des arcs du graphe des k-ppv regroupant les classes.

EC_{c_i} : Somme minimum des arcs pour diviser en deux la classe C_i .

Proximité relative RC : Les concepts évoqués pour la proximité relative sont analogues à ceux définis pour l'inter-connectivité relative. La proximité absolue entre deux classes C_i et C_j est la moyenne pondérée des arcs (alors que l'inter-connectivité absolue est la somme des arcs) qui connectent un élément de C_i à un élément de C_j .

$$RC(C_i, C_j) = \frac{\bar{S}_{EC_{\{c_i, c_j\}}}}{\frac{|C_i|}{|C_i| + |C_j|} \bar{S}_{EC_{c_i}} + \frac{|C_j|}{|C_i| + |C_j|} \bar{S}_{EC_{c_j}}} \quad (3)$$

- $|C_i|$: Nombre d'éléments de la classe C_i .

- $S_{EC_{\{c_i, c_j\}}}$: Moyenne des poids des arcs connectant les sommets de C_i à ceux de C_j ;

- $S_{EC_{c_i}}$: Moyenne des poids des arcs reliant les deux sous classes du cluster C_i .

Pour regrouper deux classes en utilisant ces deux mesures, plusieurs approches sont possibles, pour notre part, nous utilisons une approche qui maximise une fonction combinant les deux mesures, donnée par

$$RI(C_i, C_j) \times RC(C_i, C_j)^a \quad (4)$$

Le but de cette fonction consistant à regrouper deux classes en fonction des deux mesures. La valeur de α va accroître l'importance d'une mesure par rapport à l'autre. Si $\alpha=1$, alors les deux mesures ont la même importance. Par contre, si on veut donner plus d'importance à la proximité relative, alors on prend $\alpha > 1$; pour $\alpha < 1$ alors on donne plus d'importance à l'inter-connectivité relative.

3.1. Application de CHAMELEON

L'exemple suivant va faire l'objet de démonstration de l'adaptation de la méthode CHAMELION dans notre prototype. L'exemple ne contient que 11 images satellites (Fig.1), par la suite nous donnons des résultats pour des bases d'images de tailles plus importantes.

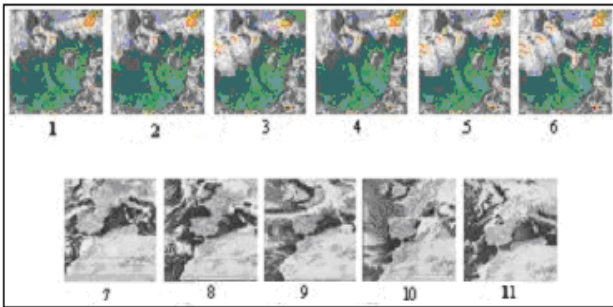


Fig. 1 Exemple D'une Base D'images Généraliste.

Dans notre prototype, on peut dérouler la méthode avec l'une des trois distances T, Q ou V pour calculer la distance entre une paire d'images i et j, représentée sous forme d'une matrice et de réaliser des opérations telles que la construction du graphe k-ppv, à partir duquel le graphe de partitionnement est conçu et représente en réalité les classes initiales.

Lorsque la matrice des similarités est calculée, on peut tracer le graphe k-ppv. Dans notre application le graphe est représenté sous forme d'un tableau (TAB.1) où chaque ligne représente une classe (une image avec ses k plus proches images). La matrice des distances nous permet de construire le graphe des k-ppv illustré dans la FIG.2 avec k=2.

Classes	Images	1 ^{ier} voisin	2 ^{ième} voisin
C1	1	2	4
C2	2	4	1
C3	3	5	2
C4	4	2	1
C5	5	3	6
C6	6	5	3
C7	7	10	9
C8	8	9	10
C9	9	8	10
C10	10	9	11
C11	11	10	8

Tab.1 Les Classes Du Graphe 2 – Ppv.

Comme le montre la FIG.2 il existera un lien entre deux images s'il fait partie de ces deux plus proches images à titre d'exemple : pour l'image 1 les deux plus proches images sont 2 et 4. (arc en bleu).

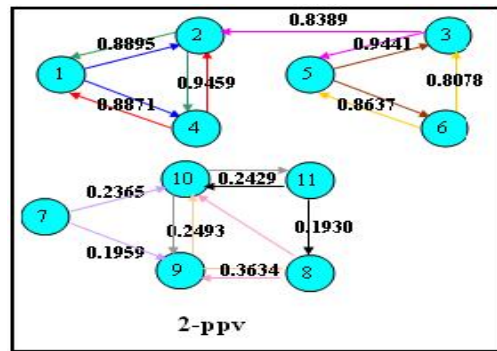


Fig.2 Graphe 2-Ppvs Du Tableau 3.2.

Chaque couleur d'arc dans la FIG.2 représente une classe dans le graphe 2-ppv. On passe maintenant à la construction de classes initiales (voir la FIG.3), qui illustre le principe de partitionnement entre quatre classes du graphe 2-ppv de la FIG.2.

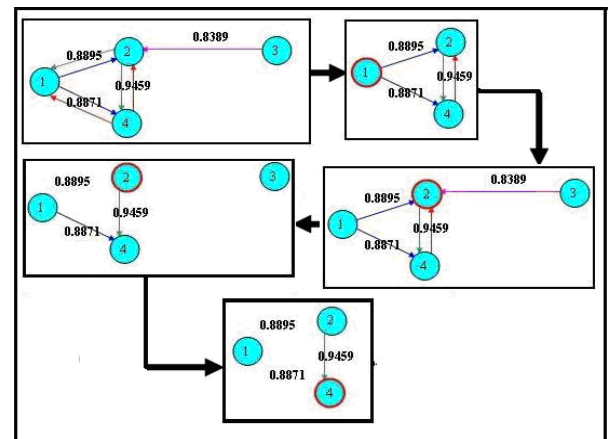


Fig.3 Application D'algorithme De Partitionnement Sur Le Graphe De La Fig.2.

La FIG.4 Présente quatre classes inter connecté entre elle (arcs vert, bleu, rouge, et rose).

L'algorithme de partitionnement parcourt les classes du 2-ppv pour chaque classe on vérifie la présence de ses éléments dans les autres classes, élément par élément.

Commençant par la classe 1 (a) ayant les images 1, 2 et 4 (un lien entre 1 et 2, 1 et 4) ; on sélectionne le premier élément de la classe en cours ; on constate que l'élément (1) appartient à 3 classe (arcs bleu (1, 2), arcs vert (2, 1), rouge(4, 1)) ; on a trouver deux liens plus fort ;le lien (1, 2) de la classe en cours dont il est propriétaire et le lien (2, 1) de la deuxième classe (arc en vert) qui sont identique(0,8895) ; dans ce cas il sera éliminé de la 2^{ième} classe est maintenu dans ça propre classe (b).

Concernant le deuxième élément (2) il appartient à quatre classes différentes (c) (arcs vert, bleu, rouge,

rose) deux liens plus fort trouvé, le lien (2, 4) arc en vert et le lien (4, 2) arc en rouge qui ont la même valeur (0,9454). On garde le lien (2, 4) arc en vert de la classe dont l'élément (2) est propriétaire. Le troisième élément (4) appartient à deux classe (d) (arc bleu et vert), le lien le plus fort est (2, 4) de la deuxième classe (e) qui sera maintenu avec la valeur 0,9454.

On réitère le même procédé pour les autres classes du graphe 2.-ppv. Les classes initiales résultantes. Sont présentées dans FIG.4.

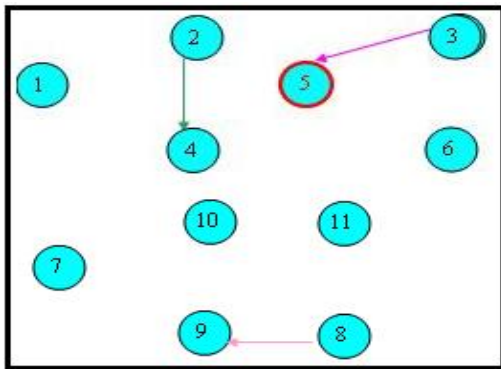


Fig.4 les classes initiales sous forme de graphe.

Afin de fusionner les paires de classes les plus proches, on calcule les deux mesures de similarité l'inter-connectivité relative et la proximité relative.

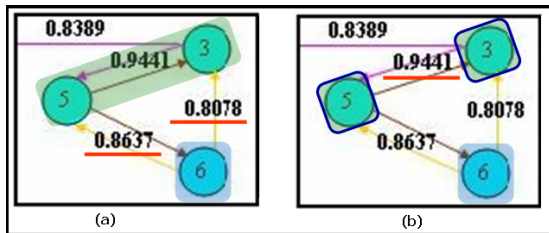


FIG.5 Deux Classes Initiales.

Pour calculer l'inter-connectivité relative on doit d'abord calculer l'inter-connectivité absolue et l'inter connectivité interne :

Inter - connectivité absolue. : Est calculer entre toutes les paire de classe ; pour la paire de classe C₃, C₄ ; C'est la somme des poids des arcs [(5, 6) et (6, 5)] et [(6, 3)] qui les relie de la fig.5 (a) :

$$EC \{c_3, c_4\} = (0.8637 * 2) + (0.8078) = 2.535.$$

Inter-connectivité intern. : Chaque classe étant coupée en deux sous-classes de taille identique, le calcul de l'inter-connectivité interne consiste à additionner les poids des arcs reliant les deux sous-classes ; A titre d'exemple, la classe 3 contient deux élément ces deux sous-classes seront alors C₃₁ avec l'élément (5) et la C₃₂ avec l'élément (3) qui sont reliés à travers deux arcs [(5, 3) et (3, 5)] de poids identique : $EC_{c_3} = 0.9441 * 2 = 1.88$. Par conséquent, si une classe ne comporte qu'un élément dans cette classe alors son inter-connectivité interne est égale à zéro, à titre d'exemple la classe (4) :

$$EC_{c_4} = 0.$$

Pour calculer la proximité relative on doit d'abord calculer la proximité absolue et la proximité interne :

Proximité absolue : Par déduction la proximité absolue est la moyenne de l'inter-connectivité relative. $S_{EC \{c_3, c_4\}} = EC \{c_3, c_4\} / \text{nombre d'arcs qui les relie} = 2.535 / 3 = 0.845$.

Proximité interne : Même principe la proximité interne est la moyenne des poids des arcs reliant les deux sous-classes. On peut aussi la calculer en fonction de l'inter-connectivité interne sur le nombre d'arcs qui relie les deux sous classes. La proximité interne de la classe (3) : $S_{EC_{c_3}} = EC_{c_3} / \text{nombre d'arcs qui relie les deux sous-classes} = 1.88 / 2 = 0.944$.

Le calcul de la proximité relative se fait entre chaque paire de classes ; La proximité relative des deux classes C₃ et C₄ :

$$RC(C_3, C_4) = \frac{0.8389}{\frac{|3|}{|1|+|2|} * 0.944 + \frac{|1|}{|1|+|2|} * 0} = 1.343$$

Idem pour les paires de classes restante, pour décider de la fusion d'une paire de classes C_i, C_j on a utilisé l'approche qui maximise la fonction :

$RI(C_i, C_j) \times RC(C_i, C_j)^\alpha$ en fonction des deux mesures calculées précédemment. Pour notre exemple :

$$RI * RC_{\{c_3, c_4\}}^\alpha = 2.865 \times 1.343 = 3.606 \text{ avec } \alpha = 1.$$

$RI * RC_{\{C_i, C_j\}}^\alpha$	CI ₁	CI ₂	CI ₃	CI ₄	CI ₅	CI ₆	CI ₇	CI ₈
CI ₁	- /	-5.2920	0	0	0	0	0	0
CI ₂	5.292-	- /	-0.394	0	0	0	0	0
CI ₃	0	0.394-	- /	-3.606	0	0	0	0
CI ₄	0	0	3.606-	- /	0	0	0	0
CI ₅	0	0	0	0	- /	-0.4314	0	0
CI ₆	0	0	0	0	0.4314-	- /	-1.956	0.4266
CI ₇	0	0	0	0	0	1.956-	- /	0
CI ₈	0	0	0	0	0	0.4266	0	- / -

TAB.2 Similarité a maximisé (matrice combinatoire)

Deux classes peuvent être fusionnées s'ils ont la même valeur maximale ; exemple des paires de classes ayant la même valeur maximale du TAB.2, sont : (CI₁ = {CI₁, CI₂}, CI₂ = {CI₃, CI₄}) ; par contre la valeur maximale de la classe CI₅ est de 0,4314 avec la classe CI₆ mais la valeur maximale de la classe CI₆ est avec la classe CI₇ de 1,956 et la classe CI₇ est aussi la même valeur maximale avec cette dernière donc la classe C₆ sera fusionnée avec la classe CI₇. Les classes résultantes de la première itération sont :

$$(CI'_1 = \{CI_1, CI_2\}, CI'_2 = \{CI_3, CI_4\}, CI'_3 = CI_5, CI'_4 = \{CI_6, CI_7\}, CI'_5 = CI_8).$$

On arrête le processus de regroupement des classes intermédiaires une fois que la matrice combinatoire ne contient que des zéro ; pour cet exemple les classes finale sont obtenue au bout de la troisième itération FIG.6.

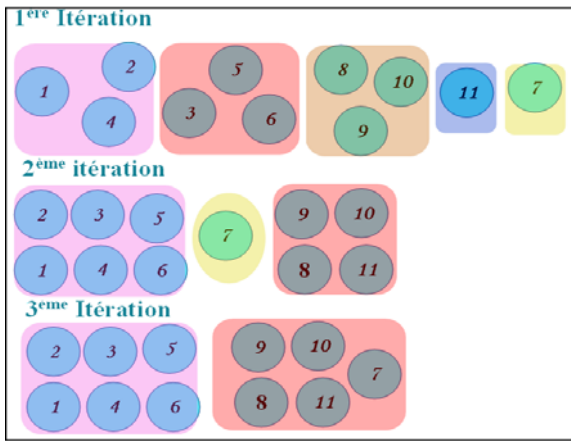


FIG.6 Les Classes Résultantes De Chaque Itération.

4. Expériences, resultants et critiques

La méthode CHAMELEON est validée sur une base d'images représentées sous forme d'arbres quaternaires [BEL 05] la similarité entre deux images est définie par la distance (T , Q ou V) entre les arbres les représentants. Une fois que les groupes d'images sont conçus, chaque groupe d'images va faire l'objet d'un arbre quaternaire générique.

La FIG 7 montre un exemple d'application de notre prototype de base d'images. Les classes finales résultantes pour cet exemple sont présentées dans la Fig.8 (les classes (c) et (g)) sont représentées dans deux classes séparément, cela revient au choix de la valeur de k ; si on veut que les classes soient fusionnées, on augmente la valeur de k (FIG.9); la même chose pour les classes (d) et (h) de la FIG.8 qui sont aussi proches.

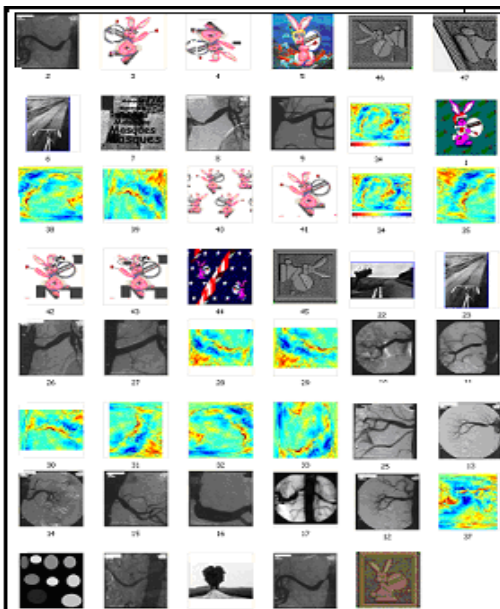


FIG.7 Base d'images Généraliste.

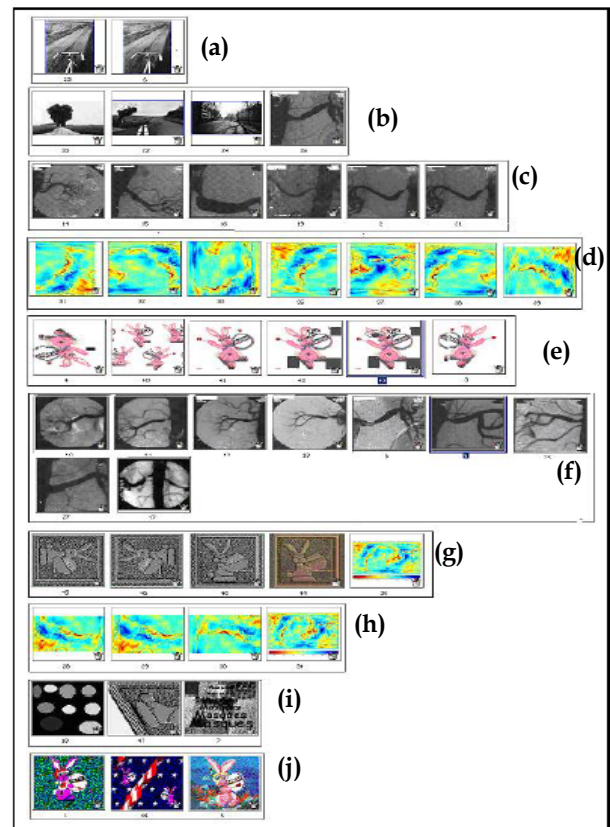


FIG.8 les classes finales avec $k=2$.

Après plusieurs tests, on a constaté que le graphe des k -ppv influe considérablement sur la construction des classes initiales. La construction du graphe k -ppv repose sur la matrice des similarités. Lors de sa construction l'algorithme prend pour chaque image les k premières images considérées comme proches sans prendre en compte que la distance de similarité soit faible ou forte. On remarque aussi que si le nombre k dépasse le nombre d'images similaires à une image donnée cela va influencer négativement sur la construction des classes initiales, et par conséquent, on peut trouver des images divergentes dans une même classe (voir la classe (g) de la FIG.9). Pour éviter cette faille, et afin d'améliorer cette méthode on a ajouté un seuil appelé seuil k ppv ayant pour objectif de limiter la plage des plus proches voisins. Ce dernier est vérifié lors de la construction du graphe k -ppv.

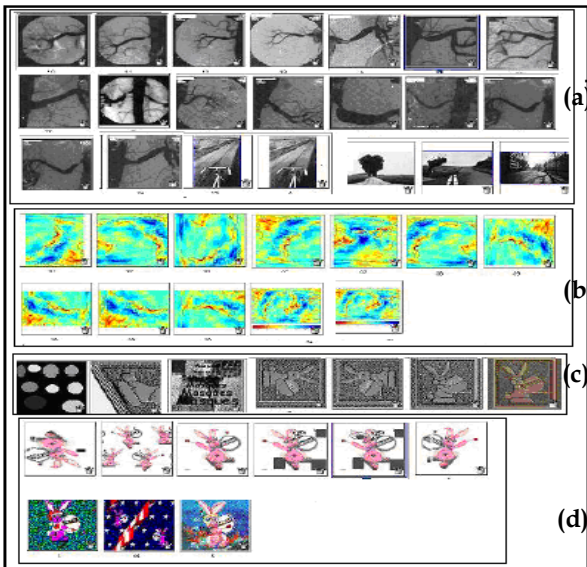


FIG. 9 Les classes finales pour k= 5 et seuilppvs =40.

Plusieurs essais ont été appliqués sur notre prototype. La Fig.10 donne des résultats pour une base d'images. Des valeurs de k sont choisies de zéro à dix et, pour chaque valeur de k, elle est testée avec 10 valeurs différentes du seuil *seuilppvs*. Le temps d'exécution de ces expériences prend seulement six minutes pour chaque expérience, sur un PC Pentium II, 400 MHz.

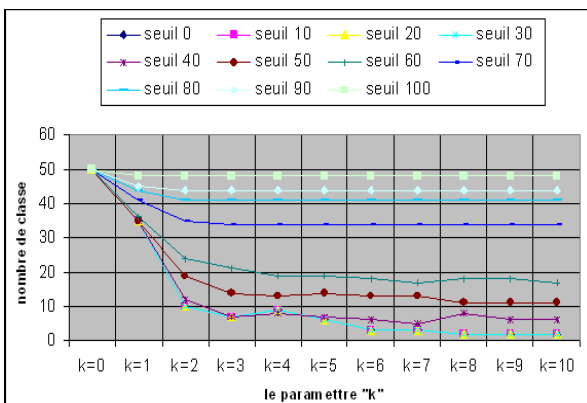


FIG.10 L'influence du seuilppvs et paramètre k sur les classes finales.

Le découpage de l'image en quadrant suivant l'organisation en arbre quaternaire se fait selon un critère particulier (ex : homogénéité de la couleur des pixels). Lorsque l'image est en niveau de gris, et/ou contient des nuances de couleurs, le découpage de l'image par ce critère peut aller jusqu'au niveau des pixels ; dans ce cas, la taille de l'image sera plus grande que la taille au format BMP (FIG.11, FIG.12). Pour remédier à cela, il est possible de fixer un seuil d'homogénéité pour limiter ce découpage.

Dans la FIG.12, un seuil d'homogénéité choisi a partir de 5, donne des tailles de fichiers au format QT plus petites que ceux enregistrés au format BMP (par rapport a l'image1). Par contre, pour l'image 2, la FIG.13 indique des tailles de fichier au format QT

plus petites que ceux enregistrés au format BMP pour un seuil d'homogénéité choisi à partir de 50.

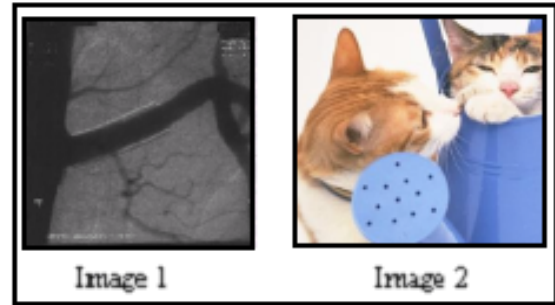


FIG.11 Exemple d'images.

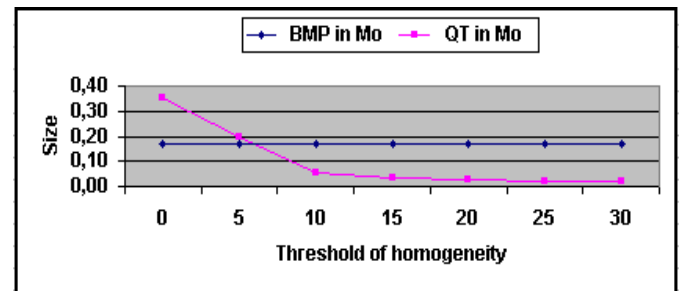


FIG.12 L'influence du seuil d'homogénéité sur la taille de l'image1 (FIG 11).

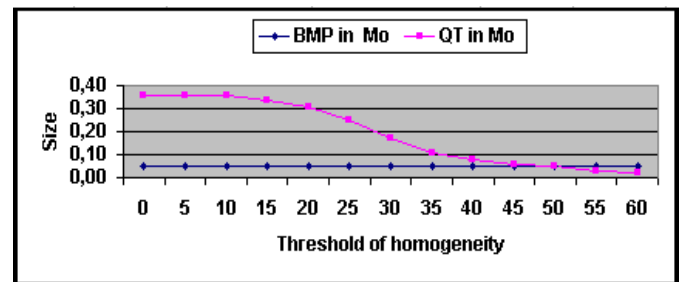


FIG. 13 l'influence du seuil d'homogénéité sur la taille de l'image2 (FIG 11).

La FIG.14 comporte trois images format QT (a), (b) et (c) avec différentes valeurs du critère d'homogénéité de la couleur ((a) : 10%, (b) : 25% et (c) : 30%), et la Fig.15 montre l'amélioration de la taille de l'image1 selon le critère d'homogénéité de la couleur. On remarque que le meilleur résultat en gain d'espace est de 89,64% pour l'image(c) (FIG.15), mais on perd en qualité d'image, l'image (a) donne un meilleur résultat avec 53,92% en gain d'espaces.

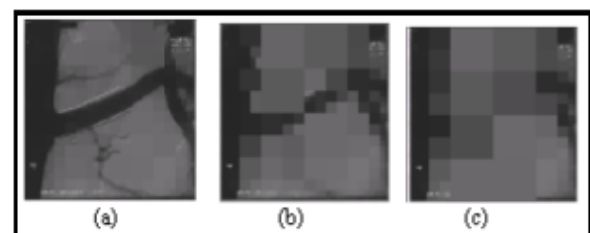


FIG.14 Les résultats obtenus avec différentes valeurs du seuil d'homogénéité pour l'image1.

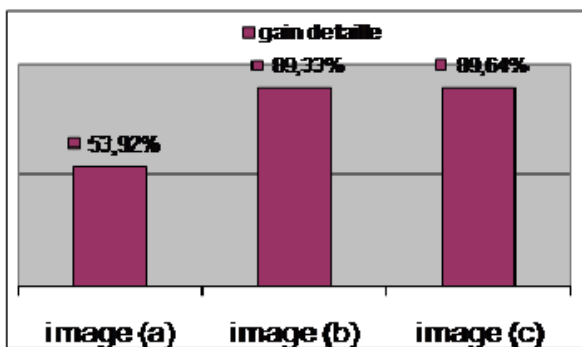


FIG.15 Gain d'espace selon le critère d'homogénéité pour l'image1.

Les paramètres de la FIG.14 sont employées avec l'image2 (FIG.11), et la FIG.16 donne les images résultantes avec différentes valeurs du critère d'homogénéité, et la FIG.17 montre l'amélioration des tailles. Nous avons noté que le meilleur résultat en gains d'espace de 91,49% pour l'image (c) (FIG.15), mais avec perte en qualité d'image, pour l'image (a) le meilleur résultat en gain d'espace est de 71,66%.

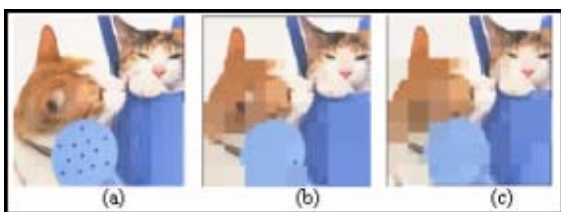


FIG.16 Les résultats obtenus avec différentes valeurs du seuil d'homogénéité pour l'image2.

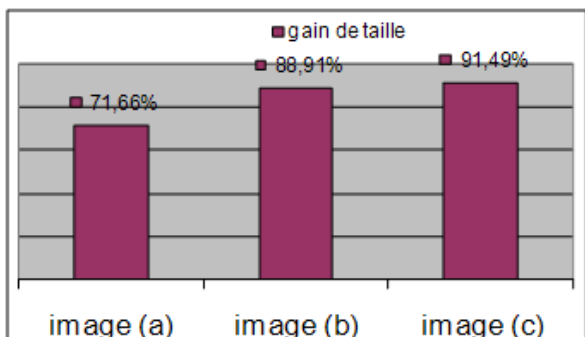


FIG.17 Gain d'espace selon le critère d'homogénéité pour l'image2.

La Fig.18 présente l'amélioration des tailles entre le système REQUIT [BEL 05] et notre nouveau système ARICHA appliqué à quatre différentes bases d'images. Les meilleurs résultats en taille du format Qt sont donnés par ARICHA (Amélioration De La Recherche D'images Par Le Contenu Intégration De La Méthode CHAMELEON), cela est dû aux modifications apportées aux structures de données utilisées dans ce format.

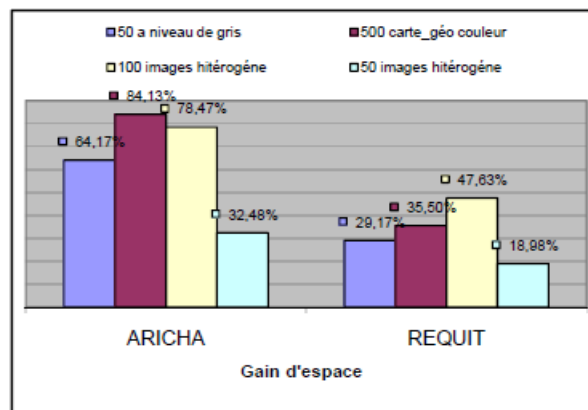


FIG.18 Amélioration des tailles entre le système REQUIT et notre nouveau système ARICHA appliqué à quatre différentes bases d'images.

Dans notre prototype, la recherche d'image similaire procède dans un premier temps à la recherche de la classe la plus similaire, puis on cherche les images similaires à l'image requête dans la classe sélectionnée. Le temps de la recherche d'images similaires dans une base généraliste de mille images avant la segmentation est de 2132,35s le temps de recherche a été réduit de 216,13s après application de notre prototype (22,23s pour la recherche de la classes et 193,9s pour la recherche des images dans la classe) soit un rapport de 10.

5. Conclusion

Après l'application de la méthode CHAMELEON, la recherche de similarité d'une image requête par rapport aux images stockées dans la base passe par deux étapes: La première étape consiste à calculer la similarité de l'image requête avec les images représentant chacune leurs classes, afin de repérer la classe la plus proche à l'image requête, en utilisant l'une des distances de similarité T, Q ou V (parcours de l'ensemble d'images résumées). La seconde étape consiste en la recherche d'images similaires dans le groupe de la classe dont elle est la plus proche.

Le nombre d'itération pour les regroupements des paires de classe dépend du paramètre k, en augmentant k, le nombre d'itération diminue. Le choix du critère de découpage dépend du domaine d'application, de l'expert et de la nature des images traitées, ainsi le choix du paramètre k de l'algorithme est aussi expérimental si le nombre k est faible on aura plus de classe plus de précision, est vice versa d'un coté; de l'autre l'apparence du paramètre *seuil_kppv* qui limite la plage des plus proches voisins, nous a évité de tomber sur des images divergentes en terme de distance dans une même classe au moment de construction du graphe des k-ppv sur lequel se base le graphe Hmétis pour la construction des classes initiales.

Références

- [NAC 98a] D.NACKACHE « *Data warehouse et Data Mining* », Conservatoire National des arts et Métiers de Lille, juin 1998.
- [DEL 02] J.M.DELORME « *L'apport de la fouille de données dans l'analyse de texte* », Conservatoire National des Arts et Métiers, Centre régional de Montpellier, avril 2002.
- [GIL 00] R.GILLERON, M.TOMMASI, « *Découverte de connaissances à partir de données* », Université de Lille3, 2000.
- [BRA 03] A.BRAHMI, A.GUEDDACHE, « *Data Mining et SMA* »; exposé RFIA USTO », 2003.
- [NAC 98b] D.NACKACHE ; « *probatoire en ingénierie des systèmes décisionnels : Data Mining sur Internet* »; conservatoire National des arts et métiers de Lille; 15-12-1998.
- [ABD 02] D. ABDELKADERZIRE, R. RAKOTOMALALA « *Extraction des connaissances à partir des données (ECD), Data Mining* » ; 2002.
- [KAR 99] G.KARYPIS, E.HAN, V.KUNER « *CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling* » 1999.
- [RAY 94] T.Ng RAYMOND and J.HAN, Member, « *CLARANS: A Method for Clustering Objects for Spatial Data Mining* » IEEE Computer Society », 1994.
- [ZHA 98] T. ZHANG, R. RAMAKRISHAN « *BRICH. An Efficient Data Clustering Method for Very Large Data Bases* » .Mirou 1998.
- [DJE 07a] S.DJERROUD, L.ZAOUI, « *Accélération de la recherche d'images stockées en arbre quaternaire dans les bases d'images généralistes par CHAMELEON* » , mémoire de magistère de l'université d'USTO Oran, soutenu le 02 juillet 2007.
- [DJE 07b] S.DJERROUD, L.ZAOUI, « *Improvement of Content Based Image Retrieval: intégration of CHAMELEON method* », 3emes journées internationales sur l'informatique graphique JIG'2007, université de Constantine, 28-29 octobre 2007.
- [DJE 07c] S.DJERROUD, L.ZAOUI, « *Accélération de la recherche d'images stockées en arbre quaternaire dans les bases d'images généralistes par CHAMELEON* », colloque internationale MOAD'07 METHODE ET OUTILS D'AIDE A LA DECISION, université de Béjaia 18-20 novembre 2007.
- [DJE 08] S.DJERROUD, L.ZAOUI, « *Amélioration de la recherche d'images par le contenu en utilisons une méthode issue du domaine du data mining* », 10ème Conférence Maghrébine de la technologie d'information (Conférence on Software Engineering and Artificial Intelligence) MCSEAI'08, université d'USTO Oran, 28-30 avril 2008.
- [GUH 98] S.GUHA, R.RASTAGI, K.KSHIM, « *CURE: An Efficient Clustering Algorithm for Large Data Bases* », 1998.
- [BEL 05] A.Belal, M.A.Djelil. « *Similarité entre images basées sur les arbres quaternaires* », PFE, Informatique Université Usto ,2005.
- [VAN 93] P.VAN. OOSTROM. « *Reactive Data Structures for Geographic Information systems* », Oxford University Press, ISBN: 0-19-823320-5, 1993.
- [MAN 00] M.Manouvrier. « *Objets de grande taille dans les bases de données* », Thèse de doctorat informatique, université de Paris, Jan 2000.
- [MAN 02] M.Manouvrier, M.Rukoz, G.Jomier « *Quadtree representations for storage and manipulation of clusters of images* », Image and Vision Computing, vol. 20, n° 7, 2002.