Ecole Centrale de Lyon

THESE

Présentée par

Youssef Chahir

Pour obtenir le grade

DOCTEUR DE L'ECOLE CENTRALE DE LYON

Discipline : Informatique

Indexation et Recherche par le contenu d'informations visuelles

Directeur de Thèse : Liming CHEN

JURY

Marie-France BRUANDET, Université Joseph Fourrier de Grenoble (*Rapporteur*)
Liming CHEN, Ecole Centrale de Lyon
Bertrand David, Ecole Centrale de Lyon
Georges GARDARIN, Université de Versailles (*Président*)
Robert LAURINI, Institut National des Sciences Appliquées, Lyon
José MARTINEZ, Université de Nantes (Co-*Rapporteur*)
Noureddine MOUADDIB, Université de Nantes (*Rapporteur*)

Table des Matières

Table des Figures

Liste des Tableaux

I. Introduction

A. Contexte

Une information de plus en plus visuelle est une conséquence majeure de la convergence entre l'informatique et l'audiovisuel [IEEE96]. De plus en plus d'applications produisent, utilisent et diffusent des données visuelles incluant des images fixes et animées. Si dans un passé encore récent, les différentes sources d'informations visuelles étaient encore produites et exploitées souvent d'une façon indépendante et isolée par des domaines d'application bien spécifiques, confiés à des rares spécialistes, comme par exemple les Systèmes d'Informations Géographiques (SIG), les Systèmes d'Imagerie Médicale, ou encore des applications de surveillance militaire s'appuyant sur des photos satellites, la conjugaison des avances technologiques, comme par exemple ATM pour la communication à très haut débit, DVD pour les stockages de données à très grande capacité, PC puissants, Web, etc., démocratisent leurs utilisations dans tous les domaines, y compris le secteur de loisir du grand public. Le potentiel gigantesque de la retombée financière fait de cette industrie multimédia un enjeu majeur de l'économie de cette fin du millénaire.

Ainsi, la télévision devient numérique, les agences d'images s'informatisent, les satellites nous inondent de leurs images. L'INA (Institut National de l'Audiovisuel) archive chaque année de l'ordre d'une dizaine de téra-octes de données numériques de type vidéo [Jol98]. Sur l'internet, le moteur de recherche Lycos a recensé plus de 18 millions d'images alors que l'agence de photos Sygma en possède 700 000. Par ailleurs, le marché des appareils photo numériques est en train d'exploser : 85 000 unités vendues en France en 1998, 160 000 prévues pour 1999, les ventes d'appareils photo numériques doublent tous les ans. Selon les chiffres fournis par Olympus, environ 1,2 millions d'appareils seront vendus en Europe en 1999. Le marché mondial devrait croître de 36% cette année pour atteindre les 4,3 millions vendus contre 200 000 en 1995.

B. Problématique et objectifs

Cette profusion d'images, réclame un système intégré de gestion d'informations visuelles [FM95] offrant des outils puissants, tels ceux qu'offraient les SGBDs classiques pour les données alphanumériques, afin de permettre aux utilisateurs de manipuler, de stocker et de rechercher par leur contenu les informations visuelles. En effet, le simple archivage n'est pas suffisant, il faut aussi garantir un accès rapide et efficace aux images stockées. Face à une masse toujours croissante d'images numériques, fixes ou animées, il est donc nécessaire d'associer à celles-ci des index performants. Malheureusement les images fixes et animées, à la différence des données textuelles classiques telles que nous avons l'habitude de manipuler, se présentent le plus souvent sous une forme brute, tout simplement comme un train de bits n'ayant a priori aucune sémantique, même si l'on connaît le format dans lequel elle est codée. D'un autre côté, l'indexation manuelle, l'approche généralement suivie pour les bases d'images fixes ou animées, est d'une part fastidieuse - donc impraticable dans une perspective de bases de millions d'images ou de centaines de milliers d'heures de vidéos accessibles en réseau -, et d'autre part subjective quant à la description du contenu des images. Il est impératif d'avoir

des techniques permettant une indexation automatique ou semi-automatique pour faciliter la recherche d'une information pertinente dans une masse d'images ou de vidéos [Nas97].

Dans nos travaux de thèse, nous nous sommes principalement intéressés à ce problème d'indexation par le contenu d'images fixes. A la différence d'une annotation manuelle, nous suivons l'approche, apparue au début des années 90, qui consiste à indexer les images par leur propre contenu visuel [GR95, Jai97, Nar95, PP96, SC96]. L'objectif ici est de mettre au point des techniques permettant d'extraire pour chaque image une signature qui résume au mieux le contenu visuel de celle-ci à partir des attributs tels que la couleur, la texture et les formes. Ces signatures sont rassemblées et structurées en des index. La recherche se fait alors par la similarité selon une distance appropriée [Chen98].

Le principe d'une telle recherche par la similarité est le suivant. On invite d'abord l'utilisateur à fournir une image, appelée image de requête, en la dessinant ou la sélectionnant, le moteur de recherche cherche ensuite dans la base d'images celles qui ressemblent à l'image de requête sur la base d'une comparaison de signatures visuelles.

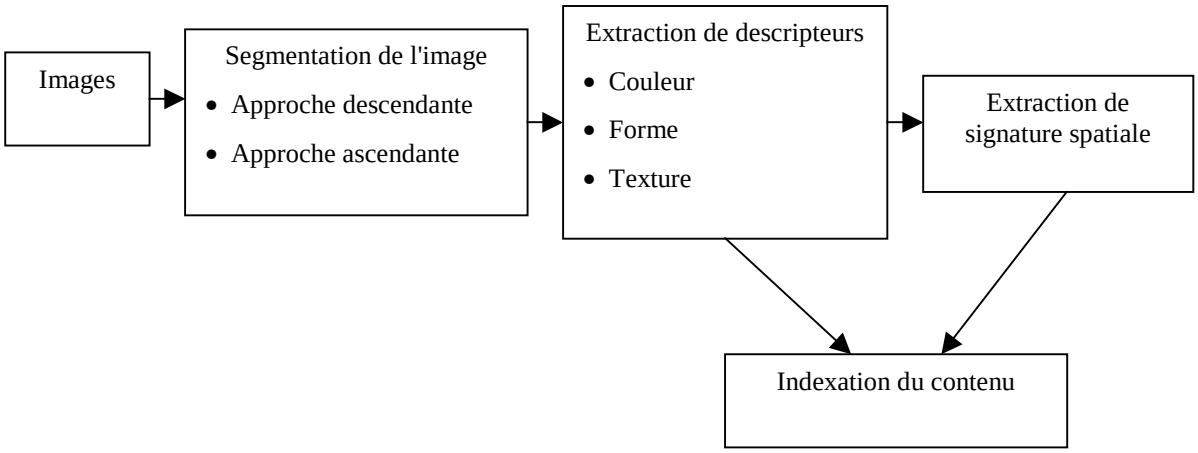
Les applications de systèmes de recherche d'images par le contenu sont nombreuses. On pourrait citer parmi d'autres [Nas97]:

- *Les applications scientifiques*, par exemple pour retrouver des invariants sur des images satellites.
- *La médecine*, par exemple pour retrouver des images pathologiques dans le cadre de l'imagerie médicale.
- *Les applications grand public*, par exemple, le web et le commerce électronique.
- *La surveillance et l'authentification*, par exemple pour la détection de visages ou des contrefaçons.
- *L'art et l'éducation*, par exemple dans une recherche encyclopédique d'illustrations.
- Les télécommunications, par exemple pour le codage des images par leur contenu, cf. MPEG-4 et MPEG-7 [MPEG7].
- *La publicité et le design*, par exemple pour illustrer une publicité par une image adéquate, rechercher une texture spécifique.
- *L'audiovisuel*, par exemple pour rechercher un plan spécifique d'un film.
- *L'archivage*: lieu, événement, personne, logo, plan technique.

C. Notre démarche

La majorité des travaux actuellement proposés dans la littérature utilisent des signatures globales pour effectuer une recherche par le contenu d'image [GP97, Fli+95,OS95]. La simplicité d'une telle approche ne doit pas cacher ses défauts évidents quant à la précision dans les recherches. En effet, dans le cas où l'attribut visuellement perceptible serait basé sur l'histogramme de couleurs par exemple, une image ayant une même distribution de couleurs mais très différente de l'image de requête peut être évaluée comme une réponse pertinente. Quelques travaux tentent de remédier à un tel défaut en proposant d'associer des informations spatiales aux différentes caractéristiques visuelles. G. Pass et al [PZM96] proposent de considérer des vecteurs de cohérence de couleur pour prendre en compte la cohérence des couleurs spatialement voisines. Le projet VisualSEEk [SC96b] fait un pas en avant et propose d'extraire des régions et caractéristiques spatialement localisées. Dans nos travaux, nous

proposons de réaliser d'abord une compréhension visuelle profonde d'une image en segmentant les objets visuellement significatifs selon des critères de couleurs, textures ou de formes par des méthodes finalement peu coûteuses. Des descripteurs et relations spatiales sur ces objets visuellement homogènes sont ensuite extraits pour aboutir à un résumé visuel profond de l'image. Le schéma suivant illustre une telle approche dans le processus de l'indexation.



En ce qui concerne le mécanisme de la recherche d'images, c'est à dire le moteur de recherche, nous proposons d'utiliser les structures de R-tree et les clés de Peano pour organiser et indexer les descripteurs d'objet et les relations spatiales extraits d'images. On ramène ainsi un problème de recherche par le contenu visuel d'images à celui qui est classique, le problème de la recherche dans des données alphanumériques. Ce qui nous a permis de définir un algorithme simple de votes comme mécanisme de recherche.

D. Nos contributions

La majeure partie de nos travaux concerne l'indexation et la recherche par le contenu d'images fixes ; il s'agit alors de concevoir des méthodes de segmentation et de mettre en place des techniques d'indexation en utilisant les structures de données déjà existantes qui font le plus souvent partie de l'état de l'art. Par la suite, ces même techniques seront aussi exploitées pour la problématique de recherche par le contenu d'images animées.

Notre principale contribution est l'élaboration de méthodes de segmentation souples qui répondent à nos exigences en matière de cohérence et d'homogénéité des objets, qui permettent d'extraire une signature spatiale du contenu de l'image. Une telle signature permet de représenter quantitativement les relations spatiales entre les objets d'une image et peut être considérée comme un terme composé (descripteur, mot clé). A la différence des méthodes classiques de recherche basées uniquement sur les indices globaux, nous nous proposons d'intégrer les informations locales et spatiales du contenu de l'image. A l'instar des systèmes récents de recherche par le contenu [EQVW], en combinant des indices visuels tels que la couleur, la texture, la forme et la disposition spatiale d'objets, l'indexation de nos signatures visuelles spatialisées permet de répondre à une variété de requêtes (superficie, position, disposition, etc.…).

1. Sur les images fixes

Notre travail concernant les images fixes s'articule autour de trois axes : la segmentation, l'extraction d'un résumé visuel et spatial, appelée parfois aussi signature visuelle et spatiale, par des descripteurs d'objet et leurs relations spatiales dans l'image, l'indexation du résumé visuel et le mécanisme de recherche. La première phase de segmentation consiste à écrire des algorithmes de segmentation d'une image et d'extraction de régions d'intérêt homogènes. L'extraction de signature permet de déduire un descripteur qui sera considéré comme un terme dans les bases de données traditionnelles. La phase d'indexation s'appuie sur des techniques de classification utilisées en analyse de données et des structures de données couramment utilisées dans les bases de données, telles que le hachage, fichiers inversés, et des structures d'arbres type R-trees. Les résultats de recherche sont ensuite évalués en terme de précision et de rappel.

Dans un premier temps, nous avons proposé une méthode de segmentation, descendante, basée sur les quadtrees, en tenant compte de critères d'homogénéité de la couleur, et une indexation multidimensionnelle basée sur les clés de Peano pour le codage et l'organisation. Ce travail a fait l'objet d'une communication dans une conférence internationale SPIE 97[CC97]. Pour des raisons de performance, nous avons tenté d'améliorer nos résultats en tenant compte de la position spatiale des couleurs et en intégrant d'autres indices globaux tels que la texture et la forme. Cette amélioration a fait l'objet d'une présentation à NTIF'98 [Che+98] et d'une communication dans la conférence internationale avec comité de lecture CATA'98 [CC98]. Ensuite, ce même travail, complété par une description détaillée des caractéristiques extraites des objets visuels, a fait l'objet d'une publication dans la revue IJCTA [CC99a]. L'avantage d'une telle approche basée sur les quadtrees, tient d'abord à sa généralité et à la possibilité de représenter des vues multiples d'une même image : une vue physique, spatiale symbolique et structurelle, comme cela est souhaitable pour une implémentation efficace d'une base de données d'images.

L'inconvénient de la méthode de segmentation par quadtree réside dans son processus de découpage systématiquement en quatre quadrants carrés. En plus la méthode est assez coûteuse lorsqu'il s'agit de reconstituer un objet cohérent dans l'image. Aussi, avons nous proposé une autre méthode de segmentation rapide, *ascendante*, qu'on peut appliquer localement ou entièrement à l'image, et qui tient compte des critères de voisinage et d'homogénéité de la couleur directement lors du processus de segmentation. Cette construction a fait l'objet d'une communication à IEEE ICMCS'99 [CC99b]. Basé sur une telle segmentation *ascendante* d'objets, nous nous sommes intéressés aux objets dominants d'une image, et avons proposé une signature spatiale, basée sur une variante des 2D-Strings, qui décrit l'arrangement spatial de ce contenu représentatif. A partir de cette signature, nous avons construit un graphe qui reflète toutes les relations spatiales entre les objets de l'image que nous avons mémorisé dans des structures traditionnelles telles que les fichiers inversés. Ainsi, on peut affiner nos requêtes et répondre à des requêtes basées sur une combinaison de critères. Cette approche fait l'objet d'une publication dans la revue internationale JVCIR [CC99c].

Nos travaux sur les images fixes ont fait l'objet d'un travail supplémentaire par des élèves d'ingénieur qui, en s'appuyant sur les techniques et méthodes de segmentation que nous avons développées, ont implémenté une interface conviviale en Java en respectant une architecture client/serveur.

2. Sur les images animées

En ce qui concerne l'indexation par le contenu d'images animées, nous avons d'abord traité le problème désormais classique de la segmentation d'une vidéo en plans et en séquences (plans de coupe, plans alternés)[IP 97]. Nous avons amélioré la segmentation en plans à partir des histogrammes locaux en travaillant dans l'espace de couleur L*u*v* avec une distance appropriée. Ensuite, nous avons proposé une technique simple qui permet d'extraire d'un plan une image qui résume au mieux le contenu visuel de celui-ci, appelée image clé ou image représentative. Enfin, nous avons procédé à la formation de clusters, en se basant sur les images représentatives ainsi que sur les bornes de chaque plan, pour détecter les plans de coupe ou les plans alternatifs. Ce travail général a fait l'objet d'une publication dans une conférence internationale SPIE [CC99d].

L'ensemble de nos travaux s'est inscrit dans le cadre du projet national **TransDoc**1 [Che+95] labellisé d'expérimentation d'intérêt publique et soutenu par le Ministère de l'Industrie pour le programme national d'autoroutes de l'information dont l'objectif est de mettre en place des techniques et outils pour faciliter les accès aux documents multimédias dans le contexte de Systèmes d'Information Globaux comme celui de WWW.

E. Organisation de la thèse

Le présent manuscrit présente nos travaux sur l'indexation d'informations visuelles par le contenu. Le chapitre II est un état de l'art sur les travaux concernant la recherche et indexation d'informations documentaires textuelles et d'images fixes. Le chapitre contient également un panorama sur quelques prototypes de recherche et produits commerciaux pour la recherche d'images par le contenu. Nous introduisons au chapitre III les définitions sur les indices visuels, couleurs, textures et formes, et leurs descripteurs associés, qui sont à la base de tout système de recherche d'images par le contenu. Nous y analysons aussi les différentes structures de représentation des descripteurs ainsi que les distances de similarité proposées dans la littérature. Ces structures constituent les points de départ sur lesquels s'appuient nos travaux proprement dits qui sont abordées aux chapitres suivants. Dans le quatrième chapitre, nous détaillons les deux approches de segmentation, une approche descendante basée sur les quadtree et une approche ascendante basée sur la croissance des régions, que nous avons proposées en vue d'extraire des objets visuellement homogènes. Nous discutons de nos algorithmes et des paramètres utilisés dans ce cadre. Nous terminons ce chapitre par une description des caractéristiques visuelles d'une région homogène. Le chapitre V est consacré au moteur de recherche des images. Nous y introduisons l'architecture utilisée, le prototype en JAVA et notre approche et algorithmes d'indexation des images à partir de descripteurs d'objets et de leurs signatures spatiales. Les résultats d'expérimentation y sont également présentés et analysés. Le chapitre VI traite nos travaux sur la vidéo. Nous y définissons les problématiques, faisons un tour rapide des travaux du domaine pour aboutir à une présentation de nos contributions sur la segmentation de la vidéo en plans, l'extraction d'images représentatives et la classification de plans en clusters. Le dernier chapitre contient les conclusions avec quelques réflexions sur les ouvertures possibles et sur nos travaux futurs.

1 . Transdoc: projet Autoroutes de l'information labellisé par le Ministère de l'Industrie, convention N.196.2.93.0385

II. Etat de l'art

Les images constituent un prolongement de données alphanumériques, donc une nouvelle frontière d'informations numérisées. Un système de recherche d'images (SRI) par le contenu nécessite un effort de recherche interdisciplinaire et une intégration de résultats des différents domaines concernés, par exemple la recherche d'information, la vision par ordinateur, les bases de données, etc. [BS95, CCH92, SB88 , SM83, Sha95 ,All95]. Bien que la nature d'une image est autrement plus riche et complexe par rapport à un texte, on s'aperçoit finalement qu'en résumant une image par ses aspects visuels comme cela se fait le plus souvent dans la littérature, on tente de ramener le problème de recherche d'images par le contenu à celui de données alphanumériques pour lequel on peut appliquer les techniques élaborées pour les textes [Meh+97a, Ort+97], par exemple l'utilisation du bouclage de la pertinence dans la recherche d'images[RHM97a,Rui+98]. Aussi, commence-t-on ce chapitre d'état de l'art sur les SRI par un rappel très rapide sur quelques principes et techniques fondamentaux sur la recherche d'information documentaire. Ensuite, nous nous concentrons sur les travaux à proprement parler sur la recherche d'images par le contenu. Nous terminons notre chapitre sur un panorama de quelques prototypes de recherche et systèmes commerciaux actuellement disponibles sur le marché.

Pour des raisons de taille et de concision, nous avons volontairement séparé ce chapitre d'état de l'art de celui qui suit sur les indices visuels, où nous tâcherons de donner les définitions précises des caractéristiques visuelles telles qu'elles sont utilisées dans la plupart des travaux sur l'indexation d'images par le contenu : couleur, texture et forme. Le chapitre sur les indices visuels prépare donc les autres chapitres consacrés à nos travaux.

A. Recherche d'information documentaire

1. Problématique

Un système de recherche d'information est un système stockant des éléments d'information que l'on a besoin de traiter, de rechercher et de diffuser vers des utilisateurs potentiels. L'information traitée peut être classée selon le média (image fixe, image animée, texte, ou son) ou selon le degré de sa structuration. La structuration peut se faire par rapport au contenu du document ou par rapport à des références plus ou moins externes. Un autre critère important de classification concerne le type d'accès à l'information: accès par requête portant sur la sémantique de l'information, accès par la structure ou accès par navigation.

Il existe des relations étroites entre la recherche d'information et les bases de données [Ull82] ne serait-ce que par la fonction de recherche qui est au cœur des deux problématiques. Chaque domaine possède ses modèles et méthodes propres bien que l'on constate une tendance à l'intégration des fonctionnalités d'un SGBD avec celles d'un système de recherche d'information [Chr94] .

Tout système de recherche d'information comporte deux fonctions principales: l'indexation et l'interrogation. L'indexation est l'étape fondamentale pendant laquelle un document se voit conférer un statut conceptuel dans la base gérée par le système de recherche. A partir de ce

moment là, un document devient candidat aux demandes potentielles. L'indexation d'un document est souvent un processus de détermination d'une liste de mots destinés à représenter le document lors de la recherche. Deux objectifs contradictoires sont assignés à l'activité d'indexation: résumer au mieux la teneur d'un document et permettre un maximum d'accès ultérieurs à ce document [Sma98] .

L'interrogation est la fonction principale dans un système général de recherche d'information. Elle offre à l'utilisateur les moyens d'exprimer son besoin selon une syntaxe plus ou moins calquée sur un modèle de requête. Celle-ci contient des critères décrivant les caractéristiques souhaitées des documents souhaités. Intervient alors l'étape cruciale qu'est la mise en correspondance entre la requête d'une part et les documents indexés d'autre part. Le résultat de la mise en correspondance est un ensemble de documents jugés pertinents.

Les différentes stratégies d'indexation sont évaluées par la performance obtenue en *rappel* et en *précision* selon la tradition des expériences de cette nature. Le rappel mesure la proportion des documents pertinents extraits par rapport au total des documents pertinents dans le système et la précision mesure la proportion des documents pertinents par rapport au total des documents extraits. Pour utiliser ces deux mesures, il faut pouvoir déterminer les documents pertinents pour chaque requête. Il faut alors disposer de jugements de pertinence pour chaque requête la liste de documents pertinents. Ces jugements de pertinence sont généralement donnés par des personnes connaissant la collection de documents.

Puisqu'il y a deux mesures de performance (rappel et précision), et dans le but de comparer plusieurs stratégies d'indexation, le rappel est fixé à des valeurs déterminées (10%, 20%, …) et la précision est calculée pour chacun de ces niveaux de rappel. Après ce calcul, une stratégie A est dite meilleure qu'une stratégie B si, pour chaque niveau de rappel, la précision de la stratégie A est supérieure à celle de la stratégie B. Si ceci n'est pas vrai pour tous les niveaux de rappel, la moyenne de quelques précisions est calculée, et la comparaison se fait à l'aide de cette moyenne [RJB89]. Enfin, une troisième mesure peut être définie aussi. Elle concerne le *taux de silence* qui donne la proportion de documents non retrouvés pour une question donnée.

2. Modèles et approches

Les méthodes statistiques d'indexation par termes simples (termes formés d'un seul mot) basées sur l'analyse des caractéristiques fréquentielles des mots dans une collection de documents montrent des performances en rappel et en précision comparables aux méthodes manuelle [Sal86]. Plusieurs approches ont été proposées pour pondérer les termes et les avantages respectifs des diverses combinaisons possibles ont été mis en lumière. Une des voies à considérer pour améliorer ces méthodes est la génération de termes composés[2] permettent généralement de limiter l'ambiguïté des termes et d'augmenter la précision.

L'évolution de la recherche d'information de ces 30 dernières années a été dominée par les techniques statistiques, basées pour la plupart sur la représentation des documents par des espaces vectoriels [SYW75, SB87] et les modèles probabilistes [BS75, CM78]. Les besoins en matière d'extraction d'information et en analyse du contenu vont croissant tout comme la quantité de données à manipuler. Cela implique l'élaboration de techniques et la création d'outils simples à mettre en œuvre et à utiliser mais efficaces et congruents en terme de résultat pour le traitement, la recherche et l'exploration de bases de données.

[2] Terme composé dans le sens d'un terme d'index composé de plusieurs mots

Il existe principalement deux types de stratégies pour accéder à l'information qui sera pertinente suite à une requête particulière d'un utilisateur:

- Stratégie basée sur l'utilisation de mots clés fournis par l'auteur du document, ou extraits automatiquement éventuellement en lien avec un thésaurus;
- Stratégie issue de méthodes utilisées en statistique lexicale et en linguistique de corpus, par un accès sémantique au contenu textuel, grâce à la sélection de contextes pertinents et riches sémantiquement.

Méthode de pondération statistique

Parmi les techniques d'indexation à base de calculs statistiques on distingue:

- La méthode de pondération par fréquence inverse. Elle repose sur l'hypothèse qu'il existe une relation inversement proportionnelle entre l'importance d'un terme pour l'indexation d'un document et le nombre total de documents contenant ce terme dans la base documentaire.
- La méthode de pondération par calcul du rapport signal/bruit. Elle part de l'idée de base que l'importance d'un message est une fonction inverse à sa probabilité d'apparition.
- La méthode de pondération par valeur discriminatoire. Elle consiste à définir la valeur d'un terme dans la base documentaire selon la méthode des cosinus de SALTON. Cette méthode calcule la similarité globale (entre deux documents) et la valeur discriminatoire d'un terme ou descripteur

Il existe plusieurs méthodes pour calculer l'importance (ou poids) d'un terme simple dans un document. Une des approches utilisées est celle décrite dans Salton et Buckley [SB88], selon laquelle le poids d'un terme simple est déterminé à partir de trois composantes: sa fréquence dans le document (C1), sa fréquence dans la collection de documents (C2) et un facteur de normalisation (C3). Le tableau 1 présente les méthodes de calcul de ces trois composantes. Il existe donc 18 combinaisons possibles (3 x 3 x 2). Chaque combinaison sera représentée à l'aide de trois lettres: chaque lettre désignant le paramètre utilisé pour chacune des composantes. Ainsi, la méthode de calcul de poids tfx signifie que pour la composante C1 le paramètre t a été utilisé, que pour la composante C2 le paramètre f a été utilisé, et que pour la composante C3 le paramètre x a été utilisé.

C1 : Fréquence du terme		
b	1 ou 0	poids binaire égal à 1 si le terme est présent
t	Ft	fréquence du terme
n	$\frac{1}{2}+\frac{1}{2}\cdot\frac{\text{ft}}{\max\ \text{ft}}$	fréquence normalisée augmentée
C2 : Fréquence dans la collection		
x	1	Aucun changement
f	$\log\frac{N}{fd}$	fréquence dans la collection inverse

p	$\log \frac{N - fd}{fd}$	fréquence dans la collection inverse probabiliste
C3 : Facteur de normalisation		
x	1	Aucune normalisation
c	$\frac{1}{\sqrt{\sum_{termes} p_i^2}}$	normalisation

Tableau 1 - Composantes du poids d'un terme simple

où

- N représente le nombre de documents

- fd est le nombre de documents auxquels le terme est associé

- p_i donne le poids du terme non normalisé = C1 x C2

En ce qui concernent les termes composés, la détermination du poids est une question qui n'a pas encore de réponse précise [Far+96]. Cependant, pour simplifier, ils peuvent être considérés comme un terme simple.

Modèle de repérage

a) Le modèle vectoriel

Dans le modèle vectoriel chaque document i indexé est représenté à l'aide d'un vecteur de la forme $D_i=(d_{i1}\ d_{i2}\ldots d_{in})$ où n est le nombre de termes et d_{ik} représente le poids (ou degré de représentativité) du terme k dans le document i. Les descripteurs non attachés au document ont un poids nul.

Ainsi, une collection *C* de *m* documents peut être par une matrice de descripteurs-documents de taille *m* x *n*, chaque ligne représentant un vecteur document tandis que chaque colonne les apparitions d'un descripteur dans la collection.

$$C = \begin{bmatrix} d_{11}d_{12}...d_{1n} \\ d_{21}d_{22}...d_{2n} \\ . \quad . \quad ... \ . \\ . \quad . \quad ... \ . \\ d_{m1}d_{m2}...d_{mn} \end{bmatrix}$$

Pour effectuer une recherche, l'utilisateur soumet une requête R_j au système qui analyse celle-ci et calcule le poids des termes de la requête. Il représente alors la requête à l'aide du vecteur $R_j= (r_{j1}, r_{j2}, \ldots, r_{jn})$. Les différentes méthodes de pondération décrites à la section précédente peuvent être appliquées non seulement aux documents mais aussi aux requêtes.

Le code *D/bxx* représente le fait que la méthode de pondération bxx a été utilisée pour les documents et *R/bxx* pour les requêtes. Les poids des termes des requêtes sont calculés selon 6 méthodes au lieu des 18 possibles. En effet, dans le cas des requêtes, il n'est pas jugé nécessaire d'utiliser la composante de fréquence dans la collection (f et p) puisque dans la réalité les requêtes sont fournies les unes après les autres au système et non toutes ensembles comme c'est le cas dans les expériences [Far+96].

La mise en correspondance peut alors être réalisée par une mesure de similarité entre les vecteurs d'un espace. Une telle mesure fournit, pour chaque document D_i de la collection, un degré de similarité avec le vecteur requête. Il existe plusieurs formules pour calculer ce degré

de similarité, dont la mesure classique est celle du cosinus de l'angle entre un vecteur document et le vecteur requête :

$$\mathrm{sim}(D_i, R_j) = \frac{\sum_{k=1}^{n} d_{ik} . r_{jk}}{\sqrt{\sum_{k=1}^{n} d_{ik}^2 . \sum_{k=1}^{n} r_{jk}^2}} = \frac{D_i . R_j}{\|D_i\| \times \|R_j\|}$$

D'autres mesures de similarité on été proposées. MacGill et al. en ont dénombré 64 qu'ils ont classé en 24 classes [WWY92]. Après avoir calculé ce coefficient de similarité pour tous les documents, le système trie les documents par ordre croissant par rapport à celui-ci et présente à l'utilisateur une telle liste triée. De cette façon, les premiers documents sont les plus similaires à la requête.

Des structures de fichiers inversés permettent d'améliorer considérablement l'efficacité de la recherche: seuls les vecteurs-documents comportant au moins un terme de la requête sont alors considérés.

b) Le modèle booléen

La recherche booléenne est actuellement utilisée dans la plupart des systèmes de recherche documentaire interactive [Sal89]. Chaque item est représenté par une liste de termes d'index utilisés dans la formulation des requêtes. Les capacités de formulation de requête booléenne sont bien adaptées au contexte où l'utilisateur a une idée précise de ce qu'il cherche et connaît bien le langage d'indexation du système. Ce modèle est basé sur l'algèbre de Boole pour la représentation d'une requête. Trois opérateurs *et*, *ou*, *non* servent à lier les critères de recherche formant une requête. Ainsi, *et* lie deux critères devant être simultanément présents dans un document pour qu'il soit considéré comme pertinent. L'opérateur de négation *non* permet d'exclure l'ensemble des documents indexés par le critère affecté par la négation. Au moment de la recherche de documents, les opérateurs logiques peuvent être interprétés par des opérations sur les ensembles de documents indexés par chacun des critères de recherche (ces ensembles sont appelés listes inverses). L'opérateur *et* (respectivement *ou*) peut s'interpréter comme l'intersection (respectivement union) des ensembles de documents indexés par les deux critères arguments. L'opérateur unaire *non* correspond à la complémentarité de l'ensemble des documents indexés par le critère par rapport à toute la collection de documents [Sma98]. Son inconvénient majeur est son caractère "tout ou rien ": le résultat d'une recherche n'est pas trié et est difficile à contrôler.

c) Le bouclage de pertinence (relevance feedback)

Le principe du bouclage de pertinence consiste à prendre en compte la pertinence jugée par l'utilisateur pour améliorer les performances du système en tenant compte de ses performances passées. Ce mécanisme crée une coopération entre l'utilisateur et le système pour tenter de converger vers la formulation idéale du besoin aux documents désirés. Il est ainsi possible d'améliorer, en quelques étapes, le rappel et la précision de la recherche.

Le scénario de déroulement d'une recherche est alors le suivant: l'utilisateur formule une requête approximative de son besoin permettant au système de recherche d'information de proposer des documents qu'il juge pertinents. Tout en prenant connaissance des propositions du système, l'utilisateur est invité à évaluer la pertinence de chaque document. Une reformulation automatique de la requête initiale intervient alors en tenant compte des retours de l'usager. La requête reformulée sert de point de départ à une nouvelle étape de recherche.

Notons que le mécanisme de bouclage de pertinence est particulièrement puissant lorsque le média des documents est spatial, comme c'est le cas des images, par opposition aux médias séquentiels. L'appréhension d'un document spatial étant quasi instantanée, la phase de visualisation et d'évaluation des propositions est moins contraignante pour l'utilisateur [CH89].

Des travaux ont porté sur la reformulation automatique dans le cadre du modèle booléen [Hal89]. Mais le modèle vectoriel est sans aucun doute celui qui permet la mise en œuvre la plus aisée de la reformulation automatique de requêtes. En effet, cela consiste à augmenter le poids attaché à un terme dans la requête lorsqu'il indexe un document pertinent et à diminuer ce poids lorsqu'il s'agit d'un terme indexant un document non pertinent pour l'utilisateur. Plusieurs méthodes de reformulation ont été proposées pour ce modèle. Une fonction de reformulation d'un vecteur-requête Q_0 par un vecteur-requête Q_1 peut être [SG83]:

$$Q_1 = Q_0 + \frac{1}{n_1} \sum_{D_i \in D_P} D_i - \frac{1}{n_2} \sum_{D_j \in D_{NP}} D_j$$

Où D_P est l'ensemble des vecteurs-documents proposés pour Q_0 et jugés pertinents par l'utilisateur; D_{NP} est l'ensemble des vecteurs-documents proposés pour Q_0 et jugés non pertinents; n_1 est le nombre de documents pertinents (ou $||D_P||$); n_2 est le nombre de documents non pertinents (ou $||D_{NP}||$); $\sum$ représente l'addition de vecteurs.

B. Indexation textuelle d'images fixes

Si le domaine de l'indexation d'images est encore relativement jeune en comparaison avec celui de l'indexation textuelle, il est en pleine effervescence depuis quelque temps. Dès la fin des années 70, de nombreuses institutions voulurent mettre en valeur leurs collections visuelles. Ils mirent donc sur pied des projets de thesaurus et de classification de grande envergure destinés à mieux organiser leurs images. Toutefois, ces projets soulevèrent de nombreux problèmes vis-à-vis l'indexation des indices visuels et moussèrent l'intérêt scientifique dans le domaine des bibliothèques numériques, des sciences de l'information et d'autres domaines encore.

Les premiers systèmes de recherche d'images s'appuient sur les systèmes de recherche documentaire classiques, et font appel, dans leur processus d'indexation et de recherche d'images, à des informations textuelles primaires ou méta données, telles que la source de l'image, la date et le temps où l'image a été prise, la résolution, la méthode de compression, et notamment les annotations textuelles. Les annotations et les mots clés sont des descriptions textuelles libres d'une scène. Si elles sont naturelles pour les utilisateurs et on ramène ainsi un problème de recherche d'images à celui de recherche textuelle, cette approche présente le défaut que deux utilisateurs peuvent décrire la même scène de façons différentes. Ils peuvent utiliser des mots différents, souligner différents aspects de l'image et décrire des détails différents. Un moyen de gérer les différentes descriptions de la même scène est d'étendre la requête et les descriptions de l'image à un thesaurus [SQ96, ATY+97]. Cependant, l'ambiguïté dans le langage naturel et les descriptions courtes peuvent rendre cette tâche de désambiguisation très difficile[Voo93], comme nous allons voir par la suite.

Dans cette section, nous introduisons d'abord deux classes de systèmes de recherche d'informations visuelles basés sur une description textuelle [BA95]: la première utilise un thesaurus visuels tandis que la deuxième s'appuient sur les langages de "*description pictorielle*". Ensuite, nous présentons les défauts majeurs de ces techniques qui ont motivé la recherche d'indexation d'images par le contenu visuel.

Cependant, avant d'aller plus loin, il est important de distinguer les notions, malheureusement souvent confuses, de *catalogage*, *classification* et *indexation* :

- *Le catalogage* consiste à décrire physiquement un document, quel que soit son format, permettant d'une part de l'identifier de façon unique et d'autre part de le repérer par le biais d'une caractéristique qui n'a pas rapport à son contenu (numéro ISBN, nom de l'auteur, etc.).
- *La classification* permet de placer un document, après en avoir analysé le contenu de façon générale, dans l'ensemble des documents qui traitent du même sujet. Le document est ici considéré comme une entité. C'est un peu comme placer le document dans une boîte étiquetée: Animaux, Meubles, Romans français du 19ème siècle.
- Quant à *l'indexation*, on ne considère plus le document comme une entité distincte mais on considère plutôt les éléments d'information qui s'y trouvent. Le but de l'indexation est toujours de créer des regroupements de documents sur un même sujet, mais la description est plus précise.

1. Thesaurus visuels

Les méthodes d'indexation basées sur thesaurus ou système de classification utilisent du texte pour décrire les images. Les thesaurus sont des listes hiérarchiques de termes, en général par ordre alphabétique, avec pour chacun des termes des renvois vers des termes plus génériques, plus spécifiques ou des termes associés. Quant aux systèmes de classification, ce sont des grandes catégories qui permettent de faire des regroupements thématiques. Toutefois, le langage visuel qui est à la base des thesaurus visuels, de la reconnaissance de formes, et de bien d'autres systèmes de classification nécessite de solides connaissances dans le domaine mathématique et informatique.

Une nouvelle vague de recherche a exploré la création de systèmes pour lesquels le texte ne servira plus à indexer ou à repérer les images mais sera remplacé par des éléments visuels. Toutefois, les expériences menées jusqu'ici utilisent toujours des mots. Les systèmes se basent sur le concept du " browsing " qui utilise le processus de sélection le plus performant qui soit, i.e. la combinaison de l'œil et du cerveau. Ces nouveaux systèmes sont efficaces pour de petites collections où les images sont relativement simples. Pour une plus grande collection, il faudrait que l'indexation soit assez exhaustive pour que le taux de rappel ne soit pas trop mauvais.

Il existe ainsi un projet à la NASA au Johnson Space Centre où les images concernant l'espace sont regroupées dans une banque de données qui ne fonctionne pas par la logique booléenne mais plutôt à l'aide d'algorithmes basés sur des statistiques. Le processus de repérage affiche ainsi les données par ordre d'importance par rapport à la requête de l'usager. De plus, le repérage affiche à la fois le terme du thesaurus choisi par l'usager avec les images qui l'accompagnent avec en plus les termes génériques, spécifiques et associés [Sel90].

Le système Phraséa permet d'archiver et rechercher tous types de fichiers: textes, images, vidéo et sons. Il permet un accès global à l'ensemble des collections et est orienté Intranet/Internet. Ce système est utilisé par un très grand nombre d'entreprises gérant de grands stocks d'images, dans le monde entier: les agences de presse (Reuter), les plus grands journaux (Die Welt, Der Spiegel, New York Times), les télévisions (FR3) ..etc.

2. Langages de description pictorielle

Ces systèmes utilisent un langage spécial (algorithmes en général) où la description de l'image peut être codée et lue par une machine. De telles techniques demandent un indexeur très spécialisé et ne sont efficaces que dans le cas de petites collections regroupant des images simples [BA95]. Nous décrivons ci-dessous trois exemples de systèmes de description pictorielle :

On propose dans [Leu+92] une méthode qui se base en fait sur la narratologie de l'image pour la décrire. Le projet Entité-Attribut-Relation utilise des "phrases" avec nom, adjectif et verbe pour décrire ce qui se passe sur l'image. Par exemple, une requête pourrait ressembler à : assis, (chat noir, chaise)

Bordogna et al. se sont penchés sur la description de la silhouette d'une image grâce aux coordonnées de points spécifiques. Le code ainsi créé ressemblerait à ceci : " ARM FROM (9,19) TO (9,23) WITH ENDPOINT IN (3,25) ". Jusqu'ici, le prototype a servi à décrire des images d'ellipses en astronomie [Bor+90].

Des chercheurs taiwanais ont développé des chaînes de caractères qui décrivent la relation spatiale entre les différents objets dans une image [CW92]. En guise d'exemple : T = {(A,B,7), (B,C,8), (C,D,1), (D,E,4)} où 1 = Nord de l'objet de référence, 7 = Est de l'objet de référence, 8 = Nord-Est de l'objet de référence et où A, B, C et D sont des objets.

3. Problèmes liés à l'indexation textuelle d'images

- *Polysémie de l'image*: L'interprétation d'une même image peut être bien différente selon les utilisateurs.
- *Choix des termes*: Selon Michael Krause [Kra88], un des grands problèmes porte sur le choix des termes pour l'indexation des images. En effet, comment trouver sous quelles catégorie sera placée l'image ? La difficulté ne consiste pas à choisir le système d'indexation pour organiser les images mais plutôt de son application, i.e. comment définir les sujets et quels aspects de l'item ou du sujet devraient être mentionnés et les termes même qu'il faudrait choisir. Il explique qu'il existe deux types d'indexation, le "hard indexing" ou le "ofness" d'une image (l'image est de …), i.e. ce que l'indexeur voit dans l'image, soit un chat ou une femme par exemple. Et le " soft indexing " ou le " aboutness " d'une image (l'image est à propos de …) qui porte sur la signification.
- *Constance de l'indexation*: James Turner [Tur94] s'est rendu compte que le terme le plus populaire était le même pour environ 60% des participants. Il en découle que la constance inter-indexeur sera plus forte pour le sujet principal et les aspects objectifs que pour les sujets secondaires ou les aspects subjectifs d'une image. Par ailleurs, Sara Shatford [Sha94] mentionne dans un article plusieurs auteurs ayant étudié la constance de l'indexation.
- *Les besoins des usagers* : Kevin Roddy [Rod91], quant à lui, relève que même si les images sont emmagasinées de façon intelligente, elles restent inaccessibles étant donné qu'elles possèdent trop peu de descripteurs. Et le problème des descripteurs est qu'ils sont ambigus, arbitraires et portent à confusion ou à des désaccords. Il indique qu'il serait important que le système indique des valeurs aux items repérés par degré de rapprochement avec la requête. De plus, le plus grand problème de l'accès aux images est

l'impossibilité de fournir de l'information sur ce qui serait une session typique de recherche d'information. Il faudrait pouvoir anticiper les besoins de l'usager.

- *Transfert de la signification*: Un autre grave problème dans le domaine de l'indexation des images est le transfert de la signification d'un aspect visuel vers du verbal. On croit en effet que ce transfert cause la perte d'une partie de l'information [Mur84]. Par ailleurs, certains croient qu'il est impossible de traduire de l'information visuelle à l'intérieur d'une structure verbale et même que le langage verbal est inadéquat pour exprimer le langage visuel. Elaine Svenonius [Sve94] se penche sur la question de transfert du visuel vers le textuel et Dennis Hogan [BA95] dit même qu'il faut utiliser autre chose que du verbal pour indexer le non-verbal.
- *Normes* : Par ailleurs, Baxter et Anderson [BA95] mentionnent le manque de normes pour l'indexation des images. Déjà au début des années 80, Thomas Ohlgren [Ohl82] faisait état du besoin évident de normes pour la classification et la description d'images.

Pour résoudre les problèmes posés par la recherche basée sur la description, plusieurs méthodes ont été développées, par exemple la restriction à certains types de phrase, utilisation des règles d'inférence, le bouclage de pertinence [HCK90,JG95,ATY+95] et des descriptions structurées. Les descriptions structurées peuvent être des phrases avec restriction en langage naturel, des descriptions symboliques ou iconiques concernant les objets, les attributs et les relations [Li+97, ATY+97, YM98].

Cependant, la plupart des objets et des concepts ne peuvent pas être extraits efficacement par des méthodes automatiques, il faut en effet, prévoir des méthodes semi-automatiques qui prennent en compte les aspects objectifs visuels d'une image.

C. Indexation d'images par le contenu visuel

Nous venons de voir les limites d'une approche basée sur une indexation textuelle d'images. L'approche à l'heure actuelle tente de pallier ces défauts en utilisant ce que l'on peut appeler *résumé visuel* d'une image. Le résumé visuel d'une image s'appuie essentiellement sur l'extraction de caractéristiques de nature statistiques représentant son aspect visuel telles que la couleur, la forme et la texture.

Soulignons cependant que les caractéristiques représentant le contenu visuel d'une image peuvent être générales qui concernent tous les domaines, par exemple la couleur, ou spécifiques à un domaine tel que la reconnaissance des visages ou des empreintes digitales. Les caractéristiques spécifiques à des domaines particuliers ont été largement évoquées en reconnaissance des formes. Généralement, cette catégorie de caractéristiques fait appel parfois à des connaissances a priori du domaine d'application et nécessite beaucoup de calcul pour l'extraction de celles-ci.

Dans cette section, nous nous concentrons sur les travaux basés sur les caractéristiques générales, à savoir la couleur, la texture, la forme et l'arrangement spatial entre les objets d'une image, visant les bases de données d'images sans connaissance a priori du domaine d'application qui est une hypothèse de base dans le domaine d'indexation d'images par le contenu. A la différence des méthodes basées sur une description textuelle, ces travaux tentent de résumer le contenu d'une image en utilisant des calculs statistiques et des analyses mathématiques de déploiements de pixels.

1. Caractéristiques visuelles: Couleur, Texture et Forme

La couleur

La couleur joue un rôle très important dans le processus de recherche d'images. Différents espaces de représentation de la couleur ont été proposées tels que l'espace RGB, l'espace YUV, le système de chromacité et de luminance de CIE, L*u*v* , ainsi que d'autres. L'espace RGB est celui qui est le plus utilisé dans les périphériques d'affichage, d'où une majorité d'images numériques sont de ce format. Toutes les couleurs perceptibles peuvent être reproduites par combinaison des trois couleurs principales qui sont le rouge, le vert et le bleu. Une image de couleur en RGB de 24 bits par pixel donne 2^{24} ou approximativement 16.7 million de couleurs distinctes. Cependant, deux couleurs très différentes au niveau de leurs valeurs dans cet espace RGB peuvent être perçues comme proches par l'être humain. Aussi, d'autres espaces de couleur tels que les espaces de couleur HSV et HSI reflétant mieux la perception humaine de la couleur sont aussi proposés. Des études représentatives de la perception de la couleur et des espaces couleur peuvent être trouvés dans les références [MMD76, Miy88 ,WYA97].

Lorsque l'attribut significatif est la couleur, on utilise généralement l'histogramme de couleurs comme signature d'image. Cet histogramme fournit la distribution des couleurs dans l'image. L'utilisation des histogrammes de couleur dans la recherche d'image a été évoquée dans [SB91, Nib+93, SC96, EM95, PZM96]. Les éléments fondamentaux de cette approche consistent en la sélection d'un espace de couleurs et d'une distance métrique de l'histogramme. Il n'y a pas de consensus sur le choix de l'espace de couleur pour la recherche d'image par les histogrammes de couleur. Le problème vient du fait qu'il n'y a pas d'espace de couleur universel , et du fait de la subjectivité de la perception de la couleur [WYS82].

En pratique, plusieurs espaces de couleur sont utilisés dans le contexte de la recherche d'image. Par exemple, Swain et Ballard [SB91] utilisent un système de couleur avec des axes opposés [BB82] quantifiés en 2048 couleurs dans leur système de recherche d'images en couleur. Ils ont utilisé l'intersection des histogrammes et la distance de Manhattan (L_1) comme mesures de similarité à partir des histogrammes de couleur. Le système QBIC d'IBM [Nib+93] a d'abord quantifié l'espace de couleur RGB en 4096 couleurs (16 niveaux pour chaque composante de couleur). Chaque couleur est ensuite transformée dans l'espace de couleur de Munsell par la transformée de MTM [MY88]. Finalement, le clustering détermine les k meilleures couleurs de Munsell (k = 64). Vellaikal et C.C.J. Kuo ont utilisé l'espace HSV quantifié en 16 entrées pour H, 8 pour S et 8 pour V [VK95]. Gray [Gra95] propose de transformer d'abord les images l'espace RGB en espace CIE-LUV et puis de diviser l'espace de couleur CIE-LUV en 512 couleurs.

Pour calculer la similarité entre deux couleurs proches, Ioka [Iok89] et Niblack et al. [Nib+94] ont utilisé la distance euclidienne (L_2) dans la comparaison des histogrammes. Faloutsos et al. [FBF+94] utilisent une matrice (cross corrélation) dans laquelle chaque entrée a_{ij} donne la similarité entre les couleurs i et j. Cette matrice est utilisée dans la distance quadratique entre deux histogrammes.

Considérant que la plupart des histogrammes de couleur sont sensibles au bruit, Stricker et Orengo proposent d'utiliser des histogrammes cumulatifs de couleur. Leurs résultats de recherche montrent effectivement certains avantages de leur approche par rapport à d'autres approches conventionnelles des histogrammes de couleur [SO95]. Mehtre et al. présentent dans [Meh+95] une étude comparative des histogrammes de couleurs dans différents espaces de couleur, avec différentes métriques.

Cependant, l'utilisation des histogrammes de couleur n'est pas totalement discriminante. En effet, une image ayant la même distribution des couleurs qu'une image de requête peut être évaluée comme une réponse pertinente même si les deux images sont visuellement très différentes. Aussi, dans un but d'améliorer ces critères globaux, une approche simple est de diviser l'image en blocs et d'extraire la couleur pour chacun de ces blocs [Fal+93,CTO97]. Dans ce cas, la comparaison de deux images revient à la comparaison de leurs blocs respectifs. Une autre variante proposée par Lu et al. [LOT94] est la décomposition d'une image en quadtree. Ils calculent alors l'histogramme de couleur pour chaque nœud de l'arbre. Si le principe est simple, la rigidité de la division des blocs rend cependant l'information locale inutile et inadéquate. En effet, la comparaison de deux images ne peut malheureusement pas être ramenée à celle de leurs arbres respectifs à cause de l'inadéquation entre les nœuds des arbres.

Pour localiser l'information dans l'espace, des régions de couleur significatives sont extraites et leurs positions sont enregistrées [HCP95, SC96b]. Les régions sont généralement représentées par des Rectangles Minimum d'Encadrement (RME) et enregistrées dans des structures du type des arbres R. Une recherche sur une couleur spécifique à une position donnée peut alors être exécutée en deux étapes, une sur la couleur et l'autre sur la position. L'intersection des résultats de ces recherches renvoie les images qui satisfont les deux conditions.

J.R. Smith et Shih-Fu Chang proposent d'extraire des régions et caractéristiques spatialement localisées dans leur projet VisualSeek [SC96b]. Ils ont d'abord transformé l'espace de couleur RGB en l'espace HSV qui est un espace perceptiblement uniforme. Puis ils ont quantifié l'espace transformé de couleur en M=166 entrées. Ensuite, ils utilisent une technique de projection en arrière [SC95a, SC95b] qui génère une image par couleur. Pour chaque couleur donnée de l'ensemble, ils extraient toutes les régions de l'image qui lui correspondent.

Soit c un ensemble de couleurs et k un indice de couleur au point [x,y] de l'image d'origine, l'image B de la projection binaire en arrière est générée par: B[x,y]=c[k]. Pour pouvoir calculer les similarités entre les couleurs, une image de projection corrélée en arrière est générée de la manière suivante:

$$B[x,y] = \max_{j=0,\ldots M-1}(A_{j,k} c[j])$$

où $A_{j,k}$ mesure la similarité des couleurs j et k. et M est le nombre de couleurs.

Un ensemble de couleur est défini alors comme une sélection des couleurs à partir de l'espace quantifié de couleur qui limitera l'ensemble des régions à extraire. Cet ensemble peut être obtenu par un simple seuillage sur l'histogramme. Un nouvel histogramme de couleur est généré à partir de cet ensemble. La similarité entre deux ensembles de couleurs est calculée par la distance quadratique entre ces nouveaux histogrammes. Un objet est représenté par un ensemble de couleurs $\subset \{c_1, c_2, \ldots c_n\}$, à qui on associe un vecteur binaire dont chaque entrée est composé de 0 ou de 1 suivant la présence de la couleur dans l'objet. Pour chaque paire de régions R_q et R_t ils calculent les caractéristiques suivantes:

Position absolue de la région: ils calculent la distance entre deux régions par la distance euclidienne entre leurs centres de gravité:

$$d^s_{q,t} = [(x_q - x_t)^2 + (y_q - y_t)^2]^{1/2}$$

Taille: La taille est caractérisée par la distance entre les surfaces et entre les RMEs.

La distance entre deux surfaces est définie par la différence absolue:

$d^a_{q,t} = \left|S_q - S_t\right|$ avec S_x désigne la surface de la région x.

Chaque MBR est définie par sa longueur L et sa largeur l. La distance entre deux RMEs est définie par la distance euclidienne :

$$d^m_{q,t} = [(L_q - L_t)^2 + (l_q - l_t)^2]^{1/2}$$

La distance générale entre deux régions q et t est définie par:

$d_{q,t} = \alpha_c d^{set}_{q,t} + \alpha_s d^s_{q,t} + \alpha_a d^a_{q,t} + \alpha_m d^m_{q,t}$ où α_i est le poids associé à chaque caractéristique

G. Pass, R. Zabih et J. Miller [PZM96] utilisent simplement l'espace de couleur RGB, quantifié en 64 couleurs, dans leur système de recherche d'images. Ils proposent de considérer des vecteurs de cohérence de couleur pour prendre en compte la cohérence des couleurs spatialement voisines. Un pixel est considéré cohérent si la taille de la composante connexe auquel il appartient dépasse un seuil τ, sinon il est considéré comme incohérent. Les couleurs sont quantifiées en 64 en espace RGB. Pour chaque entrée j de l'histogramme h, on calcule le nombre de pixels cohérents α et le nombre de pixels incohérents β. Ainsi h[i] peut être représentée par α_i et β_i. La distance entre deux histogrammes de couleurs h_1 et h_2 est calculée par la métrique L_1.

$$\Delta_H = d(h_1, h_2) = \sum_{k=0}^{n} \left|h_1[k] - h_2[k]\right| = \sum_{k=0}^{n} \left|(\alpha_1[k] + \beta_1[k]) - (\alpha_2[k] + \beta_2[k])\right|$$

Ils proposent d'utiliser une distance basée sur la quantité qui est définie par:

$$\Delta_G = \sum_{k=0}^{n} \left|\alpha_1[k] - \alpha_2[k]\right| + \left|\beta_1[k] - \beta_2[k]\right|$$

En général $\Delta_H \leq \Delta_G$.

Xia.Wan et C.-C.J. Kuo [WK96a] travaillent dans le domaine compressé. Les images traitées sont sous format JPEG [PM93]. Ils commencent par transformer l'espace RGB en espace YCbCr où Y représente la composante de luminance, et Cb et Cr sont les deux composantes de chrominance. Chaque image en YCbCr est segmentée en 8 x 8 blocks et la DCT est appliqué à chaque bloc. Les coefficients de la transformation sont quantifiés, scannées en zig-zag et codés par l'entropie. Le block DCT 8 x 8 est utilisé comme une unité de base pour l'extraction de la couleur. Avec le DCT, le coefficient DC F(0,0) est défini par:

$$F(0,0) = \frac{1}{8} \sum_{i=0}^{7} \sum_{j=0}^{7} f(i, j)$$

Les images sont reconstruites seulement avec les coefficients DCT, ce qui fournit une approximation de la couleur qui sera utilisée lors de la recherche. Les couleurs sont ensuite quantifiées et représentées par un octree [GP88]. Deux attributs indispensables pour chaque nœud de l'arbre sont le nombre de pixels qui passe par ce nœud (nombre de passes) et le centre de gravité des couleurs du nœud. En filtrant par la couleur dominante d'une image de requête, dont le nombre de passes est le plus élevé, plusieurs images non pertinentes peuvent être éliminées. Une image riche en couleurs possède un arbre à plusieurs niveaux et avec plusieurs feuilles. Il lui correspond alors un arbre très profond. Ce critère est aussi utilisé pour éliminer les images qui ne sont pas très colorées. De même, deux images qui sont similaires doivent avoir des arbres de quantifications proches. Plus l'intersection entre deux images est large, plus elles sont proches.

Rickman et Stonham proposent dans [RS96] d'utiliser les histogrammes des tuples de couleur. Ils remplissent d'abord un dictionnaire de code qui décrit toutes les combinaisons possibles des teintes de couleur quantifiées, et construisent un histogramme de teintes rencontrées à l'intérieur des régions d'une image.

Pour s'affranchir des effets de la quantification utilisées dans les histogrammes de couleur, Stricker et Orengo ont proposé l'utilisation des moments de couleur [SO95], dont le fondement mathématique est que la distribution de la couleur peut être caractérisée par ses moments. De plus, puisque l'information majeure est concentrée dans les moments d'ordre bas; seulement le premier moment (la moyenne) et les moments centrés d'ordre 2 et 3 (variance et skewness) sont extraits pour représenter la caractéristique couleur. Une distance euclidienne avec poids est utilisée pour calculer la similarité entre les couleurs. Stricker et Dimai [SD96] ont utilisé les trois premiers moments pour indexer la couleur.

Huang et al. [Hua+97] proposent d'utiliser des mesures de similarités basées sur les correlogrammes de couleurs. Ce sont des structures qui prennent en compte la disposition des couleurs à partir des matrices de cooccurrence des couleurs. Leurs résultats expérimentaux ont montré que cette approche était plus robuste que l'approche conventionnelle des histogrammes en terme d'efficacité de recherche.

X.Wan et C.-C. Jay Kuo [WK97] proposent d'utiliser une autre représentation de la couleur basée sur un octree taillé. Cet arbre est construit à partir d'une classification des couleurs réelles d'une image et non à partir des couleurs quantifiées pour l'ensemble de la base d'image. A partir de cette structure souple, ils déterminent des caractéristiques telles que la largeur de la couleur, sa profondeur , sa moyenne et les distributions pour différentes résolutions.

Les histogrammes de couleur peuvent contenir des centaines de couleur voire des millions. Afin de calculer efficacement les distances des histogrammes, le nombre de dimension doit être réduit. Des méthodes de transformation telles que K-L et la transformée discrète de Fourier (DFT), la transformée discrète du cosinus (DCT) ou diverses transformées d'ondelettes peuvent être utilisées pour réduire le nombre de dimensions significatives. Une autre manière de réduire le nombre de dimensions est de trouver les couleurs significatives, par l'extraction des régions de couleur, et de comparer les images seulement sur la base (par la présence) des couleurs significatives [SC96b]. Des stratégies de subdivision spatiale telle que la technique des pyramides [BBK98] réduisent aussi les espaces des n dimensions en un espace à une dimension et utilisent l'arbre B+ pour la gestion des données transformées. La structure d'accès qui en résulte montre une meilleur performance pour le grand nombre de dimensions comparé aux méthodes telles que des variantes de R-tree.

Récemment Androutsos et al. [APV99] présentent une méthode de recherche des images de couleur qui prend en considération la perception humaine concernant le nombre de couleurs présents dans une image. Pour cela, ils segmentent l'image en différentes régions de couleurs dans l'espace HSV, à partir desquels ils construisent un vecteur de couleurs pertinents. La comparaison entre les images se fait par le biais d'une distance de similarité basée sur l'angle entre les vecteurs et une fonction d'appartenance aux couleurs perçues, au lieu des histogrammes de couleur .

La texture

La texture est la deuxième caractéristique visuelle qu'on peut extraire automatiquement d'une image. Elle correspond aux motifs visuels qui caractérisent un grand nombre d'éléments visibles qui sont disposés d'une manière dense et uniforme. Elle n'est pas le résultat de la

présence seulement d'une couleur ou de l'intensité [SC96c]. De par son importance et son utilisation dans les domaines de la reconnaissance des formes et de la vision par ordinateur, il existe des résultats de recherche très riches qui ont été obtenus depuis ces trente dernières années. Ces résultats trouvent naturellement leur application dans le domaine de la recherche d'images.

Un élément de texture, est la répétition d'une région d'intensité uniforme d'une forme simple. La texture peut être analysée au niveau d'une fenêtre de pixels ou au niveau de l'élément de la texture. La première approche est appelée une *analyse statistique* et la seconde une *analyse structurelle*. Généralement, l'analyse structurelle est utilisée quand les éléments de la texture sont clairement identifiés, tandis que l'analyse statistique est appliquée s'agissant des textures fines (micro-textures) [TT90]. La segmentation de la texture implique la détermination des régions de l'image qui possèdent des textures homogènes. Une fois que les régions sont déterminées leurs rectangles circonscrits peuvent être utilisés dans une structure d'accès telle que R-tree.

Les mesures statistiques peuvent être utilisées pour caractériser la variation de l'intensité dans une fenêtre de texture. Parmi les exemples de mesure, on peut citer le contraste, le caractère grossier "coarseness", et la directivité. Les spectres de Fourier sont aussi utilisées pour caractériser les textures. En obtenant la transformée de Fourier d'une fenêtre de texture, une signature est générée. Les fenêtres qui possèdent des signatures égales ou proches peuvent être combinées pour former des régions de texture.

L'analyse structurelle de la texture extrait des éléments de texture de l'image, détermine leurs formes et estime les règles d'organisation. L'analyse structurelle de la texture consiste à extraire des éléments de texture d'une image, à déterminer leurs formes simples et à en donner une estimation de règles d'organisation. Ces règles décrivent la façon dont les éléments d'une texture sont organisés entre eux dans l'image. Elles s'appuient sur des mesures telles que le nombre de voisins directs (connexité), le nombre d'éléments dans un espace unitaire (densité), et la manière dont ils sont disposés (régularité). En analysant les déformations dans les formes des éléments de texture et leurs règles de placement, on peut obtenir plus d'informations à partir de la scène et les objets qu'elle compose. Par exemple, une croissance de la densité et une décroissance de la taille le long d'une direction peut indiquer une augmentation de la distance dans cette direction.

Au début des années 70, Haralick et al. [HSD73] propose la représentation de la texture par matrices de cooccurrence. Cette approche a exploré la dépendance spatiale au niveau de gris de la texture. D'abord, elle construit une matrice de cooccurrence à partir de l'orientation et la distance entre les pixels de l'image puis elle extrait des statistiques significatives à partir de la matrice pour la représentation de la texture. D'autres chercheurs ont suivi la même démarche et proposent des versions améliorées. Ainsi, Gotlieb et Kreyszig, en étudiant les statistiques proposées dans [HSD73], ont trouvé expérimentalement que le contraste, le moment inverse, et l'entropie ont une grande puissance discriminante [GK90].

Motivés par des résultats issues de la recherche psychologique sur la perception visuelle humaine de la texture, Tamura et al. explorent la représentation de la texture à partir d'un angle différent [TMY78]. Ils développent des approximations de calcul des propriétés visuelles de la texture qui ont été estimées importantes dans les études psychologiques. Six propriétés visuelles d'une texture ont été avancées : "coarseness", le contraste, la directivité, "linelikeness", la régularité et la rugosité "roughness".

La distinction majeure entre la représentation de texture proposée par Tamura et al. et celle basée sur la matrice de cooccurrence est que toutes les propriétés de texture dans la représentation de Tamura et al. sont visuellement significatives, ce qui n'est pas le cas dans la

représentation par matrice de cooccurrence. Cette caractéristique rend la représentation de la texture de Tamura et al. très attractive dans la recherche d'images, puisqu'elle peut fournir une bonne interface pour l'utilisateur. Le système QBIC [EN94] et le système [HMR96, Ort+97] ont utilisé d'ailleurs cette représentation de la texture.

Au début des années 90, après que la transformation par ondelettes a été introduite et ses fondements soient établis, plusieurs chercheurs ont commencé à étudier son utilisation dans la représentation de la texture [SC94, CK93, LF93, Gro+94, KC92, TNP94]. Ainsi, Smith et Chang utilisent dans [SC94. SC96c] la moyenne et la variance qui sont extraites à partir des sous bandes d'ondelettes comme une représentation de la texture. Cette approche donnait jusqu'à 90% de succès sur des images de texture de 112 Brodatz. Afin d'explorer les caractéristiques de la bande moyenne, Chang et Kuo [CK93] proposent une transformée d'ondelettes structurée en arbre pour améliorer la précision de la classification.

La transformée d'ondelettes a été aussi combinée avec d'autres techniques pour avoir de meilleurs performances. Gross et al. appliquent cette technique avec la transformée K-L pour rendre plus performante l'analyse de la texture [Gro+94]. Thyagarajan et al. [TNP94] et Kundu et al. [KC92] combinent la transformée d'ondelettes avec les matrices de cooccurrence pour avoir les avantages des deux techniques.

Des travaux ont été également réalisés pour comparer les différentes techniques d'analyse de texture. Ainsi, Weszka et al. comparent la performance des descripteurs de Fourier, les matrices de cooccurrence, et des statistiques du premier ordre des différences de niveaux de gris [WDR76]. Ils ont testé les trois méthodes sur deux ensembles de terrains et en ont conclu que la méthode de Fourier n'était pas performante alors que les deux autres étaient comparables.

Dans [OD92] Ohanian et Dubes comparent et évaluent quatre types de représentation de texture: la représentation de Markov [CJ83], le filtrage multi-canaux, la représentation basée sur les fractals [Pen84], et la représentation par matrice de cooccurrence. Ces représentations de textures ont été évaluées sur quatre ensembles de test dont deux synthétiques (fractal et Markov), et les deux autres naturels (cuir et surface peinte). Ils ont trouvé que la représentation par la matrice de cooccurrence donne de bonnes performances sur ces ensembles tests.

Dans [MM95a] Ma et Manjunath évaluent l'annotation de la texture d'une image par des différentes représentations de la transformée d'ondelette, comprenant les transformées d'ondelettes orthogonales et bi-orthogonales, la transformée d'ondelettes structurée en arbre et la transformée d'ondelettes de Gabor. Ils en concluent que la transformée d'ondelettes de Gabor est la meilleure parmi toutes les autres candidates [SC96c].

D'autres représentations de texture sont aussi possibles. Dans un but de l'indexation et la recherche d'images, Hang et al. [HCA95] proposent d'utiliser les attributs fractals pour capturer l'auto-similarité des régions de texture. Li et al [LHR97] présentent un ensemble de caractéristiques des textures qui sont basées sur des résidus morphologiques obtenus après les opérations de l'ouverture et de la fermeture, et proposent un algorithme de recherche optimum de cet ensemble.

La forme

Pour de nombreuses applications comme par exemple la CAO ou les applications médicales, l'attribut visuellement significatif peut concerner la forme d'objets. La forme d'une image peut être considérée comme celle formée uniquement de ses contours. Aussi, si on est capable

de segmenter les contours d'une image, le problème de la recherche d'une forme dans une base revient finalement au problème général de la recherche d'une image de couleur [JV95].

Cependant on peut aussi être tenté d'approximer les formes par des formes plus simples. Par exemple les triangles ou les rectangles peuvent être utilisés pour représenter une forme irrégulière: la forme serait alors approximée par une collection de triangles ou de rectangles dont les dimensions et les positions sont enregistrées. L'avantage d'une telle approche est qu'elle n'est pas coûteuse en mémoire ni en comparaison, la forme initiale pouvant être retrouvée assez rapidement avec une certaine marge d'erreur.

Soulignons que la recherche d'images par la forme est l'un des problèmes les plus durs dans le domaine. En effet, une telle recherche suppose qu'on soit capable de segmenter des objets d'intérêt dans les images, par exemple une personne, des voitures ou des bâtiments. En raison de la complexité de calcul, la recherche de formes dans une image est typiquement limitée aux objets significatifs de celle-ci [FBF+94 , PPS94].

Le contour d'un objet n'est pas toujours facile à extraire ou à détecter surtout quand l'image est bruitée. Dans ce cas, on lui fait subir un filtrage. Cette opération de pré-traitement dépend du domaine de l'application. Si l'objet d'intérêt est connu d'avance, par exemple il est plus foncé que le fond, alors un simple seuillage d'intensité peut isoler le bruit. Pour des scènes plus complexes, les transformations invariantes au changement d'échelle, à la translation et à la rotation peuvent être nécessaires.

Une fois l'objet détecté et localisé, sa forme peut être trouvée par un des algorithmes de détection et de suivi du contour [SHB93,Wee96]. Par contre, il est plus difficile de détecter et de caractériser les formes des objets dans une scène complexe où il y a beaucoup d'objets avec des occlusions et des ombres.

Une fois que les bords de l'objet sont déterminés, sa forme peut se caractériser par des mesures telles que la surface, l'excentricité, la circularité, la signature de la forme, les moments de la forme [SHB93], la courbure, la dimension fractale (degré d'auto similarité), etc. Toutes ces caractéristiques sont représentées par des valeurs numériques et peuvent être utilisées comme des clés dans une structure d'index multidimensionnel pour faciliter la recherche.

En général les représentations de la forme d'un objet peuvent être divisées en deux catégories: celles basées sur le contour et celles qui sont basées sur les régions. La première catégorie utilise seulement le contour externe de la forme alors que la seconde s'appuie sur la région entière de la forme [RSH96a]. Les descripteurs de Fourier et les moments invariants sont utilisés dans les deux approches. L'idée de base des descripteurs de Fourier consiste à utiliser les contours transformés par Fourier comme la caractéristique d'une forme. On peut trouver les premiers travaux dans ce domaine dans [ZR72,PF77].

Dans le processus de recherche d'images, on peut exiger parfois que la représentation de la forme soit invariante à la translation, à la rotation, et au changement d'échelle. Ainsi, Rui et al. proposent un descripteur modifié de Fourier qui est invariant aux transformations géométriques, de plus robuste au bruit. Une autre représentation possible, également invariante aux transformations, est celle basée sur les moments d'une région. Hu identifie dans [Hu62] sept moments invariants. A partir de son travail plusieurs versions améliorées ont émergé. Si la proposition dans [YA94] est basée sur la version discrète du théorème de Green, Yang et Albregtsen proposent une méthode rapide en calcul des moments pour les images binaires.

Constatant que la plus part des invariants proposés dans la littérature ont été trouvés par des expérimentations, Kapur et al. développent des algorithmes qui cherchent et génèrent

systématiquement les invariants d'une géométrie donnée [KLS95]. Quant à Gross et Latecki, ils ont développé une approche qui préserve la géométrie différentielle, des bords de l'objet même après la numérisation d'une image.

Dans [CL95, LKC95] un travail sur les courbes algébriques et les invariants est proposé pour représenter les objets complexes dans une scène découpée en plusieurs parties. Pentland et al. [PPS96] proposent d'utiliser la méthode des éléments finis pour décrire un contour. Cette méthode définit une matrice de "stiffness" (fermeture) qui décrit comment chaque point de l'objet est connecté aux autres points. Les vecteurs propres de cette matrice sont appelés des modes et représentent l'espace des caractéristiques. Toutes ces formes sont d'abord arrangées dans cet espace et la similarité est alors calculée à partir des valeurs propres.

Arkin et al. [Ark+91] développent une méthode basée sur la fonction de Turning pour comparer à la fois les polygones convexes et concaves. Dans [CK96], Chuang et Kuo utilisent la transformée d'ondelettes pour décrire la forme de l'objet. Elle possède des propriétés telles que la représentation par multi-résolution, l'invariance, le caractère unique, la stabilité et la localisation spatiale. Barrow et al. [Bar+77] ont proposé les premiers la technique de correspondance de Chamfer qui compare deux collections de fragments de la forme avec un coût proportionnel à la dimension linéaire plutôt que la surface.

Dans [LM95], Li et Ma proposent la représentation des formes par la méthode des moments géométriques (basée sur la région) et les descripteurs de Fourier (basés sur les contours) qui sont suivis par une simple transformation linéaire. On trouve d'autres travaux concernant la représentation de la forme dans [LF93].

La question de performance est également largement étudiée dans la littérature. Dans [BKL97], Babu et al. comparent la performance de la représentation basée sur le contour (la chaîne de caractères représentant le code, les descripteurs de Fourier, et les descripteurs de Fourier UNL), les représentations basées sur la région (les invariants du moment, les moments de Zernike, et les pseudo moments de Zernike), et des représentations combinées (les invariants du moment et les descripteurs de Fourier, les invariants du moment et les descripteurs de Fourier UNL). Leurs expériences montrent que les représentations combinées donnent de meilleures performances que les représentations simples.

Dans [WW80], Wallas et Wintz présentent une technique de normalisation des descripteurs de Fourier qui retiennent toute l'information de la forme, et dont le calcul est efficace. Ils ont aussi pris avantage de la propriété d'interpolation du descripteur de Fourier qui résulte d'une représentation efficace des formes 3D. Taubin [Tau91] propose d'utiliser un ensemble d'invariants algébriques du moment pour représenter à la fois les formes 2D et 3D, qui réduit largement le temps de calcul de traitement de la forme. Schettini [Sch94] propose une reconnaissance de la forme par approximation polygonale des contours.

D'autres chercheurs [Cor+94, HJ94], proposent la représentation des formes des objets dans une image sous forme de chaînes et utilisent les techniques de traitement des chaînes pour la recherche des images. Huttenlocher et al. comparent les images représentées symboliquement en utilisant la distance de Hausdorff [HKR93].

Recherche par le dessin

La recherche par le dessin peut être considérée comme un cas particulier de la recherche par la forme. L'utilisateur peut décrire un objet ou l'image entière par l'arrangement des objets qu'elle comporte. La recherche par les croquis peut être simplifiée par la détection de contour suivi d'un amincissement [Kat+92]. L'opération d'amincissement fournit une image binaire de

contour (noir et blanc) qui réduit considérablement la quantité d'information à enregistrer et à comparer.

Différentes solutions à ce problème ont été proposées. Dans QBIC, une fonction de corrélation est mesurée entre le croquis et les contours de l'image. Les fortes valeurs de corrélation donnent les images qui sont proches de la requête. Cette méthode n'est pas invariante à la variation de la taille de l'image, à la position ainsi que l'orientation du croquis. La forme a été représentée aussi par des caractéristiques globales telles que la superficie, l'excentricité, l'orientation des axes des moments. Cette représentation est invariante par rapport à certains groupes de transformations géométriques. Cependant, elle ne donne pas de garantie sur la notion de proximité perçue.

Afin de modéliser cette notion de similarité comme elle peut être perçue par l'utilisateur, M. Mokhtarian et A.K. Mackworth proposent dans [MM92, Mok95] une méthode basée sur la déformation du croquis de l'utilisateur.

S. Matusiak et M. Daoudi proposent dans [MM98] de modéliser le croquis par les courbures euclidiennes qui paraissent dans le croquis. L'utilisateur fait apparaître les courbures dans sa requête pour caractériser la forme qu'il souhaite rechercher. Les courbures sont invariantes par rapport au groupe de similitude. Mathématiquement la courbure est définie par :

$$k(u) = \frac{x'(u)y''(u) - x''(u)y'(u)}{(x'(u)^2 + y'(u)^2)^{3/2}}$$

Elle permet de définir différents points caractéristiques de la courbe tels que les points d'inflexion, points de forte courbure, etc..). Cette courbure ne peut pas être utilisée comme telle, elle est très sensible au bruit dû au dessin approximatif. Son caractère local pose le problème du choix de l'échelle d'analyse.

M. Matusiak et M. Daoudi, proposent de construire une représentation multi-echelle , basée sur l'emplacement des points d'inflexion pour différents échelles d'écart-type σ. Ils définissent un espace de courbure multi-échelle CSS. [Curvature Scale Space]. Pour l'indexation, ils définissent une distance entre deux CSS par la distance géodésique suivante :

$$d((s_q, \sigma_q), (s_t, \sigma_t)) = \log \frac{\sigma_2 (1 + \sqrt{1 - (\varphi\sigma_1)^2}}{\sigma_1 (1 + \sqrt{1 - (\varphi\sigma_1)^2} - \varphi L}$$

$$\text{où } \varphi = \frac{2L}{\sqrt{(\sigma_1^2 - \sigma_2^2)^2 + L^2(L^2 + 2(\sigma_1^2 - \sigma_2^2)^2)}}$$

et $L = | s_1 - s_2 |$ le module de la différence entre les abscisses curvilignes.

Ils utilisent également le maximum de CSS pour représenter une image dans la base.

2. Indexation multidimensionnelle

La structure d'indexation dans un système de recherche d'images dépend de la quantité d'images à traiter [Ale+95, ZZ95]. Une approche simple pour indexer les caractéristiques visuelles est d'obtenir des valeurs numériques pour les n caractéristiques et puis de représenter l'image ou l'objet comme un point dans un espace de n dimension.

Avant de proposer une technique d'indexation quelconque il est recommandé d'abord de réduire le nombre de dimension [Vet846 ,Vet95]. On étudiera dans le chapitre suivant des méthodes qui permettent de réduire le nombre de dimension d'un vecteur de caractéristiques.

Après la réduction du nombre de dimension des vecteurs des caractéristiques, nous avons besoin de sélectionner des algorithmes appropriés d'indexation multidimensionnelle pour indexer ces vecteurs certes réduits mais de grande dimension.

La recherche dans le domaine de l'indexation multidimensionnelle a évolué grâce à la contribution des trois axes de recherche que sont la géométrie, la gestion de base des données et la reconnaissance de formes. L'histoire des techniques d'indexation multidimensionnelle remonte au milieu des années 70 quand les méthodes basées sur les cellules ont été introduites. Ces algorithmes se basent sur une division régulière de l'espace, par exemple une grille, contenant l'ensemble des données, chaque partition étant assigné à un paquet ("bucket") qui regroupe un ensemble de points voisins.

L'arbre kd et ses variantes [Ben75, Ben90,Spr91,AM93] sont les structures de données les plus utilisées en pratique pour une recherche des plus proches voisins d'un point en mémoire centrale [WJ96a]. Clarkson [Cla94] et Arya et al. [Ary+94], ont proposé un algorithme optimal pour une recherche approximative des plus proches voisins d'un point en respectant la taille de l'ensemble des données.

Des structures multidimensionnelles, telles que les fichiers grilles "grid files" [NHS84,HN83] et les quadtrees linéaires [Sam89, AS91] et d'autres ont été proposées. Malheureusement elles ne sont pas adaptées pour gérer des dimensions très élevées [Fal+93]. On peut néanmoins citer Petrakis et Faloutsos [PF95] qui utilisent la structure en quadtree dans leur système de recherche d'images.

Une utilisation de plus en plus généralisée de systèmes d'information géographiques (SIG) a suscité une forte demande de méthodes d'indexation spatiales. Guttman a été le premier à proposer la structure d'indexation de R-tree en 1984 [Gut84] . Son travail a généré d'autres variantes de R-tree. Sellis et al. proposent l'arbre R+ dans [SRF87]. Greene suggère une autre variante de l'arbre R dans [Gre89]. Beckman et al. définit dans [Bec+90] la meilleure variante dynamique de R-tree : R*tree. Kamel et Faloutsos introduit le Hilbert R-tree [KF94] qui est une méthode d'indexation des rectangles à partir de clés d'Hilbert.

Ces structures précédentes ainsi que celles basées sur les arbres kd, telles que l'arbre kdb [Rob81] et l'arbre hB^{II} [ELS94], peuvent être éventuellement adaptées aux moyennes et grandes dimensions.

Fleurissent encore d'autres propositions dans la littérature, telles que TV-trees [LJF94] et SS-trees[WJ96b], hB-tree[LS90], SR-trees[Kat97]. Mais malheureusement elles ne gèrent pas le problème de chevauchement. Pour y remédier, Bertchold et al. [Ber+96] ont proposé l'arbre X. On peut trouver dans [WJ96a, NS96] une bonne étude de l'état de l'art et une comparaison de ces différentes techniques d'indexation utilisées dans le domaine de la recherche d'images.

On peut d'ailleurs remarquer que, s'agissant d'une recherche dans un espace à très haute dimension, les méthodes de classification et les réseaux de neurones peuvent être utilisés dans la recherche d'images [Rui+97c, ZZ95] comme cela se pratique déjà en reconnaissance des formes. Ainsi, Zhang et Zhong proposent dans [ZZ95] d'utiliser, comme outil, une carte d'auto-organisation, qui est un réseau de neurones, pour construire une structure d'indexation en arbres pour la recherche d'images. Les avantages de l'utilisation de ces cartes sont d'une part les capacités d'apprentissage non supervisé, et d'autre part un clustering dynamique et la capacité de supporter des mesures de similarité quelconques. Leurs résultats d'expérience sur une base de texture Broodiez, montrent que c'est une technique d'indexation prometteuse.

Un effort d'adaptation de techniques d'indexation traditionnelles est toutefois nécessaire dans le domaine de recherche d'images, dans la mesure où celles-ci pour le plus souvent ont été conçues pour répondre à des requêtes sur des points ou sur des champs et non sur des

structures qui devraient être adaptables aux requêtes sur la similarité entre les objets. White et Jain proposent dans [WJ96] des algorithmes d'indexation qui sont d'usage général, donc indépendants du domaine de l'application. Ils ont proposé l'arbre VAM k-d tree et VAMSplitR-tree qui sont des variantes des arbres k-d et de R. Selon leurs expériences, le VAMSplitR-tree fournit la meilleure performance. Yang et al. [YVD95] introduisent l'arbre MB+, une structure linéaire possédant les caractéristiques de l'arbre B+ et permettant l'utilisation des mesures de similarités (distance euclidienne avec poids). Tagare [Tag97] développe une structure adaptable d'un arbre qui s'affine en éliminant les nœuds qui sont inefficaces pour des requêtes de similarité.

Dans [Char+97], Charikar et al. avancent une technique incrémentale de clustering pour une recherche dynamique d'informations. Ils utilisent une structure dynamique capable de manipuler des données de grande dimension et qui fonctionne avec des mesures de similarité non euclidiennes(MSNE). Rui et al. introduisent eux aussi dans [Rui+97c] une méthode de classification rapide et efficace en utilisant des MSNEs.

Pour résumer, trois problèmes doivent être résolus pour qu'une structure d'indexation soit adaptable à la recherche d'images par le contenu:

Choix de la distance : la plupart des méthodes multidimensionnelles travaillent avec l'hypothèse que les différentes dimensions sont indépendantes, et par conséquent la distance euclidienne est applicable ;

Codage de l'information spatiale : si l'information spatiale n'est pas codée, elle est détruite. En d'autres termes les emplacements de ces caractéristiques ne peuvent plus être récupérés à partir de l'index ;

Réduction du nombre de dimensions : plus le nombre de dimension croît plus les structures de l'index deviennent inefficaces.

3. Détection et identification d'objet

La détection d'objet implique la vérification de sa présence dans l'image et sa localisation avec précision pour son identification. On a besoin de transformer les images dans un autre espace pour pouvoir manipuler les changements d'illumination (luminosité), de taille et d'orientation. Les caractéristiques globales et locales jouent un rôle important dans l'identification de l'objet. Pour l'identification d'un objet, un ou plusieurs indices locaux sont extraits et les régions d'intérêts sont représentés en terme de ces indices. Par exemple, un visage peut être modélisé par la taille des yeux, la distance entre l'œil et le nez, etc. L'identification se transforme alors en un problème d'appariement de graphe.

Quant à la reconnaissance d'un objet par indices globaux, l'objet considéré dans sa globalité comme un modèle de l'objet désiré est comparé à toutes les images cibles [JZL96]. Par exemple, pour identifier une personne, une image faciale inconnue (ou sa transformée) est comparée (comme un tout) aux images (ou transformées) de personnes connues. Des études psychophysiques et neurophysiologiques attestent que les indices locaux et globaux jouent un rôle dans l'identification du visage humain [CWS95].

Les méthodes de transformés telles que celles de Fourier, des ondelettes ou de K-L fournissent aussi des caractéristiques qui peuvent être utilisées pour détecter des objets d'intérêt [CSW95, PPS94]. La transformation d'un objet, ayant un spectre unique, par ces méthodes, permet de générer une signature qui peut être employée pour détecter sa présence dans une image.

4. Relations spatiales

Les relations spatiales entre les objets, telles que "à gauche de", "à l'intérieur de", et "au dessus", décrivent les arrangements spatiaux d'objets dans une image, apportent une précision particulièrement intéressante sur le contenu d'une image. La déduction de relations spatiales, telles que l'objet A est *"à gauche de"* l'objet B, l'objet B est *"à gauche de"* l'objet C $\Rightarrow$ l'objet A est *"à gauche de"* l'objet C, doivent être employées pour trouver les images ayant des relations spatiales qui ne sont pas explicites dans la requête de l'utilisateur. Plusieurs méthodes d'indexation et de recherche d'images s'appuient sur ces relations spatiales [Ege93, CSY87 , YM98].

Par exemple, les travaux de Chu et al. pour une application médicale [HCT96] proposent d'abord de détecter des objets caractéristiques tels que des os dans les radios, et des tumeurs dans le cerveau. Les objets caractéristiques sélectionnées et leurs relations spatiales sont par la suite représentés dans un modèle sous forme d'une hiérarchie d'abstraction . Le système SEMCOG [Li+97] développé chez NEC implémente un moteur d'inférence de relations spatiales. Des relations topologiques à l'intérieur d'un rectangle d'encadrement sont explorées dans [Pap+95].

Récemment J.R. Smith et C-S. Li proposent [SL2000] d'abord de segmenter une image en différentes régions étiquetées de couleur, puis de faire un nombre fixe de partitions verticales de l'image. En parcourant l'image de haut en bas de gauche à droite, ils obtiennent une chaîne de caractères qui prend en compte les locations relatives des régions de couleur.

5. La perception humaine du contenu de l'image

Des études sont faites sur la perception du contenu d'une image par l'être humain, et l'intégration d'un modèle humain dans un système de recherche d'images. Les premiers travaux de recherche ont été conduit indépendamment par les équipes MIT [PM96,MP96], NEC[Cox98a,Cox98b,Pap+98] et UIUC [RHM97,Rui+98]. Ce sont ces thèmes qui ont mené à l'étude du bouclage de pertinence dans la recherche d'images.

Dans [Rog+98], Rogowitz et al. ont conduit une série d'expériences sur l'analyse de la perception psychophysique du contenu de l'image. Selon leurs résultats, les indices visuels ne reflètent pas le sens sémantique global des images, mais seulement une partie. Ce résultat les a encouragé à développer des caractéristiques de l'image, basées sur la perception, et des métriques appropriées pour aboutir à des recherches significatives sémantiquement.

D. Les systèmes

Depuis le début des années 90, la recherche d'images par le contenu devient un domaine de recherche très actif; On trouve à l'heure actuelle des prototypes dans la communautés de recherche et même quelques produits sur le marché.

La plupart de ces systèmes de recherche d'images supportent plusieurs options suivantes [CEM98] :

- browsing aléatoire
- recherche par l'exemple
- recherche par le dessin

- recherche par le texte
- navigation

Dans la suite nous allons faire un panorama rapide de quelques systèmes représentatifs et mettre en lumière leurs caractéristiques principales. La plupart de ces systèmes sont originaires des Etats-Unis et totalement dédiés à la recherche d'images. Si les premiers prototypes manifestent des performances voisines et sont limitées en terme de rappel et précision, les travaux récents tels que ceux dans [Sri95, SC97a ,SC97b] montrent des résultats prometteurs, car ils s'appuient à la fois sur les caractéristiques textuelles et visuelles dans le processus de l'indexation automatique d'images. Nous commençons par étudier quelques prototypes issues de la recherche puis quelques systèmes qui sont actuellement disponibles sur le marché.

1. Prototypes issus de la recherche

Photobook

Photobook [PPS94, PPS96] est un prototype développé au MIT Media Lab. C'est un ensemble d'outils interactifs pour la navigation et la recherche d'images selon des critères de texture, de forme et d'image en utilisant les caractéristiques extraites au moyen de techniques de compression d'images (transformée de Wold pour la texture, analyse modale pour la forme et eigenimage pour l'image).

Les eigenimages sont obtenues par transformée de K-L d'images de résolution réduite. La comparaison entre les images se fait à partir de coefficients principaux. L'analyse modale pour la comparaison de la forme repose sur une modélisation physique par éléments finis.

Dans une version modifiée de Photobook, Picard et al. ont proposé d'associer l'utilisateur à la recherche et l'annotation de l'image [PM96,MP96,Pic96a,Pic95,Pic96b,PMS96,Nar95b].

Parmi ses composantes intéressantes on peut citer un module de reconnaissance du visage, recherche d'images par similarité de texture, et des annotations semi-automatiques en fonction d'étiquettes fournies par l'utilisateur[MP96,PMS96].

Cypress et Chabot

Le projet Chabot a été réalisé à UC Berkeley pour le stockage et la recherche dans des collections d'images. Il combine un système de base de données relationnel avec des techniques d'analyse de contenu visuel.

Le système **Cypress** [OS95], réalisé dans le cadre de Chabot, permet aux utilisateurs de définir des concepts en utilisant des caractéristiques visuelles comme la couleur. Par exemple: l'utilisateur peut définir le concept de plage par combinaison des couleurs jaune, beige et bleu.

Le système de recherche Chabot est accessible sur le Web à l'adresse: http://elib.cs.berkeley.edu/cypress.html

Netra

Netra est un prototype de recherche d'images développé dans le cadre du projet ADL (Alexandria Digital Library) à UCSB[MM97]. Il permet aussi la recherche à partir des caractéristiques de couleur, texture, forme et l'information spatiale sur des régions segmentées. Sa particularité réside dans son utilisation des filtres de Gabor pour analyser la

texture[Ale+95, MM95a, MM96a], et la construction d'un thesaurus de l'image basé sur des réseaux de neurones[MM95b,MM96b ,All95].

La démonstration peut s'obtenir en ligne à 'adresse http://vivaldi.ece.ucsb.edu/Netra/.

Mars

MARS (Multimedia Analysis and Retrieval System) a été développé à l'université de l'Illinois à Urbana-Champaign [HMR96,Meh+97b,Meh+97a,Ort+97,RSH96b,RSH96a, Rui+97a , RHM97a, Rui+97b ,Rui+98, RHM97b]. Mars est un système de recherche où il y a un effort interdisciplinaire. Sa contribution principale réside dans son intégration d'un système de gestion de base de donnée, d'un système de recherche d'information [HMR96, Ort+97] et de l'intégration de la recherche et de l'indexation [Rui+97c]. Le système propose une architecture de bouclage de pertinence[Rui+98] et intègre cette technique à différents niveaux du processus de la recherche, comprenant l'affinage du vecteur de requête [RHM97], l'adaptation automatique de caractéristiques [Rui+97b, Rui+98]. La démonstration en ligne est à http://jadzia.ifp.uiuc.edu:8000 .

Surfimage

Surfimage est un logiciel d'indexation et de recherche d'images qui a été développé à l'INRIA depuis 1995. Il est plus sophistiqué que des systèmes commerciaux tels que Qbic d'IBM ou Virage. Il offre des fonctionnalités telles que la combinaison de signatures, la classification automatique d'un lot d'images, les requêtes multiples et l'affinage des requêtes[Nas+98a , Nas+98b]. En plus, ce système permet des requêtes sur des bases spécialisées telles que la reconnaissance de visages.

Une démonstration WWW de Surfimage peut être trouvée à http://www-rocq.inria.fr/cgi-bin/imedia/surfimage.cgi .

Spot It

C'est un système conçu pour une base d'images de visages [BM95]. Les eigenimages sont construites pour chaque région correspondant à des caractéristiques visuelles, par exemple les yeux, le nez, la bouche, etc.. Toutes les images sont représentées par des coefficients principaux. L'utilisateur peut choisir les caractéristiques qui serviront à la comparaison. Le principe du bouclage de la pertinence est pris en compte en permettant à l'utilisateur de reconstruire sa requête en modifiant les coefficients associés aux caractéristiques.

Visual Seek

VisualSEEk [SC96b, SC96c] est un prototype de recherche de caractéristiques visuelles et WebSEEk [SC97c] un système de recherche d'images et de textes orienté World-Wide Web, tous les deux étant développés à l'université de Columbia.

Les caractéristiques visuelles utilisées dans leurs systèmes concernent les couleurs et des caractéristiques de textures basées sur la transformée d'ondelette [SC95a, SC95b , SC94 , SC96c]. Pour accélérer le processus de recherche, ils ont aussi développé un algorithme d'indexation basé sur un arbre binaire[SC96d,30, SC96 ,143]. En plus, VisualSEEk supporte des requêtes basées à la fois sur des caractéristiques visuelles et leurs relations spatiales.

Les démonstrations en ligne sont à http://www.ctr.columbia.edu/~sfchang/demos.html .

Jacob et CVEPS

Les systèmes tels que JACOB [CA96] et CVEPS [CSM96] sont plutôt orientés vidéo et permettent une décomposition automatique de la vidéo à partir d'images clés ou d'objets. Le système JACOB utilise le réseau des neurones pour une détection automatique de plans. Les utilisateurs peuvent utiliser les composantes de la recherche et d'analyse d'image pour classer (indexer) et rechercher les objets de la vidéo ou les images clés à partir de leurs caractéristiques visuelles ou de leur arrangement spatial.

Le système **CVEPS** [CSM96] fournit également ces fonctions mais aussi le parcours de la vidéo dans le domaine compressé.

2. Les systèmes commerciaux

QBIC

Le système semi-automatique Query By Image Content (QBIC) [Nib+94, Fli+95, Fal+93, EN94, SAF94, Lee+94,Dan+93] conçu par Niblack et al. chez IBM considère les couleurs, textures et formes d'une image qu'il transforme en formules algébriques[Nib+93]. Il était le premier système commercial de recherche d'images par le contenu et a beaucoup marqué les systèmes de recherche récents.

La couleur prise en compte dans Qbic est la moyenne des trois composantes principales dans les espaces suivants (R,G,B), (Y,I,Q) et (L,a*,b*) [Fal+93]. Sa caractéristique de texture est une version améliorée des représentations de texture de Tamura [TMY78], c'est à dire une combinaison de contraste et directivité [EN94]. La forme est caractérisée par la surface, la circularité, l'excentricité, l'orientation des axes et un ensemble algébrique invariant de moments [SAF94, Fal+93].

Qbic est l'un des rares systèmes à prendre en compte l'indexation de caractéristiques de dimensions élevées. Dans son système d'indexation, la transformée de K-L est d'abord utilisé pour réduire le nombre de dimensions et ensuite l'arbre R+ est utilisé comme la structure d'indexation multidimensionnelle [Lee+94, Fal+93]. Il permet aussi la recherche de texte à partir de mots clé, la combinaison de la recherche de similarité par le contenu avec la recherche de texte par les mots clé.

Une démonstration de Qbic se trouve sur http://wwwqbic.almaden.ibm.com [FBF+94] et permet aussi des recherches par la forme pour des objets segmentés semi-automatiquement.

Virage

Le système Virage (http://www.virage.com) [Gup95,Bac+95,GJ97] permet la recherche par le contenu, comme en Qbic à partir de la couleur, la texture et la forme, de plus il permet la recherche à partir de la position des caractéristiques.

Les utilisateurs peuvent combiner différentes caractéristiques et affiner les poids associés aux caractéristiques de recherche .

Le système intégré à Informix (http://www.informix.com , autrefois Illustra) permet de valider le traitement du contenu défini par l'utilisateur et les procédures d'analyse à enregistrer dans la base de données multimédia. Les "blades" de données pour le texte, les images, le son et la vidéo deviennent disponibles par les fournisseurs d'Informix.

RetrievalWare (Excalibur)

RetrievalWare est un système de recherche par le contenu développé par Excalibur Technologies Corp. (http://www.excalibur.com) [Dow93].

Le système visuel RetrievalWare permet des recherches sur la forme en niveau de gris, sur la forme en couleur, sur la texture et sur la couleur en utilisant les techniques de reconnaissance de forme adaptatives. En effet, il utilise un réseau de neurones dans la recherche de l'image[Dow93].

Excalibur permet aussi des couplages "blade" avec les bases de données d'Informix. Un exemple de "blade" de données est le détecteur de changement de scène pour une vidéo. Le "blade" de données détecte les changements de plans ou de la scène dans la vidéo et produit un résumé de celle-ci par des images exemples à partir de chaque plan.

Une page de démonstration se trouve à http://www.excalibur.com/cgi-bin/sdk/cst/cst2.bat

3. Les systèmes pour le World Wide Web

Le système WebSEEK (http://disney.ctr.columbia.edu/WebSEEk/) [SC96a] construit plusieurs indexes pour les images et les vidéos aussi bien à partir des indices visuels tels que la couleur qu'à partir des indices non visuels tels que les termes clés, des expressions concernant les sujets et les types d'image/vidéo.

Pour classer les images et les vidéos dans des catégories de sujet, un dictionnaire de termes clés est construit à partir des termes choisis apparaissant dans une URL (Uniform Resource Locator). Les termes sont sélectionnés à partir de leur fréquences d'occurrence et de leur signification pour les termes des sujets. Le dernier critère est pris en compte manuellement. Par exemple l'URL (http://www.chicago.com/people/michael/final98.gif) produirait les termes suivants : people, michael, final.

Une fois que le dictionnaire des termes clés est construit, la partie du répertoire de l'image et celle de la vidéo de lURL sont parcourues et analysées. Cette analyse produit un ensemble initial de catégorie d'images et de vidéos qui est alors vérifié manuellement. Les vidéos sont résumées en sélectionnant une image par seconde de la vidéo dont l'ensemble forme une image GIF animée.

Le projet WebSeer [SFA96] vise à classer les images à partir de leur indices visuels. Il comprend d'autres caractéristiques nouvelles telles que la classification de l'image en différentes catégories, par exemple les photographies, les graphiques, etc., l'intégration du détecteur de visage CMU [RBK95], des mots clés de recherche associés au texte tels que la référence http, titre de page.

Le surfer d'images Yahoo (http://isurf.yahoo.com) utilise Excalibur et Visual RetrievalWare pour rechercher les images et la vidéo sur le World Wide Web.

4. Autres systèmes

ART MUSEUM [HK92], développé en 1992, est l'un des premiers systèmes de recherche d'images par le contenu. Il utilisait le contour comme caractéristique visuelle pour la recherche.

À l'université de Syracuse, un autre projet se dédie à l'identification de feuilles de plantes mais pourrait éventuellement être utilisé pour identifier des fossiles, des os, de la monnaie, etc. Il

s'agit de choisir à l'écran, parmi les exemples présentés, une forme similaire à celle de la feuille que le chercheur veut identifier. Par la suite, le chercheur peut manipuler la forme grâce à des " points de contrôle " pour qu'elle corresponde le plus exactement possible à ce qu'il recherche. Le système retrouvera ensuite, dans la banque d'images, celles qui ressemblent le plus à la feuille à identifier. Ici, l'utilisation de mots pour effectuer une recherche a été complètement mis de côté [Hog+91].

À l'université de Californie à Berkeley, le système basé sur le logiciel IMAGEQUERY permet aux usagers d'effectuer en premier lieu une recherche textuelle. Le système affiche par la suite une mosaïque d'imagettes que l'usager peut regarder pour ensuite choisir celles qui concorderaient le plus avec ses besoins. Il s'agit d'un modèle hybride qui tente de s'éloigner de l'indexation textuelle des images en utilisant à la fois du texte et du visuel pour effectuer le repérage du matériel visuel [Bes90].

Le système *Blobworld* [Car+97] développé à UC-Berkeley permet la localisation de régions cohérentes en couleur et en texture. Il permet à l'utilisateur d'explorer la représentation interne de l'image de requête, et les résultats afin de pouvoir consulter les images qui ne sont pas pertinentes et de pouvoir modifier la requête en conséquence.

Le système *CAETIIML* (http://www.videolib.princeton.edu/test/retrieve) construit à l'université de Princeton, permet une combinaison de recherche en ligne par la similarité et une recherche hors ligne par sujets [YW97].

Parmi les systèmes qui utilisent des légendes ou des annotations pour la recherche d'image, il y a le système de recherche d'images par légende ("caption") de l'université de Dublin qui utilise WordNet[Mil95], un dictionnaire électronique/thesaurus, pour l'extension de requête [SQ96].

Rohinni et Srihari [RS95] décrivent un système qui utilise un modèle sémantique pour interpréter les "caption" dans le but de guider l'identification de personnes.

Le système SCORE [ATY+97, ATY+95] utilise un modèle étendu d'entités/associations pour représenter le contenu de l'image et le WordNet pour étendre les requêtes aussi bien que les descriptions de la base de données.

Le système *SEMCOG* [Li+97] permet l'identification semi-automatique d'objets. D'autres systèmes de recherche des images peuvent être trouvés dans [FSA96 , ASF97, STL97, SW95,Gon+94 ,Wu+95,OAH96,DFB97].

E. Conclusions

Jusqu'à présent, nous n'avons pas abordé les problèmes liés aux langages de requête. Or, si pour les SGBD traditionnels, la sémantique d'une requête basée sur le contenu reste relativement simple, car grosso modo elle consiste à trouver les données correspondant aux mots clés spécifiés dans la requête. Pour une base d'images l'expression d'une requête et l'accès aux données deviennent bien plus difficiles. Nous avons vu déjà les défauts d'une approche purement textuelle qui consiste, pour exprimer une requête, à utiliser une description textuelle des images recherchées. D'autres approches sont aussi possibles. Par exemple, Smaïl et al. [Sma94] propose une méthodologie du raisonnement à base de cas où les résultats des sessions des précédentes sont stockés en vue d'une recherche d'information interactive ultérieure après adaptation. Les performances des différents sessions de recherche peuvent être comparées à partir de critères quantitatifs tels que l'effet sur l'utilisateur, ou le

temps de réponse, et de critères qualitatifs tels la précision, le bruit et le rappel, pour la pertinence des documents trouvés en réponse à la requête.

D'une façon générale, un langage de requête doit permettre à l'utilisateur d'interroger les bases d'images par le contenu suivant différents critères. Une recherche doit pouvoir s'appuyer sur l'une ou une combinaison de plusieurs critères que l'on peut lister ci-dessous :

Recherche par attributs définis par l'utilisateur**:** L'utilisateur précise les valeurs pour les attributs définis. Par exemple, retrouver les images qui ont été prises à Paris le 4 Juillet avec une résolution d'au moins 300 dpi. Il s'agit donc les recherches classiques basées sur le texte.

Recherche simple de caractéristique visuelle. L'utilisateur précise certaines valeurs en pourcentage pour une caractéristique. Par exemple, retrouver les images qui contiennent 50% du rouge, 25% du bleu et 25% de jaune.

Recherche par une combinaison de caractéristiques. L'utilisateur combine différentes caractéristiques et indique leur valeurs et leur poids. Par exemple, rechercher les images de couleur verte et de texture d'un arbre où la couleur a un poids de 75% et la texture un poids de 25%.

Recherche de caractéristiques localisées. L'utilisateur indique des valeurs de caractéristiques et les positions en précisant l'arrangement des régions. Par exemple, rechercher les images où la couleur bleu ciel se trouve sur la moitié supérieure de l'image et le vert sur la moitié inférieure.

Recherche par concept. Certains systèmes permettent à l'utilisateur de définir des concepts simples à partir de caractéristiques extraites par le système [OS95,Als+96,GP97]. Par exemple, l'utilisateur définit le concept d'une plage comme un petit cercle jaune en haut qui symbolise le soleil, sur une grande région bleue au milieu qui symbolise la mer, sur la couleur du sable dans la partie inférieure.

Recherche par l'exemple**.** Le système fournit un ensemble d'images présélectionnées. L'utilisateur choisit une de ces images et recherche les images qui lui ressemblent dans la base. La similarité peut être déterminée en fonction des caractéristiques choisies par l'utilisateur. Par exemple, retrouver les images qui contiennent des textures semblables à cet exemple. Une version légèrement différente de ce type de requête est la possibilité offerte à l'utilisateur pour extraire des régions à partir de l'image exemple et les dispose sur une image de requête. La recherche par le dessin peut être considérée comme un cas particulier de la recherche par l'exemple.

Recherche d'objets dans une image. L'utilisateur peut décrire les caractéristiques d'un objet dans une image au lieu de décrire l'image entière. Par exemple, retrouver les images contenant une voiture verte près du centre.

Recherche par les relations spatiales entre objets**.** L'utilisateur indique les objets, leurs attributs et les relations spatiales entre eux. Par exemple, rechercher les images où un homme se trouve à gauche d'une voiture rouge.

Soulignons cependant que rares sont les travaux qui permettent une recherche par objets et leurs arrangements spatiaux qui résument au mieux le contenu visuel d'une image. Le plus souvent les systèmes d'indexation d'images proposent une recherche d'images dite par l'exemple où la requête est constitué par l'image toute entière : l'utilisateur fournit, choisit voire dessine une image similaire à l'image qu'il recherche.

Dans nos travaux qui sont exposés dans la suite, nous proposons des techniques simples et efficaces permettant de segmenter et d'indexer les objets visuellement homogènes dans une

image. Cette indexation d'objets visuellement homogènes et localisés permet par la suite une recherche sophistiquée par objets et par leurs arrangements spatiaux.

III. Les indices visuels

Les indices visuels sont des objets extraits d'une image caractérisant la perception visuelle de celle-ci. En général, ces indices concernent les couleurs, les textures ou les formes représentées par soit des contours soit des régions d'une image. Remarquons cependant que contours et régions sont complémentaires car on définit qualitativement les régions comme les zones de l'image homogènes au sens d'un certain critère, tandis que les contours sont les zones de transition entre des régions homogènes.

Les propriétés requises par les indices visuels dépendent de l'application et peuvent être caractérisées par les critères suivants :

- **Compacité**. La représentation de l'images doit être aussi concise que possible pour réduire la complexité des algorithmes ultérieurs.
- **Le fait d'être intrinsèque**. Les indices visuels doivent correspondre à la projection dans l'image d'objets physiques; en particulier, ils doivent être invariants par changement de point de vue.
- **Robustesse.** La représentation doit être peu sensible aux petites variations d'intensité dans l'image provoquées par des bruits divers tels que ceux liés à l'acquisition, la digitalisation, etc.
- **Discrimination.** Les indices visuels doivent posséder des propriétés qui permettent de les discriminer, afin de faciliter la mise en correspondance entre deux descriptions (pour la reconstruction tridimensionnelle ou pour l'appariement avec un modèle).
- **Précision.** La position des indices visuels doit pouvoir se calculer avec précision car la qualité de la localisation des objets en dépend.
- **Densité.** La densité des indices visuels doit être suffisante pour représenter tous les objets intéressants de la scène.

Dans la suite, nous allons d'abord définir d'une manière précise les indices visuels couramment utilisés dans le domaine de l'indexation d'images par le contenu visuel, à savoir les couleurs, les textures et les formes. Ensuite, nous introduisons quelques mesures de similarité sur les histogrammes et des techniques pour la réduction de dimensions car s'agissant d'une recherche par essence de similarité, nous ne pouvons pas faire l'économie du choix d'une métrique de distance et de l'utilisation d'une technique de réduction de dimensions pour des raisons de performance. Mais commençons par le commencement. Nous donnons tout de suite une définition de l'image.

A. Définition de l'image numérique

Une image est une forme discrète d'un phénomène continu obtenue après discrétisation. Le plus souvent, cette forme est bidimensionnelle. L'information dont elle est le support est caractéristique de l'intensité lumineuse (couleur ou niveaux de gris). Suivant les méthodes employées pour la traiter, l'image numérique peut être considérée comme

- Un signal bidimensionnel à support et à valeurs bornées,

- Un processus stochastique I[s]
- Un vecteur aléatoire $I = (I_1,I_2,\ldots I_s,...)$ ou I_s est une variable aléatoire associée au site s
- Une surface (i,j,I[i,j]) de l'espace N^3.

Nous nous intéressons ici à la représentation d'une image comme un signal bidimensionnel à support et à valeurs bornés que l'on notera:

$I[x,y]$ avec $[x,y] \in N^2$ et $0 \leq x \leq L-1$; $0 \leq y \leq C-1$.

$I \rightarrow [0,M]^p$ définit une image de L lignes et C colonnes dont l'information portée est définit dans un espace à p dimensions.

- Si I est une image binaire, alors (p,M)=(1,1)
- Si I est une image en niveaux de gris, alors p = 1 et le plus souvent M = 255
- Si I est une image couleur, alors p = 3 et le plus souvent M = 255

L'image résulte de l'échantillonnage du signal continu I(u,v). On désigne par s=[x,y] un site de coordonnées [x,y] dont la valeur sera notée I[s]ou I[x,y]. Les sites constituent un ensemble S organisé par un maillage. Pour s donné, on appelle pixel le couple (s,I[s]) [Coq+95].

B. La couleur

La couleur d'un objet dépend de sa géométrie, de la source de lumière qui l'éclaire, de l'environnement, et du système visuel humain. L'appréciation d'une couleur comprendra donc une partie subjective et une partie objective.

Si la perception des images en couleur est un processus complexe, pour simplifier on estimera que la couleur est uniquement fonction des pixels. Cela revient à dire qu'elle n'est pas influencée par les couleurs qui l'entourent et que l'on ignore aussi les conditions telles que la lumière ambiante, l'adaptation à l'angle de vue ou à la distance, et la qualité d'affichage de l'image. En général, il est difficile de contrôler ces paramètres dans des environnements d'applications générales tels que c'est supposé dans la recherche d'images par le contenu.

1. Colorimétrie

La colorimétrie est une théorie psychophysique descriptive qui repose sur le fait que la plupart de couleurs peuvent être reproduites par un mélange de trois couleurs indépendantes fixes.

Traiter des images réelles avec des effets de transparence, des textures ou des ombrages nécessite une grande quantité de couleurs qu'il faut évidemment coder. Rappelons que la lumière est une onde électromagnétique qui possède une distribution spectrale en fréquence ou en longueur d'onde. La lumière visible (par l'œil humain) correspond à un spectre compris entre 380 et 770 nanomètres, allant donc de l'ultra violet à l'infra rouge.

Une lumière monochromatique correspond à une longueur d'onde bien précise (en fait à une bande de largeur 1 nanomètre). On montre expérimentalement que l'on peut reconstituer toute lumière visible à partir de trois lumières monochromatiques bien choisies, appelées couleurs primaires.

En 1931, la C.I.E. (Commission Internationale de l'Eclairage) a défini trois courbes spectrales $x(\lambda)$, $y(\lambda)$,$z(\lambda)$ (en particulier $y(\lambda)$ est la courbe de sensibilité de l'œil).

Pour une lumière de distribution spectrale P(λ), on a :

$$X = \int P(\lambda)x(\lambda)d\lambda\,,\ Y = \int P(\lambda)y(\lambda)d\lambda\,,\text{et}\ Z = \int P(\lambda)z(\lambda)d\lambda$$

Les trois nombres X, Y, Z caractérisent la couleur de distribution P(λ) et sont appelés les valeurs de trichromacité. Par suite, à toute couleur correspond un vecteur dans un espace à trois dimension :

C = X.u+Y.v+Z.w

où u,v,w sont des vecteurs unitaires représentatifs de chaque couleur primaire.

On peut normaliser en posant

$$\mathbf{x = \frac{X}{X+Y+Z}} \qquad \mathbf{y = \frac{Y}{X+Y+Z}} \qquad \mathbf{z = \frac{Z}{X+Y+Z}}$$

On a donc x + y + z = 1, ce qui permet de calculer z en fonction de x et y.

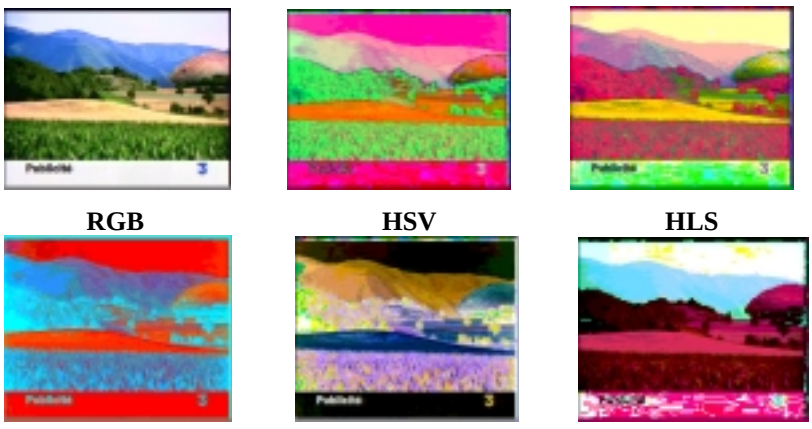
Le système XYZ a été développé par la C.I.E. en 1931. Les trois couleurs primaires X, Y, Z sont reliées linéairement aux couleurs primaires plus naturelles R, V, B (rouge, vert, bleu) :

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{vmatrix} 0.607 & 0.174 & 0.201 \\ 0.299 & 0,587 & 0.114 \\ 0,000 & 0,066 & 1.117 \end{vmatrix} \begin{pmatrix} R \\ V \\ B \end{pmatrix}$$

Pour désigner les couleurs dans un processus en recherche d'image, plusieurs modèles de représentation peuvent être utilisés.

2. Choix de l'espace de couleur

Le choix de l'espace de couleur est très important pour la perception des couleurs proches par l'utilisateur. En recherche des images par le contenu, il est à la fois un espace de segmentation des objets et de visualisation des images. Les images sont souvent représentées et affichées en espace RGB. Suivant les applications, les caractéristiques sont plus perceptibles dans certains espaces plutôt que dans d'autres. La figure 1 illustre la représentation d'une image dans quelques espaces de couleurs.



RGB **HSV** **HLS**

CMY **YUV** **YIQ**

Figure 1 – Représentation d'une image dans différents espaces de couleur

Le modèle RGB

La Commission Internationale de l'Eclairage (CIE) a choisi en 1931 les trois longueurs d'onde suivantes pour représenter les trois couleurs fondamentales:

* Bleue 435,8 nm
* Vert 546,1 nm
* Rouge 700 nm

Dans un tel modèle, les trois axes correspondent aux couleurs primaires Rouge, Vert, Bleu (cf. figure 2).

La diagonale principale représente les niveaux de gris. Ce modèle constitue le principe de base les moniteurs de télévision et des écrans à balayage; en effet , c'est par superposition de rouge, de vert et de bleu que l'affichage couleur est réalisé.

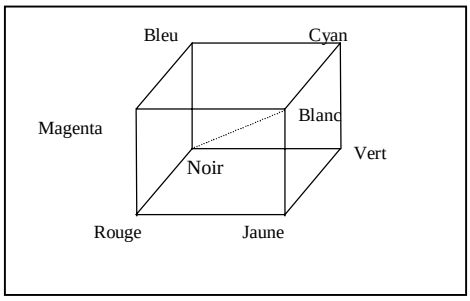


Figure 2 - Le système RGB

Le système RGB normalisé correspond aux chromacités en RGB.

$$r = \frac{R}{R+G+B};\ g = \frac{G}{R+G+B};\ b = \frac{B}{R+G+B}$$

La distance entre deux couleurs C_1= (R_1,G_1,B_1) et C_2 =(R_2,G_2,B_2) est définie par la distance Euclidienne suivante :

$$d_E(C_1,C_2) = \sqrt{(R_1-R_2)^2+(G_1-G_2)^2+(B_1-B_2)^2}$$

Les modèle CMY et CMYK

Les couleurs primaires Cyan, Magenta et jaune sont ici les complémentaires des précédentes. Ce système est utilisé par certaines imprimantes couleurs à impact, et à jet d'encre . On parle de quadrichromie ou de modèle CMYK (K pour noir!) pour qui quatre cartouches d'encre sont nécessaires à l'impression couleur, la cartouche d'encre noire permet d'obtenir un noir parfait, le mélange des autres couleurs ne donnent pas généralement un résultat satisfaisant.

$$\begin{pmatrix} C \\ M \\ Y \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} r \\ g \\ b \end{pmatrix}$$

$$\begin{pmatrix} C_k \\ M_k \\ Y_k \end{pmatrix} = \begin{pmatrix} C \\ M \\ Y \end{pmatrix} - \begin{pmatrix} K \\ K \\ K \end{pmatrix} \text{ avec } K = \min(C,M,Y)$$

Le modèle YIQ

Il s'agit d'un recodage du système RGB par NTSC (National Television Standards Committee):

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{vmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{vmatrix} \begin{pmatrix} R \\ V \\ B \end{pmatrix}$$

Y est la luminance, I et Q sont liés à la teinte et à la saturation. C'est un modèle qui est fondé sur des observations psychophysiques [Rus95].

Le modèle YUV

L'espace de couleur YUV est utilisé dans le PAL et SECAM des standards de la télévision couleur [NH88].

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{vmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.438 \\ 0.615 & -0.515 & -0.100 \end{vmatrix} \begin{pmatrix} R \\ V \\ B \end{pmatrix}$$

Le modèle YCrCb

L'espace de couleur YCrCb est utilisé dans le standard JPEG.

$$\begin{pmatrix} Y \\ Cr \\ Cb \end{pmatrix} = \begin{vmatrix} 0.2990 & 0.5870 & 0.1140 \\ 0.5000 & -0.4187 & -0.0813 \\ -0.1687 & -0.3313 & 0.5000 \end{vmatrix} \begin{pmatrix} R \\ V \\ B \end{pmatrix}$$

Munsell

*a) Le modèle La*b**

En 1976, la CIE propose l'espace couleur CIE L*a*b*, appelé aussi le CIE 1976 L*a*b*, dont les transformations, issues de l'espace XYZ, sont définies par:

$$\begin{cases} L^* = \begin{cases} 903.3(Y/Y_0)^{1/3} & \text{pour } 0 \le Y < 0.01 \\ 25(100Y/Y_0)^{1/3} - 16 & \text{pour } Y \ge 0.01 \end{cases} \\ a^* = 500\,[(X/X_0)^{1/3} - (Y/Y_0)^{1/3}] \\ b^* = 200\,[(Y/Y_0)^{1/3} - (Z/Z_0)^{1/3}] \end{cases}$$

où X_0,Y_0,Z_0 sont les coordonnées du blanc de référence.

L* est la luminance de la couleur, a* et b* représentent les composantes qui définissent la composante chromatique de la couleur. Dans ce système, la métrique euclidienne est significative des différences de couleur perçues[Kan96].

La distance entre deux couleurs C_1= (L^*_1,a^*_1,b^*_1) et C_2 =(L^*_2,a^*_2,b^*_2) est définie par la distance suivante :

$d_E(C_1,C_2) = \sqrt{(\Delta_{L^*})^2 + (\Delta_{a^*})^2 + (\Delta_{b^*})^2}$ avec Δ_{L^*}=L^*_1-L^*_2 ; Δ_{a^*}=a^*_1-a^*_2 et Δ_{b^*}=b^*_1-b^*_2 ;

*b) Le modèle L*u*v**

La même année, en 1976, l'espace L*u*v* basé sur le système colorimétrique de Munsell est devenu également un standard [Kun93]. L'espace L*u*v* est défini à partir de l'espace XYZ par les équations suivantes:

$$\begin{cases} L^* = 25[\frac{100Y}{Y_0}]^{1/3} - 16 \\ u^* = 13\,L^*(u' - u_0) \\ v^* = 13\,L^*(v' - v_0) \end{cases}$$

$$\text{and } u' = \frac{4X}{X + 15Y + 3Z};\ v' = \frac{9Y}{X + 15Y + 3Z};$$

$$(Y_0, u_0, v_0) = (1.000, 0.201, 0461)$$

De même que le modèle L*a*b*, u* et v* définissent la partie chromatique de la couleur.

La distance entre deux couleurs C_1= (L^*_1,u^*_1,v^*_1) et C_2 =(L^*_2,u^*_2,v^*_2) est définie par la distance suivante :

$d_E(C_1, C_2) = \sqrt{(\Delta_{L^*})^2 + (\Delta_{u^*})^2 + (\Delta_{v^*})^2}$ avec Δ_{L^*}=L^*_1-L^*_2 ; Δ_{u^*}=u^*_1-u^*_2 et Δ_{v^*}=v^*_1-v^*_2 ;

c) Le modèle LHC

Cet espace perceptuel [Cel90] tente de rendre plus homogène les régions colorimétriques. Il dérive de l'espace L*a*b* en utilisant les formules suivantes:

$L = L^*$

$H = \tan^{-1}(\frac{b^*}{a^*})$

$C = \sqrt{(a^*)^2 + (b^*)^2}$

Le modèle HSV

Le modèle Teinte-Saturation-Luminance ou HSV (Hue, Saturation, Value) est plus proche de la perception de la couleur, ce modèle utilise un espace en forme d'hexagone dont l'axe est celui de la luminance L. Pour L = 1, on a les couleurs d'intensité maximale.

La teinte T est donnée par l'angle entre l'axe rouge et un point de l'hexagone. La saturation S est donnée par la distance entre l'axe de la luminance et un point de l'hexagone. L'espace de couleur TSL est en forme cylindrique(cf. figure 3).

La séparation d'une couleur en différentes composantes de teinte, saturation et luminance sont intuitives pour l'utilisateur [Hun89].

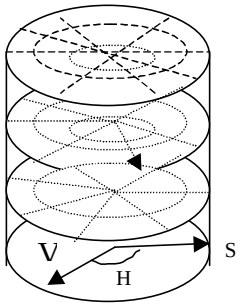


Figure 3 - Espace de couleur HSV

$$H = \begin{cases} \dfrac{60.(g-b)}{\max(r,g,b)-\min(r,g,b)} & \text{pour } r = \max(r,g,b) \\ 120 + \dfrac{60.(g-b)}{\max(r,g,b)-\min(r,g,b)} & \text{pour } g = \max(r,g,b) \\ 240 + \dfrac{60.(g-b)}{\max(r,g,b)-\min(r,g,b)} & \text{pour } b = \max(r,g,b) \end{cases}$$

$$S = \begin{cases} \dfrac{\max(r,g,b)-\min(r,g,b)}{\max(r,g,b)+\min(r,g,b)} & \text{pour } 0 < L \le 0.5 \\ \dfrac{\max(r,g,b)-\min(r,g,b)}{20-\max(r,g,b)+\min(r,g,b)} & \text{pour } 0.5 \le L < 1.0 \end{cases}$$

$$V = \max(r,g,b)$$

si H < 0 alors H= 360 + H

La distance entre deux couleurs C_1= (H^*_1,$S^*_{1,}V^*_1$) et C_2 =(H^*_2,$S^*_{2,}V^*_2$) est définie par la distance suivante:

$$d_E(C_1,C_2) = \sqrt{(S_1 \sin H_1 - S_2 \sin H_2)^2 + (S_1 \cos H_1 - S_2 \cos H_2)^2 + (V_1 - V_2)^2}$$

Le modèle HLS

Le modèle Teine-Saturation-Intensité ou HLS (Hue, Lightness, Saturation), également proche de la perception de la couleur, utilise un espace en forme de double cône (cf. figure 4) . L'axe allant du noir au blanc est l'axe de l'intensité; la teinte est déterminée par l'angle polaire et la saturation par le rayon polaire.

$$L = \frac{\max(r,g,b)+\min(r,g,b)}{2}$$

L et S sont compris entres 0 et 1. La distance entre deux couleurs C_1=(H^*_1,L^*_1,S^*_1) et C_2 = (H^*_2,L^*_2,S^*_2) est définie par la distance suivante :

$$d_E(C_1,C_2) = \sqrt{(S_1 \sin H_1 - S_2 \sin H_2)^2 + (S_1 \cos H_1 - S_2 \cos H_2)^2 + (L_1 - L_2)^2}$$

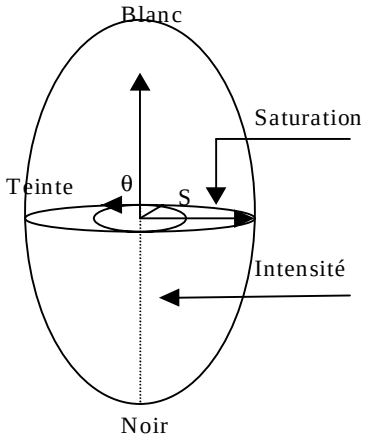


Figure 4 - Représentation de l'espace HLS

3. Couleur vraie et couleur indexée

On peut diviser les systèmes graphiques en deux catégories :

- Ceux qui codent un pixel sur 16, 24 ou 32 bits. On parle dans ce cas de "vraie couleur".

En effet, avec 24 bits, on peut avoir 16,7 millions de nuances colorées ce qui est largement au-dessus de la sensibilité de l'œil, mais utile pour la retouche photographique par exemple.

- Ceux qui codent un pixel sur 8 bits ou moins. On peut avoir au maximum 256 nuances.

Toutefois, avec la technique de la palette, on peut obtenir le même nombre de couleurs que précédemment. On parle alors de couleur indexée. Dans tous les modèles précédents, la couleur est codée avec trois nombres. Si un pixel est codé sur p bits, on peut obtenir 2^p couleurs différentes à l'affichage :

- p = 8 256 couleurs : suffisant pour les graphiques, mais pas pour des images réalistes

- p = 24 16 millions de couleurs : satisfaisant, mais place importante de l'image.

D'où la notion de palette de couleur : les p bits ne représentent plus une couleur mais une adresse dans une table de couleurs (palette) ; l'élément correspondant de la table, codé sur b bits, représente une couleur. Pour p = 12 bits, on a 2^{12} adresses, soit 4096 adresses ; les éléments de la palette peuvent être de longueur b = 3x8 (pour les trois doses de couleurs primaires) et représentent donc 4096 couleurs parmi 16 millions.

4. Quantification d'une image couleur

Les caractéristiques candidates utilisées dans le processus d'indexation et de recherche sont celles qui sont obtenues après un processus de transformation (T) en espace de représentation approprié puis une réduction de dimension du vecteur associé avec une méthode (M). En général, on utilise un espace de couleur où la caractéristique visuelle est la mieux perçue, puis on procède à une réduction du nombre de ces attributs avec la quantification par exemple pour un espace de couleur donné. Le but de ce double processus, est d'identifier la transformation et la quantification pour la couleur et la texture qui génèrent des espaces complets, compacts et uniforme pour la perception.

Pour cela, on opère d'abord un changement de coordonnées et les nouvelles T_k sont quantifiées indépendamment les unes des autres. La transformation doit être choisie de manière à ce que la différence de perception due à la quantification soit minimisée. La figure 5 montre le principe de quantification d'une image en couleur.

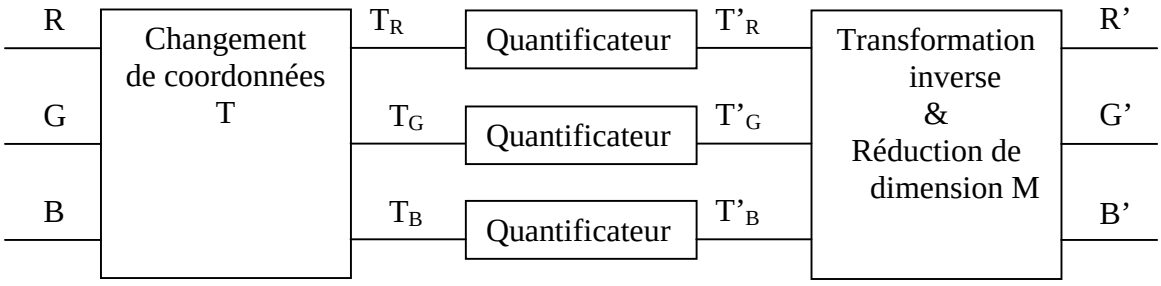


Figure 5 - Transformation et quantification d'une image en couleur

Par exemple, pour réduire le nombre de couleurs est de transformer les composantes R, G, et B en composantes RG, BY et WB par :

- RG = R – G
- BY = 2 x B – R – G
- WB = R + G +B

Cette représentation permet une meilleure quantification de l'intensité WB par rapport à RG et BY. Un histogramme de 2048 indices peut être généré si les composantes RG et BY sont divisées en 16 sections et WB est divisée en 8 sections [IP97].

Wan et Kuo [WK96] ont étudié la quantification de la couleur dans différents espaces de couleurs tels que les espaces RGB, HSV, YUV, et les espaces de Munsell. Ils en concluent que la quantification dans l'espace HSV donne de meilleurs résultats, lors de la recherche des images, que les autres espaces de couleurs.

Couleur

Si on suppose que chaque pixel est quantifié sur p bits, et on désire extraire le $n^{ième}$ bit le plus significatif.

$$u = k_1 2^{p-1} + k_2 2^{p-2} + ... + k_{p-1} 2 + k_p$$

La sortie doit s'écrire v = {N_{Max} si k_n=1; 0 sinon}

On peut montrer que $k_n = i_n - 2i_{n-1}$ avec i_n=Ent[$u/2^{p-n}$]. Cette transformation est utile pour déterminer le nombre de bits ayant une signification visuelle dans une image.

Niveaux de gris

L'image est caractérisée par une quantité de lumière affectée à chaque point et correspond à un niveau de gris mesuré par une intensité. Il est commode d'associer une échelle quantitative aux niveaux d'intensité: 0 pour le noir, 1 pour le blanc. Entre deux il y a les gris. Supposons que l'on veuille définir n intensités différentes, du noir vers le blanc. Soit I_0 l'intensité la plus basse (noir), et 1 l'intensité la plus forte (blanc). Les différents niveaux sont:

$I_0 = I_0$, $I_1 = rI_0$, $I_2 = rI_1 = r^2 I_0$ ….. $I_{n-1} = r^{n-1} I_0 = 1$

Donc $r = (1/I_0)^{1/n-1}$ et $I_j = I_0^{(n-1-j)/n-1}$

5. Quelques manipulations utiles:

Renforcement de contraste:

Un défaut d'éclairage ou un défaut d'un capteur peut engendrer des images peu contrastés. Il peut, donc, être nécessaire d'augmenter la dynamique en renforçant le contraste , pour compenser cet effet. Une modification de contraste est une application:

$$v = \begin{cases} \alpha u & 0 \le u < a \\ \beta(u-a) + v_a & a \le u < b \\ \gamma(u-b) + v_b & b \le u < N_{Max} \end{cases}$$

La pente est choisie supérieure à l'unité dans la région dont on veut augmenter le contraste.

Découpage de niveau de luminance:

Cette opération permet l'extraction de régions d'intérêt en niveau de luminance(entre a et b) du reste de l'image et le représenter par un niveau L.

$$v = \begin{cases} L & a \le u < b \\ 0 & \text{ailleurs : on ne conserve pas le fond} \\ u & \text{ailleurs : on conserve le fond} \end{cases}$$

Cette technique est utile quand on connaît un indice visuel de l'image caractérisé avec un niveau donné.

C. La texture

B. Julesz [Jul62] était le premier à s'intéresser aux statistiques du second ordre pour la discrimination des textures. Quant à A. Gagalowicz [Gag83], il a constaté sur des expériences psychovisuelles, que l'ensemble des attributs discriminants d'une texture homogène est supérieur à l'ensemble formé par les moments des deux premiers ordres et inférieure à l'ensemble des moyennes d'espace du second ordre. Ces dernières sont rassemblées dans les matrices de cooccurrence.

Un des descripteurs les plus simples d'une texture, pour une macro-segmentation éventuelle, est d'utiliser les moments de l'histogramme d'une image (ou d'une région). En effet, A. Gagalowicz [Gag83] a pu synthétiser des textures différenciables à l'œil et possédant des moments identiques jusqu'à l'ordre 4.

1. Indices basés sur les histogrammes normalisés

En notant u la variable aléatoire représentant une couleur (niveau de gris), les éléments de l'histogramme sont estimés par :

$$h[i] = \frac{\text{Nombre de pixels de couleur (niveau de gris) de i}}{\text{Nombre total de pixels}} \qquad i = 0,1,\ldots G_{Max}$$

On peut caractériser un histogramme par une série de paramètres tels que :

Moments d'ordre n

$$m_n = \sum_{i=0}^{G\,max} i^n \times h[i]$$

Moments absolus d'ordre n

$$m'_n = \sum_{i=0}^{G\,max} |i|^n \times h[i]$$

Moments centrés d'ordre n

$$\mu_n = \sum_{i=0}^{G\,max} (i - m_1)^n h[i]$$

Moyenne

$$\overline{m} = m_1 = \sum_{i=0}^{G\,max} i \times h[i]$$

Variance

$$\sigma^2 = \mu_2 = \sum_{i=0}^{G\,max} (i - m_1)^2 h[i]$$

Skewness

$$S = \mu_3 = \sum_{i=0}^{G\,max} (i - m_1)^3 h[i]$$

Kurtosis

$$K = \mu_4 = \sum_{i=0}^{G\,max} (i - m_1)^4 h[i]$$

Entropie

$$E = -\sum_{i=0}^{G\,max} h[i] \times \log h[i]$$

Le premier moment non centré traduit la moyenne de l'image (m_1) la variance (σ^2) traduit la dispersion. On peut en déduire (smoothness)

$$S_1 = \frac{\sigma^2}{1 + \sigma^2}$$

S_1 est nul si l'intensité est constante et tend vers 1 pour des grandes valeur de σ^2.

Le troisième moment traduit la dissymétrie de l'histogramme (skewness), tandis que le quatrième moment traduit son caractère plus ou moins plat (kurtosis).

2. Probabilité conjointe ou cooccurrence

L'inconvénient des caractéristiques présentées ci-dessus est ne pas fournir d'information spatiale, et de ne pas intégrer la position des pixels les uns par rapport aux autres.

Pour remédier à ce problème on considère une fonction :

$$w(dx, dy) = |f(x + dx, y + dy) - f(x, y)|$$

Pour chaque couple (dx,dy), on calcule l'histogramme correspondant

$$h_{(dx,dy)}[i] = \frac{\text{Nombre de pixels de niveau de gris i dans w(dx, dy)}}{\text{Nombre total de pixels dans w(dx, dy)}}$$

La moyenne de w(dx,dy) peut alors être décrite par :

$$\overline{m}_{(dx,dy)} = \sum_{i=0}^{G\,max} i \times h_{(dx,dy)}[i]$$

Cette valeur décrit la taille de la texture d'une région.

Toutes les fonctions définies précédemment qui utilisent h[i] peuvent être maintenant calculées à partir de $h_{(dx,dy)}[i]$

3. Matrices de cooccurrence

Ce sont en fait des histogrammes à deux dimensions. Soient deux pixels $u_1$et u_2 situés à une distance r dans une direction θ. On peut construire l'histogramme à deux dimensions correspondant :

$$c[i, j] = f(r, \theta, i, j)$$

L'histogramme est calculé par la formule :

$$c[i, j] = \frac{\text{Nombre de paires de pixels tels que } u_1 = i \text{ et } u_2 = j}{\text{Nombre total de couple de pixels dans la région}}$$

La relation géographique entre u_1 et u_2 est fixée suivant l'utilisation : par exemple, u_r doit être à une distance r de u_1, dans une direction θ.

Cet histogramme se représente par une matrice L x L, appelée matrice de cooccurrence. Puisque cette matrice dépend de r et de θ, il est possible de détecter une texture donnée en choisissant correctement ces deux paramètres. En pratique, on réduit généralement cet ensemble de niveaux de gris à 8 ou 16 valeurs [Phil88].

Les matrices de cooccurrence contiennent une masse d'information trop importante et difficilement manipulable dans son intégralité. Haralick a proposé un ensemble de paramètres pour résumer l'information portée par une matrice de cooccurrence C={c[i,j] }. Il a défini quatorze indices, prenant en compte l'ensemble de la matrice C, qui correspondent à des caractères descriptifs des textures comme les caractéristiques suivantes:

Maximum

$$P_{max} = \max_{i,j}(c[i, j])$$

C'est un indice qui donne une indication sur la plus forte densité dans la configuration (r,θ) donnée.

Moment d'ordre n

$$\sum_i \sum_j (i - j)^n c[i, j]$$

Cet indicateur a de faibles valeurs quand les grandes valeurs de C se produisent près de la diagonale principale ((i-j) faible).

Moment inverse d'ordre n

$$EDM_n = \sum_i \sum_j \frac{c[i,j]}{|i-j|^n} \qquad i \neq j$$

Ce paramètre a l'effet inverse de l'indicateur précédent.

Entropie

$$E = \sum_i \sum_j c[i,j] \log c[i,j]$$

L'entropie est une mesure de l'effet aléatoire (maximum quand tous les éléments de C sont égaux). Elle fournit un indicateur du désordre que peut présenter une texture.

Uniformité

$$\sum_i \sum_j c[i,j]^2$$

Inversement, à l'entropie, cette valeur est la plus petite quand les termes de C sont tous égaux.

Homogénéité

$$\frac{1}{M^2} \sum_i \sum_j c[i,j]^2$$

M est le nombre de couples (i,j).

Cet indice est d'autant plus élevé que l'on retrouve souvent le même couple de pixels, ce qui est le cas quand le niveau de gris est uniforme ou quand il y a périodicité dans le sens de la translation [Coc+95a].

Contraste

$$\frac{1}{M(L-1)^2} \sum_{k=0}^{L-1} k^2 \sum_{|i-j|=k} c[i,j]$$

Chaque terme de la matrice C est pondéré par sa distance à la diagonale. Cet indice est élevé quand les termes éloignés de la diagonale de la matrice sont élevés, i.e. quand on passe souvent d'un pixel très clair à un pixel très foncé ou inversement.

Directivité

$$\frac{1}{M} \sum_i c[i,i]$$

La directivité est importante s'il y a des pixels de même niveau de gris séparés par une translation [Coc+95a].

4. Spectre de Fourier

La transformée de Fourier permet de passer d'une représentation de l'image dans le domaine spatial (coordonnées m,n) à une représentation dans le domaine fréquentiel (coordonnées u,v) [Coc+95a].

Pour une image I[m,n], avec m et n entiers, $0 \leq m \leq M-1$ et $0 \leq n \leq N-1$qui est un signal à support borné , sa transformée de Fourier discrète est:

$$F[u,v] = \frac{1}{MN} \sum\sum I[m,n].e^{-2j\pi(\frac{um}{M}+\frac{vn}{N})}$$

avec $0 \leq u \leq M-1$ et $0 \leq v \leq N-1$

Le spectre est très riche en information. On peut en extraire les composantes fréquentielles les plus énergétiques de l'image. Il peut être exploité aussi pour calculer des filtres passe-bande, etc.

5. Autocorrélation

La disposition spatiale des texels peut être représentée par le caractère plus ou moins étendu de la fonction d'autocorrélation r(k,l) traduisant le lien statistique spatial [Dub99] :

$$r(k,l) = \frac{m_2(k,l)}{m_2(0,0)} = \frac{1}{N} \frac{1}{m_2(0,0)} \sum_m \sum_n [u(m-k,n-l)]^2$$

avec: $m_i(k,l) = \frac{1}{N_R} \sum_{(m,n)\in R} \sum [u(m-k,n-l)]^i$

u étant la variable aléatoire représentant le niveau de gris et N_R est le nombre de pixels de la région R.

Le caractère grossier "coarseness" de la texture est proportionnel à la largeur de la fonction d'autocorrélation, représentée par la distance entre x_0 et y_0 tels que

$r(x_{0,}0)=r(0,y_0)=1/2$

D'autres mesures sont déduites de la fonction génératrice M(k,l) :

$$M(k,l) = \sum_m \sum_n (m-\mu_1)^k (n-\mu_2)^l r(m,n)$$

La fonction d'autocorrélation n'est pas discriminante pour les textures : des textures différentes peuvent présenter des fonctions identiques.

D. La forme

L'indice de forme peut être traitée sous deux angles de vue. Il peut être associé aux contours des régions qui sont préalablement déterminées dans l'image, il sera donc caractérisé par l'aspect géométrique de ces régions que nous présentons au chapitre suivant. Il peut être également représenté par un histogramme des couleurs(ou niveaux de gris) de l'image obtenue après la détection de ses contours.

On peut considérer cette deuxième approche comme globale, puisqu'elle ne décrit pas avec précision la forme d'une région ou d'un objet de l'image.

Dans ce paragraphe, on présente brièvement les méthodes globales de représentation de la forme et les méthodes générales d'extraction des contours.

1. Représentation par les moments

Le moment d'une image f(x,y) est une fonction bornée sur un support $\Re$. Le moment d'ordre (p+q) est défini par:

$$m_{pq} = \iint f(x,y) x^p y^q dxdy \quad p,q=0,1,\ldots$$

$$m_{pq} = \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} x^p y^q f(x,y)$$

Si on fait f(x,y) égal à 1, on obtient le moment de la forme de la région. Les résultats suivants sont donc aussi bien applicables à des distributions de niveaux de gris qu'à des formes d'objets.

On définit aussi le moments central μ_{pq} par:

$$\mu_{pq} = \iint f(x,y)[x-\bar{x}]^p [y-\bar{y}]^q dxdy \quad p,q=0,1$$

$$\mu_{pq} = \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} [x-\bar{x}]^p [y-\bar{y}]^q f(x,y) = \sum_{r=0}^{p} \sum_{s=0}^{q} C_r^p C_s^q (-\bar{x})(-\bar{y}) m_{p-r,q-s}$$

avec $x_G = \bar{x} = \frac{m_{10}}{m_{00}}$ et $y_G = \bar{y} = \frac{m_{01}}{m_{00}}$ qui sont les cordonnées du centre de gravité G (x_G,y_G).

$$C_r^p = \frac{p!}{r!(p-r)!}$$

Alors,

$$\mu_{00} = m_{00} = \mu$$
$$\mu_{10} = \mu_{01} = 0$$
$$\mu_{20} = m_{20} - \mu\bar{x}^2$$
$$\mu_{11} = m_{11} - \mu\bar{x}\bar{y}$$
$$\mu_{02} = m_{02} - \mu\bar{y}^2$$
$$\mu_{30} = m_{30} - 3m_{20}\bar{x} + 2\mu\bar{x}^3$$
$$\mu_{21} = m_{21} - m_{20}\bar{y} - 2m_{11}\bar{x} + 2\mu\bar{x}^2\bar{y}$$
$$\mu_{12} = m_{12} - m_{02}\bar{x} - 2m_{11}\bar{y} + 2\mu\bar{x}\bar{y}^2$$
$$\mu_{03} = m_{03} - 3m_{02}\bar{y} + 2\mu\bar{y}^3$$

Certaines fonctions des moments sont invariantes par rapport aux transformations géométriques usuelles: leur utilisation interviendra pour reconnaître des objets de forme connue qui ont subi une telle transformation. Si on retient par exemple, de représenter la forme de l'objet par p moments, un objet se représente comme un point dans un espace de dimension p et les méthodes de reconnaissance des formes statistiques peuvent être utilisées pour faire la reconnaissance. Par exemple, les moments centrés μ_{pq} sont invariants par translation [Dub99].

Plusieurs attributs de la forme peuvent être déterminés à partir de ces moments[Lev85].

- Superficie S = m00
- Variance

En plus du centre de gravité, on calcule les variance σ_x et σ_y dans les deux directions par :

$$\sigma_x = \sqrt{\frac{\mu_{20}}{m_{00}}} \quad \sigma_y = \sqrt{\frac{\mu_{02}}{m_{00}}}$$

- L'angle θ entre l'axe principale de la forme et l'axe horizontal qui est défini par :

$$\frac{1}{2}\arctan\left[\frac{2\mu_{11}}{\mu_{20}-\mu_{02}}\right]$$

- L'excentricité est définie par :

$$\sqrt{\frac{\mu_{02}\cos^2\theta+\mu_{20}\sin^2\theta-\mu_{11}\sin 2\theta}{\mu_{02}\sin^2\theta+\mu_{20}\cos^2\theta+\mu_{11}\cos 2\theta}}$$

- Les moments-invariants: Les sept premiers invariants des moments M_1, …M_7 proposés par Hu [Hu61] sont définis par :

$M_1 = \mu_{20}+\mu_{02}$

$M_2 = (\mu_{20}-\mu_{02})^2+4\mu_{11}^2$

$M_3 = (\mu_{30}-3\mu_{12})^2+(3\mu_{21}-\mu_{03})^2$

$M_4 = (\mu_{30}+\mu_{12})^2+(\mu_{21}+\mu_{03})^2$

$M_5 = (\mu_{30}+3\mu_{12})(\mu_{30}+\mu_{12})[(\mu_{30}+\mu_{12})^2-3(\mu_{21}+\mu_{03})]^2 +$
$(3\mu_{21}-\mu_{03})(\mu_{21}+\mu_{03})[3(\mu_{30}+\mu_{12})^2-(\mu_{21}+\mu_{03})^2]$

$M_6 = (\mu_{20}-\mu_{02})[(\mu_{30}+\mu_{12})^2-(\mu_{21}+\mu_{03})^2]+[4\mu_{11}(\mu_{30}+\mu_{12})(\mu_{21}+\mu_{03})]$

$M_7 = (3\mu_{21}-\mu_{03})(\mu_{30}+\mu_{12})[(\mu_{30}+\mu_{12})^2-3(\mu_{21}+\mu_{03})^2] -$
$(\mu_{30}-3\mu_{12})(\mu_{21}+\mu_{03})[3(\mu_{30}+\mu_{12})^2-(\mu_{21}+\mu_{03})^2]$

- Les moments de rotation ϕ_{pq} sont définies par :

$$\phi_{pq} = \sum\sum (-1)^{q-s} C_r^p C_s^q (\cos\theta)^{p-r+s} (\sin\theta)^{q-s+r} \mu_{p-r+q-s,r+s}$$

- Les moments standards sont alors définis par :

$$N_{pq} = \frac{\phi_{pq}}{\phi_{00}^{\gamma}} \text{ où } \gamma=1+(p+q)/2.$$

2. Détection de contour

La notion de contour est reliée à celle de variation en chaque pixel. Une variation existe si le gradient est localement maximum ou si la dérivée seconde présente un passage par zéro. Le

gradient repose sur la définition de deux masques H_1 et H_2 qui calculent le gradient de l'image dans deux directions orthogonales. Les principaux masques utilisées sont les suivants :

H_1 H_2

Roberts :

0	1
-1	0

1	0
0	-1

Prewitt

-1	0	1
-1	0	1
-1	0	1

-1	-1	-1
0	0	0
1	1	1

Sobel

-1	0	1
-2	0	2
-1	0	1

-1	-2	-1
0	0	0
1	2	1

Isotropique

-1	0	1
$-\sqrt{2}$	0	$\sqrt{2}$
-1	0	1

-1	$-\sqrt{2}$	-1
0	0	0
1	$\sqrt{2}$	1

D'autres algorithmes principaux pour la détection de contours sont utilisés tels que les masques de Kirsh, et les méthodes de Canny et de Dériche, [Der91]…etc..

Méthodes de relaxation

On considère que chaque pixel fait partie d'une région unique de l'image et que la frontière est fonction du gradient entre les pixels adjacents. On part du principe qu'il n'existe que des contours horizontaux et verticaux. La méthode consiste à modifier les connaissances qu'on a sur un contour en un point en fonction des informations qu'on a sur les pixels voisins. A chaque contour, on associe une mesure de confiance : un contour peut être de "faible confiance" ou de "forte confiance ".

Transformée de Hough

Cette méthode repose sur un changement de repère, on passe dans l'espace des paramètres. Elle suppose que la forme peut être représentée sous forme paramétrique connue.

Détection de droites

Pour détecter des droites quelconques, on doit travailler sur un tableau à deux entrées puisqu'il y a deux paramètres. L'équation générale de ces droites est : $y = mx + c$

On préfère la représentation en (ρ,θ) suivante : $\rho = x \cos\theta + y \sin\theta$

La transformée de Hough de cette droite est un point dans l'espace des paramètres (ρ,θ).

Par un point (x_i,y_i), il passe une infinité de droites d'équation ci-dessus : ce sont toutes les combinaisons (ρ,θ) qui vérifient cette équation. Si on discrétise les valeurs possibles de (ρ,θ), on obtient plusieurs courbes dans l'espace (ρ,θ). Pour un ensemble de points alignés, ces courbes se coupent en un point qui correspond aux paramètres de la droite liant les points.

Détection d'arcs de cercles

Considérons des cercles d'équation : $(x-a)^2 +(y-b)^2 = r^2$

Il suffit ici d'incrémenter un tableau à trois paramètres et rechercher les maxima.

Pour des arcs de cercle passant par un point donné que l'on choisit pour origine, l'équation s'écrit :

$(x-a)^2 +(y-b)^2 =a^2 + b^2$

Un point (x_i,y_i) sera situé dans l'espace des paramètres (a,b) sur une droite d'équation :

$2x_ia+2y_ib =x_i^2+y_i^2$

Si de nombreux points sont sur un cercle de centre (a',b'), les droites correspondantes vont se couper au point de coordonnées (a',b').

Les calculs peuvent être diminués en tenant compte des directions de gradient. La transformée de Hough peut être généralisée pour la détection d'objets de forme analytique inconnue.

E. Histogrammes

1. Définition et normalisation

Les caractéristiques de la couleur ou de la texture peuvent être définie par des histogrammes. Un histogramme h reflète la distribution des caractéristiques comme elles apparaissent dans l'image ou dans la région. Un histogramme est représenté par un vecteur de M dimensions où M est le nombre des indices (entrées) de l'histogramme.

Soit i une caractéristique donnée (un niveau de gris, une couleur quantifiée ou un élément de texture) h[i] traduit le nombre d'occurrences (apparitions) du niveau de gris i dans l'image. Par exemple pour un pixel de niveau de gris i, on calcule son nombre total en niveaux de gris dans l'image.

Pour rendre un tel histogramme insensible et invariant aux différentes opérations géométriques telles que le changement d'échelle , en plus de la rotation et de la translation, on définit un histogramme normalisé h^N qui est défini par:

$$h^N[i] = \frac{h[i]}{\sum_{k=0}^{M-1} h[k]}$$

h^N est un histogramme dont les valeurs sont comprise entre 0 et 1.

Pour chaque i on divise son nombre d'occurrences par le nombre total des occurrences dans l'histogramme. Chaque entrée i correspond alors à sa probabilité d'apparence dans l'image. La

définition de l'histogramme se généralise aux images multi-bandes, l'histogramme est alors une fonction de p variables où p désigne le nombre de canaux.

Suivant l'espace de représentation de la caractéristique, un histogramme peut être une distribution discrète mono-dimensionnelle ou n-dimensionnelle avec n le nombre de composantes de l'espace de représentation. Par exemple, pour un espace de couleur TSL quantifié en k teintes (T) l saturations (S) et m luminances (L), l'histogramme correspondant peut être représenté soit par trois vecteurs de dimensions respectives k, l et m. Chaque vecteur correspond à une composante de tailles suivantes: k pour T, l pour S et m pour L. L'histogramme peut être représenté aussi avec un vecteur de dimension k x l x m, qui représente la probabilité "de jointure" des couleurs (intensités) des 3 plans de couleur

Les distances que nous présentons dans le paragraphe suivant peuvent être appliquées aux deux représentations de l'histogramme.

2. Choix des distances

Soient q et t deux images qui sont représentées par leurs histogrammes respectifs h_q et h_t .

Définition d'une métrique

Dans un espace métrique la distance entre deux vecteurs h_i et h_j doit satisfaire aux conditions:

$d(h_i, h_j) \geq 0$ (= 0 si et seulement si $h_i \equiv h_j$)

$d(h_i, h_j) = d(h_j, h_i)$

$d(h_i, h_j) \leq d(h_i, h_k) + d(h_k, h_j)$

Ces conditions ne sont pas toujours respectées par les distances utilisées en traitement d'image et il est en général plus correct de parler de mesure de dissemblance ou encore de mesure de distorsion (comme en traitement de la parole).

Métrique générale de Minkowski

La métrique de Minkowski est un indice de proximité entre deux images q et t décrites par des vecteurs (histogrammes) h_q et h_t, est défini par $d(h_q, h_t)$:

$$L_r = d(h_q, h_t) = \left(\sum_{k=0}^{M-1} \left| h_q[k] - h_t[k] \right|^r\right)^{1/r} \qquad r \geq 1$$

Trois valeurs particulières de r correspondent aux trois distances les plus utilisées:

r = 1

Distance en valeur absolue ou L_1 (Manhattan ,"city-bloc"):

$$L_1 = d(h_q, h_t) = \sum_{k=0}^{M-1} \left| h_q[k] - h_t[k] \right|$$

r = 2

Distance Euclidienne ou L_2:

$$L_2 = d(h_q, h_t) = \sum_{k=0}^{M-1} (h_q[k] - h_t[k])^2 = (h_q - h_t)^T (h_q - h_t)$$

r➔∞

Distance de Chebychev , Distance de l'échiquier (chessboard metric) ou L_∞:

$$L_\infty = d(h_q, h_t) = \underset{k=0,p}{Max} |h_q[k] - h_t[k]|$$

On vérifie facilement que ces distances sont des vraies distances.

Ces distances correspondent aux mesures de similarités entre deux images quelconques représentées par leurs vecteurs discriminants (histogrammes normalisés en particulier).

Intersection des histogrammes

L'intersection des histogrammes a été utilisée dans la recherche des images de couleur par Swain et Ballard dans [SB91]. Leur objectif était de trouver des objets donnés dans les images en utilisant les histogrammes de couleur. Quand la taille de l'objet (q) était plus petite que la taille de l'image (t), et les histogrammes ne sont pas normalisés, alors $| h_q | \leq | h_t |$ et l'intersection des histogrammes est définie par:

$$d'_{q,t} = 1 - \frac{\sum_{k=0}^{M-1} \min(h_q[k], h_t[k])}{\sum_{k=0}^{M-1} h_q[k]}$$

Cette distance $d'_{q,t}$ n'est pas une métrique puisque $d'_{q,t} \neq d'_{t,q}$. Cette distance a été utilisée aussi par R. Schettini dans son approche pour localiser les objets multicolores [Sch94].

Cependant, cette distance peut être modifiée pour produire une vraie distance :

$$d_{q,t} = 1 - \frac{\sum_{k=0}^{M-1} \min(h_q[k], h_t[k])}{\min(|h_q|, |h_t|)}$$

Mais si, les histogrammes h_q et h_t sont normalisés alors $| h_t | = | h_q |$ et les distances ci-dessus sont de vraies distances. d'(q,t) lors d(q,t). Il a été montré dans [SB91] que quand $| h_t | = | h_q |$, la distance en valeur absolue est égale à $d'_{q,t}$.

Cette distance d'intersection normalisée est insensible à la résolution de l'image, la taille de l'histogramme, l'occlusion, et la profondeur. Cependant, cette distance ne tient pas compte des similarités entre les différentes entrées des histogrammes.

Distance du cosinus

La métrique du cosinus est couramment utilisé pour mesurer la similarité entre des documents textuels [WMB94]. Elle calcule la différence dans une direction, indépendamment de la longueur des vecteurs, et la distance est donnée par l'angle entre les deux vecteurs.

$$\cos\theta = \frac{h_q^T h_t}{|h_q||h_t|}$$

Distance quadratique

La distance quadratique compare chaque élément de l'histogramme source avec tous les autres éléments des l'histogramme cible.

Une distance quadratique a été utilisée dans le système QBIC d'IBM pour la recherche d'images à partir d'histogrammes de couleur [Fli+95]. Dans [Ort+97], il a été reporté qu'une distance quadratique entre deux histogrammes de couleur donne de meilleurs résultats que les autres distances. La distance quadratique entre deux histogrammes h_q et h_t est définie par:

$$d(q,t) = (h_q - h_t)^T A (h_q - h_t) = \sum_{i=0}^{M-1}\sum_{j=0}^{M-1} a[i,j](h_q[i] - h_t[j])(h_q[i] - h_t[j])$$

où $A = [a_{ij}] = a[i,j]$ représente la similarité entre les couleurs correspondant respectivement aux indices i et j. Notons que si A est la matrice unité cette mesure devient la distance Euclidienne.

La distance quadratique est une vraie distance métrique quand $a_{ij} = a_{ji}$ et $a_{ii}=1$.

Cette distance est plus coûteuse que les métriques de Minkowski puisqu'elle calcule la similarité entre tous les éléments. Cependant, il y a des techniques de décomposition et d'approximation qui permettent d'améliorer le temps de calcul [SC97c].

Quand $A = \Sigma^{-1}$, on parle de distance de Mahalanobis. C'est un cas particulier de la distance quadratique dans laquelle la matrice de transformation est remplacée par une matrice de covariance. Pour appliquer la distance de Mahalanobis, l'histogramme h est traité comme un ensemble de variables aléatoires $\{x_0,x_1,x_2,\ldots x_{M-1}\}$.

Cette dernière est définie par Σ où $\Sigma=\{\sigma^2_{ij}\}$ et $\sigma^2_{ij} = E\{x_i x_j\} - E\{x_i\}E\{x_j\}$. Avec $E\{y\}$ la moyenne de la variable aléatoire y.

La distance de Mahalanobis entre deux histogrammes est définie par :

$$d(q,t) = (h_q - h_t)^T \Sigma^{-1} (h_q - h_t)$$

3. Autres distances

On présente ici d'autres distances qui sont utilisées en recherche d'images. La distance principale des composantes est calculée à partir de la moyenne de chaque composante de la caractéristique de l'image. Par exemple, un vecteur de couleurs moyennes $v = (\bar{r}, \bar{g}, \bar{b})$ est obtenu par calcul de la couleur principale dans chacune des trois composantes (r,g,b). La distance entre deux vecteurs de couleur principales v_q et v_t est définie par:

$$d(v_q,v_t) = (v_q - v_t)^T (v_q - v_t)$$

Notons que chaque entrée dans l'histogramme de couleur h est rattaché à un point (r,g,b) dans l'espace tridimensionnel RGB. V est calculé à partir de h par v = c.h, où c possède taille de M x 3, et c[i] donne le triplet (r,g,b) correspondant à l'entrée i de l'histogramme. En général, pour K composantes d'une caractéristique et M entrées d'un histogramme, c possède la taille K x M. La distance entre deux images q et t est définie alors par:

$$d(q,t) = (h_q - h_t)^T C^T C (h_q - h_t)$$

De même que pour la moyenne des composantes, d'autres moments des composantes peuvent être considérées, tels que la variance ou "skewness". Stricker et Orengo ont proposé l'utilisation des moments de couleur [SO95] qui sont calculés directement à partir des histogrammes , et définissent la distance entre moments par la formule suivante.

$$d(h_q, h_t) = \sum_{i=1}^{n} w_{i1}\left|\bar{h}_q - \bar{h}_t\right| + w_{i2}\left|var(h_q) - var(h_t)\right| + w_{i3}\left|\sigma_q - \sigma_t\right|$$

où $\bar{h}_x$ est la moyenne de l'histogramme , var (h_x) est la variance et σ_m le moment de degré 3 de h_x. Cette distance a été aussi évoqué par C. Djeraba et al dans [DFB97].

Jain et A. Vailaya [JV95] ont utilisé trois histogrammes 1-D dans leurs expériences. Les distances de similarité utilisées dans ce cas entre deux images sont définies par les équations suivantes :

Intersection d'histogramme:

$$d(h_q,h_t)=\frac{\sum_r \min(h_q[r],h_t[r])+\sum_g \min(h_q[g],h_t[g])+\sum_b \min(h_q[b],h_t[b])}{3*\min(|h_q|,|h_t|)}$$

Distance euclidienne:

$$d(h_q,h_t)=1-\sqrt{\frac{\sum_r (h_q[r]-h_t[r])^2+\sum_g (h_q[g]-h_t[g])^2+\sum_b (h_q[b]-h_t[b])^2}{2*3}}$$

On note ici que les valeurs de ces distances sont comprises entre [0,1]

Nastar et al.[Nas+98] proposent une combinaison de signatures qui est linéaire qui utilise la moyenne μ_i et l'écart-type σ_i de la distance d pour chaque attribut i pour calculer la distance normalisée:

$$d'(x^{(i)},y^{(i)})=\frac{d(x^{(i)},y^{(i)})-(\mu_i-3\sigma_i)}{6\sigma_i}$$

Où $x^{(i)}$ et $y^{(i)}$ sont les signatures des images X et Y pour l'attribut d'image i. La nouvelle distance d' prend essentiellement des valeurs dans [0…1] et peut être combinée de façon linéaire avec les distances normalisées des autres attributs de l'image.

Récemment Androutsos et al. [APV99] présentent une mesure basée sur la distance angulaire entre deux vecteurs h_q et h_t. Cette mesure est une combinaison de l'angle entre deux vecteurs et de la magnitude de la différence entre leurs vecteurs. Elle est définie par la formule suivante:

$$d(h_q,h_t)=1-\underbrace{\left[1-\frac{2}{\pi}\cos^{-1}\left(\frac{h_q.h_t}{|h_q||h_t|}\right)\right]}_{\text{angle}}\underbrace{\left[1-\frac{|h_q-h_t|}{\sqrt{3.255^2}}\right]}_{\text{magnitude}}$$

Le facteur de normalisation $\pi/2$ dans la portion de l'angle est attribué au fait que l'angle maximum qui peut être atteint est $\pi/2$. De même, le vecteur maximum de différence qui peut exister est (255,255,255) et sa magnitude $\sqrt{3\times 255^2}$.

Ces deux facteurs de normalisation contribuent à ce que d() soit dans l'intervalle [0,1].

4. Bouclage de pertinence

L'utilisateur détermine à partir des résultats d'une recherche celles qui sont pertinentes et celles qui le sont pas. Le système utilise ensuite cette information pour reformuler la requête pour trouver les images que l'utilisateur souhaite[BM95]. En utilisant les histogrammes de couleur, le bouclage de pertinence est défini par:

$h_q^{k+1} = \left\| \alpha h_q^k + \beta \sum_{i \in I_P} h_i - \gamma \sum_{j \in I_{NP}} h_j \right\|$ où ||.|| indique une normalisation. Les nouvelles images ou vidéos sont trouvées en utilisant h^{k+1}.

F. Réduction de dimension

La méthode de réduction des données ne doit pas conduire à une perte d'informations. Deux approches sont souvent citées dans la littérature, la transformée de Karhunen-Loeve (K-L) et les méthodes de clustering .

Dans [NS96], NG et Sedighian utilisent l'approche des "eigenimage" pour effectuer la réduction de dimension. Dans [FL95] Faloutsos et Lin proposent une approximation rapide de la transformée K-L pour réduire le nombre de dimension. Leurs résultats expérimentaux montrent que la plus part des ensembles de données réelles (vecteurs des indices visuels) peuvent être considérablement réduits en dimension sans une dégradation significative dans la qualité de la recherche [NS96,FL95, WJ96a].

On utilise aussi le clustering pour réduire le nombre des dimensions. C'est une méthode qui est largement utilisée dans les domaines de la reconnaissance de forme [DH73], l'analyse de la parole [RJ93] et la recherche d'information [SC97a] .

Il s'agit d'une méthode de classification automatique qu'on utilise généralement pour regrouper les objets similaires par exemple des formes, des signaux ou des documents, en vue d'une reconnaissance ultérieure [SM83]. Parmi les autres méthodes proposées dans le contexte de la recherche des images, on trouve l'analyse du discriminant de Fisher qui peut donner aussi de bons résultats [SC94].

1. Transformée de Karhunen-Loève

La méthode optimale visant à réduire des points de n dimensions en points de k dimensions est la transformée de Karhunen-Loève (K-L) [DH73, Fuk90]. K-L est optimal dans la sens où il minimise le MSE (mean square error) , où l'erreur est la distance entre chaque point et son image . La figure 6 : Illustration de la transformation de Karhunen-Loève montre un ensemble de points 2-d, et les 2 directions (x' et y') que suggère la transformation K-L.

K-L est souvent utilisé en reconnaissance des formes [Fuk90] pour choisir les caractéristiques importantes d'un ensemble de vecteurs. Il calcule les vecteurs propres de la matrice de covariance, les trie dans l'ordre décroissant des valeurs propres, et donne une approximation de chaque vecteur par les projections de ses k premiers vecteurs propres .

Cependant, la transformée de K-L a deux inconvénients:

- Elle n'est pas applicable à toutes les distances
- Le calcul devient fastidieux pour des grandes bases de données (N >> 1) avec plusieurs attributs (n>>1)[FD92,Dum94]

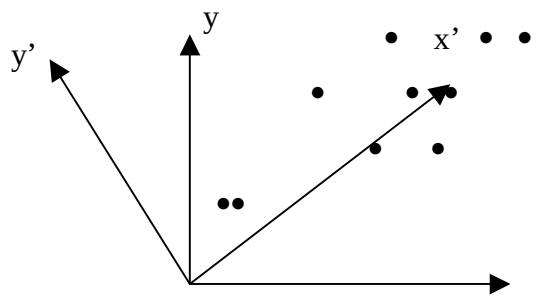


Figure 6 - Illustration de la transformation de Karhunen-Loève - Le meilleur axe de projection est x'

2. Coalescence

Les méthodes de coalescence ("clustering") cherchent à séparer un ensemble de données en différentes classes, c'est à dire à réaliser une partition de ces données. L'étude de ces méthodes n'est pas l'objet de ce document, aussi nous nous limiterons ici à citer les méthodes les plus connues :

- Les méthodes hiérarchiques. Elles peuvent être ascendantes ou descendantes. Les méthodes ascendantes partent d'autant de classes que de nombre de données. Elle regroupe ensuite les deux plus proches classes. A l'étape suivante, elle regroupe les deux classes les plus proches. Ceci jusqu'à ce que l'on obtienne plus qu'une seule et unique classe. Les méthodes descendantes partent d'une unique classe que l'on sépare en deux. L'intérêt principal de ces méthodes est la non - spécification d'un nombre de classes a priori.
- Les méthodes qui se fixent un nombre de classe au départ (c); Ce sont les algorithmes ISODATA dont le plus connu est le cas des moyennes mobiles (C-means) : on se fixe c centres de classes. On calcule les énergies inter et intraclasses. On modifie les centres et les affectations des données. On cherche ensuite de façon itérative les meilleurs centres ainsi que la meilleure classification à c donné.

Soit X l'ensemble des données à partitionner. On peut représenter le problème de coalescence par la détermination d'une fonction **d** :

$$\mathbf{d} \mid X \rightarrow N^*$$

$$\mid x \rightarrow \mathbf{d}(x)$$

où **d**(x) est le numéro de la classe à laquelle on associe x.

Par exemple, pour quantifier un espace de couleurs en K niveaux, on peut utiliser un algorithme de coalescence qui minimise le MSE [Nib+93].

G. Conclusion

Le processus d'indexation d'une image par le contenu peut être décomposé en quatre étapes essentielles [Mar97]:

- un prétraitement de l'image: amélioration, restauration, détection des contours…
- une extraction d'attributs visuels (formes, bords, textures) par seuillage, extraction de contours, etc.
- la segmentation proprement dite qui consiste à partager l'image en régions significatives. Pour cela diverses techniques sont classiques:
  - la segmentation par relaxation (méthode itérative) ;

    - la segmentation par les régions elles-mêmes (croissance de régions, fusion et scission);
    - la segmentation des textures (par exemple utilisation des matrices de cooccurrence).
- la classification et description des régions, des relations entre elles, de leurs similitudes et de différences. A ce stade, des mesures, par exemple nombre, aire, périmètre, orientation, forme, etc., servant à caractériser les régions correspondant à des objets sont souvent nécessaires.

Une part essentielle des travaux de recherche dans le domaine s'est portée sur l'étape d'extraction d'attributs visuellement significatifs, puisqu'ils permettent de résumer le contenu visuel d'une image. Comme nous l'avons vu déjà précédemment, lorsque l'attribut significatif est la couleur comme c'est souvent le cas, on utilise généralement l'histogramme de couleurs comme signature d'image. Cet histogramme fournit la distribution des couleurs dans l'image. L'indexation automatique fait appel parfois à une connaissance a priori du domaine d'application et nécessite beaucoup de calcul lors de l'extraction des caractéristiques.

Quant à l'étape de la recherche proprement dite, le principe est une recherche par la similarité pour laquelle il est nécessaire de fournir une image requête d'où est extraite, par le même mécanisme d'indexation, une série de primitives visuelles qui seront comparées à celles, extraites durant l'étape de l'indexation par le contenu visuel, des images présentes dans la base.

Un système de recherche d'images sophistiqué devrait permettre ce qu'une description de requête puisse faire référence à des régions de l'image qui se distinguent les unes aux autres sur la base de leur aspect visuel. Cependant, la facilité magique avec laquelle le système de perception visuelle de l'homme est capable de distinguer, de différencier et d'interpréter les différentes régions d'une image est, certes stimulante, mais le plus souvent frustrante pour celui qui cherche à confier cette tâche de segmentation à une machine. On ne connaît pas encore en profondeur les mécanismes de la perception visuelle chez l'homme et il serait vain de tenter de définir des critères généraux pour segmenter les images. Il n'y a pas de procédure universelle pour effectuer ce type d'analyse et on ne peut juger de l'efficacité d'une technique particulière que par l'intérêt de la segmentation obtenue dans le cadre de l'application considérée.

Aussi, allons nous présenter au chapitre suivant notre approche d'indexation d'images par le contenu basée sur la segmentation d'objets selon des critères d'homogénéité visuelles, s'agissant d'une recherche dans des bases d'images générales.

IV. Segmentation d'image en régions homogènes

La segmentation est l'étape essentielle dans un processus de recherche d'images par le contenu. Elle fait l'objet de nombreux travaux en traitement d'images. Pour l'indexation d'images par le contenu, les techniques de segmentation d'image ont pour but de mettre en évidence les régions homogènes, appelées aussi objets, de l'image et d'en extraire ses indices visuels. Une région homogène est une partition de l'image qui possède une certaine uniformité pour une ou plusieurs caractéristiques visuelles (couleur, intensité lumineuse, teinte, texture, …) et que n'en ont pas les régions voisines. Elle constitue un moyen efficace dans la réduction des données en vue d'une description économique plus proche du contenu visuel de l'image.

Les critères de segmentation et les caractéristiques des régions dépendent de la qualité des images à traiter et de l'usage qui sera fait des images segmentées. Les exigences en qualité de segmentation peuvent être différentes selon les critères choisis et selon l'application. Si pour l'extraction de la forme d'un objet, une segmentation juste est nécessaire, une segmentation grossière est suffisante pour extraire l'arrangement spatial des objets dans une image.

La segmentation d'une image I utilisant un prédicat d'homogénéité P est communément définie [Zuc76, HP74, Pav86] comme une partition $S = R_1, R_2, \ldots R_n$ de I telle que:

- $I = \cup R_i$, i [1…n]
- R_i est connexe, i [1…n]
- $P(R_i)$ = vrai, i [1…n]
- $P(R_i \cap R_j)$ = faux, i j, pour tout couple (R_i, R_j) de régions connexes.

Il est important de remarquer que ces conditions ne définissent pas, en général, une segmentation unique. Les résultats de segmentation dépendent par conséquent de l'ordre et la manière avec lesquelles les données sont traitées et non pas seulement de l'information contenu dans l'image. La segmentation en régions est plus souvent NP-difficile, ce qui nécessite parfois l'utilisation d'heuristiques. La segmentation en régions peut se ramener à un problème d'optimisation [HM93].

Une des techniques les plus utilisées, approche ascendante, est dite de croissance des régions (region growing). Cette technique part de la représentation de l'image comme un ensemble de pixels, les regroupe selon un double critère d'homogénéité et de voisinage. On parle de segmentation par agrégation de pixels. On peut aussi supposer que l'image est d'abord constituée de régions unitaires (obtenu après une première segmentation), le processus de croissance de régions est obtenu en réalisant une nouvelle segmentation de la configuration des étiquettes : les régions adjacentes voisines et similaires sont regroupées jusqu'à ce que les régions adjacentes voisines soient suffisamment différentes.

Cette croissance est conduite par l'utilisation d'un prédicat d'homogénéité qui permet d'identifier le critère de regroupement (mergion criterion) qui est une ou plusieurs contraintes à satisfaire par les régions.

Par opposition, l'approche descendante, au lieu de procéder par regroupement, procède par division (splitting). L'image est divisée, par la technique des quadtrees, puis les régions similaires sont regroupées toujours suivant ce double critère de voisinage et d'homogénéité. C'est l'algorithme de division - regroupement (split and merge).

Dans la suite du chapitre, nous définissions d'abord les critères d'homogénéité utilisés dans notre processus de segmentation. Ensuite, nous introduisons les deux approches, descendante et ascendante, qui ont été doublement appliquées dans nos travaux. Enfin, nous décrivons nos techniques permettant caractériser les relations spatiales entre les régions ainsi que les indices permettant de leur définir une signature.

A. Critères d'homogénéité

1. Choix de la distance

Pour tout pixel d'une image qui est caractérisé par un couple de coordonnées (x,y), on peut calculer des distances entre pixels. Les distances les plus courantes pour deux pixels $P(x_p,y_p)$ et $Q(x_q,y_q)$ sont :

- **Distance de Manhattan ou distance du "city-bloc":**

$d_C(P,Q) = | x_p - x_q | + | y_p - y_q |;$

- **Distance Euclidienne:**

$d_E(P,Q) = [(x_p - x_q)^2 + (y_p - y_q)^2] ^{1/2};$

- **Distance de l'échiquier (chessboard metric):**

$d_\infty(P,Q) = \text{Max } \{| x_p - x_q | , | y_p - y_q |\};$

On vérifie facilement que ces distances sont des vraies distances. Elles sont reliées par la propriété suivante:

$d_\infty (P,Q) \leq d_E(P,Q) \leq d_C(P,Q)$

2. Voisinage d'un pixel

Les définitions de voisinages et les métriques associées sont liées au type de maillage. Le maillage est obtenu de manière implicite par échantillonnage de l'image analogique lors de la numérisation [Coc+95b].

Le maillage est l'arrangement spatial géométrique des pixels dans l'image. La plupart des capteurs échantillonne en maillage carré (ou rectangulaire), les autres maillages triangulaire et hexagonal peuvent être simulés. La figure 7-a montre l'exemple d'un maillage hexagonal utilisé en imagerie. Il est évident que la première propriété irremplaçable est la connexité entre ces pixels.

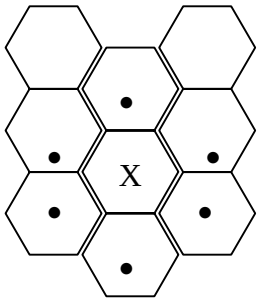
Deux pixels dans l'image sont "connexes" s'il existe au moins une suite de pixels qui les joint, et telle que deux pixels consécutifs de la suite satisfont à la condition de "connexité immédiate". La connexité immédiate entre deux pixels traduit le fait que ces pixels partagent des caractéristiques communes et qu'ils sont "voisins". Elle est conditionnée par la définition du "voisinage" d'un pixel.

Deux points de I sont dits connectés s'il existe au moins un chemin qui les relie, pour deux points p et q on notera cela par p ~ q.

La relation '~' est une relation d'équivalence qui factorise I en classes d'équivalence. Une classe d'équivalence sera appelée une composante connexe. Par définition Card(I,~) est l'ordre de connexion de l'image. Le type de voisinage utilisé influe bien sûr sur les objets résultants.

Par extension, on définit le voisinage de tout composante C de l'image par l'union de tous les voisinages des pixels de C.

Maillage hexagonal



(a)

	X	
X	p(x,y)	X
X	X	X

y pair

(b)

X	X	X
X	p(x,y)	X
	X	

y impair

(c)

Figure 7 - (a) Maillage hexagonal, (b) - (c): Simulation de voisinage v_6 sur une maille carrée

Ce type de forme peut être obtenu en simulant un maillage carré en prenant les 6 sites voisins de [x,y] de manière différente suivant que l'indice de colonne y est pair ou impair (cf. figures 7-b et 7-c) . Elle permet d'associer à P, un voisinage $v_6(P)$

Si y est pair [x-1,y], [x,y-1], [x,y+1], [x+1,y-1], [x+1,y], [x+1,y+1],

Sinon [x-1,y-1], [x-1,y], [x-1,y+1], [x,y-1], [x,y+1], [x+1,y],

Maillage carré

On distingue classiquement quatre types de connexité, selon la distance métrique utilisée: le 4-voisinage, 4-voisinage diagonal ,8-voisinage et le m-voisinage. Les deux métriques couramment utilisées en maillage carré sont d_C et d_∞.

4-voisinage

La métrique d_C permet d'associer à P, un ensemble de points $v_4(P)$ appelé 4-voisinage et défini par:

$v_4(P) = \{ Q \in S, d_C(P,Q) \leq 1 \}$

Connexité simple:

Le 4-voisinage ou "voisinage simple" d'un pixel est défini comme l'ensemble des pixels qui l'entourent et qui ne se trouvent pas sur une des deux diagonales passant par ce pixel.

Ce voisinage est représenté sur la figure 8-a. Le point P possède 4 voisins adjacents non diagonaux situés à une distance d_C=1.

Connexité diagonale :

Le 4-voisinage diagonal ou "voisinage diagonal" est défini à partir de la connexité diagonale entre les pixels .

Ce voisinage est représenté sur la figure 8-b. Le point P possède 4 voisins adjacents diagonaux situés à une distance d_C=1.

8-voisinage

La métrique d_∞ permet d'associer à P, un voisinage $v_8(P)$ défini par:

$v_8(P) = \{ Q \in S, d_\infty (P,Q) \leq 1\}$

Connexité d'ordre 8

Le voisinage 8-connexe d'un pixel est défini comme l'union des deux types de 4-voisinage. Il est défini comme l'ensemble de tous les pixels qui l'entourent (situés à une distance d_∞=1.).

Ce voisinage est représenté sur la figure 8-c. On dit que v_8 est un voisinage 8-connexe. La difficulté principale avec le 8-voisinage est qu'il produit des chemins multiples entre les pixels.

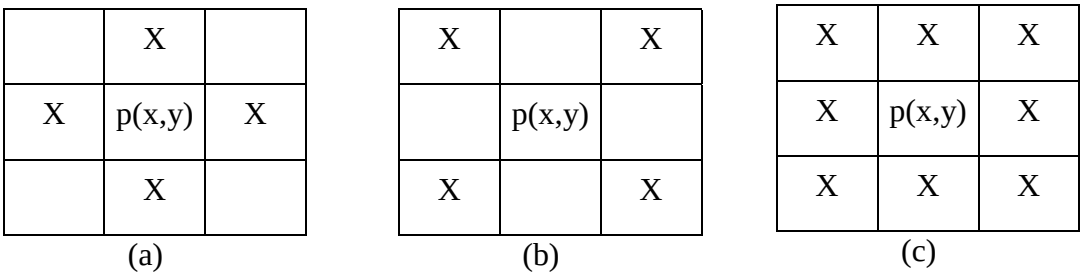


Figure 8 - Voisinage d'un pixel p(x,y): (a) 4-voisins (b) voisins diagonaux et (c) 8-voisins

Connexité d'ordre m

Deux points A et B sont m-connexes si et seulement si

- A et B sont 8-voisins
- A et B partagent le même critère d'homogénéité
- Il n'y a pas d'intersection entre l'ensemble du 4-voisinage de A avec B et celui de B avec A.

Ce type de connexité permet d'éliminer les chemins multiples entre deux pixels en enlevant le chemin diagonal si la connexité d'ordre 4 existe déjà entre les pixels (situés à une distance d_∞=1.).

La figure 9-a montre un exemple de connexité d'ordre 4 entre les pixels A et B avec les pixels de niveau de gris de 5. La figure 9-b montre la connexité d'ordre 8, où deux chemins existent entre les pixels A et B. Tandis que la figure 9-c montre la connexité d'ordre m entre les pixels A et B, où il n'y a pas de chemins multiples entre deux pixels successifs..

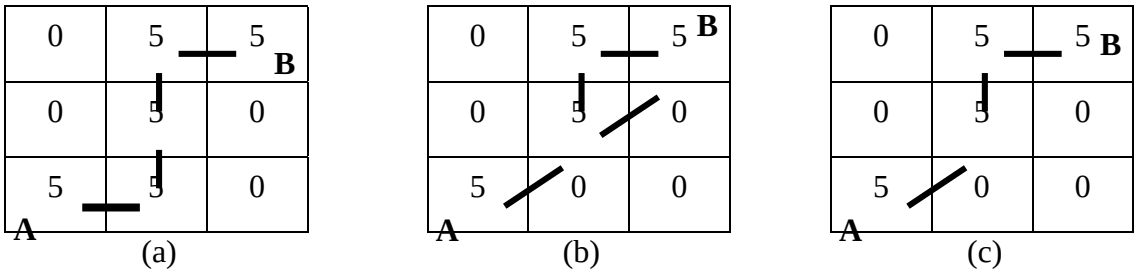


Figure 9 - Exemple de (a) connexité d'ordre 4, (b) connexité d'ordre 8, et (c) connexité d'ordre m

Remarque

Cependant, avec ces définitions, on peut avoir des paradoxes : sur la figure 10, *B* et *C* sont-ils connexes ? Avec la connexité d'ordre 4, les segments 1, 2, 3, et 4 sont considérés comme disjoints. Avec la connexité d'ordre 8, ces segments sont connexes, mais la zone intérieure est aussi connexe avec la zone extérieure.

On peut éviter ces problèmes en prenant des connexités différentes pour les objets et le fond. Une grille hexagonale, utilisée en morphologie mathématique, ne présente pas ces difficultés.

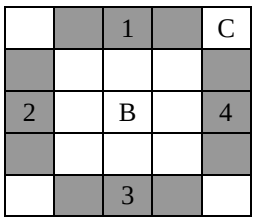


Figure 10 - Paradoxe de la connexité

B. Quadtrees

1. Principe de base

Les arbres quaternaires "quadtrees" sont une variante hiérarchique de la représentation spatiale d'une image encore appelée énumération spatiale. Cette représentation a été découverte indépendamment par plusieurs chercheurs à partir des années 1960 [HS79,Sam82]. L'idée fondamentale était de respecter le paradigme classique des informaticiens - diviser pour mieux régner- tout en permettant la représentation hiérarchique d'une surface [Mol96].

A partir d'une image carrée (2^L x 2^L) de pixels, on peut effectuer une division successive de l'image initiale en 4 blocs carrés ou quadrants ou quadtrees [Sam89], de taille moitié, jusqu'à obtenir des blocks homogènes suivant le (les) critère(s) d'homogénéité. Le quadtree est donc une arborescence dont la racine est l'image toute entière et dont chaque nœud (sauf les nœud terminaux) possède exactement 4 fils. Le quadtree est défini de manière récursive. A chacun

de ces quadrants est associé un nœud fils de la racine. Puis le processus de découpage en quatre quarts est itéré pour chacun des fils sans chevauchement des blocks. On peut découper ainsi l'image jusqu'aux pixels, l'arborescence possède alors L+1 niveaux. L'image est ainsi constituée d'un ensemble de quadrants ; chaque quadrant est défini par sa taille (longueur du coté du carré) et par sa couleur.

Chaque nœud du quadtree correspond à un bloc carré dans l'image d'origine. Chaque bloc possède 4 côtés qui composent son contour et 4 coins. Les quatre bords d'un bloc sont appelés Nord (N), Est (E), Sud (S) et Ouest (O). On a coutume de coder les nœuds et les feuilles dans l'ordre N, E, S et O ; par extension, les 4 coins sont notés NO, NE, SO et SE. La figure 11 illustre cette convention de désignation des blocks dans un quadtree. Les fils sont étiquetés avec une chaîne de 2 bits, où le premier bit indique

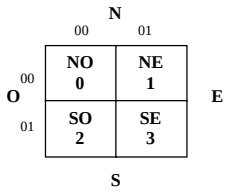


Figure 11 - Etiquetage des blocs

la direction horizontale ('gauche/droite' respectivement '0/1') et le second bit indique la direction verticale ('bas/haut' respectivement '0/1'). Les directions NO, NE, SO, and SE sont représentées respectivement par 0, 1, 2, et 3. Par exemple : NE = $(0\ 0\ 0\ 1)_2 = (1)_{10}$

Chaque quadrant pouvait être plein, à moitié plein ou vide ce qui était visualisé respectivement par les couleurs noire, grise ou blanche. La subdivision effectuée récursivement se poursuivait jusqu'à l'obtention de quadrants homogènes, vides ou pleins, ou jusqu'à ce qu'une profondeur choisie, dite de coupure, était atteinte. La figure suivante illustre ce procédé sur quatre itérations:

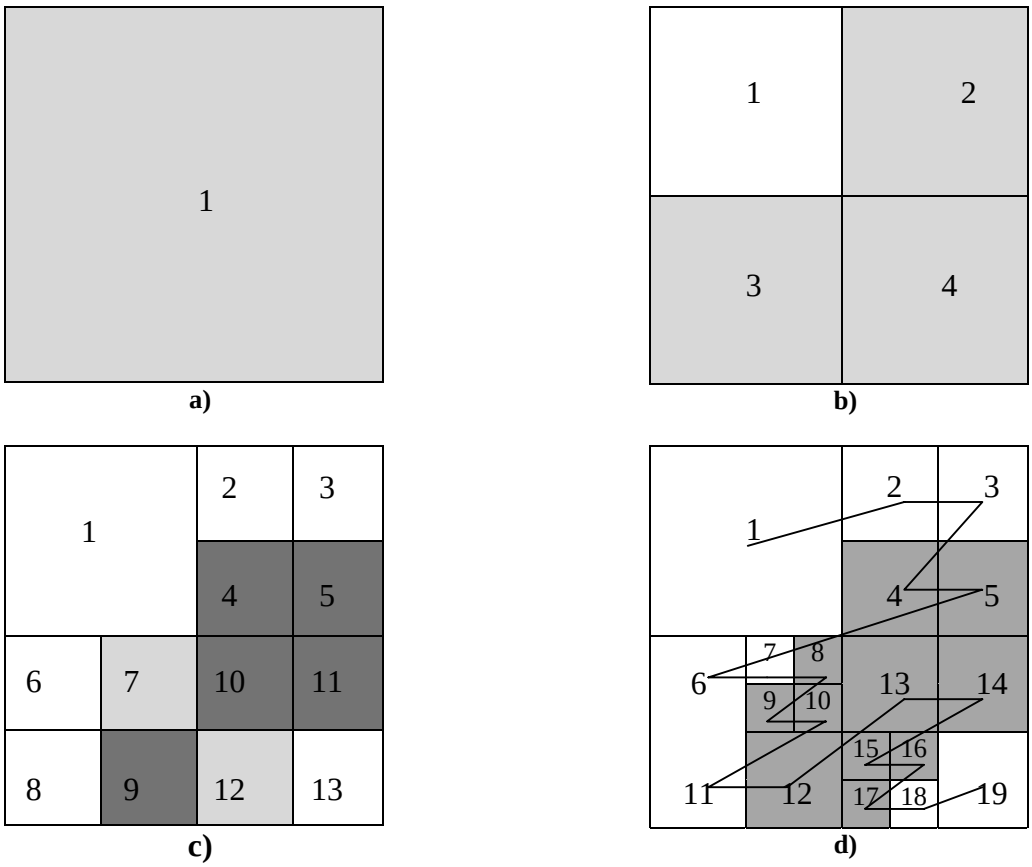


Figure 12 - Principe de décomposition d'un quadtree selon l'ordre de Peano

L'avantage de cette structure est la présence de différents niveaux de l'arbre correspondant à la taille des quadrants donc liés à la résolution. Si l'on coupe l'arbre à un certain niveau, on a une approximation du quadtree final et donc de l'image(*cf* image 13) . Cette propriété est abondamment utilisée dans les algorithmes [Lau89, NP94].

La construction de la pyramide est essentiellement un processus ascendant: le niveau le plus bas de la pyramide représente l'image à pleine résolution. Les niveaux supérieurs sont des représentations à une résolution de plus en plus réduite de l'image initiale.

Si on veut faire une recherche approximative rapide, ou si la zone de recherche est très grande, on peut tout d'abord travailler à basse résolution pour l'image. Si, on trouve plusieurs régions correspondantes, ces régions sont explorées avec une résolution plus grande pour réduire la zone de recherche. La région finalement trouvée est explorée à résolution normale. Cette méthode du grossier au précis (coarse-fine search) est relativement précise (cf. fig. 13).

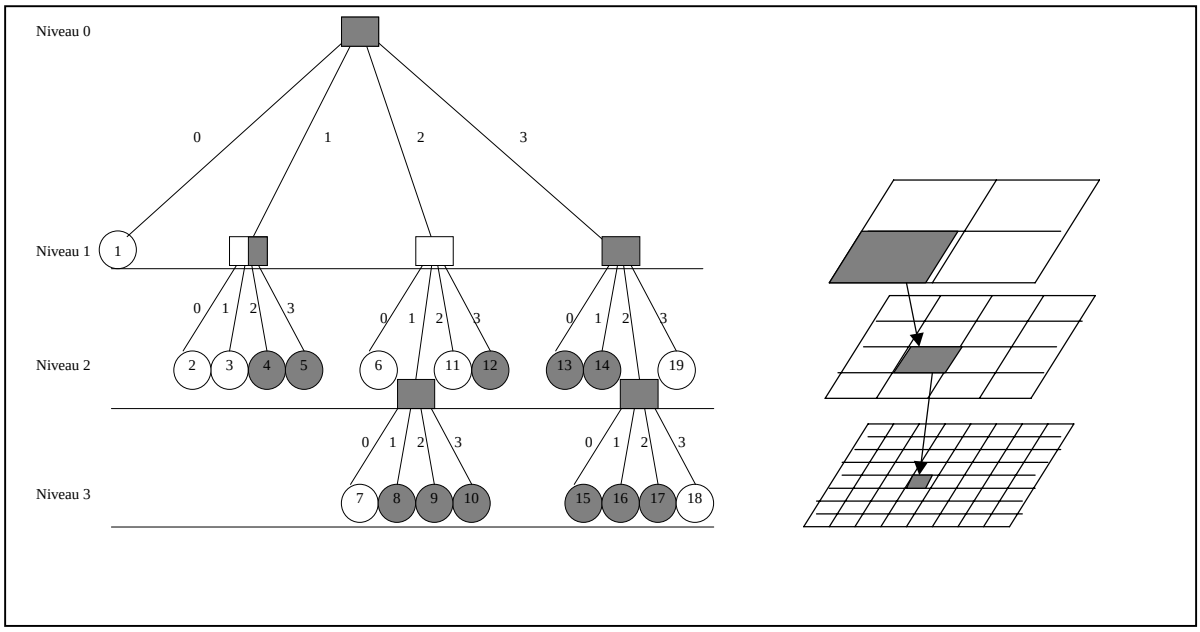


Figure 13 - Arbre quaternaire et structure pyramidale à différents niveaux

Le problème majeur de cette structure provient de la rigidité des divisions réalisées sur l'image, empêchant une bonne adéquation aux données. En particulier, une simple translation de l'image d'un seul pixel perturbe complètement la structure du quadtree.

2. Linéarisation du quadtree

Dans la pratique, la représentation du quadtree sous forme arborescente pose plusieurs problèmes d'implantation. Premièrement, la mise en œuvre directe du découpage récursif serait très coûteuse en temps de calcul, puisque le test d'homogénéité d'un bloc nécessite la consultation de tous les pixels qui le constituent. Le quadtree est construit à partir de la racine, mais chaque pixel est consulté un nombre de fois égale à la profondeur du bloc pour lequel il apparaît dans l'arborescence. De plus la représentation en mémoire implique l'utilisation de pointeurs, qui conduisent à une augmentation prohibitive du volume de codage. Enfin la manipulation de la structure arborescente conduit aussi à utiliser des algorithmes de

rafraîchissement d'arbres, pour les opérations les plus élémentaires, qui sont coûteux en temps de calcul [Coc+95b].

Il est possible de linéariser la structure arborescente en balayant les feuilles de l'arbre dans l'ordre de gauche à droite. La principale caractéristique de cet ordre est d'être lié aux courbes fractales qui remplissent tout un espace, en particulier à la courbe en Z de Peano. La figure 12 montre le parcours pour les blocs de niveaux 1, 2 et 3. La longueur de la chaîne (le nombre de chiffres du chiffre) indique le niveau de l'arbre (profondeur) du bloc correspondant.

Cela conduit, si la décomposition de l'image est faite jusqu'au niveau du pixel, à une courbe de parcours de tous les points du plan, appelée courbe de Peano en Z. On appelle clé de Peano p(x,y) la bijection de N^2 dans N qui associe à tout coordonnées d'entiers (x,y) l'entier obtenu en entrelaçant les bits de x et de y (*cf.* fig.14). Le balayage des quadrants se fait cette fois dans l'ordre des clés de Peano. Ainsi, chaque pixel n'est visité qu'une seule fois et seuls les sommets présents dans le quadtree final sont crées.

La chaîne d'étiquetage (code) d'un carré de niveau k est la concaténation des chaînes associées aux nœuds, à partir de la racine jusqu'au nœud du quadtree qui correspond au bloc de niveau k. Par exemple, la feuille 12 est représentée par la chaîne "23" , donc elle se trouve au niveau 2.

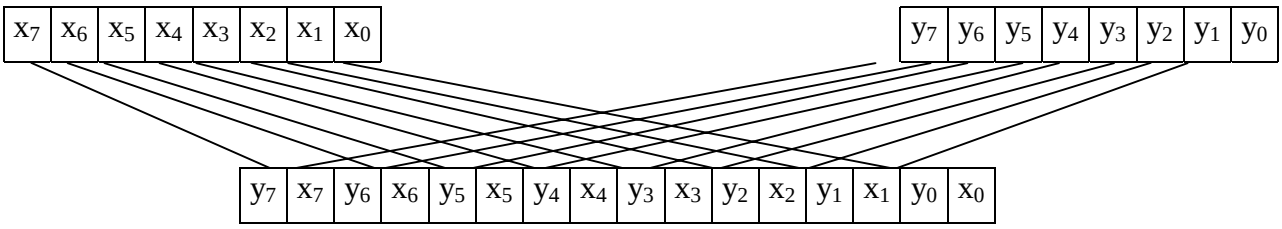


Figure 14 - Clé de Peano: Relation entre coordonnées d'un pixel et son numéro sur la courbe de Peano

L'arbre de segmentation d'une image, de dimension N x N, avec $N=2^L$, possède la structure suivante :

- Chaque nœud de l'arborescence est une région carrée caractérisée par son localisant (clé de Peano associé au son sommet supérieur gauche) de coordonnées [x,y] et par sa taille t.
- Les feuilles pixels sont des blocs de taille t=1, situées à une profondeur $L = \log_2 N$. La racine est à une profondeur 0, sa taille t = N, les coordonnées de son localisant sont [1,1].
- Les nœuds b de l'arbre sont définis de manière récursive:

Un nœud de profondeur k, a une taille $t=N/2^k$, son localisant a pour coordonnées [x,y]. Ses quatre successeurs sont de taille t/2 et les coordonnées de leur localisant sont: [x,y], [x+t/2,y],[x,y+t/2] et [x+t/2,y+t/2]. L'arbre quaternaire construit n'est pas complètement mémorisé. Seul l'ensemble des feuilles est stocké sous forme d'un tableau de taille maximum N x (taille du vecteur d'attributs attaché à chaque bloc). La figure 15 montre le modèle conceptuel d'un tel arbre.

Figure 15 - Modèle conceptuel des quadrants arborescents avec l'ordre de Peano

Bloc	Feuille	(x,y)	Taille	Niveau
4	12	(4,2)	2	2
5	13	(6,2)	2	2
8	211	(3,4)	1	3
9	212	(2,5)	1	3
10	213	(3,5)	1	3
12	23	(2,6)	2	2
13	30	(4,4)	2	2
14	31	(6,4)	2	2
15	320	(4,6)	1	3
16	321	(5,6)	1	3
17	322	(5,7)	1	3

Tableau 2 - Codage du quadtree précédent

On peut déduire du tableau 2 les caractéristiques suivantes:

- Profondeur de l'arbre est la longueur maximale des feuilles: 3
- Niveau = longueur de Feuille
- Taille = Profondeur - Niveau +1

La clé est déduite de la Feuille étendue. Par exemple, la feuille 30 représente toutes les feuilles

$\overbrace{30uuuu}^{\text{profondeur}}$ avec u = 0, 1, 2 ou 3.

Avec u= 0, on obtient la clé 300 = 110000 , après décomposition $x=(100)_2 = (4)_{10}$ et $y= (100)_2 = (4)_{10}$, donc (x,y) = (4,4)

3. Algorithme de base

Le principe simplifié de décomposition, totalement récursif et aisément parallélisable dans un environnement multiprocesseurs, écrit en pseudo-code est le suivant:

```
Séparation(zone)
Début
        Si critère(zone) est vrai
                Alors classer
                sinon diviser zone en 4: Z1, Z2, Z3, Z4
                Pour chaque Zi Faire
                        Séparation(Zi)
                Finpour
        Finsi
Fin
```

4. Compression des données

On distingue deux types fondamentalement différents de compression qui sont:

- La compression conservative qui, après un cycle de compression - décompression, permet une copie exacte de l'original, ce qui est par exemple absolument nécessaire pour les fichiers de base de données, les exécutables, les fichiers binaires, etc.…
- La compression non conservative qui tolère une perte de précision dans les données après un cycle de compression - décompression, ce qui est envisageable pour les images, les fichiers sons, etc.

Thierry Molinier, a fait une étude de l'état de l'art des méthodes de compression des données et a fait une étude comparative en utilisant la décomposition en quadtree avec ces méthodes de compression sur des images fournies par les satellites de télédétection [Mol96]:

- Algorithme de Huffman
- Algorithme de Huffman adaptatif
- Algorithme arithmétique d'ordre n
- Algorithme JPEG (Joint Photographic Experts Group)

Il en conclut que la compression par décomposition en "quadtrees" présente l'avantage, contrairement à beaucoup d'algorithmes embarqués, de compresser très fortement les zones homogènes peu porteuses d'informations utiles, sans dégrader les singularités radiométriques telles que les linéiques et les ponctuels qui représentent très souvent les points caractéristiques

de l'image. D'autre part, son caractère récursif se prête bien à la parallélisation dans un environnement multiprocesseurs.

5. Transformation en Ondelette

L'idée de base de la transformation en ondelettes est décrire l'image par approximations successives à des niveaux de résolution décroissantes. Pour conserver une description complète et non redondante, on extrait à chaque pas l'information perdue au pas précédent. Une image originale est donc décomposée en:

- une image de résolution moitié (suppression d'un pixel sur deux en ligne et en colonne): procédé de décimation
- Trois images filtrées de l'image précédente par un filtre horizontal, vertical et diagonal.

Itération des deux étapes précédentes jusqu'au taux de compression désiré;

6. Notre approche

A l'heure actuelle la majorité de travaux proposés dans la littérature utilisent des signatures globales pour effectuer une recherche par le contenu d'images. La simplicité d'une telle approche ne doit pas cacher ses défauts évidents. C'est le cas où la signature est basée uniquement sur l'histogramme de couleurs. Ainsi une image ayant une même distribution de couleurs mais très différente de l'image requête peut être évaluée comme une réponse pertinente. La figure 16 présente un exemple de fausse détection, où deux images possèdent la même distribution de couleurs , pourtant elles sont très différentes.

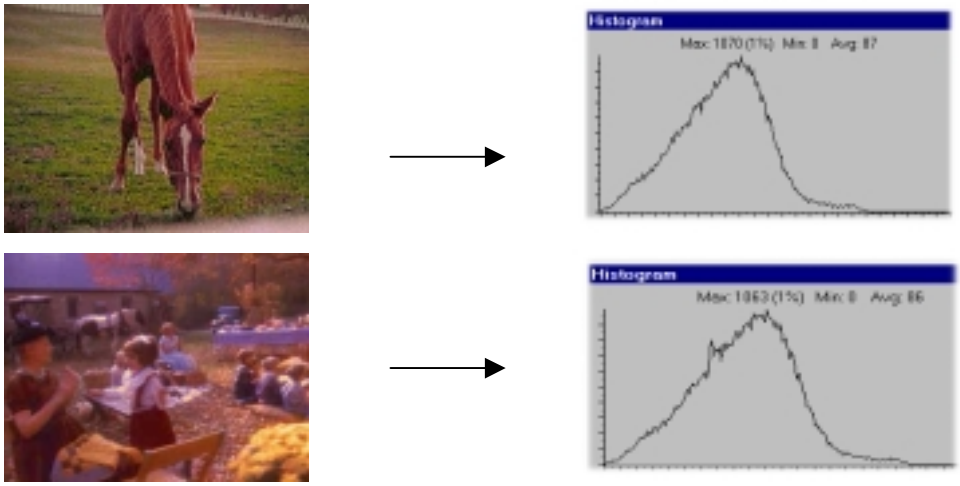


Figure 16 - Un exemple de fausse détection

Quelques travaux récents tentent de remédier un tel défaut en proposant d'associer des informations spatiales aux différentes caractéristiques visuelles. G. Pass et al. [PZ96] proposent de considérer des vecteurs de cohérence de couleur pour prendre en compte la cohérence des couleurs spatialement voisines. Le projet VisualSeek [SC96b] fait un pas en avant et propose d'extraire des régions et caractéristiques spatialement localisées.

Nos travaux [CC97,CC98] proposent d'utiliser les quadtrees [Sam89] pour localiser spatialement les différentes caractéristiques d'une signature visuelle et les clés de Peano[MJF96] pour coder et organiser ces informations spatialisées. Dans un contexte de la recherche d'images par le contenu, les avantages d'une approche basée sur les quadtrees sont multiples. D'abord, elle donne la possibilité d'avoir une approximation de l'image par une représentation de celle-ci en quadtree d'un niveau k largement inférieur à la résolution finale. Ensuite, en segmentant les régions homogènes après la décomposition selon différents critères comme par exemple couleur ou texture, nous constituons aussi une représentation spatiale, symbolique et structurelle des objets contenus dans une image. Une telle représentation de vues multiples d'une même image est souhaitable dans une perspective d'extension de systèmes de recherche d'information (SRI) aux images [BM91].

Critères de décomposition

Dans une telle approche, au moment de son insertion dans la base chaque image est décomposée en quadtree en utilisant l'un ou plusieurs tests d'homogénéité suivants :

Le test radiométrique

Il permet la décomposition de l'image à partir du minimum, et du maximum local du compte numérique des pixels comparés .

Images en niveau de gris:

Min et Max : le minimum et le maximum de la luminance sur le quadrant

Max - Min ≤ seuil1

Figure 17 – Une image du film « Un indien dans la ville »

Prenons à titre d'exemple un pixel origine de radiométrie 124, un test radiométrique fixé à 10 et les deux pixels suivants, dans le sens de la lecture, dont les comptes numériques sont respectivement de 128 et de 117:

La lecture du premier pixel (124) initialisa le minimum et le maximum à 124. La lecture de la radiométrie du pixel suivant (128) détermine le nouveau maximum et la différence maximum - minimum est supérieure à 10 (11 exactement),

Nous réitérons ce processus dans chacun des quatre quadrants.

Images en couleur:

Il s'agit de calculer la similarité entre la couleur avec le taux le plus élevé et la couleur avec le taux le moins élevé:

d(c_max,c_min) ≤ seuil1'

d est une distance de similarité choisie selon l'espace de couleur utilisée.

Le test entropique

Il consiste, dans un premier temps, à calculer l'entropie en octets de la zone à décomposer, de la comparer quantitativement à la valeur choisie, de découper l'image en quadrants si elle a

une valeur supérieure au seuil fixé et de réitérer ce processus dans chacun des quatre quadrants.

E = Entropie ≥ seuil2

Notion d'entropie:

Intéressons-nous au problème suivant: mesurer la quantité d'information apportée par une couleur donnée dans une image. On appelle quantité d'information relative au niveau i, la grandeur suivante:

$Q_i = - \log_2 (p_i)$ où p_i est la probabilité d'apparition du niveau i.

L'entropie d'une image est définie par la quantité moyenne d'information apportée par chaque niveau sur l'ensemble de l'image:

$E = \sum p_i Q_i$

Cette quantité permet de dire quelle est la meilleure image entre deux images représentant les mêmes objets.

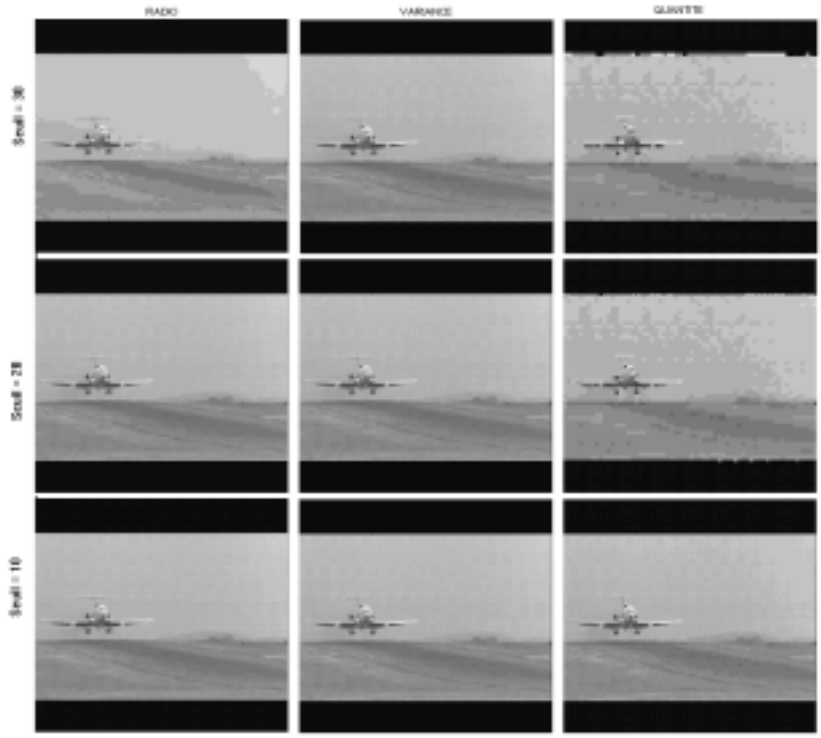


Figure 18 – Reconstitution de l'image précédente après l'avoir décomposée en quadtree selon les tests sur la radiométrie, la variance, et le taux de remplissage, et en utilisant des seuils différents.

Le test statistique basé sur la variance

Le principe du test statistique est exactement le même que le test entropique, si ce n'est que la variable de test est la variance de la zone à décomposer, puis de la même manière la variance de chacun des quatre quadrants. La formule qui exprime le moment centré d'ordre k de la statistique (X_i,N_i) est :

$$\mu_k = \frac{1}{N}\sum_{i=0}^{n} N_i (X_i - \overline{X})^k$$

Avec k =2, nous obtenons la variance (V) , c'est-à-dire le carré de l'écart type.

V ≥ seuil3

Le taux de remplissage

Il s'agit d'un test de remplissage du carré par la couleur dominante du quadrant. En fait, il s'agit du taux de remplissage par la couleur dominante qu'on compare au seuil .

Taux ≤ seuil4

Les carrés qui sont pleins/partiellement pleins sont représentés par la couleur dominante du quadrant. La couleur d'un nœud est la couleur dominante de son quadrant (cf fig.13). La figure 18 montre un exemple d'une image obtenue après décomposition suivant les différents tests.

Critères d'unification

Après la décomposition de l'image par la méthode du quadtree, on obtient un arbre dont les feuilles représentent des portions (carrés) unicolores de l'image. Après, on procède à la constitution de régions, à partir de ces parties (feuilles), suivant deux critères d'unification :

La couleur : Deux parties appartiennent à une même composante si :

- elles ont la même couleur
- leurs couleurs sont proches : c.à.d. la similarité, qui peut être une métrique, entre leurs couleurs est inférieure à un seuil $_c$.

La distance peut être choisie parmi l'une des métriques présentées dans les chapitres précédents. Nous avons choisi, pour notre part d'utiliser l'espace HSV puisqu'il est proche de la perception humaine, en le quantifiant en un ensemble compact de 166 couleurs au lieu de 256. Quant à la similarité entre deux couleurs définies dans l'espace HSV (h_1,s_1,v_1) et (h_2,s_2,v_2) elle est définie par:

$$a_{i,j} = 1 - 1/\sqrt{5[(v_i - v_j)^2 + (s_i \cosh_i + s_j \cosh_j)^2 + (s_i \sinh_i - s_j \sinh_j)^2]}$$

Nous avons choisie cette distance normalisée, proposée par J.R. Smith et Shih-Fu Chang dans [SC95b]. La normalisation par $1/\sqrt{5}$ permet d'avoir des similarités: $a_{i,i}$=1 , et $a_{i,j}$=0 pour des couleurs indexées par i et j qui sont séparées par distance maximale en espace HSV.

La distance: Deux parties appartiennent à une même composante si:

- elles sont connexes (adjacentes)
- la distance qui les sépare est inférieure à un seuil $_d$ = nombre de pixels

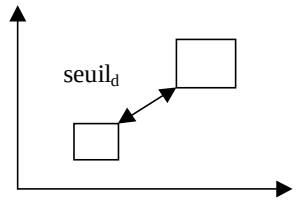


Figure 19 - Distance entre deux quadrants

Le seuil que nous avons choisi est en terme de pixels. Le meilleur choix de distance peut être soit la distance de Manhattan ou la distance de l'Echiquier. Nous avons choisi cette dernière.

Pour cette étape d'union, on étudie tous les couples de régions voisines (X_k,X_i). Si l'union de ces deux régions vérifie le critère d'homogénéité, alors, on fusionne les régions. La principale difficulté de cette approche réside bien sûr dans le parcours de l'ensemble de tous les couples de régions voisines.

L'intégration du critère de la couleur ne pose pas problème lors de la constitution de la région à partir des blocs adjacent. En fait, ici la région est une simple composante connexe qui est constituée de l'ensemble des blocs homogènes de couleurs proches.

Dans la plupart des techniques d'union de quadrants, soit on considère que les blocs constituent une partition initiale pour une future croissance des régions, soit on utilise l'une des techniques de recherche de nœuds adjacents pour constituer une composante connexe initiale. Une telle composante connexe est un ensemble des feuilles de l'arbre (de blocs). Puis en deuxième étape, on doit procéder à agréger les composantes connexes qui sont voisines et qui possèdent les mêmes caractéristiques. Cette dernière approche est gourmande en temps de calcul.

En notant d(p,q) la distance entre deux pixels p et q, la comparaison entre deux composantes connexes nécessite la comparaison de chaque élément d'une composante avec l'ensemble de l'autre selon une distance définie d'une manière classique.

Soient p(x,y) un pixel et C une composante (connexe ou pas) de l'image

- d(p,C) = d(C,p) = min d(p,q) pour tout q pixel de C

Soient A et B deux composantes de I

- d(A,B) = d(B,A) = min d(p,q) pour tout pixel q de A et tout pixel q de B.

Aussi, pour calculer la distance minimale entre deux composantes connexes, il faut calculer toutes les distances qui séparent tous les blocs entre eux. Ce qui est énorme en terme de complexité. Pour remédier à cet inconvénient, nous avons proposé d'intégrer ce critère de distance directement lors de l'assemblage. Pour cela, quand on cherche les quadrants voisins homogènes d'un nœud donné, on ne se contente pas d'examiner seulement les voisins adjacents, mais aussi les voisins qui ne le sont pas.

En fait, pour un quadrant courant, un quadrant voisin homogène est soit un nœud adjacent homogène, soit un nœud homogène qui se trouve à une distance inférieure au $seuil_d$.

Soit H(q) l'ensemble des voisins adjacents et homogènes d'un quadrant q, et $\overline{H}(q)$ l'ensemble des voisins adjacents non homogènes de q. On définit le voisinage proche $V_P(q)$ tel que:

$V_P(q) = \{x \in H(q) \text{ ou } y \in H(u) / u \in \overline{H}(q) \text{ et } d(y,q) \leq seuil_d\}$

V contient l'ensemble de tous les voisins de q situés à une distance inférieure à seuil.

L'algorithme correspondant est le suivant:

```
Voisinage ( q , seuil, V )
Début
        V = Vois_adj_hom(q);
        Pour chaque élément x de Vois_adj_non_hom(q) Faire
```

```
                T = taille(x);
                Si T < seuil Alors
                        V = V ∪ Vois_adj_hom(x);
                        Pour chaque élément y de Vois_adj_non_hom(x) Faire
                                Voisinage(y, seuil - T, V)
                        Fin
                Fin
        Fin
Fin
```

Les algorithmes de recherche de voisins adjacents sont présentés dans le même chapitre.

Critère de confiance

Etant donné une image originale dont le niveau d'un pixel (x,y) est noté I(x,y) et une image obtenue après décomposition dont le niveau d'un pixel (x,y) est noté I'(x,y), on peut définir des critères objectifs permettant de juger la qualité de la décomposition basée sur l'erreur quadratique moyenne ou le rapport signal sur bruit, PSNR (Peak Signal to Noise Ratio),de la théorie du traitement du signal. Pour le calculer et afin de permettre la comparaison de l'image origine et de l'image après décomposition dans de bonnes conditions, indépendantes de l'aspect très subjectif de la visualisation par un expert, nous calculons tout d'abord deux variables intermédiaires que sont MSE (Mean Square Error) et le RMS (Root Mean Square) données par les formules suivantes:

$$MSE(I,I') = \frac{1}{N \times N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} [I'(x,y) - I(x,y)]^2$$

$$RMS(I,I') = \sqrt{MSE(I,I')}$$

$$PSNR(I,I') = \sqrt{\frac{\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} I'^2(x,y)}{N \times N * RMS(I,I')}}$$

où l'image est prise ici de dimension N x N.

Proximité et Voisinage dans les quadtrees

La notion de voisinage nécessaire pour traiter une image, peut aussi être mise en œuvre dans un quadtree. Il existe, selon la méthode d'implantation utilisée pour coder un quadtree, divers algorithmes permettant d'accéder aux nœuds de l'arbre qui lui sont voisins dans l'image[Sam82].

Soient deux nœuds P et Q qui ne se chevauchent pas, et une direction D, on définit un prédicat d'adjacence tel que adjacent(P,Q,D) est VRAI s'il existe deux pixels p et q, appartenant respectivement à P et Q, tel que:

Si q est adjacent à un côté (horizontal ou vertical) de p dans la direction D: on parle de voisins de côté. Si q est adjacent à un coin de p dans la direction D: on parle de voisins diagonaux. On dit que les nœuds P et Q sont voisins s'ils sont adjacents.

Par exemple: les nœuds 6 et 9, dans la figure 13, sont des voisins de côté car 6 se trouve à l'Ouest de 9. Alors que les nœuds 8 et 4 sont des voisins diagonaux car 4 se trouve au Nord-Est de 8. Deux blocs peuvent être à la fois voisins de côté et voisins diagonaux. Par exemple: 1 est à la fois au Nord et au Nord-Est de 6.

Algorithmes associés au quadtrees

On présente ici les algorithmes concernant l'extraction de composantes homogènes. Pour cela, nous présentons les algorithmes de recherche de voisins.

Constitution de régions homogènes à partir d'un quadtree linéaire

```
FONCTION Composante_Connexe :
/*      Recherche d'une composante connexe :
        -1- On commence par prendre un bloc quelconque de l'image
        -2- On le met dans l'ensemble Afaire des carrés a traiter
        -3- On prend un point de l'ensemble Afaire
        -4- On cherche les voisins de ce bloc avec les caractéristiques données
        -5- On ajoute ces voisins dans l'ensemble Afaire
        -6- On recommence a partir de l'étape –3- tant que l'ensemble Afaire n'est pas vide
        PARAMETRES :
        courant : noeud courant
        prof  : profondeur de l'arbre                                                   */
Ensemble Composante_connexe(courant, prof)
Début
        Ensemble Composante, Afaire , Tamp;
        Si (courant est non nul)et (Afaire n'est pas vide) Alors
        Début
        /* Cette fonction vérifie si courant est un quadrant de couleur proche et s'il ne fait pas
        partie d'une autre composante (n'a pas été intégré) */
        verif = Verifier(courant);
                Si  (verif =FAUX) Alors Afaire->AjouteElement(courant);
                Tant que ( Afaire est non vide) Faire
                Début
                        Tant que (Verifier (courant) =VRAI  ) et (Afaire n'est pas vide) Faire
                        Début
                                courant = Afaire->RetirePremierElement();
                        Fin
```

```
                    Composante ->AjouteElement(courant);
                    Si (longueur(courant)= prof) Alors   Tamp=Voisins(courant);
                    Sinon          Tamp = VoisinsQuelconque(courant,prof);
                    Finsi
                    Union(AFaire,Tamp);  Marquer courant intégré;
            Fintantque
        Finsi
        Retourner(Composante);
Fin
```

FONCTION Voisins :

```
/*      Recherche des voisins adjacents du carré courant, ce dernier étant obligatoirement une feuille
        de profondeur égale à celle de l'arbre. Il suffit de chercher les voisins dans les 8 directions(NO,
        N, ...)
        PARAMETRES :
            carre : noeud courant
            voisin : au retour, contient les voisins du carré courant                    */
Ensemble Voisins(carre )
Début
        Ensemble Vois;
        Pour i parmi {N,O,E,S,NO,NE,SO,SE } Faire
        début
            chaine = FONC( i, carre, longueur(carre)-1));
            Si chaine contient  '0' Alors Vois -> AjouteElement(Feuille(chaine);
        Fin
        Retour(Vois);
End
```

FONCTION FONC :

```
/*      Elle retourne le bloc se trouvant à la direction D du carré  courant
        PARAMETRES :
            d: une direction donnée parmi les 8 directions (NO,N,...)
            carre : nœud courant
            index  : variable interne                                                    */
Chaine FONC (D, carre ,index)
Début
```

```
        Si index = 0 Alors      carre[0]='0';
        Sinon
                T = carre[index];          carre[index] = Renvoie (D,T);
                C = Côté_Commun (D,T);
                Si (C) Alors carre = FONC( C, carre, index-1);
                Sinon   Si Adjacent (D,T) =VRAI Alors carre= FONC(D, carre, index-1);
                Finsi
        Finsi
        Retour(carre);
Fin
```

FONCTION VoisinsQuelconque :

```
/*      Recherche des voisins adjacents du carré courant, ce dernier étant une feuille de profondeur
        inférieure à celle de l'arbre. On commence par chercher les voisins pour chaque direction
        principale(N, O, E, S) puis chaque voisin dans les directions NO, NE, SO, SE.
         PARAMETRES :
                carre : carré courant
                voisin : au retour, contient les voisins du carre courant
                prof   : profondeur de l'arbre                                          */
Ensemble VoisinsQuelconque(carre , voisin, prof)
Début
        /* • est une opération de concaténation */
        buf = carre;
        Pour i=0 à prof - longueur(carre) Faire buf = buf • 'NO';
        Pour D parmi {N,O,E,S} Faire
                Voisin = Direction(longueur(carre) , buf, prof-1, D);
        FinPour

        Pour D parmi { NO,NE,SO,SE } Faire
                buf = carre;
                Pour i=0 à prof- longueur(carre) Faire buf = buf • D;
                chaine = buf;
                FONC(D , chaine, longueur (buf)-1);
                Voisin -> AjouteElement(Feuille(chaine);
        FinPour
Fin
```

FONCTION Direction

```
/*      Recherche des voisins du carre courant dans une direction donnée
        PARAMETRES :
                i   : profondeur du carre
                buf : variable interne
                prof: profondeur de l'arbre - 1
                D   : direction de recherche de voisins parmi N, O, E et S
*/
Ensemble Direction(i, buf, prof, D) )
Début
        Ensemble Voisin;

        buf[i]= Tab_Etiq[D][0];
        Pour j =1 à prof-i Faire buf[j+i]='NO';        buf[prof+1]=NULL;

        Si (i < prof) Alors     Direction(i+1, buf, prof, D);
        Sinon
                chaine = buf;
                chaine = FONC(D, chaine, longueur(chaine)-1);
                Voisin -> AjouteElement(Feuille(chaine);
        Fsi

        buf[i]= Tab_Etiq[D][1];
        Pour j=1 à prof-i Faire buf[j+i]='NO';         buf[prof+1]=NULL;

        Si (i < prof) Alors     Direction(i+1, buf, prof, D);
        Sinon
                chaine = buf;
                chaine = FONC(D, chaine, longueur(chaine)-1);
                Voisin -> AjouteElement(Feuille(chaine);
        Fsi
        Retour Voisin
Fin
```

L'étiquette qui correspond à la racine est initialisée à 'NO'

0 étant réservé à la racine, NO = 1, NE = 2, SO = 3, SE = 4, N = 5, O = 6, E = 7 et S = 8

Tab_Etiq est un tableau indiquant la direction choisie, il est initialisé par les éléments suivants :

Nord = Tab_Etiq[0][2] = {'NO','NE'} ={'1', '2'};

Ouest = Tab_Etiq[1][2] = {'NO','SO'} ={'1', '3'};

Est = Tab_Etiq[2][2] = {'NE','SE'} ={'2', '4'};

Sud = Tab_Etiq[3][2] = {'SO','SE'} ={'3', '4'};

Adjacent (I,P) est vrai si et seulement si le quadrant P est adjacent à un côté ou coin dans la direction I, faux sinon. Par exemple, Adjacent('O','SO') est vrai.

Renvoi(I,P) retourne le résultat de *Type* du quadrant de taille égale (pas nécessairement un quadrant frère) qui partage un côté ou coin dans la direction I.

Par exemple, Renvoie('N','SO') = 'NO';

La fonction *Feuille(P)* retourne la feuille correspondant au quadrant. Par exemple, selon notre implémentation:

Feuille("1")="";

Feuille("11")="11";

Feuille("11") = Feuille("112") = Feuille("112343 = ….= "11";

Côté_Commun(J,P) retourne le côté qui est commun entre le quadrant P et ses voisins dans la direction du coin J. Par exemple, Côté_Commun('SO','NO')='O';

Les fonctions Adjacent, Renvoi et Côté_Commun sont présentés ci dessous sous forme de tableaux :

Table: Adjacent (I,Q)

	Quadrant Q			
Direction I	**NO**	**NE**	**SO**	**SE**
N	V	V	F	F
E	F	V	F	V
S	F	F	V	V
O	V	F	V	F
NO	V	F	F	F
NE	F	V	F	F
SO	F	F	V	F
SE	F	F	F	V

Table: Renvoie (I,Q)

Direction I	NO	NE	SO	SE
N	SO	SE	NO	NE
E	NE	NO	SE	SO
S	SO	SE	NO	NE
O	NE	NO	SE	SO
NO	SE	SO	NE	NO
NE	SE	SO	NE	NO
SO	SE	SO	NE	NO
SE	SE	SO	NE	NO

Table: Côté_Commun (I,Q)

Direction I	Quadrant Q			
	NO	**NE**	**SO**	**SE**
NO	-	N	O	-
NE	N	-	-	E
SO	O	-	-	S
SE	-	E	S	-

Avec, notre approche, la manipulation des quadrants et la recherche de leurs voisins s'effectuent tout simplement en manipulant les chaînes de caractères.

C. Croissance de région

La plupart des travaux qui présentent la segmentation par recherche de région effectuent la segmentation en étapes successives. Ils définissent d'abord une partition de l'image en régions de base, en fonction d'une ou plusieurs hypothèses ou connaissances a priori, puis modifient successivement ces régions.

Il existe deux façons d'effectuer une partition initiale de l'image avant de lancer le processus de détection de régions par fusion :

- chaque pixel de l'image initiale constitue une région,
- on essaie de constituer dans l'image initiale des petites régions de base bien homogènes puisque le processus de fusion s'interdit de pouvoir les diviser ultérieurement. Par exemple, elle peut être obtenue, soit par un balayage séquentiel soit à partir d'une segmentation obtenue, à partir d'une exploitation d'un arbre quaternaire.

Une partition initiale est appelée un germe. La méthode la plus classique repose sur un balayage séquentiel de l'image et de lancer la croissance de régions par agrégation des pixels adjacents.

1. Formation par agrégation de pixels

On utilise la notion de chemin par la distance d et on définit une relation de connexité par le d-chemin. La connexité par le d-chemin entre deux pixels est conditionnée par la définition du "voisinage" d'un pixel.

Soient deux points $p(x_p, y_p)$ et $q(x_q, y_q)$ des pixels dans une image. La distance métrique entre p et q est définie par :

$d(p,q) = Max \{| x_p - x_q | , | y_p - y_q |\};$

Soit ρ un entier. On appelle voisinage d'ordre ρ du pixel P et l'on note $V_\rho(P)$ l'ensemble des pixels q situés à ρ pixels de P et définit par:

$V_\rho (p)=\{ \forall q / d(p, q)\leq \rho \}$

On définit dans l'image I un chemin de longueur n et d'ordre ρ (n-chemin d'ordre ρ) comme $\{a_1,\ldots..a_n\}\subset I$ avec $d(a_i,a_{i+1})\leq \rho$

p_{36}	p_{35}	p_{34}	p_{33}	p_{32}	p_{31}	p_{30}
p_{37}	p_{16}	p_{15}	p_{14}	p_{13}	p_{12}	p_{29}
p_{38}	p_{17}	p_4	p_3	p_2	p_{11}	p_{28}
p_{39}	p_{18}	p_5	•p	p_1	p_{10}	p_{27}
p_{40}	p_{19}	p_6	p_7	p_8	p_9	p_{26}
p_{41}	p_{20}	p_{21}	p_{22}	p_{23}	p_{24}	p_{25}
p_{42}	p_{43}	p_{44}	p_{45}	p_{46}	p_{47}	p_{48}

Figure 20 – Parcours des N-voisins d'un pixel p

Les voisins d'ordre 1 d'un pixel p sont les 8 voisins directs. Pour les voisins indirects (ρ>1) , selon la figure 20, on a :

- $v_1(p)=\{p_1,p_2,p_3,p_4,p_5,p_6,p_7,p_8\}$, $card(v_1(p))= 8$;
- $v_2(p)= v_1(p)\cup\{p_9,p_{10},\ldots p_{24}\}$, $card(v_2(p)) = 24$;
- $v_3(p)= v_2(p)(\{p_{25},p_{26},...p_{48}\}$, $card(v_3(p)) = 48$

Généralement , pour ρ = n, on obtient :

- $v_n(p)= v_{n-1}(p)(\{p_x,p_{x+1},...p_{\Sigma 8*i}\}$, et le cardinal associé est : $\sum_{i=1}^{n} 8*i$

2. Principe

Le principe consiste à regrouper les pixels de l'image, qui partagent les mêmes caractéristiques, en une même région. Chaque région satisfait un prédicat indicateur d'homogénéité de la couleur. On part d'un pixel d'une couleur donnée et on joint à la liste tous les voisins de même classe de couleur (couleurs proches). Le processus continue ainsi pour tous les pixels non visités de l'image.

- Si les images sont en niveau de gris, on considère qu'un pixel p "ressemble" à une composante R si et seulement si:

voisin(p,C) **et** min(C) - ε1 ≤ niveau_gris(p) ≤ max(C) + ε1 **et** niveau_gris(p) = moy(C) ± ε2

où

- moy(C) est la moyenne des niveaux de gris de la population de la région R.

- min(C) et max (C) sont les bornes de l'intervalle de variation de la dynamique, c'est à dire, la plus petite et la plus grande valeur en niveau de gris de l'ensemble des pixels formant la région.

- niveau_gris(p) représente le niveau de gris du point p

ε1 et ε2 sont des seuils qui représentent des valeurs de tolérance

- voisin(p,C) est vrai s'il existe un point de la composante C qui appartient au voisinage de p.

- Si les images sont en couleur, on considère qu'un pixel p "ressemble" à une composante C si et seulement si:

voisin(p,C) et d(p,moy(C)) ≤ ε3

où

- moy(C) = (R_M,G_M,B_M) est la moyenne des couleurs (de chaque composante) de la population de la région C, et ε3 est un seuil qui représente le seuil de similarité des valeurs de tolérance

- voisin(p,C) est vrai s'il existe un point de la composante C qui appartient au voisinage de p.

- d est une distance qui permet de calculer la similarité entre deux couleurs **(§xx).**

La méthode d'agrégation de pixels, que nous avons proposé, part d'un ensemble de pixels de départ, les regroupe selon le double critère d'homogénéité et de voisinage. Partant du constat que seules les composantes dominantes capturent l'attention humaine, nous avons opté pour une segmentation de l'image orientée objets dominants. Pour cela, on peut utiliser l'heuristique suivante :

« Une région dominante a de fort probabilité d'être composée de couleurs dominantes » , surtout qu'on on utilise un voisinage d'ordre supérieur à 1 (ρ =3) qui permet de regrouper des pixels dispersés qui est souvent le cas dans les images texturées.

Autrement dit, si on parcourt l'histogramme dans l'ordre de ses distributions maximales, on a plus de chance d'extraire des régions dominantes.

La procédure de croissance de région va être exécutée à partir des pixels de couleur dominante, puis ainsi de suite dans un ordre décroissant jusqu'à satisfaction d'un seuil de représentation. D'autre part, puisque le but de notre segmentation est l'extraction de caractéristiques visuelles représentatives, nous avons utilisé plusieurs paramètres qui permettent le réglage et l'affinage de la segmentation.

- ρ : Seuil sur l'ordre de voisinage de balayage des pixels.
- μ: Seuil d'homogénéité de la couleur, il s'agit d'un seuil qui est utilisé lors de la constitution des classes de couleur.
- λ : Seuil de cohérence d'une région homogène: une région homogène est ignorée si elle n'est pas représentative, c.a.d si sa taille est en dessous d'un seuil donné, Une région homogène n'est considérée comme cohérente qu'à la condition de représenter plus de λ % de l'image

- θ: Le nombre de régions dominantes à extraire. Une image contient au plus θ régions cohérentes.
- Un indicateur si oui ou non, il faut considérer le fond de l'image comme une région homogène. On considère que les régions en question, sont des régions dominantes dont les contours touchent les bords de l'image au delà d'un seuil .

Le sens d'examen des pixels candidats à une fusion est important. Par exemple, soient trois régions A, B et C deux à deux adjacentes. Les fusions de A avec B et de B avec C peuvent être correctes, alors que les fusions de C avec A∪B ou de A avec B∪C ne le sont pas. Le critère de "ressemblance" utilisé pour la fusion est rarement transitif: A "ressemble" à B et B" ressemble" à C n'entraîne pas A "ressemble" à C.

Pour s'affranchir du problème du choix des points de départ, nous avons procédé à un clustering des couleurs, en utilisant une palette unique de 512 couleurs. Pour cela, on détermine la cross-corrélation entre les couleurs en vue de calculer la similarité de perception entre elles. Cela nous a permis de réduire le nombre de couleurs, de diminuer la complexité de calcul, d'utiliser les représentants des classes de couleurs comme des étiquettes et par conséquent faciliter le processus de segmentation et d'extraction d'objets cohérents et homogènes. Ainsi, le résultat de la segmentation ne dépendra plus du pixel de départ, mais seulement des critères de segmentation tels que l'ordre de voisinage et les seuils de détermination des classes de couleurs. La coalescence des couleurs s'effectue en calculant les distances euclidiennes entre les couleurs.

Classification des couleurs

Pour chaque paire de couleur C_i et C_j on calcule $d(C_i,C_j)$, on trie les résultats par ordre croissant. Ainsi, pour chaque couleur on détermine les couleurs qui lui sont proches. A la fin, on obtient des classes de couleurs dont chacune sera représentée par une couleur représentante. Pour l'affichage, on choisit la couleur dominante comme la couleur élue.

Une classe de couleur K comporte toutes les couleurs proches de la couleur représentante. On peut avoir plusieurs configurations possibles pour la constitution des classes. On a choisit de privilégier les grandes classes plutôt que des petites classes.

Supposons que nous partions du tableau suivant des distances entre les couleurs d'une palette:

Palette	***1***	***2***	***3***	***4***	***5***	***6***	***7***
1	0						
2	11	0					
3	13	12	0				
4	16	15	13	0			
5	17	16	14	11	0		
6	21	20	18	15	14	0	
7	26	25	23	20	19	15	0

On représente, pour chaque couleur, le vecteur de ses couleurs proches avec les distances correspondantes pour un éventuel seuillage. Les valeurs soulignées sont les représentants des classes. On a choisi d'utiliser la stratégie "max " . Les agrégations successives suivant les seuils conduisent aux configurations suivantes :

Seuil = 11 :

	{1,2}	***3***	***4***	***5***	***6***	***7***
{1,2}	0					
3	13	0				
4	16	13	0			
5	17	14	11	0		
6	21	18	15	14	0	
7	26	23	20	19	15	0

	{1,2}	***3***	***{4,5}***	***6***	***7***
{1,2}	0				
3	13	0			
{4,5}	17	14	0		
6	21	18	15	0	
7	26	23	20	15	0

On obtient les classes suivantes : {{1,2},{3}, {4,5} {6},{7}}

Seuil = 13 :

	{1,2,3}	***{4,5}***	***6***	***7***
{1,2,3}	0			
{4,5}	17	0		
6	21	15	0	
7	26	20	15	0

Seuil = 15 :

On a le choix entre la fusion {4,5,6} et la fusion {6,7}.

	{1,2,3}	***{4,5,6}***	***7***
{1,2,3}	0		
{4,5,6}	21	0	
7	26	20	0

ou

	{1,2,3}	***{4,5}***	***{6,7}***
{1,2,3}	0		
{4,5}	17	0	
{6,7}	26	20	0

Seuil = 20 : ↓

	{1,2,3}	***{4,5,6,7}***
{1,2,3}	0	
{4,5,6,7}	26	0

ou

Seuil = 17 : ↓

	{1,2,3,4,5}	***{6,7}***
{1,2,3,4,5}	0	
{6,7}	26	0

Nous avons privilégié les grandes classes par rapport aux petites. En effet, pour cet exemple, on choisit la configuration suivante :

{{1,2,3,4,5},{6,7}

On peut représenter le processus de constitution des classes sous forme arborescence. L'arbre obtenu est présenté dans la figure suivante.:

En conséquence, la méthode "max " permet d'éviter de constituer des classes comprenant des

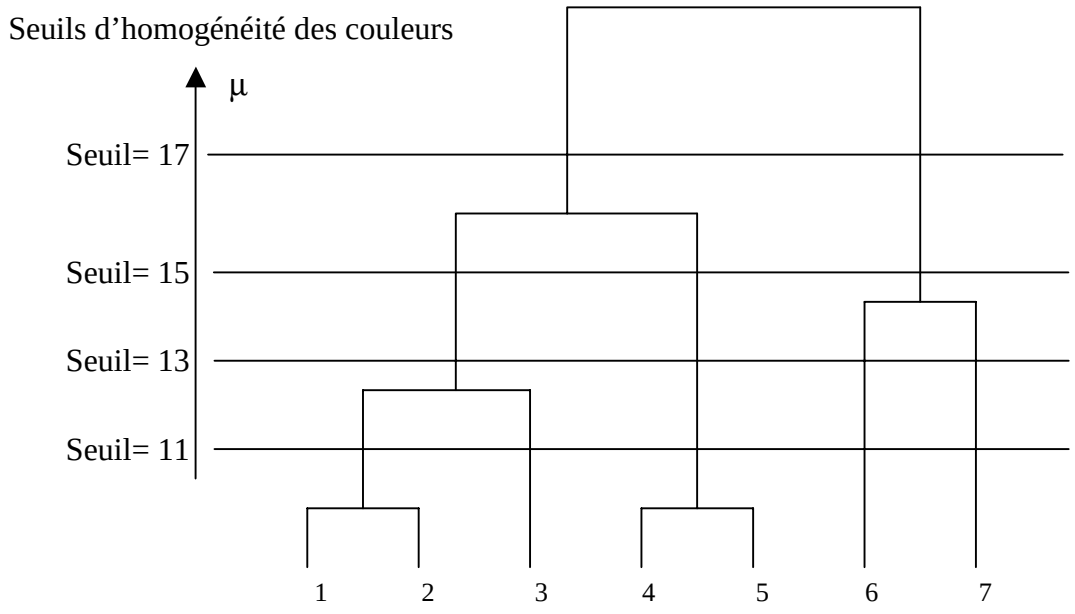


Figure 21 - Hiérarchie de classes de couleurs

éléments très éloignés les uns des autres; elle réduit le "diamètre" des classes.

3. Algorithme

Un objet homogène est une région homogène selon les deux critères de voisinage des pixels et de la couleur. La procédure de croissance des régions consiste à assembler tous les pixels voisins selon les deux critères de voisinage des pixels et de la similarité des couleurs. En effet, le processus commence avec un pixel de départ et essaie d'attirer tous les pixels voisins à cette région qui s'agrandit au fur et à mesure. La région continue à croître jusqu'à ce qu'il n'y ait plus de pixels qui puissent satisfaire aux critères d'homogénéité citées précédemment.

Function Croissance_Region( )

Entrée:

P : un point de départ dans l'image I;

ρ: seuil de voisinage en nombre de pixels {1, 2, ou 3 };

μ : seuil d'homogénéité de la couleur;

Sortie :

La région cohérente R contenant le pixel de départ P;

Initialisation :

Empiler le pixel P dans VCN ;

{Coordonnées du rectangle circonscrit RME}

xi = xs = P.x; yi = ys = P.y;

{Coordonnées du centre de gravité }

sx_centroid = 0; sy_centroid = 0;

nb_pixels = 0;

{Couleur de la région pour l'affichage}

Couleur_R = couleur_de_P ;

{Nombre de pixels pour chaque quadrant et nombre total}

nb_pixel1 = nb_pixel2 = nb_pixel3 = nb_pixel4 = nb_pixels =0

Etape 1: {Un nouveau pixel cohérent est ajouté à la région }

Dépiler un pixel Q de la pile VCN;

Insérer Q dans l'ensemble R;

Marquer le pixel Q comme intégré;

{Mise à jour du nombre de pixels pour chacun des 4 quadrants}

Si Q $\in$ quadrant 1 alors nb_pixel1 ++;

Si Q $\in$ quadrant 2 alors nb_pixel2 ++;

Si Q $\in$ quadrant 3 alors nb_pixel3 ++;

Si Q $\in$ quadrant 4 alors nb_pixel4 ++;

{Mise à jour des coordonnées du RME }

Si (Q.x < xi) alors xi = Q.x; Si (Q.y < yi) alors yi = Q.y;

Si (Q.x > xs) alors xs = Q.x; Si (Q.y > ys) alors ys = Q.y;

{accumulation pour le calcul du barycentre}

sx_centroid += Q.x; sy_centroid += Q.y;

{Couleur_R est la moyenne des couleurs de R}

Couleur_R = moyenne (R)

Etape 2 : {attraction des pixels voisins qui sont candidats}

Pour chaque pixel T voisin de Q, situé dans le périmètre ρ

{couleurs de Q et de T sont proches suivant μ}

Si T n'est pas encore intégré et d_E ($couleur_Q$, $couleur_T$) $\leq \mu$ alors empiler T dans VCN ;

Etape 3 : {boucle}

Répéter les étapes 1 et 2 jusqu'à ce que la pile VCN soit vide;

Etape 4 : {Objet homogène O extrait }

```
O.region = R;
{RME de la région }
O.xi = xi ;O.yi=yi; O.xs=xs; O.ys=ys;
{Couleur représentative de la couleur }
O.couleur = couleur_R;
{calcul de la surface de la région }
nb_pixels = nb_pixel1 + nb_pixel2 + nb_pixel3 + nb_pixel4;
{calcul du barycentre}
O.x_centroid = sx_centroid /nb_pixels;
O.y_centroid = sy_centroid /nb_pixels;
```

Pseudo algorithme de la croissance de région

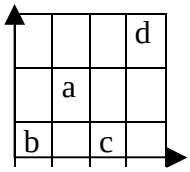
Cette fonction décrit le processus de la croissance de région par agrégation des pixels voisins dans un périmètre ρ et retourne ses caractéristiques. Le processus démarre avec un pixel p = (x, y) et continue à croître dans l'image jusqu'à ce qu'il n'y ait plus de pixels qui puissent être intégrés à la région. Ce processus d'agrégation utilise une pile VCN des pixels voisins candidats non visités. Le processus continue tant que cette pile n'est pas vide. Pour extraire toutes les régions homogènes dans l'image, on répète ce processus pour tous les pixels qui ne sont pas intégrés dans l'image.

La structure suivante caractérise les attributs d'une région cohérente: un identifiant de la région, sa couleur représentative, son rectangle circonscrit (RME), son barycentre, et l'ensemble des pixels qui composent la région homogène. Pour cela, nous avons utilisé les bibliothèques standards des modèles (STL, Standard Template Library)

```
struct pixel {
bool flag; /*intégration du pixel */
     int color;  /*Couleur du pixel*/
     int x,y;  /*Coordonnées du pixell*/
};
typedef vector <pixel> VectorOfPixel;
Struct OBJECT {
     /*Identifiant de l'objet */
     int num_reg;;
     int couleur, classe;
     /* Coordonnées du RME */
     int xi, xs, yi, ys;
     /* barycentre */
     int x_centroid,y_centroid;
     /* Ensemble des pixels */
     VectorOfPixel Region;
}
```

D. Arrangement spatial

Supposons qu'une image est constituée d'un ensemble E de régions repérées par des clés de Peano ou d'Hilbert E={a,b,c,d,e,f}



Il y a deux manières pour décrire l'arrangement spatial entre ces points: les chaînes 2D [Cos+92] et les relations spatiales simples entre les paires de points.

Représentation par les 2D-Strings:

Elles représentent un objet par des points qui sont projetés respectivement sur l'axe des x et des y. La chaîne 2-D qui représente l'ensemble E correspondant à l'image est:

Vertical gauche→droite

Horizontal bas→haut

(Vertical ,Horizontal)= (b < f a < c < e d , f e < b c < a < d)

Cette chaîne 2-D (Vertical ,Horizontal) peut représenter symboliquement la disposition des coordonnées des objets dans une image.

Représentation par des relations spatiales[LH92]

Si on considère les relations suivantes :

→ vers la droite; ≡0 en code de Freeman :

← vers la gauche; ≡ 4 en code de Freeman

↑ vers le haut; ≡ 2 en code de Freeman

↓ vers le bas, ≡ 6 en code de Freeman

On peut représenter les relations spatiales entre les régions par :

a →→ ↑d ou a ↑→→d;

b →↓ f; b →↑a ; a →↓c; f →↑c; c → ↑↑d ; c →↓e;

e ←←←↑b; etc…

L'inconvénient de ce type de représentation est le fait qu'elle ne fournit pas une signature unique, contrairement à la chaîne 2-D. Pour rendre ce type de représentation unique, il faut définir un ordre de priorité pour le parcours ou établir des règles. La procédure suivante définit un modèle de représentation unique mais qui reste sensible aux petits changements des positions.

Par exemple, la relation spatiale entre le couple (a,e) est définie par (2→,2↓). De même pour (b,f) = (→,↓), (a,b) = (←,↓), et

(d,b) = (3←,2↓).

Relation_Spatiale(u,v)

Soit $u(u_x, u_y)$ et $v(v_x, v_y)$

$r_x = u_x - v_x$; $r_y = u_y - v_y$,

Si $r_x < 0$ alors il y a r_x → entre u et v;

Sinon il y a r_x ← entre u et v;

1. Structure de l’arbre-R

L'ensemble des RMEs est indexé en utilisant la structure dynamique R-tree. La structure R-tree que nous avons utilisé est une variante de R-tree proposée par Kamel et Faloutsos [KF94], où nous avons utilisé les clés de Peano à la place de Hilbert. La figure 22 illustre un exemple d'organisation des rectangles en arbre-R de Peano et sa structure interne.

La principale innovation apportée dans les arbres-R est que les nœuds pères peuvent se chevaucher. Avec cela, ils garantissent au moins 50% de l'utilisation de l'espace. La performance des arbres-R dépend du type de groupement des rectangles de données en un nœud. On utilise les fractales, en particulier les courbes d'Hilbert ou de Peano, pour imposer un agencement linéaire des rectangles de données.

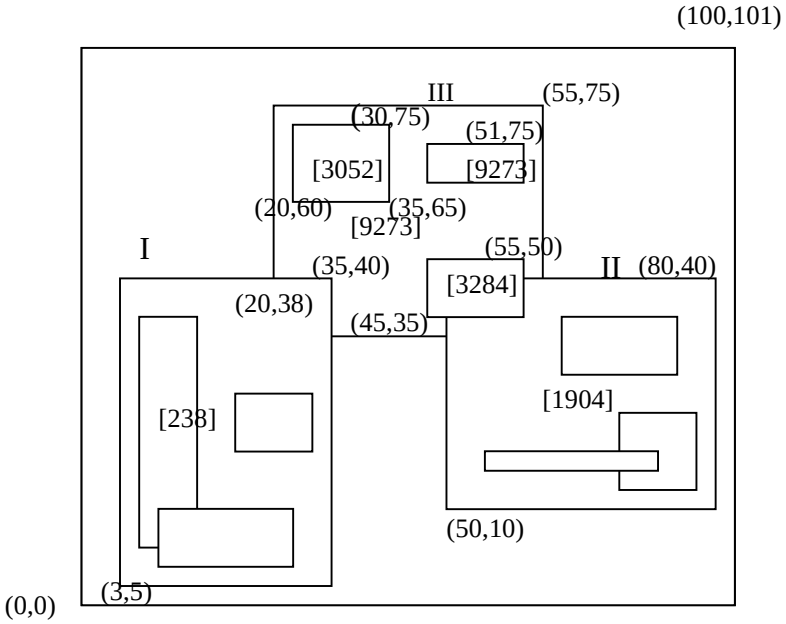
La valeur de Peano d'un rectangle est définie comme la valeur de Peano du centre de l’objet qu’elle encadre. L'idée principale est de créer une structure d'arbre qui peut :

- se comporter comme arbre-R lors de la recherche
- supporter l'éclatement lors d'une insertion, en utilisant la valeur de Peano du rectangle de données inséré comme la clé primaire.

Donc, pour chaque nœud n de l'arbre, on mémorise ses RMEs, et la plus grande valeur de Peano (P.V.P.) des rectangles de données qui appartiennent au sous arbre avec la racine n.

L'arbre-R de Peano proposé possède alors la structure suivante :

- La feuille contient au plus C_l entrées, de la forme : (R, id_obj) où C_l est la capacité de la feuille, R est le RME de l'objet réel de coordonnées (x_b,x_h,y_b,y_h) et id_obj est un pointeur sur l'article de description de l'objet.
- Le nœud interne contient au plus C_n entrées de la forme :(R, ptr, P.V.P.) où C_n est la capacité d'un nœud interne, R est le RME qui enferme tous les fils de chaque nœud, ptr est un pointeur vers le nœud fils, et P.V.P. est la plus grande valeur de Peano contenue dans R.



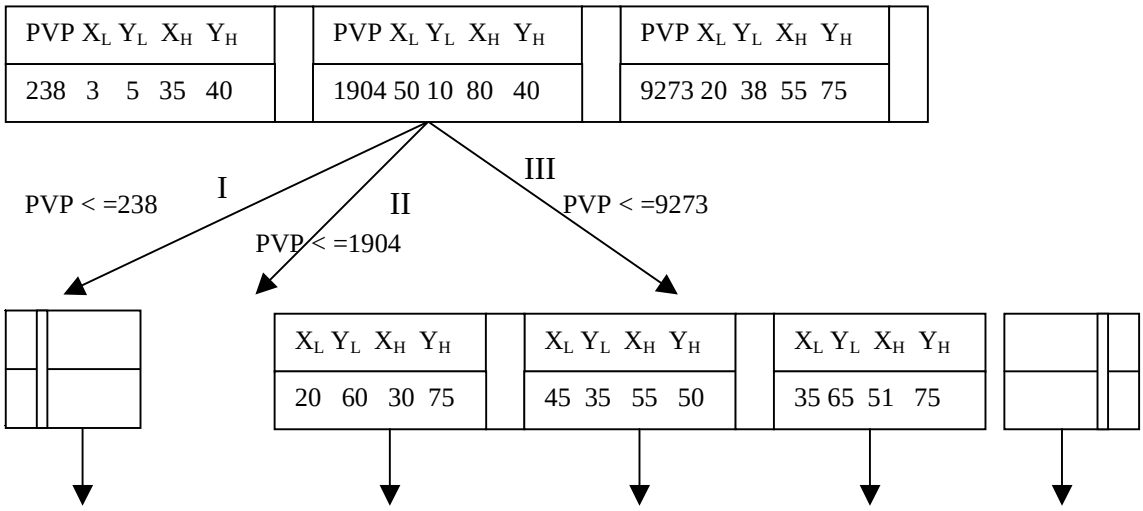


Figure 22 - Organisation des rectangles en arbre-R de Peano et sa structure

2. Signature 2D-R++-String

Une signature spatiale d'une image peut être obtenue à partir des relations spatiales entre les objets d'une image. L'arrangement spatial entre les objets d'une image peut ou ne pas être décrit avec précision. La représentation d'une image par une signature codée de l'arrangement spatial des objets qui la composent permet non seulement d'améliorer l'efficacité et la qualité des résultats de la recherche par le contenu, mais aussi de répondre à des requêtes sur des parties de l'image et de permettre à l'utilisateur d'affiner sa requête selon ses besoins .

La plupart des approches proposées, pour décrire l'arrangement spatial entre les objets dans une image, sont basées sur les 2D-strings [Cha+87] et ses variantes (2-D-E-strings [Jun88], 2-D-G-strings [CJL89], 2-D-C-strings [LH92] and 2-D-B-strings [Lee+92]).

Les deux inconvénients majeurs de telles représentations sont leur sensibilité et leur description floue puisque les opérateurs associés ne donnent pas une description complète des relations spatiales qui peuvent exister entre les objets. Par exemple, quand un objet est inclus dans un autre objet, leur représentation avec un seul point qui est le centre de gravité ne suffit pas pour décrire cette relation particulière. La figure 23 illustre un autre cas où deux images ont la même signature 2D-string mais dont l'arrangement spatial est différent pour chacune des images.

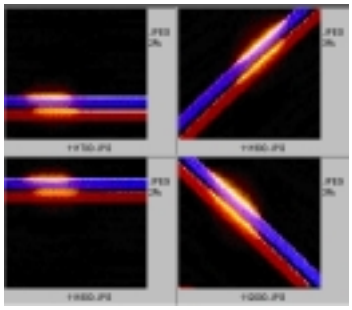


Figure 23 - Images qui ont la même chaîne 2-D mais qui sont visuellement différentes

Cette considération nous a conduite à étendre la représentation 2-D String à d'autres attributs caractéristiques des objets qui sont extraits lors du processus de la segmentation tels que le MBR, la taille relative et leurs dispersions spatiales au sein des quatre quadrants de l'image. La meilleure représentation de la relation spatiale entre les objets d'une image serait de tenir compte de tous les points du contour de chaque objet. Mais, une telle méthode serait très coûteuse en temps de calcul et en mémoire.

Pour capturer l'essentiel des relations spatiales entre les objets dans une image, nous avons proposé une signature 2-D-R++ qui est une variante de 2D-String. Chaque objet est décrit par les caractéristiques suivants:

- Un RME qui est utilisé pour encadrer l'objet dans l'image.
- Chaque objet est localisé par un identifiant qui est une clé de Peano, qui indique le centre de gravité de l'objet
- La taille de l'objet dans l'image est définie par sa surface relative.
- La dispersion hiérarchisée de l'objet à travers l'image est représentée par un code suivant quatre fenêtres qui ne sont que les quatre quadrants de l'image: 0 (SO), 1 (NO), 2 (SE) et 3 (NE). Ce code fournit plus de détails sur l'aspect visuel des objets. On peut ainsi faire des recherches à différents niveaux suivant les taux de distribution dans les quadrants.

Tous ces paramètres sont calculés lors du processus de segmentation. Notre approche [CC99b, CC99c] consiste à ajouter de nouveaux symboles pour mieux décrire l'arrangement spatial des objets, et permettre ainsi de décrire avec précision des relations du type "se chevaucher", "contenir", "rencontrer", "début", "fin", "égal", etc. La taille des objets permet de distinguer deux objets de couleurs identiques mais de tailles différentes.

Une fois ces relations spatiales sont représentées, la recherche des images en utilisant ces relations devient un simple problème de traitement de chaînes de caractères .

Puisqu'un objet est encadré par un RME, il est représenté par deux paires de coordonnées de début et de fin, une pour l'axe des ordonnées et l'autre pour l'axe des abscisses. Nous avons utilisé les sept opérateurs cités dans [Hua97] et qui sont résumés dans le tableau suivant.

Notations	Conditions
A<B	fin(A)<début(B)
A=B	début(A)=début(B), fin(A)=fin(B)
A\B	fin(A)=début(B)
A%B	début(A)<début(B), fin(A)>fin(B)
A]B	début(A)=début(B), fin(A)>fin(B)
A[B	début(A)<début(B), fin(A)=fin(B)
A/B	début(A)<début(B)< fin(A)<fin(B)

Tableau 3 - Définition des opérateurs spatiaux

Nous avons étendu ces opérateurs avec les caractéristiques définies précédemment, comme l'a proposé Huang dans [Hua97].

♦ Chaque objet R de taille t est noté R_t, où t est la taille relative de l'objet dans l'image.

♦ Un objet R est noté $_{\chi}R$, où χ représente le code qui correspond à l'ordre décroissant de dispersion de R à travers les quatre quadrants..

♦ L'opérateur "<" est étendu pour comprendre la distance d entre deux objets A et B. On note $A <_d B$, où d = début(B)- fin(A)

♦L'opérateur "%" est utilisé avec deux distances (d,d') entre les objets A et B. On le note $A \%_{d,d'} B$, où d = début(B)- début(A) et d' = fin(A) - fin(B)

♦ L'opérateur "]" est enrichi avec la distance d entre les objets A et B. On note $A]_d B$, où d = fin(A)- fin(B)

♦ L'opérateur "[" est étendu avec la distance d entre les objets A et B, noté par $A [_d B$, où d = début(B) - début(A)

♦ L'opérateur "\" est paramétré avec les distances d_1,d_2,d_3 entre les objets A et B, noté par $A \backslash_{d1,d2,d3} B$, où d_1=début(B) - début(A),d_2=fin(A) - début(B), d_3=fin(B)- fin(A)

3. Graphe de relations spatiales

Après la segmentation des objets homogènes, on s'intéresse aux relations spatiales entre les objets dans l'image. Une première signature spatiale peut être capturée par une chaîne 2-D appliqué aux trois points de chaque objet (deux pour le RME et un pour le barycentre). A partir de là, on extrait toutes les relations entre toutes les paires d'objets. Toutes les relations spatiales sont sauvegardées dans un graphe total de relations spatiales (GRS).

Dans ce graphe, chaque nœud représente un objet dans l'image tandis que chaque arc représente la relation spatiale entre les deux nœuds (objets) de l'extrémité. Cette relation spatiale est représentée par une paire d'opérateurs entre les deux objets respectivement sur l'axe des x et des y.

Pour simplifier la représentation du graphe GRS dans notre prototype, chaque nœud est représenté par un identifiant unique de la classe de couleurs à laquelle il appartient. Dans la figure 24, on présente un exemple de segmentation d'une image de personne, par notre méthode de croissance de région, et d'extraction de leurs régions homogènes. L'image d'origine et l'image obtenue après segmentation sont représentées respectivement par les figures 24-a et 24-b. . La figure 24-c décrit l'image optimisée où nous gardons seulement les six premières régions dominantes dont les caractéristiques sont détaillées dans le tableau 4. La figure 24-d illustre l'arrangement spatial des RMEs associés à ces six premières régions homogènes. L'image de test est de taille 352 x 288.

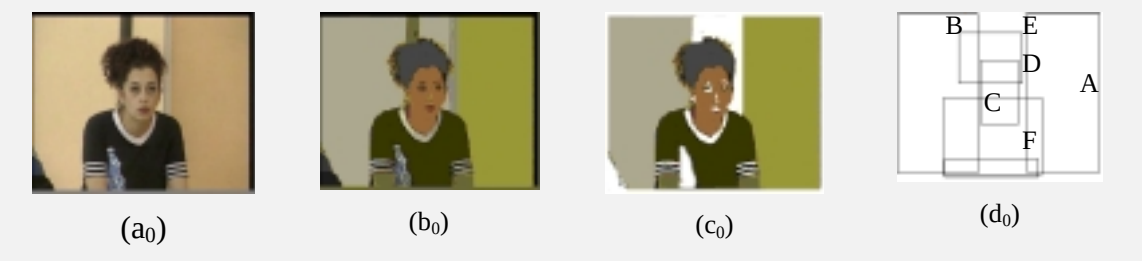


Figure 24 - Un exemple de segmentation et d'extraction de régions

Chaque région homogène a un RME [x_i, y_i, x_s, y_s], un barycentre (x_centroid, y_centroid), une taille (t) et un code représentant l'ordre de distribution (c). Dans cet exemple, chaque région homogène est étiqueté par une lettre (A... F).

Une chaîne 2D-R++ représente une signature spatiale globale riche dont les relations locales spatiales entre toute paire de d'objet sera extraite dans une phase ultérieure. Cette signature est représentée par un triplet (début, centre de gravité, fin) de 2D-String.

L'objet A possède une taille de 30579 pixels, qui fait 30579/352*288 ~ 30% du total de l'image. En plus, elle possède plus de pixels dans le quadrant NE que dans SE. L'objet A est représenté par $_{32}A_{30}$.

De même pour les autres objets, ils seront représentés par les caractéristiques suivants:

$_{10}B_{27}$, $_{02}C_{11}$, $_{3102}D_4$, $_{13}E_3$, and $_{02}F_2$.

La signature 2D-R++-String associée à l'exemple ci-dessus est :

- $B <_{55} B <_{24} C <_2 F <_{38} E <_{15} B <_7 D <_{15} E <_6 C <_5 D <_1 F <_{30} D <_6 E <_{13} A <_{14} F <_6 C <_{43} A <_{56} A$
- $A = B <_{38} E <_{30} E <_{14} D <_{32} E <_1 B <_{15} A <_5 D <_{15} C <_{41} D <_{21} C <_{37} F <_7 C <_{18} B = F <_1 A <_5 F$

On peut la décomposer en un triplet :

I: Début des RMEs:	*II: Barycentres des objets:*	*III: Fin des RMEs:*
$B <_{79} C <_2 F <_{38} E <_{22} D <_{76} A$	$B <_{101} E <_6 C <_5 D <_1 F <_{112} A$	$B <_{64} D <_6 E <_{27} F <_6 C <_{99} A$
$A = B <_{38} E <_{44} D <_{68} C <_{99} F$	$E <_{47} B <_{15} A <_5 D <_{77} C <_{62} F$	$E <_{77} D <_{84} A <_1 B <_2 C <_4 F$

La première composante de cette chaîne est la projection de toutes les régions homogènes sur l'axe des abscisses, tandis que la deuxième est la projection sur l'axe des ordonnées. On peut déduire de ces descriptions que l'objet A se trouve plus au Nord-Est qu'au Sud-Est. On peut même dire qu'il se trouve à l'Est de l'image et qu'il représente 30% de l'image.

Par exemple, B<C dans la première composante indique que l'objet homogène d'étiquette B se trouve à gauche de l'objet homogène d'étiquette C.

Le tableau 5 résume les relations spatiales entre les objets dominants de l'image et le tableau 6 simplifie leurs notations. Comme on peut le voir dans du tableau 5, la relation spatiale déduite de la chaîne 2D-R++ entre les objets dominants A et B est (B<A,B]A).

Généralement, on note une relation spatiale entre un couple d'objets(U,V) par (Γ_x, Γ_y). Cela veut dire que les relations spatiales déduites sont : U Γ_x V et U Γ_y V. On note qu'en général la relation spatiale U Γ V est différente de la relation spatiale V Γ U. L'opérateur Γ n'est pas symétrique, si le sens de la relation est inversé on le précise avec un signe - (-Γ) :U Γ V = V -Γ U

Par exemple, dans le tableau 5, on note que la relation spatiale entre le couple d'objets dominants A et B est (-<,-]). Cela veut dire que les relations extraites sont (B<A,B]A) et non (A<B,A]B). L'opérateur] peut être étendu ($]_d$) pour inclure la distance d = fin(A) – fin(B).

Pour mémoriser toutes les relations spatiales entre les objets dominants dans une image, on construit un graphe de relations spatiales (GRS) [GR95]. La figure 25 montre le graphe associé à l'image de la figure 24.

Régions homogènes	RME	Barycentre	Taille	Code de dispersion
A 	(218,6) (337,281)	(281,136)	30579	32
B 	(1 , 6) (135 ,280)	(56,121)	27064	10
C 	(80,156) (238, 282)	(163,218)	11104	02
D 	(142,88) (199, 197)	(168,141)	3891	3102
E 	(120, 44) (205, 120)	(157, 74)	3463	13
F	(82,255) (232,286)	(169,280)	2050	02

Tableau 4 - sélection des régions dominantes par ordre décroissant

$\Gamma_{X,Y}$	A	B	C	D	E	F
A	*	B<A, B]A	C/A, A/C	D<A,A%D	E<A, A%E	F/A, A/F
B	*	*	B/C, B/C	B<D, B%D	E/B, B%E	B/F ,B/F
C	*	*	*	C%D, D/C	C%E, E<C	C%F ,C/F
D	*	*	*	*	E%D ,E/D	F%D, D<F
E	*	*	*	*	*	F%E, E<F
F	*	*	*	*	*	*

Tableau 5 - Résumé des relations spatiales entre les objets dominants

			V			
	$U\Gamma_{x,y}V$	B	C	D	E	F
	A	(-<,-])	(-/,/)	(-<,%)	(-<,%)	(-/,/)
U	B		(/,/)	(<,%)	(-/,%)	(/,/)
	C			(%,-/)	(%,-<)	(%,/)
	D				(-%,-/)	(-%,<)
	E					(-%,<)

Tableau 6 - Notation simplifiée des relations spatiales

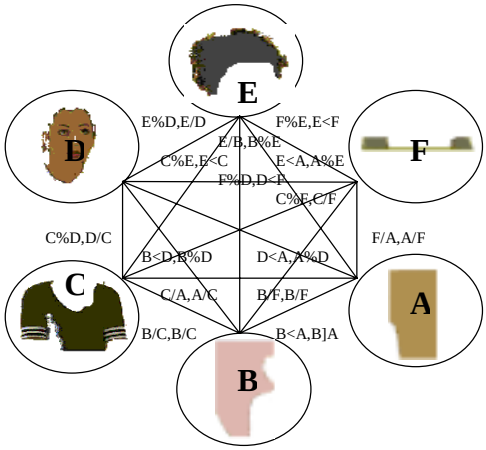


Figure 25 - Le GRS correspondant à l'image exemple

Le parcours des arcs du graphe, composé de six nœuds, représentant les six régions homogènes dominantes, nécessite 15 opérations. Généralement, le parcours d'un GRS composé de n nœuds nécessite S opérations, qui est calculée par la formule suivante:

$$S = (n-1) + (n-2) + \ldots + 1 = \sum_{i=1}^{n}(n-i) = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2}$$

Pour chaque image composé de n régions homogènes, on doit enregistrer n(n-1)/2 relations spatiales. Chaque relation spatiale concernant deux objets possède deux volets l'une suivant l'axe horizontal et l'autre suivant l'axe vertical. La figure 26 montre d'autres résultats de segmentation avec différents paramètres. Les images sont représentées respectivement dans le format 320 x 240 352 x 288, 352 x 288, 352 x 288, 320 x 240, et chaque pixel est codé en 24 bits. Nous avons fixé le seuil de voisinage ρ à 3, le seuil d'homogénéité de la couleur μ à 6, le seuil λ = 0.15 % et le seuil θ = 20. Le tableau 7 montre les résultats de la segmentation d'autres images variées.

Image	R_T	R_E	R'_E	T_1	T'_1	T_2
a_0	29918	4401	11	5	4	1
a_1	56649	12942	11	7	4	2
a_2	69609	17695	13	8	5	3
a_3	59445	14955	19	7	5	3
a_4	49514	7858	9	5	4	2
a_5	49502	3058	4	3	2	1

Tableau 7 - Nombre de régions dominantes avant et après le seuillage et temps de calcul.

Avec :

R_T : le nombre total des composantes connexes, sans seuillage.

R_E: le nombre de régions homogènes extraites en tenant compte seulement des seuils ρ et μ.

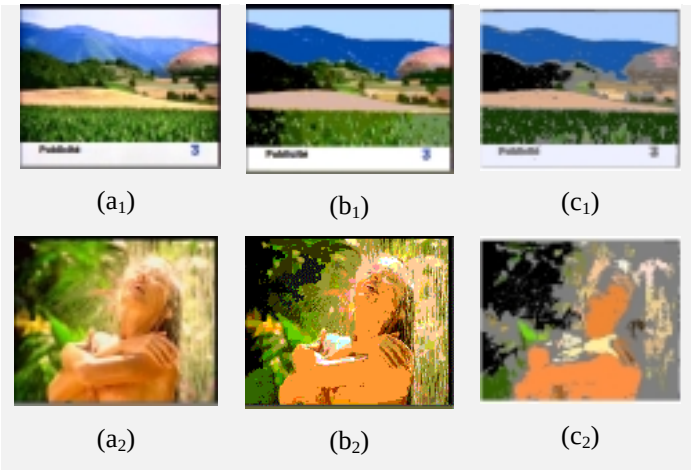
R'_E: Le nombre des régions homogènes réellement considérés dont on caractérise les relations spatiales, en tenant compte des seuils ρ, μ, λ, et θ.

T_1: Temps CPU en secondes demandé par le processus de segmentation quand on extrait les R_E régions homogènes.

T'_1: Temps CPU en secondes demandé par le processus de segmentation quand on extrait les R'_E régions homogènes.

T_2: Temps CPU en secondes pour l'extraction de la signature 2-D-R++-String et la construction du GRS.

Les mesures du temps de calcul sont en secondes et elles sont effectuées sous Pentium-Pro 200.



(a_1) (b_1) (c_1)

(a_2) (b_2) (c_2)

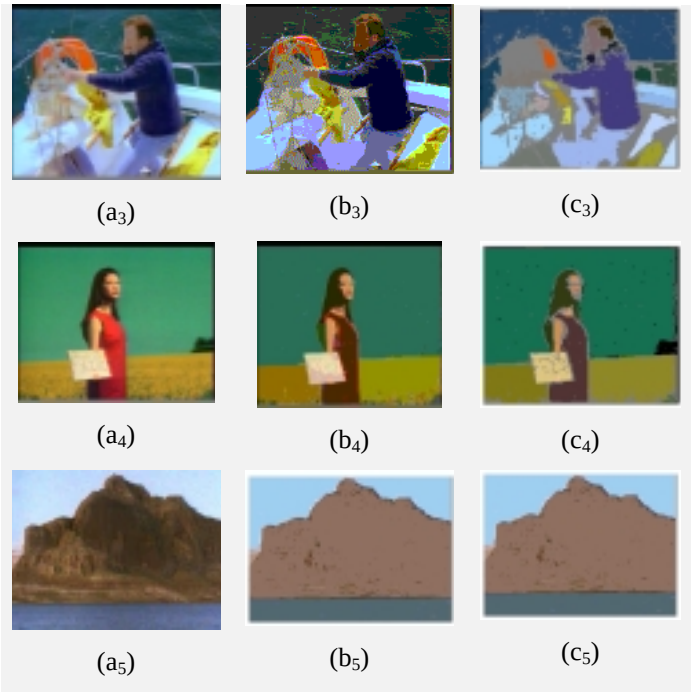


Figure 26 - Résultats de segmentation.(a) originaux, (b) résultats de segmentation avec seuillage par ρ & μ et (c) résultats avec utilisation supplémentaire des seuils λ et θ

E. Caractéristiques de région

Une fois une région homogène est extraite, on peut déterminer deux types de caractéristiques: une géométrique et l'autre basée sur la texture de sa surface (moments) [Dub99, Wee96].

1. Indices basés sur les moments

Les indices que nous avons présenté pour caractériser les indices visuels peuvent être utilisées pour caractériser les régions.

2. Indices géométriques

Pour chaque région homogène R, composée d'un ensemble de pixels $p_i(x_i,y_i)$, on peut déterminer les caractéristiques suivantes:

Aire

$$A = \iint_{\Re} dxdy = \int_F y(t)\frac{dx(t)}{dt}dt - \int_F x(t)\frac{dy(t)}{dt}dt$$

$\Re$ et F désignent respectivement la région concernée et sa frontière. En fait, la surface n'est que l'ensemble total de pixels de la région.

Centre de gravité

Etant donné N pixels (x_0,y_0), (x_1,y_1), …(x_{N-1},y_{N-1}) qui composent la région homogène R, le centre de gravité G, de coordonnée (x_G,y_G) est défini par:

$$x_G = \frac{1}{N}\sum_{i=0}^{N-1} x_i \quad \text{et} \quad y_G = \frac{1}{N}\sum_{i=0}^{N-1} y_i$$

Périmètre

$P = \int \sqrt{x^2(t) + y^2(t)}dt$ t désigne une paramétrisation du contour.

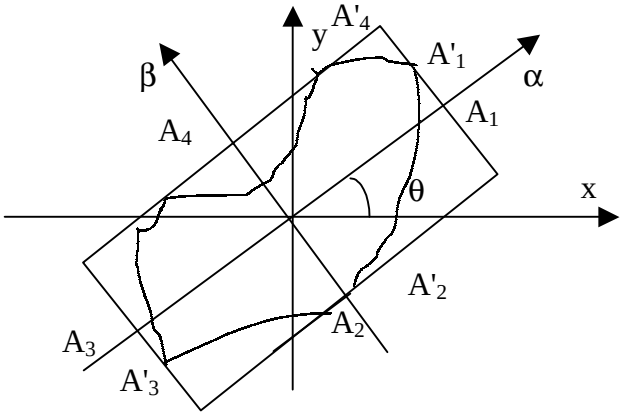
Il est en général calculé comme la somme des distances entre les pixels consécutifs du contour.

Soit M le nombre de pixels du contour. Il peut être calculé à partir du contour intérieur ou extérieur, même si on préfère souvent le contour extérieur:

$$P = \sum_{i=0}^{M-2} \sqrt{(b_{x,i+1} - b_{x,i})^2 + (b_{y,i+1} - b_{y,i})^2}$$

Rectangle d'encadrement

Il s'agit du plus petit rectangle(bounding rectangle) de même orientation qui encadre la région homogène R.



θ étant connu, on fait la transformation pour les points du contour:

$\alpha = x \cos\theta + y \sin\theta$

$\beta = -x \sin\theta + y \cos\theta$

Le but est de chercher α_{min}, α_{max}, β_{min} et β_{max} qui donnent les points A'_3, A'_1, A'_2 et A'_4, à partir desquels, on déduit le rectangle. En particulier sa largeur(L) et sa hauteur (H).

Un cas particulier de ces rectangles circonscrits, est le rectangle avec $\theta = 0$. C'est le rectangle minimum d'encadrement RME (Minimum Bounding Rectangle MBR) qu'on utilise généralement pour encadrer R.

Aspect géométrique

Elongation

C'est le rapport entre le rayon maximal et le rayon minimal (aspect ratio).

Le choix du rayon peut être différent:.

- les distances maximale et minimale entre le centre de gravité et le contour.
- La longueur et la largeur du rectangle d'encadrement
- La longueur et la largeur maximales de la région suivant le rectangle d'encadrement

En général, on choisit simplement la longueur et la largeur du RME.

Aspect circulaire

C'est le rapport entre le périmètre de la région et sa surface. Il est défini par :

$$\frac{P^2}{A}$$

Courbure et son énergie

Elle détermine le changement de direction dans un contour, ce qui permet de déterminer les coins d'une région. Si t est l'abscisse curviligne sur le contour, la courbure c(t) s'écrit:

$$|c(t)|^2 = \left[\frac{d^2y}{dt^2}\right]^2 + \left[\frac{d^2x}{dt^2}\right]^2$$

et son énergie est :

$$E = \frac{1}{M}\int_0^M |c(t)|^2 dt$$

Soit x_i et y_i la ième coordonnée du contour. La courbure de ce contour à la ième position est définie par:

$$C_i = \sqrt{(2x_i - x_{i-1} - x_{i+1})^2 + (2y_i - y_{i-1} - y_{i+1})^2}$$

où i = 1, 2, … M-2.

On peut ainsi définir l'énergie de courbure:

$$E = \frac{1}{M}\sum_{i=1}^{M-2} C_i^2$$

où M est le nombre des pixels du contour

Moyenne, et variance

La moyenne ou distance moyenne des pixels d'une région à partir du centre de gravité est définie par:

$$Moy_{dist} = \frac{1}{N}\sum_{i=0}^{N-1}\sqrt{(x_i - x_G)^2 + (y_i - y_G)^2}$$

$$Var_{dist} = \frac{1}{N}\sum_{i=0}^{N-1}[\sqrt{(x_i - x_G)^2 + (y_i - y_G)^2} - Moy_{dist}]^2$$

où N est le nombre total de la région et (x_G,y_G) sont les coordonnées du centre de gravité.

Nombre de connexité

Il est déterminé par le nombre d'Euler (E), qui lui est calculé à partir du nombre de composantes connexes (C) et du nombre de trous (H).

$E = C - H$

Ce critère est utilisée généralement en images binaires.

3. Importance de la région

Index de la région

Une région est d'abord identifiée par son centre de gravité, puis par son rectangle minimum d'encadrement. On associe au centre de gravité de la région (x_G,y_G),la clé de Peano correspondante .

La couleur de la région sera représentée par la couleur dominante lors de l'affichage de celle-ci. A chaque région, correspond un ensemble de couleurs (ou de niveaux de gris), donc un nombre de couleurs. Pour chacune de ces couleurs, correspond un indice de cohérence à partir du nombre de pixels connexes et de ceux qui ne sont pas connexes.

Rapport de superficie

C'est le rapport entre le nombre de pixels dans la région et le nombre total de pixels dans l'image. Il représente le pourcentage que représente la région de l'image.

$$\text{Arearatio}_R = \frac{\sum \text{pixel}_R}{\text{Longueur} \times \text{L arg eur}}$$

Où $\sum \text{pixel}_R$ est le nombre total de pixels dans la région R.

Position

Elle indique la position du centre de gravité de la région par rapport au centre de l'image

$$\text{Position}_R = \sqrt{(c_x - x_G)^2 + (c_y - x_G)^2}$$

où (c_x ,c_y) est le centre de l'image, et (x_G ,y_G) est le centre de gravité de la région R.

Compacité

Elle décrit la compacité de la région R . Elle est égale à 1 si la région est ronde, et très petite si le contour de la région R est compliqué .

$$\text{Compacité}_R = \frac{4\pi \times (\text{aire}_R)}{(\text{perimetre}_R)^2}$$

où aire_R et périmètre_R sont respectivement l'aire et le périmètre de la région R.

Bordure

Elle permet de savoir si la région fait partie de l'arrière-plan où non.

$$\text{Bordure}_R = \frac{\sum \text{connect}_R}{2 \times (\text{Longueur} + \text{Largeur})}$$

où $\sum \text{connect}_R$ est le nombre de pixels qui sont à la fois sur le contour de la région R et sur la frontière de l'image.

Poids

Il permet de savoir le poids d'une région par rapport à ses voisines. On considère qu'une région est très marquante (a plus de poids) s'il y a une grande différence de couleurs entre elle et ses voisines, et si elle est plus grande et plus proche que ses régions voisines.

$$\text{Poids}_R = \sum_{k=1,k\neq R}^{n} \left\| \text{color}_R - \text{color}_k \right\|^2 \times (1 - \text{distance}_{R,k}) \times \text{Arearatio}_k$$

où

$$\text{distance}_{R,k} = \frac{\sqrt{(x_G - x_k)^2 + (y_G - x_k)}}{\sqrt{\text{Longueur}^2 + \text{Largeur}^2}}$$

(x_G,y_G) et (x_k,y_k) sont respectivement le centre de gravité de la région R et de la région k .

Forme

Code de Freeman

Dans la majorité des cas le point P_n n'a que deux candidats potentiels pour P_{n+1} dont un est le point P_{n-1}. Il n'y a donc pas d'ambiguïté. Cependant, il faut aussi prendre en compte les points anguleux n'ayant aucun suivant. La technique la plus simple consiste à supprimer de l'image de départ toutes les configurations de ce type :

transition L:

0 0 0
0 1 0
0 1 1

transition I

0 0 0
0 1 0
0 1 0

Chacune de ces figures ne représente qu'une des configurations possibles. Il y a ainsi 8 configurations de chaque type, que l'on obtient par une rotation de 45°. Le contour est représenté par une chaîne de Freeman représentant la forme de la région suivant les 8 directions {0, …, 8}.

Une fois le contour extrait, on peut se contenter de ne

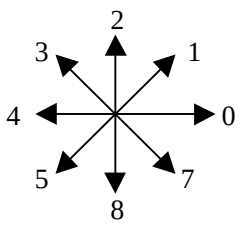


Figure 27 - Code de Freeman

mémoriser que les coordonnées du point de départ et la suite des directions d (d suivant les 8 directions (cf fig. 27) plutôt que les coordonnées des points successifs.

La suite des directions traduit la forme et le point de départ décrit sa localisation spatiale. Ce type de codage est très utile pour la reconnaissance de formes car beaucoup y ont vu une signature invariante d'une forme. De plus, de très nombreux paramètres de forme peuvent être mesurés directement sur le codage de Freeman (périmètre, aire, centre de gravité, axes d'inertie, …), et la chaîne peut être compressée.

Descripteurs de Fourier

Une autre méthode de description du contour d'un objet est l'utilisation de la transformée de Fourier discrète à une dimension. Le contour est une suite de valeurs échantillonnées x_i, y_i (i = 0,…,N-1). Si le contour est fermé, on obtient une suite périodique de valeurs (période i) auxquelles on peut appliquer la transformée de Fourier discrète qui est :

$f(r)= x_r + j\ y_r \qquad r = 0,1,2,\ldots N-1$

$$F(n) = \frac{1}{N}\sum_{r=0}^{N-1} f(r)\exp\left[\frac{-j2\pi nr}{N}\right]$$

et son inverse est :

$$f(r) = \sum_{n=0}^{N-1} F(n)\exp\left[\frac{j2\pi nr}{N}\right]$$

Les F(n) sont les descripteurs de Fourier, elles fournissent des propriétés intéressantes pour le contour. Par exemple, F(0) décrit le centre de gravité du contour. Pour n=0, la partie réelle de F(n) devient :

$$x_G = \frac{1}{N}\sum_{r=0}^{N-1} x_r$$

et la partie imaginaire de F(n) devient:

$$y_G = \frac{1}{N}\sum_{r=0}^{N-1} y_r$$

$|F(n)|$ est invariant au choix de l'origine, à une opération de rotation ou de symétrie. De même, $\frac{F(n)}{|F(n)|}$ est invariant au changement d'échelle. On peut utiliser ces propriétés pour reconnaître des objets indépendamment de leurs tailles ou de leurs orientations.

En reconnaissance de forme, quand on dispose de deux formes $f_1(r)$ et $f_2(r)$, similaires à un changement d'origine n_0, une translation u_0, une rotation θ_0 et un changement d'échelle α près dont les descripteurs respectifs $F_1(n)$ et $F_2(n)$. Ces deux formes seront dites similaires si leur distance est faible. Cette distance est définie par :

$$d(u_0, \alpha, \theta_0, n_0) = \sum_{n=0}^{N-1} \left| f_1(n) - \alpha f_2(n + n_0)\exp(j\theta_0) - u_0 \right|^2$$

On peut alors rechercher le meilleur jeu de paramètres minimisant cette distance.

Si

$$\sum_n f_1(n) = \sum_n f_1(n) = 0$$

et si n_0 est fixé, cette distance est minimisée pour :

$$u_0 = 0\,;\, \alpha = \frac{\sum_k c(k)\cos(\varphi_k + k\phi + \theta_0)}{\sum_k |F_2(k)|^2}\,;\, \tan\theta_0 = -\frac{\sum_k c(k)\sin(\varphi_k + k\phi)}{\sum_k c(k)\cos(\varphi_k + k\phi)}$$

$$F_1(k) * F_2(k) = c(k)\exp(j\varphi_k) \quad c(k) \text{ réel}$$

$$\phi = -\frac{2\pi n_0}{N}$$

La valeur minimale de la distance est alors :

$$d_{min} = \min_{\phi}\left[\sum_k |F_1(k) - \alpha F_2(k)\exp[j(k\phi + \theta_0)]|^2\right]$$

Cette quantité est évaluée pour différentes valeurs de $\phi=\phi(n_0)$ (n_0 = 0,1,…,N-1). Et on choisit la valeur la plus petite. d est une mesure de la différence entre deux formes.

F. Conclusion

Nous venons donc de présenter nos deux méthodes permettant la segmentation d'objets visuellement homogènes qui sont ensuite caractérisés par des descripteurs de couleur, de texture et de forme, ainsi que l'extraction de leurs relations spatiales. De toute façon, les objets peuvent être considérés visuellement important s'ils attirent l'attention de l'œil humain ou s'ils ont une signification pour lui. La partie qui exprime le sujet de l'image est toujours considérée plus importante, pour l'homme, que les parties qui n'en ont pas de rapport [NM96]. En ce qui concerne l'importance d'un objet visuel, les travaux dans [Zha+96] proposent des règles avec un raisonnement flou à partir des caractéristiques que nous avons présentées et des concepts suivants :

- Point isolé: C'est un point dont le nombre de 8-voisins est égal à 0, on parle aussi de région isolée si elle n'a pas de régions voisines.
- Arc : le contour est appelé un arc si ce n'est pas un point isolé et sa surface effective est égale à 0
- Une ligne horizontale: c'est une région dont y_2-y_1=1 et x_2-x_1>>1
- Une ligne verticale: c'est une région dont x_2-y_1=1 et x_2-x_1>>1
- Sommet : Un point ou une ligne horizontale est appelé sommet si les voisins de gauche et de droite sont tous situés au-dessus ou en dessous .
- Une conjonction de chemins multiples : Un point est appelé une conjonction de chemins multiples quand il n'est pas d'arc et est passé au moins deux fois par le contour.
- Contour simple: Une courbe fermée est appelée un contour simple si elle ne contient pas d'arc ni de conjonction de chemins multiples.
- Forme arbitraire: Le résultat de la segmentation est une forme arbitraire qui peut contenir des contours simples, des arcs et des conjonctions de chemins multiples.

Quoi qu’il en soit, une fois les objets visuellement homogènes segmentés et une image résumée par leurs descripteurs et leurs relations spatiales, il faut des mécanismes de recherche pour répondre aux requêtes de l’utilisateur. C’est le sujet que nous allons traiter au chapitre suivant.

V. Moteur de recherche par le contenu

Une fois les indices visuels et les objets homogènes segmentés et stockés, il s'agit de définir le mécanisme de recherche permettant de retrouver les images de la base afin de répondre aux requêtes utilisateurs. Dans ce chapitre, nous présentons d'abord l'architecture de notre prototype et les fonctionnalités de l'interface, puis nous décrivons les structures de données implémentées et l'algorithme de recherche basé sur un mécanisme de votes. Enfin, nous présentons et commentons quelques résultats expérimentaux.

A. Architecture

Le prototype de notre système est constitué de deux composantes principales: l'indexation et la recherche par similarité.

1. Indexation

Cette partie concerne l'indexation des caractéristiques visuelles telles que la couleur, la forme et la texture. Elle s'appuie sur les techniques et algorithmes que nous avons décrits au chapitre précédent. Nous rappelons ici brièvement les principales étapes.

Les images sources sont stockées sous différents formats tels que GIF, JPEG, etc. Au moment de leur insertion dans la base les indices visuels sont automatiquement extraits. Afin de résumer une image, on en extrait alors les caractéristiques visuellement significatives telles que la texture dominante, la couleur dominante, les objets homogènes, leurs formes, etc. L'utilisateur peut aussi éventuellement identifier des objets et les annoter textuellement. Certains objets peuvent être classés en tant que primitives visuelles telles que les cercles, ligne, etc..

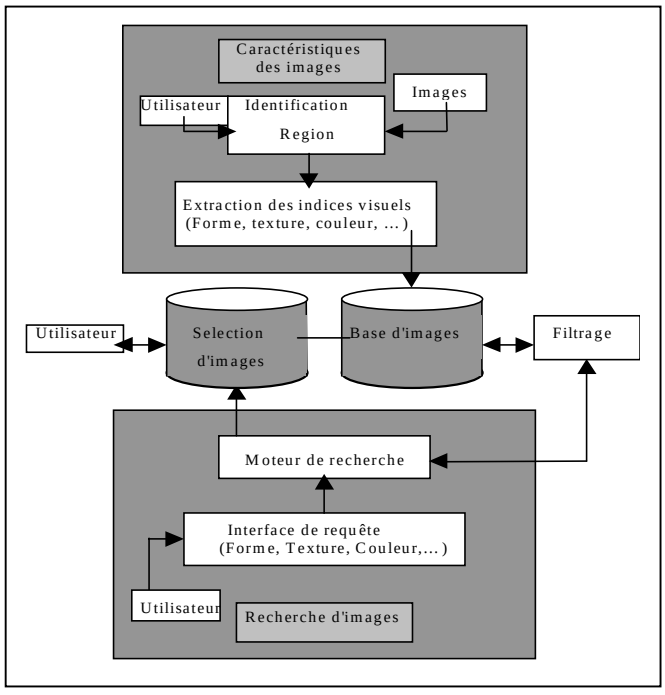


Figure 28 - Architecture du système de recherche

Ce processus d'indexation d'une image peut être résumé par l'exemple donné dans la figure 29. La palette de couleurs associée à l'image de l'exemple est réduite à 3 classes de couleur dont elles sont représentées par les couleurs Noir, Gris et Rouge. L'image d'origine, en haut à

gauche, contient 6 régions homogènes. Le processus de segmentation en identifie 5 objets homogènes compte tenu du voisinage des couleurs et des positions. Le premier qui est composé de deux régions voisines et de couleurs proches, est représenté par la couleur noir .Le 2nd et le 3ème objet, triangle et cercle, possèdent des couleurs qui appartiennent à la troisième classe qui représentée par la couleur rouge. Enfin, le 4ème et le 5ème objets sont des régions cohérentes avec deux couleurs de la deuxième classe.

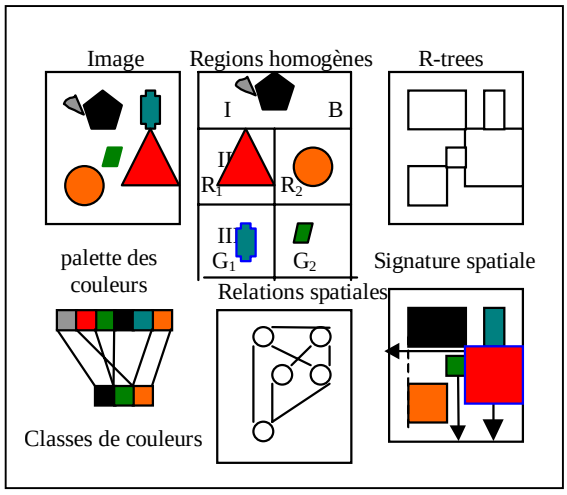


Figure 29 - Processus de segmentation et d'indexation

2. La recherche

Cette composante réalise la recherche par la similarité d'images proches par rapport à une requête utilisateur. Pour formuler sa requête, l'utilisateur peut dessiner une image ou la choisir directement de la base. Il a à sa disposition un browser graphique interactif qui lui permet de naviguer dans la base d'image. Il peut aussi identifier des régions comme objets de recherche. Pour cela, plusieurs distances de similarité, que nous avons présentées au chapitre précédent, peuvent être utilisées, par exemple les distances euclidienne, quadratique et d'intersection d'histogramme.

Si l'image de requête existe dans la base, ses caractéristiques ont été déjà extraites. Sinon, on procède à sa segmentation en vue d'en extraire des objets homogènes suivant un ou plusieurs critères d'homogénéité par les méthodes de segmentation que nous avons décrites au chapitre précédent. Quand l'utilisateur fournit une requête au moyen d'un browser, les caractéristiques visuelles extraites sont comparées à celles enregistrées dans la base. La figure 28 illustre cette architecture utilisée

Soulignons aussi que la plupart des systèmes structurent au préalable la base d'images en classes d'images ayant des propriétés communes, pour que la recherche ne s'effectue que sur les classes significatives pour la requête, gagnant ainsi en efficacité et en robustesse.

Notre question centrale ici est donc l'implémentation d'un moteur de recherche qui, à partir d'une image requête, déterminera les images de la base qui ressemblent à celle-ci par rapport à une distance de similarité. Nous décrivons au paragraphe suivant les structures de données utilisées et l'algorithme de vote réalisant la fonction de recherche.

B. Structures de données et algorithme de recherche

La recherche d'images par la similarité se fait donc par le biais d'une ou plusieurs distances. Les réponses doivent est triées par ordre de similarité. Pour des raisons d'efficacité au moment de la requête, nous calculons, au moment de l'insertion d'une image dans la base, un index d'images qui lui sont proches. Ensuite, pour réduire la quantité d'images candidates à une image requête, une première sélection d'images est réalisée sur la simple base d'histogrammes

de couleurs. Ce n'est qu'après cette première sélection qu'on utilise les informations spatiales d'objets homogènes qui ont été segmentés pour affiner et aboutir à des résultats finaux de recherche. Mais avant tout, pour toute image déjà dans la base, on réalise un pré-calcul de distance afin d'obtenir les premières images qui lui sont similaires.

1. Calcul de distance globale entre les images

On utilise un nombre N pour paramétrer le nombre des premières images qui sont les plus proches à une image de la base. Soient NbreImages le nombre d'images de la base et Distance[I,J] un tableau à deux dimensions mesurant la distance entre l'image I et l'image J. Soit Tab_Distance un vecteur, de dimension NbreImages, de structure SIM qui est définie par :

```
Type SIM  =   enregistrement
                    indice : entier ; //indice de l'image
                    dist:     Réel ;  //distance de l'image
              Fin SIM
```

Le vecteur bidimensionnel Proche contiendra pour chaque image I, les indices de toutes les images qui lui sont proches. La procédure suivante permet de réaliser ce pré-calcule des distances entre toute paire d'images dans la base afin d'aboutir à cet index d'images Proche. qui détermine les N premières images proche d'une image donnée.

```
Pour I = 0 à NbreImages - 2 Faire
        Distance[I,I] = 0;
        Pour J = I +1 à NbreImages-1 Faire
                Tab_Distance[J].dist =Distance[I,J]= Calcul_Distance(I,J);
                Tab_Distance[J].indice=J ;
        Fin
        Pour J = 0 à I Faire
                Tab_Distance[J].dist=Distance[J,I]=Distance[I,J];
                Tab_Distance[J].indice=J ;
        Fin
        /* Tri des distances par ordres croissants */
        Trier le tableau Tab_Distance suivant les distances dist
        /*Mettre dans Proche[I] les anciens indices correspondants*/
        Pour J = 0 à NbreImages-1 Faire
        Proche[I,J] = Tab_Distance[J].indice;
Fin
```

Dans la procédure, Calcul_Distance(I,J) est une fonction qui fait appel à une distance de similarité pour calculer la distance entre les deux images I et J. Elle peut s'appuyer sur plusieurs signatures qui prennent compte des indices visuels, notamment les histogrammes

pour la couleur, les moments pour la texture et les descripteurs de Fourier pour la forme. Des coefficients de poids sont utilisés pour pondérer ces différentes signatures.

A la fin de cette procédure, on a donc pour chaque image la liste des images qui lui sont proches par ordre de similarité croissante comme illustre la figure 30. Généralement, on se limite à la première vingtaine des images proches pour l'affichage. Lors de l'insertion d'une nouvelle image, on calcule sa distance par rapport aux autres images et on met à jours les priorités.

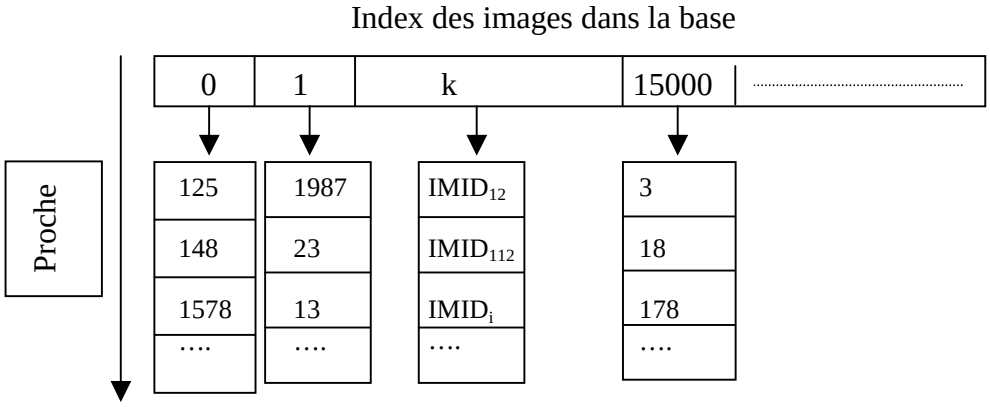


Figure 30 - Structure à jour des images proches

2. Les structures de données

Une fois l'utilisateur formule une image requête, une première sélection d'images peut se faire par les histogrammes de couleurs qui donne un ensemble d'images candidates, réduisant les images à considérer pour les critères de similarité. L'étape suivante consiste alors à utiliser les relations spatiales d'objets homogènes pour en extraire la liste finale des images ayant la même disposition spatiale que l'image requête. La comparaison des images se fait par la comparaison de leurs graphes de relations spatiales (GRSs) respectifs. Pour cela, plusieurs structures de données sont utilisées, capturant les caractéristiques des régions segmentées par les techniques étudiées au chapitre précédent.

Par exemple, pour une image qui est identifiée par un identifiant *ImId*, les informations d'une région sont mémorisées dans une relation REGION dont la structure est présentée dans le tableau 8 où nous stockons pour chaque région la couleur représentative, les coordonnées du rectangle minimum englobant (RME), le barycentre, la taille de la région ainsi que sa chaîne code qui représente la dispersion spatiale de l'objet dans l'image. Une image étant divisée en quatre quadrants repérés respectivement par 0, 1, 2 et 3 (donc codé en 2 bits), une région 02 ayant pour code 310 ou en binaire 110100 est dispersée sur les quadrants 3, 1 et 0.

IMID	*REGID*	C_R	(x_i,y_i)	(x_s,y_s)	(x_c,y_c)	*S(%)*	…	*Code*
0001	01	10	6	37	38	25	…	00011011(0123) =$1B
0001	02	165	57	49	194	1.2	…	110100(310) =$34
0002	01	13	3	256	815	1.7	…	11(3) =$03
…								
0003	01	110	228	601	800	26	…	10010011(2103) =$93

Tableau 8 - La relation REGION et ses attributs

La relation REGION contient la couleur représentative C_R, les coordonnées du RME $[x_i,y_i,x_s,y_s]$, le barycentre (x_c,y_c), la taille S de la région et la chaîne code (code). Les coordonnées sont représentées par leurs clés de Peano correspondante

Pour chaque image, on mémorise aussi les relations spatiales entre ses régions qui la composent dans le Tableau 9. Dans un but de simplification, chaque région étant représentée par sa couleur dominante, nous ne représentons que les paires de relations spatiales entre les couleurs dans l'ordre d'importance des régions. Chaque région dominante est codée par le numéro de l'image suivi de son ordre d'importance parmi les régions. Par exemple dans le Tableau 9, si l'image 101 possède cinq régions dominantes, elles vont être identifiées par 101**01**, 101**02**, 101**03**, 101**04** et 101**05**, la région 10101 étant la plus importante, vient ensuite la région 10102, etc. Dans notre implémentation, nous ne considérons au maximum que les 20 premières régions dominantes. Dans un souci de compacité, chaque ligne du tableau correspond à une image et code l'ensemble des relations spatiales de ses régions dominantes. Chaque relation spatiale entre deux régions A et B étant projetée sur les deux axes x et y, elle est caractérisé par deux relations spatiales telles que nous les avons présentées au chapitre précédent. Nous ne mémorisons que les opérateurs spatiaux, assumant implicitement que la comparaison se fait toujours dans l'ordre : première région avec la seconde, puis avec la troisième, puis la quatrième, etc. ensuite la deuxième région avec la troisième, puis la quatrième, etc. Le symbole '-' est utilisé pour signifier le sens inversé de l'opérateur. Par exemple la première ligne du Tableau 9, mémorise les relations spatiales des cinq régions dominantes de l'image identifiée 101 avec la chaîne "<[-%=-<=[[<\\<]-]<<\\ <-<", ce qui traduire les relations spatiales suivantes :

(10101,10102)=(<,[) ; (10101,10103)=(-%,=) ; (10101, 10104)=(-<,=) ; (10101, 10105)=([,[) ;

(10102, 10103)=(<,\) ; (10102, 10104)=(\,<) ; (10102, 10105)=(],-]) ;

(10103, 10104)=(<,<) ; (10103, 10105)=(\,\) ;

(10104, 10105)=(<,-<) ;

Nous avons donc 10 relations spatiales.

IMID	*Relations spatiales entre les régions*
0101	<[-%=-<=[[<\\<]-]<<\\ <-<
0102	<<-\=%%
…	
1020	%%%%-%-%

Tableau 9 - Tableau des relations spatiales associées aux images

Pour des raisons de performance, cette table des relations spatiales a été implémentée par une structure de "fichier inversé" comme l'illustre Tableau 10. En effet, pour chaque relation spatiale composée de deux opérateurs spatiaux sur les deux axes, par exemple < sur l'axe x, et \ sur l'axe y, nous avons un fichier séparé à deux colonnes. Chaque région étant représentée par sa couleur représentante, les ligne du tableau indiquent alors toutes les relations spatiales entre les deux couleurs représentantes extraites de toutes les images de la base. Par exemple, la première du Tableau 10 indique que la relation spatiale entre (couleur12, couleur15) est (<,\) et elle a été identifiée 12 fois dans l'image 11, puis 2 fois dans l'image 111 et 3 fois dans un sens inversé, c'est à dire la relation pour (couleur 15 ,couleur 12) est (<,\) , puis 4 fois dans l'image 151, et enfin une fois inversée dans l'image 189.

Paires de couleurs des objets	***Identifiants des images***
12,15	11(+12)111(-3+2)151(+4)189(-1)
123 ,145	11(+5)45(-2)456(+13)
…	
C_i, C_j	$I_x(u), I_y(v), \ldots, I_z(w)$

Tableau 10 - Structure du fichier inversé pour les relations spatiales

Il est donc important de connaître le nombre de régions dominantes. En réalité, connaissant le nombre d'opérateurs codant les relations spatiales des régions d'une image, il est possible d'en déduire le nombre de régions dominantes de celle-ci. Ainsi si on trouve dans notre exemple 20 (N_O= 20) opérateurs dans la chaîne, cela veut dire que l'image comporte cinq régions dominantes (N_R =5). En effet, remarquons d'abord que la représentation de n nœuds nécessite S opérations spatiales où :

$$S = \frac{n(n-1)}{2}$$

Sachant que chaque relation spatiale nécessite 2 opérateurs l'un sur l'axe horizontal et l'autre sur l'axe vertical, on obtient le nombre d'opérateurs N_0 :

N_0=2*S = 2*n(n-1)/2

En développant cette formule, nous obtenons : $n^2 - n = N_O$ ou encore $n^2 - n - N_O = 0$. D'où le nombre de régions dominantes N_R est

$$N_R = \frac{1+\sqrt{1+4N_O}}{2}$$

L'algorithme suivant permet de réaliser ce calcul. Rappelons simplement que le signe – est utilisé pour indiquer le sens de la relation spatiale, il n'est donc pas pris en compte lors du comptage des opérateurs.

```
Procédure Calcul_NR
Début
        NR = 0 ;  // Nombre de régions
        NO = 0 ;  // Nombre d'opérateurs spatiaux
        I = 0 ;    // indice de lecture des caractères de la ligne
```

Pour chaque ligne Faire

Si ligne[I] <> '-' alors $N_O = N_O + 1$;

I= I +1 ;

Finpour

$$N_R = \frac{1+\sqrt{1+4N_O}}{2}$$

Fin

Algorithme de recherche par le vote

Le problème ici est de trouver les images les plus ressemblantes de la base, étant donné une image requête. Le principe est celui des votes. Pour l'image de requête, nous segmentons aussi les régions dominantes et en extrayons les caractéristiques visuelles ainsi que les relations spatiales régissant sur ces régions. Ensuite, il s'agit de vérifier pour chaque relation spatiale Γ entre deux régions, soit (C_i,C_j), si elle est également présente dans une image candidate. Si c'était le cas, nous incrémentons un compteur qu'est le vote de l'image par le nombre de sa fréquence. Ainsi, à la fin d'un tel processus les images qui ont un maximum de vote sont celles qui sont considérées comme étant les plus similaires. L'algorithme de vote est présenté dans la procédure suivante :

Procédure Vote ( )

Début

Pour chaque relation spatiale Γ entre deux régions homogènes par (C_i,C_j) de l'image de requête Faire

Début

Ouvrir Fichier Γ

Si $C_j < C_i$ Alors

couple = (C_j,C_i);

signe = '-' ;

Sinon

couple = (C_i,C_j) ;

signe = '+' ;

Finsi

Chercher la ligne qui contient le couple ;

Répéter

Si signe = '-' alors

Pour chaque image I qui possède un –V alors

Vote[I] = Vote[I]+V ;

Sinon

```
                    Pour chaque image I qui possède un +W alors
                        Vote[I] = Vote[I]+W ;
            Jusqu'à fin de ligne
            Fermer Fichier Γ
        Finpour
        Trier les votes des images par ordre décroissant
Fin
```

Le principe de vote tel que nous venons de présenter ne tient pas en compte de l'importance des régions ni des satisfactions partielles. On peut cependant améliorer facilement notre algorithme pour qu'il tienne compte d'autres caractéristiques comme la distribution ou la taille d'une région.

C. Implémentation du Prototype Web

Afin de tester la viabilité de l'ensemble de nos techniques pour la recherche d'images par le contenu, nous avons aussi réalisé une interface Web permettant la recherche d'images par similarité, à partir de zones prédéfinies ou à partir d'une image importée [AM98]. Ce prototype utilise donc les algorithmes que nous avons développés pour permettre d'insérer, rechercher et archiver les images dans une base de données. Les ingrédients d'une telle réalisation s'appuie notamment sur ODBC et JDBC pour la connexion à une base de données via une applet Java.

1. Présentation de l'interface utilisateur

Les deux schémas présentés ci-dessous présentent l'interface requête de l'utilisateur. Le premier représente une recherche d'images par similarité dans une base de données à partir d'une image importée par l'utilisateur, tandis que le deuxième représente le même type de recherche à partir d'une image dessinée par l'utilisateur.

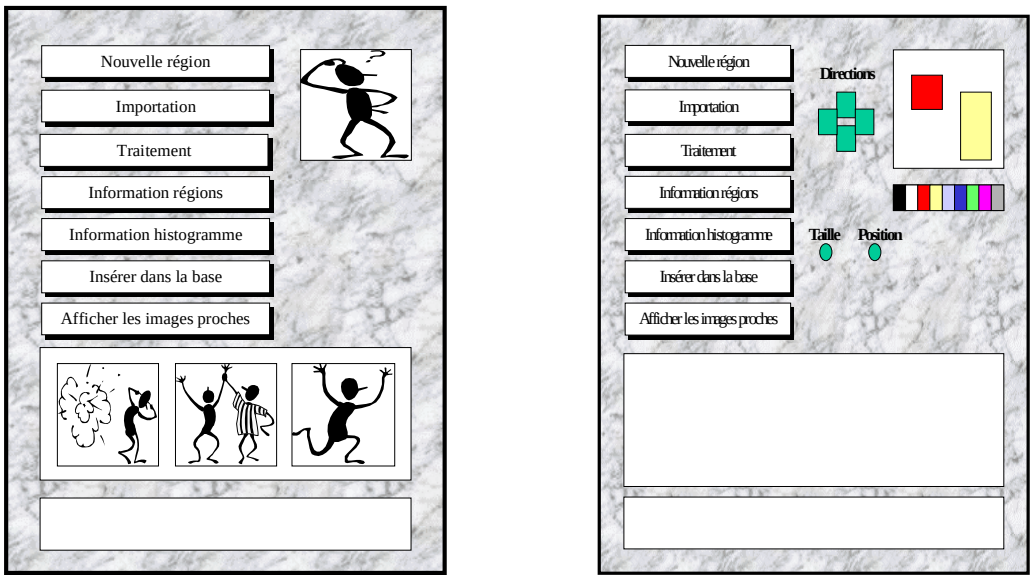


Figure 31 - Type d'interface utilisateur

Cette interface permet donc les fonctionnalités suivantes :

- Dessiner de nouvelles régions.
- Importer une nouvelle image via un explorateur.
- Exécuter un traitement permettant d'analyser l'image importée.
- Afficher l'histogramme global d'une image importée.
- Afficher les informations relatives aux régions d'une image importée.
- Insérer une image dans la base Access si elle n'existe pas déjà dans la base.
- Afficher les images similaires stockées dans la base et relatives à l'image importée.

2. Le fonctionnement général

Le schéma ci-dessous résume le fonctionnement général du prototype.

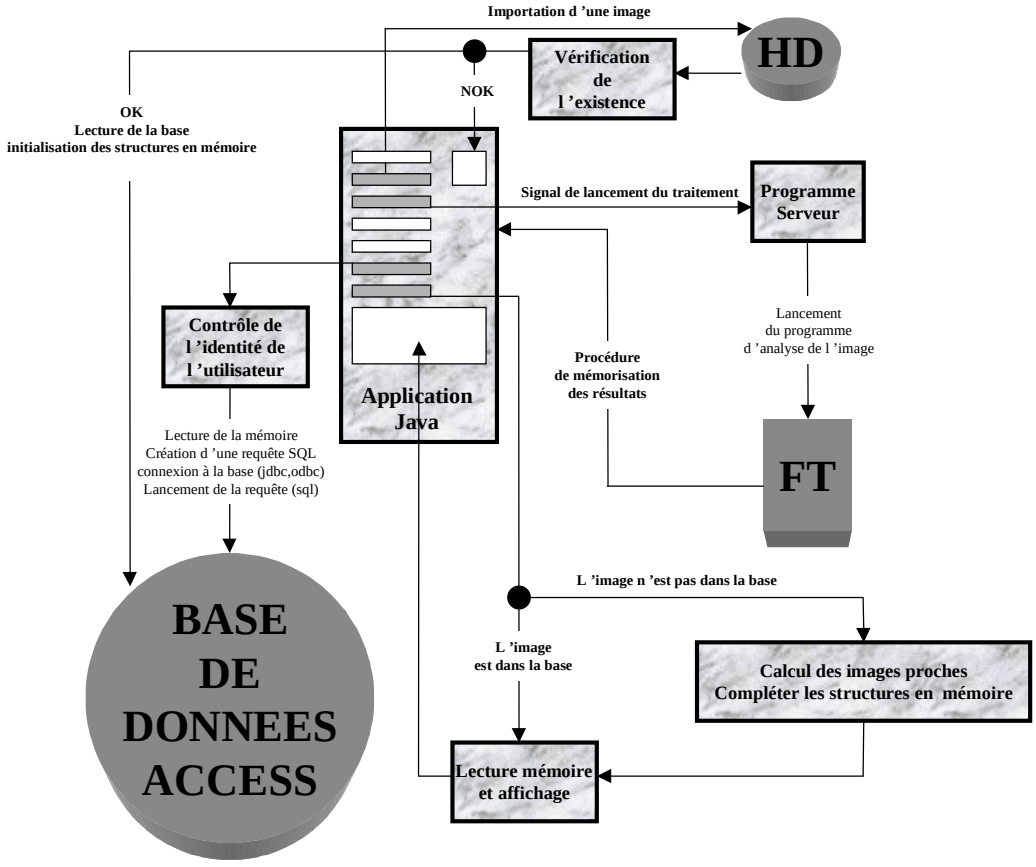


Figure 32 - Fonctionnement général du prototype

Comme nous pouvons le voir, notre prototype Web propose de réaliser les fonctionnalités de la manière suivante :

- L'utilisateur déclare une « *Nouvelle région* ». A chaque click sur ce bouton, une nouvelle région apparaît dans le cadre prévu à cet effet. L'utilisateur peut alors choisir la couleur de la zone, diminuer ou augmenter sa taille et la positionner n'importe où dans le cadre, à l'aide des boutons de commandes de *directions*.
- L'utilisateur « *Importe une image* ». Un explorateur d'arborescences apparaît pour aider à sélectionner une image particulière dans un classeur d'images. Une fois la sélection effectuée, on établit une connexion avec la base de données Access et on lance une requête afin d'initialiser les structures en mémoire relative à l'image importée, puis on charge l'image dans le cadre dédié à cet effet.
- L'utilisateur appelle « *Traitement de l'image* ». Ce bouton n'est accessible que si l'image requête ne provient pas de la base, et par conséquent il est nécessaire d'appliquer nos techniques pour réaliser une segmentation de régions homogènes et ses indices visuels.
- Un programme serveur, en attente d'un message, tourne en tâche de fond, parallèlement au browser et à l'application. Lorsque l'utilisateur sélectionne ce bouton, l'application Java (Applet) lance un message au programme serveur, qui lance le traitement de l'image en prenant cette dernière en paramètre d'entrée. Le résultat fournit par ce programme d'analyse est un fichier texte dans lequel figurent les informations concernant l'image (histogrammes et régions). Ce fichier texte est utilisé pour initialiser les structures en mémoire.
- L'utilisateur appelle « *Affichage de l'histogramme* ». Cette fonction affiche l'histogramme de l'image importée en s'appuyant sur le fichier texte généré à la suite du traitement précédent.
- L'utilisateur appelle « *Affichage des régions* ». De même cette fonction affiche les données relatives aux régions de l'image importée en s'appuyant sur le fichier texte généré à la suite du traitement précédent.
- L'utilisateur désire « *Insérer une image dans la base* ». Cette fonction est destinée à archiver les données de l'image importée dans la base de données Access. Cette fonction ne peut être exécutée que si l'image n'existe pas dans la base et si le traitement et le calcul des images proches ont été préalablement effectués. Ainsi, pour valider cette fonction il faut que les structures relatives aux informations de l'image importée (histogramme, régions, images proches, etc.) soient initialisées en mémoire. Cette fonction lit les structures mémoires, crée plusieurs requêtes SQL d'insertion, réalise une connexion à la base de données Access, et exécute les requêtes d'insertion sur les tables de la base. L'accès à la base de données est protégé par un login et un mot de passe.
- L'utilisateur désire « Afficher l*es images similaires* ». L'affichage des images similaires relatives à l'image importée se présente sous deux formes : Soit l'image requête est une image de la base, auquel cas les distances par rapport à d'autres images ont été déjà calculées ; il suffit alors de scruter les structures en mémoire pour obtenir la liste des images similaires, puis lancer une requête pour obtenir leur chemin d'accès et enfin charger les images. Soit l'image n'existait pas dans la base, auquel cas il faut alors appliquer l'algorithme de recherche par vote que nous avons présenté au paragraphe précédent, et on affiche les premières images de la base qui seront jugées plus proches de l'image requête.

3. Réalisation du lien Java-Access

L'un des objectifs de notre prototype est de permettre l'accès à une base d'images à partir d'un poste client sur lequel s'exécute un navigateur (Internet Exploreur, Netscape, etc.). Aussi, sommes nous naturellement tournés vers Java qui est considéré comme étant un langage de programmation pour les applications Internet. En effet, le langage Java propose des dispositifs de communication entre un poste client et un serveur d'applications, notamment l'interface « JDBC DriverManager » et des outils SQL permettant d'interroger ou modifier une base de données distante. D'un autre côté, il faut aussi une interface du côté Access jouant le rôle du pont pour la compréhension et le dialogue entre le serveur d'applications et le serveur de la base de données. Pour résumé, l'interface « ODBC DriverManager » et le pont « JDBC-ODBC Bridge » assurent ces deux fonctions de communiation.

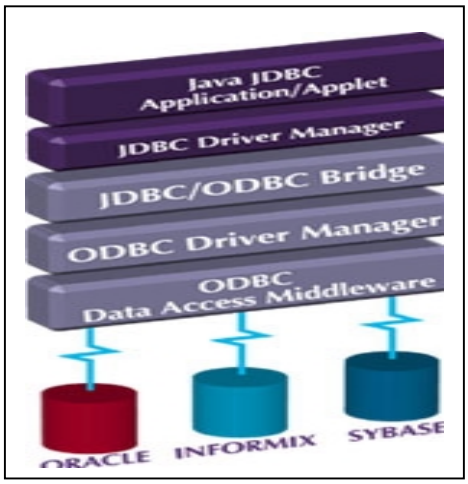


Figure 33 - Interconnexion Java-Access

Passerelle JDBC-ODBC

Java est une plate-forme puissante pour la programmation des applications client/serveur sur les réseaux du type Internet/Intranet. Du côté client il y a les "applets" qui sont des programmes résidents dans un serveur Web appelés par un client via un navigateur Web. Côté serveur, il y a les "servlets". Ce sont des applications sans interface graphique résidents et exécutées dans le serveur. Elles sont déclenchées en fonction des appels des clients. La figure 33 décrit l'ensemble des outils nécessaires à l'interconnexion Java-Access .

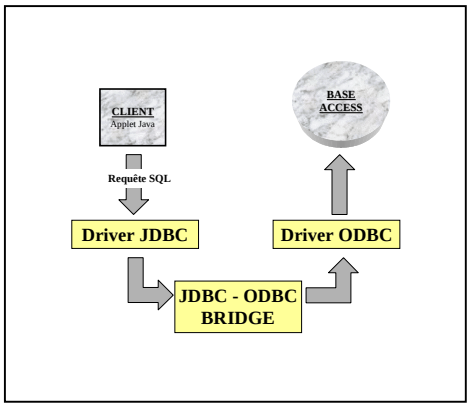


Figure 34 - Lien Java-Access

Interconnexion entre une applet Java et une base Access

De la requête SQL (recherche, insertion, modification), formulée depuis l'applet Java, à la base Access, les outils de communication utilisés sont des classes SQL (import java.sql.*), un driver JDBC (import sun.jdbc.odbc.*), un driver ODBC (cf. configuration du driver), le pont étant transparent à l'utilisateur. La figure 34 décrit le schéma d'une discussion Java-Access .

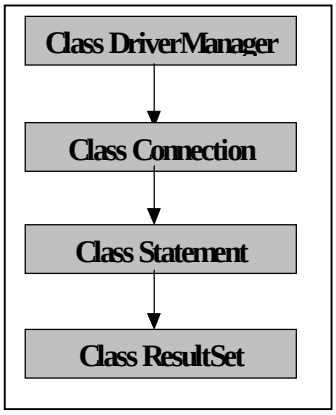


Figure 35 - Ordre de la hiérarchie

Configuration du driver JDBC

Pour configurer le driver JDBC il faut impérativement importer deux classes java : import java.sql.* et import sun.jdbc.odbc.*. Ensuite, il faut respecter une certaine hiérarchie dans les déclarations, telle celle présentée par la figure 35. En effet, pour pouvoir lancer une requête SQL à partir d'une applet, il faut avant tout instancier les classes Java dans le bon ordre. La classe Statement sert à exécuter la requête SQL et la classe ResultSet sert à récupérer et scruter les résultats.

Voici comment se présente le code dans une Applet Java :

Connection **cnx**;

Statement **stmt**;

ResultSet **rs**;

// Chargement du package Sun pour le pont JDBC-ODBC

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

// Initialisation de la connexion

Cnx = DriverManager.getConnection(DriverNameODBC,ODBC_Login,ODBC_Code);

// Initialisation d'un état pour le lancement des requêtes SQL

stmt = cnx.createStatement();

// Lancement de la requête « request » et récupération des résultats dans rs

rs = stmt.executeQuery(request);

Tout le travail de configuration de la base de données, d'établissement de la connexion, et d'initialisation des données utiles au commencement de l'application, est réalisé au chargement de l'applet. Notre base de données Access se compose de plusieurs tables concernant les images, les régions, les relations spatiales et d'autres.

D. Résultats expérimentaux

Nous avons expérimenté notre prototype sur une base de 1,500 images hétérogènes. Dans les exemples qui vont suivre, l'image de requête est toujours classée à la première colonne en haut à gauche. Les images qui ressemblent à l'image de requête sont classées de gauche à droite et du haut en bas. Ces images sont rangées par leur degré de similarité à l'image requête suivant une métrique donnée.

Cependant, nous avons besoin de critères de performances pour évaluer nos techniques de recherche d'images par la similarité [Jai93,JPP95]. Pour notre part, on juge la qualité d'une recherche d'images par deux mesures classiques 'Rappel' et 'Précision' qui sont largement appliquées dans les systèmes d'information textuelle. Ainsi, pour chaque image de requête, les images trouvées sont classées en deux catégories : groupe pertinent ou non pertinent. Dans notre cas de recherche d'images par la similarité, le rappel représente la proportion des images pertinentes dans la base qui sont retrouvés en réponse à une requête, alors que la précision représente la proportion d'images retrouvés qui sont pertinentes pour une requête [Jon81, Rij81].

D'une manière précise, soient A et B respectivement l'ensemble des images pertinentes et l'ensemble des images retrouvées. Si *a,b,c,* et *d* sont des ensembles d'images tels que c'est représenté sur la figure 36 c'est à dire que *a* représente le nombre des images pertinentes qui sont correctement retrouvées, *c* le nombre d'images pertinentes non retrouvées par notre prototype et *b* le nombre d'images qui sont retrouvées par erreur, les taux de rappel et de précision sont définies en terme de probabilité conditionnelle par les formules suivantes:

$$Rappel = P(B \setminus \mathrm{A}) = \frac{P(A \cap B)}{P(A)} = \frac{a}{a+c} = \frac{nombre\ \mathrm{d'images\ pertinentes\ retrouvées}}{\mathrm{nombre\ d'images\ pertinentes}}$$

$$\mathrm{Précision} = P(A \setminus \mathrm{B}) = \frac{P(A \cap B)}{P(B)} = \frac{\mathrm{a}}{\mathrm{a+b}} = \frac{\mathrm{nombre\ d'images\ pertinentes\ retrouvées}}{nombre\ \mathrm{d'images\ retrouvées}}$$

Dans les formules ci-dessus, a+c donne le nombre total des images pertinentes, tandis que a+b donne toutes les images considérées par le système comme proches à l'image de requête.

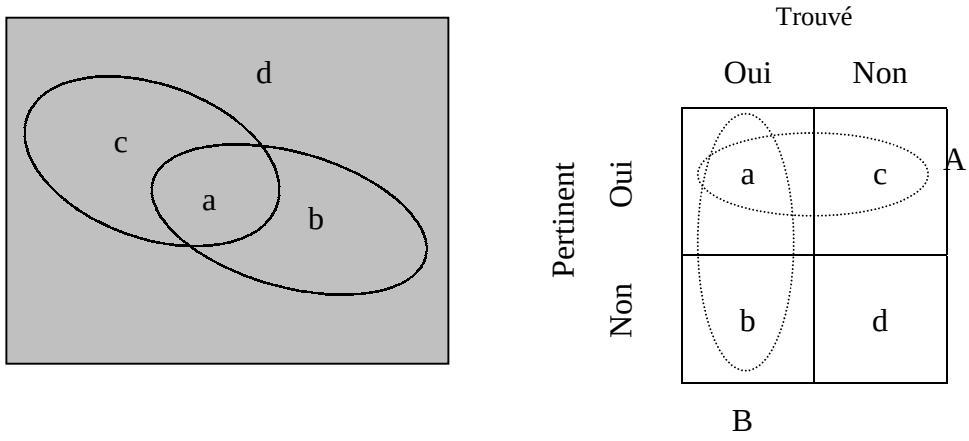


Figure 36 - Efficacité de la recherche

Dans la suite, nous allons présenter deux séries de test. Dans les deux séries d'expérimentation, on utilise la couleur comme critère principal d'homogénéité. La première série est réalisée sur des types d'images spécifiques que nous avons segmentées par notre méthode basée sur les quadtrees. Dans cette première série de test, seule l'information spatiale tenue en compte est le centre de gravité de régions homogènes. Les requêtes comportent aussi une combinaison de différentes caractéristiques visuelles.

Dans la seconde série de test, les images sont segmentées par l'autre méthode de segmentation basée sur la croissance de région. Les relations spatiales entre les objets d'images ont été évaluées par notre algorithme de vote. Les images ont été évaluées seulement sur la base de la couleur et des informations spatiales.

1. Première série de tests : segmentation par quadtree

Dans cette série de tests [CC98], nous avons évalué les résultats de notre méthode de segmentation, basée sur les quadtrees qui identifient les objets dominants, et d'indexation des images fixes en tenant compte des caractéristiques visuels, comprenant les histogrammes de

couleur, la texture et quelques formes simples. Nous avons aussi discuté quelques résultats expérimentaux en fonction des distances de similarités proposées.

L'espace HSV est utilisé pour représenter la base de 1500 images. Le système calcule les mesures de 'Rappel' et de 'Précision' pour chacune des images pertinentes qui sont ensuite aussi prises comme des images de requêtes à leur tour. Quatre distances D_1,D_2,D_3 et D_4, respectivement la distance de Manhattan, Euclidienne, d'intersection normalisée d'histogramme et quadratique, sont utilisées pour évaluer leur pertinence. Les mesures obtenues sont la moyenne des résultats, les images pertinentes étant évaluées par un expert. Afin de comparer la pertinence de ces quatre distances, nous avons lancé une série de requêtes sur trois sous-bases de la base d'images :

- Nature. Des images de la nature contenant des images à textures variables grossières, telles que les arbres le ciel, montagnes etc. Cette base a été utilisée pour répondre à des requêtes sur des textures globales.
- Globe. Des images de la terre, qui contiennent des formes circulaires pour des requêtes de type forme.
- Barres. Cette base contient des images synthétiques de barres de couleurs différentes avec plusieurs dispositions spatiales. Elle a été utilisée pour la comparaison des images sur la base de l'arrangement spatial du contenu. Les tests ont été réalisés seulement à partir des centres de gravité des objets dominants.

Les requêtes ont été formulées par une combinaison d'indices visuels. La signature de l'image basée sur la couleur a été calculée à partir des histogrammes. Celle de la texture est basée sur les premiers moments de l'histogramme. Quant à la forme d'une image, elle est basée sur la forme d'une région dominante de couleur donnée. Chaque région est encadrée par un RME, et ses descripteurs de Fourier sont extraits.

Couleur et Macro-texture

La première requête Q1 consiste à trouver les images de la base nature qui ressemblent à l'image de requête sur la base de deux indices visuels que sont la couleur et la (macro) texture. La figure 37 montre les 19 premières images trouvées. L'image de requête est la première image en haut à gauche. 40 images on été désignées comme étant images pertinentes. Le tableau 11 résume les résultats trouvés en terme de 'Rappel' et 'Précision'.

40 images de nature	D_1	D_2	D_3	D_4
Rappel	0,827	0,825	0,53	0,83
Précision	0,555	0,441	0,7	0,409

Tableau 11 - Rappel et Précision de la requête 1

Figure 37 - Résultat de recherche à partir de la couleur et une macro-texture

On remarque que les performances des distances D1,D2, et D4 sont très proches en rappel et en précision.

Couleur et Forme

La seconde requête consiste à rechercher les images à partir de leurs couleurs dominantes ainsi que de la forme de leurs principales régions. La figure 38 montre un exemple de résultats obtenus : Couleur dominante bleue et forme circulaire. 20 images sont considérées comme étant pertinentes. Le résultat est reporté sur le tableau suivant :

40 images de la terre	D_1	D_2	D_3	D_4
Rappel	0,871	0,770	0,769	0,857
Précision	0,423	0,5	0,51	0,582

Tableau 12 - Rappel et Précision de la requête 2

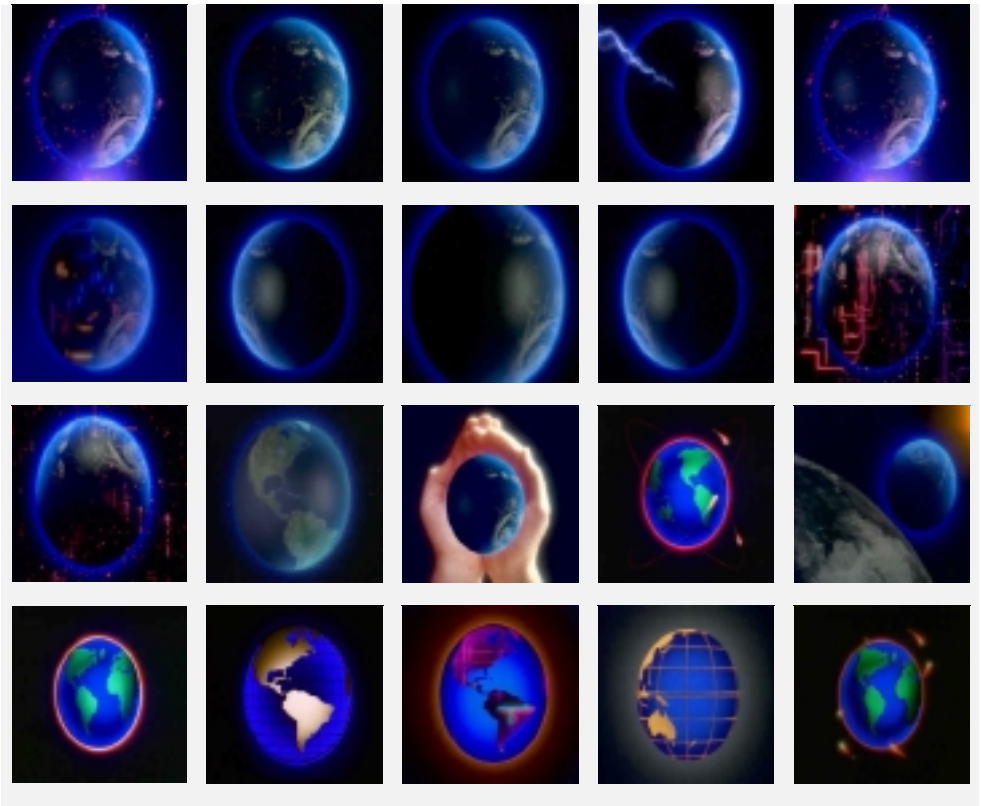


Figure 38 - Résultat de recherche à partir de la couleur et une forme circulaire

On constate que la distance quadratique D_4, est meilleure que D_3 dans les deux cas. La distance quadratique donne une meilleure performance que la distance euclidienne. Cependant, pour l'efficacité D1 est légèrement meilleure que les autres distances.

Couleur et Arrangement spatial

La troisième requête consiste à trouver images suivant leurs couleurs dominantes et le positionnement de celles-ci (régions dominantes) . La figure 39 montre quelques résultats concernant ce type de requête. Les couleurs dominantes des premiers objets dominants sont le noir, le rouge et le bleu. Le bleu se trouve sur le rouge.

40 images de barres	D_1	D_2	D_3	D_4
Rappel	0,755	0,756	0,754	0,761
Précision	0,629	0,532	0,562	0,4

Tableau 13 – Rappel et Précision de la requête 3

Les performances des métriques D_1, D_2 et D_3 sont proches. Elles ont fournit de bons résultats, sept en moyenne sur les dix premiers trouvés.

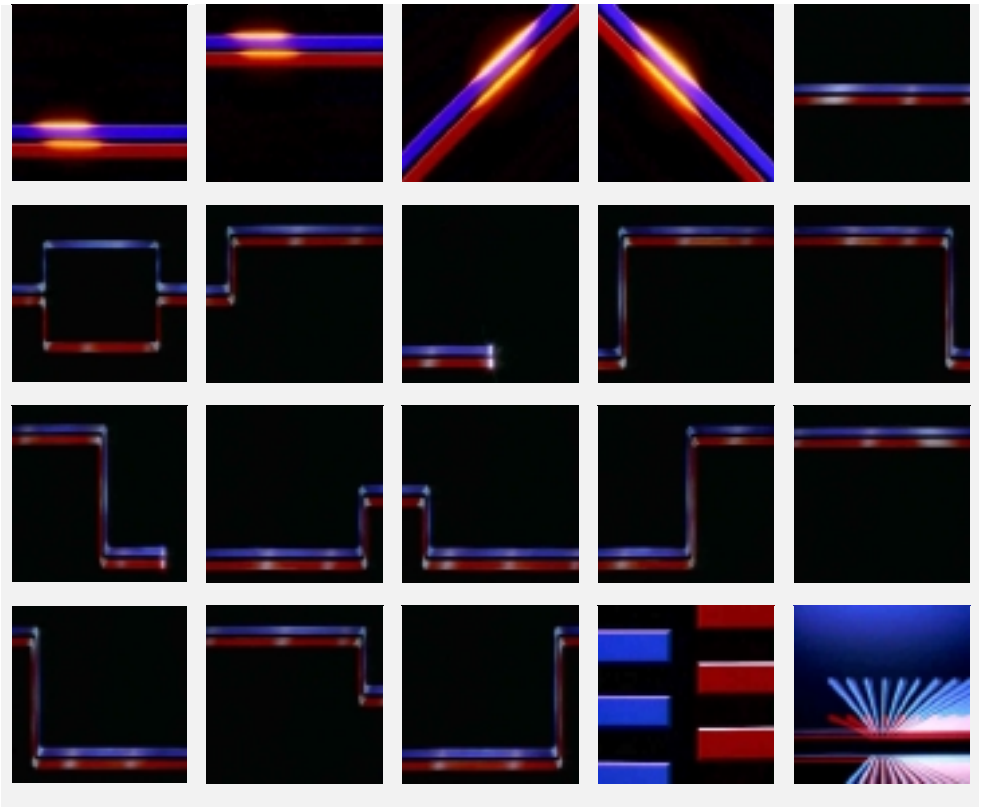


Figure 39 - Résultat de recherche à partir de la couleur et l'arrangement spatial des 3 premiers objets dominants

Conclusion

En conclusion, nous avons pu constater que les histogrammes de couleurs associés à d'autres caractéristiques visuels tels que la texture et la forme fournissent de bons résultats. Aussi, en tenant compte de la position des caractéristiques visuels et leur arrangement dans l'image, on obtient encore de meilleurs résultats. Quant au choix des distances, la distance quadratique reste la meilleure, mais compte tenu du temps de calcul, on lui préfère la distance d'intersection . Nous avons pu constater aussi que quand 'Rappel' augmente, 'Précision' baisse.

2. Deuxième série de tests : segmentation par la croissance de région

Dans cette expérimentation, nous avons voulu voir l'influence d'informations spatiales sur les résultats. L'évaluation des performances est toujours effectuée en terme de rappel et précision qui sont représentés par les courbes de la figure 43. La courbe Type I correspond aux résultats de recherches quand seulement les histogrammes de couleur sont utilisés, ne prenant donc pas les informations spatiales. La courbe Type II correspond aux résultat de la recherche quand les couleurs des objets dominants sont utilisées. Tandis que la courbe Type III représente les résultats de la recherche quand les relations spatiales sont utilisées et évaluées par notre algorithme de vote. Les résultats de recherche sur la base uniquement d'histogrammes de couleurs ne sont affichés.

La première requête, illustrée par la figure 40, cherche les images de la base ayant un nombre maximum de couleurs dominantes, ne prenant donc en compte les informations spatiales. La deuxième requête, dont les résultats sont illustrées par la figure 41, affine la première en tenant compte des arrangements spatiaux d'objets dominants, ici le ciel sur les montagnes sur l'eau. Enfin, dans la troisième requête, l'utilisateur choisit deux objets A pour la pelouse et B pour le cheval comme image de requête. Les résultats, prenant en compte la disposition spatiale B est dans A, sont illustrés par la figure 42.

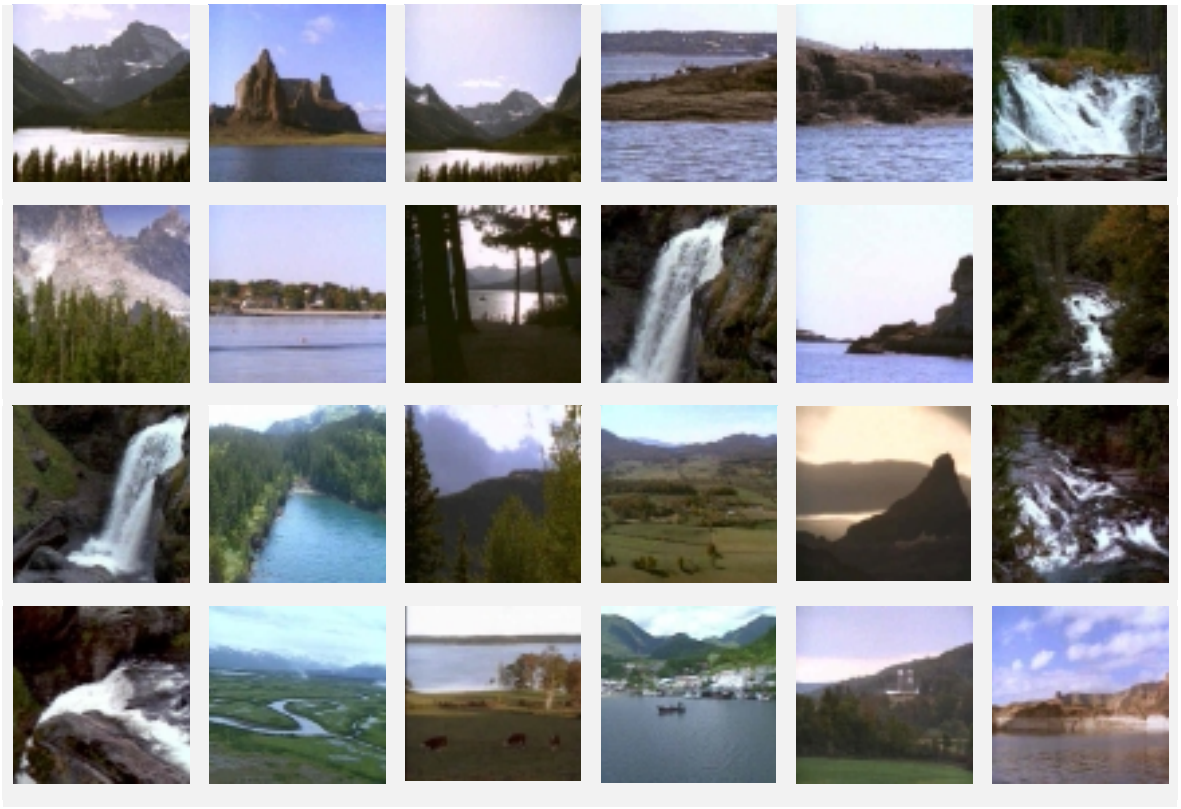


Figure 40 - Résultats de recherche basée sur les couleurs dominantes

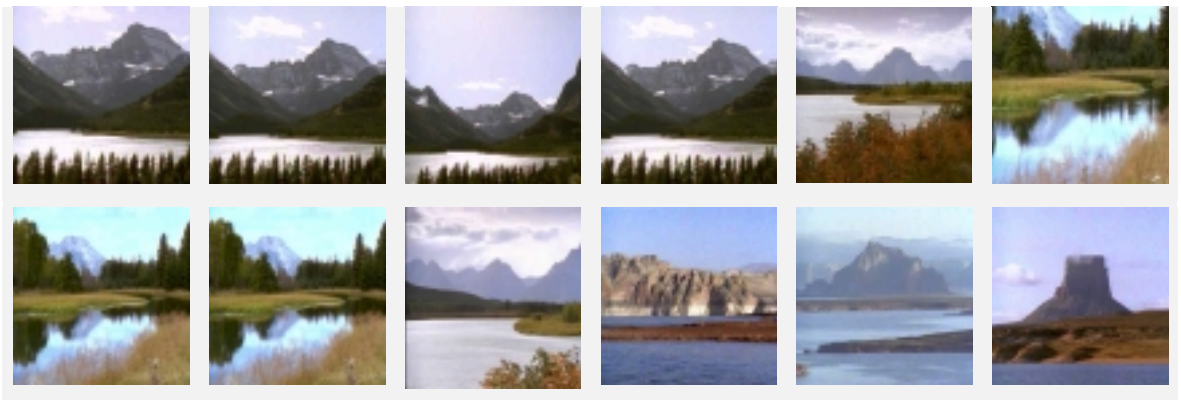
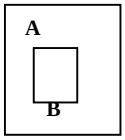
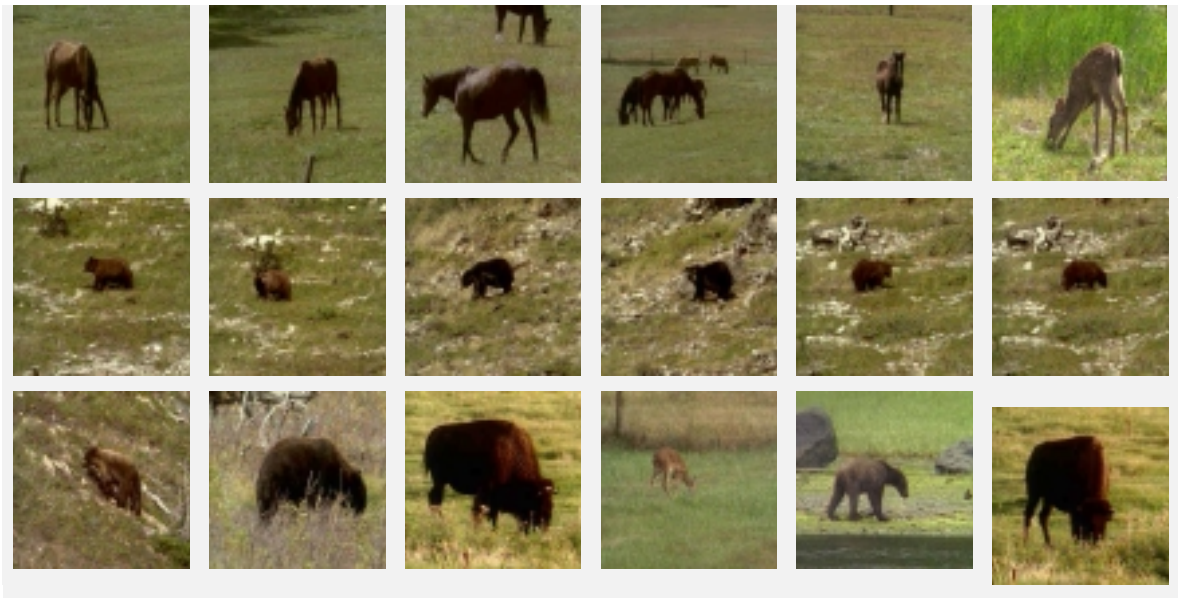


Figure 41 – Résultat de recherche automatique globale basée sur l'arrangement spatial des objets dominants



Le but est de chercher dans la base les images qui ont la même disposition spatiale que l'image de requête. La requête est formulée à partir des rectangles d'encadrement.



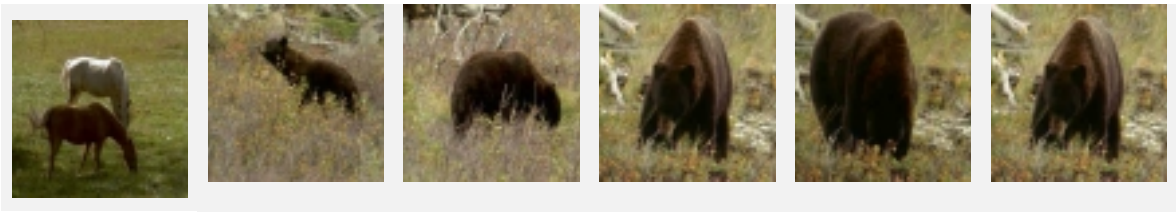


Figure 42 - Résultat de recherche basée sur l'arrangement spatial des objets dominants A et B

Conclusion

D'après les résultats présentés dans la figure 43 on remarque que la recherche à partir des couleurs d'objets dominants (Type II) donne de meilleur résultats que les requêtes basées uniquement sur les histogrammes de couleur (Type I). Aussi, nous avons remarqué que la recherche combinée à partir des histogrammes de couleurs et des relations spatiales des objets dominants (Type III) en utilisant notre algorithme de vote donne encore de meilleurs résultats que les deux autres. Dans les 100 requêtes utilisées, au pire des cas, nous avons eu un rappel de 0.6 et une précision de 0.8.

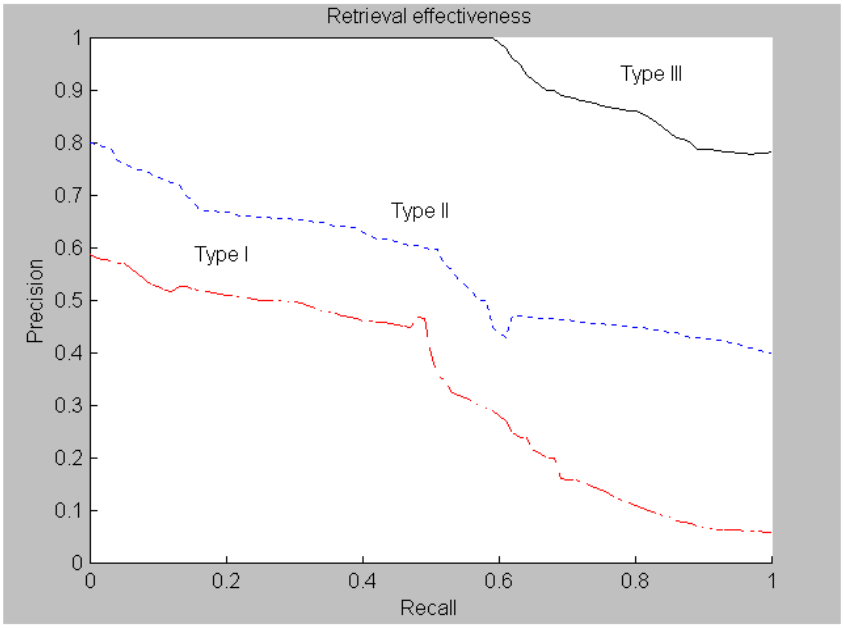


Figure 43 -Evaluation de la recherche en terme de précision et de rappel

VI. Images animées

Nous venons donc de présenter aux deux chapitres précédents un ensemble de techniques que nous avons développées pour l'indexation d'images fixes. Or une vidéo est une succession continue d'images fixes à laquelle on accède séquentiellement. Aussi, avons nous voulu appliquer les techniques sur les images fixes que nous avons étudiées aux chapitres précédents pour l'indexation de la vidéo. Dans ce chapitre, nous commençons par une description de la structure de la vidéo afin de définir quelques problématiques dans le domaine de l'indexation par le contenu de la vidéo. Ensuite, après un rapide tour d'horizon des travaux existants, nous présentons nos contributions sur la segmentation de la vidéo en plans, l'extraction de l'image représentative et la classification des plans en vue de former des clusters, chaque méthode que nous avons développée étant illustrée des résultats d'expérimentation sur le corpus de l'INA.

A. La structure de la vidéo

Une vidéo telle qu'un film de fiction ou un reportage à la télévision a une structure narrative et une organisation. La structure narrative est définie par les unités hiérarchiques que sont les séquences narratives, les scènes, et les plans. Un *plan* est une séquence d'images filmées sans interruption temporelle, ou une portion de film comprise entre deux collures [Chan94,Aum88]. Ils constituent une unité fondamentale pour la manipulation (production, représentation et indexation) de la vidéo. Une scène est une unité sémantique en général composée de plusieurs plans. Elle se déroule généralement dans un même intervalle temporel continu et un même environnement avec les mêmes acteurs. Au plus

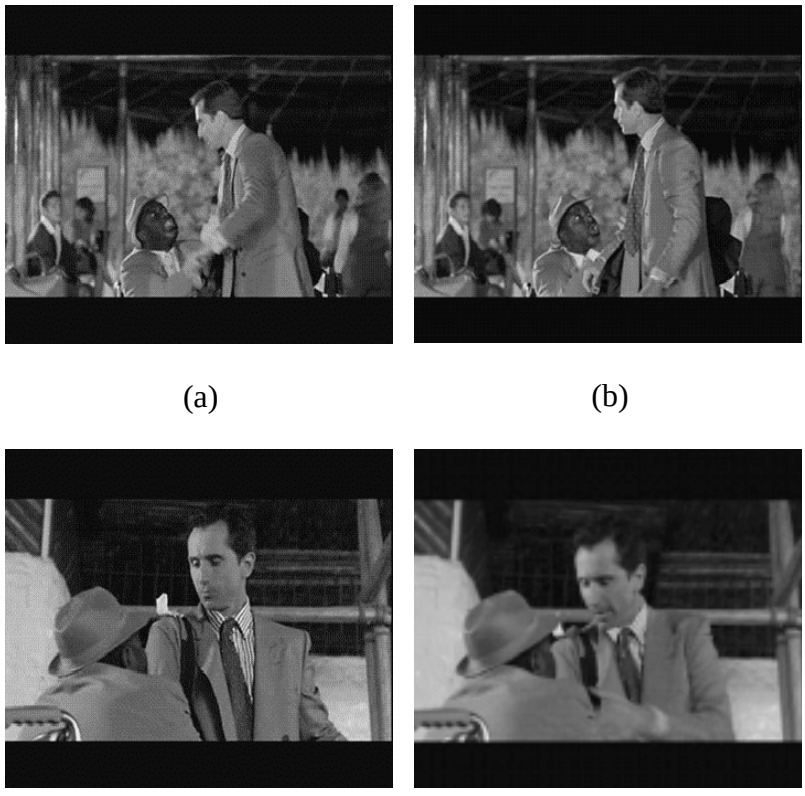


Figure 44 - Un Cut : les images a et b dans un premier plan, c et d dans un second plan

haut niveau sémantique, nous retrouvons les séquences narratives. Une *séquence narrative* est une unité narrative composée de plusieurs scènes reliées entre elles par leur force émotionnelle et narrative. Enfin, le scénario ou le script d'un film décrit une histoire sous la forme d'une suite de séquences narratives.

Le montage d'un film consiste à éditer celui-ci, à couper et à coller les prises de vue afin de donner de la meilleure manière possible le rendu du scénario. Durant le montage, deux séquences peuvent être reliées entre elles par un effet de transition comme par exemple un *fondu enchaîné* ou un *volet*. Les plans sont séparés des uns des autres par les *cuts*.

Le *cut* est signalé quand un brusque changement entre deux images successives est détecté, comme l'est illustrée par la figure 44. L'objectif est de montrer une action continue se déroulant dans plusieurs lieux (les caméras d'alors étant fixes). Ce type de montage est le plus fréquent. Son utilisation dépend plus ou moins de la nature du document, mais il est vraisemblable que le *cut* représente largement plus de 95% des transitions dans les documents d'origine cinématographique. Par contre, la réalisation actuelle des émissions de télévision tend à réduire cette proportion à 60% du nombre total des effets de transition.

Un *fondu enchaîné* se produit lorsque les images du premier plan deviennent de plus en plus faibles, superposées progressivement puis remplacées complètement par les images du second plan. Un *fondu au noir* apparaît lorsque la luminosité d'images du premier plan change graduellement pour devenir enfin une image noire. Un *volet* se produit lorsque les images du deuxième plan remplace celles du premier selon une forme bien précise, comme par exemple une ligne à partir de la gauche de l'image. L'effet de transition progressif (fondus enchaînés, fondus au noir, volets, etc.) représente une marque syntaxique dans le document, interprétée de façon quasi-inconsciente par le spectateur (un peu comme la mise en forme d'un paragraphe dans un texte). Par exemple le fondu au noir apparaît dans les documents le plus souvent pour marquer un changement thématique important ou une rupture majeure dans la narration. Les effets de transition progressifs peuvent aussi intervenir dans des cas variés, isolement pour marquer une discontinuité entre les plans par exemple [AJL95].

B. Problématique

L'objectif ici est de briser le caractère opaque d'une vidéo et d'aboutir à une structuration de celle-ci en des segments sémantiques, basés sur le contenu, pour accéder à la vidéo autrement que par un parcours séquentiel. Placé sous une telle perspective, le découpage d'une vidéo en différentes plans n'est pas suffisant. En effet, la représentation d'une vidéo par ses plans ne permet pas de décrire le contenu narratif de celle-ci. En plus, compte tenu de la durée de chaque plan, en moyenne trois seconds, la segmentation en plans d'une vidéo de 90 minutes peut fournir 1800 plans [Roh90], rendant l'accès encore difficile. D'ailleurs beaucoup de plans ont un contenu similaire. Aussi, un travail supplémentaire consiste à réorganiser les plans d'une vidéo en des unités sémantiques plus significatives, appelées scènes ou séquences, l'idéal étant d'aboutir à un storyboard d'une vidéo qui, par une arborescence hiérarchique comme l'illustre la figure 45, résume le contenu sémantique de celle-ci.

Dans la suite, après une rapide étude de l'état de l'art, nous allons utiliser les techniques que nous avons étudiées et développées sur les images fixes pour traiter quelques problèmes classiques dans le domaine de l'indexation de la vidéo : la segmentation de plans, l'extraction d'images clés, la classification de plans en vue de former des unités sémantiques plus macroscopiques, ici des clusters. Chaque technique que nous allons développée sera aussi expérimentée sur le corpus INA.

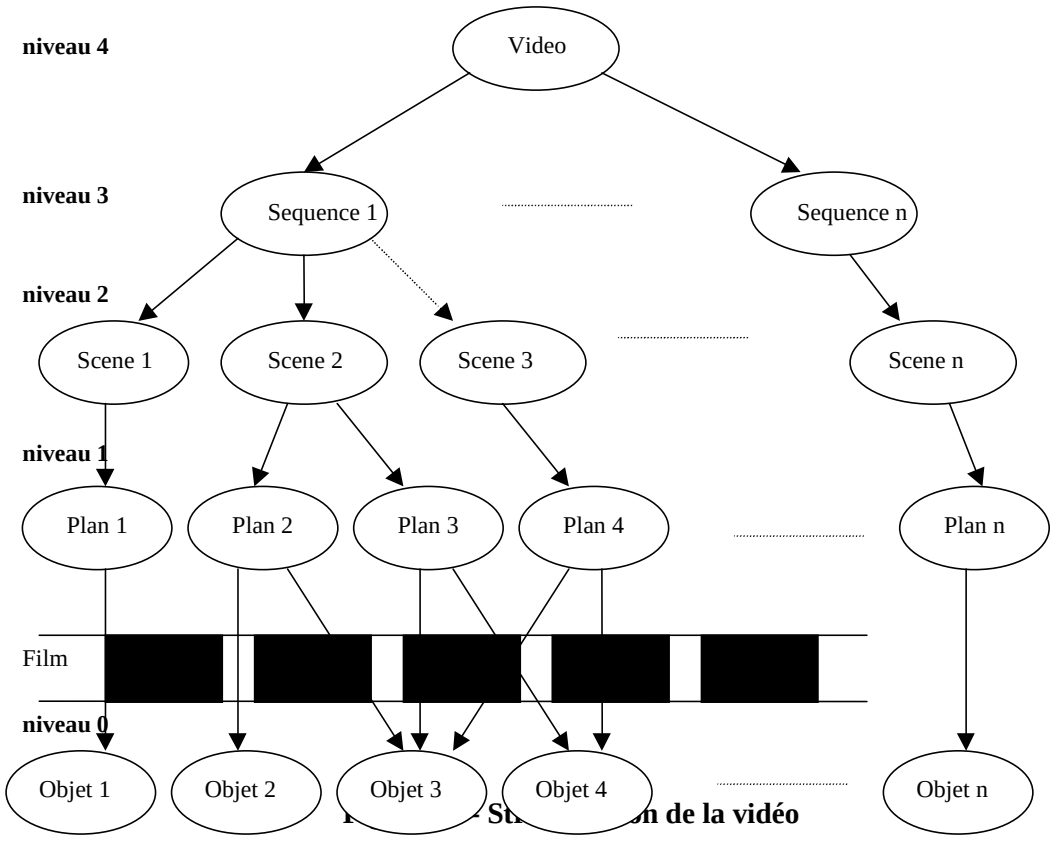


Figure 1 : Structuration de la vidéo

C. Un petit tour d'horizon

Beaucoup de travaux ont été réalisés déjà dans le domaine de l'indexation par le contenu de la vidéo. On peut trouver à ce sujet un état de l'art dans [AZP96]. Aussi, allons nous limiter notre petit tour d'horizon aux trois aspects de l'indexation de la vidéo qui nous intéressent, à savoir la segmentation de plans, l'extraction d'images clés ou représentatives et la classification de plans en vue de former des scènes.

Les indices utilisés sont très divers. On trouve des techniques utilisant des images non segmentées, et calculant donc le plus souvent des indices globaux. D'autres se servent d'images préalablement segmentées. Les primitives utilisées sont variées: points, contours, droites, courbes. Ces éléments peuvent alors être caractérisés par des invariants algébriques. Ces invariants sont placés dans une table de hachage et servent directement à la reconnaissance [LG96]. D'autres techniques essaient de tenir compte aussi de l'information photométrique en calculant des invariants utilisant les niveaux de gris autour de points particulier[Sch96]. D'autres techniques sont basées sur le mouvement et sa caractérisation. La prise en compte du mouvement a d'abord concerné le segmentation d'une vidéo en éléments élémentaires [Gro97]. Ont été exploitées des techniques de différence d'images plus ou moins améliorées, et plus efficacement des différences d'histogrammes appropriées [ZKS93].

1. La segmentation de plans

La première étape de la structuration consiste à découper une vidéo en unités temporelles de base que sont les plans. Le principe est le suivant. On associe généralement un signal continu au flux d'images de la vidéo, et un changement brutal de ce signal indique un cut, ou un changement de plans. Les premières méthodes s'appuient sur les histogrammes de couleur des images successives pour produire ce signal continu [AJ94]. Cependant, d'autres indices, appelés parfois aussi primitives, peuvent aussi être utilisés, comme par exemple les points, les contours, les droites, etc. Les différentes méthodes peuvent être évaluées par un taux de fausse détection et un taux d'omission. Le taux de fausse détection mesure le pourcentage de changements de plan faussement annoncés par la méthode, alors que le deuxième taux mesure le pourcentage du nombre de plans non détectés.

P. Aigrain et P. Joly proposent dans [AJ94] une méthode basée sur un modèle statistique de variation d'intensité des pixels pour mettre en évidence les deux effets de transition qui sont les coupures et les fondus. Lors d'une coupure, il y a une forte variation de l'intensité des pixels, alors que lors d'un fondu, il existe une proportion de pixels dont l'intensité varie faiblement. Entre deux images successives, I_i et I_{i+1}, pour N le niveau maximum de gris (256), N/2 est considéré comme le seuil au delà duquel il y a forte variation et N/6 le seuil en dessous duquel il y a faible variation. Ils calculent le nombre total de pixels $DP1_i$ qui varie fortement et le nombre total de pixels $DP2_i$, qui changent faiblement. Le cut et les transitions de type fondu et volet sont détectés sur les courbes de $DP1_i$, $DP2_i$ en exploitant le même modèle statistique et en appliquant un filtrage temporel pour éliminer les fausses alarmes dues aux mouvements importants.

Salazar et Valero implémentent cette méthode sur les composantes DC pour le format JPEG Movie [SV95]. Xiong et al. [XLI95]proposent une méthode similaire pour la détection du cut en utilisant un pas de 20 pixels entre deux images sans coupure pour ne pas tenir compte d'un mouvement maximum de 10 pixels observé lors du mouvement des objets ou de la caméra.

Ueda et al. [UMY92]. proposent une méthode basée sur la corrélation des couleurs entre deux images consécutives analysées par blocs. Cette mesure de corrélation est définie par :

$\Delta(k) = \rho_k - \rho_{k-1}$. avec ρ_k le taux de corrélation entre la $k^{ième}$ et la $(k+1)^{ième}$ image qui est définie par la formule suivante:

$$\rho_k = \frac{\sum_{x,y} (I_{k,x,y} - m_k).(I_{k+1,x,y} - m_{k+1})}{\sqrt{\sum_{x,y} (I_{k,x,y} - m_k)^2.(I_{k+1,x,y} - m_{k+1})^2}}$$

où :

$I_{k,x,y}$ est l'intensité (ou le niveau de gris) du pixel situé en (x,y) de l'image k, et m_k est la moyenne des intensités des pixels de l'image k.

Akutsu et al. [Aku+92] proposent une méthode par appariement de blocs avec compensation de mouvement. Le calcul de la corrélation normalisée est effectuée pour des blocs voisins offrant le meilleur appariement dans les deux images successives. Le calcul est effectué par la formule ci-dessus. Le cut est détecté lorsque le taux de corrélation moyenne est supérieur à un certain seuil.

Pour différencier les changements dus au cut de ceux dus aux mouvements de caméra et/ou d'objets, le taux de changement de corrélation est utilisé plutôt que le changement de magnitude de corrélation. Kasturi et Jain [KJ91] présentent une métrique basée sur les

caractéristiques statistiques des intensités des blocs de l'image. Les images sont découpées en blocs et le taux de ressemblance (likelihood ratio) TR_i entre deux images successives est évaluée. Les blocs correspondants dans deux images successives I_i et I_{i+1}sont comparés par la formule suivante:

$$TR_i = \frac{\left[\frac{\mu_i + \mu_{i+1}}{2} + \frac{\sigma_i^2 - \sigma_{i+1}^2}{2}\right]^2}{\sigma_i^2 \times \sigma_{i+1}^2}$$ où μ_i et μ_{i+1} sont les moyennes des intensités et σ^2_i et σ^2_{i+1} sont les variances des intensités des pixels des blocs correspondants des images I_i et I_{i+1}.

Le cut est détecté quand le nombre total des blocs (Nb_i) dépasse un seuil S_N .La formule s'écrit:

$$Nb_i = \sum_n b\left(TR_i - S_T\right) > S_N$$

b(x) est une fonction qui rend 1 quand x est positive et 0 dans le cas contraire. b() n'est positif que si le taux de ressemblance dépasse un seuil S_T.

Le principe des méthodes utilisant les histogrammes d'intensité ou de couleur consiste à calculer la différence entre les histogrammes h_i et h_{i+1} de deux images successives[NT91, ZKS93, HJW95a, CB95, DK95]. [YL95] et [She97] travaillent directement dans le domaine comprimé.

La distance est en général une des distances utilisées en images fixes appliquée soit à chaque composante de la couleur (par exemple R,G et B) soit l'histogramme global (notamment pour l'intensité, ou pour un histogramme de couleurs fixes).

Nagasaka et Tanaka [NT91] étaient les premiers à proposer les deux distances suivantes

Distance en valeur absolue: Cette mesure sert à calculer la variation des intensités des pixels ou de leurs couleurs

$$\Delta_i = \sum_j \left|H_i(j) - H_{i+1}(j)\right|$$

Test du χ^2 : Cette mesure amplifie les variations des cuts .

$$\Delta_i = \sum_{j=1} \frac{\left|H_i(j) - H_{i+1}(j)\right|^2}{\left(H_{i+1}(j)\right)^2}$$

Yeung et al [Yeu+95] propose l'utilisation d'une mesure d'intersection des histogrammes qui est définie comme suit:

$$\Delta_i = \sum_j \min\left(H_i(j), H_{i+1}(j)\right)$$

Dailianas et al. présentent dans |DAE95] une comparaison des méthodes basées sur les histogrammes, y compris leurs méthode basée sur les histogrammes égalisés (Différence en valeur absolue des histogrammes égalisés) .

Zhang et al. [ZKS93] détectent l'effet de transition du type fondu en utilisant un seuil S_t qui est situé juste au dessus de la valeur moyenne des Δ_i obtenue en dehors des effets de transition. Si, pendant une durée *k*, les valeurs des Δ_i franchissent le seuil S_t et si la formule

suivante est satisfaisante, alors on en déduit la présence d'un fondu. M est le point de début d'un fondu. Cette méthode nécessite deux balayages du document pour l'ajustement des seuils. Elle ne fonctionne pas en temps réel.

$S_t + \sum_{i=m}^{m+k} (\Delta_i - S_t) > S_c$ S_c est un seuil fixe pour la détection de plan.

Au lieu de travailler sur les histogrammes des niveaux de gris, on peut travailler sur des histogrammes de couleur. Zhang et al [ZKS93] quantifient l'ensemble des couleurs possibles en ne conservant que trois bits significatifs sur chaque canal pour se ramener à une taille d'histogramme plus en rapport avec la définition de l'image et donc la pertinence des données à produire. Gargi et al. [Gar+95] rapportent le résultat des comparaisons dans les différents espaces de couleur. Selon eux, la précision décroît dans l'ordre suivant: YIQ, L*a*b* suivi par HSV et L*u*v* et finalement RGB. Le coût de calcul pour les conversion croît dans l'ordre suivant: RGB, HSV, YIQ, L*a*b* et L*u*v*.

D'autres méthodes proposées utilisent le nombre de vecteurs de mouvements de blocs et exploitent les phénomènes suivants [Kim97]: Quand une image contient beaucoup de blocs intra-codés, cela signifie que la probabilité est forte d'avoir un cut entre une image précédente et l'image courante. Si, pour une image, la majorité de blocs sont codées par la prédiction de mouvement en avant (resp. en arrière), alors il y a une forte probabilité de cut entre l'image suivante (resp. précédente) et l'image courante.

Pour la détection du fondu, Meng et al. [MJC95] cherchent un segment temporel dans lequel la courbe de variance, obtenue par les coefficients DC, a la forme d'une parabole selon le modèle de l'évolution temporelle de la variance entre les images initiale et finale pour un fondu:

$V(t) = (V_i - V_f)\lambda^2(t) - 2 V_i\lambda(t) + V_i$

$\lambda(t) = (t - t_i) / (t - t_f)$

où V_i ,V_j, t_i et t_j et sont les variances et les positions temporelles des images initiale et finale du fondu.

Sethi et Patel [SP95]calculent sur les trois images consécutives I_i, I_{i+1}, I_{i+2}, les valeurs de similitude $S_{i,i+1}$ et $S_{i+1,i+2}$ pour les paires d'images I_i, I_{i+1}et I_{i+1}, I_{i+2}. La cohérence des mouvements de l'observateur (Cmo) est définie comme suit pour la mesure de détection du cut [KIM97]:

$$Cmo(i, i+1, i+2) = \left| \frac{S_{i,i+1} - S_{i+1,i+2}}{S_{i,i+1} + S_{i+1,i+2}} \right|$$

En cas de cut, la valeur de Cmo tend vers 1, sinon vers 0. La similitude entre deux images ,$S_{i,i+1}$, est calculée par l'énergie de différence normalisée (Edn) ou par la différence absolue au taux de somme (Dats).

$$Edn = \frac{\sum_{x,y} [I_i(x,y) - I_j(x,y)]^2}{\sum_{x,y} I_i^2(x,y) \times \sum_{x,y} I_j^2(x,y)} \quad , \qquad Dats = \frac{\sum_{x,y} [I_i(x,y) - I_j(x,y)]}{\sum_{x,y} [I_i(x,y) + I_j(x,y)]}$$

Arman et al. [AHC93a, AHC93b] proposent une technique pour la détection de plans en JPEG en utilisant les coefficients de DCT. La présence d'un cut est déterminée quand la valeur obtenue par la formule suivante dépasse un certain seuil.

$$\psi = 1 - \frac{|V_i . V_j|}{|V_i||V_j|}$$

où V_i et V_j sont les vecteurs de coefficient des images successives i et j.

Zhang et al. [Zha+94,Zha+95] présentent une technique de comparaison qui est basée sur la différence absolue normalisée au bloc (i,j) qui est définie par la formule suivante:

$$d(I_u, I_{u+1}, i, j) = \frac{1}{64} \sum_{k=1}^{64} \frac{|c(I_u, i, j, k) - c(I_{u+1}, i, j, k)|}{\max(c(I_u, i, j, k), c(I_{u+1}, i, j, k))}$$

où $c(I_u,i,j,k)$ est le kème coefficient du bloc (i,j) dans l'image I_u.. Si la différence $d(I_u,I_{u+1},i,j)$ est supérieure à un seuil, le bloc (i,j) est considéré comme changé. Si le nombre de blocs qui ont changé est supérieur à un autre seuil, alors un cut est déclaré.

Liu et Zick [LZ95] présentent une technique dans le domaine comprimé basée sur le nombre de vecteur de mouvement et du signal d'erreur. Un cut entre deux images augmente l'énergie de l'erreur. Cette énergie fournit une mesure de similarité entre l'image I_k et l'image I_{k+1} compensant le mouvement est définie par:

$$d(I_k, I_{k+1}) = \frac{\sum_{i=1}^{F_p} E_i}{F_P^2}$$

où E_i est l'énergie d'erreur du macrobloc i et F_p est le nombre des macroblocs prédits en avant. Un cut est déclaré si la différence entre F_p et le nombre de macroblocs prédits en arrière change du positif au négatif.

Zabih et al. [ZMM95] proposent une méthode qui tient des contours de l'image courante. Ils calculent le taux du nombre des pixels de contour qui apparaissent loin des contours de l'image précédente, et le taux du nombre des pixels de contour qui disparaissent. Le cut est détecté quand il y a un maximum local qui dépasse un seuil dans l'évolution temporelle de la valeur maximum des deux taux. Les transitions des effets de fondu et de volet sont détectées par l'analyse de ces valeurs. Pour compenser les mouvements de caméra, les images sont alignées avant le calcul.

Arman et al. proposent la comparaison des moments géométriques qui sont invariants pour les transformations linéaires et le changement d'échelle [DAE95].

Bouthemy et Ganansia proposent dans [BG96] une approche qui permet de détecter efficacement des changements de plans et des transitions progressives à l'aide de test statistique qui est basé sur une estimation robuste du mouvement dominant apparent entre deux images successives. Un test de Hinkley avec $\xi = n_d / n_0$ est utilisé pour la détection des changements de plan:

$$S_k = \sum_{t=0}^{k} (\xi - m_0 + \frac{\delta_{min}}{2}) \quad (k \geq 0)$$

$M_k = \max(S_i)\ 0 \leq i \leq k$; détection si $(M_k - S_k > \alpha)$

$$T_k = \sum_{t=0}^{k} (\xi - m_0 - \frac{\delta_{min}}{2})$$

$N_k = \min(T_i)\ 0 \leq i \leq k$ détection si $(T_k - N_k > \alpha)$

où m_0, δ_{min} sont une moyenne empirique et l'amplitude minimale de saut.

Mann et Picard [MP95]détectent un cut quand les paramètres de mouvement de caméra du modèle projectif n'est pas obtenu après dix itérations.

M. Ardebilian et al [Ard+96] proposent d'utiliser les points de fuite d'une image pour détecter un plan. Ces indices 3D sont déterminés par une double application de la transformée de Hough. Ses résultats donnent des résultats sensiblement meilleurs que celles basées sur les histogrammes car la méthode est insensible au changement brutal d'éclairage.

Pour résumé, les méthodes peuvent être divisées en trois classes: les méthodes de comparaison basées sur les pixels ou les blocs de pixels, des méthodes de comparaison d'histogrammes d'intensité ou de couleur, et les méthodes utilisant des coefficients de DCT dans les séquences MPEG[Gün97]. On trouve dans la littérature plusieurs articles sur la comparaison des différentes méthodes [DAE95, BR96, Jol96]. Les critères de comparaison sont en général le temps de calcul, la sensibilité au choix des seuils, le taux d'erreurs, le domaine utilisé (comprimé ou non comprimé), la détection des effets de transition, la robustesse aux mouvements de la caméra et aux changement de luminosité dans un plan. Ils concluent presque tous que l'utilisation des histogrammes même simples donnent toujours de bons résultats. Toutes les méthodes perdent de fiabilité quand il y a mouvement de caméras ou d'objets ou un changement brusque de luminosité. La normalisation par égalisation de l'histogramme peut diminuer ces taux d'échec. Les méthodes globales telles que les histogrammes sont insuffisantes. Il faut d'autres critères d'analyse du contenu en plus d'informations toujours utiles comme les mouvements de caméra.

2. Sélection de l'image représentative

L'image clé est aussi appelée l'image représentative du plan. Elle permet de caractériser un plan au niveau de la recherche et l'indexation. La méthode la plus simple pour choisir une image clé consiste à prendre une image prédéterminée: l'image du début, du milieu ou de la fin du plan. Cherfaoui et Bertin [CB94] proposent de sélectionner l'image en fonction des mouvements de caméra dans le plan. Ils choisissent une image pour un plan fixe et trois images (la première, la dernière et l'image du milieu) pour les autres plans qui contiennent des mouvements de caméra tels que le zoom. La sélection se fait manuellement .

Wolf [Wol95] propose une méthode qui repose sur l'échantillonnage non-linéaire par recherche d'un minimum local de mouvement.

Madrane et Goldberg [MG95] proposent d'utiliser une image de trace, pour représenter un plan, obtenue par combinaison linéaire des images. L'image doit représenter les mouvements dans un plan par la somme des mouvements en chaque point à travers l'ensemble des images d'un plan.

$$\mathrm{Image}_T = \sum_{t=0}^{N} \left[q^t + \frac{1 - \sum_{i=0}^{N} q^t}{N+1} \right] I_t$$: l'image de trace.

La valeur de q (q <1) est un paramètre de contrôle de distribution des poids sur l'ordre des images.

Yeung et Liu [YL95] déterminent une image représentative lorsque la mesure de dissemblance de l'image courante avec l'image représentative précédente est plus grande qu'un

seuil. La première image est considérée comme image représentative. La mesure de dissemblance est calculée par l'intersection normalisée d'histogrammes d^h qui est définie par:

$$d_{ij}^{h} = 1 - \frac{\sum_{k} \min(h_i(k) - h_j(k))}{\sum_{k} h_j(k)}$$ où h_i est l'histogramme de l'image i.

Elle peut être également calculée à partir de la projection des luminances d^p qui est définie par la formule suivante:

$$d_{ij}^{p} = \frac{1}{L \times (J+K)} \left(\frac{1}{J} \sum_{n}^{K} \left| l^r(i)_n - l^r(j)_n \right| + \frac{1}{J} \sum_{n}^{J} \left| l^c(i)_m - l^c(j)_m \right| \right)$$ où L est le nombre de niveaux de gris, J est le nombre de colonnes, K le nombre de rangs et $l^r(i)_n$ et $l^c(i)_n$ sont les projections des luminances pour le rang n et la colonne c de l'image i.

La sélection d'images représentatives dépend des changements qui surviennent dans le contenu de la scène.

3. Analyse du mouvement

H.J. Zhang et al [Zha+97] proposent d'explorer des caractéristiques statistiques du mouvement aussi bien que la trajectoire de l'objet. Les caractéristiques sont dérivées du flot optique calculé entre les images successives d'un plan. Elles comprennent la distribution directionnelle des vecteurs de mouvement qui est définie par:

$$d_i = \frac{N_i}{N_{mt}} \quad i = 1,\ldots M$$

où i représente une des M directions, N_i le nombre de points en mouvement dans la direction i et N_{mt} le nombre total des points qui ont bougé dans toutes les directions. N_{mt} peut être remplacé par le nombre total de points dans le flot optique de manière à estimer la superficie du mouvement dans la scène.

Le flot optique est un vecteur de mouvement translationnel de chaque point d'une image à un instant t. Le calcul de flot optique est effectué soit sur l'unité de bloc par transformée de Fourier ou appariement de bloc ou sur un pixel par des méthodes de gradient.

Ils estiment la vitesse moyenne et la déviation standard dans une direction donnée par:

$$\bar{s}_i = \frac{\sum_{j=1}^{N_i} s_{ij}}{N_i}, \sigma_i = \sqrt{\frac{\sum_{j=1}^{N_i} (s_{ij} - \bar{s}_i)}{N_i - 1}}, i = 1,\ldots, M$$

où s_{ij} est la vitesse du $j^{ème}$ point en mouvement dans la direction i.

Ces deux caractéristiques fournissent une description générale de la distribution de la vitesse.

Une autre méthode d'analyse du mouvement est l'utilisation d'appariement de blocs. Il indique le déplacement du bloc et donc le vecteur de mouvement local. Une image est décomposée en blocs de N x N pixels. Pour chaque bloc, on cherche le bloc qui présente le meilleur appariement dans une fenêtre donnée. En général, on choisit une fenêtre carré de côté N + 2w sur l'image précédente où w est associé à la taille de la fenêtre de recherche. La qualité de l'appariement est évaluée soit par un critère de corrélation soit par des formules du moindre carré ou de l'erreur moyenne des valeurs absolues [Kim97]:

$$\frac{1}{N^2}\sum_{m}^{N}\sum_{n}^{N}(I_t(m,n)-I_{t+1}(m+i,n+j))^2$$

$$\frac{1}{N^2}\sum_{m}^{N}\sum_{n}^{N}\left|I_t(m,n)-I_{t+1}(m+i,n+j)\right|$$

S. Benayoun et al. [Ben+98] utilisent l'approche développé par Bouthémy et al. [BG96] pour extraire les objets d'une séquence. Ils étiquettent les pixels de l'image en deux catégories selon qu'ils sont conformes ou non au mouvement dominant. Ils font ensuite l'hypothèse qu'une partie connexe et non conforme correspondra à un objet physique. Ce qui n'est toujours pas valide surtout lorsque les projections de deux objets sont connexes dans l'image ou lorsqu'un objet est immobile. Ensuite, l'objet est automatiquement extrait sur l'ensemble du plan par application du mouvement dominant.

4. La segmentation de scènes

H.J. Zhang et al. [Zha+97] définissent la similarité entre deux plans P_i et P_j composés respectivement de deux ensembles d'images $K_i = \{f_{i,m}, m=1,\ldots,M\}$ et $K_j = \{f_{j,n}, n=1,\ldots,N\}$ par:

$S_k(P_i,P_j) = \max [d(f_{i,l},f_{j,1}), d(f_{i,l},f_{j,2}) ,\ldots, d(f_{i,l},f_{j,N}),\ldots, d(f_{i,m},f_{j,1}), d(f_{i,m},f_{j,2}) ,\ldots, d(f_{i,M},f_{j,N})]$ où d est distance de similarité entre deux images.

Haddad et al. [HBB96] tirent une analogie entre les notions de mots-clés d'un texte et les images représentatives des plans et proposent un modèle vectoriel de recherche d'information pour la vidéo

Chen et al.[CFH98] utilisent la sémantique associée à un ensemble de relations temporelles proposé par Allen [All83] pour extraire des scènes. Il y a principalement trois types de relations qui sont les suivantes[Ham97]:

- "meets" : elle sépare deux clusters (un cluster est défini comme un ensemble de plans similaires et situés dans une période de temps limité) appartenant à deux scènes consécutives différentes
- "before": elle sépare deux clusters appartenant à deux scènes dans deux séquences consécutives.
- "during/overlaps": elle décrit un changement dans la distribution temporelle de plans de la même scène.

D. Détection de plan

L'objectif ici est d'identifier les changements de plan, les cuts, afin de détecter les unités de base que sont les plans. Comme nous l'avons déjà vue, la technique basée sur les histogrammes de couleurs globaux conduit à des omissions lorsque deux plans ont à peu près la même distribution de couleurs, ce qui se produit finalement assez souvent dans les émissions de TV. Nous avons proposé une méthode robuste qui utilise les histogrammes de couleur locaux [CC99d]. Pour palier à ce problème, nous avons expérimenté une technique qui consiste à localiser les histogrammes de couleurs des images. En plus, en localisant les informations, ici les histogrammes de couleurs, notre technique est moins sensible au bruit qui apparaît souvent dans les émissions TV.

1. Principe

Pour cela, chaque image est divisée en un ensemble de partitions horizontales (h) et verticales (v). Comme on travaille en général sur des images de résolutions 352 x 288 pixels, ce qui est le cas du corpus de l'INA, nous avons choisi h =6 et v= 4, ce qui nous donne 24 blocs de l'image. Au lieu de comparer deux images, on compare chaque paire de blocs correspondants. En fait, on compare les histogrammes correspondant dans l'espace L*u*v*, pour détecter des cuts locaux . Un cut global est détecté si le nombre des cuts locaux est supérieur à un seuil (deux tiers = 16). La figure 46 illustre ce processus.

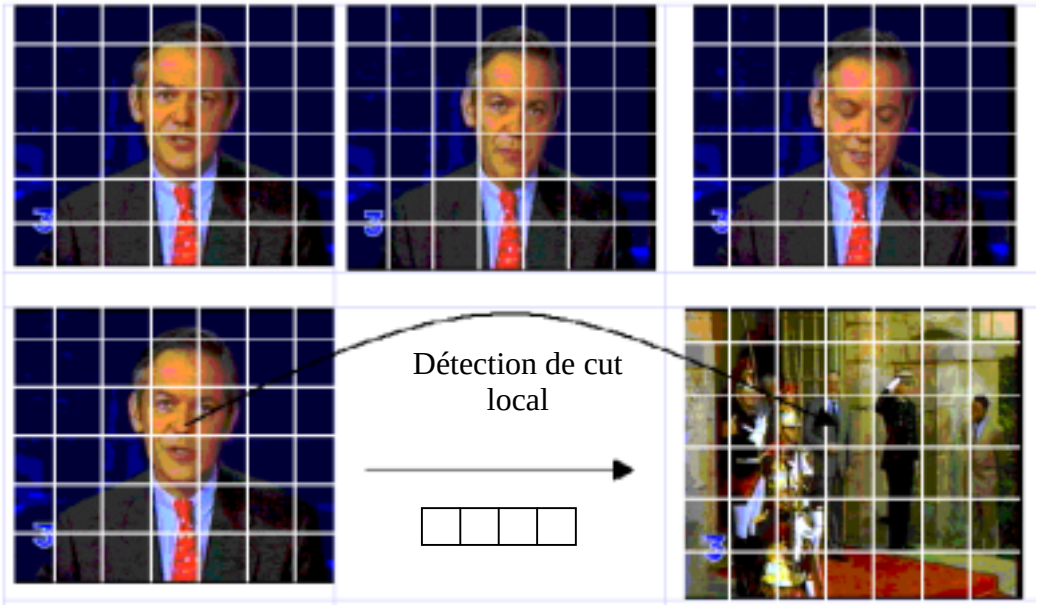


Figure 46 - Détection de cuts locaux

Séparément, nous calculons les caractéristiques visuelles de l'image et utilisons la somme des distances avec des poids comme mesure totale de l'image.

Indépendamment de l'espace de couleur, une couleur dans une image peut être représentée par un histogramme à trois dimensions simple ou trois histogrammes 1-D séparés. Soient $H_i^{L^*}$, $H_i^{u^*}$ et $H_i^{v^*}$ respectivement $H_j^{L^*}$, $H_j^{u^*}$ et $H_j^{v^*}$ les histogrammes normalisés de couleur de l'image courante f_i et de l'image suivante f_j. La similarité entre ces deux images consécutives est calculée par la formule suivante :

$$\Delta_H = d(H_{f_i}, H_{f_j}) = \frac{\sum_{L^*}\min(H_{f_i}^{L^*}, H_{f_j}^{L^*}) + \sum_{u^*}\min(H_{f_i}^{u^*}, H_{f_j}^{u^*}) + \sum_{v^*}\min(H_{f_i}^{v^*}, H_{f_j}^{v^*})}{\left|H_{f_i}\right| * 3}$$

Notons que la valeur de Δ_H se trouve dans l'intervalle [0,1]. Si les images sont identiques alors $\Delta_H = 1$. Dans cette distance, la mesure d'intersection est incrémentée par le nombre de pixels qui sont communs aux deux images successives.

Si nous représentons la distribution de la couleur par ses moments centrés (premiers quatre moments), on peut caractériser en même temps les attributs de texture et de la forme. La comparaison entre deux images successives f_i et f_j est définie par la formule suivante:.

$$d(f_i, f_j) = \frac{w_0\Delta_H + w_1\Delta_{\mu 1} + w_2\Delta_{\mu 2} + w_3\Delta_{\mu 3} + w_4\Delta_{\mu 4}}{\sum_{i=0}^{4} w_i}$$

où les w_i sont les poids associés à chaque similarité visuelle. Des expériences ont montré que cette mesure est plus robuste en traitement des images de couleur que les histogrammes de couleur [SO95], et donc utilisée comme l'une des mesures de similarité entres images d'un plan. Un cut est déclaré si cette distance est inférieure à un seuil.

2. Résultats expérimentaux

Nous avons expérimenté notre méthode sur le corpus de l'INA sur quatre séquences d'émissions TV. L'environnement du test est un Pentium-Pro 300 MHz sous Windows. Toutes les images ont une résolution de 288 * 352 avec chaque pixel est codé en 24 bits. Les résultats sont obtenus dans l'espace de couleur L*u*v* sur ces 56 minutes de la vidéo prise des programmes de Télévision comprenant des spots publicitaires et des journaux télévisés.

En ce qui concerne le choix de l'espace couleur, la figure 47 montre un exemple de comparaison de plans dans l'espace RGB et l'espace L*u*v qui explique notre préférence pour l'espace L*u*v. Car en utilisant la distance d'intersection, on remarque que les pics sont plus clairs en espace L*u*v* qu'en RGB, donc moins sensible au changement de luminosité qu'en espace RGB.

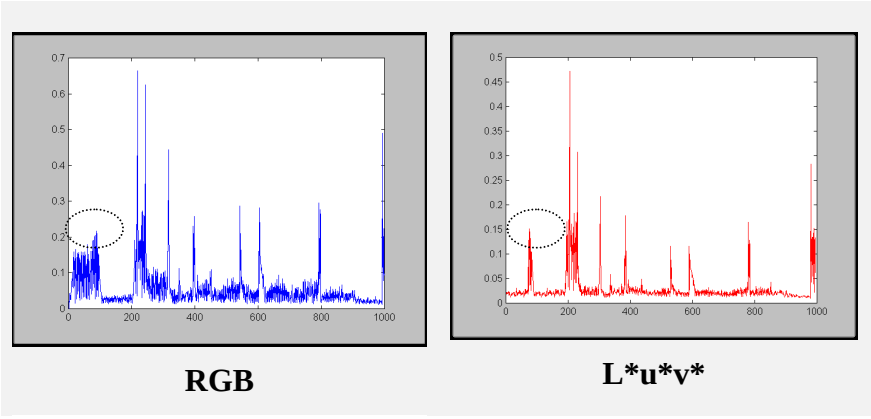


Figure 47 - Comparaison de détection de plans en utilisant la distance d'intersection dans les espaces RBG et L*u*v*

La figure 48 réalise une comparaison entre la méthode de détection de plans basée sur l'histogramme global et la nôtre utilisant des histogrammes locaux, montrant que notre méthode est de loin meilleure pour éviter les omissions de détection. En effet, comme on peut le voir sur la figure 48, alors qu'aucun cut n'est détecté pour la zone encerclée par la méthode basée sur l'histogramme global (fig. 48-a), notre méthode détecte 20 cuts locaux au niveau des blocs comme l'illustre la (fig. 48-b) et on déclare en conséquence un cut global.

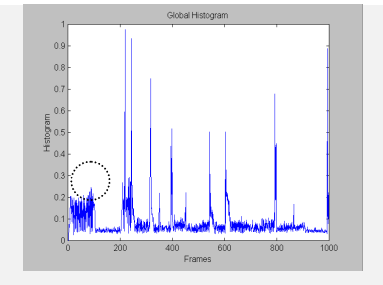


Figure 48-a: Détection de plan basée sur l'histogramme global

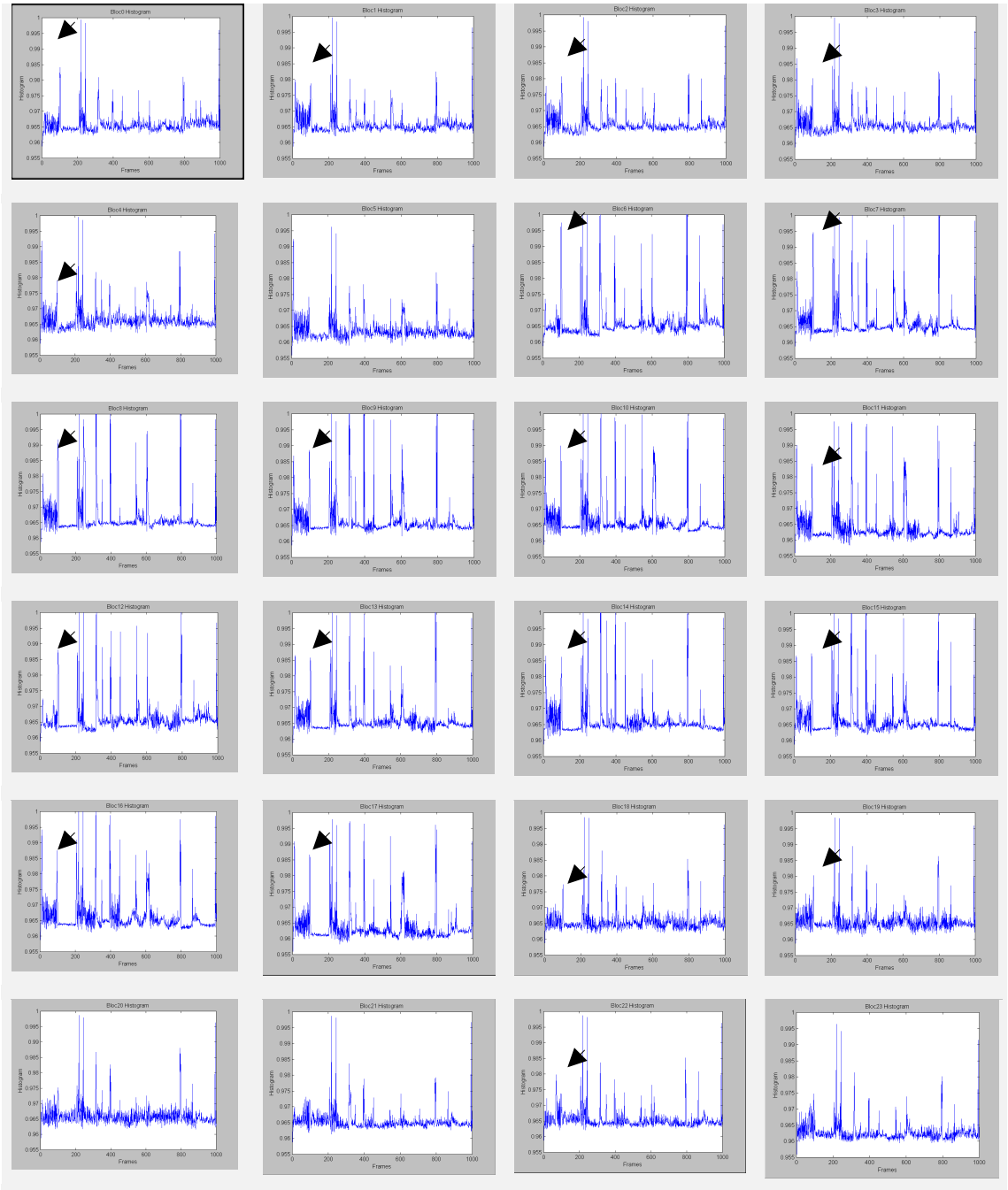


Figure 48-b - Détection de cuts locaux en utilisant la distance d'intersection

Les résultats de segmentation par notre méthode sur les quatre vidéos du corpus de l'INA sont reportés dans le tableau 14. n_c, n_m, et n_f, représentent respectivement les nombres de plans correctement détectés, non détectés et de faussement détectés. Les résultats peuvent être évalués en terme de précision et de rappel qui sont: précision = n_c/n_c+n_f et rappel =n_c/n_c+n_m. La figure 49 montre les premières images des premiers plans détectés.

Vidéo	**Total des images**	**Temps total**	**Nb total de plans**	**n_c**	**n_f**	**n_m**	**Précision**	**Rappel**
AIM1MB02	22082	14:43	369	335	13	21	0.96	0.94
AIM1MB03	18572	12:22	118	108	7	3	0.939	0.972
AIM1MB04	19928	13:17	188	170	9	9	0.949	0.949
AIM1MB05	23144	15:25	166	150	11	5	0.931	0.967

Tableau 14 - Evaluation de la méthode de détection de plans de 4 vidéos du corpus de l'INA

On remarque que les taux de rappel et de précision sont au dessus de 0.93 dans les deux cas, ce qui est un bon résultat sachant que ces vidéos sont très hétérogènes et variées.

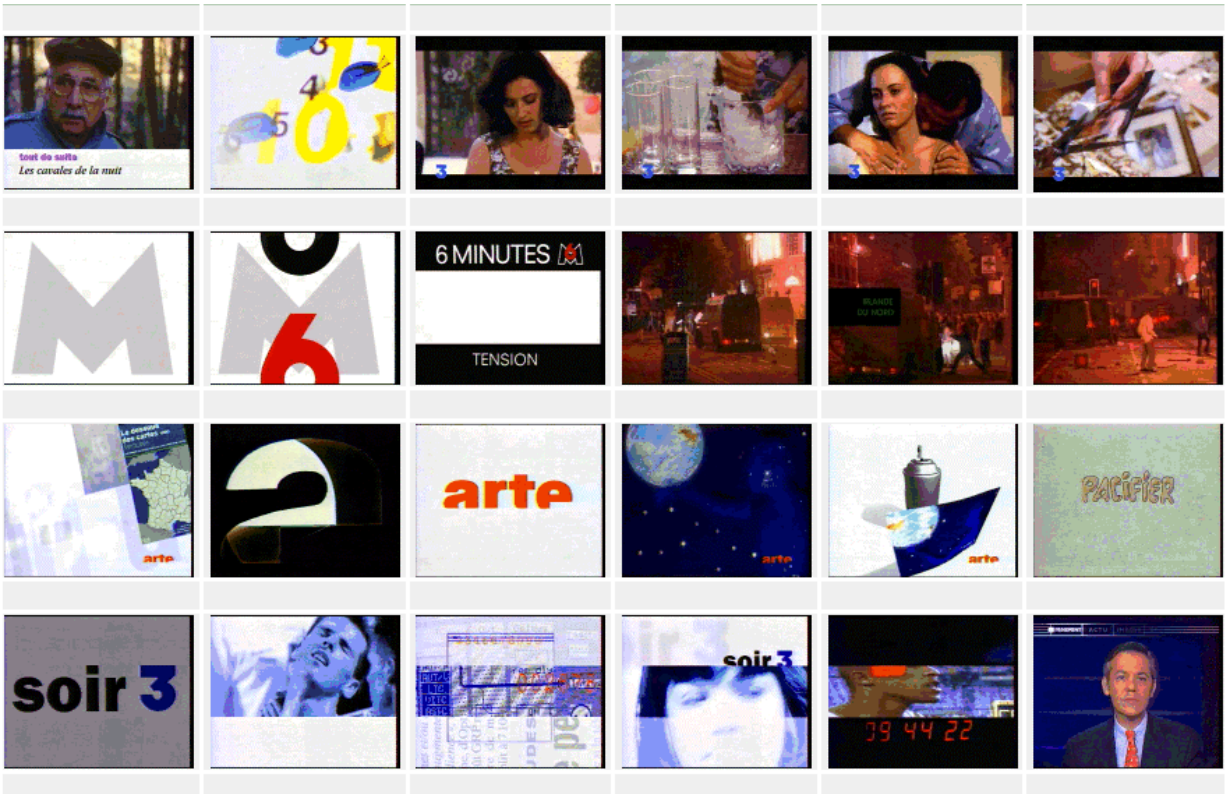


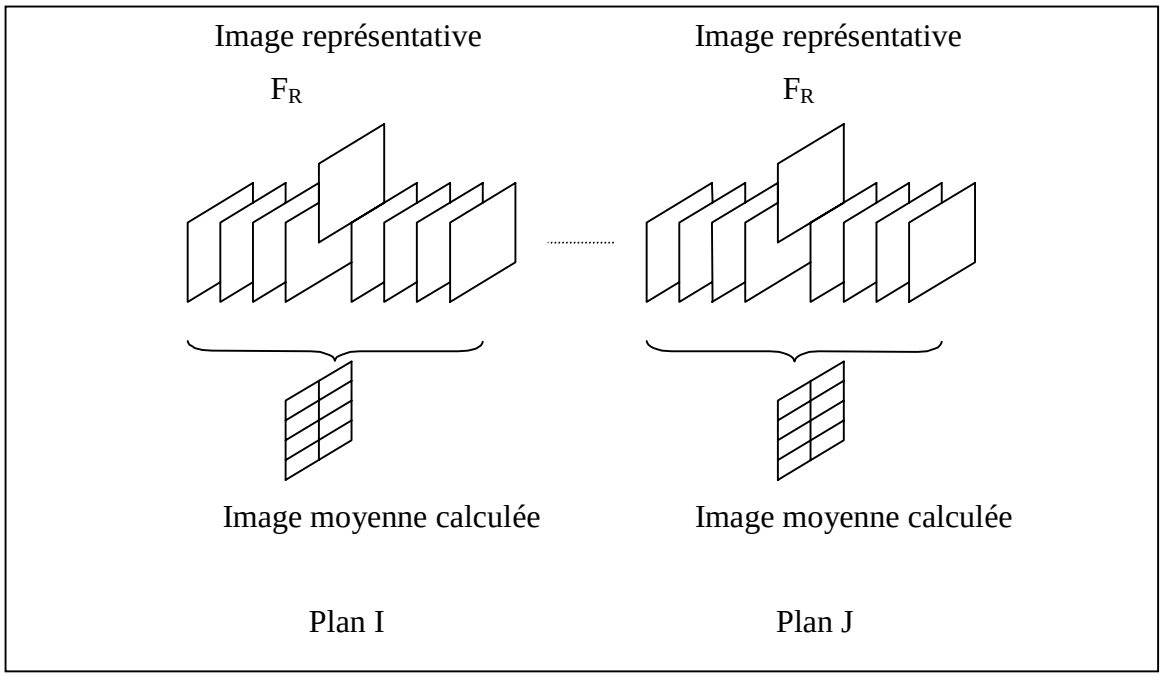
Figure 49 - Exemples de plan détectés de quatre vidéo du corpus de l'INA qui sont dans l'ordre (par ligne) : AIM1MB02, AIM1MB03, AIM1MB04, et AIM1MB05

E. Extraction des images clés

Une fois les plans segmentés, on procède en général à l'extraction d'une image clé, appelée aussi image représentative, pour chaque plan. Un plan étant une succession d'images sans interruption temporelle, il s'agit de choisir celle qui résume le contenu du plan aussi précis et complet que possible. Ce critère de *synthèse* exige donc que cette image représentative doit donc refléter non seulement les actions principales mais également le décor ou l'environnement donnant une idée sur le lieu et le temps. Force est de constater que les approches généralement suivie, qui consiste à choisir statiquement comme image clé la première image ou celle du milieu ou encore la dernière image d'un plan, sont insatisfaisantes au regard de ce critère de synthèse, car elle conduit très souvent à l'omission soit du décor, soit du temps soit de l'action principale.

1. Principe

Dans notre approche, nous considérons d'abord une image qui résulte de la moyenne des histogrammes des images d'un plan, celle-ci constitue une mesure permettant de résumer à la fois les actions principales et l'environnement. Cependant nous ne pouvons pas simplement renvoyer cette image synthétique comme image clé du plan car elle pourrait être complètement incompréhensible bien qu'elle résume la perception visuelle sur le plan des histogrammes de couleurs. Aussi, procédons à une comparaison des images du plan avec cette image synthétique, celle qui est la plus proche de l'image synthétique est choisie comme image représentative, comme l'illustre la figure suivante.



La composition de l'image synthétique est réalisée de la manière suivante. Le calcul se fait pour chaque composante L*, u*, et v* de l'histogramme sur le nombre total des images du plan. Les trois moyennes sont définies par:

$$H_M^{L*}(i) = \frac{1}{k}\sum_f^k H_f^{L*}(i) \;\; ; \;\; H_M^{u*}(i) = \frac{1}{k}\sum_f^k H_f^{u*}(i) \;\; ; \;\; H_M^{v*}(i) = \frac{1}{k}\sum_f^k H_f^{v*}(i)$$

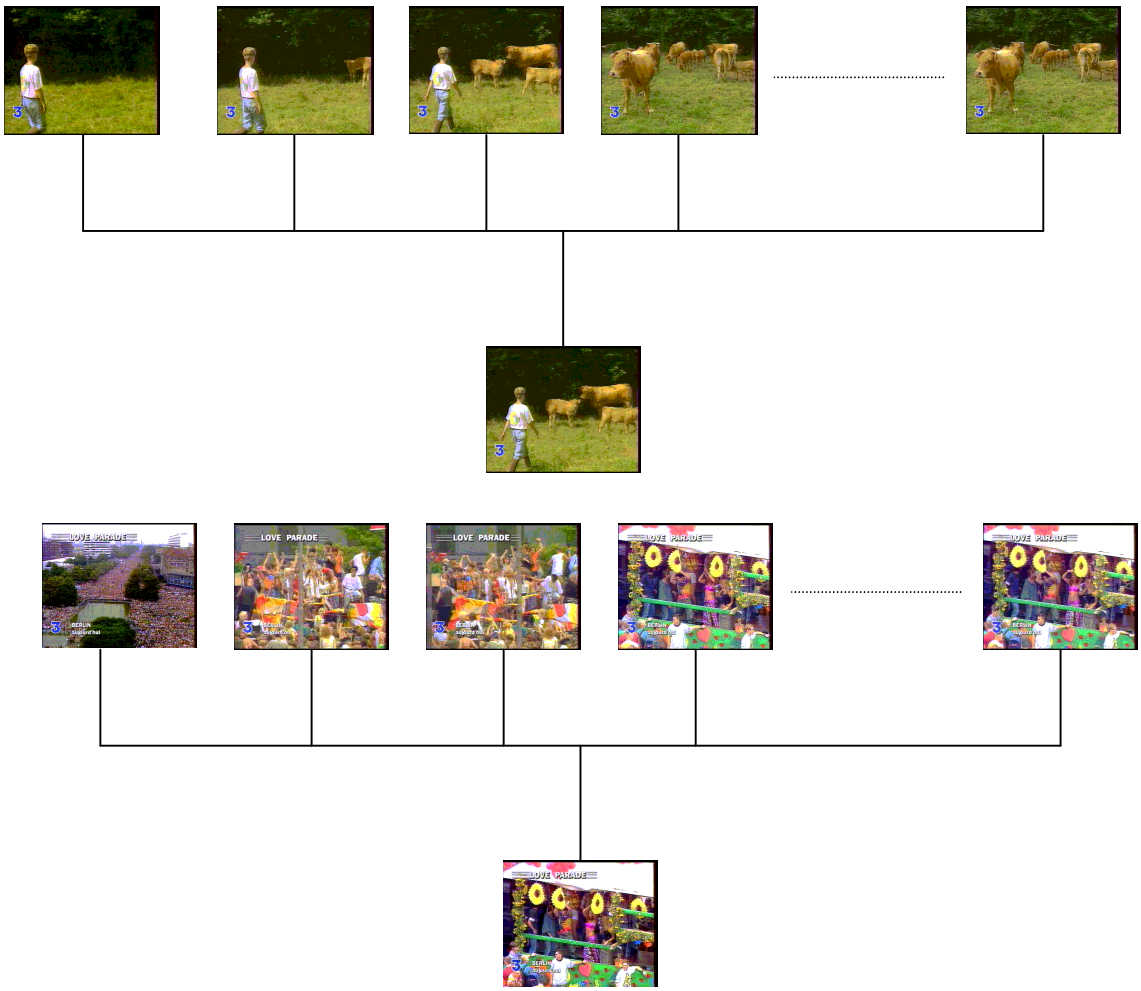
avec i un indice de l'histogramme., et k le nombre des images du plan.

Une fois l'image synthétique déterminée, nous identifions l'image clé par l'examen de chaque image f_i du plan comparée avec l'image synthétique M par la distance χ^2 qui est définie par:

$$\chi^2 = d(H_{f_i}, H_M) = \sum_{L^*} \frac{(H_{f_i}^{L^*} - H_M^{L^*})^2}{(H_{f_i}^{L^*} + H_M^{L^*})^2} + \sum_{u^*} \frac{(H_{f_i}^{u^*} - H_M^{u^*})^2}{(H_{f_i}^{u^*} + H_M^{u^*})^2} + \sum_{v^*} \frac{(H_{f_i}^{v^*} - H_M^{v^*})^2}{(H_{f_i}^{v^*} + H_M^{v^*})^2}$$

2. Résultats expérimentaux

Dans les image suivantes, on montre la première image du plan qui est présentée à gauche suivie d'autres images du même plan. L'image représentative que nous extrayions est présentée comme l'image résumée du plan. On voit bien, la différence entre ces images représentatives et celles des débuts de plans. En plus, si le début du plan est faussement détecté, l'image de début n'aurait pas de rapport avec le contenu du plan alors que l'image synthétique générerait une image représentative qui va donner un aperçu du contenu du plan même si les bornes du plan sont faux.



F. Classification des plans

Nous avons donc introduit une nouvelle méthode pour la segmentation de vidéos en plans qui sont représentés à leur tour par une image représentative sélectionnée par la technique de l'image synthétique présentée au paragraphe précédent. Or comme nous l'avons vu, une représentation d'une vidéo par ses plans est encore largement insuffisant pour le but de l'indexation de la vidéo et la navigation dans celle-ci. En effet, rappelons que cette représentation d'une vidéo par un ensemble d'images représentatives fixes ne décrit pas son contenu narratif. En plus, le nombre de plans dans un document audiovisuel est en général très élevé. Ainsi, dans « Octobre » de S.M. Eisenstein, il est de 3225, et 605 dans « Le Grand Sommeil » de Howard Hawks. D'une façon générale, un long métrage peut comporter en moyenne de 600 à 2000 plans. Afin de mieux structurer la vidéo et faciliter plus la navigation et la recherche, un découpage de la vidéo en des unités sémantiques plus macroscopiques (scènes, séquences, etc.) est souhaitable, l'objectif étant de produire éventuellement d'une manière automatique une vue synoptique de la vidéo.

1. Principe

Nous avons abordé ce problème dans nos travaux et proposons une technique permettant de regrouper les plans de contenus similaires en des clusters. Selon les travaux de Chen et al. [Che98], un *cluster* est un ensemble de plans, similaires et situés dans une période de temps limitée. La similarité implique en général un décor semblable ayant les mêmes objets. La constitution de clusters permet notamment de détecter les plans alternatifs, les plans de coupe très souvent utilisés dans la publicité, les scènes formés de plans continus. Dans notre approche, la similarité du contenu entre deux plans est simplement déterminée par une distance entre leurs images représentatives. Dans un but de comparaison, nous avons aussi implémenté une distance entre deux plans P_i et P_j proposée par H.J. Zhang et al. [Zha+97] qui est basée sur une comparaison d'images à images de deux plans. Soient plans P_i et P_j composés respectivement de deux ensembles d'images $K_i = \{f_{i,m}, m=1,\ldots,M\}$ et $K_j = \{f_{j,n}, n=1,\ldots,N\}$, cette distance est définie par:

$$D(P_i, P_j) = \frac{1}{M} \sum_{m=1}^{M} \underset{k=1}{\overset{N}{\text{Max}}} [d(f_{i,m}, f_{j,k})] \qquad \textbf{(Eq. 1)}$$

où d est la distance de similarité entre deux images définie précédemment. Remarquons déjà que le calcul de cette distance nécessite deux boucles imbriquées, parcourant les images de P_i puis celles de P_j, ce qui est assez coûteux.

Dans notre implémentation, nous représentons chaque plan par trois images: la première, la dernière et l'image représentative. La comparaison du contenu des plans se fait par comparaison de leurs images clés. Cela nous permet de déterminer pour chaque plan, les plans qui lui sont proches. A partir de là, on examine toutes les paires d'images composées par l'image de fin du plan courant et l'image du début d'un autre plan postérieur. Si les deux images sont suffisamment proches d'après un seuil, nous agrégeons ces deux plans en un cluster. L'algorithme que nous proposons utilise la distance de χ^2 et il est détaillé ci-dessous :

```
Entrée :
        Debut  : tableau de début des plans
        Fin: tableau de fin des plans
        Plan_Proche :  une matrice de vecteurs des plans proches basé sur les similarités entre les plans sur la base de leurs images clés.
        k = Plan_Proche (i,j) signifie que k est le j^ème plan proche au plan i
Etape 1:
        Pour i=0  à n Faire Plan_Proche (i,j) =  non visité;
          k=0;
Etape 2:
        Pour i=0 à n Faire
                J = -1;
                plan i est visité;
                Répéter
                        j = j + 1;
                Jusqu'à (  ( D (Fin(i) , Debut(Plan_Proche (i,j)) ) > seuil )
                        ET  (j<n)
                        ET (Plan_Proche (i,j) est non visité)) ;
                Si ( j<n) Alors
                        scene(k) = scene(k) U {plan j} ;
                        Plan_Proche (i,j) est visité ;
                        k  = k +1;
                Fin
        Fin
Sortie :    Constitution de k clusters
```

2. Résultats expérimentaux

Les expériences ont été menées sur le corpus de l'INA *"AIM1MB02"* qui dure 14:43 minutes qui totalise 22082 images réparties sur 369 plans. Il s'agit des publicités TV sur France Télévision (FR2, FR3) et M6. Il s'agit d'une façon générale des matières très dures pour la segmentation de scènes car les plans sont très courts et changent très rapidement, en plus ils ne respectent pas non plus le critère classique de continuité temporelle qui est respecté par tous les films de fiction pour la compréhension de la narration par le public.

Nous avons comparé notre méthode basée uniquement sur la similarité d'images représentatives avec celle proposée par Zhang et al. qui détermine la similarité de deux plans à partir de l'Eq. 1, comparant donc l'ensemble des images des deux plans. En principe, cette dernière traduit mieux la similarité des deux plans car toutes les images composant les plans sont consultées. Dans nos expérimentations, nous allons voir que les deux méthodes conduisent à des résultats pratiquement identiques alors que notre méthode, ne comparant que les images représentatives, est beaucoup moins coûteuse.

Les tableaux 15 et 16 donnent les mesures de similarité entre plans calculées respectivement par la méthode de Zhang et la nôtre. On en déduit dans le tableau 17 pour chaque plan ceux qui lui sont proches dans un ordre de similarité décroissante. On voit que les résultats sont pratiquement identique puisque seul le plan 4 produit un résultat légèrement différent dans la mesure où notre méthode place le plan 5 en troisième position alors que la méthode de Zhang le place en avant dernière position. De cette expérimentation, on est donc tenté de dire que la comparaison du contenu de deux plans peut être ramené à celle de leurs images représentatives respectives.

Le tableau 18 montre les résultats de la segmentation de clusters par notre algorithme. La figure 50 illustre une partie du résultats à partir des 91 premiers plans de AIM1MB02. On y voit que le premier cluster est formée d'un ensemble de plans coupe qui se produisent fréquemment dans les publicités TV, tandis que le deuxième cluster est formé uniquement des plans d'annonce de publicité. Par rapport à la définition de cluster telle que nous avons introduits au début de la section, les résultats sont tout à fait cohérents.

AIM02	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**
S1	-	0.0769	0.0519	0.0291	0.0334	0.0603	0.0955	0.0551	0.0589	0.0598
S2	0.0769	-	0.0613	0.0332	0.0375	0.0744	0.0673	0.0625	0.0629	0.0672
S3	0.0519	0.0613	-	0.0177	0.0186	0.0348	0.0316	0.0299	0.0301	0.0318
S4	0.0291	0.0332	0.0177	-	0.0190	0.0321	0.0291	0.0272	0.0277	0.0288
S5	0.0334	0.0375	0.0186	0.0190	-	0.0305	0.0264	0.0244	0.0248	0.0271
S6	0.0603	0.0744	0.0348	0.0321	0.0305	-	0.0216	0.199	0.0199	0.214
S7	0.0955	0.0673	0.0316	0.0291	0.0264	0.0216	-	0.185	0.0178	0.192
S8	0.0551	0.0625	0.0299	0.0272	0.0244	0.199	0.185	-	0.0163	0.166
S9	0.0589	0.0629	0.0301	0.0277	0.0248	0.0199	0.0178	0.0163	-	0.153
S10	0.0598	0.0672	0.0318	0.0288	0.0271	0.214	0.192	0.166	0.153	-

Tableau 15 - Mesures de similarité des plans utilisant toutes les images

AIM02	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**
S1	-	0.564	0.300	0.228	0.298	0.432	0.785	0.312	0.384	0.401
S2	0.564	-	0.439	0.399	0.409	0.503	0.498	0.468	0.479	0.482
S3	0.300	0,439	-	0.153	0.159	0.390	0.228	0.261	0.300	0.238
S4	0.228	0.399	0.153	-	0.259	0.381	0.219	0.795	0.295	0.208
S5	0.298	0.409	0.159	0.259	-	0.291	0.278	0.190	0.267	0.207
S6	0.432	0.503	0.390	0.381	0.291	-	0.287	0.209	0.222	0.801
S7	0.785	0.498	0.228	0,219	0.278	0.287	-	0.781	0.218	0.789
S8	0.312	0.468	0.261	0.795	0.190	0.209	0.781	-	0.178	0.645
S9	0.384	0.479	0.300	0.295	0.267	0.222	0.218	0.187	-	0.543
S10	0.401	0.482	0.238	0.208	0.207	0.801	0.789	0.645	0.543	-

Tableau 16 - Mesures de similarité des plans à partir de leurs images clés

Plans	**Plans proches par ordre de similarité**	
	A partir de l'ensemble des images	**A partir des images représentatives**
S1:	7,2,6,10,9,8,3,5,4	7,2,6,10,9,8,3,5,4
S2:	1,6,7,10,9,8,3,5,4	1,6,7,10,9,8,3,5,4
S3:	2,1,6,10,7,9,8,5,4	2,1,6,10,7,9,8,5,4
S4:	*2,6,1,7,10,9,8,5,3*	*2,6,5,1,7,10,9,8,3*
S5:	2,1,6,10,7,9,8,4,3	2,1,6,10,7,9,8,4,3
S6:	10,8,2,1,3,4,5,7,9	10,8,2,1,3,4,5,7,9
S7:	10,8,1,2,3,4,5,6,9	10,8,1,2,3,4,5,6,9
S8:	6,7,10,2,1,3,4,5,9	6,7,10,2,1,3,4,5,9
S9:	10,2,1,3,4,5,6,7,8	10,2,1,3,4,5,6,7,8
S10	6,7,8,9,2,1,3,4,5	6,7,8,9,2,1,3,4,5

Tableau 17 - Comparaison des résultats des plans proches obtenus par les deux similarités précédentes

Vidéo	Nombre de Scènes	Classification correcte	Classification incorrecte	Non classée	Succès (%)
AIM1MB02	32	29	1	2	91%

Tableau 18 - Evaluation des scènes de AIM1MB02

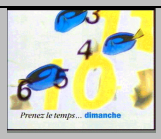
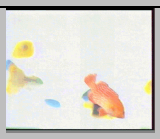
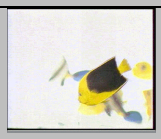
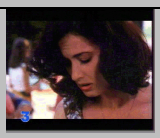
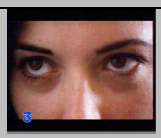
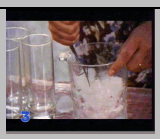
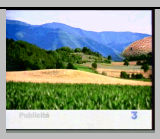
Cluster 1	 Plan 2	 Plan 27	 Plan 31	 Plan 33	 Plan 62	 Plan 79
Cluster 2	 Plan 3	 Plan 30	 Plan 64	 Plan 89	 Plan 91	….
Cluster 3	 Plan 4	 Plan 16	 Plan 19	 Plan 24		
Cluster 4	 Plan 5	 Plan 15	 Plan 21	 Plan 23	 Plan 25	
Cluster 5	 Plan 35	 Plan 37	 Plan 42	 Plan 45		
Cluster 6	 Plan 38	 Plan 40				
Cluster 7	 Plan 32	 Plan 61	….	….	….	
Cluster 8	 Plan 34	 Plan 46	 Plan 60	…	….	

Figure 50 - Classification des plans

VII. Conclusion

Les systèmes de recherche d'informations visuelles visent la maîtrise d'une nouvelle frontière dans l'univers d'informations numériques, celles d'images fixes et animées. Il s'agit d'un défi majestueux et prometteur car il nécessite des efforts interdisciplinaires, en bases de données, en analyse et traitement d'images et en recherche d'informations documentaires, etc. pour la réalisation d'une telle édifice. Le présent manuscrit de thèse apporte notre modeste pierre à cette construction. De nombreux problèmes restent encore ouverts et vont constituer l'objet de nos travaux à venir.

D'abord en ce qui concerne les images fixes, si par nos techniques de segmentation d'objets visuellement homogènes qui sont ensuite caractérisés par leurs signatures visuelles et spatiales, nous avons transformé un problème de recherche d'images par le contenu visuel à celui d'une recherche dans des données qui sont finalement alphanumériques, nous n'avons pas encore le temps d'explorer complètement les bénéfices d'une telle transformation. D'abord, en considérant une image comme étant une composition d'objets visuellement homogènes et en faisant l'équivalence de résumés visuels avec les termes tels que nous trouvons dans des documents textuels, nous aurions pu appliqué un modèle statistique pour affiner nos mécanismes de recherche dans la richesse d'expression de requête et la précision quant à la précision dans les réponses. En effet, si l'on décrit le visage humain approximativement comme étant composé de deux régions homogènes représentant les yeux, une région homogène représentant la bouche et une autre région représentant les cheveux, l'ensemble de ces objets visuellement homogènes étant régis par des relations spatiales identifiés, il est alors possible de formuler des requêtes sur le concept qu'est le visage humain en s'appuyant des recherches utilisant des modèles vectoriels. Ensuite, pour l'instant nous n'avons pas non plus le temps d'aborder le problème du bouclage de pertinence qui est alors couramment utilisé dans les recherches documentaires. Or, dans le domaine de recherche d'images, comme les critère de recherche initiaux d'un utilisateur sont nécessairement imprécis - car la perception humaine étant toujours subjective -, on peut s'attendre beaucoup d'un tel mécanisme pour améliorer au fur et à mesure les résultats de recherche.

Ensuite au chapitre des travaux sur les images animées, il serait intéressant de réaliser encore une meilleure analyse de la vidéo en indexant les événements d'un objet visuel. En effet, en utilisant les relations spatiales entre les objets de deux images successives d'une vidéo, on peut en déduire certaines caractéristiques sur l'objet, telles que sa trajectoire, sa vitesse, son ordre dans l'image, etc., aboutissant à un graphe qui capture les mouvements de celui-ci. Les mouvements d'un objet dans un flux continu d'images peuvent être définis par huit événements que propose Courtnay [Cour97] :

- Apparition ("Appearence"): un objet qui apparaît dans une scène.
- Disparition("Disappearance"): un objet disparaît de la scène.
- Entrée("Entrance"): un objet en mouvement entre dans la scène.
- Sortie("Exit"): un objet en mouvement quitte la scène.
- Dépôt ("Deposit"): un objet immobile est ajouté à la scène.
- Enlèvement ("Removal"): un objet immobile est enlevé de la scène.

- Mouvement ("Motion"): un objet qui était au repos qui commence à bouger
- Repos ("Rest"): Un objet qui était en mouvement s'arrête.

Si on résume une image d'un plan par ses premiers objets visuellement dominants, nous pouvons tracer un graphe de mouvement, comme l'illustre la figure 51, en suivant les objets d'une image à une autre. Dans un tel graphe, les objets sont classés par ordre décroissant en fonction de leurs tailles (superficie). Par exemple, l'objet $_{0123}A_{30}$ qui était l'objet le plus important avec une taille de 30 dans l'image I_0 est devenu second dans l'image suivante I_1 avec une taille de 28, alors que l'objet D va disparaître de l'image I_1 à I_2 qui voit apparaître l'objet E, etc.

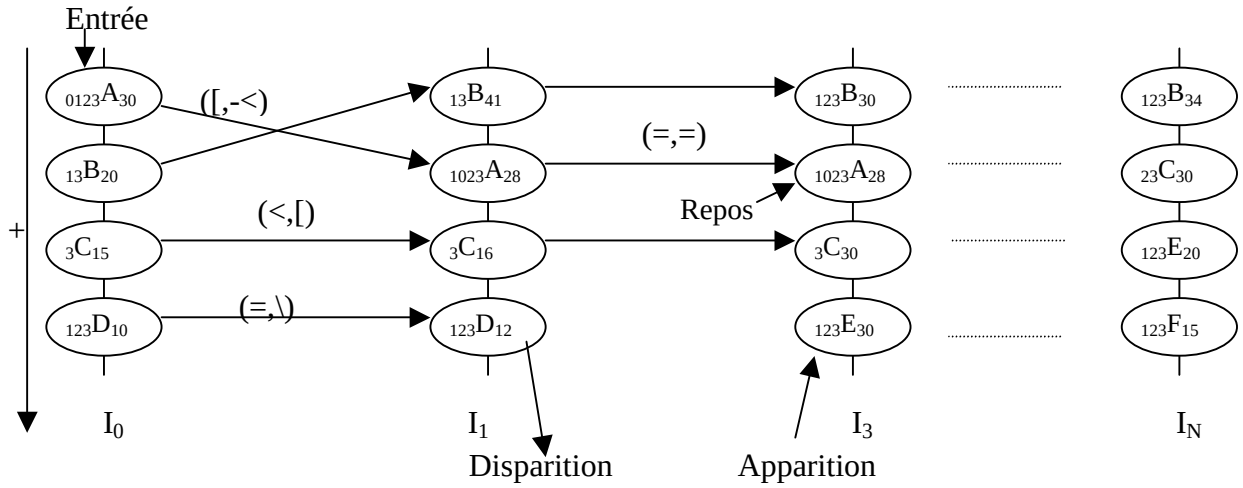


Figure 51 - Graphe de mouvement des objets dominants

Sachant qu'un objet est défini par sa taille relative et sa dispersion, en fait on peut déduire la direction du mouvement d'un objet et le type d'événement le caractérisant, en comparant les dispersions et les positions d'un objet (cordonnées du centre de gravité et de son RME) dans deux images successives. En examinant les coordonnées d'un objet entrant, on peut aussi estimer son sens d'entrée pour savoir s'il est de gauche, de droite, de haut ou de bas.

Enfin, la vitesse d'un objet X caractérisé par son centre de gravité (x,y) entre deux instants t_i et t_j est définie par:

$$V_j = \frac{\sqrt{((x_j - x_i)^2 + (y_j - y_i)^2)}}{t_j - t_i}$$

On peut ainsi déduire si un objet est stationnaire. La trajectoire d'un objet peut être obtenu en suivant son mouvement à travers les images successives. La détection d'un objet en mouvement dans l'image suivante, sauf s'il a disparu, se fait dans une fenêtre de recherche, comme l'illustre la figure 52. Le choix de cette fenêtre dépend du RME de l'objet dans l'image précédente en tenant compte d'un seuil de tolérance. A l'intérieur de cette fenêtre, on pourrait exécuter notre procédure de croissance de région à partir d'un point de départ qui est le centre de gravité de l'objet dans l'image précédente. A ce sujet nous allons mener bientôt des expérimentations pour valider nos idées proposées ici.

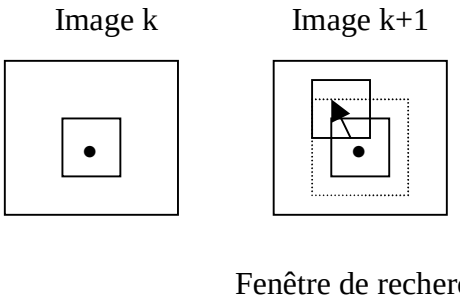


Figure 52 - Suivi d'un objet en mouvement dans deux images successives

Nous avons décrit la problématique dans le cadre de l'exploitation d'indices visuels, et réalisé la transformation qui peut nous faire profiter pleinement des résultats acquis depuis des décennies dans le domaine du texte.

Nous souhaitons également investir dans la recherche floue dans les images. En effet, si la recherche précise, basée sur la logique booléenne, s'applique fort bien aux données alphanumériques, elle s'avère inadaptée lorsqu'il s'agit de rechercher par le contenu dans une base d'images fixes ou animées, car les critères de recherche dans un tel contexte ne sont jamais précis. Il serait aussi souhaitable d'explorer les caractéristiques visuelles des images animées, comme par exemple les mouvements de caméra, les effets spéciaux comme le zoom et le panorama, et les trajectoires et les formes des objets en mouvement.

Dans un tel contexte, nous nous intéressons également à contribuer à la définition du futur standard MPEG-7 qui vise un langage commun de description de vidéos, dont les enjeux applicatifs importants montrent que l'indexation des informations visuelles présente de riches perspectives de transfert technologique.

VIII. Références

[AHC93a] F. Arman, A. Hsu, and M-Y. Chiu, Image processing on compressed data for large video databases, Proc. SPIE: Storage Retrieval Image Video Databases 1993, pp. 267-272

[AHC93b] F. Arman, A. Hsu and M.-Y Chiu, Feature management for large video databases, ACM Multimedia 93, 1908, February 1993, p. 2-12

[AJ94] P. Aigrain and P. Joly. The automatic real-time analysis of film editing and transition effects and its applications. Computer & Graphics, 18(1):93-103, 1994

[AJL95] Ph.Aigrain, Ph.Joly, V.Longueville, « Medium Knowledge-based Macro-segementation of Video into Sequences », Proc.IJCAI Workshop on Intelligent Multimedia Information Retrieval, Ed.Mark Maybury, 1995

[Aku+92] Akihito Akutsu, Yoshinobu Tonomura, Hideo Hashimoto and Yuji Ohba, Video indexing using motion vectors, Proc. of Visual Communication and Image Processing 1992, Boston, SPIE Proc. series Vol. 1818, pp 1522-1530

[Ale+95] A.D. Alexandrov, W.Y. Ma, A. El Abbadi, end B.S. Manjunath. Adaptive filtering and indexing for image databases. Storage and Retrieval for Image and Video Databases III, SPIE Vol. 2420, pp12-23

[All83] J.F. Allen. Maintainning Knowledge about temporal intervals, CACM 1983, Vol. 26, pp. 832-843

[All95] James Allan. Relevance feedback with too much data. In Proc of SIGIR'95, 1995.

[Als+96] P. Alshuth, Th. Hermes, J. Kreyb, and M. Roper. Video Retrieval with IRIS. In Proceedings of ACM Multimedia Conference, Boston MA, page 421, 1996

[AM93] S. Arya and D.M. Mount. Algorithms for fast vector quantization. In Storer, J.A. and Cohn, M. editors, Proc. of DCC'93: Data Compression Conference, pages 381-390, IEEE press

[AM98] Mohamed AKHERAZ et Martial MUZARD, Elaboration d'une interface pour la gestion et l'analyse, rapport de TX en automne 1998

[APV99] D. Androutsos, K.N. Plataniotis and A.N. Venetsanopoulos. A perceptual motivated technique to query-by-example using color cardinality. In Proc. of SPIE Multimedia Storage and Archiving Systems IV. September 1999, Vol. 3846, pp.137-145

[Ard+96] Mohsen Ardebilian, Xiao Wei Tu, Liming Chen and Pascal Faudemay, Video Segmentation Using 3-D hints contained in 2-D images, Proc. SPIE, Multimedia Storage and Archiving Systems, Vol. 2916, pp. 236-242, 1996

[Ark+91] E.M. Arkin et al. An efficiently computable metric for comparing polygonal shapes. IEEE Trans. Patt. Recog. and Mach. Intell., 13(3), March 1991

[AS91] Walid G. Aref and Hanan Samet. Optimization strategies for spatial query processing. Proc. of Very Large Data Bases, pages 81-90, September 1991

[ASF97] Vassilis Athitsos, Michael J. Swain, and Charles Frankel. Distinguishing photographs and graphics on the world wide web. In Proc IEEE Workshop on Content-Based Access of Image and Video Libraries, 1997.

[ATY+95] Aslandogan, Y.A., Thier, C., Yu, T.C., Liu, C., and Nair K. Design, implementation and evaluation of SCORE (a system for Content based Retrieval of Pictures). In proceedings of IEEE ICDE'95, pages 280-287, March1995.

[ATY+97] Aslandogan, Y.A.lp, Thier, Charles, Yu, T.Clement, Zou, Jun, and Rishe, Naphtali. Using Semantic Contents and WordNet TM in Image Retrieval. In Proceedings of ACM SIGIR Cnoference, 1997.

[Aum88] M. Marie Aumont, L'analyse de films, 2é édition, Nathan, 1988

[BA95] G. Baxter et D. Anderson. 1995. Image indexing and retrieval : some problems and proposed solutions. New Library World 96, no 1123 : 4-13.

[Bac+95] J.R. Bach, C. Fuller, A. Gupta, A. Hampapur, R. Jain, and C. Shue The Virage image search engine: An open framework for image management. In Proc. SPIE Storage and Retrieval for Image and Video Databases.1995

[Bar+77] H.G. Barrow et al.. Parametric correspondence and chamfer matching: Two new techniques for image matching. In Proc rth Int. Joint Conf. Artificial Intel. 1977

[BB82] D. H. Ballard and C. M. Brown. Computer Vision. Prentice-Hall, Inc, Englewood Cliffs, NJ, 1982.

[BBK98] Stephan Berchtold, Christian Bohm, and Hans-Peter Kriegel. The Pyramid-Technique: Towards Breaking teh Curse of Dimensionality. In proceedings of ACM SIGMOD, pages 142-153,1998.

[Bec+90] N. Beckmann, H-P. Kriegel, R. Schneider, and B. Seeger. The R*-tree: an efficient and robust access method for points and rectangles. In Proc. ACM SIGMOD, 1990

[Ben+98] Serge Benayoun et al. Structuration de vidéos pour des interfaces de consultation avancées,

[Ben75] J.L. Bentley. Multidimensional binary search trees used for associative searching. Communications of the ACM, 18(9):509-517

[Ben90] J.L. Bentley. K-d trees for semi dynamic points sets. In Proc. of the Sixth ACM Annual Symposium on Computational Geometry, pages 187-197

[Ber+96] S. Berchtold et al. The X-tree: AN index structure for high-dimensional data. VLDB'96

[Bes90] Howard Besser 1990. Visual access to visual images : the UC Berkley Image Database Project. Library Trends 38, no 4 (Spring) : 787-798.

[BG96] P. Bouthemy and F. Ganansia. Video partitioning and camera motion characterization for content-based video indexing. In Proc. of 3rd IEEE Int. Conf. on Image Processing, volume I, pages 905-909, September 1996

[BKL97] Babu M. Mehtre, M. Kankanhalli, and Wing Foon Lee. Shape measures for content based image retrieval: A comparison. Information Processing &Management, 33(3), 1997.

[BM91] C.Berrut, M.Mechkour, « Representation of images in databases : a preliminary study », Proc. of IFIP-WG2.6 2nd Conference on Visual Database Systems, Budapest, 1991

[Bor+90] Bordogna et al. 1990. Pictorial indexing for an integrated pictorial and textual environment. Journal of Information Science 16, no 3 : 165-173.

[BR96] J. S. Boreczky and Lawrence A. Rowe, Comparison of video shot boundary detection techniques, Storage and retrieval for Image and Video Databases IV, Proc. of IS&T/SPIE 1996 Int'l Stymp. On Elec. Imaging: Science and technology, 1996

[BS75] A. Bookstein and D.R. Swanson, A decision theoretic foundation for indexing, Journal of the american society for information science, 28: 45-50, 1975

[BS95] Chris Buckley and Gerard Salton. Optimization of relevance feedback weights. In Proc. of SIGIR'95, 1995.

[Car+97] Chad Carson, serge Belongie, Hayit Greenspan, and Jitendra Malik. Region-based image querying. In Proc of IEEE Workshop on Content-based Access of Image and Video Librairies, in conjunction with IEEE CVPR'97, 1997.

[CB94] Mourad Cherfaoui and Christian Bertin, Two-stage strategy for indexing and presenting video, Proc. SPIE'94 Storage and Retrieval for Video Databases, San Jose, 1994

[CB95] J.M. Corridoni and A. Del Bimbo, Film Semantic Analysis, Proc. of ACAIP, Prague, 1995

[CC97] Y. Chahir, L. Chen, "Peano key rediscovery for content based retrieval of images", Proc. of SPIE Conf. on Multimedia Storage and Archiving Systems, Vol. 3229, Dallas, USA, Nov. 1997, ISBN 0-8194-2662-8

[CC98] Y. Chahir, L. Chen, "Spatialized Visual Features-Based Image Retrieval", Proc. of the ISCA, 14 th International Conference on Computers and Their Applications, ISBN: 1-880843-27-7 , pp. 174-179, Cancun, Mexico ,April 7-9, 1998

[CC99a] Y. Chahir, L. Chen, "Spatialized Visual Features-Based Image Retrieval", article à paraître dans la revue "International Journal of Computers and Their Applications", au mois de décembre 99.

[CC99b] Y. Chahir, L. Chen, " Efficient Content-Based Image Retrieval based on color homogenous objects segmentation and their spatial relationship characterization ", IEEE Multimedia Systems'99, International Conference on Multimedia Computing and Systems ,June 7-11, 1999, Florence, ITALY

[CC99c] Y. Chahir and L. Chen , "Searching Images on the basis of color homogeneous objects and their spatial relationship", Papier accepté, à la revue "Journal of Visual Communication and Image Representation" , à apparaître fin 1999

[CC99d] Y. Chahir, L. Chen, "Automatic video segmentation and indexing", Proc.of SPIE Conf. on Intelligent Robots and Computer Vision XVIII : Algorithms,

Techniques, and Active Vision, ISBN 0-8194-3430-2, pp :345-357, 19-22 September 1999 Boston

[CCH92] James P. Callan, W. Bruce Croft, and Stephen M Harding. The inquiry retrieval system. In proc of 3rd Int Conf on Database and Expert System Application, Sept 1992.

[Cel90] M. Celenk, A color clustering technique for image segmentation, Computer Vision, Graphics, and image processing, vol. 52, pp. 145-170, 1990

[CEM98] S.-F. Chang, A. Eleftheriadis, and Robert McClin . Next-generation content representation, creation and searching for new media applications in education. IEEE Proceedings, 1998.

[CFH98] Liming Chen, Dominique Fontaine et Riad Hammoud. La segmentation sémantique de la vidéo basée sur les indices spatio-temporels. Coresa 1998, Lannion

[CH89] M. Créhange, and G. Halin, Machine Learning Techniques for Progressive Retrieval in an Image Database, in Proceedings Datenbanksysteme in Buro, Technik und Wissenschaft, T. Harder, ed. Zurich, march 1989, Springer-Verlag, pp. 314-322

[Cha+87] S.K. Chang et al.. Iconic indexing by 2D Strings. IEEE Transaction on Pattern Analysis and Machine Intelligence, p(3):413-428, 1987

[Cha+97] Moses Charikar, Chandra Chekur, Thomas Feder, and Rajeev Motwani. Incremental clustering and dynamic information retrieval. In Proc. of the 29th Annual ACM Symposium on Theory of Computing, pages 626-635, 1997.

[Cha94] D .Chandler, The Grammar of Television and Film , UWA, 1994. http://www.aber.ac.uk/~dgc/gramtv.html

[Che+95] L.Chen, P.Faudemay, D.Donsez, L.Sonké, P.Maillé, C.Ragot, « TransDoc : Centre de Documentation Multimédia Sans Mur », projet n°617 labbelisé d'expérimentation d'intérêt publique par le Minsitère de l'Industrie dans son appel à propositions « Autoroutes de l'Information », 1995

[Chen98] Liming Chen, Une contribution pour les accès intelligents aux documents multimédias dans les systèmes d'information globaux, Rapport scientifique pour l'obtention d'une Habilitation à Diriger des Recherches, UTC-Compiègne, 8 Janvier, 1998

[Chr94] V. Christophides, Recherche documentaire par structure: une approche comparative entre SRI et SGBD, dans Le Traitement électronique du document, ADBS Editions, Paris, 1994

[CJ83] G.C. Cross and A. K. Jain. Markov random field texture models. IEEE Trans. Patt. Recog. and Mach. Intell. 5:25-39, 1983

[CJL89] S.Chang, E. Jungert, and Y. Li, Representation and retrieval of symbolic pictures using generalized 2-D strings, Proc. SPIE: Visual Commun. Image Process. IV 89, 1360-1372

[CK93] Tianhorng Chang and C.-C Jay Kuo. Texture analysis and classification with tree-structured wavelet transform. IEEE Trans. Image Proc., 2(4):429-441, October 1993.

[CL95] D. Copper and Z. Lei. On representation and invariant recognition of complex objects based on patches and parts. Springer Lecture Notes in Computer Science series, 3D Object Representation for Computer Vision, pages 139-153, 199. M. Hebert, J. Ponce, T.Boult, A. Gross, editors

[CM78] W.S. Cooper and M.E. Maron, Foundation of probabilistic and utility theoretic indexing, Journal of the ACM, 25(1): 67-80, 1978

[Coc+95a] J.P. Cocquerez, et al. Analyse d'images: filtrage et segmentation, Masson, ISBN: 2-225-84923-41995 Chap II, p:14

[Coc+95b] J-P. Cocquerez et al. Analyse d'images: filtrage et segmentation. Masson, ISBN: 2-225-84923-41995, Chap. III, pp.39-63

[Cor+94] G. Cortelazzo et al. , Trademark shapes description by string-matching techniques, Pattern Recognition, vol. 27, no.8, pp. 1005-1018, 1994

[Cos+92] Gennaro Costaglio et al., Representing and Retrieving Symbolic Pictures by Spatial Relations; Visual Database Systems, II 1992 IFIP

[Cour97] Jonathan D. Courteney, Automatic video indexing via object motion analysis, Pattern Recognition, Vol. 30, No. 4, pp. 607-625, 1997, Special issue: image databases, Guest Editors, John C.M. Lee and Anil Jain

[Cox98a] I. J. Cox, M. L. Miller, Thomas P. Minka, and P. N. Yianilos. An optimized interaction strategy for bayesian relevance feedback; In IEEE Conf. CVPR, 1998.

[Cox98b] I. J. Cox, M. L. Miller, S. M. Omohundro, and P. N. Yianilos. Target testing and the pichunter bayesian multimedia retrieval system. In Advanced Digital Libraries Forum, Washington D.C., May.

[CSY87] S.K.Chang, Q.Shi, and C.Yan. Iconic indexing by 2-d string. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1987.

[CTO97] T.S Chua, K-L Tan and B-C Ooi. Fast signature-based color-spatial image retrieval. In Proc. IEEE Conf. on Multimedia Computing and Systems, 1997

[CW92] Chang, C.-C. et Wu, T.-C. 1992. Retrieving the most similar symbolic pictures from pictorial databases. Information Processing and Management 28, no 5 : 581-588.

[CWS95] S.F. Chang, J.Smith, and H.Wang. Automatic Feature Extraction and Indexing for Content-Based Visual Query. Technical Report CU/CTR 414-95-20, Columbia University, January 1995.

[DAE95] Apostolos Dailianas, Robert Allen and Paul England, Comparison of automatic video segmentation algorithms, Proc. of SPIE Photonics West, 1995

[Dan+93] D. Daneels, D. Campenhout , W. Niblack, W. Equitz, R. Barber, E. Bellon, and F. Fierens. Interactive outhning: An improved approach using active contours. In Proc. SPIE Storage and Retrieval for Image and Video Databases, 1993.

[Der91] Rachid Deriche, Fast algorithms for Low-level Vision, IEEE Transaction on Pattern Anal. Mach. Intell, (PAMI) N° 1, pp. 78-87, 1991

[DFB97] Chabane Djeraba, Patrick FARGEAUD and Henri BRIAND, Retrieval by Content in an Image Database, Journée CNET, Coresa97, Paris

[DH73] R.O. Duda and P.E. Hart. Pattern Classification and Scene Analysis. Wiley, New York, 1973

[DK95] Sadashiva Devadiga, david A. Kosiba, Ullas Gargi, Scott Oswald et Rangachar Katsuri, A semiautomatic video database system, SPIE vol. 2420, pp 262-266, 1995

[Dow93] J. Dowe. Content-based retrieval in multimedia imaging. In Proc. SPIE Storage and Retrieval for Image and Video Databases, 1993.

[Dub99] B. Dubuisson. Vision et Image. Cours d'option SY23, Printemps 1999, UTC Compiègne

[Dum94] Susan Dumai. Latent semantic indexing (LSI) and TREC-2. In D. K. Harman, editor, The Second Text Retrieval Conference (TREC-2), pages 105-115, Gaithersburg, MD, March 1994. NIST. Special Publication 500-215

[Ege93] Max J.Egenhofer. What's Special About Spatial? Database Requirements for Vehicle Navigation in Geographic Space. In Proceedings of ACM SIGMOD, pages 398-402, 1993.

[ELS94] G. Evangedelis, D. Lomet and B. Salzberg. The hB^{II}-Tree: A concurrent and recoverable multi-attribute access method. Technical report NU-CCS-94-12, Northeastern University, College of CS , 1994

[EN94] W. Equitz and W. Niblack. Retrieving images from database using texture-algorithms from the QBIC system. Technical Report RJ 9805, Computer Science, IBM Research Report, May 1994

[EQVW] http://www.excalibur.com,http://wwwqbic.amaden.ibm.com, http://www.virage.com/ http://disney.ctr.columbia.edu/WebSEEk/

[Fal+93] C. Gfaloutsos et al. Efficient and effective querying by image content. Technical report, IBM Research Report, 1993

[Far+96] N. Faraj, R. Godin, R. Missaoui, S. David et P. Plante (1996). Analyse d'une méthode d'indexation automatique basée sur une analyse syntaxique de texte, Revue de l'information et de la bibliothéconomie, 21(1), 1-21

[FBF+94] Faloutsos C., Barber R., Flickner M., Hafner J., Niblack W., PetkovicD., and Equitz W. Efficient and Effective Querying by Image Content. Journal of Intelligent Information Systems, 3(1):231-262,1994.

[FD92] Peter W. Foltz and Susan T. Dumais. Personalized information delivery: an analysis of information filtering methods. Comm. Of ACM (CACM), 35(12):51-60, December 1992

[FL95] C. Faloutsos and King-Ip (david) Lin. Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In Proc. of SIGMOD, pages 163-174, 1995

[Fli+95] Myron Flicker et al. . Query by image and video content: the QBIC system, Computer vol 28, number 9, September 1995

[FM95] B. Furht, M. Milenkovic, "A Guided Tour of Multimedia Systems and Applications", IEEE Computer Society Press, 1995

[FSA96] C. Frankel, M.J. Swain, and V. Athitsos. Webseer: An image search engine for the world wide web. Technical Report TR-96-14, Computer Science Department, University of Chicago, 1996.

[Fuk90] Keinosuke Fukunaga. Introduction to Statistical Pattern Recognition. Academic Press, 1990. 2nd Edition

[Gag83] A. Gagalowicz. Vers un modèle de textures. Thèse d'Etat, Université Pierre et Marie Curie, Paris VI, 1983

[Gar+95] Ullas Gargi, Scott Oswald, David Kosiba, Sadashiva Devadiga and Rangachar Kasturi, Evaluation of video sequence indexing and hierarchical video indexing, SPIE Vol. 2420, pp 144-151

[GJ97] A. Gupta and R. Jain. Visual information retrieval. Communications of the ACM, 40(5), 1997

[GK90] C.C. Gotlieb and H.E.Kreyszig. Texture descriptors based on co-occurrence matrices. Computer Vision, Graphics, and Image Processing, 51, 1990

[Gon+94] Y. Gong, H. Zhang, H.C. Chuan, and M. Sakauchi. An image database system with content capturing and fast image indexing abilities. In Proc IEEE 1994, 1994.

[GP88] M. Gervautz and W. Purgtathofer, A simple method for color quantization , New Trends in Computer Graphics (Magnenat-Thalmann and Thalmann, eds.), pp. 219-231, Springer-Verlag

[GP97] F. Golshani and Y. Park.Content-based image indexing and retrieval in Image RoadMap, . In Proc. SPIE Multimedia Storage and Archiving Systems II, Vol. 3229, pp. 194-205, February 1997

[GR95] V. Gudivada and V. Raghavan, "Design and evaluation of algorithms for image retrieval by spatial similarity", ACM Trans. In Inf. Syst., 13(1): 115-144, April 1995

[Gra95] R. S. Gray. Content-based image retrieval: Color and edges. Technical report, Dartmouth University Department of Computer Science technical report #95-252, 1995.

[Gre89] Diane Green. An implementation and performance analysis of spatial data access. In Proc. ACM SIGMOD, 1989

[Gro+94] M.H. Gross, R. Koch, L. Lippert, and A. Dreger. Multiscale image texture analysis in wavelet spaces. In Proc. IEEE. Conf. on Image Proc. 1994

[Gro+97] P. Gros, R. Mohr, M. Gelgon et P. Bouthemy. Indexation de vidéos par le contenu, Coresa 1997, Paris

[Gün97] B. Günsel et al. "Hierarchical Temporal Video Segmentation and Content Characterization", Proc. Of SPIE, Multimedia Storage and Archiving Systems II, pp.46-56, Dallas 1997

[Gup95] Gupta, A.Visual Information Retrieval Technology, A VIRAGE Perspective. White paper, Virage Inc.,1995.

[Gut84] A. Guttman. R-trees: A dynamic index structure for spatial searching. In proceedings of ACM SIGMOD Conference, pages 47-57, June 1984.

[Hal89] G. Halin, Apprentissage pour la recherche interactive et progressive d'images,: processus EXPRIM et prototype RIVAGE, Thèse de Doctorat de l'Université de Nancy I, Octobre 1989

[Ham97] R. hammoud. La segmentation de la vidéo en scènes basée sur les indices spatio temporels, Rapport de DEA Contrôle De Systèmes (CDS), Université de Technologie de Compiègne, Septembre 1997

[HBB96] H. Haddad, C. Berrut, M.F. Bruandet, Un modèle vectoriel de recherche d'informations adapté aux documentd vidéo, CORESA' 96, Grenoble, pp. 281-289, 1996

[HCA95] D. Hang, B. Cheng, and R. Acharya. Texture-based Image Retrieval Using Fractal Codes. Technical Report 95-19, Department of Computer Science, State Unv. Of New York at Buffalo, 1995

[HCK90] Halin, G., Crehange, M., and Kerekes, P.Machine Learning and Vectorial Matching for an Image Retrieval Model: EXPRIM and the system RIVAGE. In Proceedings of ACM-SIGIR 1990, Brussels, Belgium, pages 99-114, 1990

[HCP95] Wynne Hsu, T.S.Chua, and H.K.Pung. An Integrated Color-Spatial Approach to Content-Based Image Retrieval. In Proceedings of ACM Multimedia Conference, pages 305-313,1995.

[HCT96] Chih-Cheng Hsu, Wesley W. Chu, and Rick K.Taira. A Knowledge-Based Approach for Retrieving Images by Content. IEEE-TKDE, 8:522-532, 1996.

[HJ94] P.W. Huang and Y.R. Jean, Using 2DC+-strings as spatial knowledge representation for image database systems, Pattern Recognition, vol. 27, no.9, pp.1249-1257, 1994

[HJW95a] Arun Hampapur, ramesh jain and Terry E. Weymouth, Production Model Based Digital Video Segmentation, Multimedia Tools and Applications, Vol.1, 1995 pp 9-46

[HK92] Kyoji Hirata and Toshikazu Kato. Query by visual example. In Proc of 3rd Int Conf on Extending Database Technology. 1992

[HKR93] D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge, Comparing images using the Hausdorff distance, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, pp. 850-863, September 1993

[HM93] Radu Horaud et Olivier Monga. Vision par ordinateur. Hermes Chapitre 4, Segmentations d'images en régions, pages 103-129, ISBN 2-86601-370-0

[HMR96] T.S. Huang, S. Mehrotra, and K. Ranmchandran. Multimedia analysis and retrieval system (MARS) project. In Proc. of 33rd Annual Clinic on Library Application of Data Processing - Digital Image Access and Retrieval, 1996

[HN83] K. Hinrichs and J. Nievergelt. The grid file: a data structure to support proximity queries on spatial objects. Proc of the WG'83 (Intern. Workshop on Graph. Theoretic Concepts in Computer Science), pages 100-113, 1983

[Hog+91] Hogan, M. et al. 1991. The visual thesaurus in a hypermedia environment : a preliminary exploration of conceptual issues and applications, International Conference on Hypermedia and Interactivity in Museums, Pittsburgh, October 1991, Archives and Museum Informatics, Pittsburgh, 1991 : 202-221.

[HP74] S.L Horowitz and T. Pavlidis. Picture segmentation by a directed split-and-merge procedure. In Proc. of the 2nd International Joint Conference on Pattern Recognition, pages 424-433, 1974

[HS79] G.M. Hunter and K. Steiglitz. Linear transformation of pictures represented by quadtrees. Computer Graphics and Image Processing 10, (3), 289-296

[HSD73] Robert M. Haralick, K. Shanmugam, and Its'hak Dinstein. Texture features for image classification. IEEE Trans. On Syst. Man. And Cyb. SMC-3(6), 1973

[Hu61] M.K. Hu, Pattern Recognition by Moments Invariants, Proc. of the IEEE, vol. 49, no. 9, September 1961, p. 1428

[Hu62] M.K. Hu. Visual pattern recognition by moment invariants, computer methods in image analysis. IRE Trans. On Information's Theory, 8, 1962

[Hua+97] J. Huang et al. Image indexing using color correlogram. In Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, 1997

[Hua97] P. Huang, "Indexing pictures by key objects for large-scale image databases", Pat. Rec., Vol.30, N. 7 p1229-1237, 1997

[Hun89] R. W. G. Hunt, Measuring Color. Ellis Horwood series in applied science and industrial technology. Halsted Press, New York, NY, 1989.

[IEEE96] IEEE 1996. Special Section on Digital Libraries: Representation and Retrieval, IEEE Trans. On Pattern Analysis and Machine Intelligence, 18 (8)

[Iok89] Mikihiro Ioka. A method of defining the similarity of images on the basis of color information. Technical Report RT-0030, IBM Research, Tokyo Research Laboratory, Nov. 89

[IP 97] F. Idris and S. Panchanathan (1997), Review of image and video indexing techniques, Journal of Visual Communication and Image Representations, 8(2), 146-166

[Jai97] Ramesh Jain. Guest Editor, Special issue on visual information management communications of ACM, December 1997.

[JG95] Jung, G.S. and Gudivada, V. Adaptive Query reformulation in Attribute based Image Retrieval. In Intelligent Systems, pages 763-774, 1995.

[Jol96] Philippe Joly, Consultation et Analyse des documents en image animée numérique, Thèse de Doctorat, Université Paul Sabatier, 1996

[Jol98] Jean-Michel Jolion, Indexation d'images: nouvelle problématique ou vieux débat?, Rapport de Recherche 9805, Laboratoire Reconnaissance de Formes et Vision, INSA, Lyon, 1998

[Jon81] K.S. Jones. Information Retrieval Experiment. Butterworth and Co., 1981

[JPP95] R. Jain, A. Pentland, and D. Petkovic. NSF-ARPA workshop on visual information management systems. Cambridge, MA, June 1995.

[Jul62] B. Julesz. Visual pattern recognition, IEEE Trans. on Information Theory, vol. 8, 1962

[Jun88] E. Jungert, Extended symbolic projection as a knowledge structure for image database systems, Fourth BPRA Conference on Pattern Recognition, 1988, pp.343-351, Springer-Verlag

[JV95] Anil K. Jain et A. Vailaya, "Image Retrieval using Color and Shape", May 15, 1995

[JZL96] A. Jain, Y. Zhong, and S. Lakshmanan. Object Matching Using Deformable Templates. IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 408-439, March 1996.

[Kan96] H.R.Kang, "Color technology for electronic imaging devices", SPIE, ISBN 0-8194-2108-1, 1996

[Kat+92] Kato, T., Kurita, T., Otsu, N., and Hirata, K. A Sketch Retrieval Method for Full Color Image Database, Query by Visual Example. In IEEE-IAPR-11, pages 530-533, August-September 1992.

[KC92] A. Kundu and J. Chen. Texture classification using qmf bank-based subband decomposition. CVGIP: Graphical Models and Image Processing, 54(5): 369-384, Sep. 1992

[KF94] I. Kamel and C. Faloutsos. Hilbert R-tree: AN Improved R-tree using Fractals. VLDB 1994: 500-509

[Kim97] Hae-Kwang KIM, Détection automatique des mouvements de caméra et des régions de textes pour la structuration et l'indexation de documents audiovisuels, Thèse de doctorat, Université Paul Sabatier de Toulouse, 1997

[KJ91] R. Kasturi and R. Jain, Dynamic vision, in Computer Vision: Principles, pp. 469-480, IEEE Comput. Soc., Los Alamitos, CA, 1991

[KLS95] D. Kapur, Y.N.Lakshman, and T. Saxena. Computing invariants using elimination methods. In Proc. IEEE Int. Conf. on Image Proc. 1995

[Kra88] Krauze, Michael G. 1988. Intellectual problems of indexing picture collections. Audiovisual Librarian 14, no 4 (November) : 73-81.

[Kun93] M. Kunt, Traitement de l'information: volume 2, Traitement numérique des images. Presses polytechniques et universitaires romandes, 1993

[Lau89] Robert Laurini. L'ingénierie des connaissances spatiales. Chap.2: Formalismes de modélisation. et Chap. 3: Modélisation conceptuelle géométrique. Hermes

[Lee+92] S.Lee et al. Signature files as a spatial filter for iconic image database, J. Visual Lang. Comput. 92, 305-397

[Lee+94] D. Lee, R. Barber, W. Niblack, M. Flickner, J. Hafner, and D. Petkovic. Indexing for complex queries on a query-by-content image database. In Proc. IEEE Int. Conf. on Image Proc., 1994.

[Leu+92] Leung, C.H.C. et al. 1992. Picture retrieval by content description. Journal of Information Science 18, no 2 : 111-119.

[Lev85] Martin D. Levine, Vision in man and machine, Mc Graw Hill eds., Chap. 10 Shape, pp.480-544, 1985

[LF93] A. Laine and J. Fan. Texture classification by wavelet packet signatures. In IEEE Trans. Patt. Recog. and Mach. Intell., 15(11):1186-1191, 1993

[LG96] B. Lamiroy and P. Gros, Rapid object indexing and recognition using enhanced geometric hashing. In Procedings of the 4th European Conference on Computer Vision Cambridge, England, volume 1, pages 59-70, Avril 1996

[LH92] S-Y Lee and F.–J.Hsu . Spatial reasoning and similarity retrieval of images using 2D C String knowledge representation. Pattern Recognition, 25(3):305-318,1992

[LHR97] Wei Li, Véronique Haese-Coat and Joseph Ronsin. Residues of morphological filtering by reconstruction for texture classification. Pattern Recognition, Vol. 30, No.7, 1081-1093, 1997

[Li+97] Wen-Syan Li, K.S. Candan, Kyoji Hirata, and Yoshinori Hara. SEMCOG: an object-based image retrieval system and its visual query interface. In proceedings of ACM SIGMOD, pages 521-524, June 1997.

[LJF94] K-L. Lin, H. Jagadish, and C. Faloutsos. The TV-tree-An index structure for high-dimensional data. VLDB Journal, 3:517-542, 1994

[LKC95] Z. Lei, D. Keren and D.B. Cooper. Computationally fast bayesian recognition of complex objetcs based on mutual algebraic invariants. In Proc. IEEE Int. Conf. on Image Proc, 1995

[LM95] B. Li and S. De Ma. On the relation between region and contour representation. In Proc. IEEE Int. Conf. on Image Proc., 1995

[LOT94] H Lu, B. Ooi, and K. Tan. Efficient image retrieval by color contents. In Proc. of the 1994 Int. Conf. on Applications of Databases, 1994

[LS90] David B. Lomet and Betty Salzberg. The hb-tree: a multi-attribute indexing method with good guaranteed performance. ACM TODS, 15(4):625-658, December 1990

[LZ95] H.-C. H. Liu and G.L. Zick, Scene decomposition of MPEG compressed video, Digital Video Compression: Algorithms Tech. 2419, February 1995, 26-37

[Mar97] André Marion. Acquisition & visualisation des images, , pp. 32, ISBN: 2-212-08871-X, Eyrolles 1997

[Meh+95] B.M. Mehtre, M. S. Kankanhalli, A.D. Narsimhalu, and G.C. Man. Color matching for image retrieval, Pattern Recognition Letters, Vol.16, pp. 325-331, March 1995

[Meh+97a] S. Mehrotra, Y. Rui, M. Ortega-B, and T.S. Huang. Supporting content-based queries over images in MARS. In Proc. of IEEE Int. Conf. on Multimedia Computing and Systems, 1997

[Meh+97b] S. Mehrotra, K. Chakrabarti, M. Ortega, Y. Rui, and T. S. Huang. Multimedia analysis and retrieval system. In Proc. of The 3rd Int. Workshop on Information Retrieval Systems, 1997.

[MG95] Nabil Madrane and Morris Goldberg, Video representation tools using a unified object and perspective based approach, SPIE Vol. 2420, pp. 152-163, 1995

[Mil95] Georges A. Miller. Wordnet: a lexical database for English, http://www.cogsci.princeton.edu/~wn/

[Miy88] Makoto Miyahara, Mathematical transform of (r,g,b) color data to munsell (h,s,v) color data, In SPIE Visual Communications and Image Processing, Vol. 1001, 1988

[MJC95] Jianhao Meng, Yujen Juan, Shi-Fu Chang, Scene change detection in a MPEG Compressed Video Sequence, Proc. IS&T SPIE '95 Digital Video Compression: Algorithm and Technologies, San Jose, Vol. 2419, 1995, pp 14-25

[MJF96] B.Moon, H.V.Jagadish, C.Faloutsos, J.H.Saltz, « Analysis of the Clustering Properties of Hilbert Space-filling curve », IEEE Transaction on Knowledge and Data Engineering, March 1996

[MM92] F. Mokhtarian and Alan K. Mackworth, A Theory of Multiscale, Curvature-Based Shape representation for Planar Curves, IEEE PAMI, Vol. 14, pp. 789-805, 1992

[MM95a] W.Y. Ma and B.S. Manjunath. A comparison of wavelet transform features for texture image annotation. In Proc. IEEE Int. Conf. on Image Proc., 1995

[MM95b] B.S. Manjunath and W.Y. Ma. Image indexing using a texture dictionary. In Proceedings of SPIE Conference on Image Storage and Archiving System, volume 2606,.995

[MM96a] W.Y. Ma and B.S. Manjunath. Texture features for browsing and retrieval of image data. IEEE T-PAMI special issue on Digital Libraries, Nov 1996.

[MM96b] W.Y. Ma and B.S. Manjunath.Texture features and learning similarity. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pages 425-430, 1996.

[MM97] W.Y. Ma and B.S. Manjunath. Netra: A toolbox for navigating large image databases. In Proc. IEEE Int Conf. on Image Proc., 1997.

[MM98] S. Matusiak et M. Daoudi, Indexation Robuste et Recherche d'Images par le Dessin, Journées CNET, CORESA 98,

[MMD76] C.S. McCamy, H. Marcus, and J.G. Davidson. A color-rendition chart, Journal of Applied Photographic Engineering, 2(3), Summer 1976

[Mok95] F. Mokhtarian, Silhouette-Based Isolated Object Recognition Through Curvature Scale Space, IEEE PAMI, Vol. 17, No 15, pp. 539-544, 1995

[Mol96] Thierry Molinier. Décomposition en "Tétra-Arbres". Rapport de DEA, UTC de Compiègne, Juillet 1996

[MP95] S. Mann and R.W. Picard, Video orbits: characterizing the coordinate transformation between two images using the projective group, MIT Media Lab. Technical report No. 278, 1995

[MP96] T.P. Minka and R.W. Picard. Interactive learning using a "society of models". In Proc. IEEE CVPR, pages 447-452, 1996.

[MPEG7] MPEG7: Context and Objectives (v.3), ISO/IEC JTC1/SC29/WG11 N1678, http://www.cselt.stet.it/mpeg/mpeg_7.htm.

[Mur84] Murry, Rita. 1984. Criteria for Subject Indexing of Pictures : an Introductory Survey. Master diss., University of Alberta.

[MY88] M. Miyahara and Y. Yoshida. Mathematical transform of (r, g, b) color data to Munsell (h, v, c) color data. In SPIE Visual Communications and Image Processing 1988, vol. 1001 1988.

[Nar95] A.D.Narasimhalu, Guest Editor, Special Issue on Content-Based Retrieval, ACM Multimedia Systems, Vol. 3(1), Feb. 1995

[Nar95b] A.D.Narasimhalu. Special section on content-based retrieval. Multimedia Systems. 1995.

[Nas+98a] C. Nastar, M. Mitschke, C. Meilhac, N. Boujemaa. Surfimage: a flexible content-based image retrieval system, ACM-Multimedia 1998

[Nas+98b] C. Nastar, N. Boujemaa., M. Mitschke, C. Meilhac, Surfimage: un système flexible d'indexation et de recherche d'images .CORESA 98, Lannion 9-10 juin 1998

[Nas97] Chahab Nastar. Indexation d'Images par le Contenu: un Etat de l'Art, journées CNET CORESA, 1997

[NH88] A. N. Netravali and B. G. Haskell. Digital Pictures; representation and compression. Applications of Communications Theory. Plenum Press, New York, NY, 1988.

[NHS84] J. Nievergelt, H. Hinterberger and K.C. Sevcik. The grid file: an adaptable, symmetric multikey file structure. Proc. of the ACM TODS, 9(1):38-71

[Nib+93] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, and C. Faloutsos. The QBIC project: Querying images by content using color, texture, and shape. IBM RJ 9203 (81511), February 1993.

[Nib+94] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, and C. Faloutsos. The QBIC project: Querying images by content using color, texture, and shape. In Proc. SPIE Storage and Retrieval for image and Video Databases, February 1994.

[NP94] E. Nardelli and G. Poietti. A Hybrid Pointerless Representation of Quadtrees for Efficient Processing of Window Queries. IGIS 1994: 256-269, 1994

[NS96] Raymond Ng and Andishe Sedighian. Evaluating multi-dimensional indexing structures for images transformed by principal component analysis. In Proc. SPIE Storage and Retrieval for Image and Video Databases, 1996.

[NT91] Akio Nagasaka et Yuzuru Tanaka, Automatic video indexing and full-video serach for object appearances, Proc. of the IFIP, Second Working conference on Visual Database Systems, Budapest, 1991, pp 113-127

[OAH96] Atsushi Ono, Masashi Amano, and Mituhiro Hakaridani. A flexible content-based image retrieval system with combined scene description keyword. In Proc. IEEE Conf. on Multimedia Computing and Systems, 1996.

[OD92] Philippe P. Ohanian and Richard C. Dubes. Performance evaluation for four classes of texture features. Pattern Recognition, 25(8): 819-833, 1992

[Ohl82] Ohlgren, Thomas H. 1982. Image analysis and indexing in North America : a survey. Art Libraries Journal (summer) : 51-60.

[Ort+97] M. Ortega, Y. Rui, K. Chakrabarti, S. Mehrotra, and T. S. Huang. Supporting similarity queries in MARS. In Proc. of ACM Conf. on Multimedia, 1997

[OS95] VE. Ogle and M. Stonebraker. Chabot, Retrieval from a Relational Database of Images, Special issue on Content-Based Image Retrieval Systems, Computer, Vol. 28(9), Sept. 1995, pp.40-48

[Pap+95] Dimitris Papadias, Timos Sellis, Yannis Theodorakis, and Max J. Egenhofer. Topological Relations in the Worl of Minimum Bounding Rectangles: A Study with R-trees. In Proceedings of ACM SIGMOD, pages 92-103, 1995.

[Pap+98] T.V. Papathomas, T.E. Conway, I.J. Cox, J. Ghosn, M.L. Miller, T.P. Minka, and P.N. Yianilos. Psychophysical studies of the performance of an image database retrieval system. In IS&T/SPIE Conf on Human Vision and Electronic Imaging III, 1998.

[Pav86] T. Pavlidis. A critical survey of image analysis methods. In Proc. of the Eighth International Conference on Pattern Recognition, pages 502-511, October 1986

[Pen84] A.P. Pentland. Fractal-based description of natural scenes. IEEE Trans. Patt. Recog. and Mach. Intell. 6(6): 661-674, 1984

[PF77] E. Persoon and K.S. Fu. Shape discrimination using Fourier descriptors. IEEE Trans. Sys. Man, Cyb. 1977

[PF95] E.G.M. Petrakis and C. Faloutsos. Similarity Searching in Large Image Databases. Technical Report 3388, Department of Computer Science, University of Mryland, 1995.

[Phil88] S.Philipp. Analyse de texture appliquée aux radiographies industrielles. Thèse de l'Université P. et M. Curie, Paris VI, 1988

[Pic95] R.W. Picard. Toward a visual thesaurus. In Springer Verlag Workshop in Computing, MIRO 95. 1995.

[Pic96a] R.W. Picard.Digital libraries: Meeting place for high-level and low-level vision. In Proc Asian Conf on Comp. Vis. 1996

[Pic96b] R.W. Picard. Computer learning of subjectivity. In Proc. ACM Computing Surveys. 1996

[PM93] William B. Pennebaker and Joan L. Mitchell, JPEG Still Image Data Compression Standard. New York: Van Nostrand Reinhold, 1993

[PM96] R.W. Picard and T.P. Minka. Vision texture for annotation. Multimedia Systems: Special Issue on Content-based retrieval.1996

[PMS96] R.W. Picard, T. P. Minka, and M. Szummer. Modeling user subjectivity in image libraries. In Proc. IEEE Int. Conf. on Image Proc. Lausanne, Sept 1996.

[PP96] A. Pentland and R. Picard. Special issue on digital librarie. IEEE Trans. Pattern. Recognition and Machine Intelligence, 1996

[PPS94] A. Pentland, R. Picard, and S. Sclaroff. Photobook: Tools for content-based manipulation of image databases. In SPIE Paper 2185-05, Storage and Retrieval of Image and Video Databases II, San Jose, CA, pages 34-47, 1994.

[PPS96] A. Pentland, R. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. International Journal of Computer Vision 1996.

[PZM96] G.Pass, R.Zabih, J.Miller, « Comparing Images Using Color Coherence Vectors », Proc.of the 4th ACM Intl. Multimedia Conference, November 18-22, 1996, Boston, MA, pp.65-74

[RHM97a] Y. Rui, T. S. Huang, and S. Mehrotra. Content-based image retrieval with relevance feedback in MARS. In Proc. IEEE Int. Conf. on Image Proc., 1997

[RHM97b] Y. Rui, T. S. Huang, and S. Mehrotra. MARS and its applications to MPEG-7. ISO/IE JTC1/SC29/wg11 M2290, MPEG97, july 1997.

[RHM97c] Yong Rui, Thomas S. Hua,g, and Sharad Mehrotra. Content-based image retrieval with relevance feedback in MARS. In Proc. IEEE Int. Conf. on Image Proc. 1997

[Rij81] C.J. van Rijsbergen, Retrieval Effectiveness, Information Retrieval Experiment in K.S.Jones, Ed), pp.32-43, Butterworths,Stoneham,MA,1981

[RJ93] Lawrence Rabiner and Biing-Hwang Juang. Fundamentals of Speech Recognition, chapter 5. Prentice hall, Englewood Cliffs, New jersey, 1993.

[RJB89] V.V. Raghavan, G.S. Jung and P. Bollmann (1989). A Critical Investigation of Recall and Precision as Measures of Retrieval System Performance. ACM TOIS, 7(3), 205-229

[Rob81] J.T. Robinson. The K-D-B tree: A search structure for large multidimensional dynamic indexes. In Proceedings of the ACM SIGMOD International Conference on the Management of Data, pages 10-18

[Rod91] Roddy, Kevin. 1991. Subject access to visual resources : what the 90s might portend. Library Hi Tech 9 : 1 (no 33) : 45-49.

[Roh90] Eric Rohmer, "Conte de Printemps", L'Avant-Scène du Cinéma, 392, May 1990 (Le script du film en français)

[RS96] R. Rickman and J. Stonham. Content-based image retrieval using color tuple histograms. In Proc. SPIE Storage and Retrieval for Image and Video Databases, 1996

[RSH96a] Y.Rui, A.C. She, and T.S. Huang . Modified Fourier descriptors for shape representation - a practical approach. In Proc of First International Workshop on Image Databases and Multi Media Search, 1996

[RSH96b] Y.Rui, A.C. She, and T.S. Huang. Automated shape segmentation using attraction-based grouping in spatial-color-texture space. In Proc. IEEE Int. Conf. on Image Proc.,1996.

[Rui+97a] Y. Rui, T.S. Huang, S. Mehrotra, and M. Ortega. Automatic matching tool selection using relevance feedback in MARS. In Proc. of 2nd Int. Conf. on Visual Information Systems, 1997.

[Rui+97b] Y. Rui, T.S. Huang, S. Mehrotra, and M. Ortega. A relevance feedback architecture in content-based multimedia information retrieval systems. In Proc. of IEEE Workshop on Content-based Access of Image and Video Libraries, in conjunction with IEEE CVPR'97, 1997.

[Rui+97c] Y. Rui, K. Chakrabarti, S. Mehrotra, Y. Zhao, and T.S. Huang. Dynamic clustering for optimal retrieval in high dimensional multimedia databases. In TR-MARS-10-97, 1997.

[Rui+98] Y. Rui, T.S. Huang, S. Mehrotra, and M. Ortega. Relevance feedback: A power tool in interactive content-based image retrieval. IEEE Tran on Circuits and Systems for Video Technology, Special Issue on Interactive Multimedia Systems for the Internet, Sept 1998;

[Rus95] J. C. Russ. The Image Processing Handbook. CRC Press, Boca Raton, 1995.

[SAF94] B. Scassellati, S. Alexopoulos, and M. Flickner. Retrieving images by 2nd shape: A comparison of computation methods with human perceptual judgments. In Proc. SPIE Storage and Retrieval for Image and Video Databases, 1994.

[Sal86] G. Salton (1986). Another Look at Automatic Text-Retrieval Systems. Communications ACM, 29(7), 648-656

[Sal89] G. Salton, Automatic Text Processing, Addison-Weslay, 1989

[Sam82] H. Samet,. Neighbor finding techniques for images represented by quadtrees, CHIP, vol. 18, 298-303, 1982

[Sam89] Hanan Samet. The Design and Analysis of Spatial Data Structures. Addison-Wesley, 1989.

[SB87] G. Salton and C. Buckley, Text weighting approaches in automatic text retrieval, Technical Report 87-881, Cornell University, 1987

[SB88] G. Salton, C. Buckley (1988). Term Weighting approaches in automatic text retrieval. Information Processing and Management, 24(5), 513-523

[SB91] M. J. Swain and D. H. Ballard. Color indexing. International Journal of Computer Vision, 7:1 1991.

[SC94] J. R. Smith and S. Chang. Transform features for texture classification and discrimination in large image databases. In Proc. IEEE Int. Conf. on Image Proc., 1994

[SC95a] John R. Smith and Shih-Fu Chang. Single color extraction and image query. In Proc. IEEE Int. Conf. on Image Proc., 1995

[SC95b] John R. Smith and Shih-Fu Chang. Tools and techniques for color image retrieval. In IS &T/SPIE proceedings Vol. 2670, Storage and Retrieval for Image and Video Databases IV, 1995

[SC96] Bruce Schatz and Hsinchun Chen. Building Large-scale digital libraries, Computer 1996

[SC96a] J.R. Smith and S-F Chang. Intelligent Multimedia Information Retrieval, edited by Mark T. Maybury, chapter Querying by Color Regions Using the VisualSEEk Content-Based Visual Query System, 1996

[SC96b] John R. Smith and Shih-fu Chang. VisualSEEk: a fully automated content-based image query system. In proceedings of ACM Multimedia Conference, Boston MA, pages 87-98, 1996.

[SC96c] John R. Smith and Shih-Fu Chang. Automated binary texture feature sets for image retrieval. In Proc. IEEE Int. Conf. Acoust, Speech and Signal Proc. May 1996

[SC96d] J.R. Smith and S-F.Chang. Local color and texture extraction and spatial query. In Proc. IEEE Int. Conf. on Image Proc., 1996

[SC97a] John R. Smith and Shih--Fu Chang. Multi-stage classification of images from features and related text. In 4th Europe EDLOS Workshop, San Miniato, Italy Aug 1997

[SC97b] John R. Smith and Shih--Fu Chang. Enhancing image search engines in visual information environments. In IEEE 1st Multimedia Signal Processing Workshop, June 1997

[SC97c] J.R. Smith and S-F. Chang. Visually searching the web for content. IEEE Multimedia Magazine, 4(3):12-20, Summer 1997,

[Sch 94] R. Schettini, Multicolored object recognition and location, Pattern Recognition Letters, vol.15, pp. 1089-1097, November 1994

[Sch96] C. Schmid. Appariement d'images par invariants locaux de niveaux de gris. Thèse de doctorat, GRAVIR-IMAG-INRIA Rhône-Alpes, juillet 1996

[SD96] M. Stricker and A. Dimai. Color indexing with weak spatial constraints. In Proc. SPIE Storage and Retrieval for Image and Video Databases 1996

[Sel90] Seloff, G.A. 1990. Automated access to the NASA-JSC image archives. Library Trends 38, no 4 : 682-696.

[SG83] G. Salton and M.M. Gill, Introduction to Modern Information Retrieval, Mac Graw Hill Book Company, New York, 1983

[Sha94] Shatford, Sara 1994. Some Issues in the Indexing of Images. Journal of the American Society for Information Science 45, no 8 : 538-588.

[Sha95] W.M. Shaw. Term-relevance computations and perfect retrieval performance. Information Processing and Management. 1995

[SHB93] Sonka, M. and Hlavac, V. and Boyle, R. Image Proceeding, Analysis, and Machine Vision. Chapman and Hall, 1993.

[She97] Bo Shen, HDH based compressed video cut detection; HPL-97-142 971204 External, http://www.hpl.hp.com/techreports/97/HPL-97-142.html

[SL2000] John R. Smith and Chung-Sheng Li, Image Classification and Querying using Composite Region Templates*, To appear in Journal of Computer Vision and Image Understanding - special issue on Content+Based Access of Image and Video Libraries

[SM83] G. Salton and Michael.J.MacGill, Introduction to Modern Information Retrieval, Mac Graw Hill Book Company, New York, 1983.

[Sma94] Malika Smaïl, Raisonnement à base de cas pour une recherche évolutive d'information, Prototype Cabri-n. Vers la définition d'un cadre d'acquisition de connaissances, Thèse de doctorat, Université Henri Poincaré, Nancu I, 1994

[Sma98] Malika Smaïl, Vers des systèmes évolutifs de recherche d'information: un état de l'art, Techniques et Sciences informatiques, Volume 17 - n° 10/1998, pages 1193 à 1222

[SO95] M. Stricker and M. Orengo. Similarity of color images. In Storage and Retrieval for Image and Video Databases III, volume SPIE Vol. 2420, February 1995.

[SP95] Ishwar K. Dethi and Nilesh Patel, A Statistical approach to scene change detection, SPIE Vol. 2420, pp. 329-336, 1995

[Spr91] R. Sproull. Refinements to nearest-neighbor searching in k-dimensional objects. Algorithmica, 6(4): 579-589

[SQ96] Alan F. Smeaton and Ian Qigley. Experiments on Using Semantic Distances Between Words in Image Caption Retrieval. In Proceedings of ACM SIGIR Conference, 1996.

[SRF87] T. Sellis, N. Roussopoulos, and C. Faloutsos. The R+tree: A dynamic index for multi-dimensional objects. In Proc 12th VLDB, 1987

[Sri95] R.K. Srihari. Automatic indexing and content-based retrieval of captioned images. IEEE Computers Magazine, 28(9), 1995

[STL97] S. Sclaroff, L. Taycher, and M. La Cascia. Imagerover: A content based image browser for the world wide web. In Proc IEEE Workshop on Content-based Access of image and Video Libraries, 1997.

[SV95] François Salazar et Franck Valéro. Analyse automatique de documents vidéo, Rapport de Recherche, IRIT 95-28-R, Université Paul Sabatier, 1995

[Sve94] Svenonius, Elaine. 1994. Access to Nonbook Materials : The Limits of Subject Indexing forVisual and Aural Languages. Journal of the American Society for Information Science 45, no 8 : 600-606.

[SW95] D.L. Swets and J.J. Weng. Efficient content-based image retrieval using automatic feature selection. In Proc. IEEE 1995, 1995

[SYW75] G. Salton, C.S. Yang, and A. Wong, A vector space model for automatic indexing, Comm.of the ACM, 18(11):613-620, November 1975

[Tag97] H. Tagare. Increasing retrieval efficiency by index tree adaptation. In Proc. of IEEE Workshop on Content-based Access of Image and Video Libraries, in conjunction with IEEE CVPR'97, 1997

[Tau91] G. Taubin. Recognition and positioning of rigid objects using algebraic moment invariants. In SPIE Vol. 1570 Geometric Methods in Computer Vision, 1991

[TMY78] H. Tamura, S. Mori, and T. Yamawaki. Texture features corresponding to visual perception. IEEE Trans. on Sys. Man. and Cyb., SMC-8(6), 1978

[TNP94] K.S. Thyagarajan, T. Nguyen, and C. Persons. A maximum likelihood approach to texture classification using wavelet transform. In Proc. IEEE Int. Conf. on Image Proc. 1994

[TT90] Tomika, F. and Tsuji Saburo. Computer Analysis of Visual Textures. Kluwer Academic Publishers, 1990.

[Tur94] Turner, James M..1994. Indexing " Ordinary " Pictures for Storage and Retrieval. Visual Resources X : 265-273.

[Ull82] J. Ullman, Principles of databases systems, Computer science press, London (UK), 2nd edition, 1982

[UMY92] Hirota Ueda, Takaumi Miyatake et Satoshi Yoshizawa, IMPACT: An interactive natural motion-picture dedicated multimedia authoring system, INTERCHI '91, ACM, 1991, pp 343-350

[Vet84] M. Vetterli. Multi-dimensional sub-band coding: Some theory and algorithms. Signal Processing, pages 97 - 112, April 1984.

[Vet95] M. Vetterli and J. Kova?evi?. Wavelets and Subband Coding. Prentice-Hall, Inc, Englewood Cliffs, NJ, 1995.

[VK95] A. Vellaikal and C.-C. J. Kuo. Content-based image retrieval using multiresolution histogram representation. In C.-C. J. Kuo, editor,Digital Image Storage and Archiving Systems, volume 2606 of SPIE Proceedings, pages 312--323, Philadelphia, PA, USA, October 1995.

[Voo93] Ellen M. Voorhees. Using WordNet to Disambiguate Word senses for Text Retrieval. In proceedings of ACM SIGIR Conference, pages 171-180, 1993.

[WDR76] J. Weszka, C.Dyer, and A. Rosenfeld. A comparative study of texture measures for terrain classification. IEEE Trans. On Sys. Man, and Xyb. SMC-6(4), 1976

[Wee96] A.R. Weeks. Fundamentals of electronic image processing. SPIE/IEEE Series on imaging science & engineering SPIE ISBN 0-8194-2149-9

[WJ96a] D.A. White and R. Jain. Similarity indexing: algorithms and performance. In Proc.of the SPIE: Storage and Retrieval for Image and Video Databases IV, San Jose, CA, Vol. 2670, pages 62-75.

[WJ96b] D.A. White and R. Jain. Similarity indexing with the SS-tree. IN Proc. 12th IEEE International Conference on Data Engineering, pages 516-523, New Orleans, 1996

[WK96a] Xia Wan and C.-C. Jay Kuo. Image Retrieval Based on JPEG Compressed Data. Proc. SPIE Multimedia Storage and Archiving Systems, Volume 2916, pp.104-115, Nov. 1996

[WK96b] X. Wan and C.-C. J. Kuo. Color analysis and quantization for image retrieval. Storage Retrieval Still Image Video Databases IV 2470, February 1996, 8-16

[WK97] Xia Wan and C.-C.Jay Kuo. Pruned Octree Feature for Interactive Retrieval. Proc. of SPIE Multimedia Storage and Archiving Systems II, Vol. 3229,pp.182-193, Dallas, 1997

[WMB94] I. H. Witten, A. Moffat, and T. C. Bell. Managing Gigabytes: compressing and indexing documents and images. Van Nostrand Reinhold, New York, NY, 1994.

[Wol95] Wayne Wolf, Key Frame Selection by Motion Analysis, ICASSP'95

[Wu+95] J.K. W, A. Narasimhalu, B.M. Mehtre, C.P. Lam, and Y.J. Gao. Core: a content-based retrieval engine for multimedia information systems. Multimedia systems, 1995

[WW80] I. Wallace and P. Wintz. An efficient three-dimensional aircraft recognition algorithm using normalized fourier descriptors. Computer Graphics and Image Processing, 13, 1980

[WWY92] Z. Wand, S. Wong, and Y. Yao, An analysis of vector space models based on computational geometry , in ACM SIGIR International Conference on Research and Development in Information Retrieval, Copenhagen, 1992, pp. 152-160

[WYA97] Jia Wang, Wen-Jann Yang, and Raj Acharya. Color clustering techniques for color-content-based image retrieval from image databases. In Proc. IEEE Conf. on Multimedia Computing and Systems, 1997

[WYS82] G. Wyszecki and W. S. Stiles. Color science : concepts and methods, quantitative data and formulae. The Wiley series in pure and applied optics. John Wiley & Sons, Inc., New York, NY, 2nd ed. edition, 1982..

[XLI95] Wei Xiong, John Chung-Mong Lee and Man-Ching Ip, Net Comparison : a fast and effective method for classifying image sequencess, Storage and Retrieval for Image and Video Databases III, SPIE Vol. 2420, 1995, pp318-328

[YA94] L. Yang and F. Algregtsen. Fast computation of invariant geometric moments: A new method giving correct results. In Proc. IEEE Int. Conf. on Image Proc., 1994.

[Yeu+95] Minerva Yeung, Boon-Lock Yeo, Wayne Wolf and Bede Liu, Video browsing using clustering and scene transitions on compressed sequences, Proc. Miltimedia Computing and Networking, -6-8 February 1995, San Jose, USA SPIE, Vol. 2417, pp 389-398

[YL95] B.L. Yeo and B. Liu. Rapid scene analysis on compressed video, IEEE Trans. on circuits and systems for video technology, vol. 5, 533-544, 1995

[YM98] Clement T. Yu and Weiyi Meng. Principles of Database Query Processing for Advanced Applications. Data Management Systems. Morgan Kaufmann, 1998.

[YVD95] Qi Yang, Asha Vellaikal and Son Dao. MB+-Tree: A New Index Structure For MultimediaDatabases.1995, ttp://biron.usc.edu/~vellaika/papers/mmdbms95.ps

[YW97] Hong Heather Yu and Wayne Wolf. Hierarchical, multi-resolution algorithms for dictionary-driven content-based image retrieval. In Proc. IEEE Int. Conf. on Image Proc., 1997.

[Zha+94] H.J. Zhang, C.Y.Low, Y. Gong, S.W. Smoliar, and S.Y. Tan, Video Parsing compressed data, Proc, SPIE: Image Video Processing II 2182, 1994, 142-149

[Zha+95] H.J. Zhang, C.Y.Low, S.W. Smoliar, and S.Y. Tan, Video Parsing and browsing using compressed data, Multimedia Tools Applications 1, 1995, 89-111

[Zha+96] J. Zhao, Y. Shimazu, K. Ohta, R. Hayasaka, and Y. Marsushita, JPEG Codec Adaptive to Region Importance, ACM Multimedia'96, 92140, ISBN 0-201-92140-X, Nov 1996, pp. 209-218

[Zha+97] Hong Jiang Zhang, Jianhua Wu, Di Zhong and Stephen W. Smoliar, An integrated system for content-based video retrieval and browsing, Pattern Recognition, Vol. 30, No. 4, pp. 643-658, 1997

[ZMM95] Ramin Zabih, Justin Miller and Kevin Mai, The feature-based algorithm for detcting and classifying scene breaks, ACM Multimedia'95, San Fransisco, November 1995, pp 189-200

[ZR72] C.T. Zahn and R.Z. Roskies. Fourier descriptors for plane closed curves. IEEE Trans. On Computers, 1972

[Zuc76] S.W. Zucker. Region growing: Chilhood and adolescence. Computer Graphics and Image Processing, 5:382-399, 1976

[ZZ95] HongJiang Zhang and Di Zhong. A scheme for visual feature based image indexing, Storage and Retrieval for Image and Video Databases III, SPIE Vol. 2420, pp36-45

FIN

[CC97] Y. Chahir, L. Chen, "Peano key rediscovery for content based retrieval of images", Proc. of SPIE Conf. on Multimedia Storage and Archiving Systems, Vol. 3229, Dallas, USA, Nov. 1997, ISBN 0-8194-2662-8

[CC98] Y. Chahir, L. Chen, "Spatialized Visual Features-Based Image Retrieval", Proc. of the ISCA, 14 th International Conference on Computers and Their Applications, ISBN: 1-880843-27-7 , pp. 174-179, Cancun, Mexico ,April 7-9, 1998

[CC99a] Y. Chahir, L. Chen, "Spatialized Visual Features-Based Image Retrieval", article à paraître dans la revue "International Journal of Computers and Their Applications", au mois de septembre ou décembre 99.

[CC99b] Y. Chahir, L. Chen, " Efficient Content-Based Image Retrieval based on color homogenous objects segmentation and their spatial relationship characterization ", IEEE Multimedia Systems'99, International Conference on Multimedia Computing and Systems ,June 7-11, 1999, Florence, ITALY

[CC99c] Y. Chahir and L. Chen , "Searching Images on the basis of color homogeneous objects and their spatial relationship", Papier accepté, à la revue "Journal of Visual Communication and Image Representation" , à apparaître fin 1999

[CC99d] Y. Chahir, L. Chen, "Automatic video segmentation and indexing", Proc.of SPIE Conf. on Intelligent Robots and Computer Vision XVIII : Algorithms, Techniques, and Active Vision, ISBN 0-8194-3430-2, pp :345-357, 19-22 September 1999 Boston

[IEEE96] IEEE 1996. Special Section on Digital Libraries: Representation and Retrieval, IEEE Trans. On Pattern Analysis and Machine Intelligence, 18 (8)

[Jol98] Jean-Michel Jolion, Indexation d'images: nouvelle problématique ou vieux débat?, Rapport de Recherche 9805, Laboratoire Reconnaissance de Formes et Vision, INSA, Lyon, 1998

[FM95] B. Furht, M. Milenkovic, "A Guided Tour of Multimedia Systems and Applications", IEEE Computer Society Press, 1995

[Nas97] Chahab Nastar. Indexation d'Images par le Contenu: un Etat de l'Art, journées CNET CORESA, 1997

[Jai97] Ramesh Jain. Guest Editor, Special issue on visual information management communications of ACM, December 1997.

[Nar95] A.D.Narasimhalu, Guest Editor, Special Issue on Content-Based Retrieval, ACM Multimedia Systems, Vol. 3(1), Feb. 1995

[PP96] A. Pentland and R. Picard. Special issue on digital librarie. IEEE Trans. Pattern. Recognition and Machine Intelligence, 1996

[SC96] Bruce Schatz and Hsinchun Chen. Building Large-scale digital libraries, Computer 1996

[Chen98] Liming Chen, Une contribution pour les accès intelligents aux documents multimédias dans les systèmes d'information globaux, Rapport scientifique pour l'obtention d'une Habilitation à Diriger des Recherches, UTC-Compiègne, 8 Janvier, 1998

[MPEG7] MPEG7: Context and Objectives (v.3), ISO/IEC JTC1/SC29/WG11 N1678, http://www.cselt.stet.it/mpeg/mpeg_7.htm.

[Fli+95] Myron Flicker et al. . Query by image and video content: the QBIC system, Computer vol 28, number 9, September 1995

[OS95] VE. Ogle and M. Stonebraker. Chabot, Retrieval from a Relational Database of Images, Special issue on Content-Based Image Retrieval Systems, Computer, Vol. 28(9), Sept. 1995, pp.40-48

[EQVW] http://www.excalibur.com,http://wwwqbic.amaden.ibm.com, http://www.virage.com/ http://disney.ctr.columbia.edu/WebSEEk/

[IP 97] F. Idris and S. Panchanathan (1997), Review of image and video indexing techniques, Journal of Visual Communication and Image Representations, 8(2), 146-166

[Che+95] L.Chen, P.Faudemay, D.Donsez, L.Sonké, P.Maillé, C.Ragot, « TransDoc : Centre de Documentation Multimédia Sans Mur », projet n°617 labbelisé d'expérimentation d'intérêt publique par le Minsitère de l'Industrie dans son appel à propositions « Autoroutes de l'Information », 1995

[BS95] Chris Buckley and Gerard Salton. Optimization of relevance feedback weights. In Proc. of SIGIR'95, 1995.

[CCH92] James P. Callan, W. Bruce Croft, and Stephen M Harding. The inquery retrieval system. In proc of 3rd Int Conf on Database and Expert System Application, Sept 1992.

[SM83] G. Salton and Michael.J.MacGill, Introduction to Modern Information Retrieval, Mac Graw Hill Book Company, New York, 1983.

[Sha95] W.M. Shaw. Term-relevance computations and perfect retrieval performance. Information Processing and Management. 1995

[All95] James Allan. Relevance feedback with too much data. In Proc of SIGIR'95, 1995.

[Meh+97a] S. Mehrotra, Y. Rui, M. Ortega-B, and T.S. Huang. Supporting content-based queries over images in MARS. In Proc. of IEEE Int. Conf. on Multimedia Computing and Systems, 1997

[RHM97a] Y. Rui, T. S. Huang, and S. Mehrotra. Content-based image retrieval with relevance feedback in MARS. In Proc. IEEE Int. Conf. on Image Proc., 1997

[Rui+98] Y. Rui, T.S. Huang, S. Mehrotra, and M. Ortega. Relevance feedback: A power tool in interactive content-based image retrieval. IEEE Tran on Circuits and Systems for Video Technology, Special Issue on Interactive Multimedia Systems for the Internet, Sept 1998;

[Ull82] J. Ullman, Principles of databases systems, Computer science press, London (UK), 2nd edition, 1982

[Chr94] V. Christophides, Recherche documentaire par structure: une approche comparative entre SRI et SGBD, dans Le Traitement électronique du document, ADBS Editions, Paris, 1994

[Sma98] Malika Smaïl, Vers des systèmes évolutifs de recherche d'information: un état de l'art, Techniques et Sciences informatiques, Volume 17 - n° 10/1998, pages 1193 à 1222

[RJB89] V.V. Raghavan, G.S. Jung and P. Bollmann (1989). A Critical Investigation of Recall and Precision as Measures of Retrieval System Perfomance. ACM TOIS, 7(3), 205-229

[Sal86] G. Salton (1986). Another Look at Automatic Text-Retrieval Systems. Communications ACM, 29(7), 648-656

[SYW75] G. Salton, C.S. Yang, and A. Wong, A vector space model for automatic indexing, Comm.of the ACM, 18(11):613-620, november 1975

[SB87] G. Salton and C. Buckley, Text weighting appraches in automatic text retrieval, Technical Report 87-881, Cornell University, 1987

[BS75] A. Bookstein and D.R. Swanson, A decision theoretic foundation for indexing, Journal of the american society for information science, 28: 45-50, 1975

[CM78] W.S. Cooper and M.E. Maron, Foundation of probabilistic and utility theoretic indexing, Journal of the ACM, 25(1): 67-80, 1978

[SB88] G. Salton, C. Buckley (1988). Term Weighting approaches in automatic text retrieval. Information Processing and Management, 24(5), 513-523

[Far+96] N. Faraj, R. Godin, R. Missaoui, S. David et P. Plante (1996). Analyse d'une méthode d'indexation automatique basée sur une analyse syntaxique de texte, Revue de l'information et de la bibliothéconomie, 21(1), 1-21

[WWY92] Z. Wand, S. Wong, and Y. Yao, An analysis of vector space models based on computational geometry , in ACM SIGIR International Conference on Research and Development in Information Retrieval, Copenhagen, 1992, pp. 152-160

[Sal89] G. Salton, Automatic Text Processing, Addison-Weslay, 1989

[Sma98] Malika Smaïl, Vers des systèmes évolutifs de recherche d'information: un état de l'art, Techniques et Sciences informatiques, Volume 17 - n° 10/1998, pages 1193 à 1222

[CH89] M. Créhange, and G. Halin, Machine Learning Techniques for Progressive Retrieval in an Image Database, in Proceedings Datenbanksysteme in Buro, Technik und Wissenschaft, T. Harder, ed.

Zurich, march 1989, Springer-Verlag, pp. 314-322

[Hal89] G. Halin, Apprentissage pour la recherche interactive et progressive d'images,: processus EXPRIM et prototype RIVAGE, Thèse de Doctorat de l'Université de Nancy I, Octobre 1989

[SG83] G. Salton and M.M. Gill, Introduction to Modern Information Retrieval, Mac Graw Hill Book Company, New York, 1983

[SQ96] Alan F. Smeaton and Ian Qigley. Experiments on Using Semantic Distances Between Words in Image Caption Retrieval. In Proceedings of ACM SIGIR Conference, 1996.

[ATY+97] Aslandogan, Y.A.lp, Thier, Charles, Yu, T.Clement, Zou, Jun, and Rishe, Naphtali. Using Semantic Contents and WordNet TM in Image Retrieval. In Proceedings of ACM SIGIR Cnoference, 1997.

[Voo93] Ellen M. Voorhees. Using WordNet to Disambiguate Word senses for Text Retrieval. In proceedings of ACM SIGIR Conference, pages 171-180, 1993.

[BA95] G. Baxter et D. Anderson. 1995. Image indexing and retrieval : some problems and proposed solutions. New Library World 96, no 1123 : 4-13.

[Sel90] Seloff, G.A. 1990. Automated access to the NASA-JSC image archives. Library Trends 38, no 4 : 682-696.

[Leu+92] Leung, C.H.C. et al. 1992. Picture retrieval by content description. Journal of Information Science 18, no 2 : 111-119.

[Bor+90] Bordogna et al. 1990. Pictorial indexing for an integrated pictorial and textual environment. Journal of Information Science 16, no 3 : 165-173.

[CW92] Chang, C.-C. et Wu, T.-C. 1992. Retrieving the most similar symbolic pictures from pictorial databases. Information Processing and Management 28, no 5 : 581-588.

[Kra88] Krauze, Michael G. 1988. Intellectual problems of indexing picture collections. Audiovisual Librarian 14, no 4 (November) : 73-81.

[Tur94] Turner, James M..1994. Indexing " Ordinary " Pictures for Storage and Retrieval. Visual Resources X : 265-273.

[Sha94] Shatford, Sara 1994. Some Issues in the Indexing of Images. Journal of the American Society for Information Science 45, no 8 : 538-588.

[Rod91] Roddy, Kevin. 1991. Subject access to visual resources : what the 90s might portend. Library Hi Tech 9 : 1 (no 33) : 45-49.

[Mur84] Murry, Rita. 1984. Criteria for Subject Indexing of Pictures : an Introductory Survey. Master diss., University of Alberta.

[Sve94] Svenonius, Elaine. 1994. Access to Nonbook Materials : The Limits of Subject Indexing forVisual and Aural Languages. Journal of the American Society for Information Science 45, no 8 : 600-606.

[Ohl82] Ohlgren, Thomas H. 1982. Image analysis and indexing in North America : a survey. Art Libraries Journal (summer) : 51-60.

[HCK90] Halin, G., Crehange, M., and Kerekes, P.Machine Learning and Vectorial Matching for an Image Retrieval Model: EXPRIM and the system RIVAGE. In Proceedings of ACM-SIGIR 1990, Brussels, Belgium, pages 99-114, 1990

[JG95] Jung, G.S. and Gudivada, V. Adaptive Query reformulation in Attribute based Image Retrieval. In Intelligent Systems, pages 763-774, 1995.

[ATY+95] Aslandogan, Y.A., Thier, C., Yu, T.C., Liu, C., and Nair K. Design, implementation and evaluation of SCORE (a system for Content based Retrieval of Pictures). In proceedings of IEEE ICDE'95, pages 280-287, March1995.

[MMD76] C.S. McCamy, H. Marcus, and J.G. Davidson. A color-rendition chart, Journal of Applied Photographic Engineering, 2(3), Summer 1976

[Miy88] Makoto Miyahara, Mathematical transform of (r,g,b) color data to munsell (h,s,v) color data, In SPIE Visual Communications and Image Processing, Vol. 1001, 1988

[WYA97] Jia Wang, Wen-Jann Yang, and Raj Acharya. Color clustering techniques for color-content-based image retrieval from image databases. In Proc. IEEE Conf. on Multimedia Computing and Systems, 1997

[WYS82] G. Wyszecki and W. S. Stiles. Color science : concepts and methods, quantitative data and formulae. The Wiley series in pure and applied optics. John Wiley & Sons, Inc., New York, NY, 2nd ed. edition, 1982..

[SB91] M. J. Swain and D. H. Ballard. Color indexing. International Journal of Computer Vision, 7:1 1991.

[BB82] D. H. Ballard and C. M. Brown. Computer Vision. Prentice-Hall, Inc, Englewood Cliffs, NJ, 1982.

[Nib+93] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, and C. Faloutsos. The QBIC project: Querying images by content using color, texture, and shape. IBM RJ 9203 (81511), February 1993.

[MY88] M. Miyahara and Y. Yoshida. Mathematical transform of (r, g, b) color data to Munsell (h, v, c) color data. In SPIE Visual Communications and Image Processing 1988, vol. 1001 1988.

[VK95] A. Vellaikal and C.-C. J. Kuo. Content-based image retrieval using multiresolution histogram representation. In C.-C. J. Kuo, editor,Digital Image Storage and Archiving Systems, volume 2606 of SPIE Proceedings, pages 312--323, Philadelphia, PA, USA, October 1995.

[Gra95] R. S. Gray. Content-based image retrieval: Color and edges. Technical report, Dartmouth University Department of Computer Science technical report #95-252, 1995.

[Iok89] Mikihiro Ioka. A method of defining the similarity of images on the

basis of color information. Technical Report RT-0030, IBM Research, Tokyo Research Laboratory, Nov. 89

[Nib+94] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, and C. Faloutsos. The QBIC project: Querying images by content using color, texture, and shape. In Proc. SPIE Storage and Retrieval for image and Video Databases, February 1994.

[SO95] Markus Stricker and Marcus Orengo. Similarity of color images. In Proc. SPIE Storage and Retrieval for Image and Video Databases , 1995

[Meh+95] B.M. Mehtre, M. S. Kankanhalli, A.D. Narsimhalu, and G.C. Man. Color matching for image retrieval, Pattern Recognition Letters, Vol.16, pp. 325-331, March 1995

[Fal+93] C. Gfaloutsos et al. Efficient and effective querying by image content. Technical report, IBM Research Report, 1993

[CTO97] T.S Chua, K-L Tan and B-C Ooi. Fast signature-based color-spatial image retrieval. In Proc. IEEE Conf. on Multimedia Computing and Systems, 1997

[LOT94] H Lu, B. Ooi, and K. Tan. Efficient image retrieval by color contents. In Proc. of the 1994 Int. Conf. on Applications of Databases, 1994

[HCP95] Wynne Hsu, T.S.Chua, and H.K.Pung. An Integrated Color-Spatial Approach to Content-Based Image Retrieval. In Proceedings of ACM Multimedia Conference, pages 305-313,1995.

[SC96b] John R. Smith and Shih-fu Chang. VisualSEEk: a fully automated content-based image query system. In proceedings of ACM Multimedia Conference, Boston MA, pages 87-98, 1996.

[SC95a] John R. Smith and Shih-Fu Chang. Single color extraction and image query. In Proc. IEEE Int. Conf. on Image Proc., 1995

[SC95b] John R. Smith and Shih-Fu Chang. Tools and techniques for color image retrieval. In IS &T/SPIE proceedings Vol. 2670, Storage and Retrieval for Image and Video Databases IV, 1995

[WK96a] Xia Wan and C.-C. Jay Kuo. Image Retrieval Based on JPEG Compressed Data. Proc. SPIE Multimedia Storage and Archiving Systems, Volume 2916, pp.104-115, Nov. 1996

[PM93] William B. Pennebaker and Joan L. Mitchell, JPEG Still Image Data Compression Standard. New York: Van Nostrand Reinhold, 1993

[GP] M. Gervautz and W. Purgtathofer, A simple method for colour quantization , New Trends in Computer Graphics (Magnenat-Thalmann and Thalmann, eds.), pp. 219-231, Springer-Verlag, 1988

[RS96] R. Rickman and J. Stonham. Content-based image retrieval using color tuple histograms. In Proc. SPIE Storage and Retrieval for Image and Video Databases, 1996

[SD96] M. Stricker and A. Dimai. Color indexing with weak spatial constraints. In Proc. SPIE Storage and Retrieval for Image and Video Databases 1996

[Hua+97] J. Huang et al. Image indexing usong color correlogram. In Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, 1997

[WK97] Xia Wan and C.-C.Jay Kuo. Pruned Octree Feature for Interactive Retrieval. Proc. of SPIE Multimedia Storage and Archiving Systems II, Vol. 3229,pp.182-193, Dallas, 1997

[BBK98] Stephan Berchtold, Christian Bohm, and Hans-Peter Kriegel. The Pyramid-Technique: Towards Breaking teh Curse of Dimensionality. In proceedings of ACM SIGMOD, pages 142-153,1998.

[APV99] D. Androutsos, K.N. Plataniotis and A.N. Venetsanopoulos. A perceptual motivated technique to query-by-example using color cadinality. In Proc. of SPIE Multimedia Storage and Archiving Systems IV. September 1999, Vol. 3846, pp.137-145

[SC96c] John R. Smith and Shih-Fu Chang. Automated binary texture feature sets for image retrieval. In Proc. IEEE Int. Conf. Acoust, Speech and Signal Proc. May 1996

[TT90] Tomika, F. and Tsuji Saburo. Computer Analysis of Visual Textures. Kluwer Academic Publishers, 1990.

[HSD73] Robert M. Haralick, K. Shanmugam, and Its'hak Dinstein. Texture features for image classification. IEEE Trans. On Syst. Man. And Cyb. SMC-3(6), 1973

[GK90] C.C. Gotlieb and H.E.Kreyszig. Texture descriptors based on co-occurrence matrices. Computer Vision, Graphics, and Image Processing, 51, 1990

[TMY78] H. Tamura, S. Mori, and T. Yamawaki. Texture features corresponding to visual perception. IEEE Trans. on Sys. Man. and Cyb., SMC-8(6), 1978

[EN94] W. Equitz and W. Niblack. Retrieving images from database using texture-algorithms from the QBIC system. Technical Report RJ 9805, Computer Science, IBM Research Report, May 1994

[HMR96] T.S. Huang, S. Mehrotra, and K. Ranmchandran. Multimedia analysis and retrieval system (MARS) project. In Proc. of 33rd Annual Clinic on Library Application of Data Processing - Digital Image Access and Retrieval, 1996

[Ort+97] M. Ortega, Y. Rui, K. Chakrabarti, S. Mehrotra, and T. S. Huang. Supporting similarity queries in MARS. In Proc. of ACM Conf. on Multimedia, 1997

[SC94] J. R. Smith and S. Chang. Transform features for texture classification and discrimination in large image databases. In Proc. IEEE Int. Conf. on Image Proc., 1994

[CK93] Tianhorng Chang and C.-C Jay Kuo. Texture analysis and classification with tree-structured wavelet transform. IEEE Trans. Image Proc., 2(4):429-441, October 1993.

[LF93] A. Laine and J. Fan. Texture classification by wavelet packet signatures. In IEEE Trans. Patt. Recog. and Mach. Intell., 15(11):1186-1191, 1993

[Gro+94] M.H. Gross, R. Koch, L. Lippert, and A. Dreger. Multiscale image texture analysis in wavelet spaces. In Proc. IEEE. Conf. on Image Proc. 1994

[KC92] A. Kundu and J. Chen. Texture classification using qmf bank-based subband decomposition. CVGIP: Graphical Models and Image Processing, 54(5): 369-384, Sep. 1992

[TNP94] K.S. Thyagarajan, T. Nguyen, and C. Persons. A maximum likelihood approach to texture classification using wavelet transform. In Proc. IEEE Int. Conf. on Image Proc. 1994

[WDR76] J. Weszka, C.Dyer, and A. Rosenfeld. A comparative study of texture measures for terrain classification. IEEE Trans. On Sys. Man, and Xyb. SMC-6(4), 1976

[OD92] Philippe P. Ohanian and Richard C. Dubes. Performance evaluation for four classes of texture features. Pattern Recognition, 25(8): 819-833, 1992

[CJ83] G.C. Cross and A. K. Jain. Markov random field texture models. IEEE Trans. Patt. Recog. and Mach. Intell. 5:25-39, 1983

[Pen84] A.P. Pentland. Fractal-based description of natural scenes. IEEE Trans. Patt. Recog. and Mach. Intell. 6(6): 661-674, 1984

[MM95a] W.Y. Ma and B.S. Manjunath. A comparison of wavelet transform features for texture image annotation. In Proc. IEEE Int. Conf. on Image Proc., 1995

[HCA95] D. Hang, B. Cheng, and R. Acharya. Texture-based Image Retrieval Using Fractal Codes. Technical Report 95-19, Departement of Computer Science, State Unv. Of New York at Buffalo, 1995

[LHR97] Wei Li, Véronique Haese-Coat and Joseph Ronsin. Residues of morphological filtering by reconstruction for texture classification. Pattern Recognition, Vol. 30, No.7, 1081-1093, 1997

[JV95] Anil K. Jain et A. Vailaya, "Image Retrieval using Color and Shape", May 15, 1995

[FBF+94] Faloutsos C., Barber R., Flickner M., Hafner J., Niblack W., PetkovicD., and Equitz W. Efficient and Effective Querying by Image Content. Journal of Intelligent Information Systems, 3(1):231-262,1994.

[PPS94] A. Pentland, R. Picard, and S. Sclaroff. Photobook: Tools for content-based manipulation of image databases. In SPIE Paper 2185-05, Storage and Retrieval of Image and Video Databases II, San Jose, CA, pages 34-47, 1994.

SHB93 Sonka, M. and Hlavac, V. and Boyle, R. Image Proceeding, Analysis, and Machine Vision. Chapman and Hall, 1993.

[Wee96] A.R. Weeks. Fundamentals of electronic image processing. SPIE/IEEE Series on imaging science & engineering SPIE ISBN 0-8194-2149-9

[RSH96a] Y.Rui, A.C. She, and T.S. Huang . Modified fourier descriptors for shape representation - a practical approach. In Proc of First International Workshop on Image Databases and Multi Media Search, 1996

[ZR72] C.T. Zahn and R.Z. Roskies. Fourier descriptors for plane closed curves. IEEE Trans. On Computers, 1972

[PF77] E. Persoon and K.S. Fu. Shape discrimination using fourier descriptors. IEEE Trans. Sys. Man, Cyb. 1977

[Hu62] M.K. Hu. Visual pattern recognition by moment invariants, computer methods in image analysis. IRE Trans. On Informations Theory, 8, 1962

[YA94] L. Yang and F. Algregtsen. Fast computation of invariant geometric moments: A new method giving correct results. In Proc. IEEE Int. Conf. on Image Proc., 1994.

[KLS95] D. Kapur, Y.N.Lakshman, and T. Saxena. Computing invariants using elimination methods. In Proc. IEEE Int. Conf. on Image Proc. 1995

[CL95] D. Copper and Z. Lei. On representation and invariant recognition of complex objects based on patches and parts. Springer Lecture Notes in Computer Science series, 3D Object Representation for Computer Vision, pages 139-153, 199. M. Hebert, J. Ponce, T.Boult, A. Gross, editors

[LKC95] Z. Lei, D. Keren and D.B. Cooper. Computationally fast bayesian recognition of complex objetcs based on mutual algebraic invariants. In Proc. IEEE Int. Conf. on Image Proc, 1995

[PPS96] A. Pentland, R. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. International Journal of Computer Vision 1996.

[Ark+91] E.M. Arkin et al. An efficiently computable metric for comparing polygonal shapes. IEEE Trans. Patt. Recog. and Mach. Intell., 13(3), March 1991

[Bar+77] H.G. Barrow et al.. Parametric correspondence and chamfer matching: Two new techniques for image matching. In Proc rth Int. Joint Conf. Artificial Intel. 1977

[LM95] B. Li and S. De Ma. On the relation between region and contour representation. In Proc. IEEE Int. Conf. on Image Proc., 1995

[BKL97] Babu M. Mehtre, M. Kankanhalli, and Wing Foon Lee. Shape measures for content based image retrieval: A comparison. Information

Processing &Management, 33(3), 1997.

[WW80] I. Wallace and P. Wintz. An efficient three-dimensional aircraft recognition algorithm using normalized fourier descriptors. Computer Graphics and Image Processing, 13, 1980

[Tau91] G. Taubin. Recognition and positionning of rigid objects using algebraic moment invariants. In SPIE Vol. 1570 Geometric Methods in Computer Vision, 1991

[Sch 94] R. Schettini, Multicolored object recognition and location, Pattern Recognition Letters, vol.15, pp. 1089-1097, November 1994

[Cor+94] G. Cortelazzo et al. , Trademark shapes desscription by string-matching techniques, Pattern Recognition, vol. 27, no.8, pp. 1005-1018, 1994

[HJ94] P.W. Huang and Y.R. Jean, Using 2DC+-strings as spatial knowledge representation for image database systems, Pattern Recognition, vol. 27, no.9, pp.1249-1257, 1994

[HKR93] D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge, Comparing images using the Hausdorff distance, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, pp. 850-863, September 1993

[Kat+92] Kato, T., Kurita, T., Otsu, N., and Hirata, K. A Sketch Retrieval Method for Full Color Image Database, Query by Visual Example. In IEEE-IAPR-11, pages 530-533, August-September 1992.

[MM92] F. Mokhtarian and Alan K. Mackworth, A Theory of Multiscale, Curvature-Based Shape representation for Planar Curves, IEEE PAMI, Vol. 14, pp. 789-805, 1992

[Mok95] F. Mokhtarian, Silhouette-Based Isolated Object Recognition Through Curvature Scale Space, IEEE PAMI, Vol. 17, No 15, pp. 539-544, 1995

[MM98] S. Matusiak et M. Daoudi, Indexation Robuste et Recherche d'Images par le Dessin, Journées CENT, CORESA 98,

[Ale+95] A.D. Alexandrov, W.Y. Ma, A. El Abbadi, end B.S. Manjunath. Adaptive filtering and indexing for image databases. Storage and Retrieval for Image and Video Databases III, SPIE Vol. 2420, pp12-23

[ZZ95] HongJiang Zhang and Di Zhong. A scheme for visual feature based image indexing, Storage and Retrieval for Image and Video Databases III, SPIE Vol. 2420, pp36-45

[Vet84] M. Vetterli. Multi-dimensional sub-band coding: Some theory and algorithms. Signal Processing, pages 97 - 112, April 1984.

[Vet95] M. Vetterli and J. Kova?evi?. Wavelets and Subband Coding. Prentice-Hall, Inc, Englewood Cliffs, NJ, 1995.

[Ben75] J.L. Bentley. Multidimensional binary search trees used for associative searching. Communications of the ACM, 18(9):509-517

[Ben90] J.L. Bentley. K-d trees for semi dynamic points sets. In Proc. of the Sixth ACM Annual Symposium on Computational Geometry, pages 187-197

[Spr91] R. Sproull. Refinements to nearest-neighbor searching in k-dimensional objects. Algorithmica, 6(4): 579-589

[AM93] S. Arya and D.M. Mount. Algorithms for fast vector quantization. In Storer, J.A. and Cohn, M. editors, Proc. of DCC'93: Data Compression Conference, pages 381-390, IEEE press

[WJ96a] D.A. White and R. Jain. Similarity indexing: algorithms and performance. In Proc.of the SPIE: Storage and Retrieval for Image and Video Databases IV, San Jose, CA, Vol. 2670, pages 62-75.

[NHS84] J. Nievergelt, H. Hinterberger and K.C. Sevcik. The grid file: an adaptable, symmetric multikey file structure. Proc. of the ACM TODS, 9(1):38-71

[HN83] K. Hinrichs and J. Nievergelt. The grid file: a data structure to support proximity queries on spatial objetcs. Proc of the WG'83 (Intern. Workshop on Graph. Theoretic Concepts in Computer Science), pages 100-113, 1983

[Sam89] Hanan Samet. The Design and Analysis of Spatial Data Structures. Addison-Wesley, 1989.

[AS91] Walid G. Aref and Hanan Samet. Optimization stratégies for spatial query processing. Proc. of Very Large Data Bases, pages 81-90, September 1991

[PF95] E.G.M. Petrakis and C. Faloutsos. Similarity Searching in Large Image Databases. Technical Report 3388, Department of Computer Science, University of Mryland, 1995.

[Gut84] A. Guttman. R-trees: A dynamic index structure for spatial searching. In proceedings of ACM SIGMOD Conference, pages 47-57, June 1984.

[SRF87] T. Sellis, N. Roussopoulos, and C. Faloutsos. The R+tree: A dynamic index for multi-dimensional objects. In Proc 12th VLDB, 1987

[Gre89] Diane Green. An implementation and performance analysis of spatial data access. In Proc. ACM SIGMOD, 1989

[Bec+90] N. Beckmann, H-P. Kriegel, R. Schneider, and B. Seeger. The R*-tree: an efficient and robust access method for points and rectangles. In Proc. ACM SIGMOD, 1990

[KF94] I. Kamel and C. Faloutsos. Hilbert R-tree: AN Improved R-tree using Fractals. VLDB 1994: 500-509

[Rob81] J.T. Robinson. The K-D-B tree: A search structure for large multidimensional dynamic indexes. In Proceedings of the ACM SIGMOD International Conference on the Management of Data, pages 10-18

[ELS94] G. Evangedelis, D. Lomet and B. Salzberg. The hB^{II}-Tree: A concurrent and recoverable multi-attribute access method. Technical report NU-CCS-94-12, Northeastern University, College of CS , 1994

[LJF94] K-L. Lin, H. Jagadish, and C. Faloutsos. The TV-tree-An index structure for high-dimensional data. VLDB Journal, 3:517-542, 1994

[WJ96b] D.A. White and R. Jain. Similarity indexing with the SS-tree. IN Proc. 12th IEEE International Conference on Data Engineering, pages 516-523, New Orleans, 1996

[LS90] David B. Lomet and Betty Salzberg. The hb-tree: a multattribute indexing method with good guaranteed performance. ACM TODS, 15(4):625-658, December 1990

[Ber+96] S. Berchtold et al. The X-tree: AN index structure for high-dimensional data. VLDB'96

[NS96] Raymond Ng and Andishe Sedighian. Evaluating multi-dimensional indexing structures for images transformed by principal component analysis. In Proc. SPIE Storage and Retrieval for Image and Video Databases, 1996.

[Rui+97c] Y. Rui, K. Chakrabarti, S. Mehrotra, Y. Zhao, and T.S. Huang. Dynamic clustering for optimal retrieval in high dimensional multimedia databases. In TR-MARS-10-97, 1997.

[YVD95] Qi Yang, Asha Vellaikal and Son Dao. MB+-Tree: A New Index Structure For Multimedia Databases.1995, http://biron.usc.edu/~vellaika/papers/mmdbms95.ps

[Tag97] H. Tagare. Increasing retrieval efficiency by index tree adaptation. In Proc. of IEEE Workshop on Content-based Access of Image and Video Libraries, in conjunction with IEEE CVPR'97, 1997

[Char+97] Moses Charikar, Chandra Chekur, Thomas Feder, and Rajeev Motwani. Incremental clustering and dynamic information retrieval. In Proc. of the 29th Annual ACM Symposium on Theory of Computing, pages 626-635, 1997.

[JZL96] A. Jain, Y. Zhong, and S. Lakshmanan. Object Matching Using Deformable Templates. IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 408-439, March 1996.

[CWS95] S.F. Chang, J.Smith, and H.Wang. Automatic Feature Extraction and Indexing for Content-Based Visual Query. Technical Report CU/CTR 414-95-20, Columbia University, January 1995.

[Ege93] Max J.Egenhofer. What's Special About Spatial? Database Requirements for Vehicle Navigation in Geographic Space. In Proceedings of ACM SIGMOD, pages 398-402, 1993.

[CSY87] S.K.Chang, Q.Shi, and C.Yan. Iconic indexing by 2-d string. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1987.

[YM98] Clement T. Yu and Weiyi Meng. Principles of Database Query Processing for Advanced Applications. Data Management Systems. Morgan Kaufmann, 1998.

[HCT96] Chih-Cheng Hsu, Wesley W. Chu, and Rick K.Taira. A Knowledge-Based Approach for Retrieving Images by Content. IEEE-TKDE, 8:522-532, 1996.

[Li+97] Wen-Syan Li, K.S. Candan, Kyoji Hirata, and Yoshinori Hara. SEMCOG: an object-based image retrieval system and its visual query interface. In proceedings of ACM SIGMOD, pages 521-524, June 1997.

[Pap+95] Dimitris Papadias, Timos Sellis, Yannis Theodorakis, and Max J. Egenhofer. Topological Relations in the Worl of Minimum Bounding Rectangles: A Study with R-trees. In Proceedings of ACM SIGMOD, pages 92-103, 1995.

[SL2000] John R. Smith and Chung-Sheng Li, Image Classification and Querying using Composite Region Templates*, To appear in Journal of Computer Vision and Image Understanding - special issue on Content+Based Access of Image and Video Libraries

[Cox98a] I. J. Cox, M. L. Miller, Thomas P. Minka, and P. N. Yianilos. An optimized interaction strategy for bayesian relevance feedback; In IEEE Conf. CVPR, 1998.

[Cox98b] I. J. Cox, M. L. Miller, S. M. Omohundro, and P. N. Yianilos. Target testing and the pichunter bayesian multimedia retrieval system. In Advanced Digital Libraries Forum, Washington D.C., May.

[Pap+98] T.V. Papathomas, T.E. Conway, I.J. Cox, J. Ghosn, M.L. Miller, T.P. Minka, and P.N. Yianilos. Psychophysical studies of the performance of an image database retrieval system. In IS&T/SPIE Conf on Human Vision and Electronic Imageing III, 1998.

[CEM98] S.-F. Chang, A. Eleftheriadis, and Robert McClin . Next-generation content representation, creation and searching for new media applications in education. IEEE Proceedings, 1998.

[Sri95] R.K. Srihari. Automatic indexing and content-based retrieval of captioned images. IEEE Computers Magazine, 28(9), 1995

[SC97a] John R. Smith and Shih--Fu Chang. Multi-stage classification of images from features and related text. In 4th Europe EDLOS Workshop, San Miniato, Italy Aug 1997

[SC97b] John R. Smith and Shih--Fu Chang. Enhancing image search engines in visual information environments. In IEEE 1st Multimedia Signal Processing Workshop, June 1997

[PM96] R.W. Picard and T.P. Minka. Vision texture for annotation. Multimedia Systems: Special Issue on Content-based retrieval.1996

[MP96] T.P. Minka and R.W. Picard. Interactive learning using a "society of models". In Proc. IEEE CVPR, pages 447-452, 1996.

[Pic96a] R.W. Picard.Digital libraries: Meeting place for high-level and low-level vision. In Proc Asian Conf on Comp. Vis. 1996

[Pic95] R.W. Picard. Toward a visual thesaurus. In Springer Verlag Workshop in Computing, MIRO 95. 1995.

[Pic96b] R.W. Picard. Computer learning of subjectivity. In Proc. ACM Computing Surveys. 1996

[PMS96] R.W. Picard, T. P. Minka, and M. Szummer. Modeling user subjectivity in image libraries. In Proc. IEEE Int. Conf. on Image Proc. Lausanne, Sept 1996.

[Nar95b] A.D.Narasimhalu. Special section on content-based retrieval. Multimedia Systems. 1995.

[MM97] W.Y. Ma and B.S. Manjunath. Netra: A toolbox for navigating large image databases. In Proc. IEEE Int Conf. on Image Proc., 1997.

[Ale+95] A. D. Alexandrov, W.Y.Ma, A. El Abbadi, and B.S. Manjunath. Adaptive filtering and indexing for image databases. In Proc. SPIE Storage and Retrieval for Image and Video Databases, 1995.

[MM96a] W.Y. Ma and B.S. Manjunath. Texture features for browsing and retrieval of image data. IEEE T-PAMI special issue on Digital Libraries, Nov 1996.

[MM95b] B.S. Manjunath and W.Y. Ma. Image indexing using a texture dictionary. In Proceedings of SPIE Conference on Image Storage and Archiving System, volume 2606,.995

[MM96b] W.Y. Ma and B.S. Manjunath.Texture features and learning similarity. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pages 425-430, 1996.

[Meh+97b] S. Mehrotra, K. Chakrabarti, M. Ortega, Y. Rui, and T. S. Huang. Multimedia analysis and retrieval system. In Proc. of The 3rd Int. Workshop on Information Retrieval Systems, 1997.

[RSH96b] Y.Rui, A.C. She, and T.S. Huang. Automated shape segmentation using attraction-based grouping in spatial-color-texture space. In Proc. IEEE Int. Conf. on Image Proc.,1996.

[Rui+97a] Y. Rui, T.S. Huang, S. Mehrotra, and M. Ortega. Automatic matching tool selection using relevance feedback in MARS. In Proc. of 2nd Int. Conf. on Visual Information Systems, 1997.

[RHM97c] Yong Rui, Thomas S. Hua,g, and Sharad Mehrotra. Content-based image retrieval with relevance feedback in MARS. In Proc. IEEE Int. Conf. on Image Proc. 1997

[Rui+97b] Y. Rui, T.S. Huang, S. Mehrotra, and M. Ortega. A relevance feedback architecture in content-based mulitmedia information retrieval systems. In Proc. of IEEE Workshop on Content-based Access of Image and Video Libraries, in conjunction with IEEE CVPR'97, 1997.

[Rui+98] Y. Rui, T.S. Huang, S. Mehrotra, and M. Ortega. A relevance feedback: A power tool in interactive content-based image retrieval IEEE Tran. on.

Circuits and Systems for Video Technology, Special Issue on Interactive Multimedia Systems for the Internet, Sept 1998

[RHM97b] Y. Rui, T. S. Huang, and S. Mehrotra. MARS and its applications to MPEG-7. ISO / IE JTC1 / SC29 / wg11 M2290, MPEG97, july 1997.

[Nas+98a] C. Nastar, M. Mitschke, C. Meilhac, N. Boujemaa. Surfimage: a flexible content-based image retrieval system, ACM-Multimedia 1998

[Nas+98b] C. Nastar, N. Boujemaa., M. Mitschke, C. Meilhac, Surfimage: un système flexible d'indexation et de recherche d'images .CORESA 98, Lannion 9-10 juin 1998

[SC96a] J.R. Smith and S-F Chang. Intelligent Multimedia Information Retrieval, edited by Mark T. Maybury, chapter Querying by Color Regions Using the VisualSEEk Content-Based Visual Query System, 1996

[SC97c] J.R. Smith and S-F. Chang. Visually searching the web for content. IEEE Multimedia Magazine, 4(3):12-20, Summer 1997,

[SC96d] J.R. Smith and S-F.Chang. Local color and texture extraction and spatial query. In Proc. IEEE Int. Conf. on Image Proc., 1996

[SAF94] B. Scassellati, S. Alexopoulos, and M. Flickner. Retrieving images by 2nd shape: A comparison of computation methods with human perceptual judgments. In Proc. SPIE Storage and Retrieval for Image and Video Databases, 1994.

[Lee+94] D. Lee, R. Barber, W. Niblack, M. Flickner, J. Hafner, and D. Petkovic. Indexing for complex queries on a query-by-content image database. In Proc. IEEE Int. Conf. on Image Proc., 1994.

[Dan+93] D. Daneels, D. Campenhout , W. Niblack, W. Equitz, R. Barber, E. Bellon, and F. Fierens. Interactive outhning: An improved approach using active contours. In Proc. SPIE Storage and Retrieval for Image and Video Databases, 1993.

[Gup95] Gupta, A.Visual Information Retrieval Technology, A VIRAGE Perspective. White paper, Virage Inc.,1995.

[Bac+95] J.R. Bach, C. Fuller, A. Gupta, A. Hampapur, R. Jain, and C. Shue The Virage image search engien: An open framework for image management. In Proc. SPIE Storage and Retrieval for Image and Video Databases.1995

[GJ97] A. Gupta and R. Jain. Visual information retrieval. Communications of the ACM, 40(5), 1997

[Dow93] J. Dowe. Content-based retrieval in multimedia imaging. In Proc. SPIE Storage and Retrieval for Image and Video Databases, 1993.

[HK92] Kyoji Hirata and Toshikazu Kato. Query by visual example. In Proc of 3rd Int Conf on Extending Database Technology. 1992

[Hog+91] Hogan, M. et al. 1991. The visual thesaurus in a hypermedia environment : a preliminary exploration of conceptual issues and applications, International Conference on Hypermedia and Interactivity in Museums, Pittsburgh, October 1991, Archives and Museum Informatics, Pittsburgh, 1991 : 202-221.

[Bes90] Howard Besser 1990. Visual access to visual images : the UC Berkely Image Database Project. Library Trends 38, no 4 (Spring) : 787-798.

[Car+97] Chad Carson, serge Belongie, Hayit Greenspan, and Jitendra Malik. Region-based image querying. In Proc of IEEE Workshop on Content-based Access of Image and Video Librairies, in conjunction with IEEE CVPR'97, 1997.

[YW97] Hong Heather Yu and Wayne Wolf. Hierarchical, multi-resolution algorithms for dictionary-driven content-based image retrieval. In Proc. IEEE Int. Conf. on Image Proc., 1997.

[Mil95] Georges A. Miller. Wordnet: a lexical database for english, http://www.cogsci.princeton.edu/~wn/

[FSA96] C. Frankel, M.J. Swain, and V. Athitsos. Webseer: An image search engine for the world wide web. Technical Report TR-96-14, Computer Science Department, University of Chicago, 1996.

[ASF97] Vassilis Athitsos, Michael J. Swain, and Charles Frankel. Distinguishing photographs and graphics on the world wide web. In Proc IEEE Workshop on Content-Based Access of Image and Video Libraries, 1997.

[STL97] S. Sclaroff, L. Taycher, and M. La Cascia. Imagerover: A content based image browser for the world wide web. In Proc IEEE Workshop on Content-based Access of image and Video Libraries, 1997.

[SW95] D.L. Swets and J.J. Weng. Efficient content-based image retrieval using automatic feature selection. In Proc. IEEE 1995, 1995

[Gon+94] Y. Gong, H. Zhang, H.C. Chuan, and M. Sakauchi. An image database system with content capturing and fast image indexing abilities. In Proc IEEE 1994, 1994.

[Wu+95] J.K. W, A. Narasimhalu, B.M. Mehtre, C.P. Lam, and Y.J. Gao. Core: a content-based retrieval engine for multimedia information systems. Multimedia systems, 1995

[OAH96] Atsushi Ono, Masashi Amano, and Mituhiro Hakaridani. A flexible content-based image retrieval system with combined scene description keyword. In Proc. IEEE Conf. on Multimedia Computing and Systems, 1996.

[DFB97] Chabane Djeraba, Patrick Fargeaud, Henri Briand. Retrieval by Content in an Image Database. Coresa 97, Journées CENT, Paris, 26-27 Mars 1997

[Sma94] Malika Smaïl, Raisonnement à base de cas pour une recherche évolutive d'information, Prototype Cabri-n. Vers la définition d'un cadre

d’acquisition de connaissances, Thèse de doctorat, Université Henri Poincaré, Nancu I, 1994

[Als+96] P. Alshuth, Th. Hermes, J. Kreyb, and M. Roper. Video Retrieval with IRIS. In Proceedings of ACM Multimedia Conference, Boston MA, page 421, 1996

[GP97] F. Golshani and Y. Park.Content-based image indexing and retrieval in Image RoadMap, . In Proc. SPIE Multimedia Storage and Archiving Systems II, Vol. 3229, pp. 194-205, February 1997

[Coq+95] J-P. Cocquerez et al. Analyse d'images: filtrage et segmentation. Masson, ISBN: 2-225-84923-41995, pp.3-4

[Rus95] J. C. Russ. The Image Processing Handbook. CRC Press, Boca Raton, 1995.

[NH88] A. N. Netravali and B. G. Haskell. Digital Pictures; representation and compression. Applications of Communications Theory. Plenum Press, New York, NY, 1988.

[Kan96] H.R.Kang, "Color technology for electronic imaging devices", SPIE, ISBN 0-8194-2108-1, 1996

[Kun93] M. Kunt, Traitement de l'information: volume 2, Traitement numérique des images. Presses polytechniques et universitaires romandes, 1993

[Cel90] M. Celenk, A color clustering technique for image segmentation, Computer Vision, Graphics, and image processing, vol. 52, pp. 145-170, 1990

[Hun89] R. W. G. Hunt, Measuring Color. Ellis Horwood series in applied science and industrial technology. Halsted Press, New York, NY, 1989.

[WK96] X. Wan and C.-C. J. Kuo. Color analysis and quantization for image retrieval. Storage Retrieval Still Image Video Databases IV 2470, February 1996, 8-16

[Jul62] B. Julesz. Visual pattern recognition, IEEE Trans. on Information Theory, vol. 8, 1962

[Gag83] A. Gagalowicz. Vers un modèle de textures. Thèse d'Etat, Université Pierre et Marie Curie, Paris VI, 1983

[Phil88] S.Philipp. Analyse de texture appliquée aux radiographies industrielles. Thèse de l'Université P. et M. Curie, Paris VI, 1988

[Coc+95a] J.P. Cocquerez, et al. Analyse d'images: filtrage et segmentation, Masson, ISBN: 2-225-84923-41995 Chap II, p:14

[Dub99] B. Dubuisson. Vision et Image. Cours d'option SY23, Printemps 1999, UTC Compiègne

[Lev85] Martin D. Levine, Vision in man and machine,Mc Graw Hill eds., Chap. 10 Shape, pp.480-544, 1985

[Hu61] M.K. Hu, Pattern Recognition by Moments Invariants, Proc. of the IEEE, vol. 49, no. 9, September 1961, p. 1428

[Der91] Rachid Deriche, Fast algorithms for Low-level Vision, IEEE Transaction on Pattern Anal. Mach. Intell, (PAMI) N° 1, pp. 78-87, 1991

[SCH 94] R. Schettini, Multicolored object recognition and location, Pattern Recognition Letters, vol.15, pp. 1089-1097, November 1994

[WMB94] I. H. Witten, A. Moffat, and T. C. Bell. Managing Gigabytes: compressing and indexing documents and images. Van Nostrand Reinhold, New York, NY, 1994.

[SO95] M. Stricker and M. Orengo. Similarity of color images. In Storage and Retrieval for Image and Video Databases III, volume SPIE Vol. 2420, February 1995.

[DFB97] Chabane Djeraba, Patrick FARGEAUD and Henri BRIAND, Retrieval by Content in an Image Database, Journée CENT, Coresa97, Paris

[JV95] Anil K. Jain et A. Vailaya, "Image Retrieval using Color and Shape", May 15, 1995

[APV99] D. Androutsos, K.N. Plataniotis and A.N. Venetsanopoulos. A perceptual motivated technique to query-by-example using color cadinality. In Proc. of SPIE Multimedia Storage and Archiving Systems IV. September 1999, Vol. 3846, pp.137-145

[FL95] C. Faloutsos and King-Ip (david) Lin. Fastmap: Afast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In Proc. of SIGMOD, pages 163-174, 1995

[RJ93] Lawrence Rabiner and Biing-Hwang Juang. Fundamentals of Speech Recognition, chapter 5. Prentice hall, Englewood Cliffs, New jersey, 1993.

[DH73] R.O. Duda and P.E. Hart. Pattern Classification and Scene Analysis. Wiley, New York, 1973

[Fuk90] Keinosuke Fukunaga. Introduction to Statistical Pattern Recognition. Academic Press, 1990. 2nd Edition

[FD92] Peter W. Foltz and Susan T. Dumais. Personalized information delivery: an analysis of information filtering methods. Comm. Of ACM (CACM), 35(12):51-60, December 1992

[Dum94] Susan Dumai. Latent semantic indexing (LSI) and TREC-2. In D. K. Harman, editor, The Second Text Retrieval Conference (TREC-2), pages 105-115, Gaithersburg, MD, March 1994. NIST. Special Publication 500-215

[Mar97] André Marion. Acquisition & visualisation des images, , pp. 32, ISBN: 2-212-08871-X, Eyrolles 1997

[Zuc76] S.W. Zucker. Region growing: Chilhood and adolescence. Computer Graphics and Image Processing, 5:382-399, 1976

[HP74] S.L Horowitz and T. Pavlidis. Picture segmentation by a directed split-and-merge procedure. In Proc. of the 2nd International Joint Conference on Pattern Recognition, pages 424-433, 1974

[Pav86] T. Pavlidis. A critical survey of image analysis methods. In Proc. of the Eighth International Conference on Pattern Recognition, pages 502-511, October 1986

[HM93] Radu Horaud et Olivier Monga. Vision par ordinateur. Hermes Chapitre 4, Segmentations d'images en régions, pages 103-129, ISBN 2-86601-370-0

[HS79] G.M. Hunter and K. Steiglitz. Linear transformation of pictures represented by quadtrees. Computer Graphics and Image Processing 10, (3), 289-296

[Sam82] H. Samet,. Neighbour finding techniques for images represented by quadtrees, CHIP, vol. 18, 298-303, 1982

[Mol96] Thierry Molinier. Décomposition en "Tétra-Arbres". Rapport de DEA, UTC de Compiègne, Juillet 1996

[Lau89] Robert Laurini. L'ingénierie des connaissances spatiales. Chap.2: Formalismes de modélisation. et Chap. 3: Modélisation conceptuelle géométrique. Hermes

[NP94] E. Nardelli and G. Poietti. A Hybrid Pointerless Representation of Quadtrees for Efficient Processing of Window Queries. IGIS 1994: 256-269, 1994

[CC97] Y. Chahir, L. Chen, "Peano key rediscovery for content based retrieval of images", Proc. of SPIE Conf. on Multimedia Storage and Archiving Systems, Vol. 3229, Dallas, USA, Nov. 1997, ISBN 0-8194-2662-8

[CC98] Y. Chahir, L. Chen, "Spatialized Visual Features-Based Image Retrieval", Proc. of the ISCA, 14 th International Conference on Computers and Their Applications, ISBN: 1-880843-27-7 , pp. 174-179, Cancun, Mexico ,April 7-9, 1998

[MJF96] B.Moon, H.V.Jagadish, C.Faloutsos, J.H.Saltz, « Analysis of the Clustering Properties of Hilbert Space-filling curve », IEEE Transaction on Knowledge and Data Engineering, March 1996

[BM91] C.Berrut, M.Mechkour, « Representation of images in databases : a preliminary study », Proc. of IFIP-WG2.6 2nd Conference on Visual Database Systems, Budapest, 1991

[Cos+92] Gennaro Costaglio et al., Representing and Retrieving Symbolic Pictures by Spatial Relations; Visual Database Systems, II 1992 IFIP

[LH92] S-Y Lee and F.–J.Hsu . Spatial reasoning and similarity retrieval of

images using 2D C String knowledge representation. Pattern Recognition, 25(3):305-318,1992

[Cha+87] S.K. Chang et al.. Iconic indexing by 2D Strings. IEEE Transaction on Pattern Analysis and Machine Intelligence, p(3):413-428, 1987

[Jun88] E. Jungert, Extended symbolic projection as a knowledge structure for image database systems, Fourth BPRA Conference on Pattern Recognition, 1988, pp.343-351, Springer-Verlag

[CJL89] S.Chang, E. Jungert, and Y. Li, Representation and retrieval of symbolic pictures using generalized 2-D strings, Proc. SPIE: Visual Commun. Image Process. IV 89, 1360-1372

[LH92] S-Y Lee and F.–J.Hsu . Spatial reasoning and similarity retrieval of images using 2D C String knowledge representation. Pattern Recognition, 25(3):305-318,1992

[Lee+92] S.Lee et al. Signature files as a spatial filter for iconic image database, J. Visual Lang. Comput. 92, 305-397

[CC99a] Y. Chahir, L. Chen, " Efficient Content-Based Image Retrieval based on color homogenous objects segmentation and their spatial relationship characterization ", IEEE Multimedia Systems'99, International Conference on Multimedia Computing and Systems ,June 7-11, 1999, Florence, ITALY

[CC99b] Y. Chahir and L. Chen , "Searching Images on the basis of color homogeneous objects and their spatial relationship", Papier accepté, à la revue "Journal of Visual Communication and Image Representation" , à apparaître fin 1999

[Hua97] P. Huang, "Indexing pictures by key objects for large-scale image databases", Pat. Rec., Vol.30, N. 7 p1229-1237, 1997

[GR95] V. Gudivada and V. Raghavan, "Design and evaluation of algorithms for image retrieval by spatial similarity", ACM Trans. In Inf. Syst., 13(1): 115-144, April 1995

[Zha+96] J. Zhao, Y. Shimazu, K. Ohta, R. Hayasaka, and Y. Marsushita, JPEG Codec Adaptive to Region Importance, ACM Multimedia'96, 92140, ISBN 0-201-92140-X, Nov 1996, pp. 209-218

[AM98] Mohamed AKHERAZ et Martial MUZARD, Elaboration d'une interface pour la gestion et l'analyse, rapport de TX en automne 1998

[JPP95] R. Jain, A. Pentland, and D. Petkovic. NSF-ARPA workshop on visual information management systems. Cambridge, MA, June 1995.

[Jon81] K.S. Jones. Information Retrieval Experiment. Butterworth and Co., 1981

[Rij81] C.J. van Rijsbergen, Retrieval Effectiveness, Information Retrieval Experiment in K.S.Jones, Ed), pp.32-43, Butterworths,Stoneham,MA,1981

[Chan94] D .Chandler, The Grammar of Television and Film , UWA, 1994. http:/ / www.aber.ac.uk / ~dgc / gramtv.html

[Aum88] M. Marie Aumont, L'analyse de films, 2é édition, Nathan, 1988

[AJL95] Ph.Aigrain, Ph.Joly, V.Longueville, « Medium Knowledge-based Macro-segementation of Video into Sequences », Proc.IJCAI Workshop on Intelligent Multimedia Information Retrieval, Ed.Mark Maybury, 1995

[Roh90] Eric Rohmer, "Conte de Printemps", L'Avant-Scène du Cinéma, 392, May 1990 (Le script du film en français)

[LG96] B. Lamiroy and P. Gros, Rapid object indexing and recognition using enhanced geometric haching. In Procedings of the 4th European Conference on Computer Vision Cambridge, England, volume 1, pages 59-70, Avril 1996

[Sch96] C. Schmid. Appariement d'images par invariants locaux de niveaux de gris. Thèse de doctorat, GRAVIR-IMAG-INRIA Rhône-Alpes, juillet 1996

[Gro+97] P. Gros, R. Mohr, M. Gelgon et P. Bouthemy. Indexation de vidéos par le contenu, Coresa 1997, Paris

[AJ94] P. Aigrain and P. Joly. The automatic real-time analysis of film editing and transition effects and its applications. Computer & Graphics, 18(1):93-103, 1994

[AJ94] Philippe Aigrain et Philippe Joly, The automatic real-time analysis of film editing and transition effects and its application, Computers & Graphics, Vol. 18, No. 1, 1994 pp 93-103

[SV95] François Salazar et Franck Valéro. Analyse automatique de documents vidéo, Rapport de Recherche, IRIT 95-28-R, Université Paul Sabatier, 1995

[XLI95] Wei Xiong, John Chung-Mong Lee and Man-Ching Ip, Net Comparison : a fast and effective method for classifying image sequencess, Storage and Retrieval for Image and Video Databases III, SPIE Vol. 2420, 1995, pp318-328

[UMY92] Hirota Ueda, Takaumi Miyatake et Satoshi Yoshizawa, IMPACT: An interactive natural motion-picture dedicated multimedia authoring system, INTERCHI '91, ACM, 1991, pp 343-350

[Aku+92] Akihito Akutsu, Yoshinobu Tonomura, Hideo Hashimoto and Yuji Ohba, Video indexing using motion vectors, Proc. of Visual Communication and Image Processing 1992, Boston, SPIE Proc. series Vol. 1818, pp 1522-1530

[KJ91] R. Kasturi and R. Jain, Dynamic vision, in Computer Vision: Principles, pp. 469-480, IEEE Comput. Soc., Los Alamitos, CA, 1991

[NT91] Akio Nagasaka et Yuzuru Tanaka, Automatic video indexing and full-video serach for object appearances, Proc. of the IFIP, Second Working conference on Visual Database Systems, Budapest, 1991, pp 113-127

[HJW95a] Arun Hampapur, ramesh jain and Terry E. Weymouth, Production Model Based Digital Video Segmentation, Multimedia Tools and Applications, Vol.1, 1995 pp 9-46

[CB95] J.M. Corridoni and A. Del Bimbo, Film Semantic Analysis, Proc. of ACAIP, Prague, 1995

[DK95] Sadashiva Devadiga, david A. Kosiba, Ullas Gargi, Scott Oswald et Rangachar Katsuri, A semiautomatic video database system, SPIE vol. 2420, pp 262-266, 1995

[YL95] B.L. Yeo and B. Liu. Rapid scene analysis on compressed video, IEEE Trans. on circuits and systems for video technology, vol. 5, 533-544, 1995

[She97] Bo Shen, HDH based compressed video cut detection; HPL-97-142 971204 External, http://www.hpl.hp.com/techreports/97/HPL-97-142.html

[Yeu+95] Minerva Yeung, Boon-Lock Yeo, Wayne Wolf and Bede Liu, Video browsing using clustering and scene transitions on compressed sequences, Proc. Miltimedia Computing and Networking, -6-8 February 1995, San Jose, USA SPIE, Vol. 2417, pp 389-398

[DAE95] Apostolos Dailianas, Robert Allen and Paul England, Comparison of automatic video segmentation algorithms, Proc. of SPIE Photonics West, 1995

[Gar+95] Ullas Gargi, Scott Oswald, David Kosiba, Sadashiva Devadiga and Rangachar Kasturi, Evaluation of video sequence indexing and hierarchical video indexing, SPIE Vol. 2420, pp 144-151

[Kim97] Hae-Kwang KIM, Détection automatique des mouvements de caméra et des régions de textes pour la structuration et l'indexation de documents audiovisuels, Thèse de doctorat, Université Paul Sabatier de Toulouse, 1997

[MJC95] Jianhao Meng, Yujen Juan, Shi-Fu Chang, Scene change detection in a MPEG Compressed Video Sequence, Proc. IS&T SPIE '95 Digital Video Compression: Algorithm and Technologies, San Jose, Vol. 2419, 1995, pp 14-25

[SP95] Ishwar K. Dethi and Nilesh Patel, A Statistical approach to scene change detection, SPIE Vol. 2420, pp. 329-336, 1995

[AHC93a] F. Arman, A. Hsu, and M-Y. Chiu, Image processing on compressed data for large video databases, Proc. SPIE: Storage Retrieval Image Video Databases 1993, pp. 267-272

[AHC93b] F. Arman, A. Hsu and M.-Y Chiu, Feature management for large video databases, ACM Multimedia 93, 1908, February 1993, p. 2-12

[Zha+94] H.J. Zhang, C.Y.Low, Y. Gong, S.W. Smoliar, and S.Y. Tan, Video Parsing compressed data, Proc, SPIE: Image Video Processing II 2182, 1994, 142-149

[Zha+95] H.J. Zhang, C.Y.Low, S.W. Smoliar, and S.Y. Tan, Video Parsing and browsing using compressed data, Multimedia Tools Applications 1, 1995, 89-111

[LZ95] H.-C. H. Liu and G.L. Zick, Scene decomposition of MPEG compressed video, Digital Video Compression: Algorithms Tech. 2419, February 1995, 26-37

[ZMM95] Ramin Zabih, Justin Miller and Kevin Mai, The feature-based algorithm for detcting and classifying scene breaks, ACM Multimedia'95, San Fransisco, November 1995, pp 189-200

[BG96] P. Bouthemy and F. Ganansia. Video partitioning and camera motion characterization for content-based video indexing. In Proc. of 3rd IEEE Int. Conf. on Image Processing, volume I, pages 905-909, September 1996

[MP95] S. Mann and R.W. Picard, Video orbits: characterizing the coordinate transformation between two images using the projective group, MIT Media Lab. Technical report No. 278, 1995

[Ard+96] Mohsen Ardebilian, Xiao Wei Tu, Liming Chen and Pascal Faudemay, Video Segmentation Using 3-D hints contained in 2-D images, Proc. SPIE, Multimedia Storage and Archiving Systems, Vol. 2916, pp. 236-242, 1996

[Gün97] B. Günsel et al. "Hierarchical Temporal Video Segmentation and Content Characterization", Proc. Of SPIE, Multimedia Storage and Archiving Systems II, pp.46-56, Dallas 1997

[BR96] Jphn S. Boreczky and Lawrence A. Rowe, Comparison of video shot boundary detection techniques, Storage and retrieval for Image and Video Databases IV, Proc. of IS&T/SPIE 1996 Int'l Stymp. On Elec. Imaging: Science and technology, 1996

[Jol96] Philippe Joly, Consultation et Analyse des documents en image animée numérique, Thèse de Doctorat, Université Paul Sabatier, 1996

[CB94] Mourad Cherfaoui and Christian Bertin, Two-stage strategy for indexing and presenting video, Proc. SPIE'94 Storage and Retrieval for Video Databases, San Jose, 1994

[Wol95] Wayne Wolf, Key Frame Selection by Motion Analysis, ICASSP'95

[MG95] Nabil Madrane and Morris Goldberg, Video representation tools using a unified object and perspective based approach, SPIE Vol. 2420, pp.

152-163, 1995

[Zha+97] Hong Jiang Zhang, Jianhua Wu, Di Zhong and Stephen W. Smoliar, An integrated system for content-based video retrieval and browsing, Pattern Recognition, Vol. 30, No. 4, pp. 643-658, 1997

[Ben+98] Serge Benayoun et al. Structuration de vidéos pour des interfaces de consultation avancées, Coresa 1998, Lannion

[HBB96] H. Haddad, C. Berrut, M.F. Bruandet, Un modèle vectoriel de recherche d'informations adapté aux documentd vidéo, CORESA' 96, Grenoble, pp. 281-289, 1996

[CFH98] Liming Chen, Dominique Fontaine et Riad Hammoud. La segmentation sémantique de la vidéo basée sur les indices spatio-temporels. Coresa 1998, Lannion

[All83] J.F. Allen. Maintainning Knowledge about temporal intervals, CACM 1983, Vol. 26, pp. 832-843

[Ham97] R. hammoud. La segmentation de la vidéo en scènes basée sur les indices spatio temporels, Rapport de DEA Contrôle De Systèmes (CDS), Université de Technologie de Compiègne, Septembre 1997

[CC99c] Y. Chahir, L. Chen, "Automatic video segmentation and indexing", Proc.of SPIE Conf. on Intelligent Robots and Computer Vision XVIII : Algorithms, Techniques, and Active Vision, ISBN 0-8194-3430-2, pp :345-357, 19-22 September 1999 Boston

[SO95] M. Stricker and M. M. Orengo, "Similarity of color images", Proc. IS and SPIE. Storage and Retrieval for Image and Video Databases III, San Jose ,1995

[Cour97] Jonathan D. Courteney, Automatic video indexing via object motion analysis, Pattern Recognition, Vol. 30, No. 4, pp. 607-625, 1997, Special issue: image databases, Guest Editors, John C.M. Lee and Anil Jain