REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'enseignement supérieur et de la recherche scientifique

UNIVERSITE SAAD DAHLED BLIDA

Faculté des sciences

Département d'informatique.

Mémoire Présenté par :

- **Adidou Chaabane**
- **Salmi Mohamed Yacine**

Pour l'obtention du diplôme Master
Domaine : Mathématique et Informatique.
Filière : Informatique.
Spécialité : Informatique.
Option : Ingénierie du logiciel.

Sujet :

COMBINAISON DES DESCRIPTEURS VISUELS POUR LA RECHERCHE PAR CONTENUE D'IMAGE EN UTILISANT L'ARBRE QUATERNAIRE

Soutenue le : / /2013 devant le jury composé de :

Mr :	**Président**
Mr : Amine chérif Zahar	**Promoteur**
Mr : Ali Khalfi	**Encadreur**

Année universitaire 2012/2013

REMERCIEMENTS

Toute notre gratitude, grâce et remerciements vont à Dieu le Tout Puissant qui nous a donné la force, la patience, le courage et la volonté de mener à terme ce projet.

Nous remercions les membres de jury qui nous font l'honneur d'examiner ce modeste travail.

Toute notre gratitude va à tous les enseignants qui ont contribué à notre formation.

C'est avec une profonde reconnaissance et considération particulière que nous remercions nos de promoteurs Mr. CHERIF ZAHAR de nous avoir encadré ainsi que leurs disponibilités, orientations, et précieux conseil avec lequel ils ont suivi et guidé ce travail.

Et n'houblier pas nos chère Encadreur Khalfi Ali qui nous a donné les conseilles et la main d'œuvre.

En fin, tous nous amis, et ceux qui ont contribues de prés ou de loin pour achever ce modeste travail.

Dédicace

Je dédie ce modeste travail à :

ma chère mère

Qui m'a éclairé la vie par leur bonté et leur amour.

Mes amis auxquels je souhaite bonne chance dans leur Vie quotidienne et toute ma famille.

A tous mes amis et à tous ceux qui ont contribué à l'aboutissement de ce projet.

Adidou Chaabane

Dédicace

Je dédie ce modeste travail à :

Ma chère mère qui n'a jamais cessé de me soutenir et de m'encourager

Qui m'a éclairé la vie par leur bonté et leur amour.

A tous mes amis et à tous ceux qui ont contribué à l'aboutissement de ce projet.

Salmi Mohamed Yacine

Résumé

Notre projet consiste à realiser un système de recherche par le contenue d'images en utilisant une approche de combinaison des descripteurs visuel de bas niveau par l'utilisation de l'arbre quaternaire. Pour ce faire, dans chaque niveau de l'arbre il y a une extraction d'un vecteur visuel. Dans notre approche, chaque image de la base est représentée par un arbre quaternaire de trois niveaux obtenus par une décomposition récursive de l'image. Nous avons combiné trois descripteurs de texture, couleur et contours. Pour la couleur, nous avons utilisés l'histogramme scalable de couleur, la texture par l'extraction d'histogramme d'orientation des contours sachant que les deux derniers descripteurs sont de la norme MPEG 7, en ajoutant la détection des contours par l'approche de convolution avec un filtre gaussien d'opérateur de Sobel. Dans cette approche, la similarité est calculé selon un algorithme de calcul de distance entre les arbres quaternaires afin d'avoir une distance finale qui sera utilisée pour mesurer la similarité entre l'image requête et les images de la base.

Mots clé : arbre quaternaire, descripteurs visuel, MPEG 7.

Abstract

In this project we will realize CBIR system (Content-based image retrieval) by combining different low-level visual descriptors using QuadTree. For each depth of the QuadTree we extract a visual descriptor, in our feature each image is represented by QuadTree of three levels obtained after using recursive split of image. We were used three descriptors: texture, color and edge. For the color, we extract the scalable color descriptor of MPEG 7(SCD) standard and the edge Histogramme descriptor of the texture (EHD). Finally, the edges are detected by convolution approche using gaussian Filter with kernel of Sobel. In this feature the similarity is calculated by using algorithm of distance between QuadTrees to get final distance which is used for the matching between images.

Keyword: QuadTree, Visual descriptor, MPEG 7.

Sommaire

Sommaire

Chapitre 3 : L'arbre quaternaire

Chapitre 4 : Implémentation et test

Liste des Figures

Liste des tableaux

1. Introduction

A l'heur actuel les descripteurs sont utilisés dans plusieurs domaines d'application pour l'indexation et la recherche d'image par le contenu dans de larges bases de données notamment dans les grandes bases de donnes d'images .les chercheurs en analyse d'image ont concentrés leurs efforts a décrire le contenue visuel de l'image pour qu'il soit performant en terme de reconnaissance et pour avoir des résultats performants dans un système de recherche d'images par le contenu.

L'indexation automatique d'images directement à partir de leur contenu numérique est une des solutions possibles et prometteuses pour gérer efficacement les bases de données d'images numériques.

L'interrogation d'une base d'images peut s'appuyer sur des mesures de similarité entre les caractéristiques bas niveau d'une image requête proposée comme exemple, et le résultat attendu est celles des images présentes dans la base.

Les particularités requises pour l'indexation et la recherche d'image par le contenu dans plusieurs domaines d'applications proviennent, entre autres, des particularités des images elles-mêmes. Il existe une grande variété de modalités d'acquisition d'images en fonction des bases de données. Les images fournies dans les domaine de multimédia et de l'informatique sont très différentes en termes de résolution, contraste et rapport signal sur bruit. Elles sont très spécialisées et produisent des images porteuses d'informations différentes globalement et localement entre elles.

En plus des descripteurs visuels de bas niveaux tel que la couleur, le contour la texture ….qui décrivent l'image nous avons les méthodes arborescentes qui permettent également de structurer et présenter l'image dans une structure hiérarchique par décomposition spatiale de l'image en plusieurs région dans le cas par exemple de l'arbre R [1]ou en quatre quadrants disjoints en fonction d'un critère de découpage comme dans le cas de l'arbre quartenaire [2]

2. Problématique

Aujourd'hui l'apparence de la nouvelle technologie conduit vers la naissance des images de haute qualité. Pour l'utilisateur , ce type de bases de données d'image, la recherche de l'information est très couteuse et problématique, on a besoin d' une nouvelle technique de traitement des données. Dans ce cas , la recherche d'images par le contenu s'intéresse à découvrir des connaissances implicitement contenues dans un ensemble de données en s'appuyant sur différentes techniques qui peuvent être mises en œuvre. Ces techniques visent à décrire leur contenu, et à en extraire l'information la plus pertinente. La recherche d'images par le contenu consiste à caractériser le contenu visuel des images par des descripteurs visuels et d'effectuer des recherches par similarité visuelle à partir de ces descripteurs.

Dans ce contexte nous proposons de combiner les di scripteurs visuel couleur, forme et texture toute en exploitant les caractéristiques de l'arbre quartenaire

3. Objectif

Suivant à la problématique que nous avons citée au-dessus, on va utiliser un bon standard des descripteurs visuel pour caractériser les informations multimédias qui comportent la Norme **MPEG7**. Le but de notre travail est d'augmenter le niveau de pertinence de recherche en utilisant une combinaison des descripteurs de la norme **MPEG7** et d'autres descripteurs toute en exploitant la spécificité de l'arbre quaternaire. Dans notre cas, l'image estdevisée en 3 niveaux(Profondeur de l'arbre) etpour chacun on fait l'extraction d'un descripteur visuel parmi ceux-ci nous allons intéresser aux descripteurs suivant :

- **EOD** (Edge orientation descriptor) descripteur de contour.
- **EHD** (Edge histogramme descriptor) descripteur de texture.
- **SCD** (Scalable color descriptor) descripteur de couleur.

En fin, tous les résultatstrouvés par cette combinaison des descripteursseront mesuréeset évaluer par un protocole d'évaluation.

4. Organisation de Mémoire

Afin d'atteindre les objectifs cité ci-dessus, notre mémoire est organisé comme suite :

- Introduction général.
- Chapitre I : Principe des Système CBIR.
- Chapitre II : les descripteurs visuels.
- Chapitre III : l'arbre quaternaire.
- Chapitre IV : Implémentation et test.
- Conclusion.

Chapitre 1:
Principe des Systèmes CBIR

1. Introduction

La recherche d'images par le contenu est l'une des solutions possibles et prometteuses pour gérer les bases de données d'images numériques ,elles visent à résoudre ce problème en se basant sur un paradigme de représentation de bas niveau du contenu de l'image, par la couleur, la texture, la forme, etc.., et d'autres par une combinaison de celles-ci. Il est donc indispensable de développer des outils permettant de sélectionner les images les plus pertinentes par leur aspect visuel.

L'objectif de ce chapitre est de présenter le principe général des systèmes CBIR et aussi de citer quelque exemple des systèmes les plus connues . Puis on s'intéressera à l'étude des différentes approches d'indexation et de recherche d'images par le contenu. La littérature dans ce domaine étant très riches, nous allons présenter dans ce qui suit les techniques les plus utilisées.

2. Architecture de Système CBIR [ABE 07]

Deux aspects indissociables existent dans les systèmes de recherche d'images Par le contenu, l'indexation et la recherche.

- **La phase Indexation (Offline) :** Dans cette phase, des caractéristiques sont automatiquement extraites à partir de l'image et stockées dans un vecteur numérique appelé descripteur visuel. Grâce aux techniques de la base de données, on peut stocker ces caractéristiques et les récupérer rapidement et efficacement.
- **La phase recherche (Online):** Dans cette étape, le système analyse une ou plusieurs requêtes émises par l'utilisateur et lui donne le résultat correspond en une liste d'images ordonnées, en fonction de la similarité entre leur descripteur visuel et celui de l'image requête en utilisant une mesure de distance.

La figure1 schématise le fonctionnant d'un système de recherche et d'indexation d'images.

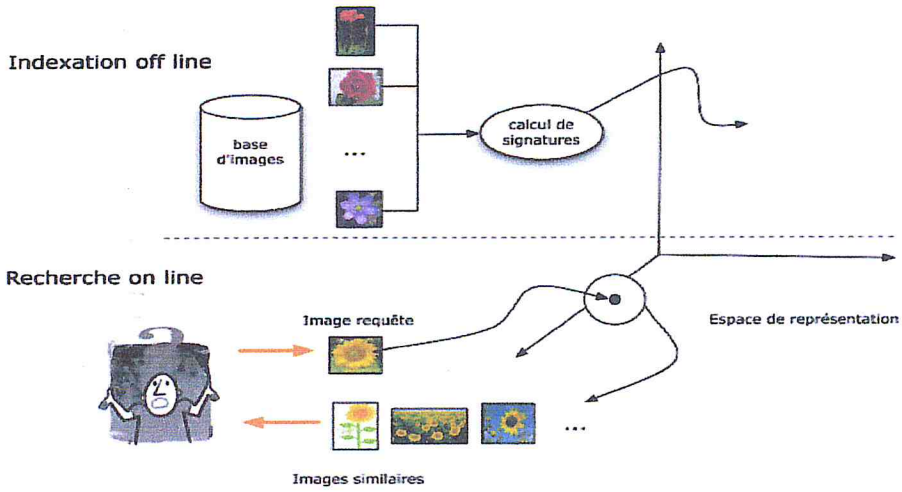


Fig.1.1. Le fonctionnant d'un système de recherche et d'indexation d'images.

3. Les types des requêtes

Il existe 3 façons de faire une requête dans un système d'indexation et recherche des images : soit une requête par mots clés, soit une requête par esquisse, soit une requête par exemple(image). La figure 2 donne une démonstration pour les trois façons.

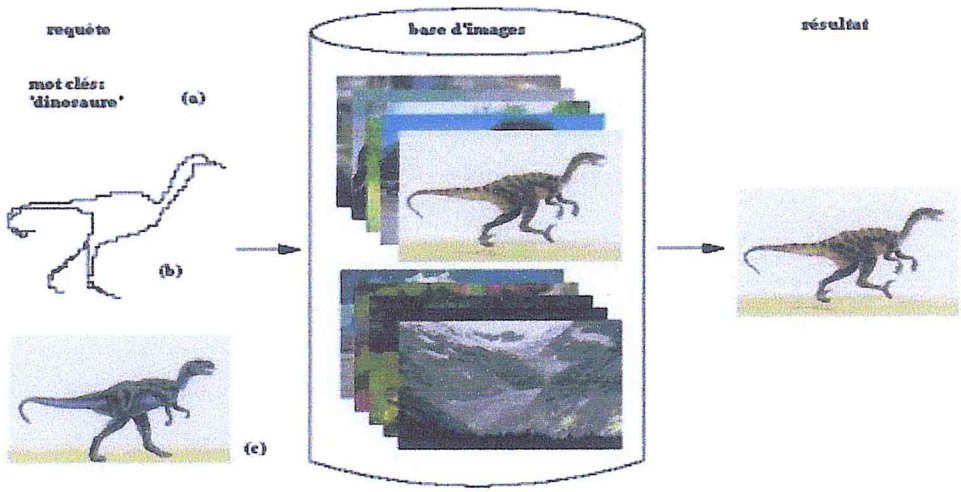


Fig.1.2. Les types des requêtes dans le Système CBIR.

3.1 .Requête par mot clé

Une des attentes des utilisateurs dans le domaine de recherche d'images se situe au niveau de sa sémantique c'est pour cela que la plupart des systèmes de recherche d'images développés utilisent des mots clés ou des descripteurs textuelles pour caractériser

chaque image de la base (ex : recherche d'images sur Internet). Ce type de caractérisation comporte un certain nombre d'inconvénients, en effet : La description textuelle est une opération longue, coûteuse et difficile à élaborer car l'information externe est manuellement attachée par l'utilisateur ce qui conditionne la qualité de recherche future, et puis elle ne décrit pas fidèlement le contenu de l'image car elle se fait de manière automatique à partir du nom, de la légende ou du texte qui l'entoure [ABE 07].

Dans ce types des requêtes les images sont recherchées suivant un ou plusieurs critères, par exemple trouver les images contenant 80% de rouge. Donc, le système se base sur l'annotation manuelle et textuelle d'images.

La figure3 illustre bien les inconvénients de ce type de requête. En effet l'utilisateur veut trouver des images qui contiennent une ou (des) voiture(s) avec le ciel cependant les premières images ne sont pas pertinentes.

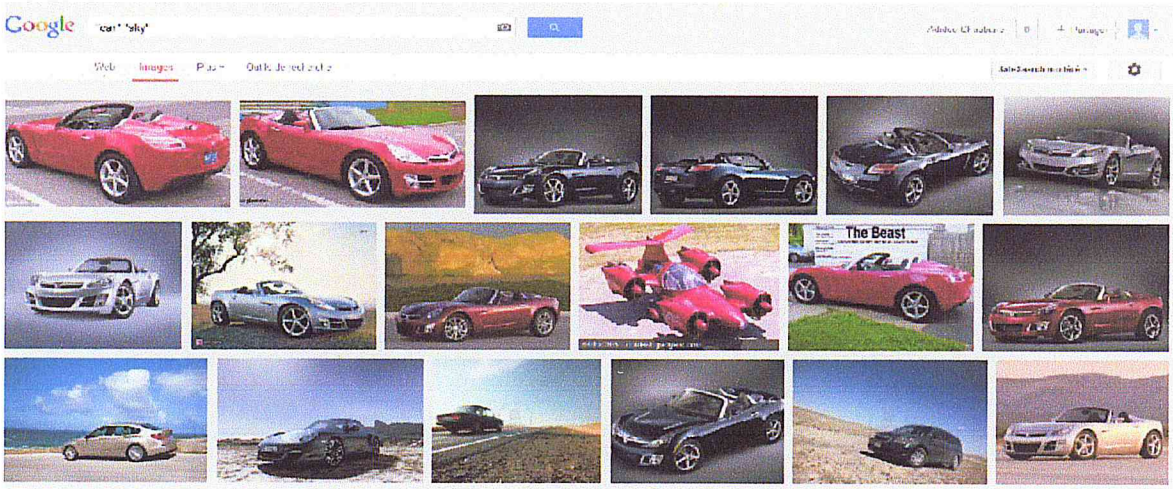


Fig.1.3. Un exemple de recherche d'images par mot clé dans Google.

Pour pallier aux inconvénients de la recherche par mots clés, une deuxième approche a été proposée : la recherche par le contenu.

3.2 Requête Par l'exemple (image)

Comme son nom l'indique, le principe de cette méthode est d'identifier des images à partir de leur contenu (c'est à dire à partir des données de l'image elles même et non à partir du texte associé aux images).

L'indexation des images, qui se fait automatiquement, nécessite l'extraction des paramètres de celles-ci au préalable. Ces paramètres "quantifient" la couleur, la texture,

l'intensité ou bien encore les formes contenues dans l'image et fournissent une "signature" de l'image [NAS 97].

Pour ces types des requêtes, l'utilisateur, pour représenter ses besoins, utilise une image (Ou une partie d'image) qu'il considère similaire aux images qu'il recherche. Cette image est appelée image exemple ou requête. L'image exemple peut soit être fournie par l'utilisateur, soit être choisie par ce dernier dans la base d'images utilisée.

3.3 Requête par esquisse

Dans ce cas, le système fournit à l'utilisateur des outils lui permettant de constituer une esquisse (dessin) correspondant à ses besoins. L'esquisse fournie sera utilisée comme exemple pour la recherche. L'esquisse peut être une ébauche de forme ou contour d'une image entière ou une ébauche des couleurs ou textures des régions d'une image. L'utilisateur choisira en fonction de la base d'images utilisée ses besoins et préférences. cette technique présente l'inconvénient majeur qu'il est parfois difficile pour l'utilisateur de fournir une esquisse, malgré les outils qui lui sont fournis.

4. L'indexation [VAL 05]

L'indexation a pour but de substituer à une image un représentant (ou descripteur) moins encombrant qui la caractérise le mieux possible et de ne travailler que sur ce modèle lors de la recherche. Cela permettra une meilleure organisation des données, de limiter la quantité de données examinées durant une recherche, d'y accéder rapidement et de confiner la recherche au maximum.

4.1 Les phases d'indexation

Système d'indexation comprend généralement deux phases de traitement :

4.1.1 Extraction des caractéristiques visuelles (indexation logique)

Cette phase consiste à extraire de l'image un résumé de son apparence visuel, par des méthodes d'analyse d'images. La méthode employée pour extraire ce résumé porte le nom de **descripteur d'image.** Après l'extraction des caractéristiques visuelles, le contenu visuel de l'image est décrit par un vecteur numérique (ou un ensemble de vecteurs), que l'on appelle généralement signature de l'image. La mise en place d'un descripteur consiste également à associer aux signatures une mesure de similarité. Au moment de la recherche, deux images seront jugées similaires si leurs signatures sont similaires au sens de cette mesure de similarité. Le descripteur n'est pas universel pour une image donnée, l'information extraite

devant être pertinente pour l'usage visé. Il doit être Fidèle au contenu syntaxique et sémantique de l'image tout en étant compact.

4.1.2 Structuration de l'espace des signatures (indexation physique)

Indexer l'espace des signatures consiste à organiser cet espace selon une structure d'index (généralement multidimensionnel) permettant d'accélérer la recherche. Si l'ensemble de l'espace et de la structure ne peuvent pas être chargés en mémoire principale, l'optimisation porte sur la réduction du nombre d'accès au disque. Les approches les plus récentes s'attachent également à réduire les couts en temps CPU correspondant généralement au calcul de la mesure de similarité. La recherche via un index doit naturellement être plus rapide qu'un parcours exhaustif de l'espace. Pour être efficace, la structure développée doit prendre en compte les propriétés de l'espace des signatures.

Des nombreuses techniques basées sur des arbres existe dans la littérature comme (B-Tree, R-Tree, KD-Tree, SS-Tree, QuadTree,...) ont été proposées. Pour qu'un système de recherche d'images soit performant, il faut que l'indexation logique soit pertinente et que l'indexation physique permette un accès rapide aux documents recherchés.

Le rapprochement de deux communautés de recherche peut poser quelques problèmes de vocabulaire. Notamment, il est important de souligner que le terme "indexation" est employé dans la communauté de l'analyse d'images comme dans celle des bases de données, mais n'y revêt pas la même signification ! En analyse d'images, indexer une image correspond à l'étape d'extraction des caractéristiques visuelles. Ici, l'index est la signature de l'image. La communauté des bases de données parle quant à elle d'indexation de la base d'images, ce qui correspond à l'étape de structuration de l'espace des signatures. Dans ce cas, l'index est la structure issue de cette organisation.

4.2 Méthodes d'arbre en Indexation physique (multidimensionnelle)

Les techniques principales de la méthode d'arbre en indexations multidimensionnelles visent à regrouper les descripteurs de base et à les englober dans des cellules faciles à manipuler (hiérarchie) [OGI 01].

Cela nous permet d'éviter de considérer tous les descripteurs dans la base lors d'une recherche en considérant seulement les groupes ou les paquets les plus pertinents et enfin, on travaille seulement avec les descripteurs dans les paquets sélectionnés.

Il y a deux grandes catégories de techniques de création de cellules : partitionnement des données et partitionnement de l'espace.

4.2.1 Partitionnement des données

Le partitionnement de données (*Data Clustering* en anglais) est l'une des méthodes statistiques d'analyse des données. Elle vise à diviser un ensemble de données en différents« *paquets* » homogènes, en ce sens que les données de chaque sous-ensemble partagent des caractéristiques communes (dans notre cas sont des images), qui correspondent le plus souvent à des critères de proximité (similarité informatique) que l'on définit en introduisant des mesures et classes de distance entre objets[WIKIPEDIA].

Les techniques de partitionnement de données créent des cellules en se basant sur la distribution des descripteurs et leur proximité relative dans l'espace.

Dans cette catégorie, on trouve des techniques comme B-Tree, la famille R-Tree, SS-Tree….. Etc.

a. B-Tree : Balanced-tree (Bayer et McCreight, 1972)

C'est un arbre balancé qui nous permet de structurer des données selon l'une des dimensions. Un nœud qui n'est pas la racine d'un arbre B-Tree d'ordre m contient entre m/2 et m nœuds fils. Toutes les feuilles sont au même niveau et contiennent l'information. Un nœud quelconque qui a k nœuds fils a k-1 éléments en ordre croissant qui sont les valeurs de séparation divisant les valeurs de l'axe choisi des nœuds fils.

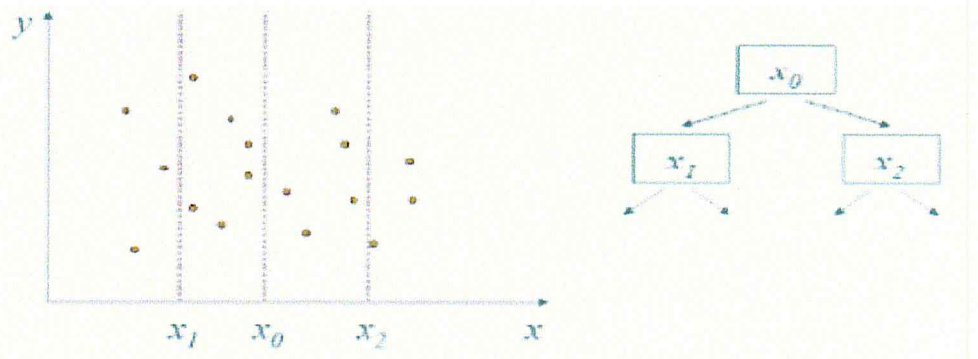


Fig.1.4. Structure de l'arbre B-Tree.

Pour la recherche, on part de la racine. Cette structure nous permet de vérifier s'il existe ou non un élément dans la base, mais elle ne convient pas pour trouver les éléments qui sont proches d'une entrée quelconque parce que chaque nœud interne contient seulement les valeurs de séparation selon un seul axe.

b. La famille R-Tree : Rectangle Tree

Dans la famille R-Tree, on trouve trois structures R-Tree, R+-Tree, et R*-Tree.

L'idée de base de cette approche est l'indexation des objets spatiaux par des rectangles englobant minimums. Dans le contexte de l'indexation multidimensionnelle, on utilise des rectangles englobants multidimensionnels (hyper-rectangle). Le principe de cette famille est la hiérarchie d'hyper- rectangles englobants et non-disjoints correspondant à la distribution des données par un arbre équilibré, données étant au niveau des feuilles [OGI].

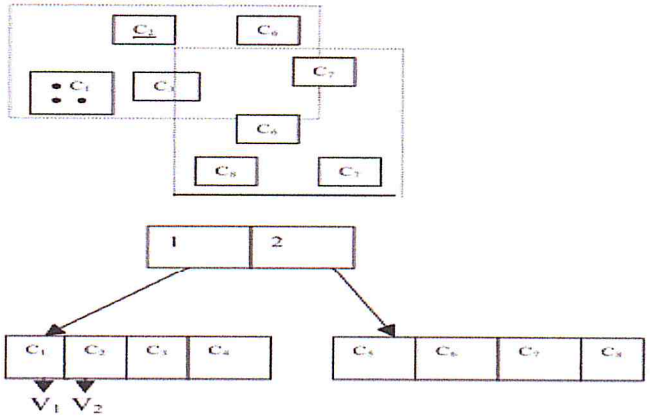


Fig.1.5. Structure de l'arbre R.

Un arbre R-Tree a les propriétés suivantes [GUTT 84] :

- ✓ Au niveau des feuilles, le rectangle englobant est le rectangle minimum recouvrant les vecteurs de données appartenant à ce nœud. Chaque nœud feuille contient au maximum M et au minimum m $\leq$M/2 éléments de données.
- ✓ Tous les nœuds, sauf la racine, qui ne sont pas des feuilles ont entre m et M nœuds fils; le rectangle englobant de ces nœuds est le rectangle minimum qui englobe les rectangles englobants des nœuds fils.
- ✓ Le nœud racine a au moins deux fils sauf quand il est une feuille. Le rectangle englobant du nœud racine recouvre tous les vecteurs de données de la base.
- ✓ Toutes les feuilles sont au même niveau.

Un rectangle englobant est déterminé par deux points $S(s_1, s_2, \ldots, s_n)$ et $T(t_1, t_2, \ldots, t_n)$; pour chaque élément $X(x_1, x_2, \ldots, x_n)$ appartenant à ce rectangle, on a :

$$Si \leq Xi \leq Ti \text{ avec } \forall i \in [1.....n]$$

Pour rechercher toutes les données appartenant à un rectangle Q quelconque, on réalise la recherche à partir du nœud racine, en descendant aux nœuds fils qui ont le rectangle englobant intersectés le rectangle et ainsi de suite jusqu'à ce qu'on rencontre les nœuds feuilles. Au niveau des nœuds feuilles, on retourne toutes les données appartenant au rectangle Q. La création d'un arbre R-Tree est réalisée en ajoutant au fur et à mesure des vecteurs dans l'arbre. L'algorithme exact pour l'insertion d'un nouvel élément (vecteur) dans l'arbre est dans [OGI] ; l'idée principale est de partir du nœud racine, puis de descendre pour trouver la feuille où ajouter le nouvel élément. A chaque nœud, on choisit le nœud fils avec le rectangle englobant le plus petit. Lorsqu'on trouve une feuille pour ajouter l'élément, s'il est plein, on doit le diviser en deux feuilles différentes en minimisant la surface totale des deux nouveaux rectangles englobants (division exhaustive, coût quadratique ou linéaire [OGI]). Le rectangle englobant de chaque nœud est créé et mis à jour au cours de l'insertion des éléments dans l'arbre pour qu'il soit le rectangle le plus petit possible qui recouvre tous les éléments dans le sous arbre de ce nœud.

La structure R-Tree est dédiée à la recherche par intervalles, elle convient pour l'indexation des données spatiales. Lors de la recherche, on peut gagner du temps car on ne doit pas considérer tous les éléments dans la base, on considère seulement les nœuds fils ayant le rectangle englobant qui intersecté l'intervalle entré. Mais la possibilité d'intersection entre les rectangles augmente quand le nombre de dimensions est grand.

Pour les structures R+-Tree[TIM 87], R*-Tree [BECK 90] ont comme but d'optimiser la recherche en minimisant le chevauchement des rectangles englobants. La structure R+-Tree évite le recouvrement entres les rectangles en divisant chaque rectangle qui recouvre un autre rectangle en plus petits rectangles jusqu'à ce qu'il n'y a plus de recouvrement. Cela peut faire augmenter la hauteur de l'arbre mais on peut gagner du temps en réduisant le nombre de sous arbres à visiter. La structure R*-Tree minimise le recouvrement entre les rectangles et minimise aussi le volume des rectangles en appliquant, quand on veut ajouter un nouveau fils à un nœud plein, le mécanisme de réinsérer quelques nœuds fils du nœud plein avant de le diviser en deux avec l'espoir de trouver des meilleurs positions pour les nœuds à réinsérer. R*-Tree est la structure ayant le plus de succès dans la famille R-Tree . Les expérimentations montrent qu'un SR-Tree peut être utilisé efficacement pour l'organisation des données multidimensionnelles et des données spatiales.

c. SS-Tree Similarity Search Tree (1996)[OGI][WHI 96]

Le SS-Tree est une structure d'indexation de similarité qui regroupe les Vecteurs de caractéristiques suivant la similarité entre eux. La mesure de similarité utilisée ici est la distance euclidienne dans le cas où les poids de toutes les dimensions dans le vecteur de caractéristiques sont pareils. La structure de SS-Tree ressemble à celle du R-Tree mais on remplace dans chaque nœud le rectangle englobant par une sphère englobante représentée par un centre et un rayon. Les données sont toujours au niveau des feuilles. Le centre de la sphère englobante d'un nœud est le centre de gravité de tous les éléments dans le sous-arbre de ce nœud. Au niveau des feuilles, la sphère englobante recouvre les éléments appartenant à cette feuille, le rayon de la sphère englobante est égal à la distance entre le centre et le point le plus loin. Et au niveau d'un nœud interne, la sphère englobante recouvre les sphères englobantes de tous les nœuds fils de ce nœud, le rayon de la sphère est toujours supérieur ou égal à la distance entre le centre et le point le plus loin parmi les points dans le sous arbre de ce nœud.

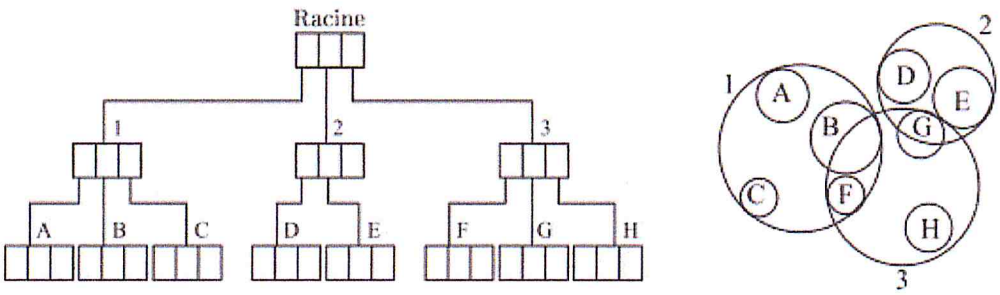


Fig.1.6 Structure de l'arbre SS-Tree.

On utilise aussi le mécanisme de réinsertion d'une partie d'un nœud plein comme dans le cas du R*-Tree afin de minimiser le recouvrement entre les sphères englobantes et le volume des sphères. Les expérimentations montre que le SS-tree est meilleur que le R*-Tree pour les applications de recherche de similarité avec des données de haute dimensionnalité.

Après le SS-tree on trouve SR-Tree qui combine les deux structures R*-Tree et SS-Tree en identifiant la région de chaque nœud par l'intersection du rectangle englobant et de la sphère englobante.

Pour plus de détaille sur le fonctionnement de SR-Tree [OGI] [KATA 97].

4.2.2 Partitionnement de l'espace

A l'égard des techniques de partitionnement de l'espace, on divise directement l'espace multidimensionnel en cellules plus ou moins complexes et régulières.

Il existe plusieurs techniques dans la littérature, mais nous allons présentera quelques techniques comme KD-tree [OGI], KDB-tree [ROB 81], LSD-tree [HEN 89].

a. KD-tree (Bentley, 1979) [OGI]

Le KD-tree (k-dimensional tree) organise les données dans un espace à k dimensions. On structure les données sous la forme d'un arbre binaire. A chaque niveau, on partitionne l'espace en deux sous-espaces successivement selon chaque dimension.

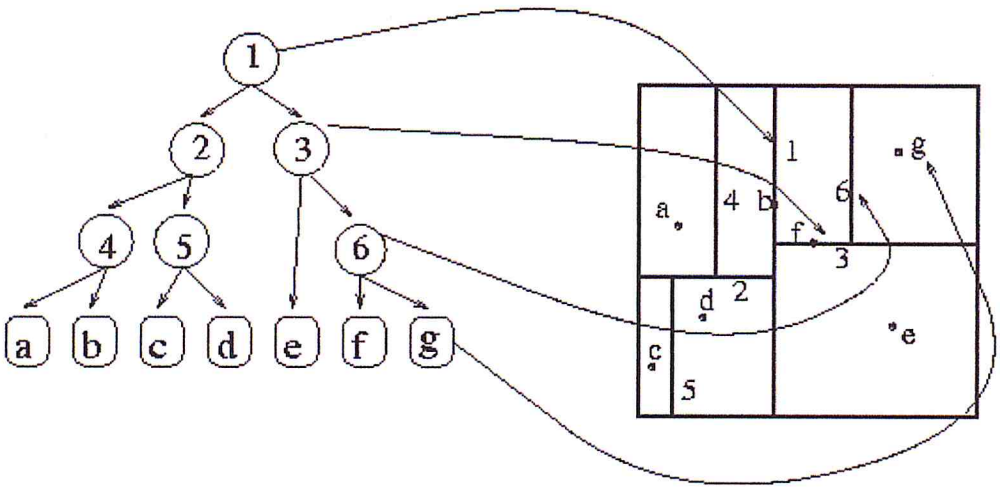


Fig. 1.7. Structure de l'arbre KD-Tree.

Un arbre KD-tree a des propriétés suivantes :

- ✓ La racine est la boîte englobante de tout l'espace.
- ✓ Un nœud correspond à un plan séparateur et deux pointeurs pointant vers deux sous-espaces construits par le plan.
- ✓ Une feuille correspond à la liste des objets de la base appartenant à l'espace de ce nœud.

On peut couper l'espace en deux au milieu par le plan médian, couper à la position de l'objet médian ou au hasard. Cette structure permet la recherche par intervalles ou par plus proches voisins. Elle permet d'accélérer le traitement des données multidimensionnelles. L'avantage de cette structure pour la recherche dépend de la concentration des données dans l'espace.

b. KDB Tree [ROB 81]

Le KDB-tree est une structure d'indexation pour les vecteurs multidimensionnels. Elle combine les propriétés du KD-tree et du B-tree en organisant les données sous forme d'un arbre balancé qui partitionne à chaque niveau l'espace de recherche en deux sous-espaces successivement selon chaque dimension. La disjonction entre les nœuds de même niveau dans l'arbre KDB-tree implique un seul chemin dans la recherche par point (point query). Mais dans le KDB-tree, lorsqu'on divise une région d'un nœud intermédiaire selon un axe, on doit aussi diviser les régions des sous nœuds selon cet axe, cela pouvant créer des nœuds vides ou presque vides. Cela implique une diminution de la performance du KDB-tree dans le cas des requêtes par intervalles ou les requêtes par plus proches voisins.

c. LSD-Tree: Local Split Decision Tree [HEN 89]

Le LSD-tree (Local Split Decision tree) est une structure de données qui supporte efficacement l'accès vers les objets géométriques. Comme d'autres structures, le LSD-tree partitionne l'espace de données récursivement en deux sous-espaces séparés à une position arbitraire. La position pour diviser est choisi pour atteindre l'optimal local, c'est-à-dire le choix de la position pour la division dépend seulement du bloc de l'espace qu'on divise, et non pas des bornes des autres blocs. C'est pour cette raison que l'on appelle cette structure Local Split Decision tree. La structure du LSD-tree ressemble à celle du KD-tree. Chaque nœud de l'arbre enregistre la dimension et la position pour la division.

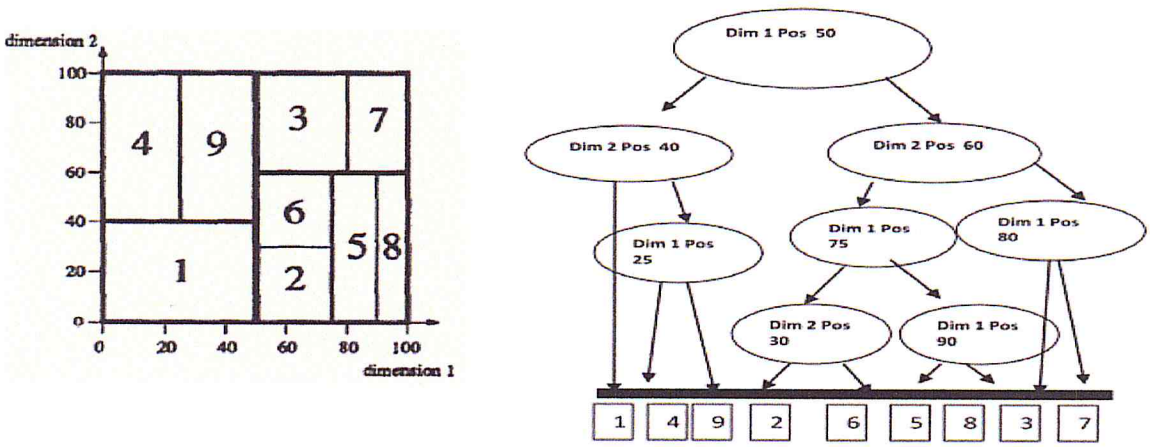


Fig. 1.8. LSD Tree.

4.2.3 Récapitulatif des Méthodes d'indexation [BER 04]

Ci-dessous, le tableau récapitule les avantages et inconvénients des différentes méthodes présentées précédemment.

Méthode	Avantages	Inconvénients
R*-Tree	Formes englobantes permettant d'affiner le filtrage.	Fort taux de chevauchement; éclatement des nœuds problématiques
SS-Tree	Petites tailles d'index	Chevauchement important
SR-Tree	Formes englobantes plus adaptées A de grandes dimensions.	Complexité des formes; recherche coûteuse; taille d'index importants.
LSD-Tree	Codage binaire des régions: Taille de l'arbre réduite et zones vides non gérées.	Organisation mémoire Complexe.

Tableau.1.1. Résumé des avantages et inconvénients des index multidimensionnels

5. La recherche d'image

Une fois la base d'images indexée, la phase de recherche peut revêtir plusieurs formes. La plus ancienne et la plus classique est la recherche par similarité visuelle à partir d'un exemple. Dans sa forme la plus simple, le Système reçoit en entre une image exemple et retourne l'ensemble des images les plus similaires à cette image, au sens de la mesure de similarité associée à la signature. La recherche implique de trouver les plus proches voisins de la signature associée à l'image requête. Ce mode est illustré par l'exemple de la Figure 8 obtenu à partir du moteur IKONA.

Fig.1.9. Recherche globale approximative, à partir de descripteurs de couleur, texture et forme.

6. Quelques systèmes de recherche d'images [BED 10]

Ces dernières années, de nombreux systèmes d'indexation et de recherche d'images par le contenu, ont vu le jour. La plupart de ces systèmes, permettent de naviguer au sein de la base d'images, et/ou d'effectuer des recherches par l'exemple et d'exprimer des requêtes au moyen d'une interface graphique conviviale et adéquate. Voici une liste des systèmes les plus réputés.

✓ QBIC

Le logiciel QBIC (Query By Image Content) d'IBM, est le premier système commercial de recherche d'images par le contenu. Il supporte différents types de requêtes : par l'exemple, par croquis…etc. Les auteurs de ce logiciel fut rassembler un large

panel de descripteurs d'informations contour, couleur et texture. Le système utilise la moyenne pour caractériser la couleur dans les espaces RGB, YIQ, Lab et Munsell. La texture est représentée par une version améliorée des caractéristiques de Tamura.

La forme est assurée par des signatures classiques comme la circularité, la surface, l'excentricité et les moments invariants. La distance Euclidienne est utilisée pour comparer les images et la distance quadratique pour comparer les histogrammes.

✓ Virage

Virage est le moteur de recherche d'images développé par la société Virage Inc. Son objectif est de construire un environnement dédié à la recherche d'images, principalement composé de primitives. Similairement à QBIC, Virage propose des requêtes portant sur la couleur, la localisation des couleurs, la texture et la structure de l'image. L'interface de Virage offre la possibilité d'ajouter et de pondérer les différentes primitives ainsi que d'utiliser le bouclage de pertinence. L'avantage de Virage par rapport à QBIC est qu'il autorise une combinaison entre les différents modes de recherche. L'utilisateur définit le poids qu'il veut attribuer à chaque mode.

✓ MARS

MARS est système inter disciplinaire qui implique plusieurs domaines de recherche: traitement d'image, gestion de base de données et recherche d'information. Pour la caractérisation visuelle, l'image est découpée en blocs de 5*5. Des indices de texture et de couleur sont calculés pour chaque bloc d'image. La couleur est représentée par un histogramme 2D (coordonnées HS), les coefficients d'ondelettes décrivent la texture. La segmentation des images se déroule en deux procédures. La première est l'algorithme des k-moyennes sur l'espace couleur/texture, la deuxième une détection de régions par regroupement selon un modèle d'attraction. Le système est paramétrable : par exemple on peut choisir la palette de couleurs. De plus Mars offre un nombre d'opérateurs logiques pour formuler la requête. La ressemblance entre couleurs est mesurée par l'intersection d'histogrammes. Pour les textures le système applique une distance euclidienne.

7. Domaine d'application

Les applications des systèmes de recherche d'images par le contenu sont variées. Citons les plus importantes : Des applications judiciaires : les services de police possèdent de grandes collections d'indices visuels (visages, empreintes) exploitables par des systèmes de recherche d'images.

Les applications militaires, bien que peu connues du grand public, sont sans doute les plus développées: reconnaissance d'engins ennemis via images radars, systèmes de guidage, identification de cibles via images satellites. Le journalisme et la publicité sont également d'excellentes applications. Les agences de journalisme ou de publicité maintiennent en effet de grosses bases d'images afin d'illustrer leurs articles ou supports publicitaires. Cette communauté rassemble le plus grand nombre d'utilisateurs de recherche par le contenu (davantage pour les vidéos) mais l'aide apportée par ces systèmes n'est absolument pas à la hauteur des espoirs initiaux. D'autres applications incluent : le diagnostic médical, les systèmes d'information géographique, la gestion d'œuvres d'art pour explorer et rechercher des peintures similaires, architecture pour retrouver des bâtiments ou des aménagements intérieurs.

8. Conclusion

Les systèmes de recherche d'images par le contenu permettent aux utilisateurs d'avoir de résultats de recherche d'images très satisfaisantes par rapport les systèmes de recherche d'information par mot clé. Mais le problème majeur lié au développement d'un système d'indexation et de recherche d'images par contenu est le temps de réponse d'une part et d'autre part les résultat de recherche ,pour cela la mise en place des techniques capables de décrire le contenu visuel des images et de prendre compte des besoins des utilisateurs et leurs différents points de vue reste toujours un axe de recherche dans la littérature.

Concernant le chapitre suivant nous focalisons les différents descripteurs visuel existent dans la littérature et les principales méthodes d'analyse de texture, forme, couleur et aussi la norme MPEG 7.

Chapitre 2:
Les descripteurs visuels

1. Introduction

L'objectif de ce chapitre est de dresser un état de l'art sur les descripteurs visuel utilisés dans le système CBIR et aussi les descripteur de la norme MPEG 7. La littérature est très riche mais on va présenter les descripteurs les plus utilisés et les plus importants.

Aujourd'hui avec le développement des systèmes multimédias et le recul de l'écrit, nous utilisons de plus en plus le contenu visuel comme support de communication dans différents domaines. En effet l'image et la vidéo numérique sont partie intégrante de tels systèmes par la densité et la richesse de leur contenu. La même image peut présenter plusieurs significations à différents niveaux : analyse, description, reconnaissance et interprétation.

En vision de la machine, les descripteurs visuels ou descripteurs d'image sont des descriptions des caractéristiques visuelles du contenu des images, des vidéos, des algorithmes ou des applications qui produisent de telles descriptions. Ils décrivent les caractéristiques élémentaires telles que la forme, la couleur, la texture ou la motion, entre autres. [Wikipédia]

2. Descripteur de couleurs

2.1 Les espaces de couleurs

Un modèle de couleur est un modèle mathématique abstrait décrivant la façon dont les couleurs peuvent être représentées comme les tuples de nombres, comme typiquement trois ou quatre valeurs de couleur ou des composants (par exemple RGB et CMYK sont des modèles de couleur). [Wikipédia]

Il existe plusieurs espace de représentation des couleurs, mais nous citions les plus importants et connues.

2.1.1 Espace RGB

Le mode colorimétrique de fondement pour les applications numériques est basé sur le fonctionnement des écrans vidéo (ordinateur, télévision…) qui restituent les teintes par l'addition en lumière (faisceaux d'électrons) de trois couleurs primaires : le rouge (R), le vert (G) et le bleu (B). Avec ce sous-système une teinte Θ est obtenu par une combinaison linaire $\Theta = \alpha*R+\beta*G+\gamma*B$ avec $\alpha, \beta, \gamma \in [0,1]^3$.

Dans le cas d'une image RGB codé sur 8 bits (cas le plus courant), chaque couleur primaire est codé sur un octet et peut donc prendre n'importe quelle valeur dans l'intervalle [0, 255]. Le noir et le blanc correspondent alors respectivement au triplet (0, 0, 0) et (255, 255, 255).

Les trois plans R, G, B sont des plans monochromes (en tons continus) dont les intensités peuvent varier du noir à la saturation (255), figure 2.1 nous montre le cube d'espace RGB.

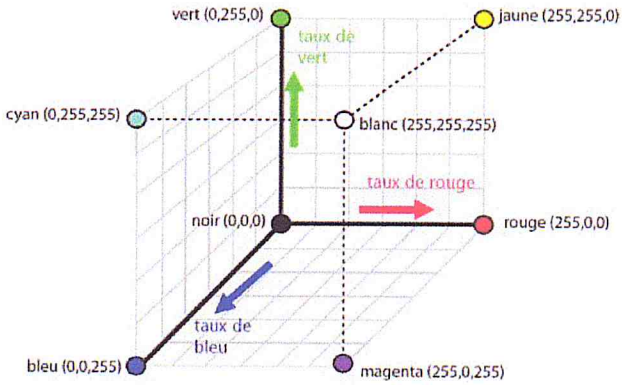


Fig.2.1. Cube d'espace RGB

2.1.2 Espace CIE Lab

CIE Lab (plus précisément L*a*b*) est un modèle de représentation des couleurs développé en 1976 par la Commission internationale de l'éclairage (CIE). Il est une version corrigée du modèle Hunter Lab créé en 1948. Comme tous les systèmes issus du système CIE XYZ, il caractérise une couleur à l'aide d'un paramètre d'intensité correspondant à la luminance et de deux paramètres de chrominance qui décrivent la couleur. Il a été spécialement étudié pour que les distances calculées entre couleurs correspondent aux différences perçues par l'œil humain.

- ✓ Le composant L* est la clarté qui va de 0 (noir) à 100 (blanc).
- ✓ La composante a* représente la gamme de l'axe rouge (valeur positive) → vert (négative) en passant par le gris (0).
- ✓ La composante b* représente la gamme de l'axe jaune (valeur positive) → bleu (négative) en passant par le gris (0).

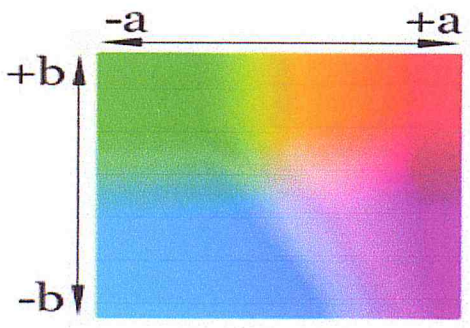


Fig.2.2. Espace de couleur CIE lab.

2.1.3 Espace CMY et CMYK

L'espace CMY est le dual de l'espace RGB. Les teintes extrêmes noir et blanc sont cette fois représentées respectivement par les triplets (1, 1, 1) (0, 0, 0) et le blanc qui correspond à l'absence de couleur (c'est la couleur de fond par défaut, celle du papier sur lequel on imprime). Les trois autres couleurs utilisées dans ce système sont le Cyan C, le Magenta M et le Yellow (jaune) Y. Ces trois couleurs sont les primaires du mode synthèse soustractive utilisé par le système CMY mais aussi pour le système CMYK (k pour le black, noir) utilisé dans les applications d'impression.

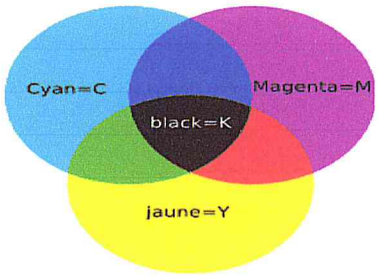


Fig.2.3. Espace CMY (Synthèse soustractive)

2.1.4 Espace HSV

Cet espace représente la couleur selon trois entités: la teinte (H), la saturation (S) et l'intensité ou luminosité (V).La figure 2.4 suivante présente l'espace HSV, ce derniere simule le comportement visuel humain..

Ce Modèle de couleur (teinte, saturation, intensité), découple la composante d'intensité de la couleur de support de l'information (teinte et saturation) dans une image en couleurs. En conséquence, le modèle HSV est un outil idéal pour le développement d'algorithmes de traitement d'images basé sur les descriptions de couleurs qui sont naturelles et intuitives pour les humains .[WOOD 01]

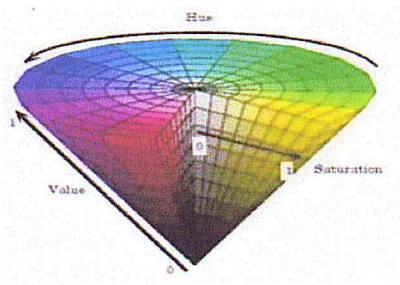


Fig.2.4. Espace de couleur HSV.

2.2 Les Histogrammes

Les histogrammes sont en général des distributions de fréquences, et les histogrammes des images décrire la fréquence des valeurs d'intensité qui se produisent dans une image. Ce concept peut être facilement expliqué par l'examen d'une image en niveaux de gris. Un histogramme H pour une image I avec valeurs d'intensité dans l'intervalle I (u,v) ∈ [0,K-1] contient exactement K entrées ,où, pour un type de 8 bits en niveaux de gris, $K=2^8 = 256$. Chaque individu entrée histogramme est défini comme :

H(i) : le nombre de pixels dans I avec la valeur d'intensité i. [BURG 09]

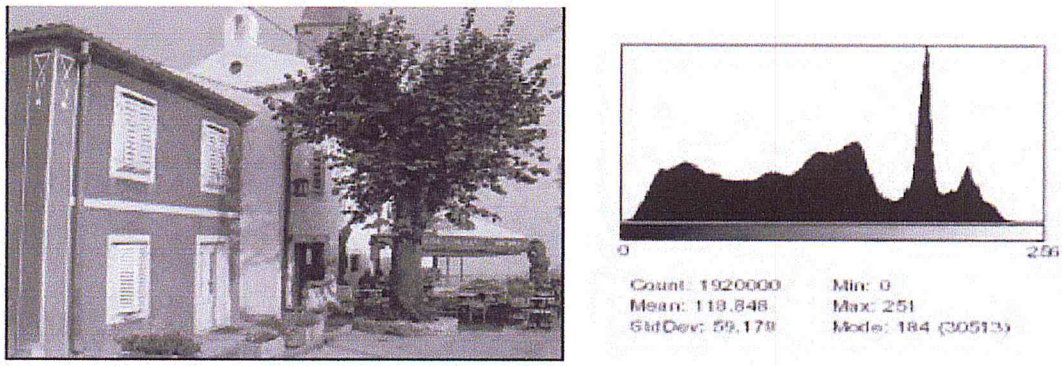


Fig.2.5. un histogramme illustrant la distribution de fréquence 256 de ses valeurs d'intensité.

H [0] est le nombre des pixels avec la valeur (intensité) 0, H [1] est le nombre des pixels avec la valeur 1 et ainsi de suite. Le résultat de calcul d'histogramme est un vecteur h à une dimension de longueur K.

Mais l'histogramme ne contient aucune information sur la disposition spatiale des pixels ou les couleurs dans l'image la figure 2.6 nous montre deux images différentes avec le même histogramme.

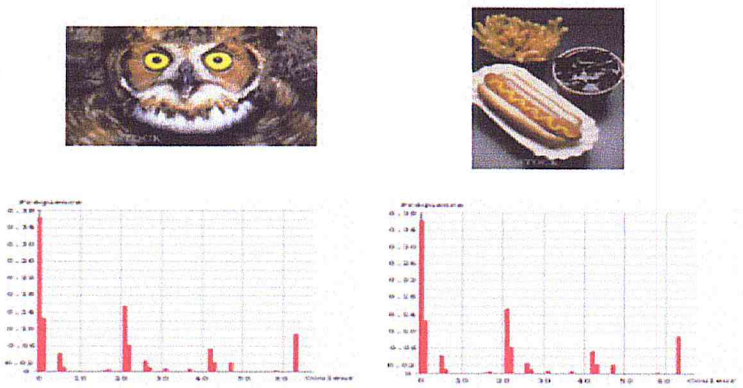


Fig.2.6. images très différentes avec des histogrammes identiques.

Cela est intentionnel puisque la fonction principale d'un histogramme est de fournir des informations statistiques (par exemple, la distribution des valeurs d'intensité) sous une forme compacte. Est-il possible de reconstruire une image en utilisant seulement son histogramme? C'est, peut-être un histogramme en quelque sorte "inversé" Compte tenu de la perte d'information spatiale, sauf dans les cas les plus triviaux, la réponse est non. A titre d'exemple, considérons la grande variété des images que vous pourriez à l'aide du même nombre de pixels d'une valeur spécifique. Ces images semblent différentes, mais ont exactement le même histogramme.

2.3 Les moments statistiques [MEH 95]

La méthode d'histogramme utilise la distribution complète de la couleur. Les données stockées sont beaucoup. Cela nous met beaucoup de temps et aussi de mémoire.

Donc pour résoudre ce problème, au lieu de calculer la distribution complète, dans les systèmes de recherche d'images, on calcule seulement des dominantes caractéristiques de couleur tels que l'espérance, la variance et d'autres moments. Cette approche consiste à calculer une somme pondérée de la moyenne, de la variance et du moment du troisième ordre pour chaque canal de couleur, pour fournir un nombre unique utilisé pour indexer.

Si P_{ij} est la valeur du pixel j pour le canal i (i : RGB), N est le nombre des pixels dans l'image, ces moments sont définis par :

$$\mu i = \frac{1}{N} \sum_{j=1}^{N} Pij$$

$$\sigma i = \sqrt{\frac{1}{N} \sum_{j=1}^{N} (pij - \mu i)^2}$$

$$si = \left(\frac{1}{N} \sum_{j=1}^{N} (pij - \mu i)^3\right)^3$$

Le moment d'ordre μi représente la couleur moyenne de l'image. Le moment d'ordre σi caractérise le contraste d'une image, plus la variance de couleur est grand, plus l'image est contrastée. Le moment d'ordre 3 si caractérise la quantité de lumière dans une image.

Concernant la distance entre l'image requête I et une image H de la base en utilisant les moments statistiques est donnée par la relation suivante :

$$Dist_{mom} = \left(\sum \left|\left((\mu(I) - \mu(H)\right) + \left(\sigma(I) - \sigma(H)\right) + \left(s(I) - s(h)\right)\right|^2\right)^{1/2}$$

2.4 Corrélogramme de couleurs [HUA 97]

Le Corrélogramme de couleurs a été proposé pour qualifier non seulement la distribution de couleurs des pixels, mais aussi la corrélation spatiale entre les pairs de couleurs. Ces informations sont représentées sous la forme d'un histogramme à trois dimensions : les deux premières dimensions représentent les combinaisons possibles de pairs de pixels et la troisième dimension représente leurs distances spatiales. Une autre définition dit que Corrélogramme est un tableau indexé par des pairs de couleurs, où le k^{ieme} entré (i, j) spécifie la probabilité de trouver un pixel de couleur j à une distance k d'un pixel de couleur i dans une image.

Soit I l'ensemble de pixels d'une image et $I_{c(i)}$ l'ensemble de pixels dont la couleur est *C(i)* Alors le Corrélogramme de couleurs est défini par :

$$Yi,j^{(k)} = Pr_{p1 \in c(i), p2 \in I}\left[p2 \in I_{c(i)}, |p2 - p1| = k\right]$$

Où (i, j) $\in \{1,2 \dots\dots\dots\dots\dots, N\}$ et k $\in \{1,2 \dots\dots\dots\dots\dots, d\}$ et $|p2 - p1|$ est la distance entre les pixels p1 et p2. Si on considère toutes les combinaisons de paires de couleurs alors la taille du Corrélogramme de couleurs est considérable.

Les Corrélogramme donnent des meilleurs résultats en termes de pouvoir de discrimination. Cependant, son implémentation est très délicate. Même si l'espace mémoire n'est pas forcément plus élevé que celui utilisé par les histogrammes, leur création est très coûteuse en temps de calcul, ce qui nécessite une implémentation très optimisée pour qu'ils soient utilisables avec un temps de calcul raisonnable.

La figure suivante nous montre un exemple de Corrélogramme sous le programme Rummager

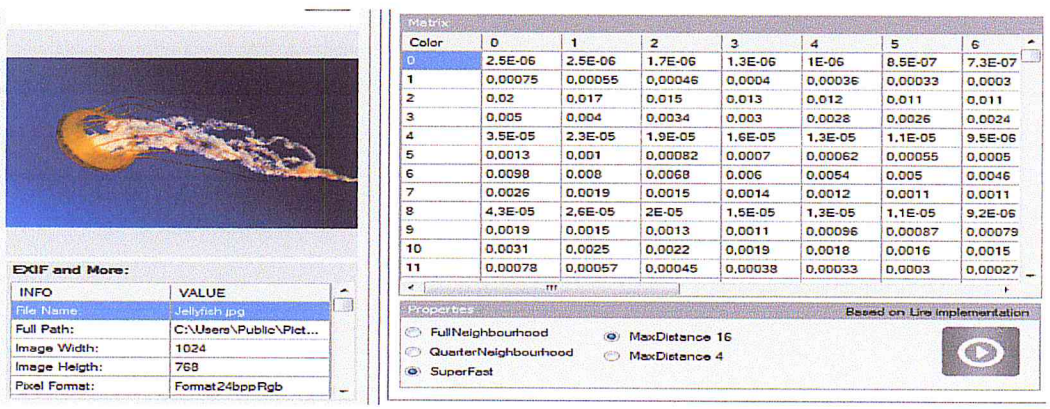


Fig.2.7. Exemple de Corrélogramme

2.5 Vecteur de cohérence de couleur [PASS 96]

Le vecteur de cohérence de couleur CCV, représente une autre variante plus détailles de l'histogramme de couleur, il a été proposé par [PASS 96], dans cette technique chaque rang de l'histogramme peut être partitionné en deux catégorie :

- ✓ Cohérent s'il appartient à une région de couleur uniforme.
- ✓ Incohérent si non.

On note α_i le nombre des pixels cohérents dans le i^{eme} rang de couleur et β_i le nombre de pixels incohérents. Le CCV d'une image est défini comme suite :
$[(\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (\alpha_n, \beta_n)]$, tel que la somme : $(\alpha_1 + \beta_1, \alpha_2 + \beta_2, \ldots\ldots\ldots, \alpha_n + \beta_n)$ donnera l'histogramme de couleurs de l'image, L'avantage de cette approche réside dans l'ajout de l'information spatiale à l'histogramme et cela à partir de leur raffinement, mais elle présente l'inconvénient d'amplifier la sensibilité aux conditions d'illumination.

2.6 Résume

L'information relative aux couleurs est particulièrement importante dans la caractérisation d'une image. Plusieurs études ont été menées pour trouver un critère de choix des descripteurs de couleurs pour l'indexation des images, mais aucune n'a abouti. Ceci peut s'expliquer par le manque de subjectivité de cette information, les descripteurs couleur ne suffisent pas à indexer efficacement une image, ni à la chercher.

3. Descripteur de texture

La texture est une information de plus en plus utilisée en indexation et la recherche d'images par le contenu, car elle permet de pallier certains problèmes posés par l'indexation par la couleur, notamment lorsque les distributions de couleur sont très proches.

Cependant, il n'existe pas de définition formelle de ce qu'est une texture, et il n'en existe pas non plus de représentation officielle, Même si la norme MPEG-7 propose une représentation pour la texture, il existe aujourd'hui de nombreuses approches différentes pour la décrire.

La majorité des méthodes d'analyse de texture sont basées sur des méthodes statistiques. Le but est de définir un vecteur décrivant au mieux les caractéristiques de la texture. Ces méthodes statistiques la caractérisent en termes de distribution spatiale de l'intensité des pixels. Les approches les plus utilisées sont les matrices de cooccurrences, les filtres de Gabor et les ondelettes.

Au même titre que la couleur, la texture est une caractéristique fondamentale des images car elle concerne un élément important de la vision humaine. De nombreuses recherches ont été menées à la fois dans les domaines de l'analyse et de la synthèse de texture.

L'étude de la texture des objets d'une image peut avoir des objectifs très divers : obtenir des informations sur la nature d'un l'objet, segmenter l'image en régions homogènes, identifier la texture afin de la réduire à un ensemble de paramètres (compression d'images), recherche d'image par contenu, etc.

D'après [DEL 99], une définition formelle de la texture est quasiment impossible. D'une manière générale, la texture se traduit par un arrangement spatial des pixels que l'intensité ou les couleurs seules ne suffisent pas à décrire. Elles peuvent consister en un placement structuré d'éléments mais peuvent aussi n'avoir aucun élément répétitif.

Dans la recherche par le contenu, la texture permet de distinguer des zones de couleurs similaires mais de sémantiques différentes. De nombreuses méthodes sont référencées dans la littérature [TUE 92] pour l'analyse de texture. Parmi les plus connues, on peut citer

- ✓ **Méthodes structurelles :** la texture est définie comme une organisation spatiale de niveaux de gris.
- ✓ **Méthodes fréquentiel :** la texture est définie comme un mélange de signaux de fréquences, d'amplitudes et de directions différentes, on retrouve: les filtres de Gabor et les ondelettes.
- ✓ **Méthodes statistiques :** la texture est considérée comme la réalisation d'un processus stochastique stationnaire.

3.1 Méthodes statistique

Ce sont les méthodes basées sur des évaluations quantitatives de la distribution de niveaux de gris. Elles étudient les relations entre un pixel et ses voisins. Elles sont utilisées pour caractériser des structures fines, sans régularité apparente. Plus l'ordre de la statistique est élevé et plus le nombre de pixels allant de 1 à n mis en jeu est important. Parmi ces méthodes on peut citer la méthode de la dépendance spatiale des niveaux de gris (SGLDM : Spatial Gray Level Dépendance Method) ou matrices de cooccurrences, caractéristiques de Tamura, la matrice de longueur de plages et la méthode de différence de niveau gris (GLDM : Gray Level Différence Method).

3.1.1 Matrice de cooccurrence [HAR 73]

Dans les années 70 Haralick et al ont proposé une des premières méthodes de caractérisation de texture baptisée matrice de cooccurrence. Cette approche consiste à explorer les dépendances spatiales des textures en construisant d'abord une matrice de cooccurrence basée sur l'orientation et la distance entre les pixels de l'image. De chacune de ces matrices Haralick a défini 14 paramètres caractéristiques de texture, comme le contraste, l'entropie ou la différence inverse des moments . La réussite de cette méthode repose sur le bon choix des paramètres qui sont : la taille de la matrice sur laquelle s'effectue la mesure, et la distance d qui sépare les deux pixels du motif.

Cette méthode permet de déterminer la fréquence d'apparition d'un "motif" formé de deux pixels séparés par une certaine distance d dans une direction particulière par rapport à l'horizontale. Afin de limiter le nombre de calculs, on prend généralement comme valeurs 0°, 45°, 90°, 135°, 180° et 1 pour la valeur de d.

L'exemple suivant nous montre comment une matrice de cooccurrence est construit à partir d'une image représenté par quatre niveau de grise, .cet exemple se limite au cas d=1, l'élément (2,3) de la matrice P $[1,0^0]$ est égal à 4 cela signifie qu'il existe 4 configurations dans l'image où un pixel de niveau de gris 2 est séparé horizontalement d'un autre pixel de niveau de gris 3 par distance 1.

On va dénombrer toutes les connexions de type 0-0 que l'on mettra dans la case (0,0) de la matrice P, puis les connexions 0-1 dans la case (0,1), etc.

0	0	1	3
0	1	3	2
0	2	3	2
1	2	3	0

Matrice en niveau de grise

$P\,[1,0^0] =$

2	2	1	1
2	0	2	1
1	2	0	4
1	1	4	0

Matrice de cooccurrence

Cette méthode a récemment été appliquée à la recherche d'images. Mais elle reste même ne donne pas des résultats pertinents en ce qui concerne l'aspect chromatique des images [GUA 12], puisque cette approche consiste à convertir l'image en niveau de grise avant l'extraction des paramètres, suivant à cette transformation l'image perdre leur contenue colorimétrique.

Les paramètres les plus utilisés dans la recherche d'image sont entropie, énergie, inertie, homogénéité. Afin d'estimer la similarité entre les matrices de cooccurrences, Haralick a proposé 14 caractéristiques statistiques extraites à partir de cette matrice. Actuellement, seulement les 4 caractéristiques les plus appropriées sont largement utilisées :

$$Energie = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left(P_{ij}\ (d,\theta)^2\right)$$

$$Inertie = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left((i-j)^2\ P_{ij}\ (d,\theta)\right)$$

$$Entropie = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left(\log(P_{ij}(d,\theta))P_{ij}(d,\theta)\right)$$

$$Homogénéité = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{P_{ij}}{1+|i-j|}$$

La figure 2.8 nous montre un exemple de cette matrice et aussi les quatre paramètres.

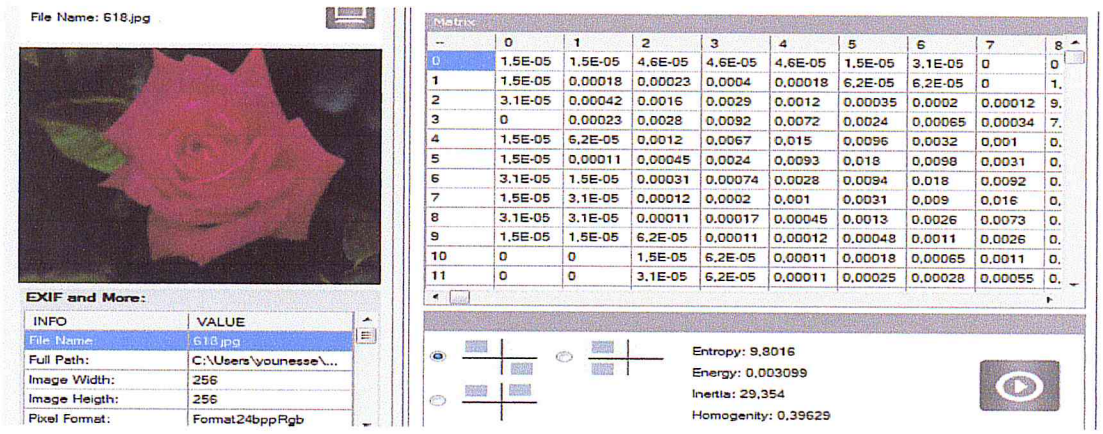


Fig.2.8. Exemple de matrice de cooccurrence.

3.1.2 Caractéristique de Tamura

L'approche Tamura et al. [TAM 78] est intéressante pour la recherche d'images car elle décrit les textures possibles selon des concepts qui correspondent à la perception visuelle humaine. Les auteurs proposent six propriétés visuelles des textures : la grossièreté, le contraste, la direction, présence de lignes (linéarité), régularité et rugosité. Chacun de ces paramètres est mesuré pour établir un vecteur de texture. La forte liaison de ces descripteurs, notamment les trois premiers, avec la perception humaine ; Fait de cette représentation des textures un champ intéressant et exploitable pour la recherche d'images par le contenu. Il semblerait que l'œil humain soit le plus sensible à la

grossièreté de la texture, puis à son contraste et enfin à la direction. Ce type de caractéristiques peut sembler intéressant pour comparer le contenu visuel des images car il correspond directement à la manière dont l'humain les perçoit. Cependant, les méthodes mises en œuvre pour calculer automatiquement ces caractéristiques n'ont donné que des résultats limités pour la recherche d'images dans des données réelles.

La figure 2.9 montre un exemple de ce descripteur.

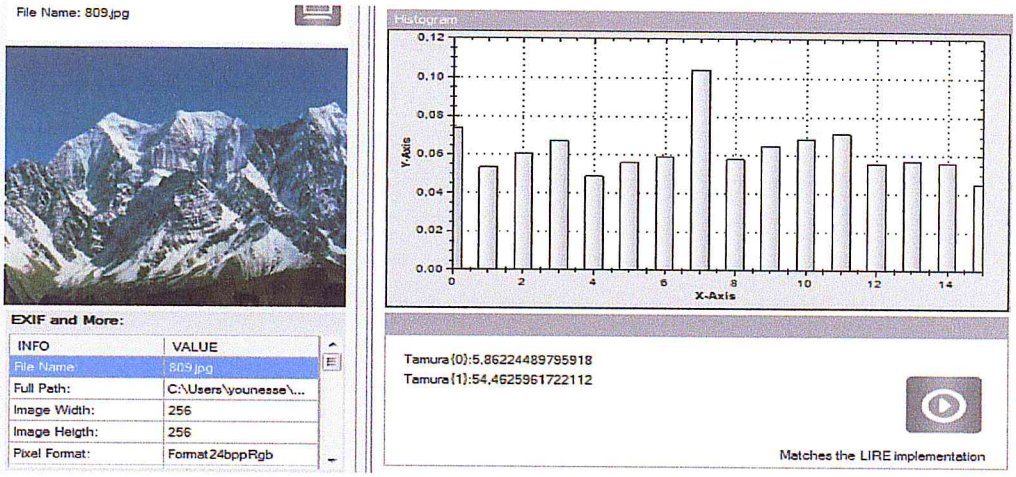


Fig.2.9. Caractéristique de Tamura.

3.2. Les méthodes fréquentielles

L'une des méthodes de description de la texture les plus utilisées concerne les propriétés fréquentielles et s'appuie sur la transformée de Fourier , le filtre de Gabor, les ondelettes , etc. Elle repose sur l'analyse d'une fonction de densité spectrale dans un domaine fréquentiel. La texture est définie comme un mélange de signaux de fréquences, d'amplitudes et de directions différentes. Ces méthodes consistent à extraire l'énergie portée par le signal dans diverses bandes de fréquence[BAJ 73].

3.2.1 Filtre de Gabor

Les filtres de Gabor sont largement utilisés en indexation, pour la description de la texture. Ils permettent une bonne résolution temporelle à haute fréquence et une bonne résolution harmonique sans grande précision temporelle à basse fréquence [MER 01]. Sommairement, les paramètres de texture sont déterminés en calculant la moyenne et l'écart type des niveaux de gris de l'image filtrée par Gabor. En fait, ce n'est pas une seule valeur de moyenne et d'écart type qui sera calculée, mais plutôt un ensemble de valeurs égal au nombre d'échelles multiplié par le nombre d'orientations utilisées.

On aura donc ce qui est parfois appelé la banque de filtre de Gabor. Mathématiquement, toutes les valeurs des moyennes et d'écarts type calculées seront regroupées dans un seul vecteur descripteur.

Un filtre de Gabor 2-D est produit d'une gaussienne elliptique dans toute rotation et un exponentiel complexe représentant une onde plane sinusoïdale [LAB 06]. On rappelle que, dans le domaine spatial, la fonction de Gabor bidimensionnelle est une somme de deux fonctions sinusoïdales, l'une paire et réelle, l'autre impaire et imaginaire, modulée par une enveloppe gaussienne. Le filtrage correspond à une convolution par des filtres de réponse impulsionnelle, pour plus de détails voir [CHE 00] [JAC 95].

Figure 2.10 montres un exemple de vecteur descripteur de Filtre de Gabor.

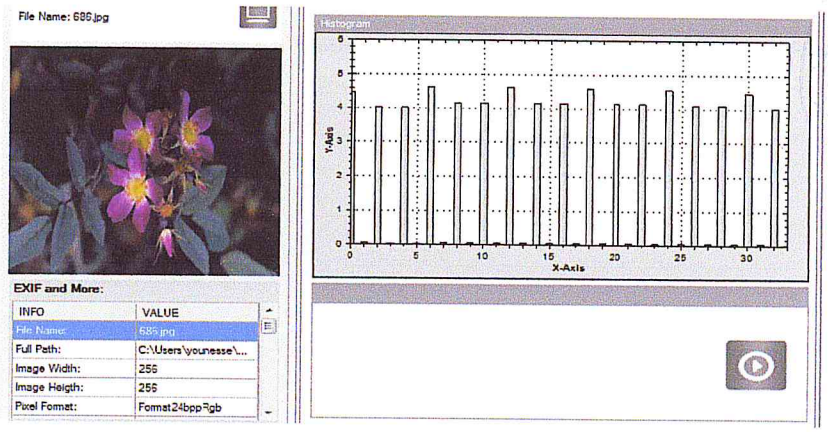


Fig.2.10. vecteur descripteur de Filtre de Gabor

3.2.2 Les ondelettes

La transformée en ondelettes est un outil de mathématique récent qui décompose un signal en fréquences en conservant une localisation spatiale. Le signal de départ est projeté sur un ensemble de fonctions de base qui varient en fréquences et en espace. Ces fonctions de base s'adaptent aux fréquences du signal à analyser. Cette transformation permet d'avoir une localisation en temps et en fréquences du signal analysé.

En outre, il est à la base de nombreuses analyses de texture, telles que les filtres de Haar. La description de texture à base d'ondelettes est utilisée dans la recherche d'images. Pour avoir plus d'information sur les fondements mathématiques de la transformée en ondelettes, le lecteur peut se référer au livre [VETTER 95].

4. Les descripteurs de formes

Au même titre que pour la texture, l'information de forme est complémentaire de celle de la couleur. La forme est généralement une description très riche d'un objet.

L'extraction d'attribut géométrique a été le fer de lance de la recherche d'image par le contenu ces dernières années. De nombreuses solutions ont été proposées pour représenter une forme, nous distinguons deux catégories de descripteurs de formes : les descripteurs basés sur les régions et les descripteurs basés sur les frontières(les contours).

Les premiers font classiquement référence aux moments invariants et sont utilisés pour caractériser l'intégralité de la forme d'une région. Ces attributs sont robustes aux transformations géométriques comme la translation, la rotation et le changement d'échelle.

La seconde approche fait classiquement référence aux descripteurs de Fourier et porte sur une caractérisation des contours de la forme. Nous présentons dans ce qui suit quelques méthodes de description de la forme.

4.1 Les moments géométriques

Les moments géométriques [SHA 78] permettent de décrire une forme à l'aide de propriétés statistiques. Ils représentent les propriétés spatiales de la distribution des pixels dans l'image. Ils sont facilement calculés et implémentés. Par contre, cette approche est très sensible au bruit et aux déformations et le temps de calcul de ces moments est très long.

La formule générale des moments géométriques est donnée par la relation suivante:

$$m_{p,q} = \sum_{p=0}^{m} \sum_{q=0}^{n} x^p y^q \ f(x,y)$$

p + q est l'ordre de moment, le moment d'ordre 0 $m_{0,0}$ représente l'aire de la forme de l'objet.

Les deux moments d'ordre 1 $m_{0,1}$ et $m_{1,0}$ associés au moment d'ordre 0 $m_{0,0}$, permettent de calculer le centre de gravité de l'objet. Les coordonnées de ce centre sont :

$$x_c = \frac{m_{1,0}}{m_{0,0}} \quad \text{et} \quad y_c = \frac{m_{0,1}}{m_{0,0}}$$

Il est possible de calculer à partir de ces moments l'ellipse équivalente à l'objet. Afin de calculer les axes de l'ellipse, il faut ramener les moments d'ordre 2 au centre de gravité :

$$m^g_{2,0} = m_{2,0} - m_{0,0}\, x_c^2$$

$$m^g_{1,1} = m_{1,1} - m_{0,0}\, x_c y_c$$

$$m^g_{0,2} = m_{0,2} - m_{0,0}\, y_c^2$$

Puis on détermine l'angle d'inclinaison de l'ellipse α comme suite :

$$\alpha = \frac{1}{2}\, arctan \frac{2m^g_{1,1}}{m^g_{2,0} - m^g_{0,2}}$$

✓ Les invariants de Hu [HU 62]

à partir des moments géométriques, Hu a proposé un ensemble de sept moments invariants aux translations, rotations et changement d'échelle. Ils sont très utilisés dans la littérature pour la description de formes en vue d'une classification ou d'une indexation, mais sont assez sensibles aux bruits. Par ailleurs cette famille de descripteurs n'est ni orthogonale, ni complète, pour plus de détails un état de l'art est détaillé sur [BED 10].

La Figure 2.11 nous montre ces différents moments.

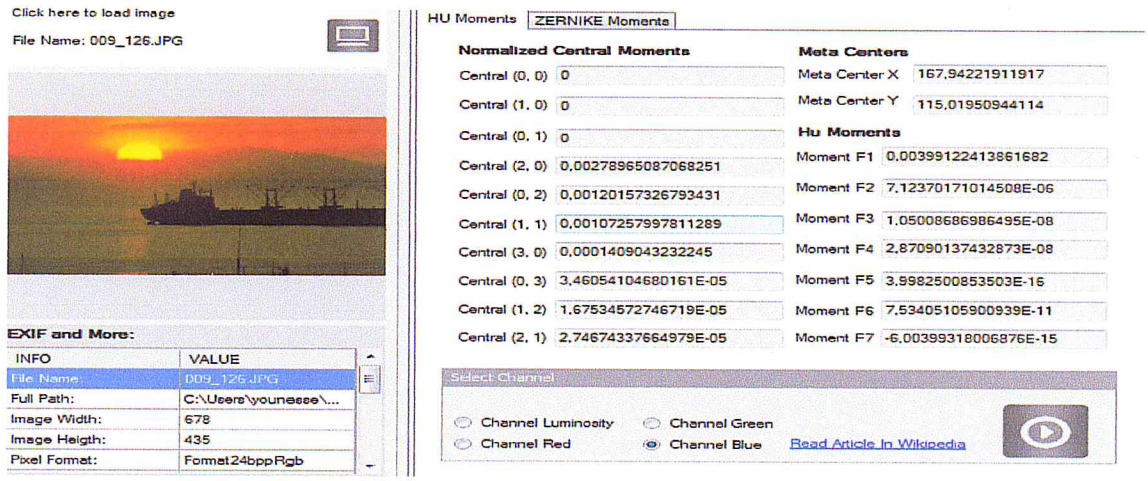


Fig.2.11. les moments de Hu

4.2 Les moments orthogonaux

Par opposition aux moments géométriques qui sont définis par rapport à une base quelconque (x^p, y^q), les moments orthogonaux, comme leur nom l'indique, sont définis dans une base orthogonale, ce qui évite la redondance des informations portées par chacun des moments. Les deux types de moments orthogonaux les plus utilisés sont : les moments de Legendre et les moments de Zernike, dont nous donnons les définitions ci-dessous.

✓ Moments de Legendre

Les moments de Legendre sont définis à partir des polynômes du même nom. Ils sont définis dans le carré unité [-1,1] x [-1,1], ce qui oblige à normaliser l'objet dont on veut calculer ces moments.

✓ Les moments de Zernike

Ce type des moments a été initialement introduit par Teague en 1978 et qui sont construits à partir de polynômes complexes et forment un ensemble orthogonal complet définie sur le disque unité. Ils sont invariants par rotation et changements d'échelles et présentent des propriétés intéressantes en termes de résistance aux bruits, efficacité informative et possibilité de reconstruction.

Pour plus d'information sur le fonctionnement de ces moments le lecteur peut se référer au [BED 10].

4.3 Descripteurs de Fourier

Les Descripteurs de Fourier DFs font partie des descripteurs les plus populaires pour les applications de reconnaissance de formes et de recherche d'images. Ils ont souvent été utilisés par leur simplicité et leurs bonnes performances en terme de reconnaissance [ZHA 05] et facilitent l''étape d'appariement. De plus, ils permettent de décrire la forme de l'objet à différents niveaux de détails. Les descripteurs de Fourier sont calculés à partir du contour des objets. Leur principe est de représenter le contour de l'objet par un signal 1D, puis de le décomposer en séries de Fourier. Les DFs sont généralement connus comme une famille de descripteurs car ils dépendent de la façon dont sont représentés les objets sous forme de signaux.

5. La Norme MPEG 7 [ZAH 01]

Détaillons à présent les différents éléments de MPEG-7 que nous considérons dans ce travail, en adoptant une approche descendante. Commençons donc par énoncer les objectifs et les applications ciblés par MPEG-7.

5.1. Contexte et objectifs de MPEG-7

L'accroissement du volume des données numériques aujourd'hui accessibles sur l'Internet, via des bases de données ou la diffusion par les bouquets numériques, requiert de disposer de modalités d'accès intelligent à ces contenus multimédias qui sont composés d'images fixes, de vidéo, d'audio et de texte. Cette nécessité fait écho à des enjeux socio-économiques importants, s'affirmant dans des contextes d'applications professionnels ou grand public aussi divers que les télécommunications (codage, téléports ,réseaux...), les services en ligne (commerce électronique, informations personnalisées, ...) et la production audiovisuelle (télévision, industrie cinématographique, vidéo, post-production, archivage, accès public aux fonds collectifs...).

Le futur standard MPEG-7 a pour objectif de fournir des descriptions standardisées des contenus multimédias et de supporter un large éventail d'applications potentielles.

Le MPEG-7 est un niveau d'ISO/IEC développé par les Experts de l'Images en Mouvement Groupent.

Le niveau MPEG-7, nommée cérémonieusement « Multimédia Interface de la Description Satisfaire », fournit un ensemble riche d'outils standardisés pour décrire le contenu multimédia afin de rendre rapide et efficace des recherches et des classifications de documents. Il permet de décrire le contenu d'un fichier multimédia avec un certain niveau d'interprétation des informations de ce fichier. Il n'est pas dédié à une architecture, à un média particulier, mais permet de standardiser un nouveau moyen de recherche multimédia et ça pour un grand nombre d'applications.

5.2 Domaines d'application de MPEG-7

MPEG-7 propose de couvrir une gamme d'applications aussi large que possible. Les divers domaines d'application ciblés par MPEG-7 sont :

- ✓ à l'archivage radio,
- ✓ à la TV et le cinéma,
- ✓ aux systèmes d'informations géographiques ou touristiques,
- ✓ au journalisme,
- ✓ à l'architecture,
- ✓ au domaine bio-médical,
- ✓ au commerce électronique.

Enfin, MPEG-7 se propose également d'aborder un certain nombre d'applications ayant un profil hautement spécialisé voire professionnel. Dans ce cadre, nous retrouvons des applications telles que :

- ➢ téléshopping,
- ➢ imagerie satellitaire (Remote Sensing Applications),
- ➢ applications éducatives (Educational applications),
- ➢ télé-surveillance (Surveillance applications).

5.3 Structuration de MPEG-7

Pour optimiser l'efficacité de ses développements, MPEG est structuré en plusieurs groupes qui rassemblent chacun des compétences spécifiques (ex. audio, vidéo) et qui visent à apporter des solutions technologiques aux requêtes et objectifs distincts et identifiés qui leur sont propres.

Ces différents groupes interviennent dans les activités liées aux différents standards MPEG. En ce qui concerne MPEG-7, il bénéficie de l'apport des groupes suivants :

- Requirements,
- Systems,
- Audio,
- Video,
- Reference Software,
- Conformance.

Dans la suite, les descripteurs visuels de bas-niveau qui constituent les éléments porteurs d'information des représentations par le contenu vont être présentés. Nous décrivons les principaux descripteurs basé sur la forme, texture, couleur de MPEG 7.Mais le lecteur peut se trouver plus de d'informations sur l'architecture et la structuration de MPEG 7 à [ZAH 01].

5.4 Descripteurs visuels MPEG-7

Les descripteurs visuels de MPEG-7 se réfèrent aux attributs largement utilisés dans le domaine de l'indexation par le contenu, de couleur, texture, forme et mouvement.

Ils incluent cinq structures de base (Grille spatiale, Repère Spatial 2D, Séries Temporelles, Interpolation Temporelle, Vues Multiples). En ci-dessous le tableau 2.1 résume le différent descripteur, en fonction de leurs types (texture, couleur, forme).

Couleur	Texture	Forme
Espace de couleur	Histogramme des orientations des contours	Localisation spatiale
Quantification de couleur	Texture homogène	Mouvement de la caméra
Histogramme de couleur scalable	Parcours rapide à partir de la texture	Localisation spatio-temporelle
Couleur dominante		Trajectoire
Couleur d'un groupe de trames		Mouvement paramétrique 2D
Couleur structurée		Activité de mouvement
Distribution spatiale de couleur		Reconnaissance de visage

Tableau 2.1 Les descripteurs visuels MPEG-7.

5.4.1 Descripteurs de couleur

La couleur a été également considérée dans le cadre de MPEG-7. Sept descripteurs de couleur sont actuellement adoptés dans le futur standard. Nous allons en donner, les plus importants et leur principales caractéristiques.

a. Histogramme Scalable (Scalable Color Descriptor)

La propriété de scalabilité de l'histogramme de couleur dans l'espace HSV est exprimée à partir de la transformée de Haar qui permet de déduire des variantes de l'histogramme initial à différentes résolution.

Le caractère hiérarchique de la transformée de Haar permet donc de comparer des histogrammes à différentes résolutions.

En outre, les coefficients sont quantifiés de manière non-linéaire en un nombre de bitplanes, et de manière optimisée pour l'espace HSV, ce qui conduit à un deuxième type de scalabilité, allant des représentations binaires des coefficients jusqu'aux représentations en pleine résolution (ce descripteur sera détaillé dans le chapitre 4).

La figure 2.12 nous montre un exemple de ce descripteur.

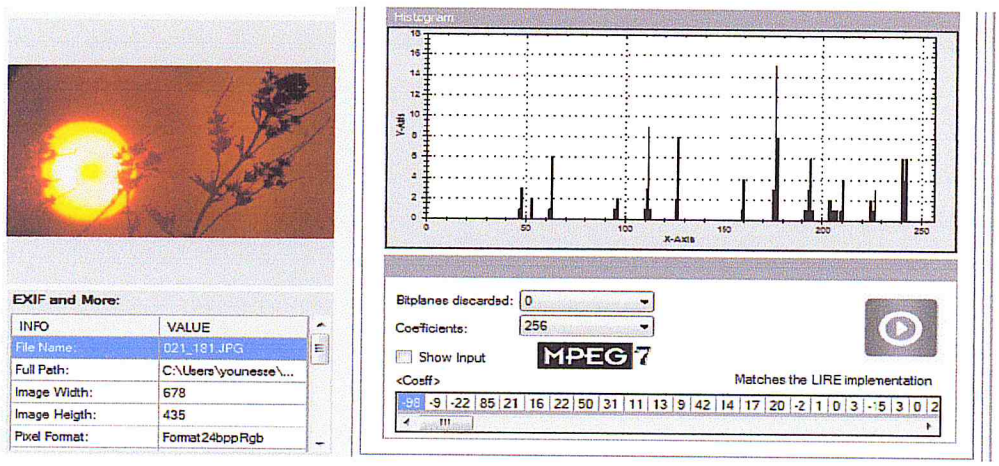


Fig.2.12. Histogramme Scalable (Scalable Color Descriptor).

Les descripteurs à base d'histogrammes de couleur offrent des représentations caractérisant globalement la couleur d'une image, et présentent un comportement d'invariance satisfaisant par rapport aux transformations de similarité. Toutefois, leur inconvénient majeur est lié à la perte de toute information de localisation spatiale. Pour y remédier, un nouveau descripteur prenant en compte un "minimum" d'information spatiale a été proposé et adopté par MPEG-7. Il s'agit de l'histogramme couleur-structure que nous présentons ci-dessous.

b. Histogramme Couleur-Structure

Ce descripteur étend et enrichit la notion d'histogramme, en introduisant un peu de cohérence spatiale locale. Pour ce faire, ce ne sont plus les pixels qui sont comptés et enregistrés dans les intervalles de l'histogramme, mais des éléments structurants (en utilisant la terminologie de la morphologie mathématique). Le principe de construction l'histogramme CS, illustrée Figure 2.13, est le suivant. L'élément structurant, représenté par un masque binaire, est translaté en chaque pixel de l'image. Tous les intervalles de l'histogramme correspondant aux couleurs présentes à l'intérieur du masque sont incrémentés. Ainsi, si l'on note H_{CS} l'histogramme CS, $H_{CS\,(i)}$ devient la fréquence relative des éléments structurants contenant la couleur d'index i.

Remarquons que l'histogramme classique est un histogramme CS, dans lequel l'élément structurant est réduit à un pixel.

Précisons qu'au niveau de MPEG-7, l'élément structurant est toujours un carré de (8 × 8) pixels. Afin de pouvoir gérer des images de taille variable et de garantir une certaine invariance par rapport aux homothéties, les images sont initialement normalisées à une taille fixe approchant au mieux une image de référence de taille (320x240 pixels).

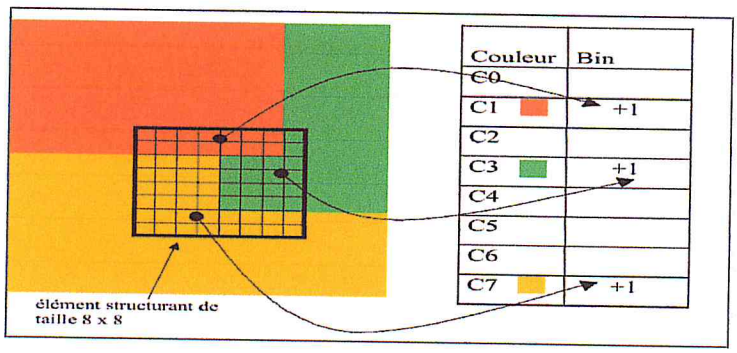


Fig.2.13 Principe de calcul de l'histogramme couleur-structure.

Ce nouveau descripteur parvient à discriminer des informations de couleur là où les histogrammes de couleur classiques échouent, comme lorsque les couleurs sont globalement représentées de manière équivalente, mais réparties différemment dans l'image (Figure 2.14). Plus précisément, on définit la cohérence d'une couleur dominante comme le rapport entre le nombre de pixels cohérents et le nombre total de pixels dans la couleur considérée. Un pixel est considéré comme cohérent si, dans un masque rectangulaire de taille prédéfinie centré en ce pixel, il existe au moins un certain pourcentage de pixels ayant la même couleur que le pixel central.

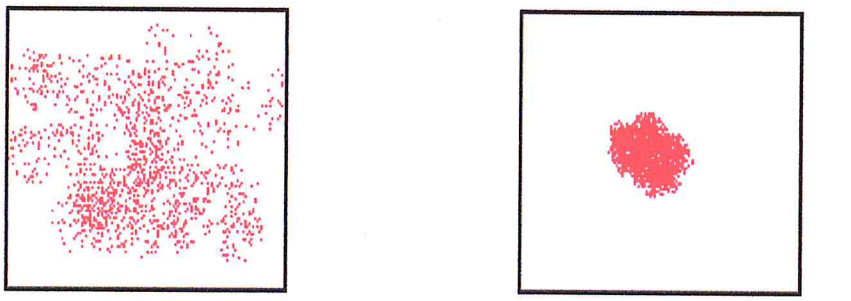


Image avec cohérence de couleur réduite. Image avec cohérence de couleur élevée.

Fig. 2.14 L'histogramme couleur-structure discrimine les deux images alors que les histogrammes classiques les confondent.

Tel que défini, l'histogramme CS est une variante simplifiée des méthodes à base de vecteurs de cohérence de couleur, proposées dans [PASS 96]. Toutefois, les descripteurs de couleur de MPEG 7 ont montré nettement la supériorité de cette technique par rapport aux autres représentations de couleur.

Tous les descripteurs de couleur mentionnés jusqu'à présent sont fondés sur des histogrammes. Ces représentations portent en elles deux limitations principales :

- ✓ La quantification de l'espace de couleur considéré est totalement arbitraire et sans aucun rapport avec le contenu spécifique de chaque image analysée ,L'histogramme contiendra par conséquent de nombreux éléments nuls

- ✓ Le Matching ou bien la mesure de similarité en utilisant la distance quadratique [HAFNER 95] a une complexité haute avec le nombre d'intervalles des histogrammes.

Pour ces raisons, un descripteur plus adapté au contenu spécifique de chaque image a été adopté dans MPEG-7. Il s'agit du descripteur par couleurs dominantes.

c. Descripteur par Couleurs Dominantes (Dominant Color Descriptor)

Son principe consiste à déterminer un nombre réduit (au maximum 8 dans MPEG-7) de couleurs principalement représentées dans l'image, appelées couleurs dominantes. Le descripteur spécifie ensuite le nombre, les valeurs et les fréquences relatives des couleurs dominantes trouvées. Ne nécessitant pas de quantification, ce descripteur peut être utilisé dans tous les espaces de couleur MPEG-7, plus de description dans [ZAH 01].

La figure 2.15 montre ce descripteur.

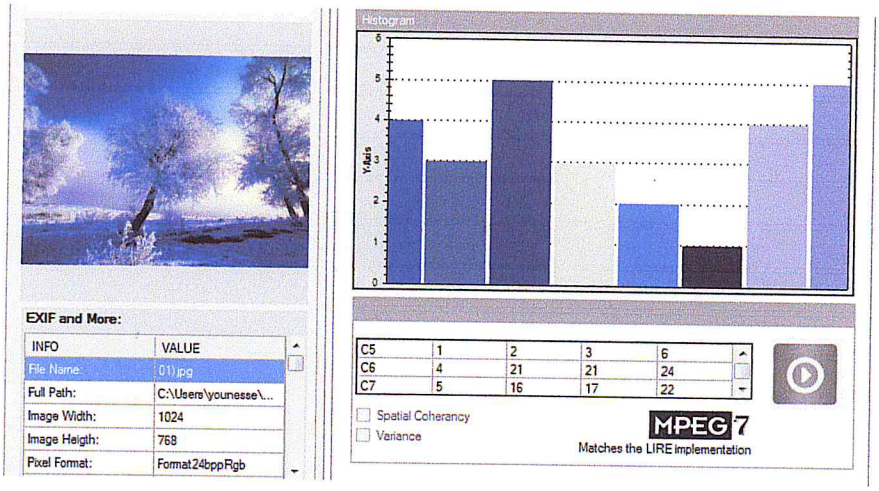


Fig. 2.15. Descripteur par Couleurs Dominantes avec les 8 couleurs dominant de l'image.

d. Distribution Spatiale de Couleur (Color Layout Descriptor)

Contrairement aux descripteurs de couleur mentionnés ci-dessus qui quantifient le taux global de présence de différentes couleurs, le descripteur Color Layout capture la manière dont les couleurs sont réparties spatialement dans l'image. Cela le rend adapté aussi bien à des applications de type requête par l'exemple qu'à des requêtes par l'esquisse (query by sketch), où l'utilisateur, à l'aide d'une interface graphique, crée rapidement une image exprimant grossièrement la répartition spatiale des couleurs souhaitée, qu'il soumet ensuite comme exemple à un système d'indexation.

Le principe de ce descripteur est le suivant : L'image est tout d'abord divisée en 64 (8x8) blocs rectangulaires. Chaque rectangle est représenté par sa couleur dominante, qui est la moyenne (marginale) des couleurs des pixels constituant le rectangle considéré. L'espace de représentation est ici YC_bC_r.

On construit ainsi trois matrices 8 x 8, une par composante de couleur. Les coefficients quantifiés des transformées en cosinus discret 2D (DCT–Discrete Cosine Transform) de chacune de ces matrices, définissent le descripteur Les coefficients sont parcourus en zigzag, à partir des coefficients basse fréquence comme illustré(Figure 2.15)

1	2	6	7	15	16	28	29
3	5	8	14	17	27	30	43
4	9	13	18	26	31	42	44
10	12	19	25	32	41	45	54
11	20	24	33	40	46	53	55
21	23	34	39	47	52	56	61
22	35	38	48	51	57	60	62
36	37	49	50	58	59	63	64

Fig.2.16 Parcours en zigzag des coefficients DCT.

Les propriétés bien connues de décorrélation et de compacité de l'énergie du signal de la DCT permettent de retenir un nombre très réduit de coefficients, aboutissant ainsi à une description très compacte. Quand ce nombre n'est pas spécifié par l'utilisateur, les valeurs par défaut sont de 6 coefficients pour la composante Y et de 3 coefficients pour chacune des composantes de couleur. Ce descripteur sera détailler plus ce descripteur dans le chapitre 4.

La figure 2.17 montre ce descripteur.

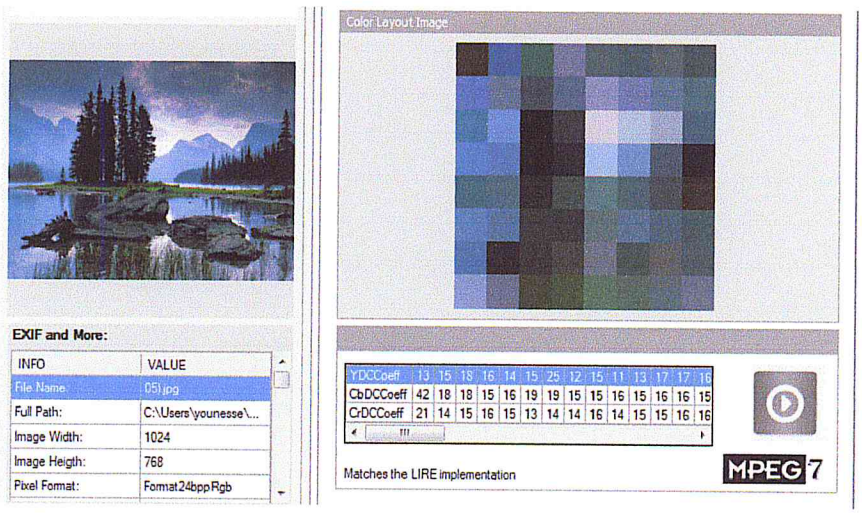


Fig.2.17 Distribution Spatiale de Couleur (Color Layout Descriptor).

5.4.2 Descripteurs de texture

Dans le cadre de MPEG-7, trois descripteurs de texture ont été considérés. Commençons par le plus simple, dédié à la description texturale d'images naturelles génériques, C'est l'Histogramme d'Orientations des Contours (Edge Histogram Descriptor).

a. Histogramme d'Orientations des Contours (EHD)

Ce descripteur représente la distribution des orientations des contours de l'image, grossièrement classifiées en 5 catégories : horizontale, verticale, diagonale à 45°, diagonale à 135° et non-directionnelle.

Pour obtenir une description plus complète, les distributions sont calculées selon trois niveaux de localisation : global, semi-global et local. Dans le premier cas, un seul vecteur de 5 éléments est généré. Il contient les fréquences relatives d'apparition de chaque type d'arête dans toute l'image. Au niveau local, l'image est divisée en 16 (4x4) blocs rectangulaires et le descripteur est calculé pour chaque bloc. Enfin, au niveau semi-global, le descripteur s'exprime sous forme de projections cumulant les distributions locales verticalement et horizontalement (Figure 2.18).

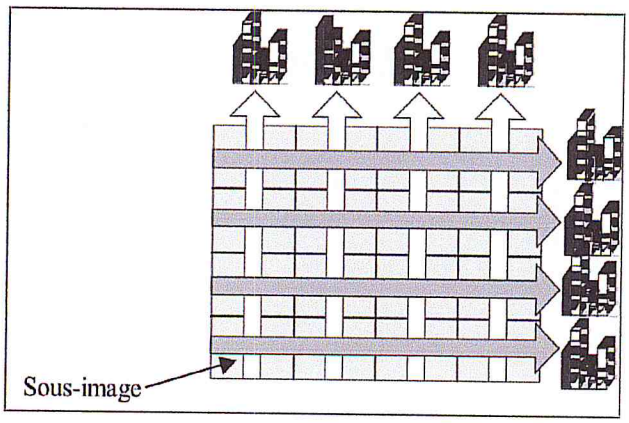


Fig. 2.18. Calcul des distributions semi-globales d'orientations des contours.

Pour déterminer ces différentes distributions d'arêtes, un procédé simple est suggéré. Tout d'abord, on réalise une partition de l'image en blocs carrés. Chaque bloc est divisé en 4 quadrants sur chacun desquels est calculée la luminance moyenne. Ensuite, les opérateurs linéaires directionnels définis par les masques présentés Figure 2.18 sont appliqués à celle-ci.

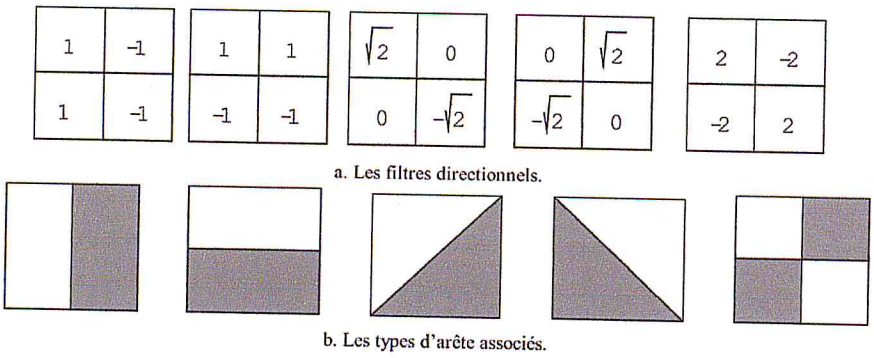


Fig. 2.19. Le principe de détection des différents types d'arête lors de l'extraction du descripteur histogramme d'orientation des contours.

Si la sortie maximale en valeur absolue de ces 5 filtres dépasse un seuil prédéfini, alors on décide que le bloc considéré contient une arrête du type correspondant à ce filtre.

La figure 2.20 montre ce descripteur.

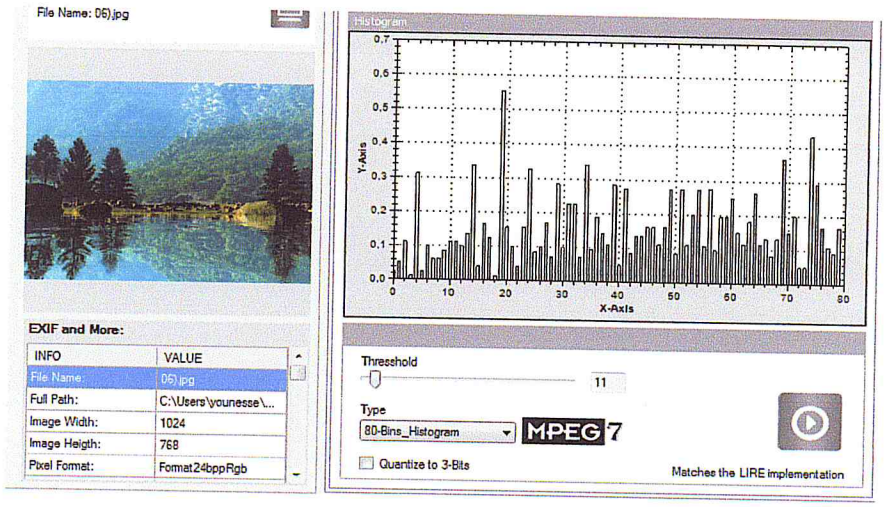


Fig.2.20 Histogramme d'Orientations des Contours.

Les histogrammes d'orientation des contours sont dédiés à des applications génériques de requêtes par similarité d'images naturelles. Toutefois, lorsqu'il s'agit de textures homogènes bien spécifiques, la description fournie se révèle trop élémentaire. MPEG-7 a donc adopté un deuxième descripteur de texture, proposant des représentations plus élaborées, à base d'une analyse spectrale énergétique multi-résolution.

b. Descripteur de Texture Homogène (HTD)

Descripteur de texture homogène (Homogeneous Texture Descriptor) est fondé sur des mesures énergétiques dans des sous-bandes du domaine spectral de Fourier correspondant à un banc de filtres de Gabor.

Les filtres de Gabor, bien connus dans la littérature pour leur propriété d'optimalité de la représentation temps-fréquence d'un signal sont en effet des filtres linéaires passe-bande dont la fonction de transfert est une gaussienne.

Ce descripteur est un vecteur composé de 62 nombre, la première et le deuxième sont le moyen et la déviation de l'image respectivement, et le reste sont l'énergie, et l'énergie de la déviation de filtre de Gabor correspondant au devisions spatial de canal dans le domaine fréquentiel (Figure 2.21), cette figure présente la réponse de cortex de cerveaux qu'il devise le spectre de signal sous des bands dans la fréquence spatial [PEN 01].

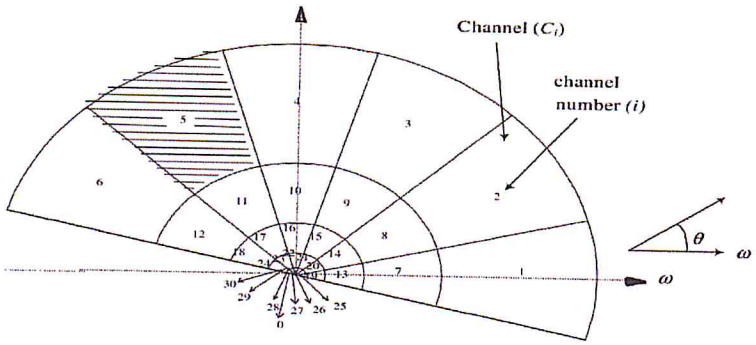


Fig. 2.21 devisons spatial de domaine fréquentiel

Contrairement au descripteur histogramme d'orientation, ce descripteur est particulièrement utile dans le cas d'applications spécifiques de requêtes par similarité dans des bases d'images texturées uniformément.

c. Descripteur de Parcours Rapide de Texture (TBD)

Ce descripteur (Texture Browsing Descriptor) est dédié à une classification très rapide et grossière des textures, visant comme domaine d'application le parcours rapide (browsing) d'une base d'images texturées.

Le principe est ici de représenter une texture par des paramètres correspondant à la perception humaine, comme la direction, la régularité, l'orientation et l'échelle (la granularité). La définition de ces différents paramètres s'appuient sur la même analyse multi résolution par banc de filtres de Gabor que précédemment.

Plus précisément, les éléments retenus sont les suivants :

- ✓ Deux paramètres de direction (θ_1, θ_2) représentant les deux orientations dominantes de la texture. Ils sont issus d'une analyse par banc de filtres de Gabor, qui réalise un filtrage directionnel et Muli échelle des textures et qui permet l'étude des variations des réponses des filtres selon les différentes orientations considérées.
- ✓ Deux mesures de la granularité des textures, mesurant la périodicité des textures selon les deux directions dominantes précédemment déterminées. Cette analyse de périodicité s'appuie sur des mesures de maxima et minima locaux des signaux 1D de projections des images filtrées sur chacune des directions principales.
- ✓ Un paramètre de régularité, noté ρ , représentant le degré de régularité (structuration) d'une texture. Cette mesure de régularité est définie à partir des autres paramètres, en étudiant le caractère persistant des projections précédemment utilisées à travers des échelles et orientations voisines.

Ce descripteur est exclusivement dédié aux applications de type parcours rapide et classification grossière des textures. Les applications associées se déclinent selon les 5 paramètres du descripteur et s'expriment en les termes suivants : "trouver les textures ayant une direction dominante de 30°", ou "trouver toutes les textures fortement régulières".

5.4.3 Descripteurs de Forme

Au niveau du MPEG-7, les aspects de représentation de forme ont été considérés aussi bien dans le cas 2D que 3D. Introduisons tout d'abord les deux descripteurs de forme 2D actuellement retenus dans le standard.

Le premier adopte une vision ensembliste, pour caractériser des formes génériques, non nécessairement représentables par un unique contour.

Parmi les descripteurs de forme de MPEG 7 on trouve en 2D Descripteur ART (Region-Based Shape Descriptor), Descripteur par Espace Echelle de Contour (Contour Based Shape Descriptor), pour les forme en 3D il y a aussi Descripteur par Spectre de Forme 3D (3D Shape Spectrum Descriptor).

En outre, la littérature de MPEG 7 est très riche de descripteur en ce qui concerne les forme et dans le cas de notre travail on s'intéresse pas à ces dernières, le lecteur peut trouver un bon état de l'art à [ZHA 01].

6. Conclusion

Nous avons analysé les principaux descripteurs standardisés de MPEG-7 en ce qui concerne l'aspect de texture, couleur, forme. Ce riche ensemble d'outils de représentation par le contenu offre, d'une manière interopérable, un solide support pour un large d'applications d'indexation et de recherche d'image par le contenue .ce qui nous permet par la suite de concevoir la technique de combinaison des descripteurs visuel.

Dans le chapitre suivant on va présentera un état de l'art sur l'arbre quaternaire et les mesures de similarité entre les images en utilisant ce arbre, et leur application dans les systèmes CBIR.

Chapitre3:
L'arbre quaternaire

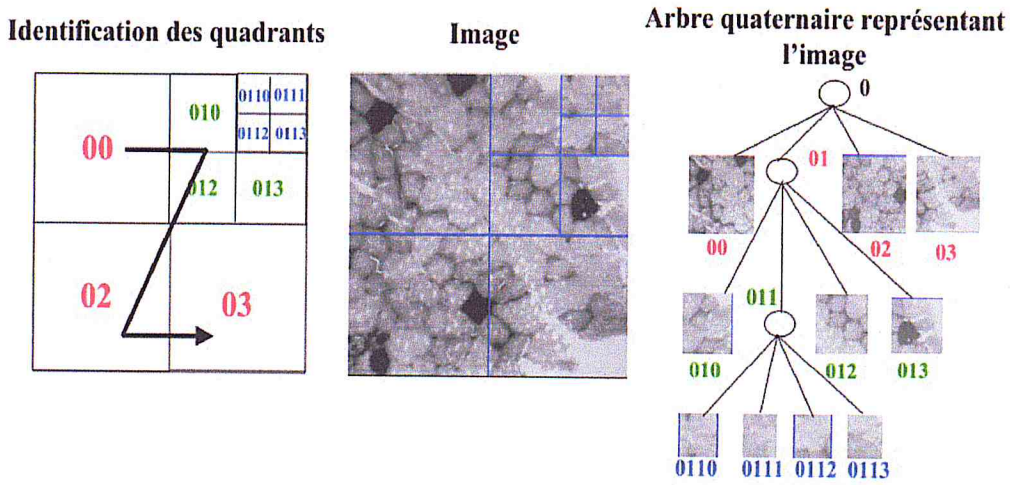


Fig. 3.1 représentation d'une image par l'arbre quaternaire.

Il existe plusieurs fonctions permettant d'associer un identificateur à un nœud d'arbre quaternaire [SAM 84]. Ces fonctions permettent de retrouver facilement, à partir de l'identificateur de l'image et du nœud d'arbre quaternaire, le quadrant associé dans l'image.

Si on suit un ordre en Z, en suivant les directions Nord-Ouest, Nord-Est, Sud-Ouest et Sud-Est et en leur associant respectivement les identificateurs 0,1,2,3 Pour commencer l'entier 0 identifie l'image entière et le nœud racine de l'arbre quaternaire, les entiers 0 à 3 procédé de 0, identifient les quatre premiers quadrants de l'image et les quatre nœuds quaternaire associés. De manière récursive, les nœuds fils d'un nœud identifié par k sont identifiés par kx avec x un entier prenant sa valeur dans [0,3], Deux nœuds de même identificateur dans deux arbres quaternaires différents sont nommés **nœuds homologues** [MAR 02].

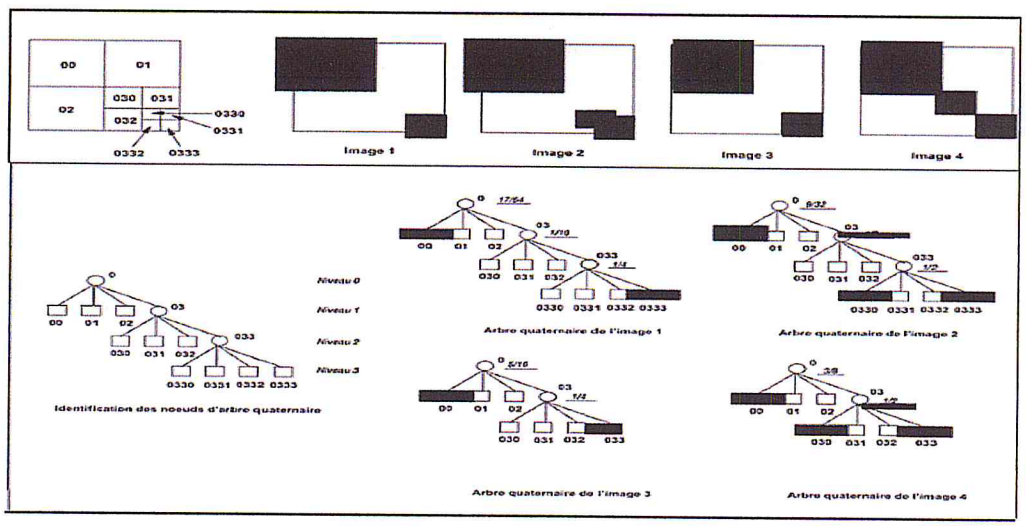


Fig. 3.2. Les arbres quaternaires des quatre premières images

La figure 3.2 présente un exemple de quatre arbres quaternaires représentant les quatre premières images avec l'identification des nœuds. Dans ce cas-là le critère de découpage est l'homogénéité de couleur.

Ce qui concerne les types de nœud de l'arbre quaternaire il y a deux types :

- ✓ **Nœud interne** : S'il est homogène Par-rapport au critère de découpage.
- ✓ **Feuille :** S'il n'est pas homogène.

Les nœuds internes d'un arbre quaternaire peuvent contenir de l'information, comme par exemple l'histogramme de couleurs ou la signature de la région correspondante ou comme dans notre cas le descripteur MPEG7, le nombre de points d'intérêt dans la région associée ou encore une information sur leurs nœuds fils. Dans la figure 3.2, les nœuds internes contiennent la proportion de nœuds noirs dans le sous-arbre (voir les valeurs soulignées dans la figure 3.2)

Dans l'approche de [AHM 97], les images sont caractérisées par des points d'intérêt. Le critère de découpage des images est, dans ce cas, que chaque nœud feuille de l'arbre quaternaire corresponde à un quadrant d'image contenant au plus un point d'intérêt. Les nœuds internes contiennent alors la somme des points d'intérêt situés dans le sous-arbre dont ils sont racines.

Certaines approches [HAD 12] indexent les images par des arbres quaternaires dont le nombre de niveaux est fixe (généralement inférieur à 3) et ça ce qui nous intéresse réellement. Chaque image est, dans ce cas, représentée par un arbre quaternaire complet équilibré. Chaque nœud (interne ou feuille) de l'arbre quaternaire contient de l'information sur la région correspondante dans l'image, comme par exemple ; l'histogramme des caractéristiques visuelles de la région (couleur, texture, forme ou une combinaison de ces caractéristiques). Une telle structure, appelée histogrammes multi-niveaux (en anglais multi-level histograms), permet de filtrer les images au fur et à mesure de la recherche. La figure 3.3 donne un exemple d'histogrammes de couleurs multi-niveaux, chaque nœud de l'arbre quaternaire contenant l'histogramme de couleurs de la région correspondante dans l'image.

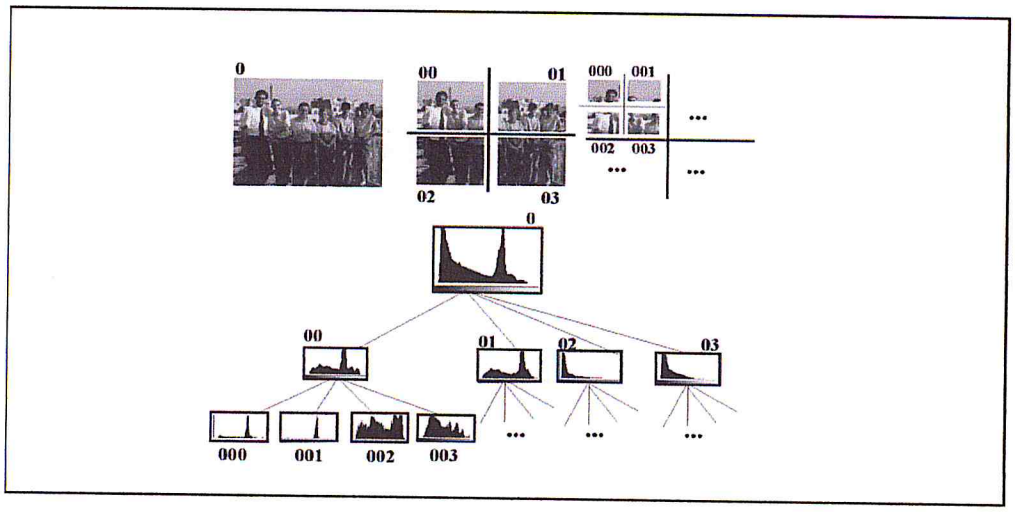


Fig. 3.3. Un exemple d'histogrammes de couleurs multi-niveaux.

3. Similarité d'images représentées par des arbres quaternaires [MAR 02]

3.1. Distances entre nœuds d'arbre quaternaires

A chaque nœud d'arbre quaternaire correspond un quadrant dans l'image associée. Un nœud d'arbre quaternaire peut contenir n'importe quelle information sur la région correspondante : un histogramme de couleurs, un vecteur de caractéristiques des points d'intérêt, la sous-image (compressée ou non) correspondant à la région, ou des descripteurs de Fourier, par exemple. Il est par conséquent possible de calculer des distances entre nœuds d'arbre quaternaires.

On note $\delta_k(i,j)$ est une distance normalisée ($\delta_k(i,j) \in [0,1]$) entre deux nœud homologue k apparaissant dans les arbres quaternaire i, j la distance δ_k peut-être n'importe quelle distance applique aux nœuds d'arbre quaternaire en fonction du contenu des nœuds [CAS 02].

Le choix δ_k dépend du critère de découpage de l'image en arbre quaternaire ainsi que de la valeur donnée aux nœuds feuilles et aux nœuds internes de l'arbre quaternaire. Si les images sont représentées par des histogrammes de couleurs multi-niveaux (voir figure 3.3), alors δ_k peut-être une distance choisie parmi les distances existantes entre les histogrammes de couleurs. Certaines signatures d'images comme les descripteurs de Fourier pour la forme sont des représentations invariantes du point de vue des transformations géométriques (translation, rotation ou homothétie). Si les images sont représentées par des arbres

quaternaires stockant les descripteurs de Fourier associés aux différents quadrants d'images, alors dans ce cas, l'invariance par transformations géométriques s'applique à chaque quadrant associé à un nœud d'arbre quaternaire.

Lorsque deux nœuds homologues sont tous les deux feuilles ou internes, la distance δ_k correspond à une distance entre valeurs de nœuds. Lorsque un nœud k est interne dans un arbre quaternaire, par exemple i mais feuille dans un autre arbre quaternaire j alors plusieurs alternatives apparaissent :

- ✓ $\delta_k\,(i,j)$ =1 parce que les nœuds ne sont pas de même type (interne et feuille).
  Dans ce cas pour tous les fils de k dans l'arbre quaternaire $i, \delta_{kx}\,(i,j) = 1$ parce que les nœuds kx, fils de k existent dans l'arbre quaternaire i, mais n'existent pas dans l'arbre quaternaire j.

- ✓ Le nœud interne k, situe dans l'arbre quaternaire i peut contenir une valeur correspondant au sous arbre dont il est racine. Cette valeur est utilisée pour établir la distance δ_k entre les deux sous arbre de racine k dans les arbres quaternaires i et j (le sous arbre de racine k est une feuille dans l'arbre quaternaire j).

- ✓ Le nœud k, est feuille dans l'arbre quaternaire j peut être transformé en nœud interne le temps du calcul de la distance de similarité, en ayant quatre nœuds fils dont la valeur dépend du critère de découpage. Pour ce faire, on remplace artificiellement le nœud feuille homogène k par un sous-arbre composé de quatre nœuds feuille homogènes selon le critère de découpage. Dans ce cas, la distance δ_k peut-être calculée pour chacun des nœuds fils de k.

Par exemple, dans la figure 3.2 $\delta_{033}(1,3)$ peut être calculée de différentes manières, Parce que le nœud 033 est interne dans l'arbre quaternaire 1 et est feuille dans l'arbre quaternaire 3 alors on peut avoir $\delta_{033}(1,3) = 1$. Dans ce cas, pour tous les nœuds feuilles $033x$, $\delta_{033x}(1,3) = 1$ puisque ces nœuds n'existent pas dans l'arbre quaternaire j.

Si, pendant le calcul de la distance, le nœud 033 de l'arbre quaternaire 3 est découpé en quatre nœuds feuilles fils de couleur noire, alors $\delta_{033}(1,3) = 0$ puisque les deux nœuds homologues sont internes, $\delta_{0330}(1,3) = \delta_{0331}(1,3) = \delta_{0332}(1,3) = \delta_{0333}(1,3) = 0$ puisque seul le nœud 0333 a la même couleur dans les deux arbres quaternaire.

3.3 Cas particuliers de la distance Δ

En fonction des valeurs des coefficients c_k et du choix de la distance δ_k plusieurs familles de distances peuvent être définies .on note T une distance entre les structures de l'arbre quaternaires dont le calcul ne tient pas compte de la valeur des nœuds feuilles, on note Q une distance entre les structures des arbres quaternaire dont le calcul tient compte de la valeur des nœuds feuilles, Une autre distance notée V permet de comparer visuellement les images en utilisant leur arbre quaternaire.

3.3.1 Distance *T*

La comparaison de la structure de deux arbres quaternaires représentant des images, sans tenir compte de la valeur des nœuds feuilles permet de savoir :

- ✓ si le découpage de deux images selon le même critère est identique.
- ✓ si le découpage d'une même image selon deux critères différents est identique.

Ce type de distance, notée T a pour valeur 0, $T(i, j)=0$ lorsque les nœuds internes et les nœuds feuilles sont exactement à la même position dans les arbres quaternaires représentant les images i et j comme le cas de figure 3.4, ou deux image dont les nœuds (internes et feuilles) des arbres quaternaires qui les représentent sont à la même position, on peut dire en quelque sort, deux arbres inversées. Et voici cette figure.

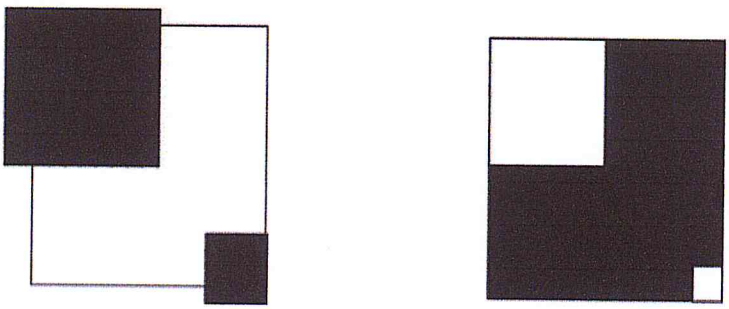


Fig. 3.4. Deux images ont la même représentation d'arbre quaternaire où les nœuds internes et les nœuds feuilles sont exactement à la même position.

Ces deux image ont comme distance T nulle (égale à 0), puisque la deuxième image correspond au complémentaire de la première image. Les arbres quaternaires de ces deux images ont la même structure mais les nœuds feuilles sont inversés.

Cette distance n'a aucune utilité que lorsque les images ne sont pas toutes représentées par des arbres quaternaires équilibrés dont le nombre de niveaux est fixe.

Pour calculer la distance T entre deux images i et j représentées par des arbres quaternaires, la distance $\delta_k(i,j)$ entre nœud d'arbre quaternaire ne prend que deux valeurs :

- $\delta_k(i,j) = 0$ les nœuds homologues k sont tous les deux internes ou tous les deux feuilles dans les arbres quaternaires i et j.

- $\delta_k(i,j) = 1$ lorsque le nœud k est interne dans un arbre quaternaire et feuille dans l'autre, ou lorsque k existe dans un arbre et n'existe pas dans un autre.

Lors du calcul de la distance T les valeurs des nœuds ne sont pas prises en compte, Par conséquent les images peuvent être visuellement très différentes (comme la figure 3.4). La distance Q est une autre distance, issue de la définition de la distance Δ qui tient compte, quant à elle, de la valeur des nœuds.

3.3.2 La distance Q

La distance Q compare deux arbres quaternaires non seulement du point de vue de leur structure, mais également du point de vue des valeurs de leurs nœuds, La distance Q entre deux arbres quaternaires i et j est nulle ($Q(i,j)=0$), lorsque tous les nœuds internes et tous les nœuds feuilles sont à la même position dans les deux arbres et lorsque tous les nœuds ont même valeur dans les deux arbres. Dans ce cas :

- $\delta_k(i,j) = 0$ lorsque les nœuds homologues k sont tous les deux internes ou tous les deux feuilles, avec la même valeur dans les arbres quaternaire i et j.
- $\delta_k(i,j) = 1$ lorsque k est feuille dans un arbre quaternaire et interne dans l'autre (sans valeur dans le nœud interne) ou lorsque k n'existe que dans un seul des arbres quaternaires i et j.
- $\delta_k(i,j) \in]0,1[$ dans les autre cas, c'est-à-dire lorsque les nœuds homologues k sont tous les deux interne ou tous les deux feuilles avec des valeurs différentes $\delta_k(i,j)$ est la distance entre les valeurs stockées dans les nœuds k.

Un exemple de cette distance est celle de la figure 3.4 $Q(i,j)=10/13$ (i et j sont les arbres quaternaire des images de figure 3.4) en effet seul les nœuds interne ont une distance δ_k nulle, la valeur de distance δ_k entre les nœuds feuilles étant 1.

Plusieurs approches, proposent de stocker des images similaires organisées en arbre quaternaire. L'objectif principal de ces approches est d'optimiser l'espace de stockage des images en partageant les parties communes entre leurs arbres quaternaires. Par conséquent, la distance Q peut être utilisée dans ces approches pour organiser les images dans la base sous arborescence d'images dans laquelle une image i est feuille de l'image j dans l'arborescence, si $\forall l$ image de l'arborescence, $Q(i, j) \leq Q(i, j)$. Plus la distance Q est petite entre deux image plus il y a plus des point commun (Quadrants) entre les images et donc, de partager des nœuds possible entre les arbres quaternaires.

Cependant, deux images dont les arbres quaternaires sont très différents $Q(i, j)=1$ peuvent apparaître très similaires visuellement (voir figure 3.5) : par exemple, lorsque le critère de découpage est l'homogénéité de la couleur, une image complètement blanche et une image blanche possédant uniquement un pixel noir l'arbre quaternaire de la première image étant composé d'une seule d'une seule feuille racine blanche. Par conséquente, il est intéressant de définir une distance visuel notée V, telle que si $V(i, j)$ est proche de zéro alors les images (i, j) sont similaires de point de vue de critères de découpage bien que leur arbres quaternaires puissent être très différents.

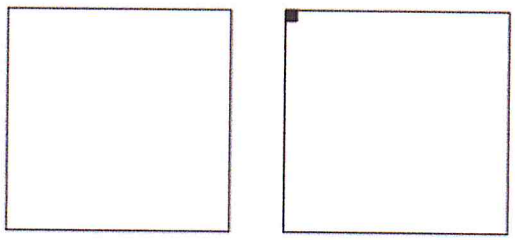


Fig. 3.5 images où ces arbres quaternaires différent mais avec comportement visuel très similaire.

3.3.3 La distance V

On note V une famille de distances visuelles entre images, calculées à l'aide des arbres quaternaires représentant les images. Lors du calcul de la distance $V(i, j)$ les arbres quaternaires des deux images doivent être complétés pour avoir la même structure $T(i, j)=0$ lorsqu'un nœud k est interne dans un arbre quaternaire i et est feuille dans l'arbre quaternaire j le nœud k devient, le temps de calcul de la distance V interne dans l'arbre quaternaire j et possède quatre nœud fils feuilles dont la valeur dépend de critères de découpage. Le calcul de la distance V ne prend en compte que les distances δ_k entre nœuds feuilles ; $\delta_k = 0$ pour tous les nœuds internes, les arbre quaternaire i et j ont la même structure lors de calcule.

Pour comparer la surface des images i et j les coefficients c_k associés aux nœuds feuilles peuvent être proportionnels à la surface des quadrants correspondants dans l'image. Par conséquent, c_k=4 $^{-p}$ si k est situé au niveau p de l'arbre quaternaire (la racine de l'arbre étant au niveau 0) en prenant comme hypothèse que la surface de l'image entière vaut 1. Plus le nœud n'est situé profondément dans l'arbre, plus la valeur de son coefficient, et donc son poids dans le calcul de la distance V est faible.

Pour calculer la distance V entre les images représentées par deux arbres quaternaires (figure 3.6) l'arbre quaternaire de l'image 3 (comme le montre la figure 3.6) doit être complété : le nœud 033 est par conséquent divisé en quatre nœuds fils noir, ainsi. $V(1,3)=$ $(\delta_{0330}(1,3)+\delta_{0331}(1,3)+\delta_{0332}(1,3)+\delta_{0333}(1,3)*\left(\frac{1}{64}\right)=\frac{3}{64}=0.05)$, Si on ne s'intéresse pas aux détails situés en dessous du niveau 2 des arbres quaternaires 1 et 3 on obtient que $V(1,3)$=1/16 car le nœud 033 situe au niveau 2, est interne dans l'arbre quaternaire de l'image 1 et feuille dans l'autre. Et voici la figure.

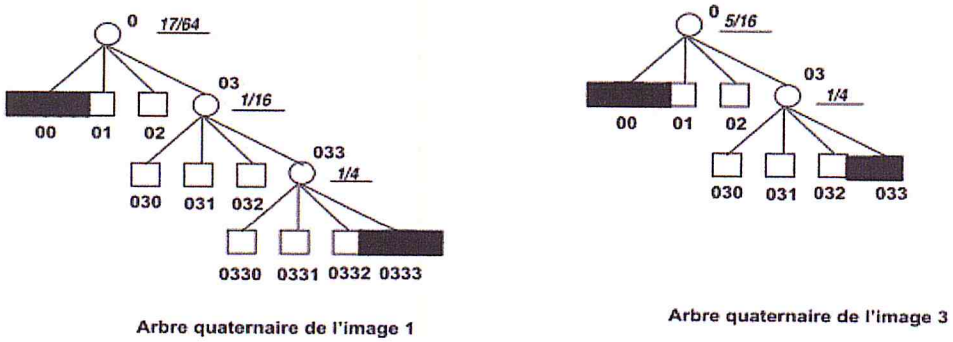


Fig. 3.6 deux arbres quaternaires pour le calcul de distance V

3.4 Perspective

Ces différentes distances que nous avons citées peuvent être utilisées pour la recherche globale d'images par le contenu selon différents critères (comparaison visuelle des images, comparaison de la structure de leur représentation en arbre quaternaire etc.) ou pour la recherche d'images par similarité de régions Dans ce cas, la similarité des images doit être définies en fonction de la distance choisie (c'est-à-dire en fonction des coefficients c_k et aussi de la distance δ_k et d'un seuil donné.

L'arbre quaternaire est utilisé dans plusieurs approches de recherche d'images par le contenu afin de prendre en compte, dans le calcul de distances entre images, la localisation spatiale des caractéristiques (couleur, texture, contour etc.) des images. Les distances, proposées dans certaines approches, apparaissent comme des cas particuliers de la distanceΔ.

Finalement, la distance Δ généralise les distances entre images organisées en arbres quaternaires. La définition de la distance Δ est générale dans le sens où, en fonction des choix des valeurs de coefficients c_k et la distance δ_k, plusieurs distances existantes et de nouvelle distance peuvent être retrouvées et définies en fonction des besoins de l'utilisateur.

En ce qui concerne le cadre de notre travail, on s'intéresse à la distance Δ .Puisque, elle est la plus adéquat dans notre travail où il n'y a pas de contexte spécifique du type des images. De plus l'approche de [HAD 12] a montré une précision spectaculaire avec laquelle il est capable de détecter deux anomalies qui affectent les globules rouges.

4. Indexation des images représentées par des arbres quaternaires[GEN 02]

Dans ce qui suite, on présente une structure d'index pour la recherche d'images par le contenu, l'arbre QUIP (acronyme anglais pour QuadTree Based Index for Image Retrieval and Image Pattern search). En effet, chaque image de la base est représentée par un descripteur dit multi-niveau, qui stocke les descripteurs des quadrants de l'image, obtenus par une décomposition de l'image en arbre quaternaire. L'arbre QUIP permet de regrouper les images en clusters, en fonction de la similarité de leurs quadrants. Cette structure d'index permet non seulement des recherches globales d'images par le contenu, en appliquant un filtrage multi-niveau via l'arbre quaternaire, mais aussi des recherches d'images similaires par région.

4.1 Contexte

Les performances de l'exécution d'une requête de similarité d'images sont fortement dépendantes du nombre d'images à comparer, puisque l'algorithme de base consiste à calculer la distance de l'image requête avec toutes les images de la base. Pour améliorer ces performances, plusieurs approches ont été proposées, telles que les approches de regroupement (ou Clustering) et les approches d'indexation.

Regrouper les images consiste à rassembler, au sein d'un même ensemble, des images ayant des caractéristiques similaires, la similarité étant calculée à l'aide d'une distance entre les descripteurs des images. Parmi les algorithmes de regroupement, la méthode des nuées dynamiques (ou méthode des centres mobiles - en anglais k-means) est la plus utilisée. Il s'agit d'une approche facile à implémenter et qui donne de bons résultats en terme de précision (Precision) et rappel (recall). Cette approche consiste à répartir les données de la base, par exemple des images, en k groupes, répartis autour de k centres de groupes, appelés noyaux ou centroïdes : une image est dans un groupe (ou cluster) *c* si l'image est plus proche en fonction d'une distance choisie, du centroïde du groupe *c* que de n'importe quel autre centroïde des autres groupes.

Il existe également des techniques d'indexation multidimensionnelles pour gérer une base d'images Le principe de ces structures est de découper l'espace et d'indexer les différentes régions de l'espace obtenues, à l'aide d'une structure arborescente, telle que l'arbre R. L'objectif est de diminuer le nombre d'images à comparer et par conséquent le nombre de distances à calculer entre descripteurs, la structure d'index donnant directement accès à un sous-ensemble des images de la base, ce sous-ensemble étant un sur-ensemble de l'ensemble résultat de la requête.

Il est possible de combiner une approche de regroupement avec une structure d'indexation. Par exemple, de partitionner l'ensemble des données à l'aide d'une méthode de regroupement comme k-means, de diminuer les dimensions de l'espace vectoriel, en appliquant une décomposition en valeurs singulières sur chaque groupe ou cluster, puis de construire un index sur l'espace ainsi transformé.

Le QUIP Tree combinant également une structure d'index, l'arbre quaternaire, et une méthode de regroupement (par ex. k-means), pour la recherche globale et partielle par le contenu d'images organisées en descripteurs multi-niveaux. Plusieurs approches représentent les images par des arbres quaternaires pour faire de la recherche d'images par le contenu. Parmi ces approches, certaines utilisent une structure d'index des arbres quaternaires représentant les images, dans le but de diminuer le nombre d'images, et donc d'arbres quaternaires, à comparer. Au lieu de faire un filtrage des images uniquement sur la racine des arbres quaternaires, en comparant les images globalement, la structure d'index, l'arbre QUIP regroupant les images en clusters en fonction de la similarité de tous leurs quadrants.

4.2 Structure de l'arbre QUIP

Dans cette approche, les images de la base sont représentées par des descripteurs multi-niveaux organisés en arbre quaternaire. Nous nous limiterons à des descripteurs de trois niveaux. Néanmoins, cette approche peut se généraliser à des images représentées par des vecteurs stockés dans des arbres quaternaires de plus de trois niveaux.

Lorsqu'un descripteur multi-niveau est représenté par un arbre quaternaire complet de trois niveaux, chaque image de la base est représentée par 21 quadrants : le quadrant 0, représentant l'image entière, les quatre premiers quadrants Nord-Ouest, Nord-Est, Sud-Ouest et Sud-Est de l'image (identifiés par 00, 01, 02 et 03) et les 16 quadrants représentant les sous-quadrants des quadrants 0x, $x \in \{0, 1, 2, 3\}$.

Chacun des 21 quadrants des images de la base est représenté par un descripteur dans un espace à plusieurs dimensions (par exemple à trois dimensions dans le cas des vecteurs moyennes de couleurs). Puis, chacun des 21 espaces à plusieurs dimensions est partitionné en utilisant une méthode de regroupement, comme celle des nuées dynamiques par exemple. En utilisant l'approche des k-means, chaque groupe ou cluster est englobé par une hyper-sphère de centre μ et de rayon d, le rayon d définissant la distance entre le centre μ et l'image la plus éloignée du groupe.

Pour simplifier les explications et les illustrations ci-après, en prenant l'exemple d'un espace de caractéristiques à trois dimensions, chaque quadrant d'image étant représenté par son vecteur moyenne des couleurs dans l'espace RGB. La partie droite de la figure 3.7 représente des descripteurs dans un espace à trois dimensions partitionné en quatre groupes, chaque groupe étant représenté par une sphère englobante. Les descripteurs des 21 quadrants d'une image sont stockés dans les nœuds de l'arbre quaternaire représentant le descripteur multi-niveau de l'image. Pour comparer les descripteurs stockés dans les descripteurs multi-niveaux, nous utilisons les distances Δ, cette approche est indépendante du choix des descripteurs de quadrants d'images et des distances utilisées entre ces descripteurs. La seule contrainte est que la distance au niveau p entre deux descripteurs multi-niveaux soit inférieure à leur distance au niveau $p + 1$.

Pour indexer ces 21 espaces multidimensionnels. L'arbre QUIP suit la décomposition des images : il s'agit d'un arbre quaternaire complet à 3 niveaux, lorsque les descripteurs multi-niveaux ont trois niveaux. Chaque nœud n de l'arbre QUIP contient des lignes une ligne l d'un nœud n de l'arbre QUIP associe un ensemble I_1 d'identificateurs d'image à un groupe de quadrants d'image. Un groupe ou un cluster est représenté par son centre et rayon étant calculé à partir de la distance δ_k entre le descripteur du centre du groupe et celui du quadrant d'image n le plus éloigné dans le cluster, la figure 3.7 nous montre cette structure.

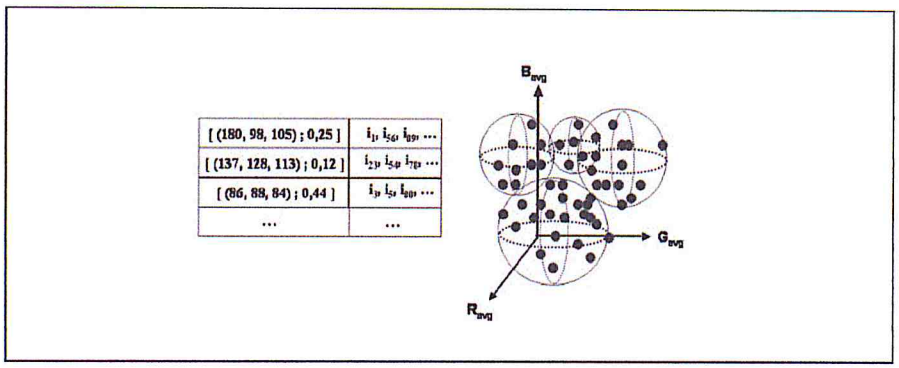


Fig. 3.7 Un exemple de nœud de l'arbre QUIP.

La partie gauche de la figure 3.7 représente un nœud n d'arbre QUIP, les quadrants n des images de la base étant représentés par des vecteurs moyens des couleurs regroupés en quatre clusters. La première ligne du nœud indique que le cluster, dont le centre a pour coordonnées (180, 98, 105) et pour rayon 0.25, regroupe les images d'identificateurs I_1, I_{56}, I_{89}, etc. Les quadrants homologues n de ces images sont donc ans le même cluster et ont une distance δ_n inférieure à 0,25 avec le centre du cluster de coordonnées (180, 98, 105).

L'arbre QUIP permet trois types de recherche d'images par le contenu : la recherche dite globale, la recherche de sous-images et la recherche partielle ou locale. La recherche globale consiste à retrouver des images similaires, selon la distance Δ, à une image requête en considérant les caractéristiques globales des images (donc tous les quadrants d'images depuis le quadrant 0). La recherche de sous-images consiste à comparer une image requête à tous les quadrants d'image stockés dans la base, quelles que soient la taille de l'image requête et celle des quadrants. La recherche locale ou partielle, quant à elle, consiste à retrouver des images ayant une région similaire, la région étant composée de quadrants d'image.

5. présentation et stockage des images similaires

5.1. Introduction

Les images numérisées sont utilisées dont nombreux application, médical, géographique, et notamment les ateliers de traitement d'images sont amenées à générer ,gérer ,stocker des images similaire ou peur différent, ces images occupe assez d'espace de mémoire .aussi des outils efficace doivent-ils développées pour stocker et gérer ces images, les applications gérant des images similaire doivent permettre en plus le stockage et l'affichage, de retrouver, modifier, comparer ces images, En réponse à ces besoin , l'arbre quaternaire générique offre une optimisation du stockage des images, par partage d'information entre les arbres représentant les images, Ce partage est réalisés de sorte qu'il est possible d'appliquer des opérations spécifiques sur le images.

5.1 L'arbre quaternaire générique [MAU 02]

L'arbre quaternaire générique est une structure de données permettant de stocker les images similaires organisées en arbre quaternaire, la similarité des images étant définie par la distance entres les arbres quaternaires les représentant. Cette structure minimise l'espace de stockage par partage des parties commune entre les images via cette structure un utilisateur peut facilement choisir une ou plusieurs images dans la base de données, un utilisateur peut également modifier une image existante dans la base, insérer ou supprimer des images, extraire des images pour construire des séquences.

5.2 Le concept de partage entre les arbres quaternaires

L'arbre quaternaire générique est basé sur le principe de partage de régions (quadrants) entre images. Soit I_m un ensemble des images .si un quadrant q a la même valeur dans un ensemble $I'_m \subset I_m$ cette valeur n'est stockée qu'une seule fois dans la base et est associée à l'ensemble des identificateurs des images I'_m. Dans ce cas on parle de partage explicite, parce que l'identificateur de chaque image partageant cette valeur apparait explicitement dans la liste des images associées à cette valeur.

Si les images de l'ensemble I_m sont organisées en arborescence, chaque image excepté la racine de l'arbre, à une mère unique et un nombre indéfini d'images filles. Par conséquent la règle de partage implicite suivante peut être introduite : '' excepté lorsque l'identificateur d'une image *i* est implicitement associé avec une autre valeur *v*, l'image *i* partage implicitement la valeur associée à son image mère ''.

Si l'arbre organisant les images est stocké, cette règle de partage implicite permet une représentation compacte de l'ensemble d'images, en particulier lorsqu'un grand nombre d'images partagent des valeurs de quadrants dans plusieurs branches de l'arbre.

5.3 Similarité entre images

Les images sont regroupées, dans la base de données ; en fonction d'une distance de similarité entre les arbres quaternaires qui représentent. Cette distance, appelée *Q-similarité* est proposée afin d'optimiser le stockage des images dans la base. La distance de *Q-similarité* entre deux images est définie par le nombre des nœuds différents (de même identificateur et de valeur différente) dans les deux arbres quaternaires représentants les images, divisé par le nombre d'identificateurs de nœud (sans doublon) apparaissant dans l'union des nœuds des arbres quaternaires des images

De manière plus formelle, on note *S (i, i')* l'ensemble des nœuds différents entre les arbres quaternaires des images *i*, *i'*. On note *U (i, j')* l'ensemble (sans doublon) des identificateurs de nœud apparaissant dans les arbres quaternaires des images *i* et *i'*. On note $|S(i,j')|$ (resp. $|U(i,i')|$) le nombre d'éléments de *S (i, i')* (resp. *U (i, i')*) .la distance de *Q-similarité* des images *i* et *i'* notée *d (i, i')*, est calculée par l'équation suivante :

$$d(i,j') = \frac{|S(i,i')|}{|U(i,i')|} = \frac{\text{nombre de nœuds différents}}{\text{Total des identificateurs de nœud (sans doublon)}}$$

- $d(i,i') \in [0,1]$, c'est une distance normalisée, c'est-à-dire :
  $d(i,i)=0$, $d(i,i')=d(i',i)$ et $d(i,i') \leq d(i,j)+d(j,i')$ (inégalité triangulaire).
- $d(i,i')=0 \Leftrightarrow$ les arbres quaternaires des images i et i' partagent tous leurs nœuds.
- $d(i,i')=1 \Leftrightarrow$ les arbres quaternaires des images i et i' ne partage aucun nœud.

Exemple : La distance de *Q-similarité* des images *a* et *b*, dont les arbres quaternaires sont représenté sur la figure 3.8 est égale a : $d(a,b) = \frac{|S(a,b)|}{|U(a,b)|} = \frac{5}{9}$.

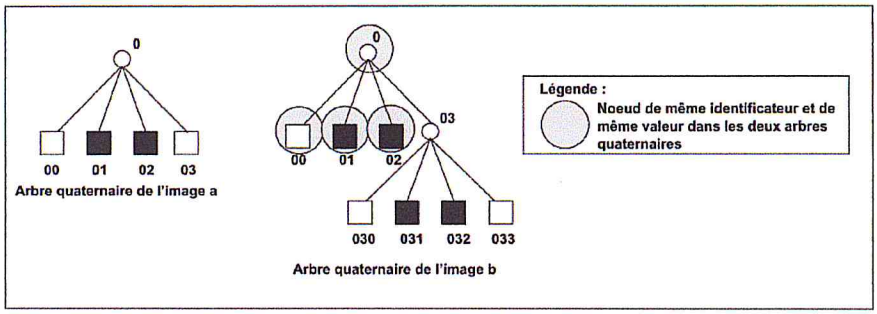


Fig.3.8 Le calcul du coefficient de *Q-similarité* des images a et b.

- ✓ $|U(a,b)|= 9$, car le nœud $0{,}0x$, et $03x$ apparaissent dans l'union des identificateurs des nœuds des arbres quaternaires des images a et b.

- ✓ $|S(a,b)|= 5$, car les nœuds 03 et $03x$, $x \in [0,3]$ sont des coefficients dans les deux arbres quaternaires.

5.4 L'arbre d'image

En conséquence de la règle de partage implicite, les images représentées par un arbre quaternaire générique sont organisées à l'aide d'une structure arborescente particulière, l'arbre d'images. Lorsqu'une nouvelle image est insérée dans l'arbre d'images, elle insérée comme fille de l'image dont elle est la plus similaire, c'est-à-dire dont la distance entre l'arbre quaternaire associée et celui de l'image à insérée est la plus proche de 0.

5.5 Nœuds génériques

La représentation et le stockage d'un ensemble des images similaire sont effectués dans un arbre quaternaire générique, dont les nœuds sont appelés nœuds générique. Pour chaque nœud apparaissant dans l'arbre quaternaire d'une image, il existe un nœud générique ayant le même identificateur dans l'arbre quaternaire générique. Un nœud générique n représente tous les nœuds n des arbres quaternaires des images de la base. Il contient toute l'information nécessaire pour recomposer la valeur du nœud de même identification dans chaque arbre quaternaire.

La valeur d'un nœud est soit ⊥ qui signifie que le nœud n'existe pas, soit *int* qui signifie qu'il est interne (il y a quatre fils). Chaque nœud générique peut être vu comme un tableau ayant plusieurs lignes. Chaque ligne *l* de nœud générique *n* contient une liste d'identificateurs d'images et une valeur *v* de nœud d'arbre quaternaire. La valeur *v* signifie que tous les nœuds identifiés par *n* ont pour valeur *v* dans les arbres quaternaires des images dont l'identificateur *i* apparait dans la liste.

De plus par application de la règle de partage implicite, on déduit que les nœuds *n* des arbres quaternaires de toutes les images descendantes des images de la liste, dans l'arbre d'images partagent implicitement cette valeur, excepté lorsqu'un identificateur d'image descendante apparait explicitement dans une autre ligne du nœud générique.

La figure 3.9 montre un exemple de construction d'arbre quaternaire générique

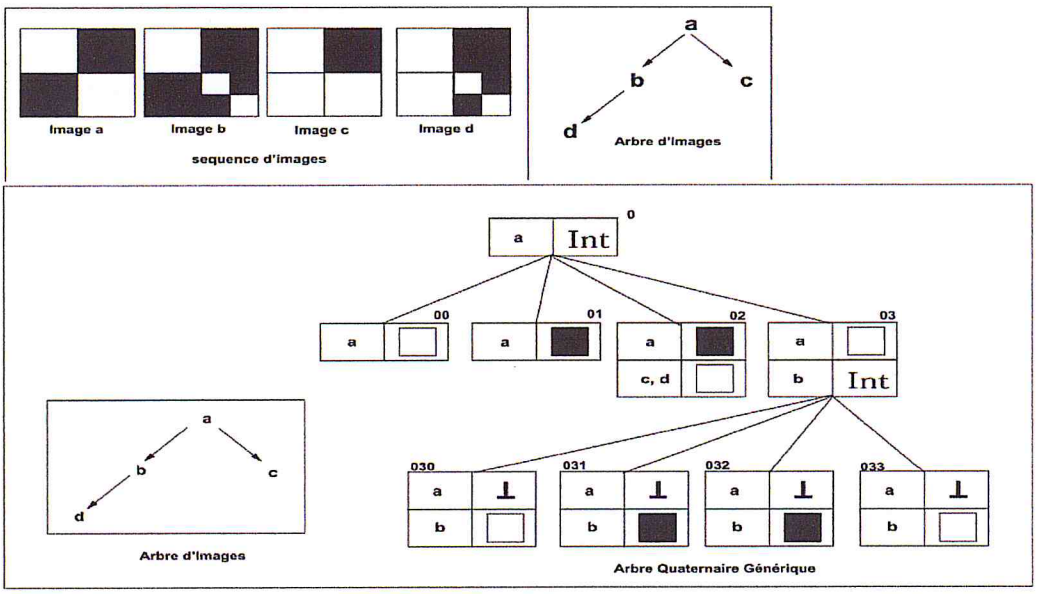


Fig. 3.9 L'Arbre Quaternaire Générique des images représentées (a, b, c, d).

6.0 Conclusion

Nous avons présenté de manière détaillée la structure de l'arbre quaternaire, et nous avons étudié son application dans les divers contextes, indexation (QUIP tree), stockage et compression (arbre quaternaire générique), la recherche par le contenue. La spécificité de cette structure nous a conduit à élaborer notre système de recherche d'images par le contenue par l'implémentation de cette dernier tout en exploitant leur caractéristique.

Chapitre 4:

Implémentation et Tests

1. Introduction:

Dans ce chapitre nous allons faire une étude de l'évaluation des performances du système d'indexation et recherche d'image par le contenu et la présentation des déférents algorithmes et techniques utilisées dans notre travail .en premier lieu nous présentons un algorithme globale pour le calcul de la distance associer aux arbres quaternaire et en deuxième lieu nous verrons les algorithme de calcul des déférents descripteurs (couleur, forme et texture) par la suite nous terminons par l'implémentation de la structure d'arbre quaternaire et en fin par une discussion de sur les déférents résultats obtenu .

2. Fonction de similarité

Comme nous avons cité dans le chapitre 3, nous avons choisi la distance générale Δ qui représente une somme pondéré de distanceδ_k entre les nœuds des arbres quaternaires.

La distance $\delta_k$peut porterdifférentes distancesentre les descripteurs visuels selon ce qu'on veut sauvegarder dans les nœuds de l'arbre quaternaire.

2.1 Distance entre descripteur visuel (distance δ_k)

Un système de recherche d'images par le contenu visuel, calcule la similarité visuelle entre le descripteur de l'image requête et les descripteurs des images de la base. Il y a différents types de mesure de similarité présentés dans la littérature.

Notre choix a porté sur la distance Euclidienne, L'idée de cette distance c'est la distance entre deux vecteur dans un espace vectoriel.

La distance entre deux descripteurs, c'est la distance entre deux points de cet espace, elle est donnée par la formule suivante :

$$dist(I_{req}, I_{cour}) = \left(\sum_{k=1}^{n} |V(K)_{requete} - V(K)_{courant}|^2 \right)^{1/2}$$

Où :I_{req} l'image requête et I_{cour} est l'image courante.

$V(K)_{requete}$ Le vecteur descripteur de l'image requête.

$V(K)_{courant}$ Le vecteur descripteur de l'image courante.

Mais dans l'arbre quaternaire, cette distance est calculée entre les descripteurs visuels contenus dans les nœuds du même niveau.

2.2 Algorithme de calcule de distance

Pour exploiter l'approche de l'arbre quaternaire dans la recherche des images, nous avons implémenté l'algorithme qui tient compte la localisation spatiale des images qui est une propriété de l'arbre quaternaire. Dans cette approche nous avons essayé de donner à l'utilisateur la possibilité de choisir le descripteur adéquat (couleur, texture , contours), pour chaque niveau, par un simple clic sur ce dernier. L'algorithme permet de calculer les distances inter quadrants par niveau.

La distanceutilisée est la distanceΔ , les coefficients $C_k$associés aux nœuds sont choisis de sorte qu'ils soient proportionnels à la surface des quadrants correspondant dans l'image. Par conséquent les coefficients choisis obéissant à la formule suivante:
$C_k = 4^{-p}$[JOM 05]Où p est la profondeur du nœud dans l'arbre (p=0 pour le niveau de racine, 1 pour le niveau suivant, et ainsi de suite).Plus le nœud est situé profondément dans l'arbre, plus son poids dans le calcul de distance Δ est faible,car l'aspect visuel qui caractérise (texture, couleur, forme) le nœud sera moins considéré dans l'étape de recherche.

Nous allons indexer nos images par des arbres quaternaires dont le nombre de niveaux est fixe (trois niveaux). Chaque image est, dans ce cas, représentée par un arbre quaternaire complet équilibré. Chaque nœud de l'arbre quaternaire contient de l'information sur la région correspondante dans l'image (couleur ,texture ,contours). En ce qui suit, nous représenterons l'algorithme permettant la manipulation de l'arbre quaternaire.

Algorithme de calcul de distance entre arbres quaternaires

Début

Entrée : deux images représentées par arbres quaternaire.

Sortie : distance entre ces deux images.

- ✓ ***Pour le niveau 0 (l'image toute entière):***

  - Calculer la distance entre la racine de l'arbre quaternaire de l'image requête avec l'une des racines de l'image de la base d'images.

  - Dans ce calcule, la distance $\delta_k(i,j)$ dépend du descripteur choisi.

  - Si le descripteur choisit est le contour(EDH), $\delta_k(i,j)$ sera la distance entre deux vecteur de contours.et C_k=1 pour p=0.

- ✓ ***Pour le niveau 1 :***

  L'image entière se trouvant dans la racine (niveau 0) se divise en quatre quadrants (Nous aurons quatre nœuds).Pour chaque quadrant il y'aura un calcul de distance δ_k Nous faisons la somme de ces distances puis nous les multiplions par le poids C_k=1/4 pour p=1.(Pour le niveau 1, p=1).

- ✓ ***Pour le niveau 2 :***

  -Chaque région du niveau 1 se divise en quatre quadrants (en totalité 16 nœuds).

  -Pour calculer les distances inter quadrant du niveau 2 nous utilisons la même démarche déjà cité au niveau 1 sachant que C_k=1/16. (Pour le niveau 2 p=2).

- ✓ *Pour chaque niveau, nous aurons une distance. Nous faisons la moyenne de ces dernières pour obtenir, à la fin, la distance finale entre l'image requête et l'une des images de la base sachant que pour calculer la distance entre deux arbres quaternaire (i,j)s, nous avons utilisé comme la distance* $\Delta(i,j)=\frac{\sum_k c_k \delta_k(i,j)}{\sum c_k}$. [MAR 02]

Fin

Algorithme 1. Calcul de distance entre arbres quaternaires.

3. Les descripteurs utilisés

Les descripteurs utilisées dans notre travail sont des descripteurs de la couleur, la forme et la texture qui ont été définies et montrés leurs efficacités dans la littérature [ZAH 01] nous allons les détaillées comme suit :

- ✓ Pour la couleur, nous calculons l'histogramme scalable de couleur (SCD) de la norme MPEG 7(qui est déjà présenté dans le chapitre 2).
- ✓ Pour la texture, nous calculons l'histogramme d'orientation des contours recommandé par MPEG 7 pour le Matching entre les textures non-uniformes.
- ✓ Pour la forme on fait la détection des contours de l'image par l'approche de convolution en utilisant un noyau (kernel) de Sobel.

3.1 Détection des contours

La détection de contours au sein d'une image est une caractéristique importante du processus de recherche d'images par le contenu. Face à un nombre important de techniques et filtres de détection de contours, il peut être difficile de choisir l'approche la plus adaptée à une collection spécifique d'images d'autant plus qu'il n'existe pas de technique optimale pour tous les cas de figures[GUE 04]. Par ailleurs, les techniques de détection de contours analysent souvent une image dans sa globalité sans tenir compte des spécificités des composantes de l'image.

3.1.1 Définition du contour

Dans une image, un contour peut être considéré de différentes manières. Nous allons décrire ici trois principales manières de considérer un contour : Premièrement, un contour peut être vu comme un changement brusque de l'intensité de l'image, Il existe plusieurs types de variations comme le montre la figure 4.1.

Fig.4.1. Profil de contours : marche, rampe, toit, pic.

Deuxièmement, une façon très proche de celle citée ci-dessus est de considérer les contours comme une différence sur la couleur.

Troisièmement, si on considère l'image comme étant un signal 2D, un contour peut être vu comme représentant les hautes fréquences du signal.

Cependant, ces représentations sont parfois insuffisantes pour représenter un contour. À titre d'exemple, nous remarquons que parfois notre cerveau est capable de repérer des contours invisibles sur l'image car il a la capacité d'utiliser ses connaissances pour extraire des formes d'une image. Par exemple, le cerveau humain est capable de voir le triangle dans l'image suivante :

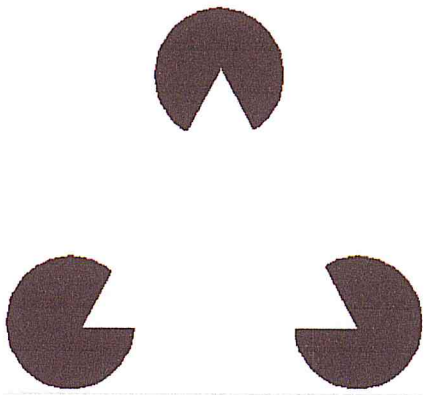


Fig. 4.2 Triangle invisible pour un ordinateur

3.1.2 Définition du filtrage

Le principe du filtrage consiste à modifier la valeur des pixels d'une image, généralement dans le but d'améliorer son aspect. En pratique, il s'agit de créer une nouvelle image en se servant des valeurs des pixels de l'image d'origine. N'entrent pas dans la catégorie du filtrage toutes les transformations de l'image d'origine : zoom, découpage, projections.

Il existe deux types de filtrage global et local :

- ✓ **Filtrage Global :**Dans le filtrage global, chaque pixel de la nouvelle image est calculé en prenant en compte la totalité des pixels de l'image de départ. Dans cette catégorie on trouve, par exemple, les opérations sur les histogrammes ou les opérations qui nécessitent de passer dans l'espace de Fourier(Lorsque le système récupère le signal il remplit un espace mathématique qui est appelé espace de fourrier).
- ✓ **Filtrage local :**Dans le filtrage local, chaque pixel de la nouvelle image est calculé en prenant en compte seulement un voisinage du pixel correspondant dans l'image d'origine. Il est d'usage de choisir un voisinage carré et symétrique autour du pixel considéré. Ces voisinages sont donc assimilables à des tableaux à deux dimensions (matrices) de taille impaire.

3.1.3 Filtrage Local Linéaire

Un filtre linéaire transforme un ensemble de données d'entrée en un ensemble de données de sortie selon une opération mathématique nommée convolution. Quand il s'agit de données numérisées comme dans le cas du traitement d'image, la relation entre les valeurs des pixels de sortie et celle des pixels d'entrée est décrite par un tableau de nombres, le plus souvent carré, nommé ***matrice de convolution***. Le temps de calcul est fréquemment diminué quand on veut séparer un filtre en deux filtres dont la convolution mutuelle sert au reconstituer. Cette remarque est utilisée surtout pour créer un filtre à deux dimensions à partir de deux filtres à une seule dimension (vecteurs) dans le sens horizontal et le sens vertical.

Il existe plusieurs filtres linéaires (uniforme, pyramidal), mais dans le cadre de notre travail on s'intéresse au filtre gaussien que nous détaillons au-dessous.

- ✓ **Filtre gaussien :** Appelé également *gaussian filtering*. Le principe de ce filtrage est une convolution avec une gaussienne.Nous rappelons l'expression d'une gaussienne en dimension 2, de moyenne nulle :

$$G_\sigma(\mathbf{x}) = \frac{1}{2\pi\sigma^2} e^{\left(-\frac{|\mathbf{x}|}{2\sigma^2}\right)}$$

Pour effectuer une convolution avec une gaussienne, on utilise un masque (kernel) de convolution obtenu par discrétisation d'une gaussienne sur un noyau généralementde taille (2p+1 x 2p+1).

Certains masques sont à coefficients entiers pourpermettre des calculs plus rapides, voire à coefficients puissance de deux (unemultiplication ou division par 2 d'un entier revientà un décalage de 1 desbits le composant).

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 2 & 1 & 1 \\ 1 & 2 & 4 & 2 & 1 \\ 2 & 4 & 8 & 4 & 2 \\ 1 & 2 & 4 & 2 & 1 \\ 1 & 1 & 2 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 2 & 2 & 2 & 1 & 1 \\ 1 & 2 & 2 & 4 & 2 & 2 & 1 \\ 2 & 2 & 4 & 8 & 4 & 2 & 2 \\ 2 & 4 & 8 & 16 & 8 & 4 & 2 \\ 2 & 2 & 4 & 8 & 4 & 2 & 2 \\ 1 & 2 & 2 & 4 & 2 & 2 & 1 \\ 1 & 1 & 2 & 2 & 2 & 1 & 1 \end{bmatrix}$$

Fig. 4.2. Exemples de noyaux gaussiens

Le lissage gaussien permet de corriger le bruit dans les parties homogènesdes images mais est moins efficace que le lissage moyenneur. Cependant, ildégrade moins les détails que le lissage moyenneur [LIN 04].

2.3 Principe de détection de contours [VIN 04]

Afin de détecter les contours la majorité des algorithmes, Utiliser un filtrage linéaire afin d'appliquer les équations de détection à l'image. Le filtrage d'un signal consiste à éliminer certaines fréquences dans ce signal. Et comme nous avons cite au-dessusla principale technique pour appliquer un filtrage linéaire à une image c'est à faire une convolution de l'image avec le filtre. Si on considère le filtre h de taille $n*m$ et une image I de taille $p*q$, alors la convolution s'écrit :

$$(I * h)(i,j) = \sum_{k=-n/2}^{n/2} \sum_{l=-m/2}^{m/2} I(i+k, j+l)h(k,l)$$

En pratique,la convolution numérique sert à parcourir les pixels de l'image et à les modifier en fonction de voisinage de pixel parcouru à un moment donné et de filtre de convolution à appliquer. Un filtre est généralement une matrice de taille impaire et symétrique comme 3 par 3, 5 par 5, 7 par 7, etc. le fait que la matrice soit de taille impaire permet de définir le pixel à

traiter situe au centre de la matrice, et son voisinage constitue au par les autre pixels, la figure 4.3 montre un exemple d'application de convolution numérique sur une image[REY 12].

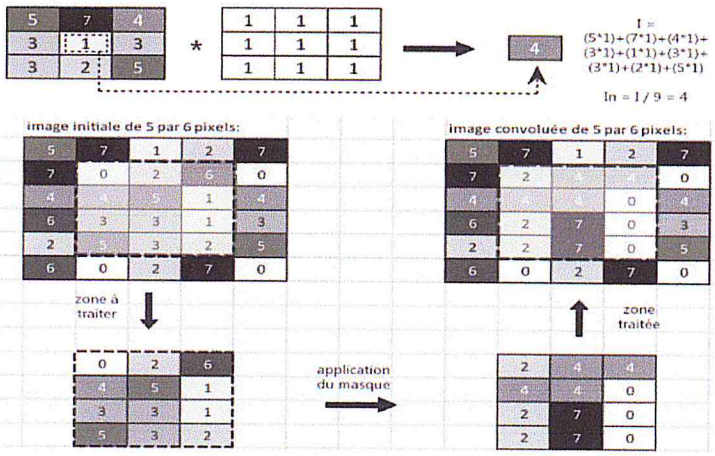


Fig.4.3exemple d'application de la convolution.

Lorsque l'on considère un contour comme une variation de l'intensité lumineuse, on pense tout naturellement à la dérivation pour déterminer les variations de l'image. Dans ce cas, on applique généralement la dérivation à la luminance de l'image. Cette dérivation va donc annuler les zones de faibles variations (intensités uniformes) et retourner de forts coefficients lors de variations importantes (contours) et cette dérivation est nommé le gradient de l'image .

Tout d'abord, afin de pouvoir effectuer une dérivation à l'image, il va falloir discrétiser les équations de dérivation car on ne peut appliquer une dérivation continue sur une fonction discrète. L'élément minimal étant le pixel, les dérivées premières (gradients) dans les directions x et y s'écrivent :

$$\frac{dI}{dx}(x,y) = I(x-1,y) - I(x,y) \text{ et } \frac{dI}{dy}(x,y) = I(x,y-1) - I(x,y)$$

Dans la suite, nous verrons plusieurs détecteurs des contours basés sur la dérivation discrète d'une image. Comme la manière de faire cette dérivation n'est pas unique, il existe un grand nombre de filtres permettant d'approcher les résultats. Ce sont ces filtres que nous présentons maintenant.

2.3.1 Détecteurs des contours basés sur le gradient

Il existe plusieurs détecteurs mais nous abordant quelque exemple, puisque notre choix est porté sur le détecteur de Sobel.

- ✓ **Détecteur de robert :** La technique de détection proposée par Robert est une application directe du calcul du gradient et consiste à utiliser deux filtres linéaires pour calculer les dérivées dans les directions π/4 et 3π/4 :

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} et \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Ce détecteur est malheureusement très localisé et donc très sensible au bruit, ce qui nuit grandement à ses performances. Les autres sont moins sensibles au bruit et offrent de meilleurs résultats.

- ✓ **Détecteurs Sobel et Prewitt**

Ces détecteurs utilisent un filtre linéaire qui a l'avantage d'effectuer deux opérations en même temps, soit un lissage de l'image et une dérivation. Ils restent sensibles au bruit mais donnent de bons résultats sur des images non bruitées. Voici les deux filtres impliquant un opérateur horizontal et un opérateur vertical.

$$\begin{bmatrix} 1 & \alpha & 1 \\ 0 & 0 & 0 \\ -1 & -\alpha & -1 \end{bmatrix} et \begin{bmatrix} 1 & 0 & -1 \\ \alpha & 0 & -\alpha \\ 1 & 0 & -1 \end{bmatrix} \begin{cases} \alpha = 1 \; pour \; prewitt \\ \alpha = 2 \; pour \; sobel \end{cases}$$

Finalement voici un exemple de la détection de contours.

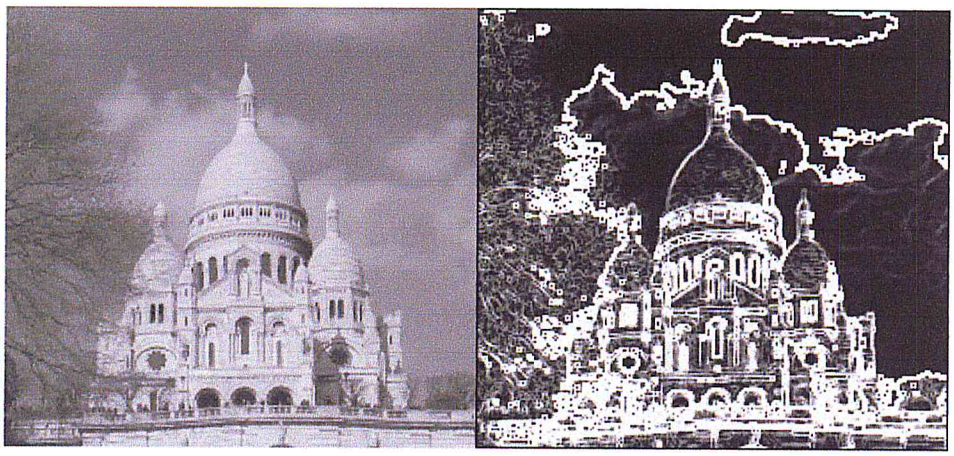


Fig. 4.4 exemple de la détection des contours

3.3 Algorithme de détection des contours

Afin de bien comprendre les étapes de détection de contours nous écrivons l'algorithme suivant

---

: *Début*

Entrée : image

Sortie : Vecteur d'orientation des contours

Gaussien⟵: matrice [3x3]

Sobel_kernel⟵: matrice [3x3]

Gradient _H⟵ : vecteur

Gradient _V⟵ : vecteur

Orientation_contour⟵: vecteur

*1-**initialiser le filtre gaussien.**// utiliser pour éliminer le bruite dans l'image.*

✓ *convertit l'image en niveau de grise.*

✓ *gaussien ⟵get_Filtre ().//calculer le filtre en utilisant la formule gaussien.*

2-Appliquer la convolution avec le filtre gaussien sur l'image en niveau de grise

✓ *Image_convolution (images grisé,gaussien).*

3-Détection des contours à partir de l'image convoluée.

✓ *affecter le kernel Sobel Horizontale.*

// Calculer le gradient par rapport le kernel Sobel Horizontale

Gradient_H⟵ caculer_gradient (images_Convoluée,Sobel_kernel_H).

✓ *affecter le kernel Sobel Verticale*

Gradient_V⟵ caculer_gradient (images_Convoluée, Sobel_kernel_V).

Orientation_contour⟵ detecte_edge (images_Convoluée,Gradient_H, Gradient_V)

Fin

Algorithme.2. Algorithme de détection des contours

3.4 Algorithme d'extraction d'histogramme orientation des contours[ZAH]

L'EHD capture la distribution spatiale des contours et voici l'algorithme correspondant.

Début

Entrée : image

Sortie : histogramme d'orientation des contours(EHD)

Histogramme_Orientation_contour ⟵: vecteur

Histogramme_Orientation_local⟵ : vecteur

1-partitionement de l'image en 4x4 sous images (16 sous images)

- ✓ *Pour chaque sous images deviser la en 16 images-blocs*

2-géneration d'histogramme local (des contours) pour chaque 64 image bloc.

- ✓ *Pour chaque image bloc.*
- ✓ *détecter les contours selon 5 orientations (déjà citées) en appliquons les filtres correspondants pour chaque images bloc.*

  Histogramme_Orientation_local ⟵ detecte_edge (images bloc, Filtre).

3-compter le nombre total des contours pour chaque sous images

- ✓ *compter le nombre total des contours pour chaque sous images, sachant que il y a 5 type des contours pour 16 images blocks, on peut définir que les nombres des histogrammes local égale à 16 X 5 =80 pour chaque sous images.*
- ✓ *Normaliser chaque histogrammes local par la devisons de ces éléments par le nombre total des blocks avec un contour dans le sous images correspondant.*
- ✓ *Pour obtenir l'histogramme final on fait l'opération de quantification*

Histogramme_Orientation_contour ⟵ : Quant(Histogramme_Orientation_local)

*La quantification se fait en appliquons un algorithme nommé : **Lyond-Max*** [DON]

Fin

Algorithme.3.algorithme d'extraction d'histogramme orientation des contours(EHD)

3.5 Algorithme d'extraction d'histogramme scalable de couleur (SCD) [ZAH]

Le SCD on peut le considéré comme étant un histogramme de couleur dans l'espace HSV encodé par le transformé de Haar,et voici l'algorithme.

Début

Entrée : image

Sortie : histogramme scalable de couleur(SCD)

Haar_Transforme_histogramme ⟵: vecteur

1- *Extraire les vecteur des composant chromatique R, G, B à partir de l'image en entré.*
2- *Pour chaque tuples(R, G, B).*
3- *Converti l'ensemble des tuples à l'espace de couleur HSV*
4- *Construire l'histogramme de couleur de l'espace HSV.*
5- *Appliquer une procédure de quantification à l'histogramme construit.*
6- *Normaliser l'histogramme de couleurs quantifié.*

Haar_Transforme_histogramme ⟵ Haar _ Transforme (Histogramme _normalisé).

Fin

Algorithme.4.algorithme d'extraction d'histogramme scalable de couleur(SCD)

Remarque :l'algorithme présenté au-dessus semble très généralisé, en réalité nous avons décrit que les fonctionnalités globales de l'étape de l'extraction. La principalefonctionnalité de ce descripteur c'est le transforme de Haar qui est un transformé signal,ce dernier offre une représentation multi-résolution(uniformité des couleurs) aux couleurs présentes dans l'image. Mais ce descripteur reste aveugle par rapport à la distribution spatiale des couleurs.

4. Protocol d'évaluation

Pour évaluer notre approche de combinaison. Nous avons définir un ensemble de catégories, regroupant dans les mêmes classes des images similaires. Pour cela, nous avons construit à partir de base de données (Corel) contient 2381 images de taille 256x256 pixels 15 ensembles d'images de test dans plusieurs différents dossiers que nous avons regroupés.

On trouve des ensembles des images contient des orientations différentes, et d'autre contient des images hétérogènes, hautement texturées, scènes diffèrent… etc.

La figure 4.5 montre un exemple sur les différentes natures des images des tests.

Fig. 4.5 différentes ensemble des images des tests

Deux critères bien connus pour l'évaluation des requêtes par similarités ont été retenus pour mesurer les performances système CBIR. Il s'agit des scores suivants :

- ✓ ***Précision*** **:** La précision est le rapport entre le nombre d'images pertinentes dans l'ensemble des images trouvées et le nombre d'images trouvées.
- ✓ ***Rappel*** **:** Le rappel est le rapport entre le nombre d'images pertinentes dans l'ensemble des images trouvées et le nombre d'images pertinentes dans la base d'images.

En ce qui concerne les deux scores de l'évaluation (précision, rappel), on donne un exemple pour mieux comprendre les choses.

Supposant que le système a retourné 15 images par rapport à 100 en totalité dont le nombre d'images pertinente est 7, alors précision=7/15=46.66 % et rappel =7/100=7 %.

5. Implémentation de l'arbre quaternaire

Notre avons programmé notre application avec le langage C# de Framework .net version 3.5, et pour l'implémentation de l'arbre quaternaire nous avons créé deux classe quadtreeNode (le nœud de l'arbre quaternaire) et class QuadTree qui représente l'arbre quaternaire.

Toutes les méthodes et les propriétés de ces classes nous détaillent maintenant :

5.1 Class QuadtreeNode

5.1.1 Les attributs

-**Position** : énumération contient les postions de nœud dans l'arbre a comme valeurs {NW, NE, SW, SE, ROOT (pour le nœud racine)}

- **ID :** c'est l'identifiant de nœud.
- **Depth :** c'est la profondeur de nœud.
- **Poids :** c'est le poids de nœud.
- **Parent :** c'est le parent de nœud (parent=nulle si le nœud est racine).
- **Fils :** collection des objets quadtreeNode contient les quatre fils d'un nœud.
- **Bound :** rectangle représente les bords de nœud.
- **Image :** image représente une partie de l'image de l'arbre quaternaire.
- **Vector :** un tableau représente le descripteur visuel sauvegardé dans le nœud.

5.1.2 Les méthodes :

- **quadtreeNode () :** constructeur par défaut de nœud
- **quadtreeNode** (parent, position, rectangle, image, x, y, x1, y1) **:** le constructeur principale de nœud.

-**subTree** (nœud) : méthodes chargé de deviser un nœud en et retourner les quatre fils.

-**Extractor** () : méthode chargé de extraire les descripteurs visuel dans le nœud.

5.2 Class QuadTree

5.2.1 Les attributs

-root : racine de l'arbre quaternaire de type quadtreeNode.

5.2.2 Les méthodes

-QuadTree () : constructeur par défaut de l'arbre quaternaire.
-QuadTree (image) : constructeur charge de crée l'arbre quaternaire.
-getAllChild () : méthode qui retourne tous fils de l'arbre quaternaire.
-getChidNode (nœud) : méthode qui retourne tous fils de d'un nœud d'arbre quaternaire.
-countIndepth (profondeur) : méthode qui compter le nombre des nœuds d'un niveau.
-getNodeIndepth (profondeur) : méthode qui retourne les nœuds d'une profondeur.
-getAllImage () : méthode qui retourne tous les images des nœuds d'arbre quaternaire.
-getImageIndepth (profondeur) : méthode qui retourne tous les images des nœuds d'arbre quaternaire pour une profondeur donné.
-getSubTree(ID) : méthode qui retourne tous les fils d'un nœud à partir de son ID.
-countInSubtree(ID) : méthode qui compter le nombre des fils (nœuds) dans un sous arbre d'un nœud à partir de son ID.
-distance (QuadTree, QuadTree) : méthode qui calcule distance entre deux arbres quaternaires.

6. Tests expérimentaux et discussion

Tous les tests ont été réalisés dans un pc portable qui possède un processeur Intel core 2 duo 2.13 GHZ et de RAM 4GB. Pour mesurer la pertinence de recherche, nous avons choisi d'utiliser plusieurs descripteurs de plusieurs niveaux de l'arbre quaternaire. En totalité, nous avons 27 tests à réaliser en utilisant tous les combinaisons possibles. Chaque combinaison parmi les 27 a été utilisée avec plusieurs catégories d'images : peu texturées, fortement texturées….etc. Chaque test comporte un calcul de deux scores (précision et rappel) qui nous sert à mesurer la qualité de la réponse du système d'une façon claire. Les résultats obtenus seront interprété selon le choix de descripteur visuel. La fenêtre principale de notre application est représentée dans l'image ci-dessous. Cette dernière, contient une interface facile à comprendre dans laquelle l'utilisateur peut s'en servir pour entrer le seuil souhaité.

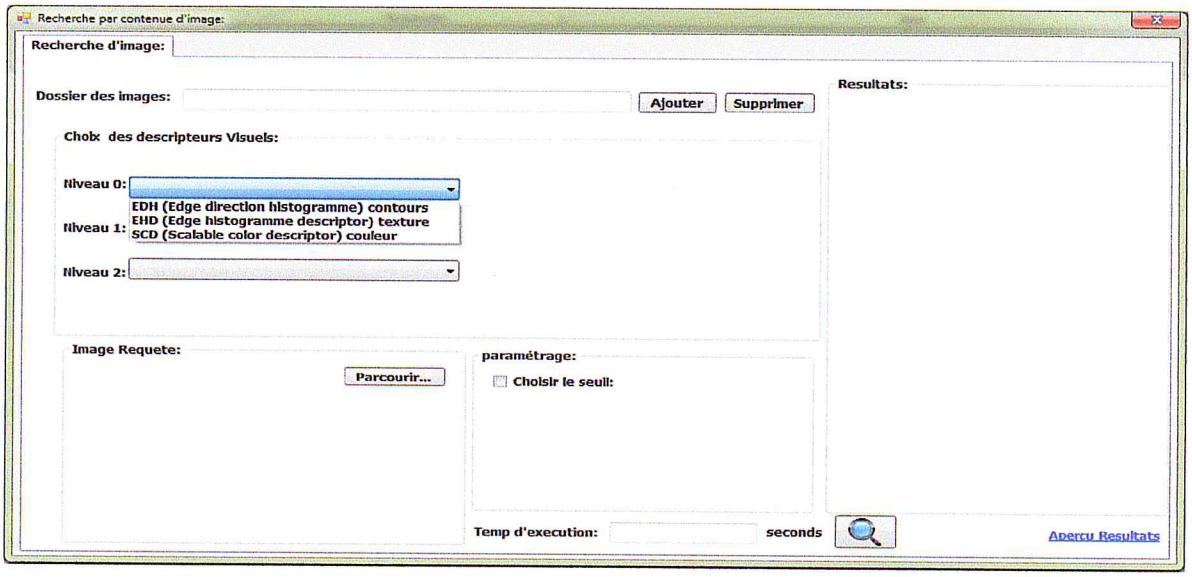


Fig.4.6 Fenêtre principale de l'application.

Pour réaliser les tests, nous avons choisi plusieurs ensemble des images et d'évaluer diffèrent configuration (choix de descripteur) sur différent requête, afin de bien évaluer les résultats obtenue, et à la fin des tests, il y aura un tableau global pour les 27 tests sur une ensemble des images avec les valeurs de la précision et le rappel.

- **Test numéro 1**(Niveau 0 : contours, Niveau 1 : contours, Niveau 2 : contours)

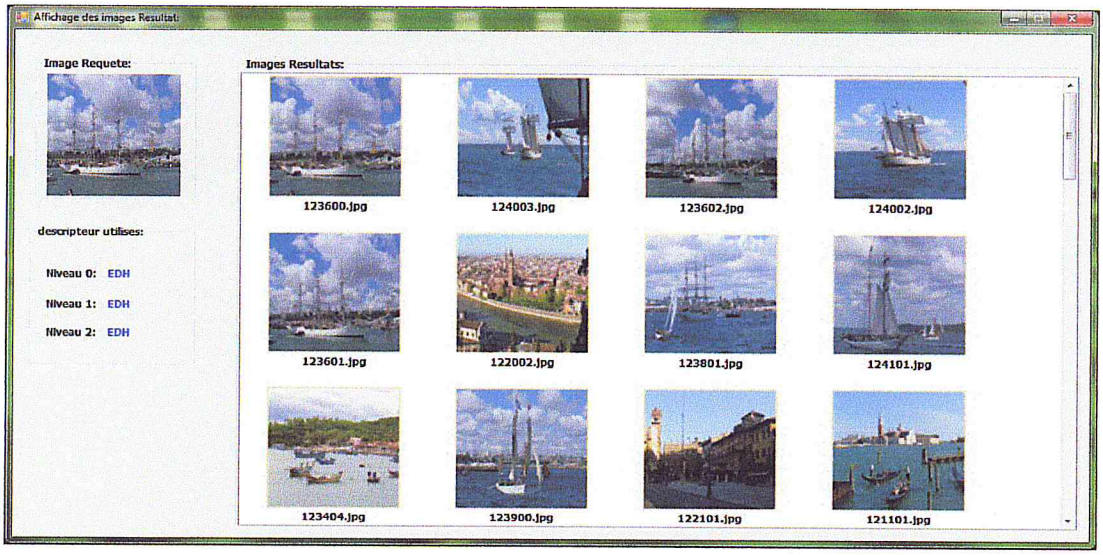


Fig.4.7 exemple de recherche en utilisant les contours pour chaque Niveau.

Description : le nombre d'images retournés ici est 52 dont il y a 17 images pertinentes alors la précision est 17/52 =32.69 % et le rappel est égale à 17/103=16.50 %.

➢ **Test numéro 2**(Niveau 0 : contours, Niveau 1 : contours, Niveau 2 : textures)

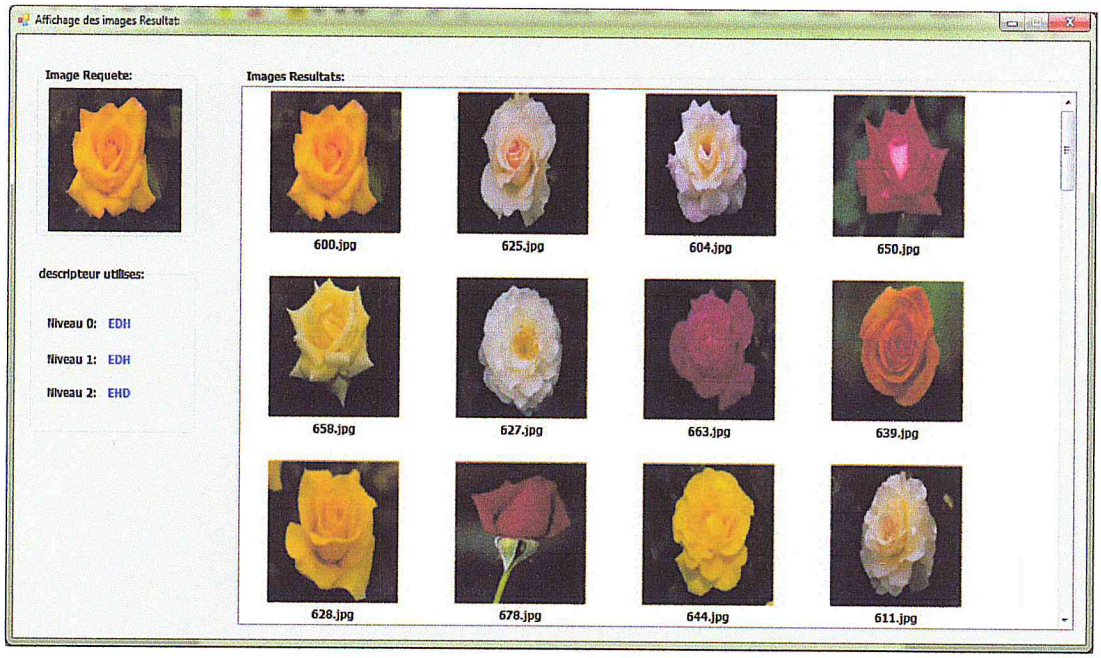


Fig.4.8 exemple de recherche en utilisant les contours (Niveau 1 et 2) et la texture.

Description : précision=40%, rappel=17.54 %.

Remarque : l'évaluation des résultats dépend de descripteur choisit pour chaque niveau comme nous l'avons dit auparavant. Le descripteur choisit au niveau haut de l'arbre aura plus de poids dans la recherche. On remarque dans cet exemple les images retournés sont similaires de point de vue de forme, mais la coloration est un peu différent, ce que est expliqué par l'absence de descripteur de couleur.

Autrement dit, la pertinence de l'image retournée diffère d'un cas à une autre. Par exemple, Si on choisit au *niveau 0* le descripteur du contour la pertinence sera mesure de point de vue de forme.si on choisit dans le même niveau le descripteur de couleur, les résultats obtenus seront basées sur la couleur. Plus précisément, au *niveau 1* la pertinence des images retournes sont bases sur la similarité entre les quadrants de celles-ci.

De plus, au *niveau 2* la pertinence de recherche d'image obtenue sera basée sur les détails de l'image, ce que nous donnent des résultats plus précis.

➢ **Test numéro 3 :** (Niveau 0 : couleur, Niveau 1 : couleur, Niveau 2 : couleur)

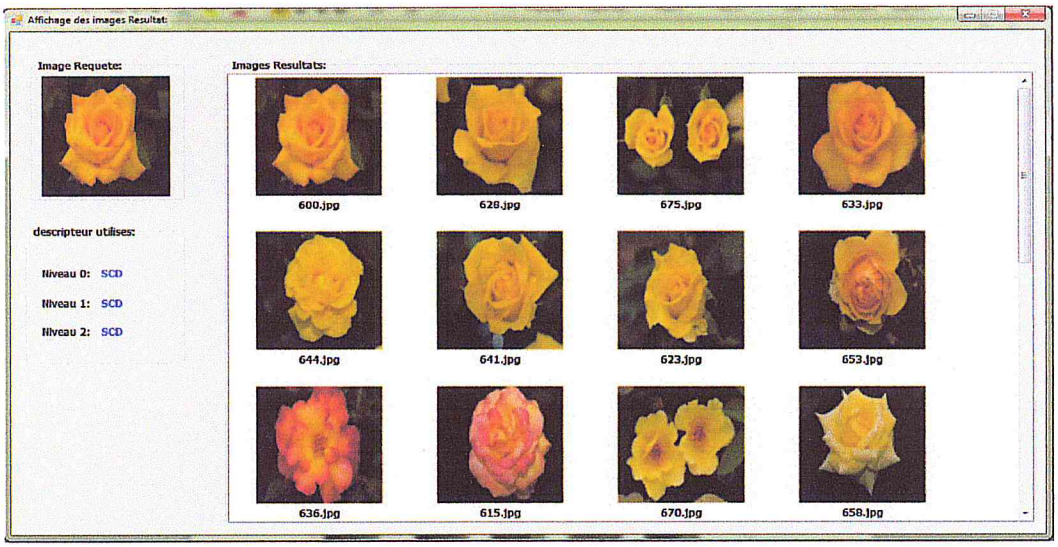


Fig.4.9 exemple de recherche en utilisant la couleur (Niveau 1 et 2 et 3).

Description : précision=50%, rappel=8.77 %.

D'une part, cette figure nous présente les résultats de recherche après qu'on a utilisé seulement la couleur. De ce fait, on a obtenu des résultats suffisamment clairs vis-à-vis la couleur. D'autre part, nous observons que le descripteur de couleur (SCD) offre des performances inferieurs (précision inferieur) quand les images à tester contient plus d'informations spatiales (textures, formes, orientations différentes, texture homogènes), ce qui nécessite d'ajouter d'autres descripteurs de texture ou contours pour que les résultats soient pertinentes. Les figures 4.10 et 4.11 nous confirment cette interprétation après qu'on a effectué une requête sur une collection de 100 images hétérogènes.

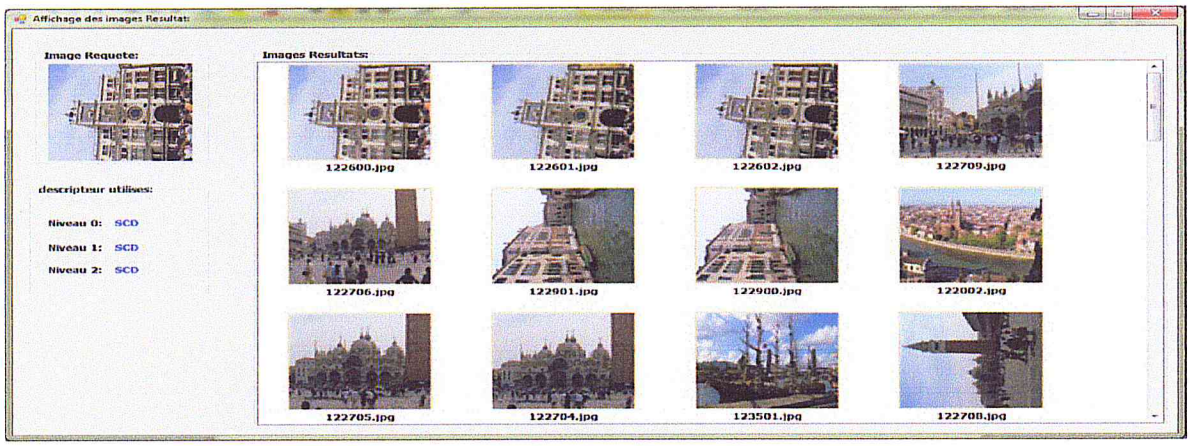


Fig.4.10 exemple de manque de précision en utilisant l'attribut couleur.

Description : précision=11.11%, rappel=5.82 %.

Suivant la figure 4.10 on constate que le système a donné des images complètement différents de l'image requête malgré que les couleurs sont proches. Cela est évidement dû à la complexité et à la richesse de contenue visuel de l'image requête. Ceci implique la nécessité d'ajouter d'autres descripteurs de texture et de forme pour que les résultats soient pertinents.

En voyons la figure 4.11 nous montre une bonne précision de recherche en utilisant les contours au niveau 0, texture au niveau 1 et la couleur au niveau 2.

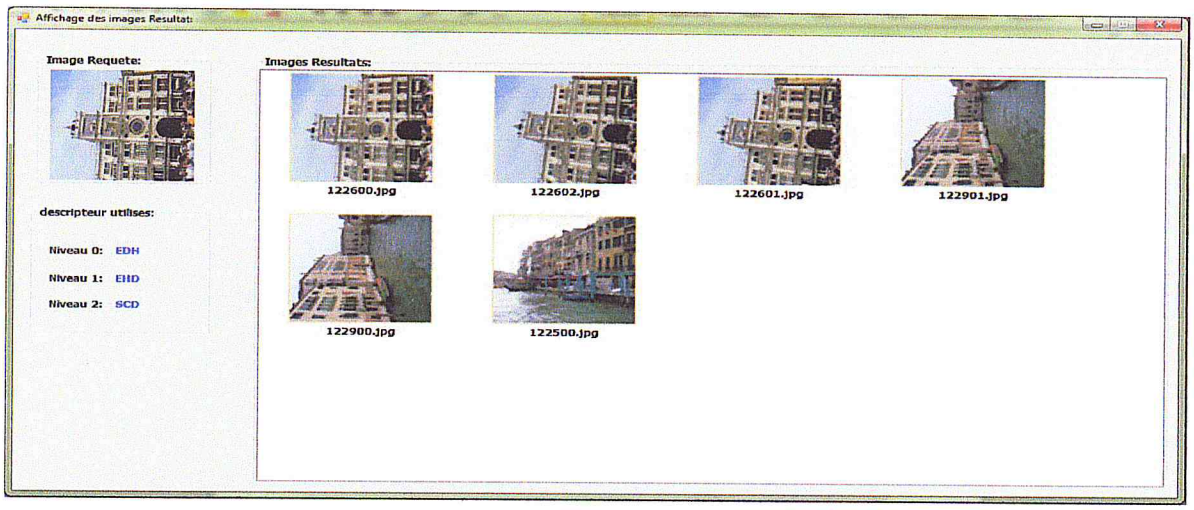


Fig.4.11 exemple de recherche avec une bonne précision.

Description : précision=50%, rappel=1.94 %.

Nous avons examiné les requêtes de similarité basées sur la couleur, et nous avons vu que le l'attribut de couleur reste insuffisants pour représenter les différentes informations visuelles de l'image d'une façon pertinente.

Examinons à présent les performances des requêtes basées sur la texture(EHD). La figure 4.12 nous montre un exemple. Le test représenté par cette figure est réalisé sur une collection des images hautement texturées contenant 364 images.

➢ **Test numéro 4 :** (Niveau 0 : texture, Niveau 1 : texture, Niveau 2 : texture)

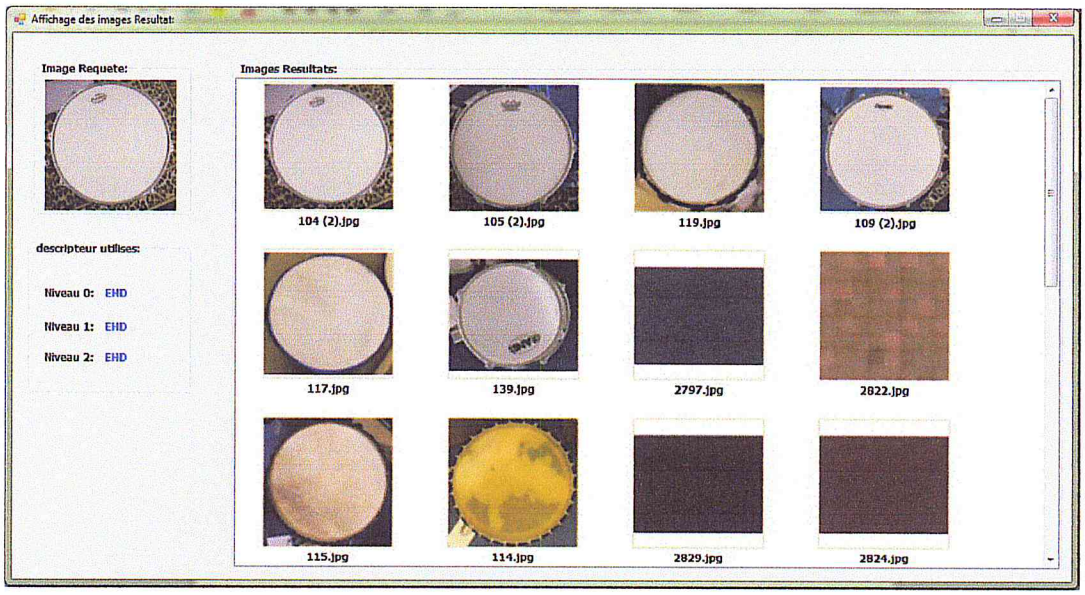


Fig.4.12 exemple de recherche en utilisant la texture (pour tous les niveaux).

Description : précision=41.66%, rappel=2.74 %.

Voyons dans cet exemple que le système donne 14 fausses images par rapport aux 25 images retournées. Ceci est relativement dû à la nature des images (hautement texturées), puisque l'histogramme d'orientation de contours (EHD) détecte les contours de l'image selon cinq directions et que ces dernières sont moins détectables si l'image est assez bruitée ou structurée (assez des détails dans l'image) ; comme le cas de l'arbre quaternaire, l'image sera plus homogène ou structurée a un niveau plus bas dans l'arbre (Niveau 3 par exemple).

De plus le EHD est recommandé par la norme MPEG 7 pour le Matching entre les images qui contient des textures non uniformes [ZAH 01].

Mais on constate que la combinaison de texture (au niveau 0) et la couleur (Niveau 1 et 2) donne des résultats plus performants. La figure 4.13 nous montre ces résultats.

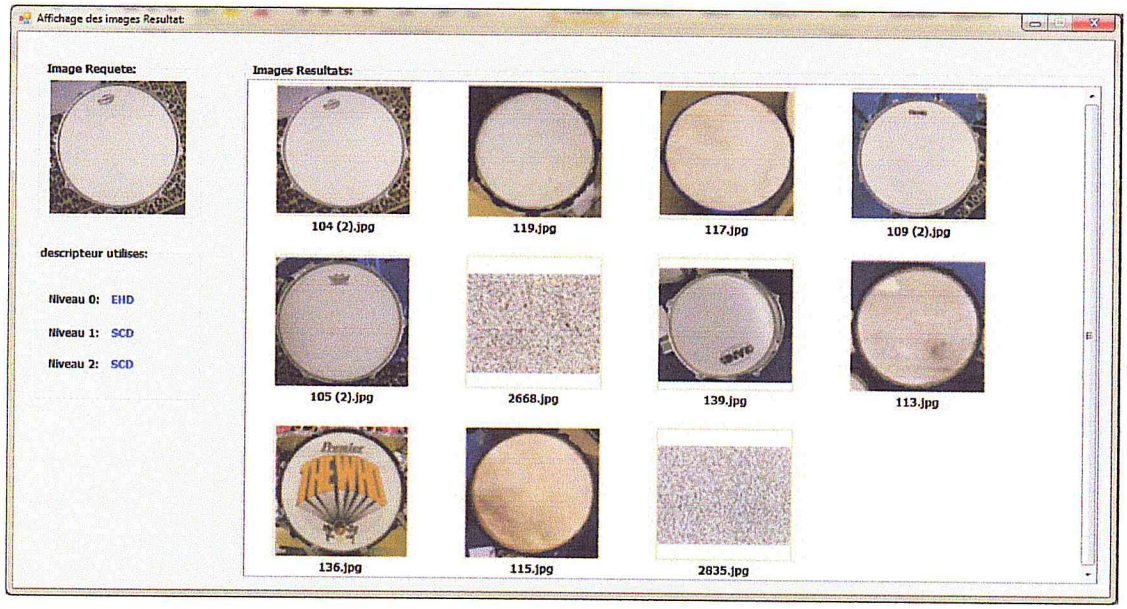


Fig.4.13 exemple de recherche avec une bonne précision (combinaison de couleur et texture)

Description : précision=81.81%, rappel=2.19%.

Remarquons que le système a retourné juste 2 images fausses. Essayons maintenant cette combinaison (couleur et texture) sur une autre collection des images non texturées, les résultats obtenus sont représentés par la figure 4.14.

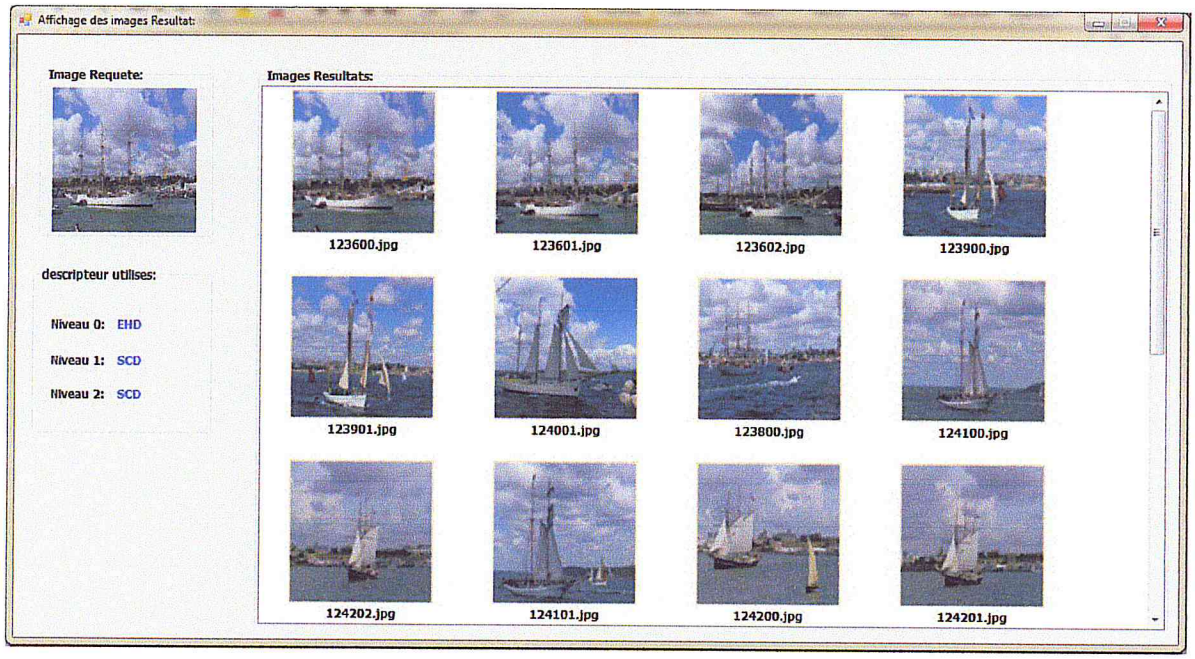


Fig.4.14 autre exemple de recherche avec une bonne précision (combinaison de couleur et texture).

Description : précision=73.91%, rappel=16.19%.

Examinons à présente le descripteur de textures sur différentes orientations, comme nous avons dit EHD calcule les orientations de contours selon 5 directions respectivement, $0^{\circ}, 45^{\circ}, 90^{\circ}, 135^{\circ}$, et non directionnelle. Pour mesurer ces orientations nous avons identifié manuellement une collection des images contenant différentes directions. Les résultats obtenus sont présentés dans la figure 4.15.

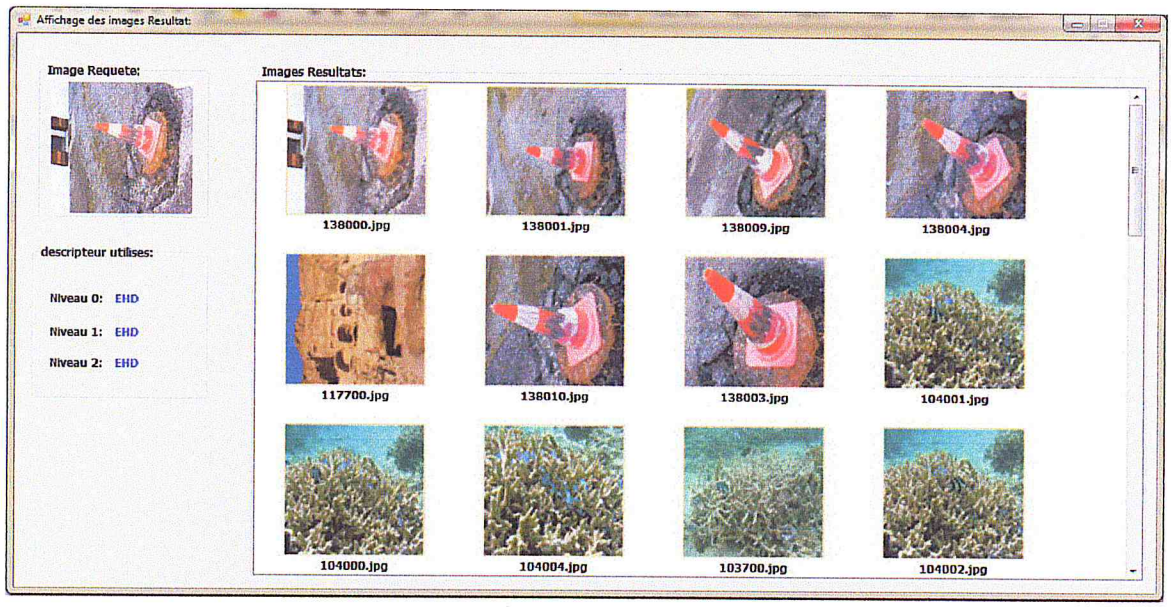


Fig.4.15 exemple de recherche sur différents orientations (utilisation de texture).

Description : précision=27.02%, rappel=9.8%.

Suivant à la figure 4.15 nous remarquons que la précision chute drastiquement (27 %), mais en ajoutant le descripteur couleur les résultats obtenus sont vraiment excellents (précision 100%). Ceci confirme notre interprétation que le descripteur de texture ne donne pas des résultats si on n'ajoute pas le descripteur de couleur.

La figure 4.16montre les résultats obtenue en ajoutant le descripteur de couleur.

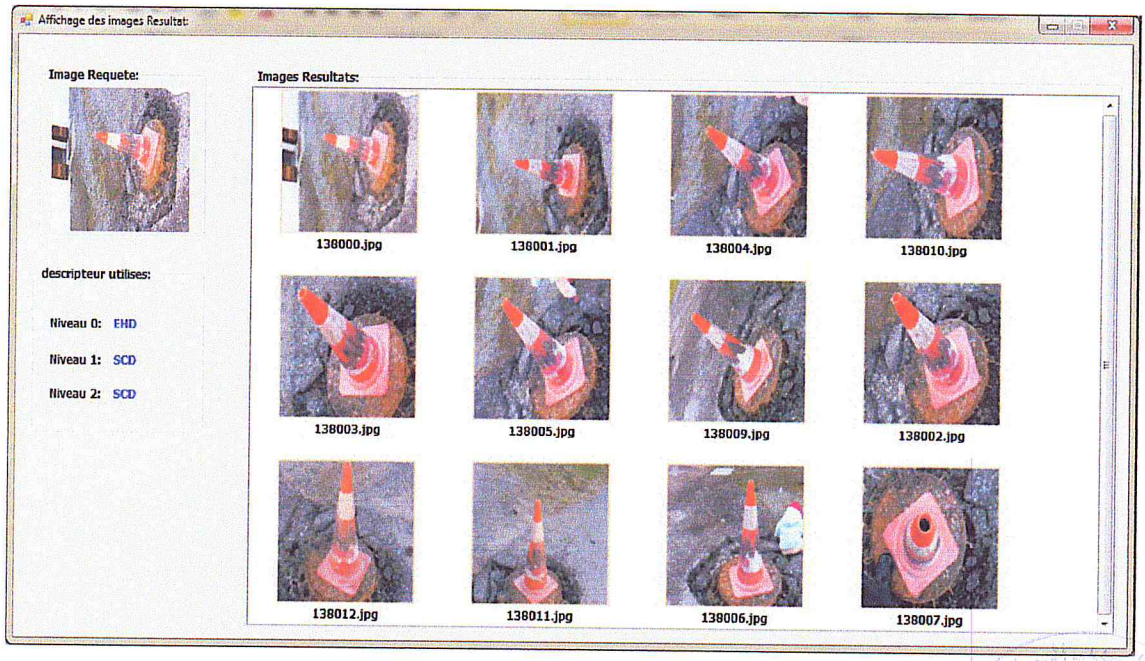


Fig.4.16 exemple de recherche avec une précision complète (utilisation de texture et couleur).

Description : précision=100%, rappel=10.78%.

Essayons maintenant un autre exemple sur une collection des images contenant des orientations différentes, mais la distribution des couleurs est quasi-uniforme.

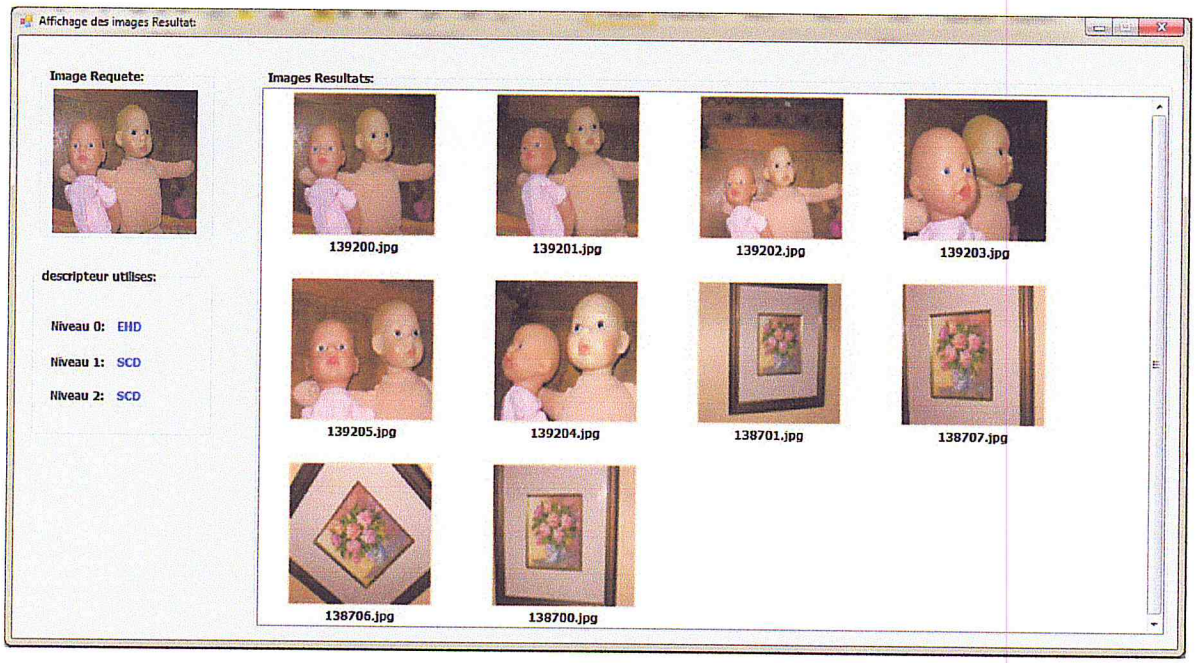


Fig.4.17 autre exemple de recherche avec une précision modéré (utilisation de texture et couleur).

Description : précision=50%, rappel=5.26%.

D'après la figure 4.17 on constate que lorsqu'il y a une distribution homogène de couleur dans les images à chercher, La combinaison de texture et couleur n'offre pas toujours des bonnes performances. Ceci est expliqué par le fait que le descripteur de couleur (SCD) qui mesure la scalabilité de couleur (la distribution des couleurs) lorsque les couleurs des images sont uniformes. Ce qui implique d'ajouter le descripteur de contours pour qu'on caractérise le contenue visuel de l'image requête d'une façon plus riche.

La figure 4.18 nous montre un exemple de combinaison de texture, couleur, contours.

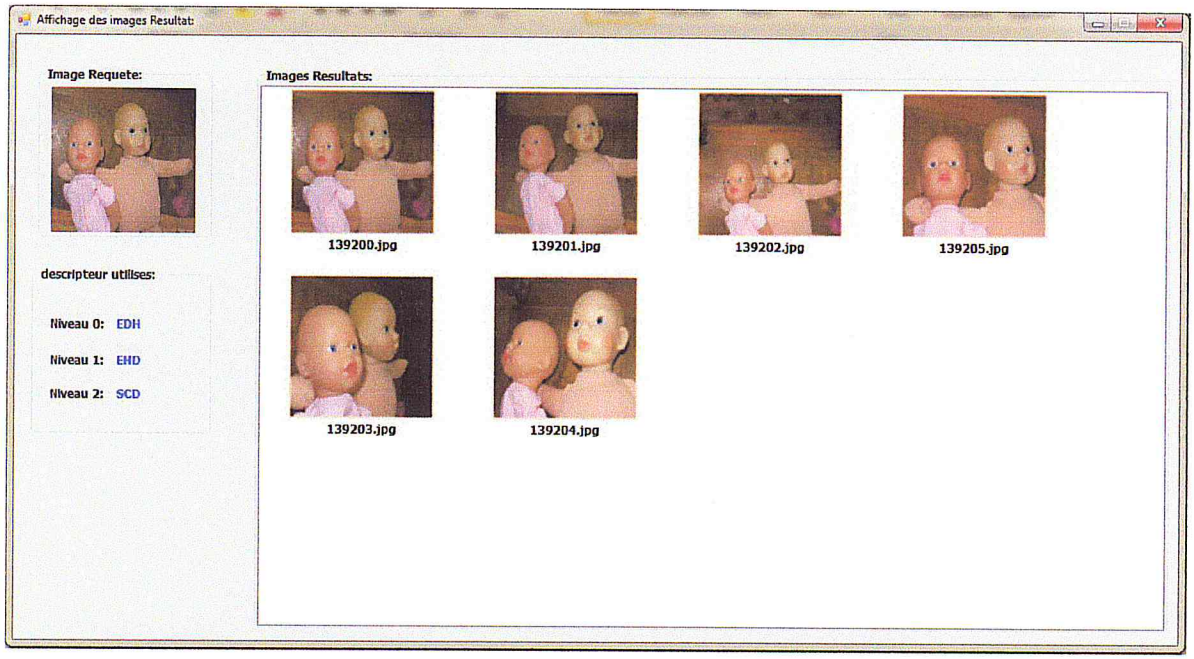


Fig.4.18 exemple de recherche avec une précision exact (utilisation de texture et couleur et contours).

Description : précision=100%, rappel=5.26%.

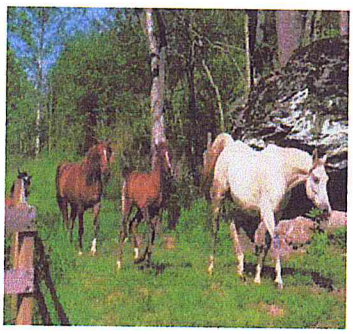
7. Perspective

Nous avons présenté une évaluation expérimentale des descripteurs visuels de MPEG 7 et aussi le descripteur de contours. Et nous avons conclu qu'il y a une relation de complémentarité entre les différents descripteurs. Les meilleurs scores obtenus (précision et rappel) sont ceux de combinaison de *texture et couleur*, et aussi la combinaison de *texture, couleur, et contours.*

Afin de valider les résultats que nous avons obtenus, on va utiliser les 27 configurations de choix descripteurs possibles. Et calculer pour chacune la précision et le rappel sur une collection de 105 images complètent hétérogènes. Les résultats obtenus sont cités au-dessous

8. Mesure de la qualité des réponses

À partir des résultats présentés dans la section précédente, nous voulons calculer les tables de rappel et précision. Par exemple, considérons la requête ***712.jpg*** pour laquelle 14 images sont pertinentes dans la base.



Images 712.jpg

Choix des descripteurs (niveau 0/1/2 resp)	Nombre des images pertinents	La précision(%)	Le rappel(%)
contours/contours/contours	13	31.70	12.38
contours/contours/texture	9	90	8.57
contours/contours/couleur	13	72.22	12.38
contours/texture/contours	13	65	12.38
contours/texture/couleur	**11**	**100**	**10.47**
contours/texture/texture	12	80	11.42
contours/couleur/contours	5	100	4.76
contours/couleur/texture	**11**	**100**	**10.47**
contours/couleur/couleur	9	100	8.57

Tableau .4.1. L'évaluation des tests en fixant au niveau 0 les contours.

Choix des descripteurs (niveau 0/1/2 resp)	Nombre des images pertinents	La précision(%)	Le rappel(%)
texture/contours/contours	13	65	12.38
texture /contours/texture	13	51.16	12.38
texture /contours/couleur	4	100	3.80
texture /texture/contours	7	100	6.66
texture /texture/couleur	**13**	**100**	**12.38**
texture /texture/texture	10	100	9.52
texture /couleur/contours	**12**	**100**	**11.42**
texture /couleur/texture	**13**	**100**	**12.38**
texture /couleur/couleur	10	100	9.52

Tableau .4.2. L'évaluation des tests en fixant au niveau 0 la texture.

Choix des descripteurs (niveau 0/1/2 resp)	Nombre des images pertinents	La précision(%)	Le rappel(%)
couleur/contours/contours	4	100	3.80
couleur /contours/texture	**11**	**100**	**10.47**
couleur /contours/couleur	6	100	5.71
couleur /texture/contours	10	100	9.52
couleur /texture/couleur	**12**	**100**	**11.42**
couleur /texture/texture	**11**	**100**	**10.47**
couleur /couleur/contours	13	65	12.38
couleur /couleur/texture	**11**	**100**	**10.74**
couleur /couleur/couleur	**11**	**100**	**10.74**

Tableau .4.3. L'évaluation des tests en fixant au niveau 0 la couleur.

9. Des résultats des tests

D'après les tableaux d'évaluations que nous avons présentées. On remarque 9 combinaisons ont données des résultats performants :

- ✓ Contours/texture/couleur (resp niveau 0/niveau 1/niveau 2).
- ✓ Contours/couleur/texture.
- ✓ Texture/texture/couleur.
- ✓ Texture/couleur/contours.
- ✓ Texture/couleur/texture.
- ✓ Couleur/contours/texture.
- ✓ Couleur/texture/couleur.
- ✓ Couleur/couleur/texture.
- ✓ Couleur/couleurs/couleur.

10. Conclusion

Finalement, les résultats obtenus montrent l'efficacité de notre approche de combinaison, et la pertinence des descripteurs de la norme MPEG 7 dans l'arbre quaternaire. Les meilleurs scores obtenus sont de la combinaison de *contours/texture/couleur*, et *texture/couleur*. Ceci est dû à la richesse de caractérisation visuelle en utilisant l'approche de combinaison dans l'arbre quaternaire (global et local), qui a montré une pertinence excellente tout dépend des caractéristiques d'une telle base de données.

Conclusion générale

Tout au long de notre travail on a implémenter l'indexation et la recherche d'image par le contenu par la structure arbre quaternaire a trois niveaux dont chaque niveau et chaque nœud de l'arbre en sauvegardant un seul descripteur de bas niveau que ce soit de la norme MPEG7 ou autre et les résultats de la combinaison de ces descripteur a été bien montré dans le dernier chapitre ce qui nous permet de tirer comme une conclusion générale que la combinaison des descripteur visuel avec l'arbre quaternaire est efficace pour différents types d'images, avec diverses orientations.

À l'issue des travaux menés dans le cadre de ce mémoire, nous dégageons quelques perspectives qui peuvent être réalisées à court et à moyen terme. En premier lieu, nous envisageons d'enrichir notre système en utilisant d'autres caractéristiques visuelles afin d'étendre la collection des descripteurs utilisables. La seconde amélioration intervient dans la phase de recherche où nous prévoyons de tester le descripteur de texture homogène(HTD) et couleurs-structure de la norme MPEG 7 en sous un seul vecteur. De plus, un système de bouclage de pertinence est une solution intéressante pour réduire le fossé sémantique. Nous pensons également d'étudier et de tester les différentes mesures de similarité afin de les comparer et de pouvoir sélectionner celles qui ont la meilleure performance.

Bibliographie

[ABE 07] Système D'Indexation et de Recherche d'Images par le contenu Houaria ABED, Lynda ZAOUI Laboratoire : Systèmes, Signaux, Données Université des Sciences et de la Technologie d'Oran - Mohamed Boudiaf 2007.

[AHM 97] AHMAD I., GROSKY W., « Spatial Similarity-Based Retrievals and Image Indexing By Hierarchical Decomposition », Int. Database Engineering and Applications Symposium (IDEAS), Montreal (Canada), 1997.

[BAJ 73] D. Mercier, R. Séguier, Utilisation des STANN en audio : illustration en reconnaissance de chiffre, Journée Valgo 2001.

[BED 10] Recherche d'images par le contenu, BEDOUHENE Saïda, MEMOIRE DE MAGISTER, ingénieur U.M.M.T.O 2010.

[BURG 09] Principles of Digital Image Processing Fundamental Techniques Wilhelm Burger Mark J. Burge 2009.

[CAS 02] CASTELLI V., « Image Databases : Search and Retrieval of Digital Imagery », cha-pitre 14 - Multidimensional Indexing Structures for Content-Based Retrieval, p. 373–433, Wiley Inter-Science, 2002, V. Castelli and L.D. Bergman (Eds) - ISBN : 0-471-32116-8.

[CHE 00] J-Y Chen, C. A. Bouman, John C. Dalton. Hierarchical Browsing and Search of Large Image Databases. IEEETIP : IEEE Transactions on Image Processing,
9(3): 442-455, 2000.

[JAC 95] C. Jacobs, A. Finkelstein, D. Salesin. Fast multiresolution image querying. Proceedings of SIGGRAPH95, Los Angeles, California, 1995.

[DEL 99] M. de Andrade, G. Bertrand, et A. Araújo. An attribute-based image segmentation method. Materials Research, 2(3):145–151, 1999.

[DON] Efficient use of local Edge descriptor Dong Kwon Park,Yoon seok Jeon,Chee Sun won, IEEE .

[GEN 02] Indexation multi-niveau pour la recherche globale et partielle d'images par le contenu, Geneviève Jomier, Maude Manouvrier, Université Paris Dauphine – LAMSADE 2002.

[GUA 12] Content-based image retrieval using color difference histogram Guang-Hai Liu, Jing-Yu Yang, Pattern Recognition 46 (2013) 188–198, journal homepage: www.elsevier.com/locate/pr 2013.

[GUTT 84] Automn Guttman, « R-tree : A dynamic index structure for spatial searching », In Proceedings of the ACM SIGMOD International Conference on Management of Data, pages 47-57, Boston, MA, June 1984.

[HAD 12] Indexation d'Images de Cellules Sanguines par Arbres Quaternaires Généralisés Karim HADDADA, Mohamed Ali MAHJOUB, SETIT 2012.

[HAR 73] Haralik, R.M., Shanmugam, K., et Dinstein, I. Textural features for images Classification. IEEE Transaction on System , Man, Cybernetics, 3,610-621, 1973.

[HAFNER 95] J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, W. Niblack, "Efficient ColorHistogram Indexing for Quadratic Form Distance Functions" IEEE Trans. on PAMI,Vol. 17, No. 7, pp. 729-736, Juillet 1995.

[HEN 89] Andreas Henrich, Hans-Werner Six, Peter Widmayer, "The LSD tree: spatial access to multidimensional point and non-point objects", In Proceedings of the 15th International Conference on Very large data bases, pages 45-53, Amsterdam, The Netherlands, July 1989.

[HUA 97] G.Pass, R.Zabih and J.Miller. ''Comparing images using color coherence vectors''. Proc. ACM Conf. On Multimedia, pp.65-73, Boston, USA, 1996.

[HU 62] M.K. Hu. Visual Pattern Recognition by moment invariants. IRE Transaction on Information Theory, Volume 8, n°2 :179–187, 1962.

[KATA 97] Noriko Katayama, Shin'ichi Satoh, "The SR-tree : An index structure for High Dimensional Nearest Neighbor Queries", In Proceedings of the ACM SIGMOD International Conference on Management of Data, pages 369-380, Tucson, Arizon USA,1997.

[LAB 06] Laboratoire L3I- Université de La Rochelle, Segmentation d'Images de Documents Anciens par Approche Texture, 2006.

[LIN 04] : COURS DE TRAITEMENT D'IMAGES Lingrand Diane Projet RAINBOW Rapport de recherche ISRN I3S/RR–2004-05–FR 22 Janvier 2004.

[MAN 02] MANOUVRIER M., RUKOZ M., JOMIER G., « Quadtree representations for storage and manipulation of clusters of images », Image and Vision Computing, vol. 20, n°7, 2002, p. 513-527.

[MAU 02] Sto ckage et gestion d'images par un Arbre Quaternaire Générique, Maude Manouvrier, Marta Rukoz Université Paris Dauphine LAMSADE2002

[MAR 02] Marta Rukoz, Maude Manouvrier, Geneviève Jomier, « Distances de similarité d'images basées sur les arbres quaternaires » 18èmes Journées Bases de Données Avancées – 21-25 Oct./2002. BDA'2002, pages 1 à 20.

[MER 01] D. Mercier, R. Séguier, Utilisation des STANN en audio : illustration en Reconnaissance de chiffre, Journée Valgo 2001.

[MEH 95] B.M.Mehtre, M.S.Kankanhalli, A.Nasrasimhalu and G.Man. ''Color matching for image retrieval''. Pattern recognition Letters, Vol.16, pp.325-331, 1995.

[NAJ 08] Thèse de Doctorat de l'université de Nantes , Najlae IDRISSI , La navigation dans les bases d'images: prise en compte des attributs de texture Faculté des sciences de Rabat 2008.

[NAS 97] « Indexation d'Images par le Contenu : un Etat de l'Art », Compression et Représentation des Signaux Audiovisuels (CORESA'97), Issy Les Moulineaux - France (1997) Journées CNET, http ://www-rocq.inria.fr/imedia/.

[OGI] :Jean-Marc OGIER, Partie3:Techniques de Clustering et d'indexation multidimensionnelles », cours Indexation MultiMedia, université de La Rochelle.

[PASS 96] G.Pass, R.Zabih and J.Miller. ''Comparing images usingcolor coherence vectors''. Proc. ACM Conf. On Multimedia,pp.65-73, Boston, USA, 1996.

[PEN 01] Texture Descriptors in MPEG-7 Peng Wu, Yong Man Ro, Chee Sun Won, and Yanglim Choi , W. Skarbek (Ed.): CAIP 2001, LNCS 2124, pp. 21–28, 2001. Springer-Verlag Berlin Heidelberg 2001.

[REY 12] le traitement d'images avec Silverlight 5, Patrice rey 2012.

[ROB 81] John T. Robinson. "The k-d-b-tree: A search structure for large multidimensional dynamic indexes". In Proceedings of the ACM SIGMOD International Conference on Management Data, pages 10–18, 1981.

[SAM 60] SAMET H., « The Quadtree and Related Hierarchical Structures », Computing Surveys, vol. 16, n° 2, 1984, p. 187-260.

[SHA 78] A. Iannino et S. D. Shapiro. A survey of the Hough transform and its extension for curve detection. Proc. IEEE PRIP'78, Chicago, mai juin, p. 32-38. 1978.

[TAM 78] H. Tamura, S. Mori, and T. Yamawaki, Texture features corresponding to Visual perception, IEEE Trans. On Systems, Man, and Cybernetics, vol. Smc-8, No. 6, Juin 1978.

[TUE 92] A.K.Jain and M.Tuceryan, "Texture analysis", chapter 11 in the Handbook of pattern Recognition and computer vision by C.H.Chen 1992.

[VAL 05] Recherche par contenu visuel dans les grandes collections d'images, Valérie Gouet-Brunet, chapitre-gouet 2005, page 2.

[VETTER 95] J. Kovacevic M. Vetterli. Wavelets and Subband Coding. Prentice Hall, 1995.

[VIN 04] détection efficace des contours , GUESDON VINCENT , université de Québec octobre2004 .

[WHI 96] David A.White, Ramesh Jain, "Similarity indexing with the SS-tree", in proceeding of the 12th International Conference on Data Engineering, page 516-523, 1996.

[WOOD 01] Digital Image Processing , Second Edition, Rafael C. Gonzalez University of Tennessee Richard E. Wood MedData Interactive. 2001.

[ZAH 01] Indexation de vidéos et de maillages 3D dans le contexte MPEG-7, Titus Bogdan ZAHARIA, UNIVERSITE RENE DESCARTES - PARIS V, Centre Universitaire des Saints-Pères 2001.

[ZHA 05] D. ZHANG, G. LU, Study and evaluation of different Fourier methods for image retrieval, Image Vision Computing, vol. 23, 2005.