REPUBLIQUE ALGERIENNE DEMOCRATINE ET POPULAIRE
MINSITERE DE L'ENSEIGNEMENT ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MENTOURI CONSTANTINE
FACULTE DES SCIENCES DE L'INGENIEUR
DEPARTEMENT INFORMATIQUE

N :

Thèse De Doctorat es Sciences en Informatique
Spécialité : Informatique

Intitulé :

RECHERCHE D'IMAGES PAR LE CONTENU

Par :
Kamel Houari

Sous la direction de :
Dr. Kholladi Mohamed-Khireddine

Devant le Jury composé de :

Président: Benmohammed Mohamed, Professeur, (Université de Constantine)
Rapporteur: Kholladi Med Khireddine, Maître de Conférences, (Université de Constantine)
Examinateurs: Kazar Okba, Maître de Conférences, (Université de Biskra)
Billami Azzedine, Maître de Conférences, (Université de Batna)
Chikhi Salim, Maître de Conférences, (Université de Constantine)

Juin 2010

Remerciements

Je tiens à remercier tout particulièrement les personnes ayant permis l’aboutissement de mes travaux de thèse :

Tout d’abord, je tiens à remercier le. Prof. Benmohammed Mohamed pour avoir accepter de présider ma soutenance.

Je remercie vivement Dr. Kazar Okba pour avoir accepté de juger mes travaux.

Je remercie vivement Dr. Billami Azzedine pour avoir accepté de juger mes travaux.

Je remercie vivement Dr. Chikhi Salim pour avoir accepté de juger mes travaux.

Je tiens à exprimer ici ma gratitude au Dr Kholladi Mohamed-Khireddine pour tout : son investissement à mon égard, la liberté qu’il m’a laissée dans mon travail, sa grande disponibilité, sa rapidité, son aide inconditionnelle et ce, quel que soit le domaine (scientifique ou non). Bref, en un mot: merci Kholladi Mohamed-Khireddine !

Je remercie vivement Dr. Youcef Chahir, qui m’a accueilli au sein de son laboratoire GREYC. Mes semaines là-bas furent parmi les meilleures et certainement les plus enrichissantes, il a été toujours disponible lorsque je le sollicitais. Son expérience, son recul se sont révélés particulièrement utiles pour se concentrer sur l’essentiel, pour faire le tri entre les détails et les aspects réellement importants.

Je remercie également Prof. M.C Baatouche, pour sa confiance, sa disponibilité et son aide si précieuse.

Merci à toutes les personnes qui m’ont aidé et entouré durant ma thèse de près ou de loin.

Je réserve le dernier merci à mes parents et à ma famille, le plus gros. Merci pour tout le soutien, la confiance. MERCI !

Dédicace.

Je dédie du fond du cœur cette thèse à ma mère et à mon père pour tous ce qu’ils ont fait pour moi, et je la dédie aussi à ma chère épouse, qui a partagé avec moi en tous sens ces précieuses années, je la remercie beaucoup très sincèrement pour ses encouragements, ses conseils, son écoute, sa patiente et son soutien, ainsi que pour toute sa compréhension et sa grande compagnie.

Résumé

Cette thèse porte sur la **recherche d'images** par le **contenu** à l'issu de laquelle une chaine complète de recherche d'images par le contenu a été élaboré puis **automatisée**. Ceci nous a permis de cerner puis ensuite de définir les contours du champ d'intervention pour apporter notre contribution à la problématique de la recherche d'images par le contenu.

Notre approche est **globale**: elle se base sur le **contenu global de l'image**. Nous avons considéré tous les **meilleurs éléments de description** d'une image par son **contenu** et **particulièrement les points d'intérêts**. Vu la **taille importante** du **descripteur** d'image ainsi que le **volume des bases d'images**, nous nous sommes penchés sur les méthodes qui nous permettent **de réduire la taille du champ de recherche et d'indexation**.

Nous avons mis au point une méthode basée sur la **classification spectrale** aprés representaion de la **base d'images** par un **graphe convexe**, et pour réduire le **temps de réponse** de notre syteme qui represente l'enjeu majeur de tous système de recherche d'images. Nous avons appliqué la **théorie de nystrom** qui permet de ne pas **recalculer** toutes les **valeurs propres** mais seulement la dernière, ce qui permet de **réduire considérablement le temps de recherche** et par conséquent le **temps de calcul.**

Chaque partie du système a été **testée** et **évaluée** sur des **bases d'images hétérogènes**. Le résultat en termes de **rappel-précision et de temps de calculs** sont jugés **satisfaisants** comparé aux méthodes classiques qui utilisent soit un descripteur donné ou bien des descripteurs hybrides.

Mots-Clés : Recherche d'images par le contenu, descripteurs d'images, primitives, texture, couleur, formes, points d'intérêts.

Abstract

These thesis concerns the content based image retrieval witch a the end it result on elaboration of a complete chains and then automatized of content based image retrieval, This enabled us to determine and define contours of the field of intervention to contribute our share to the problems of image retrieval by contents. Our approach is total i.e. it is based on the total contents of the image.

Our approach is global; it is based on the total contents of the image. We consider all the best elements of description of an image by its contents and particularly the points of interests Considering the significant size of the descriptor of image as well as the volume of the bases of images, we are leaning on the methods which enable us to reduce the size of field of research and indexing**.**

We developed a method based on spectral classification after representing the base of images by a convex-graph, and to reduce the response time of our syteme which represente the major stake of all system of image retrieval, we applied the theory of nystrom which makes it possible not to recompute all the eigenvalues but only the last one.

Each part of the system was tested and evaluated over heterogeneous bases of images. The result in terms of recall-precision and computing time are considered to be satisfactory compared with the traditional methods which uses a given descriptor or many hybrid descriptors.

Keywords: Content-Bases-Images Retrieval, feature-extraction**,** primitives, texture, color, shape, interest points.

Sommaire

Remerciements.
Dédicaces.
Résumé
Abstract
Chapitre 1. ……...12
1. Introduction générale. …….....12
1.1 Motivations……..12
1.3 Objectifs…….....12
1.3 Description de la thèse…….……13
Chapitre 2……16
2. Descripteurs d'images……16
2.1 Introduction……..16
2.2 Descripteurs de la couleur……...16
2.2.1 L'espace de Couleur……17
(a) Système RGB…….……18
(b) Système normalisé (r, g, b)……18
(c) Système XYZ……..19
(d) Système L*u*v ……..……20
(e) Système $L^*a^*b^*$ ……...……..21
(f) Système HSV …….……22
(g) Système YCrCb……..……23
Espaces couleurs : récapitulatif ……24
2.2.2 Histogrammes……..25
2.2.3 Les moments statistiques_…….....27
2.2.4 Cohérence spatiale…….....27
2.2.5 Couleurs dominantes…….....28
2.2.6 Corrélogrammes…….....30
2.2.7 Résumé ……32
2.3 Descripteurs de textures……..……32
1. Schéma général……..……34
2. La fenêtre d'analyse ……34
2.3.1 Matrices de co-occurrences……35
2.3.1.1 Algorithme: Matrice de cooccurrences…….....36
2.3.2 Descripteurs texturaux…….……37
2.3.3 Filtres de Gabor…….....39
2.3.4 Transformée en ondelettes……..….40
2.4 Les descripteurs des points d'intérêts……..41
2.4.1 Quelques méthodes de détection……..……43
2.4.2 Le détecteur de Moravec (1980)…….....43
2.4.3 Le détecteur de Harris (1988)……44
2.4.4 Extraction de caractéristiques des points d'intérêt par SIFT……..45
. Points SIFT…….…....46
. Implémentation……..49
2.5. Descripteurs de formes……...52
2.5.1 Les descripteurs géométriques de région…….…...53
2.5.2 Les moments géométriques……53
2.5.3 Transformée de Hough ……..55

. Transformation de m à 1 ……56
. Transformation de 1 à m ……...56
Algorithme: Transformée de Hough Standard……..56
2.5.4 Résumé……..…..…57
Chapitre 3…….....58
3. Les Systèmes de Recherche d'Images par le Contenu
3.1 Introduction ……58
3.2 Composants d'un CBIR……...59
(1) La base d'image……...60
(2) L'indexation……60
(3) La gestion desindex……61
(4) Les requêtes…….……....61
(5) Analyse de la requête……..61
(6) Mise en correspondance requête / base……...61
(7) La présentation des résultats……...62
3.3 Représentation des images dans un CBIR……..……62
3.3.1 Descripteurs de bas niveau……..62
3.3.2 Représentation par les couleurs……64
3.3.3 Représentation par les textures……....65
Chapitre 4……68
4. Mesure de similarité entre descripteurs…….....68
. Distance de Mahalanobis…….....68
. Intersection d'histogrammes…….....68
.Earth Mover Distance (EMD)……..……69
.Distance de Minkowski ……69
.Distance quadratique ……70
.*Distance de Bhattacharya :* ……..……70
.Distance de Kullback Leiber (KL)……70
.Divergence de Jeffrey (JD) …….....70
.Distance de Kolmogorov Smirnov ……...71
.Distance de Cramer Von Mises ……...71
4.1. Techniques de mise en correspondance …….....71
4.1.1 L'algorithme Hongrois ……71
4.1.2 Mariages stables ……72
4.1.3 Graphes attribués (Attributed Relationnal Graph ARG)……72
4.1.4 Appariement par région integrée (IRM)……73
4.2. Conclusion…….....73
Chapitre 5. …….……..75
5. Méthodes d'appariement des graphes ……..……75
5.1. Introduction …….....75
5.2. Notions de graphes ……75
5.2.1. Définitions ……75
5.2.2. Appariements de graphes…….……..75
5.2.3. Représentation par des graphes …….....76
5.3. Mesure de similarité des graphes ……76
5.3.1.Distance d'éditions……76
5.3.2. Mesure de Papadopoulos et Manolopoulos ……76
5.3.3. Mesure de similarité basée sur le plus grand sous-graphe commun……..78
5.3.4 Mesure de similarité Ó distance ……78
5.3.5. Notion de graphe Médian ……79
5.4. Conclusion ……..80
Chapitre 6……81
6. Les modèles de recherche d'information visuelle……81
6.1. Les différents types de requêtes ……81

6.1.1 Requête par l'exemple ……..81
6.1.2 Requête par crayonnage (Sketch).
6.1.3 Requête par caractéristique
6.1.4 Requête exemple et texte
6.2 Méthodes Géométriques ……..81
6.3 Méthodes basées sur un modèle ……..82
6.4 Méthodes Fréquentielles…….82
6.5 Comparaison et évaluation……83
6.6 Raisonnement spatiale……83
. 2D string et ses variantes……..83
. $\theta R\ string$ et ses variantes ……...83
6.7 Caractérisation de haut niveau……..84
Chapitre 7……..85
7. Nouvelle approche spectrale pour CBIR……85
7.1 Méthodes de clustering ……85
7.2 Graphe de représentation minimale (MST)……85
7.3 Spectral Clustering ……92
7.3.1 Définition de la notion de NCUT (Coupure Normalisée)……..92
7.3.2 Propriétés de NCUT : Notations et définitions……95
7.3.3 Reformulation du problème de segmentation par coupure normalisée(NCUT) ……..96
7.4 NCUT et méthode des K-means pondérés ……98
7.5 Expérimentation ……..99
Conclusion générale et perspectives……...102
Bibliographie Personnelle……..105
Bibliographie……106

Lexique :

RI	*Recherche d'Information*
CBIR	*Content Based Image Retrieval*
RIC	*Recherche d'Information par le Contenu*
SIFT	*Scaled Invariant Feature Transform*
RGB	*Red Green Blue*
CIE	*Commission Internationale de l'Eclairage*
SCR	*Spatial Coherence Region*
DCD	*Dominant Color Descriptor*
MPEG-7	*Moving Picture Experts Group version 7*
EMD	*l'Earth Mover Distance*
TH	*Transformée de Hough*
MS	*Mariages Stables*
SMTI	*Stable Marriage with Ties and Incomplete lists*
ARG	*Attributed Relationnal Graph*
IRM	*Integrated Region Matching*
Sketch	*Requête par crayonnage*
MRF	*Markov Random Field*
AR	*Autoregressive models*
SAR	*Simultaneous Autoregressive models*
GMRF	*Gaussian Markov Random Field*
SGLDM	*Spatial Grey Level Dependance Method*
GLDM	*Grey Level Run Difference Method*
GLRLM	*Grey Level Run Length Method*
MST	*Minimum Spanning Tree*
CUT	*Longueur de Coupure*
NCUT	*Longueur de Coupure Normalisée*
ASSOC	*Longueur d'Association*
TW	*Wavelet transform : transformée par ondelettes*
DWT	*Discret Wavelet Transform: transformée par ondelettes discrète)*
MBR	*minimum bounding rectangle*
SMA	*Système multi-agents*

Liste des Figures :

Fig. 1 Images d'une base généraliste
Fig.2 Espace de couleur RGB et HSV
Fig.3 Espace de couleur HSV
Fig.4 Espace de couleurs RGB. Toutes les couleurs sont représentées par un point appartenant au parallélépipède rectangle.
Fig.5 Espace XYZ/RGB
Fig.6 Exemple d'image couleur avec ces composantes $L^*, u^* v^*$
Fig.7 Exemple d'image couleur avec composante HSV.
Fig.8 Exemples d'histogrammes
Fig.9 Avec une comparaison cellule à cellule, les deux histogrammes dans la situation (a) ont une intersection nulle, alors qu'ils sont très proches. Dans la situation (b), les deux histogrammes ont une intersection de 50%

alors qu'ils ne sont pas visuellement plus proches que dans le cas (a),
lorsque les cellules sont ordonnées par proximité de couleurs.
Fig.10 Exemple de création de signature. Pour obtenir les couleurs dominantes,
nous avons segmenté l'image de départ, avec l'implémentation optimisée
proposée dans la bibliothèque Open Cv.
Fig.11 Limite des histogrammes et des couleurs dominantes. Ces quatre images
ont le même histogramme et les mêmes couleurs dominantes
Fig.12 Exemple de textures de la base Brodatz
Fig.13 Exemples de matrices de cooccurrences
Fig.14 {Description d'un filtre de Gabor. L'espace de Fourier est découpé en
plusieurs régions. Pour une région donnée (ω, θ)*, les termes* $\mu\theta$ *et* μ_ω
représentent respectivement l'angle par rapport à l'axe d'origine et la position
radiale, $\sigma\theta$ *et* $\sigma\omega$ *représentent leur tolérance. Ce découpage est utilisé par*
la norme MPEG-7. Ces images ont été recentrées pour que l'origine
du plan de Fourier corresponde au centre de l'image.
Fig.15 Filtre de Gabor (a) en partie réel , (b) en partie imaginaire
Fig.16 Point anguleux
Fig.17 Points anguleux.
Fig.18 Différentes configurations pour le calcul de la moyenne
Fig.19 Mikolajczyk(2002): Les extrema du Log (Laplacian of Gaussian) donne l'échelle intrinsèque
Fig.20 Images Gaussiennes groupées par octaves
Fig.21 Filtrages Gaussiens successifs d'une image
Fig. 22 Différences de gaussiennes
Fig.23 Recherche des extrema
Fig.24 Vecteur de descripteur des points d'intérêt
Fig. 25 Détection des points d'intérêt par SIFT en différents paramètres
Fig. 26 Détection des points d'intérêt par SIFT en différents paramètres
Fig.27 Détection des points d'intérêt par SIFT en différentes échelles et différentes rotations
Fig.28 Application de la transformée de Hough sur une image couleur
Fig.29 Principaux composants d'un Système de Recherche par le Contenu
Fig.30 Problème de mise en correspondance.
Fig.31 Exemple d'appariement IRM
Fig.32 Un graphe attribué qui représente les relations entre les régions de l'image. (a) image qui contient un groupe de régions (b) le graphe qui représente (a)
Fig.33 Technique de pliage d'histogramme de Papadopoulos et Manolopoulos
Fig. 34 Exemple de construction d'un MST par l'algorithme PRIM
Fig.35 Représentation de la distance entre les différents points du graphe au cours des itérations.
Fig. 36 Nombre de classes détectés en fonction du seuil
Fig. 37 Exemple de détection de groupes effectuée par notre programme
Fig. 38 Visualisation en 3 dimensions des données
Fig. 39 Mise en relief des groupes trouvés lors de la construction du MST par l'algorithme Prim
Fig. 40 Groupes finaux après K-Means
Fig. 41 Exemples pour définir les notions de CUT et d'ASSOC
Fig. 42 Exemple de base d'index d'origine
Fig.43 La matrice d'affinité générée par clustering spectrale
Fig. 44 Résultats de Classification en utilisant seulement les deux premières valeurs propres de la matrice d'affinité
Fig. 45 Résultats de Classification en utilisant seulement les deux premières valeurs propres de la matrice d'affinité

Liste des tableaux :

Tableau 1 : Intervalles de variation des canaux $L^*, u^* v^*$.Ces intervalles sont valables pour un système $L^*, u^* v^*$ obtenu à partir de l'illuminantD_{65}. Les valeurs entre crochets indiquent les couleurs dans le système RGB qui ont permis d'obtenir ces limites.

Tableau 2 : Tableau récapitulatif des espaces couleurs.

Tableau 3 : caractéristiques statistiques extraites à partir de la matrice de cooccurrences

Tableau 4 : Comparaison des descripteurs SIFT

Tableau 5 : Exemple de liste de préférence

Chapitre 1.

1. Introduction générale.

1-1 Motivations.

Les travaux de thèse sont intervenues dans une période de plein essor pour les systèmes de recherche d'images par le contenu, la baisse des coûts des équipements informatiques, du développement des télécommunications et de la disponibilité des techniques de numérisation de haute qualité, l'échanges de volumes de plus en plus importants de données multimédias numérisées. Ces données sont par essence hétérogène et leur contenu prépondérant est visuel. Les développements récents dans les domaines de traitement du signal et des bases de données offrent tous les éléments nécessaires pour l'extraction, l'indexation et la recherche du contenu visuel des données multimédias, notamment des images. La multitude des activités, traduites soit par des publications soit par des démonstrations en ligne, avaient fait naître de nombreux espoirs.

Véritablement, quelle était la réelle ambition de ces systèmes?. C'est d'appréhender les images similaires selon certains critères et aussi la sémantique même d'une image uniquement à partir de son contenu. Habituellement, ces systèmes proposaient à l'utilisateur de rechercher une image parmi d'autres selon certains critères préétablis. Pour cela, trois étapes permettaient généralement une stratégie de recherche:

- L'interface proposée à l'utilisateur. Celle-ci doit lui permettre de signifier au système ce qu'il attend et lui proposer en retour les résultats de sa recherche.
- L'extraction des données. Les paramètres numériques permettant de décrire les images sont extraits de celles-ci et archivés. Ils renseignent sur le contenu exclusivement bas-niveau et reposent sur des méthodes issues de l'analyse et du traitement d'images.
- La gestion des connaissances. Le système doit traduire la requête sous-jacente en exploitant au mieux les informations relatives à chaque image.

Néanmoins, face à l'engouement de la communauté pour ces systèmes, les utilisateurs finaux ont opposé leurs complexités et leur philosophie de la recherche d'images. Les limites des systèmes existants et les capacités réelles de ces systèmes n'a pas permis à ces derniers de se positionner en acteur incontournable de la recherche multimédia. Devant la nécessité d'une remise en question, il est donc devenu inéluctable de converger vers d'autres approches plus réalistes, Non seulement par manque de méthodes universelles mais aussi par essence même. En effet, le volume très important de la masse d'images à parcourir rend les systèmes existants très lents par rapport aux attentes des utilisateurs.

1-2 Objectifs.

Devant ces lacunes, quelle pierre peut-on toutefois apporter à l'édifice ? Sans aucun doute et en premier lieu, tester la validité et approfondir la réelle capacité de tous les outils mis en jeu, même les plus basiques. En effet, toutes les techniques d'indexation et de recherche d'images par le contenu ont pour but d'organiser un ensemble de descripteurs (un descripteur étant un vecteur

de réels décrivant le contenu visuel d'une image et pouvant être de très grande dimension) afin que les procédures de recherches soient performantes en temps de réponses. Cette organisation se traduit généralement par une structuration des descripteurs en petits ensembles et par l'application de stratégies de recherche capable de filtrer toutes les images non pertinentes qui seront évitées (non parcourue) pendant la recherche garantissant ainsi un temps de recherche acceptable par les utilisateurs.

Toutes les phases d'un système de recherche d'images par le contenu qui aboutissent à une similarité se doivent d'être analysées et évaluées de manière objective, et à partir des conclusions, faire émerger de nouvelles pistes ou construire de nouveaux outils sous le couvert des réelles capacités de chacune des phases. La portée de ces outils se doit alors d'être réaliste. La recherche par le contenu et donc par similarité doit pouvoir s'appuyer sur des descripteurs intelligibles et robustes. L'apport d'"outils ciblés" face à des "indexeurs généralistes" trouve alors pleinement sa justification. On entend par "outil ciblé", une analyse simple de l'image qui permet d'obtenir une information bas niveau. En parallèle du peu d'informations sémantiques contenues dans la description, une grande stabilité et une complète maîtrise peuvent être envisagées. Dans ce contexte, nous présenterons une extraction bas niveau de descripteurs multiples et notamment à base de points d'intérêt invariants aux transformations telle que les points SIFT d'une image combiné à un descripteur de texture calculé avec les transformées d'ondelettes.

Les techniques d'indexation et de recherche basée sur le contenu visent à extraire automatiquement des descripteurs et à les organiser dans des index multidimensionnels pour ensuite faciliter la recherche dans les grandes bases d'images. Ces techniques ont une complexité particulière liée à la nature des données manipulées. L'état de l'art dans ce domaine fait ressortir que parmi celles-ci-certaines souffrent de la malédiction de la dimension [FBM03], et d'autres, par contre, sont spécifiques à une représentation particulière des données (distribution uniforme des données, espaces métriques ,…) et de ce fait, il est nécessaire d'élaborer des techniques d'indexation multidimensionnelles qui soient adaptées aux applications réelles pour aider les utilisateurs a faire une recherche rapide et pertinente.

1-3 Description de la thèse.

Les contributions de cette thèse, intitulée *" recherche d'images par le contenu"*, vont être scindées en sept parties.

- La première partie dresse un état de l'art succinct du domaine de la recherche d'images par le contenu de ces dernières années. Il met en exergue les difficultés rencontrées face aux problèmes réels ainsi que le temps de réponse, qui semble infranchissable, entre la demande des utilisateurs et les réelles capacités des systèmes. Il permet de recentrer notre travail de thèse et présente les motivations des contributions apportées.

- La seconde partie débute par une présentation des différents descripteurs d'images à savoir les descripteurs de la couleur, de textures, de formes et les points d'intérêts usuellement utilisées dans les systèmes de recherche par le contenu. Les différentes mesures de similarités entre descripteurs et ensuite une présentation des différentes techniques de mise en correspondances de ces derniers.

- La troisième partie, présente les composants d'un système de recherche d'images par le contenu.

- La quatrième partie, présente tout l'état de l'art en matière de calcul de similarités entre les différents types de descripteurs d'images.

- La cinquième partie, présente aussi, l'intérêt des graphes dans le domaine des CBIR ainsi que l'état de l'art en matière d'appariement de ces derniers.

- La sixième partie dresse les différents modèles de recherche d'information visuelle.

- La septième partie présente les développements que nous avons introduits et contributions pour améliorer la recherche d'images par le contenu. Nous exposons ainsi notre descripteur de bas niveau, qui est robuste et invariants aux transformations affines , qui est une association du descripteur SIFT avec le descripteur de texture calculé avec les transformées d'ondelettes et l'utilisation des méthodes à noyaux ou spectrales (spectrale clustering) à base de graphe bipartite qui assure la réduction de l'espace de représentation sans perte d'information qui répond à notre logique et démarche de minimisation du temps de recherche et d'efficacité par le maintien du rapport rappel/précision dans de bonnes normes. Dans cette partie, les expérimentations réalisées seront aussi présenté

La conclusion générale présentera à la fin du document, les principaux apports de cette thèse et introduit certaines perspectives.

Chapitre 2.

2. Descripteurs d'images.

2.1. Introduction.

L'objectif de ce chapitre est de dresser un état de l'art sur les méthodes de recherche d'images par le contenu. La littérature dans ce domaine étant très riches, nous allons dans ce qui suit présenter les méthodes les plus importantes et les plus pertinentes à nos yeux.

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

La recherche d'information couvre le traitement de documents numériques impliquant la structure, l'analyse, le stockage et l'organisation des données. Dans le passé, le terme recherche d'information était lié au concept de l'information textuelle. Actuellement 'RI' est associé à tout type d'information, textuel, visuel ou autres. Cependant dû aux limitations des méthodes textuelles, le développement des méthodes basées sur le contenu visuel est devenu primordial. Ceci explique l'activité de recherche intense consacrée au système CBIR ces dernières années. Le 'RIC' est souvent confronté au problème de pertinence de la recherche, et au temps de recherche.

L'objectif de n'importe quel système CBIR est de satisfaire la requête d'un utilisateur par la pertinence des résultats. Comme l'accès à un document via sa pure sémantique est impossible, les systèmes CBIR traditionnels s'appuient sur un paradigme de représentation de bas niveau du contenu de l'image, par la couleur, la texture, la forme, etc.…, et d'autres par une combinaison de celles-ci. La recherche d'images se fait ainsi par comparaison des descripteurs.

L'analyse et la représentation du contenu des données sources mises sous forme de vecteur caractéristique. L'information obtenue dans cette étape est une sorte de résumé des images de la base (segmentation en régions, couleur, texture, relations spatiales,…). La transformation est généralement gourmande en temps de calcul.

2.2. Descripteurs de la couleur.

La couleur est l'information visuelle la plus utilisée dans les systèmes de recherche par le contenu [SO95, CLOP01, GS99, GS97, SB91]. Ces valeurs tridimensionnelles font que son potentiel discriminatoire soit supérieur à la valeur en niveaux de gris des images. Avant de sélectionner le descripteur de couleur approprié, la couleur doit être déterminée d'abord.

2.2.1. L'espace de couleur.

Chaque pixel d'image peut être représenté comme un point dans un espace 3D. Les espaces les plus communément utilisés dans les CBIR sont: l'espace RGB, Cie L*a*b*, CIE L*u*v*, HSV (ou HSL, HSB).

Typiquement les images sont codés sur trois canaux contre un seul pour les monochromes. Il semble que son efficacité soit liée au fait que l'être humain peut distinguer des milliers de couleurs et seulement 24 niveaux de gris [GW02].

Fig. 1 : Images d'une base généraliste

Plusieurs études ont été réalisées sur l'identification d'espaces colorimétriques plus discriminants [SO95, WG02]. Par exemple la projection de l'image dans l'espace HSV permet de séparer les informations relatives à la teinte, la saturation et l'intensité [SC96, CLOP01]. Il a été démontré que la teinte est mieux invariante aux conditions d'éclairage et de prise de vue [GMD97, GS99, GVDBSG01]. D'autres espaces également fréquents dans le domaine revendiquent d'être perceptuellement uniformes et indépendant de l'intensité telle que CIE, XYZ, et CIE-LUV [GS97, STC97]. La encore ce sont des modèles de représentations et il n'existe pas un espace de couleur idéal. On trouve une comparaison entre les espaces de couleurs ainsi que leurs caracteristiques et une analyse avantages /inconvenients dans [KSGA96].

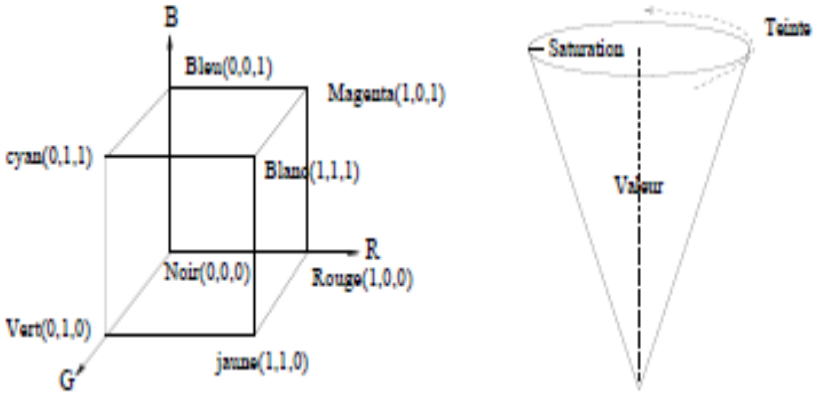


Fig. 2 : Espace de couleur RGB et HSV.

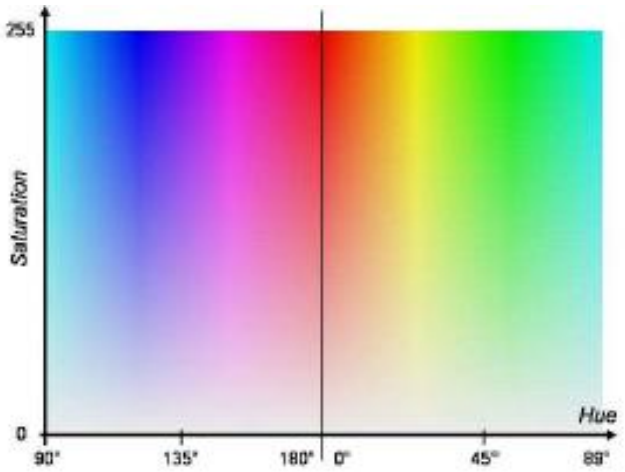


Fig. 3 : Espace de couleurs HSV.

(a) Système RGB.

Le système le plus couramment utilisé est le système RGB (Red-Green-Blue), qui est le système des trois couleurs fondamentales. Il associe à chaque couleur trois composantes (ou canaux), qui correspondent aux intensités respectives de trois couleurs primaires de la synthèse additive. Le blanc correspond à la valeur maximale pour chaque canal, tandis que le noir correspond aux trois composantes nulles.

En pratique, les valeurs de chaque canal sont des entiers compris entre 0 et N_R pour le canal rouge, entre 0 et N_G pour le canal vert et entre 0 et N_B pour le canal bleu. Ainsi, chaque couleur appartient à un parallélépipède (Figure 4). Le codage le plus couramment utilisé consiste à prendre $N_R = N_G = N_B = 255$, ce qui permet de stocker chaque composante sur un octet. C'est ce codage qui est principalement utilisé.

(b) Système normalisé (r, g, b).

Un des problèmes du système RGB est que trois canaux ne représentent pas seulement la couleur, mais aussi la luminosité. Le Système (r, g, b) permet de supprimer en partie en normalisant les trois composantes R, G, et B. On obtient alors trois canaux r, g et b, ce dernier étant facultatif car il peut être obtenu par combinaison linéaire des deux autres.

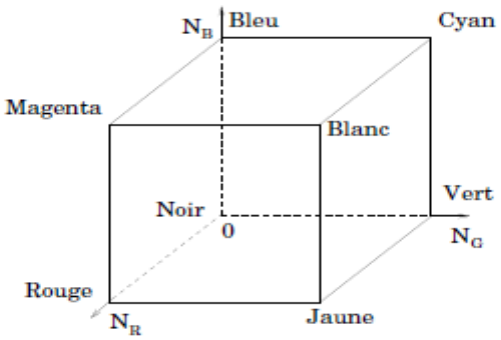


Fig.4 – Espace de couleurs RGB. Toutes les couleurs sont représentées par un point appartenant au parallélépipède rectangle.

Ainsi, on travaille en générale uniquement sur les canaux r et g, que l'on appelle système normalisé r, g, b.

Le passage du système RGB au système normalisé (r, g, b) s'effectue en divisant chaque composante par la somme des trois :

$$\begin{cases} r = \frac{R}{R+G+B} \\ g = \frac{G}{R+G+B} \\ b = \frac{B}{R+G+B} \end{cases} \tag{2.1}$$

Ainsi, la somme des trois nouvelles composantes est égale à 1. Chaque couleur normalisée est un réel appartenant à l'intervalle [0,1].

(c) Système XYZ.

La Cie (Commission Internationale de l'Eclairage) est une organisation internationale chargée d'établir des normes et des recommandations reconnues par tous les pays, afin de pouvoir quantifier la couleur : c'est la base de la colorimétrie, science de la mesure de la couleur. Le système XYZ a été établi par la CIE afin de pallier à certain inconvénients du système RGB. Ce système correspond à un changement de couleurs primaires et s'obtient simplement à partir du système RGB à l'aide d'une matrice de passage. Les coefficients de cette matrice sont déterminés par rapport à un blanc de référence que l'on appelle illuminant. Celui que nous utilisons est appelé illuminant standard D_{65}. Il existe ainsi plusieurs codage XYZ, qui dépendent de l'illuminant choisi.

Ce système de couleurs ne nous intéresse pas particulièrement, mais il constitue une étape intermédiaire pour passer au système L*u*v.

Le passage du système RGB que nous utilisons (tous les canaux sont codés par des entiers entre 0 et 255) au système XYZ se fait par le calcul matriciel suivant :

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.430574 & 0.341550 & 0.178325 \\ 0.222015 & 0.706655 & 0.071330 \\ 0.020183 & 0.129553 & 0.939180 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{2.2}$$

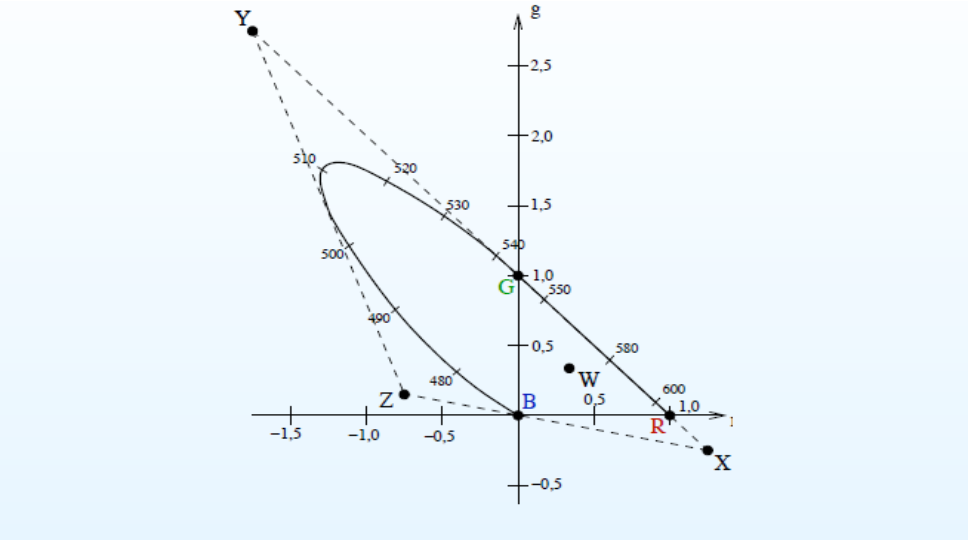


Fig . 5 : Espace XYZ/RGB.

Les coefficients de la matrice de passage sont déterminés par rapport à l'illuminant standard D_{65}.

(d) Système L*u*v.

Les systèmes RGB et XYZ ne sont pas perceptuellement uniformes: une même distance euclidienne entre deux couleurs ne sera pas perçue de la même façon par l'être humain selon les couleurs. Le problème qui se pose est le problème de la quantification des écarts de couleurs: des écarts de couleurs perceptuellement proches risquent de correspondre à des écarts importants dans le système de représentation adopté, alors que des couleurs perceptuellement très différentes risquent de correspondre à des écarts trop faibles. Le système CIE-LUV, appelé aussi L*u*v, est un système perceptuellement uniforme que l'on peut obtenir à partir du système XYZ. Il permet l'utilisation d'une distance euclidienne pour mesurer l'écart entre deux couleurs tout en évitant le problème énoncé.

On appelle **luminance** le degré de luminosité des points de l'image. Dans le système L*u*v une couleur est définie par L*, qui représente la réponse de l'œil à un niveau de luminance, et par u* et v*, qui représentent respectivement des oppositions de couleur vert-rouge et bleu-jaune.

La luminance L* se calcule par :

$$L* = \begin{cases} 116\sqrt[3]{\frac{Y}{Y_0}} - 16 \quad si \quad \frac{Y}{Y_0} > 0.008856 \\ 903.3\frac{Y}{Y_0} \quad si \quad \frac{Y}{Y_0} \leq 0.008856 \end{cases} \tag{2.3}$$

Les autres composantes sont calculées par ;

$$u^* = 13\ L^*(u' - u_0') \tag{2.4a}$$

$$v^* = 13\ L^*(v' - v_0') \tag{2.4b}$$

Avec :

$$u' = \frac{4X}{X+15Y+3Z} \quad (2.5a)$$

$$v' = \frac{9Y}{X+15Y+3Z} \quad (2.5b)$$

Où u'_0 et v'_0 sont les composantes respectives de u' et v' correspondant au blanc de référence.

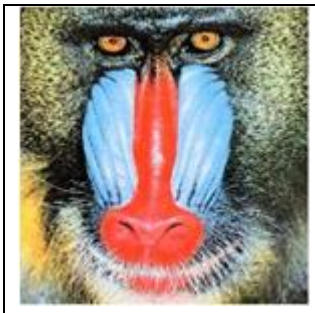
Nous avons utilisé comme blanc de référence $X_0 = 242.364495, Y_0 = 255$ et $Z_0 = 277.67358$ (obtenu en prenant R=V=B=255).

Les intervalles de variation de chaque canal $L^*, u^* v^*$ sont donnés par le tableau 1.

Ces intervalles dépendent évidement de l'illuminant choisi.

	Min	Max
L^*	0[0,0,0]	100[255,255,255]
u^*	-87.6943[0,255,0]	178.3047[255,0,0]
v^*	-133.3592[39,0,255]	105.5727[0,255,0]

Tableau 1 – *Intervalles de variation des canaux* $L^*, u^* v^*$.*Ces intervalles sont valables pour un système* $L^*, u^* v^*$ *obtenu à partir de l'illuminant* D_{65}. *Les valeurs entre crochets indiquent les couleurs dans le système RGB qui ont permis d'obtenir ces limites*.



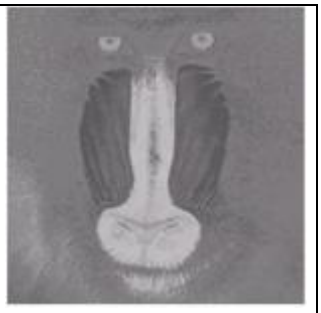


Image couleur	*Luminance Y*	*Chrominance bleue u*	*Chrominance rouge v*

Fig. 6 : Exemple d'image couleur avec ces composantes $L^*, u^* v^*$.

(e) Système $L^* a^* b^*$.

Le système $L^*a^*b^*$ est, comme le système $L^*u^*v^*$, perceptuellement uniforme. La différence entre ces deux espaces de couleur vient de l'information sur la chrominance. La

formule pour obtenir la valeur de la luminance L^* est la même, elle est donnée dans l'équation (2.3). Les composantes de chrominance sont donnée par :

$$a^* = 500\left(f\left(\frac{X}{X_0}\right) - f\left(\frac{Y}{Y_0}\right)\right) \quad (2.6a)$$

$$b^* = 200\left(f\left(\frac{Y}{Y_0}\right) - f\left(\frac{Z}{Z_0}\right)\right) \quad (2.6b)$$

Avec f défini par :

$$f(x) = \begin{cases} \sqrt[3]{x} & si\ x > 0.008856 \\ 7.787x + \frac{16}{116} & si\ x \leq 0.008856 \end{cases} \quad (2.6c)$$

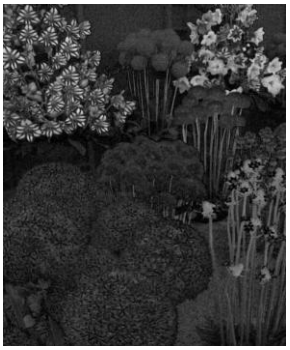
(f) Système HSV.

Le système (Hue –Saturation –Value) est défini par un cylindre qui représente la teinte, la saturation et la valeur d'une couleur. La teinte H est représentée par un angle entre 0 et 360° : elle indique la famille de couleur (rouge, jaune, vert, bleu, etc.). La **saturation** S donne une information sur la pureté de la couleur. La valeur V correspond à l'intensité lumineuse. Elle indique si la couleur est claire ou sombre. L'algorithme de transformation du système RGB vers HSV est donné par :

(1) Max = max(R, G, B)
(2) Min = min(R, G, B)
(3) V = max

V=max(R,G,B)
S=(V-min(R,G,B))*255/V if V !=0, 0 otherwise
(G-B)*60/S, if V=R
H=180+(B-R)*60/S, if V=G
240+(R-G)*60/S, if V=B
If H < 0 then H = H+360.

Image Originale


Composante saturation

Composante Teinte

Composante Valeur

Fig.7 : Exemple d'image couleur avec composante HSV.

(4) Si Max = 0
S=0
Sinon
S = (Max – Min) / Max
(5) Si Max = Min
H = 0
Sinon
Si Max = R Et G ≥ B
H = 60(G-B)/ (Max-Min)
Sinon Si Max = R Et G < B
H = 360 + 60(G-B)/(Max – Min)
Sinon Si G = Max
H = 60 (2+ (B- R)/(Max – Min))
Sinon
H = 60(4 + (R-G)/(Max-Min)

Pour obtenir des valeurs entières entre 0 et 255, la bibliothèque Open CV propose les conversions suivantes :

$$V = \max(R, G, B)$$

$$S = \frac{255(V - \min(R,G,B))}{V} \quad si\ V \neq 0, \quad 0\ sinon$$

$$H = \begin{cases} \frac{60(G-B)}{S} & si\ V = R \\ 180 + \frac{60\,(B-R)}{S} & si\ V = G \\ 240 + \frac{60(R-G)}{S} & si\ V = B \end{cases}$$

Si H < 0 alors H = H + 360
H = H/2 (pour obtenir une valeur qui tienne sur un octet).

(g) Système YCrCb.

Le système de couleur YCrCb est utilisé par les anciens standards MPEG (MPEG 1,2 et 4). On le retrouve aussi dans les images de télévisions numériques. En effet, ce système sépare l'information de **luminance et chrominance**, ce qui permet de rendre compatibles les téléviseurs couleur avec les téléviseurs noir et blanc. Il est défini par une transformation linéaire de l'espace RGB.

$$\begin{pmatrix} Y \\ Cr \\ Cb \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.500 & -0.419 & -0.081 \\ -0.169 & -0.331 & 0.500 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \qquad (2.7)$$

La distribution de couleur est une caractéristique très utilisée pour la représentation d'images. Il s'agit vraisemblablement du descripteur le plus répandu en indexation. Il est en théorie invariant aux translations et rotations, et change seulement légèrement en cas de changements de la prise de vue ou de l'échelle. La distribution de couleur est habituellement estimée à l'aide d'un histogramme, mais il existe d'autres descripteurs.

Même si la couleur semble être la signature la plus simple, son utilisation n’est pas si triviale, une pré-quantification des couleurs est nécessaire pour tous les descripteurs. En effet, dans une photographie réelle il est très rare d'avoir deux pixels ayant exactement la même couleur. Même dans une surface uniforme telle qu'un ciel bleu, il existe une légère variation des valeurs des pixels, indiscernable à l'œil. Il est donc impossible de comparer directement deux pixels pour savoir s'ils ont la même couleur. Ainsi, on remarquera que tous les descripteurs proposent une approche pour quantifier la couleur, on suppose qu'une pré-quantification a été effectuée.

Espaces couleurs : récapitulatif.

Le tableau suivant résume les principaux espaces couleurs que nous avons utilisés, ainsi que leurs qualités, du moins dans une optique de recherche par le contenu. Il peut sembler, a priori, que le panel utilisé est large mais l’influence de l’espace couleur sélectionné est grande. D’abord l’espace couleur influe sur l’algorithme lui-même, d’un point de vue conception, par exemple: discrétiser les couleurs dans l’espace RGB n’a pas le même sens que discrétiser l’espace HSV.

Quelque soit l’objectif voulu, il faut toujours rechercher la représentation, i.e. l’espace couleur, qui sera le mieux adapté aux données et à l’algorithme que l’on souhaite utiliser.

Espace Couleur	Calcul Linéaire	Distance Uniforme	Avantages Inconvénients
RGB	Oui	Non	-Format de base -Nombreux algorithmes -Axes fortement corrélés
XYZ	Oui	Non	-Espace incontournable -Décomposition Luminance/Chrominance -Nécessite de connaître les conditions d’acquisitions
X1X2X3	Oui	Non	-Axes décorrélés -Forte complexité algorithmique -Espace lié à l’image étudiée
I1I2I3	Oui	Non	-Approximation de la décorrélation -Calcul beaucoup plus rapide que X1X2X3 -Dépend des images sélectionnées pour le calcul de la matrice de passage
HSV,	Non	Oui	-Séparation Luminance, Teinte et Saturation -Bonne corrélation avec la représentation humaine des couleurs -Extraction de la teinte -Transformation non-linéaire : création d’artéfacts numériques
Y IQ	Oui	Non	-Décomposition Luminance/Chrominance

			-Meilleure décorrélation que RGB -Très utilisé en Vidéo
L_a_b_	Non	Oui	-Distance adaptée à la perception humaine -Décomposition Luminance/Chrominance -Temps de calcul important -Transformation non-linéaire : création d'artéfacts numériques -Nécessite de connaître les conditions d'acquisition -Non judicieux si la source RGB est de 8bits ou moins

Tableau 2 : Tableau récapitulatif des espaces couleurs.

2.2.2. Histogrammes.

Un histogramme est un outil statistique qui permet d'estimer la densité de probabilité d'une distribution à partir d'échantillons. L'intervalle des valeurs possibles est divisé en classes, puis pour chacune d'elles on compte le nombre d'échantillons associés.

Comme un système de couleur est composé de plusieurs canaux, son histogramme a plusieurs dimensions. L'histogramme peut alors être calculé de deux farçons :

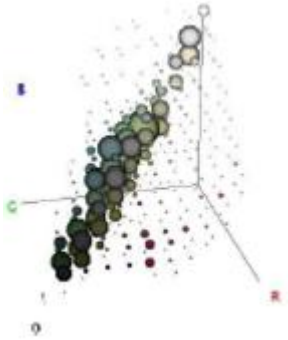
-* indépendamment sur les différents canaux. Il contient alors $\sum_{i=1}^{n} n_i$, si on note n le nombre de canaux et n_i le nombre de classes pour chaque canal.

-* par une discrétisation de l'espace complet. Le nombre de classes est alors $\prod_{i=1}^{n} n_i$.

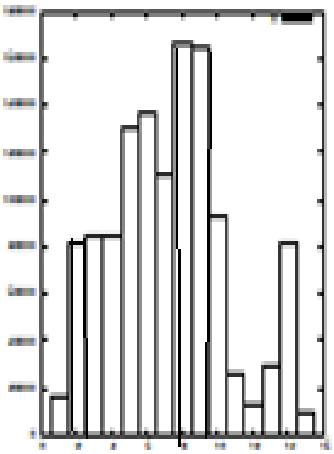
Un exemple illustrant ces deux méthodes est présenté sur la figure 5. Même si la deuxième méthode est la plus coûteuse en termes d'espace mémoire, c'est néanmoins la plus utilisée, car traiter les différents canaux indépendamment conduit à perdre beaucoup d'information sur la distribution spatiale des couleurs. En effet, dans ce cas, deux images avec des couleurs différentes peuvent avoir des histogrammes identiques.

Il existe de nombreuses mesures de similarité pour comparer deux histogrammes. Nous ne citerons ici que les plus utilisées. L'intersection d'histogrammes a été proposée par Swain et Ballard [SB91] pour l'indexation et la recherche d'images par le contenu par la couleur. Le but est de calculer le nombre de pixels en commun entre les deux histogrammes. Cette mesure a été développée pour la recherche d'images qui contiennent un objet donné, représenté par l'image requête.

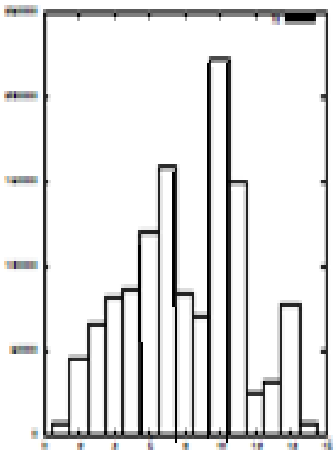
Image : RGB



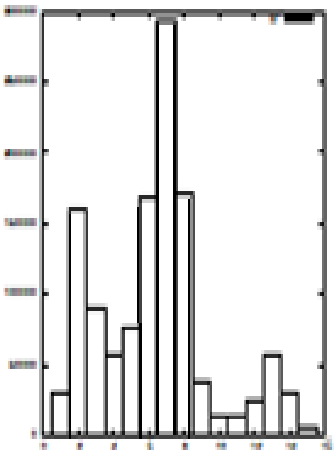
Histogramme 3D RGB



Histogramme R



Histogramme G



Histogramme B

Fig. 8 : Exemples d'histogrammes

Celle-ci est habituellement plus petite que les images de la base. Si l'on note $p = \{p_u\}\ \ u = 1,..m$, et $q = \{q_u\}\ \ u = 1,..m$, les deux histogrammes (linéarisés) à m classes, leur intersection se calcule par :

$$I(p,q) = \sum_{u=1}^{m} \min\ (p_u, q_u) \tag{2.8}$$

Pour que cette mesure de similarité soit comprise entre 0 et 1, l'intersection peut être normalisée par le nombre de pixels contenus dans l'histogramme associé à l'image la plus petite (celle servant de modèle). Par exemple, si p est l'histogramme de celle-ci, l'intersection normalisée se calcule par :

$$I_n(p,q) = \frac{\sum_{u=1}^{m} \min\ (p_u, q_u)}{\sum_{u=1}^{m} p_u} \tag{2.9}$$

La valeur 0 correspond à des histogrammes qui n'ont aucune couleur en commun, la valeur 1 signifie que les histogrammes sont identiques. L'intersection d'histogrammes donne de bons

résultats quand il s'agit de chercher un objet dans une image, car elle permet de filtrer les pixels de fond qui n'appartiennent pas à l'objet. Cependant, d'après une étude de cas proposée dans [LON03], les résultats sont inferieurs à ceux obtenus avec d'autres distances pour les comparaisons d'image à image ou d'objet à objet.

Le Khi2 (ou χ^2) est une mesure statistique, utilisée pour effectuer des tests d'hypothèses.

Cette mesure a été adaptée à la comparaison d'histogrammes ; si on note $m_u = \frac{p_u + q_u}{2}$, le χ^2 se calcule par :

$$\chi^2(p,q) = \sum_{u=1}^{m} \frac{(p_u - m_u)^2}{m_u} \tag{2.10}$$

La distance de Minkowski L_r, très utilisée en vision par ordinateur pour les mesures de corrélation, est aussi utilisée pour comparer des histogrammes [ZHANG03].

Elle se calcule par :

$$L_r(p,q) = \left(\sum_{u=1}^{m} |p_u - q_u|^r\right)^{\frac{1}{r}} \tag{2.11}$$

Où r (en général r ≥ 1) est un paramètre de la mesure. Il a été démontré qu'il existe un lien entre le cas particulier r = 1 et l'intersection d'histogrammes lorsque les histogrammes sont normalisés [SB91]. Le cas particulier r = 2 correspond quant à lui à la distance euclidienne.

2.2.3. Les moments statistiques.

La méthode d'histogramme utilise la distribution complète de la couleur. On doit stocker de nombreuses données. Au lieu de calculer la distribution complète, dans les systèmes de recherche d'images, on calcule seulement des caractéristiques dominantes de couleur tel que l'espérance, la variance et d'autres moments.

Dans [SO95], les auteurs ont prouvé que les méthodes des moments statistiques utilisées marchent plus vite et donnent des résultats meilleurs que les méthodes d'histogrammes.

L'espérance, la variance, les moments d'ordre 3 peuvent également être calculés sur chaque composante couleur par la formule suivante :

$$E_i = \frac{1}{n} \sum_{j=1}^{N} P_{ij} \tag{2.12}$$

$$\delta_i = \left(\frac{1}{N} \sum_{j=1}^{N} (p_{ij} - E_i)^2\right)^{\frac{1}{2}} \tag{2.13}$$

$$s_i = \left(\frac{1}{N} \sum_{j=1}^{N} (p_{ij} - E_i)^3\right)^{\frac{1}{3}} \tag{2.14}$$

où E_i est l'esperance, δ_i la variance et s_i est le troisième moment qui correspond à la composante couleur i.

2.2.4. Cohérence spatiale.

Ce descripteur a pour but de combler en partie, l'absence d'information spatio-colorimétrique de l'image dans le descripteur précédent. La cohérence spatiale est calculée pour chaque classe de couleur identifiée. Tout d'abord un histogramme de connexité est calculé :

$$H_I(c) = \sum_{i=0}^{X=I} \sum_{j=0}^{X=I} \delta(I(i,j), c)\alpha(i,j) \tag{2.15}$$

I est l'image segmentée de taille (X,Y), c'est la couleur du pixel (i,j), δ *est le symbole de Kronecker et* $\alpha(i,j)$ *d'éfinie par :*

$$\alpha(i,j) = \begin{cases} 1 \; si \;\; \forall\, k,k' \in (-W,W) I(i+k, j+k') + I(i,j) \\ 0 \quad sinon \end{cases} \tag{2.16}$$

La fenêtre (2W+1)(2W+1) représentant le degré de compacité souhaité. La cohérence spatiale est alors donnée par le rapport ;*

$$SCR(c) = \frac{H_I(c)}{H(c)} \tag{2.17}$$

Où H représente l'histogramme couleur et donc SRC(c) Є [0,1]. Une faible valeur de SRC(c) indiquera que la couleur c est dispersée dans l'image, tandis que pour une couleur dominante homogène SRC(c) sera proche de 1.

2.2.5. Couleurs dominantes.

L'utilisation d'histogrammes pour représenter la distribution de couleur présente quelques inconvénients. Du point de vue de l'espace mémoire, les histogrammes à plusieurs dimensions sont ((creux)), c'est-à-dire que la majorité des cellules ne comptent aucun pixel (comme sur l'exemple de l'histogramme 3D de la figure 6). Une grande partie de l'espace mémoire est utilisée inutilement. De plus, toutes les classes ont la même taille, alors qu'il serait plus intéressant d'avoir des classes plus petites dans les régions contenant des couleurs très fréquentes, et de grandes classes pour les couleurs moins répandues. Du point de vue des mesures de similarité employées, les mesures traditionnelles effectuent uniquement une comparaison cellule à cellule. Même si les histogrammes sont ordonnés, le voisinage des cellules n'est pas pris en compte quand elles ont des valeurs différentes.

Les signatures par couleurs dominantes, proposées dans [RUBNER99], permettent de résoudre ces différents problèmes. La signature $s = \{s_i = (m_i, w_i)\}$ est un ensemble de nuages de points. Chaque nuage est représenté par son mode m_i (le mode d'un nuage de point

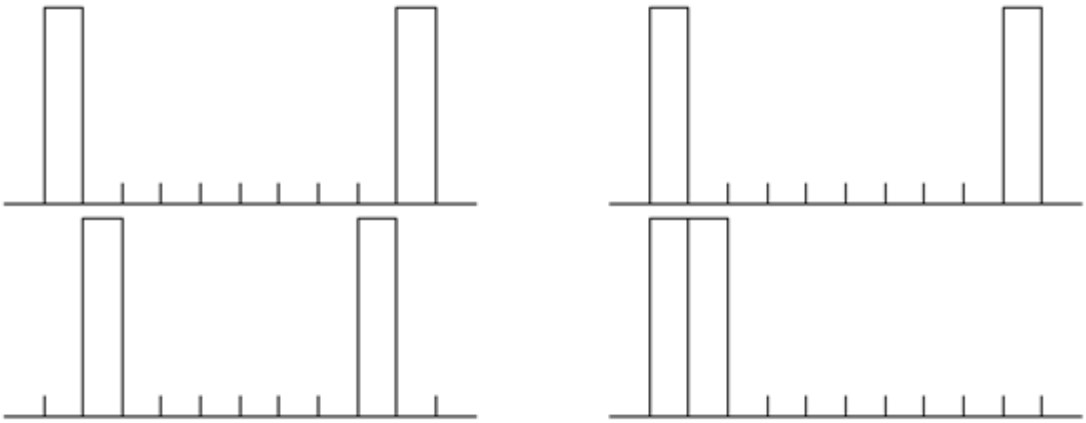
correspond à un maximum local de sa densité de probabilité), et le nombre w_i de pixels qui appartiennent au nuage.

Contrairement aux histogrammes, ces signatures ne stockent que les couleurs qui appartiennent à l'image, elles ne stockent pas les cellules vides.

La norme MPEG-7 définit un descripteur pour les couleurs dominantes, appelée DCD (Dominant Color Descriptor). D'après [PMNHD00], ce descripteur est défini par :

$$F = \{(c_i, p_i, v_i), s\} \qquad i = 1,2, \dots, N \tag{2.18}$$

Où N est le nombre de couleurs dominantes (inférieur ou égal à huit). Le terme c_i est un vecteur qui représente la $i^{ème}$ couleur dominante, attribuée à un pourcentage pi de pixels dans l'image (ou dans la région de l'image). Le terme vi représente un paramètre optionnel, la



(a) (b)

Fig. 9 : Avec une comparaison cellule à cellule, les deux histogrammes dans la situation (a) ont une intersection nulle, alors qu'ils sont très proches. Dans la situation (b), les deux histogrammes ont une intersection de 50% alors qu'ils ne sont pas visuellement plus proches que dans le cas (a), lorsque les cellules sont ordonnées par proximité de couleurs.

variance des couleurs des pixels qui sont associés à la couleur dominante c_i. Le terme s est une valeur qui représente l'homogénéité spatiale des couleurs dominantes dans l'image (en termes de connexité de régions). Le nombre d'occurrences de couleurs dominantes est variable selon les spécifications MPEG-7. Etant donné que le calcul de distance est assez complexe et pour des raisons de normalisation des calculs dans notre implémentation qui vise justement a réduire le temps de calcul, ce descripteur est peu intéressant dans un contexte d'indexation temps réel.

La mesure de similarité la plus utilisée pour comparer ces signatures est l'Earth Mover Distance, ou EMD. La distance entre deux distributions s_1 et s_2 est donnée par le coût minimum de travail nécessaire pour transformer s_1 ens_2. L'EMD peut être définie comme la solution d'un problème de transport qui peut être résolu par une optimisation linéaire.

Si on note d_{ij} la distance entre le mode m_i de s_1 et le mode m_j de s_2, et f_{ij} le point optimal entre les deux distributions, la solution de ce problème est donnée par :

$$EMD(s_1, s_2) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{m} d_{ij} f_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{m} f_{ij}} \quad (2.19)$$

Où m et n représentent respectivement le nombre de classes dans s_1 et s_2. Pour avoir plus de détails sur les problèmes d'optimisation de l'EMD, le lecteur peut se référer à [RUBNER99].

Il existe plusieurs approches pour trouver les différents nuages de points. La méthode la plus simple consiste, à partir d'un histogramme, à prendre toutes les classes qui ont un effectif non nul.

Cette approche a l'inconvénient de fournir un nombre assez élevée de classes. Une autre approche consiste à ne garder que les couleurs dominantes, et à remplacer chaque couleur de l'image par la couleur conservée la plus proche. Un exemple est donné sur la figure 10. Il existe plusieurs algorithmes pour détecter les couleurs dominantes : citons notamment la segmentation d'image, qui consiste à regrouper tous les pixels ayant une couleur proche, ou encore l'extracteur de couleurs dominantes proposée pour le descripteur de MPEG-7 par [DENMAN01]. Ce dernier effectue une classification des couleurs de l'image dans un espace perceptuellement uniforme (habituellement le système L*u*v*), à l'aide de l'algorithme de Lloyd généralisé.

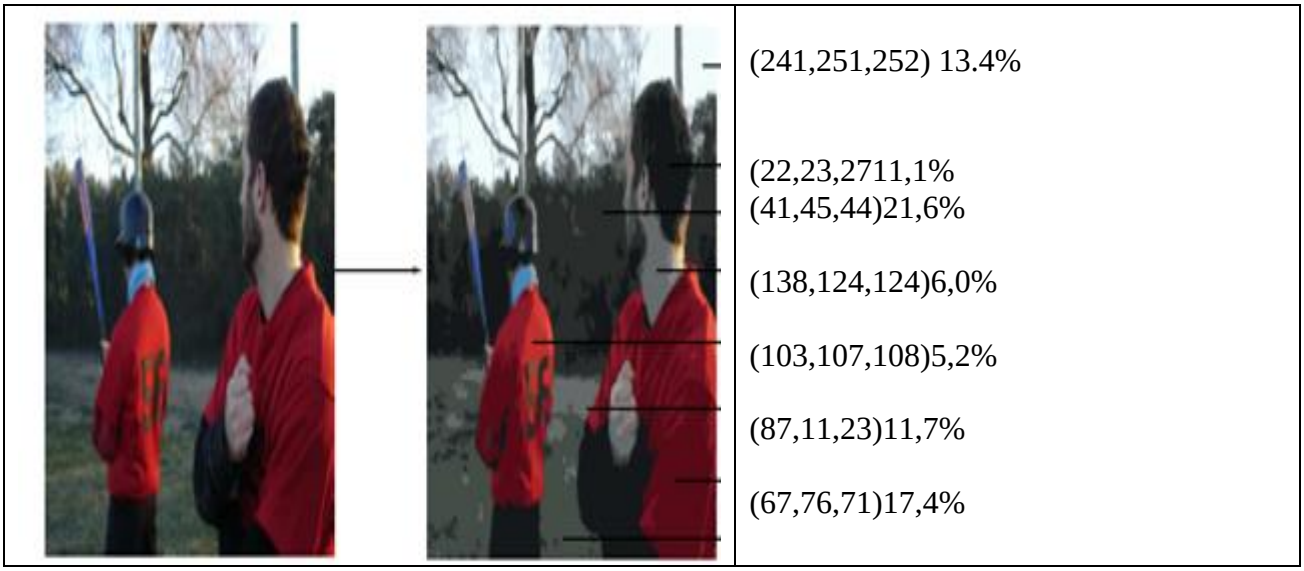


Fig.10 : Exemple de création de signature. Pour obtenir les couleurs dominantes, nous avons segmenté l'image de départ, avec l'implémentation optimisée proposée dans la bibliothèque Open Cv.

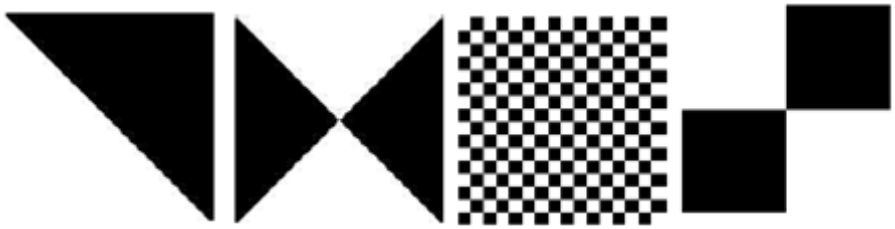


Fig.11 : Limite des histogrammes et des couleurs dominantes. Ces quatre images ont le même histogramme et les mêmes couleurs dominantes

Même si l'EMD présente de bons résultats en comparaison d'autres mesures de similarité [APCS27], son utilisation est limitée par la complexité de calcul. Même s'il existe des implémentations optimisées assez rapides, notamment dans la bibliothèque Open CV [Open CV], l'exécution demeure toujours plus lente qu'avec les distances traditionnelles appliquées sur les histogrammes.

2.2.6. Corrélogrammes.

Les histogrammes et les couleurs dominantes donnent une bonne estimation de la densité de couleur d'une image, mais ils ont l'inconvénient de perdre toute l'information sur la distribution spatiale des couleurs. Il est impossible de savoir, pour une couleur donnée, ou elle est située par rapport à une autre couleur, ou même de savoir si une couleur est repartie dans toute l'image ou si elle est présente uniquement dans une sous-partie. Ainsi, les images présentées sur la figure 2.5 sont identiques d'après leur histogramme : chaque couleur a exactement le même nombre de pixels, c'est uniquement leur répartition spatiale qui change.

Cette contrainte peut être adoucie, si l'on considère pour chaque couleur non plus le nombre de pixels ayant cette couleur, mais les probabilités pour les différentes couleurs qu'elles soient présentes, à différentes distances. Ces histogrammes améliorés sont appelés des corrélogrammes, et ont été proposés dans [ORT97]. Ils sont inspirés des matrices de cooccurrences, développées au départ pour les images en niveaux de gris [HARA79]. On note $\mathcal{L}$ l'image à indexer, et n son nombre de lignes et de colonnes (on prend l'exemple d'une image carrée pour simplifier les notations). Le terme $\mathcal{L}_c$ représente l'ensemble des pixels ayant la couleur c, et pour un entier quelconque l, on représente par [l] l'intervalle d'entiers de 1 à l. Si on quantifie l'espace de couleurs en m couleurs $c_1, c_2, \dots, c_m$, la classe c_{ij} d'un histogramme normalisé peut être reformulée en termes de probabilité par : $h_{c_i}(\mathcal{L}) = \Pr_{p\in\mathcal{L}}(p \in \mathcal{L}_{c_i})$

Pour un pixel donné de l'image, $h_{c_i}(\mathcal{L})$ donne la probabilité que sa couleur soit c_i. Si on discrétise l'intervalle réel $[0,n]$ en d parties, alors pour $(i,j) \in [m]^2$ et k $\in [d]$,

Le corrélogramme de $\mathcal{L}$ est défini par :

$$\gamma^{(k)}_{c_i,c_j}(\mathcal{L}) = Pr_{p1\in\mathcal{L}_{c_i}p2\in}(p_2 \in \mathcal{L}_{c_j}\ et\ |p_1 - p_2| = k \qquad (2.20)$$

Pour un pixel de couleur c_i de l'image, $\gamma^{(k)}_{c_i,c_j}(\mathcal{L})$ donne la probabilité qu'un pixel situé à la distance k soit de couleur c_j. Le corrélogramme a ainsi $m^2 d$ classes.

La mesure de similarité proposée dans [ORT97] pour comparer deux images $\mathcal{L}$ et $\mathcal{L}'$ est dérivée de la distance L_1, et est définie par :

$$|\mathcal{L} - \mathcal{L}'| = \sum_{i,j\in[m],k\in[d]} \frac{\gamma_{c_i,c_j}^{(k)}(\mathcal{L}) - \gamma_{c_i,c_j}^{(k)}(\mathcal{L}')}{1+\gamma_{c_i,c_j}^{(k)}(\mathcal{L}) - \gamma_{c_i,c_j}^{(k)}(\mathcal{L}')} \quad (2.21)$$

Les corrélogrammes ont été utilisées dans [ORT97, CHE03] pour l'indexation et la recherche d'images par le contenu, et semblent donner de très bons résultats en termes de reconnaissance. Cependant, l'implémentation des corrélogrammes est très délicate. Même si l'espace mémoire n'est pas forcement plus élevé que celui utilisé par les histogrammes, leur création est très couteuse en temps de calcul, ce qui nécessite une implémentation très optimisée pour qu'ils soient utilisables avec un temps de calcul raisonnable. En effet, pour chaque pixel de l'image il faut déterminer tous les pixels de chaque couleur présents à une distance k, ce qui représente une complexité $O(n^4)$ pour une image $n \times n$..

Pour diminuer la durée de création des corrélogrammes, on peut n'étudier la corrélation spatiale qu'entre les couleurs identiques. On utilise alors un autocorrélogramme, défini pour une couleur c et une distance k par :

$$\alpha_c^{(k)}(\mathcal{L}) = \gamma_{c,c}^{(k)}(\mathcal{L}) \quad (2.22)$$

Le nombre de classes passe ainsi de $m^2 d$ à md, mais le temps de création reste relativement élevé.

2.2.7. Résumé.

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

2.3. Descripteurs de textures.

La texture est une information de plus en plus utilisée en indexation et la recherche d'images par le contenu, car elle permet de pallier certains problèmes posés par l'indexation par la couleur, notamment lorsque les distributions de couleur sont très proches, comme c'est le cas sur la figure 12. Cependant, il n'existe pas de définition formelle de ce qu'est une texture, et il n'en existe pas non plus de représentation officielle.

Même si la norme MPEG-7 propose une représentation pour la texture [CHE02], il existe aujourd'hui de nombreuses approches différentes pour la décrire.

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

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

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

D’après [DEL99], une définition formelle de la texture est quasiment impossible. D’une manière générale, la texture se traduit par un arrangement spatial des pixels que l’intensité ou la couleur seules ne suffisent pas à décrire. Elles peuvent consister en un placement structuré d’éléments mais peuvent aussi n’avoir aucun élément répétitif.

De nombreuses définitions ont été proposées, mais aucune ne convient parfaitement aux différents types de textures rencontrées. Dans une définition couramment citée [POL98], la texture est présentée comme une structure disposant de certaines propriétés spatiales homogènes et invariantes par translation. Cette définition stipule que la texture donne la même impression à l’observateur quelle que soit la position spatiale de la fenêtre à travers laquelle il observe cette texture. Par contre l’échelle d’observation doit être précisée. On peut le faire par exemple en précisant la taille de la fenêtre d’observation.

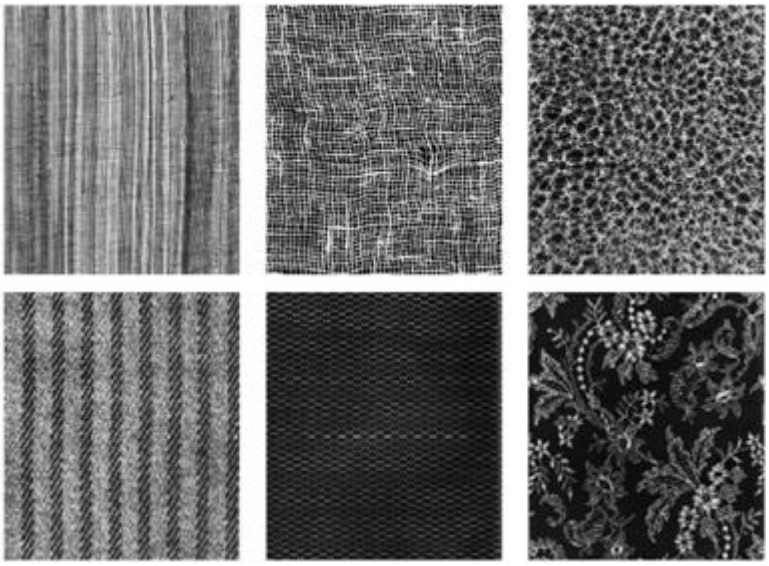


Fig. 12 : Exemple de textures de la base Brodatz.

La notion de texture est liée à trois concepts principaux:

1- un certain ordre local qui se répète dans une région de taille assez grande,
2- cet ordre est défini par un arrangement structuré de ses constituants élémentaires,
3- ces constituants élémentaires représentent des entités uniformes qui se caractérisent par des dimensions semblables dans toute la région considérée.

Il existe un grand nombre de textures. On peut les séparer en deux classes: les textures structurées (macro textures) et les textures aléatoires (micro textures).

Une texture qualifiée de structurée est constituée par la répétition d'une primitive à intervalle régulier. On peut différencier dans cette classe les textures parfaitement périodiques (carrelage, damier, etc.), les textures dont la primitive subit des déformations ou des changements d'orientation (mur de briques, grains de café, etc.).

Les textures qualifiées d'aléatoires se distinguent en général par un aspect plus fin (sable, herbe, etc.). Contrairement aux textures de type structurel, les textures aléatoires ne comportent ni primitive isolable, ni fréquence de répétition. On ne peut donc pas extraire de ces textures une primitive qui se répète dans l'image mais plutôt un vecteur de paramètres statistiques homogènes à chaque texture.

Dans tous les cas, ces objectifs nécessitent l'extraction d'un ou de plusieurs paramètres caractéristiques de cette texture. Nous désignerons ces paramètres sous le terme ***de descripteurs texturaux*** (textural features) et l'ensemble qu'ils constituent sous le terme de ***descripteur de texture.***

Certains de ces paramètres correspondent à une propriété visuelle de la texture (comme la **directionalité** ou la **rugosité**). D'autres correspondent à des propriétés purement mathématiques auxquelles il est difficile d'associer une qualification perceptive.

Un recensement ainsi qu'une classification des termes de description des textures employés par les principaux auteurs pourront être trouvés dans [PMNHD00].

Les **descripteurs texturaux** peuvent être obtenus à partir d'un ensemble assez vaste de différentes théories mathématiques. Citons notamment :

- Les descripteurs fondés sur des calculs statistiques effectués sur les niveaux de gris des pixels de l'image. C'est le cas des statistiques classiques, et des matrices de cooccurrences [HARA79, SCO90] ou de longueurs de plages ainsi que les méthodes utilisant directement la fonction de covariance ou les statistiques d'ordre supérieur [SC96, [HBME96].

- Les descripteurs obtenus à la suite de transformations orthogonales appliquées aux images (transformées de Fourier [MIT97], Ondelettes [SC96, STC97], etc.). Les descripteurs texturaux seront alors calculés dans des domaines différents de celui de la grille spatiale des luminances (domaine spectral par exemple).

- D'autres méthodes, basées par exemple sur la morphologie mathématique [HCS88, PEN94], les intégrales curvilignes [CRJP84], l'application de filtres [HSTV90, SB91, NAD94] ou sur la théorie fractale [LADT01, YLFC25] peuvent également être employées pour obtenir des descripteurs texturaux.

1. Schéma général.

Le principe le plus général de l'analyse statistique des textures est le suivant :

1- On définit un voisinage V_{ij} de tout pixel (*i*, *j*), de taille et de forme appropriés.

2- Sur le voisinage V_{ij} n on mesure des propriétés particulières de l'image : soit $\prod k$ la mesure attachée à la configuration parmi les *N* mesurées. Ces mesures seront attachées au pixel (*i, j)*.
3- On classifie les pixels à partir du vecteur formé par les $\prod k$ par l'une des nombreuses méthodes de la reconnaissance des formes.
4- Eventuellement, avant cette classification, et si le vecteur de mesures est trop grand, on réduit la dimension de l'espace par une sélection judicieuse des composantes les plus significatives ou par analyse en composantes principales.

2. La fenêtre d'analyse.

La dimension du voisinage V_{ij} est importante. C'est l'un des choix délicats de l'analyse des textures. Elle doit comprendre au moins un motif de base de la texture mesurée pour fournir des statistiques homogènes sur la texture, mais, si V_{ij} est trop grand, la précision de localisation des frontières des textures détectées sera médiocre.

La forme est généralement carrée. Taille et forme peuvent éventuellement s'adapter au signal à mesurer si l'on a quelques méthodes pour le faire (pré-segmentation, masque de contours, etc.), mais il faut s'assurer alors que la technique de classification n'est pas perturbée par des mesures faites sur des échantillons variables.

Dans notre système nous utiliserons la matrice de cooccurrences de Haralick, pour extraire la texture d'une image [HAR73].

2.3.1. Matrices de cooccurrences.

En 1973, Haralick [HAR73] a proposé une méthode en se basant sur les matrices de cooccurrences de niveaux de gris. La texture d'une image peut être interprétée comme la régularité d'apparition de couples de niveaux de gris selon une distance donnée dans l'image. La matrice de cooccurrences contient les fréquences spatiales relatives d'apparition des niveaux de gris selon quatre directions :

$\theta = 0,\ \theta = \frac{\pi}{4},\ \theta = \frac{\pi}{2},\ \theta = \frac{3\pi}{4}.$

Une matrice de cooccurrences est définie au moyen d'une relation géométrique $\Re$ entre deux pixels $(x_1, y_1) et\ (x_2, y_2)$.

La matrice de cooccurrences $P_{d,\theta}(i,j)$ est carrée et de dimension $\Delta * \Delta, où\ \Delta$ est le nombre de niveaux de gris présents dans I. Les indices de la matrice de cooccurrences sont donc les niveaux de gris de la texture étudiée.

On définit la matrice de cooccurrences

$$P_{d,\theta}\ par\ P_{d,\theta} = P_{d,\theta}(i,j) \qquad (2.23)$$

$P_{d,\theta}(i,j)$ représente le nombre de fois où un couple de points séparés par la distance *d* dans la direction θ ayant les niveaux de gris I_i et I_j. Pour obtenir de véritables fréquences relatives, il faut normaliser les éléments de la matrice en les divisant par le nombre total de paires de points élémentaires séparés par la distance *d* dans la direction θ dans toute l'image.

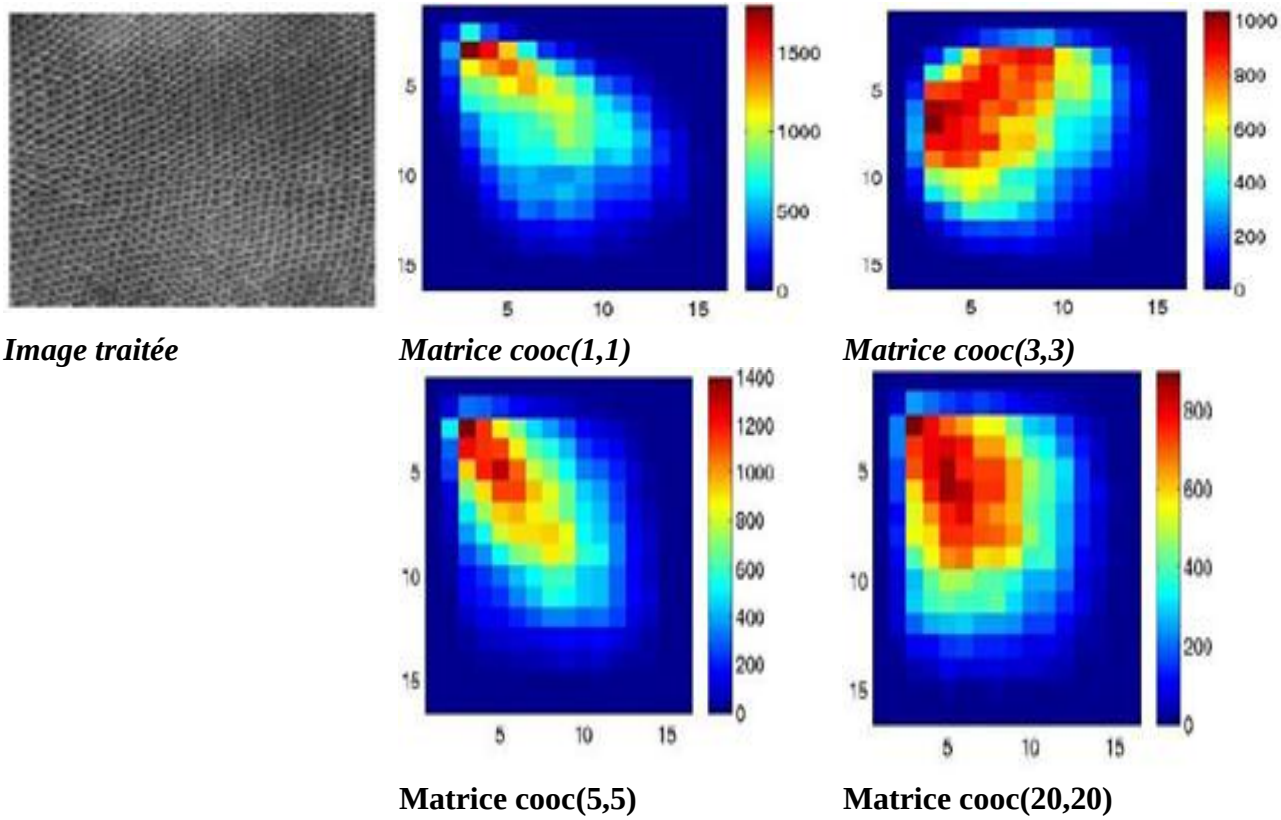


Fig. 13 : Exemples de matrices de cooccurrences.

2.3.1.1. Algorithme: Matrice de cooccurrences.

Soit $L_x = \{1,2,\dots,N_x\}$ et $L_y = \{1,2,\dots,N_y\}$ les domaines spatiaux verticaux et horizontaux de l'image et soit $\Delta = \{0,1,2,\dots,F-1\}$ les niveaux de gris.

L'image numérique $I: L_x * L_y \to \Delta$ affecte un niveau de gris à chaque pixel. Les définitions suivantes proviennent de [HAR73].

Soit 'q' une direction, et une distance '*d*' d'étude.

i- Initialisation de la matrice :$\forall i,j \in [0,L[:\ P_{d,\theta}(i,j) = 0$.

ii- Remplissage de la matrice. Pour tout point (x_1, y_1) d'intensité $I(x_1, y_1)$, le couple distance-angle (d, θ); Si la relation R entre deux pixels (x_1, y_1) et (x_2, y_2) est respectée :

$$P_{d,\theta}(i,j) = \sum_{x_i \in L_x} \sum_{y_i \in L_y} \delta(I(x_1, y_1) = i \ \ et \ I(x_2, y_2) = j) \tag{2.24}$$

Exemple : Soit l'image suivante comportant $\Delta = 4$ niveaux de gris {0,1,2,3} ;

$$I(x,y) = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 2 & 2 & 3 \\ 2 & 2 & 3 & 3 \end{bmatrix}$$

$$P_{d,0^\circ}(i,j) = \{(x_1, y_1), (x_2, y_2) \in I \ \ / \ y_1 = y_2, |x_2 - x_1| = d, I(x_1, y_1) = i \ et \ I(x_2, y_2) = j \tag{2.25}$$

Les matrices $P_{1,0^\circ}$ et $P_{1,90^\circ}$ sont donc de dimension 4 et valent

$$P_{1,0^\circ} = \begin{bmatrix} 6 & 2 & 1 & 0 \\ 2 & 2 & 0 & 0 \\ 1 & 0 & 4 & 2 \\ 0 & 2 & 2 & 2 \end{bmatrix} P_{1,90^\circ} = \begin{bmatrix} 6 & 1 & 2 & 0 \\ 1 & 2 & 1 & 1 \\ 2 & 1 & 2 & 1 \\ 0 & 1 & 1 & 2 \end{bmatrix}$$

2.3.2. Descripteurs texturaux.

Afin d'estimer la similarité entre les matrices de cooccurrences, Haralick a proposé 14 caractéristiques statistiques extraites à partir de cette matrice.

Actuellement, seulement les quatre caractéristiques les plus appropriées sont largement utilisées : l'énergie, l'entropie, le contraste et le moment inverse de différence et c'est ces caractéristiques que nous avons retenues pour notre système.

L'homogénéité h : $h = \sum_{i=1}^{n} \sum_{j=1}^{n} P_{d,\theta}(i,j)^2$ (2.26)

L'entropie e : $e = \sum_{i=1}^{n} \sum_{j=1}^{n} P_{d,\theta}(i,j) log_2 P_{d,\theta}(i,j),$ (2.27)

Le contraste c : $c = \sum_{k=0}^{n-1} (\sum_{i=1}^{n} \sum_{j=1}^{n} P_{d,\theta}(i,j)), \quad |i-j| = n$ (2.28)

Moment des Différences inverses m : $m = \sum_i \quad \sum_j \frac{1}{1+(i-j)^2} P_{d,\theta}(i,j)$ (2.29)

Les 10 autres caractéristiques statistiques extraites à partir de la matrice de cooccurrences qui permettent d'estimer la similarité entre ces matrices sont :

Entropie des sources	$es = \sum_{i=2}^{2\Delta} P_{d,\theta}(i,j) log_2 (P_{d,\theta}(i,j)$
Corrélation	$c = \frac{\sum_i \sum_j (i,j) P_{d,\theta}(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y}$ $avec\ \mu_x, \mu_y, \sigma_x, \sigma_x$: moyennes et écart types des marginales horizontales et verticales.
Variance	$v = \sum_i \sum_j (i-\mu)^2) P_{d,\theta}(i,j), avec\ \mu{:}moyenne\ des\ P_{d,\theta}(i,j)$
Moyenne des sommes	$Ms = \sum_{i=2}^{2\Delta} i P_{x+y}(i), avec\ P_{x+y}(k) = \sum_{i+j=k} P_{d,\theta}(i,j)$
Variance des sommes	$Vs = \sum_{i=2}^{2\Delta} (i - f_6)^2 P_{x+y}(i)$
Variance des différences	$Vd = variance\ des\ P_{x-y}\ avec\ P_{x+y}(k) = \sum_{i+j=k} P_{d,\theta}(i,j)$
Entropie des différences	$Ed = \sum_{i=0}^{\Delta-1} P_{x-y}(i) \log P_{x-y}(i)$
Autres mesures de corrélation	$f1 = \frac{H_{XY} - H_{XY1}}{\max\ (HX, HY)}$ $F2 = [1 - \exp(-2(H_{XY2} - H_{XY}))]^2 avec\ H_{XY} = entropie\ des\ sources\ 'es'$ $H_{XY1} = -\sum_i \sum_j P_{d,\theta}(i,j) \log\ (P_x(i) P_y(j))$ $H_{XY2} = -\sum_i \sum_j P_x(i) P_{y(j)} \log\ (P_x(i) P_y(j))$
Coefficient de corrélation maximale	$C_{cm} = (seconde\ plus\ grandes\ valeur\ de\ \mathbb{Q})^{1/2}, avec$ $\mathbb{Q}(i,j) = \sum_k \frac{P_{d,\theta}(i,k) P_{d,\theta}(j,k)}{P_x(i) P_y(k)}$

Tableau 3 : caractéristiques statistiques extraites à partir de la matrice de cooccurrences.

Considérons une image de taille M x N : le but est de décrire sa texture par des matrices de fréquences relatives $P_{\rho,\theta}(a,b)$, qui représentent le nombre de fois ou deux pixels de niveaux de gris a et b apparaissent dans l'image séparés par une distance ρ dans une direction θ. Les paramètres $\rho\ et\ \theta$ peuvent être des intervalles, si la distance et l'angle sont quantifiés. On

discrétise les niveaux de gris en *Ng* valeurs différentes, et on note f la fonction qui donne l'indice du niveau de gris discrétisé d'un pixel (i; j). Pour une distance ρ et un angle θ, un algorithme de création de la matrice de cooccurrences peut être le suivant :

(1) $\forall a \; et \; \forall b \in 1, \ldots, N_g \quad P_{\rho,\theta}(a,b) \leftarrow 0.$

(2) Pour tous les couples de pixels (i_1, j_1) $de\ l'image\ trouver\ les\ couples\ de\ pixels\ (i_2, j_2)$ à une distance $\rho\ et\ un\ angle\ \theta.$

Pour tous les pixels trouvés (i_2, j_2)

$$P_{\rho,\theta}(f(i_1,j_1), fi_2,j_2)) \leftarrow P_{\rho,\theta}(f(i_1,j_1), fi_2,j_2)) + 1$$

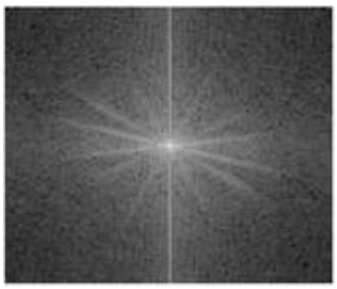
Pour la discrétisation de l'angle θ, la méthode la plus communément employée consiste à prendre quatre classes $(0^\circ, 45^\circ, 90^\circ, 135^\circ)$. La classification de textures est ensuite basée sur des critères calculés à partir de chaque matrice de cooccurrences. Ces critères sont liés à l'énergie, l'entropie, la probabilité maximale, le contraste, les moments, la corrélation, etc.

Les formules pour les calculer sont détaillées dans [HARA79].

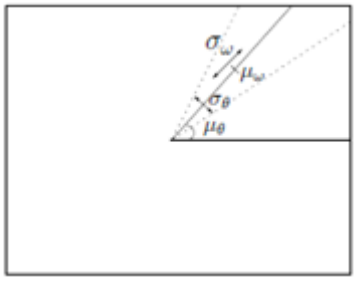
Un inconvénient de cette approche est le coût en termes de temps de calcul. Comme pour les corrélogrammes, la complexité algorithmique de la création de toutes les matrices est assez lourde, et nécessite une optimisation. Utilisé dans [HARA79], la méthode a été reprise dans de nombreux travaux, mais avec des adaptations souvent différentes [ABMC90, AKRC88, SOZ05].

2.3.3. Filtres de Gabor.

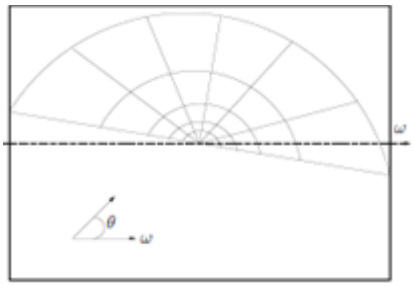
Les filtres de Gabor sont très utilisés en indexation, pour la description de la texture. Ils sont notamment utilisés par la norme MPEG-7 [PL02]. Ces filtres sont généralement exploités dans l'espace de Fourier dans le but de caractériser des textures locales.



Transformée de Fourier



Paramètres d'un filtre



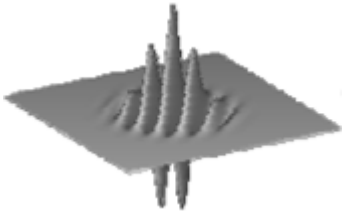
Découpage du plan de Fourier

Fig 14 : {Description d'un filtre de Gabor. L'espace de Fourier est découpé en plusieurs régions. Pour une région donnée (ω, θ), les termes $\mu\theta$ et μ_ω représentent respectivement l'angle par rapport à l'axe d'origine et la position radiale,$\sigma\theta$ et $\sigma\omega$ représentent leur tolérance. Ce

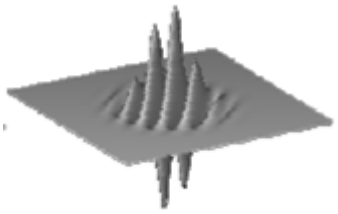
découpage est utilisé par la norme MPEG-7 [PL02]. Ces images ont été recentrées pour que l'origine du plan de Fourier corresponde au centre de l'image.

L'utilisation des filtres de Gabor consiste à analyser indépendamment différentes parties de l’espace de Fourier, à l'aide de plusieurs filtres. Chacun d’eux est défini par de plus fortes valeurs de coefficients à une fréquence et une orientation donnée : ainsi il est très sensible aux fréquences situées au centre de la région d'intérêt, et pratiquement pas partout ailleurs. Tous les filtres sont modélisés par la même loi, une gaussienne, avec des paramètres différents. Ces paramètres sont présentés sur la figure 15. Si on découpe l’espace de Fourier en M positions radiales et N angles, l'équation du filtre de Gabor ayant un indice de position r et un indice d'angle a est donnée, en coordonnées polaires sur l’espace de Fourier, et en associant l'origine au coefficient des basses fréquences, par :

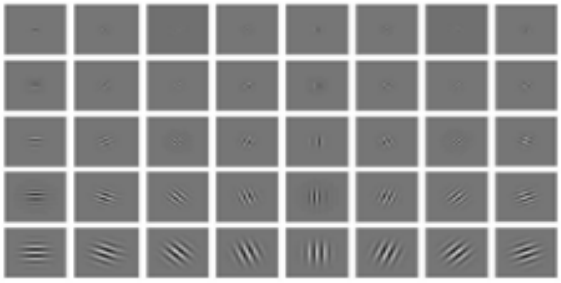
$$G_{r,a}(\omega,\theta) = \exp\left(\frac{-(\omega-\mu_{\omega_r})^2}{2\sigma\omega_r}\right)\exp\left(\frac{-(\theta-\mu_{\theta_a})^2}{2\sigma\theta_a}\right) \quad (2.30)$$



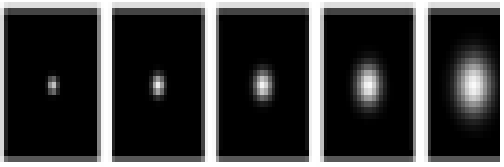
Filtre de Gabor (a) en partie réel



(b) en partie Imaginaire



Filtre de Gabor réelle (a)



Les modules de 5 fréquences (b)

Fig 15 : Filtre de Gabor (a) en partie réel , (b) en partie imaginaire.

Dans la norme MPEG-7, le découpage de l'espace de Fourier se fait en décomposant la moitié du plan en six intervalles d'angles égaux (soit un pas de 30 degrés) et en cinq intervalles de valeurs radiales inégales, ce qui génère 30 régions, appelées aussi canaux de fréquences.

Pour chaque région, deux valeurs sont stockées, la somme et l'écart-type de la convolution du filtre avec la transformée de Fourier, ce qui fournit un vecteur à 60 dimensions pour représenter une texture. Une texture donnée est ainsi décrite par l'énergie moyenne et par les variations d'énergie d'un ensemble de canaux de fréquences. Ce descripteur est appelé HTD (Homogeneous Texture Descriptor ou Descripteur de Texture Homogène).

Très utilisés pour la segmentation de texture [LASC03, AKRC88], les filtres de Gabor sont aujourd'hui une référence pour l'indexation et la recherche d'images par la texture.

2.3.4. Transformée en ondelettes.

La transformée en ondelettes est à la base de nombreuses analyses de texture, telles que les filtres de Haar [JG07]. La description de texture à base d'ondelettes est utilisée dans [HSP06] pour la recherche d'images.

Pour avoir plus d'information sur les fondements mathématiques de la transformée en ondelettes, le lecteur peut se référer au livre [VETTER95]. Comme pour la transformée de Fourier, une présentation plus pédagogique et plus historique des ondelettes peut également être trouvée dans [HUBBARD95].

L'approche continue des ondelettes pour un signal 2D est trop complexe pour être applicable rapidement sur des images. Pour résoudre ce problème, Mallat [MALA89] considère l'analyse en ondelettes comme une décomposition du signal par une cascade de filtres, en utilisant une paire de filtres pour chaque niveau de résolution (un filtre passe-haut et un filtre passe-bas). Il propose ainsi la DWT (Discrete Wavelet Transform) qui permet d'obtenir une transformée rapide. Le choix de l'ondelette mère est alors remplacé par le choix du filtre. Pour calculer une transformée en ondelettes, on n'a alors besoin que des deux filtres : au lieu de calculer le produit scalaire de l'ondelette avec le signal, on réalise un produit de convolution du signal avec ces filtres.

Une des transformées en ondelettes les plus couramment employées en analyse d'images est la transformée de Haar, mais d'autres ondelettes sont aussi largement exploitées [VETTER95]. Les filtres de Haar sont fréquemment employés en apprentissage pour obtenir la description d'un objet (comme un visage ou une personne).

Ces filtres ont été utilisés dans [PAPAGE98] pour la détection de personnes et la détection de visages, puis repris et améliorés dans [LIENHART02, VIOLA01, VIOLA03]. Pour plus d'information sur ses fondements mathématiques, une présentation détaillée de la transformée de Haar peut être trouvée dans [VETTER95, ch. 3.1.2 p. 98]), ainsi que dans [MALA89].

2.4. Les descripteurs des points d'intérêts.

Les points anguleux (ou points d'intérêt, points saillants,...) sont des points « qui contiennent beaucoup d'information » relativement à l'image. Ce sont des points aux voisinages desquels l'image *varie significativement dans plusieurs directions*.

Fig. 16 : Point anguleux.

Une mesure des variations locales de l'image *I* au point (x, y) associée à un déplacement $(\Delta x, \Delta y)$ est fournie par la ***fonction d'autocorrélation*** :

$$\chi(x,y) = \sum_{(x_k,y_k)\in W}(I(x_k,y_k) - I(x_k+\Delta x, y_k+\Delta y))^2 \qquad (2.31)$$

Où *W* est une fenêtre centrée au point (x, y).

Sachant que, en utilisant une approximation du premier ordre :

$$I(x_k+\nabla x, y_k+\Delta y) \approx I(x_k,y_k) + \left(\frac{\delta I}{\delta x}(x_k,y_k) \;\; \frac{\delta I}{\delta y}(x_k,y_k)\right).\begin{pmatrix}\Delta x\\ \Delta y\end{pmatrix} \qquad (2.32)$$

Et donc :

$$\chi(x,y) = \sum_{(x_k,y_k)\in W}\left(\left(\frac{\partial I}{\partial x}(x_k,y_k)\frac{\partial I}{\partial y}(x_k,y_k)\right).\begin{pmatrix}\Delta x\\ \Delta y\end{pmatrix}\right)^2 \qquad (2.33)$$

$$\chi(x,y) = (\Delta x \; \Delta y)\begin{pmatrix} \sum_{(x_k,y_k)\in W}\left(\frac{\partial I}{\partial x}(x_k,y_k)\right)^2 & \sum_{(x_k,y_k)\in W}\frac{\partial I}{\partial x}(x_k,y_k)\frac{\partial I}{\partial y}(x_k,y_k) \\ \sum_{(x_k,y_k)\in W}\frac{\partial I}{\partial x}(x_k,y_k)\frac{\partial I}{\partial y}(x_k,y_k) & \sum_{(x_k,y_k)\in W}\left(\frac{\partial I}{\partial y}(x_k,y_k)\right)^2 \end{pmatrix}\begin{pmatrix}\Delta x\\ \Delta y\end{pmatrix}$$

$\Xi(x,y)$: *Matrice d'autocorrélation de l'image I en* (x,y)

$$\Xi(x,y) = \begin{pmatrix} \sum_{(x_k,y_k)\in W} \left(\frac{\partial I}{\partial x}(x_k,y_k)\right)^2 & \sum_{(x_k,y_k)\in W} \frac{\partial I}{\partial x}(x_k,y_k)\frac{\partial I}{\partial y}(x_k,y_k) \\ \sum_{(x_k,y_k)\in W} \frac{\partial I}{\partial x}(x_k,y_k)\frac{\partial I}{\partial y}(x_k,y_k) & \sum_{(x_k,y_k)\in W} \left(\frac{\partial I}{\partial y}(x_k,y_k)\right)^2 \end{pmatrix} \quad (2.34)$$

La matrice d'autocorrélation X représente la variation locale de l'image *I* en (x,y) .

(x,y) sera considéré comme un point anguleux de I si pour tous les déplacements $(\Delta x, \Delta y)$, la quantité : $(\Delta x, \Delta y).\Xi(x,y).(\Delta x, \Delta y)^t$ est grande.

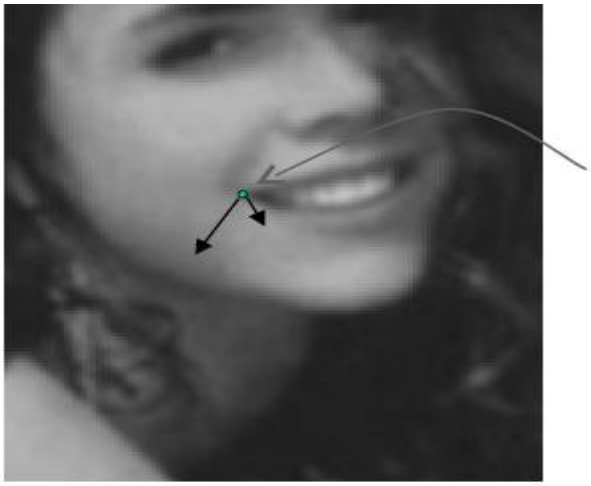


Fig. 17 : Points anguleux.

Les points anguleux sont les points (x,y) pour lesquels la matrice d'autocorrélation $X(x,y)$ a ***deux valeurs propres grandes***.

Cela correspond aux points pour lesquels il existe localement une base de vecteurs propres de X décrivant des variations locales importantes de l'image.

Le ***détecteur de Harris*** calcule une ***fonction d'intérêt*** $\Theta(x,y)$: :

$$\Theta(x,y) = det\Xi - \alpha\ trace\ \Xi \quad (2.35)$$

Le premier terme correspond au produit des valeurs propres, le second terme pénalise les points de contours avec une seule forte valeur propre.

Les points d'intérêt correspondent aux maxima locaux de la fonction Θ qui sont au delà d'un certain seuil (typiquement 1% de la valeur max de Θ).

2.4.1. Quelques méthodes de détection.

Bien avant le SIFT (Scale Invariant Feature Transform), plusieurs méthodes de caractérisation d'image furent développées, telle la méthode de Moravec et la méthode de Harris.

2.4.2. Le détecteur de Moravec (1980).

L'idée du détecteur de Moravec est de considérer le voisinage d'un pixel (une fenêtre) et de déterminer les changements moyens de l'intensité dans le voisinage considéré lorsque la fenêtre se déplace dans diverses directions. Plus précisément, on considère la fonction :

$$E(x,y) = \sum_{u,v}[w(u,v)|I(x+u,y+u) - I(u,v)|^2], \qquad (2.36)$$

Où :

w spécifie la fenêtre/voisinage considérée (valeur 1 à l'intérieur de la fenêtre et 0 à l'extérieur) ;
I(u,v) est l'intensité au pixel (u,v) ;

E(x,y) représente la moyenne du changement d'intensité lorsque la fenêtre est déplacée de (x,y).

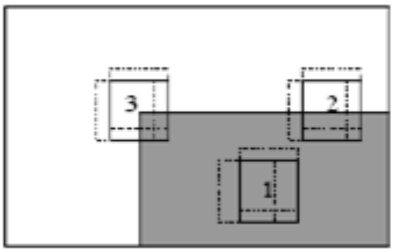


Fig. 18 : Différentes configurations pour le calcul de la moyenne.

En appliquant cette fonction dans les trois situations principales suivantes (voir la figure ci-dessus), on obtient :

1. L'intensité est approximativement constante dans la zone image considérée : la fonction 'E' prendra alors de faibles valeurs dans toutes les directions (x, y).

2. La zone image considérée contient un contour rectiligne : la fonction 'E' prendra alors de faibles valeurs pour des déplacements (x, y) le long du contour et de fortes valeurs pour des déplacements perpendiculaires au contour.

3. La zone image considérée contient un coin ou un point isolé : la fonction 'E' prendra de fortes valeurs dans toutes les directions.

En conséquence, le principe du détecteur de Moravec est donc de rechercher les maxima locaux de la valeur minimale de E en chaque pixel (au dessus d'un certain seuil).

2.4.3. Le détecteur de Harris (1988).

Le détecteur de Moravec fonctionne dans un contexte limité. Il souffre en effet de nombreuses limitations. Harris et Stephen ont identifié certaines limitations et, en les corrigeant, ils ont déduit un détecteur de coins très populaire : **le détecteur de Harris**. Les limitations du détecteur de Moravec prises en compte sont :

1. La réponse du détecteur est anisotropique en raison du caractère discret des directions de changement que l'on peut effectuer (des pas de 45 degrés). Pour améliorer cet aspect, il suffit de considérer le développement de Taylor de la fonction d'intensité I au voisinage du pixel (u, v) :

$$I(x+u, y+v) = I(u,v) + x\frac{\delta y}{\delta x} + y\frac{\delta I}{\delta y} + o(x^2, y^2), \quad (2.37)$$

$$E(x,y) = \sum_{u,v} w(u,v)[x\frac{\delta I}{\delta x} + y\frac{\delta I}{\delta y} + o(x^2, y^2)]^2 \quad (2.38)$$

En négligeant le terme $o(x^2, y^2)$ (valide pour les petits déplacements), on obtient l'expression analytique suivante :

$$E(x,y) = Ax^2 + 2Cxy + By^2 \quad (2.39)$$

Avec :

$$. A = (\frac{\delta I}{\delta x})^2 * w$$
$$. B = (\frac{\delta I}{\delta y})^2 * w \quad (2.40)$$
$$. C = \frac{\delta I}{\delta x}\frac{\delta I}{\delta y} * w$$

2. La réponse du détecteur de Moravec est bruitée en raison du voisinage considéré. Le filtre w utilisé est en effet binaire (valeur 0 ou 1) et est appliqué sur un voisinage rectangulaire. Pour améliorer cela, Harris et Stephen proposent d'utiliser un filtre Gaussien : $w(u,v) = e^{-(u^2+v^2)/2\sigma^2}$

3. Enfin, le détecteur de Moravec répond de manière trop forte aux contours, en raison du fait que seul le minimum de E est pris en compte en chaque pixel. Pour prendre en compte le comportement général de la fonction E localement, on écrit :

$$E(x,y) = (x,y).M.(x,y)^t \quad (2.41)$$

Avec :

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix} \quad (2.42)$$

La matrice M caractérise le comportement local de la fonction E, les valeurs propres de cette matrice correspondent en effet aux courbures principales associées à E :

Si les deux courbures sont de faibles valeurs, alors la région considérée a une intensité approximativement constante.

Si une des courbures est de forte valeur alors que l'autre est de faible valeur alors la région contient un contour.

Si les deux courbures sont de fortes valeurs alors l'intensité varie fortement dans toutes les directions, ce qui caractérise un coin.

Par voie de conséquence, Harris et Stephen proposent l'opérateur suivant pour détecter les coins dans une image :

$$R = Det(M) - k.Trace(M)^2, avec$$

$$Det(M) = AB - C^2 \ \ et \ trace(M) = A + B$$

Les valeurs de R sont positives au voisinage d'un coin, négatives au voisinage d'un contour et faibles dans une région d'intensité constante.

2.4.4. Extraction de caractéristiques des points d'intérêt par SIFT.

Dans les sections précédentes, les caractéristiques de couleur et de texture sont extraites. Cependant, pour la représentation pertinente de la signature d'une image, des points d'intérêt ou des régions saillantes devront être détectés et localisés afin d'obtenir les informations plus précises des nuances entre des images différentes.

Une approche représentative pour caractériser les informations des points d'intérêt est la méthode SIFT (*Scale-Invariant Feature Transform*) proposée et développée par David Lowe [LOW99, BLOW02+]. SIFT est un algorithme qui permet de détecter des points d'intérêt et d'extraire des caractéristiques distinctives de ces points pour la reconnaissance d'objet. Les caractéristiques de SIFT sont invariantes à l'échelle et à la rotation, ce qui joue un rôle très important pour la reconnaissance des images a angle de prise de vue très diverses.

- ***Points SIFT***

La détection et l'extraction de caractéristiques sur les points d'intérêt se déroulent en quatre étapes :

• détection d'extrema d'espace-échelle ("scale-space"),
• localisation des points d'intérêt,
• choix de l'orientation des descripteurs,
• calcul des descripteurs.

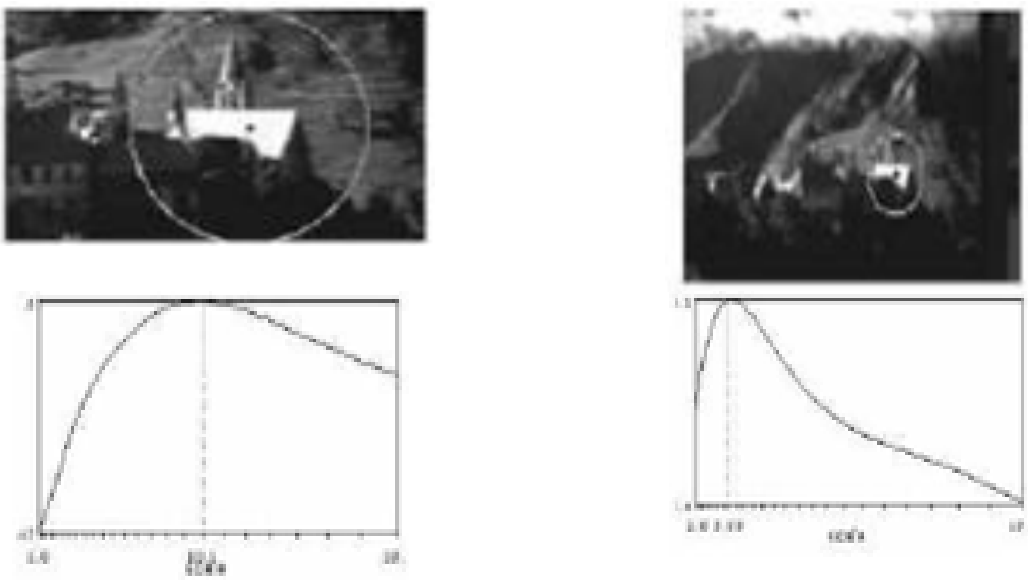


Fig.19 : Mikolajczyk(2002)[MI02]: Les extrema du Log (Laplacian of Gaussian) donne l'échelle intrinsèque

Pour la première étape de détection d'extrema d'espace-échelle, l'image est convoluée avec un noyau gaussien. L'espace-échelle d'une image est donc défini par la fonction :

$$L(x, y.\sigma) = G(x, y.\sigma) * I(x, y) \quad (2.43)$$

où I(x, y) est l'image originale et $G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$ (2.44)

A une normalisation près, cela revient à résoudre $\frac{\partial y}{\partial x} = \Delta I,$ où ΔI représente le Laplacien de I.

La *Figure 20* nous montre les images Gaussiennes groupées par quatre octaves. De gauche à droite, l'échelle augmente. De haut à bas, la taille d'image divisée par deux.

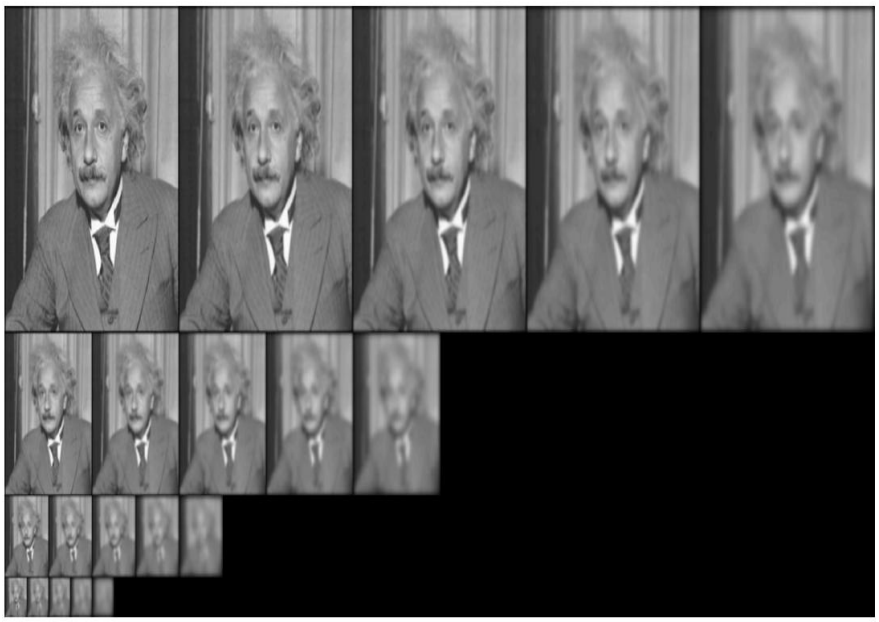


Fig.20 : Images Gaussiennes groupées par octaves

La présélection des points d'intérêt et de leur échelle est faite en détectant les extrema locaux des différences de gaussiennes (*Figure 20*):

$$D(x, y, k\sigma) = \big(G(x, y, k\sigma) - G(x, y, \sigma)\big) * I(x, y) \tag{2.45}$$

$$D(x, y, k\sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{2.46}$$

Fig.21 : Filtrages Gaussiens successifs d'une image

Notons que $D(x, y, \sigma) \approx (k-1)\,\Delta I \quad lorsque \quad k \to 1$

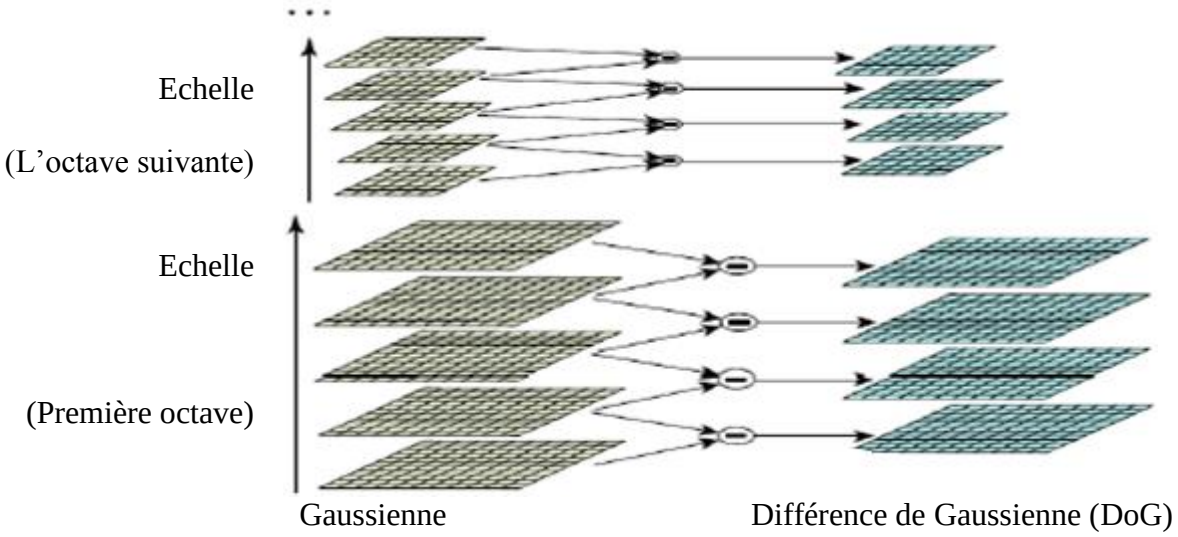


Fig. 22 : Différences de gaussiennes

Les extrema sont recherchés dans de petits voisinages en position et en échelle (typiquement 3 x 3 x 3) (*Figure 23*).

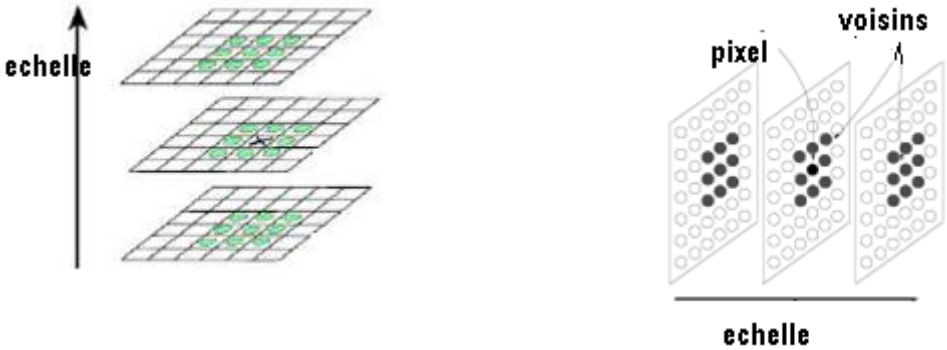


Fig.23 : Recherche des extrema

Une étape d'interpolation a pour but d'améliorer la localisation des points d'intérêt en espace et en échelle. Puis une analyse des rapports des valeurs propres de la matrice hessienne '2 x 2' permet d'éliminer les points d'intérêt situés dans des zones insuffisamment contrastées ou sur des bords présentant une courbure trop faible.

L'étape suivante consiste à assigner à chaque point une orientation. Cette orientation correspond à l'orientation majoritaire des gradients spatiaux d'intensité calculés dans un voisinage du point d'intérêt à l'échelle préalablement déterminée. Un point d'intérêt peut se voir associer plusieurs orientations. Cela entraîne par la suite une redondance des descripteurs.

Finalement, pour une position, une échelle et une orientation données, chaque point d'intérêt se voit associer un descripteur. Pour chaque image, la norme du gradient spatial $m(x,y)$ et l'orientation du gradient spatial $\theta(x,y)$ correspondants à cette échelle sont calculées :

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \quad (2.47)$$

$$\theta(x,y) = tan^{-1}\left(\frac{L(x,y+1)-L(x,y-1)}{L(x+1,y)-L(x-1,y)}\right) \quad (2.48)$$

Le descripteur est constitué d'histogrammes d'orientation du gradient spatial d'intensité pondérés par la norme du gradient spatial (*Figure 24*).

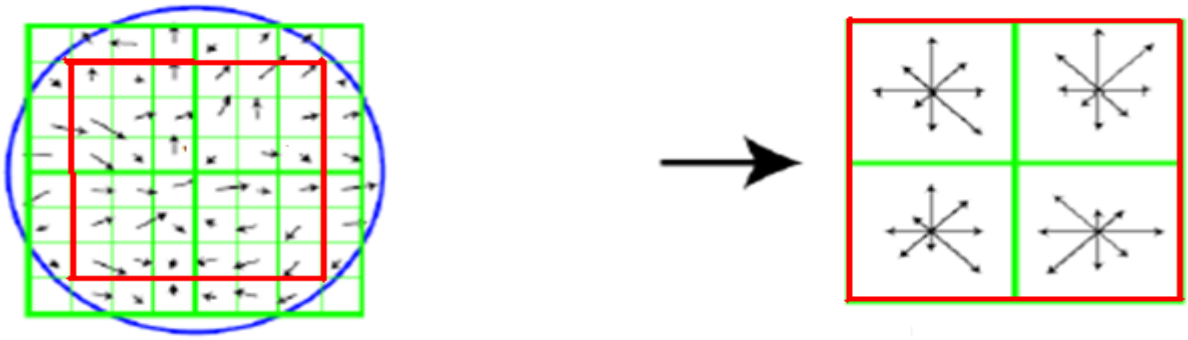


Image gradients Descripteur des points

Fig.24 : Vecteur de descripteur des points d'intérêt

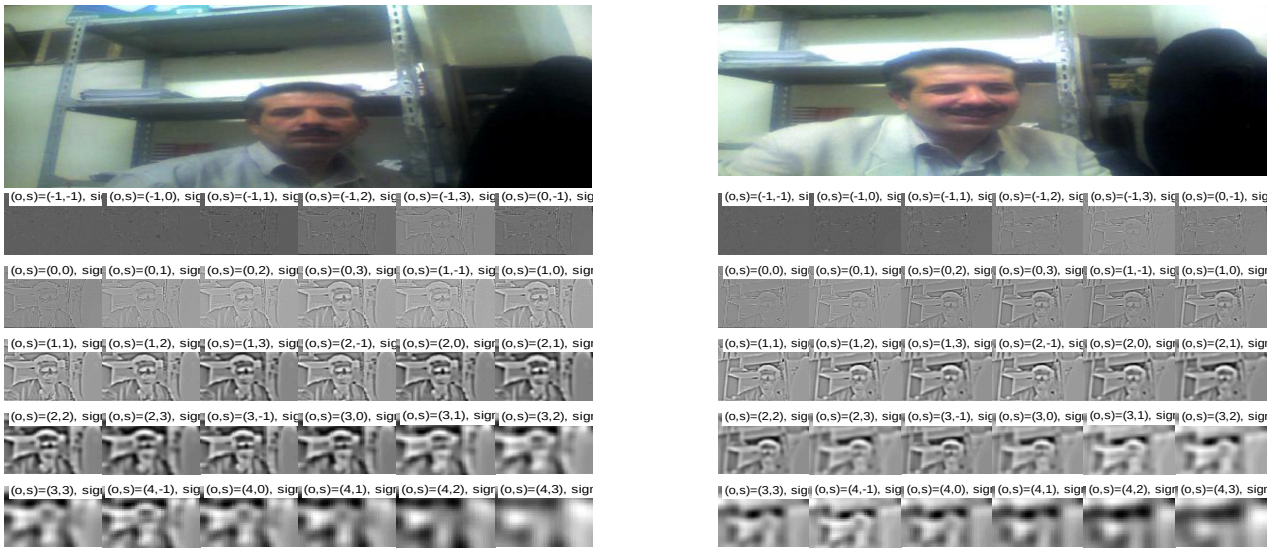
En effet, le voisinage du point d'intérêt dont la taille dépend de l'échelle subit un découpage 4 x 4 en blocs. Pour chaque bloc, un histogramme à 8 niveaux de quantification résume les orientations du gradient spatial d'intensité à l'intérieur du bloc. Le descripteur SIFT est donc un vecteur à 4 x 4 x 8 = 128 coordonnées.

- ***Implémentation***

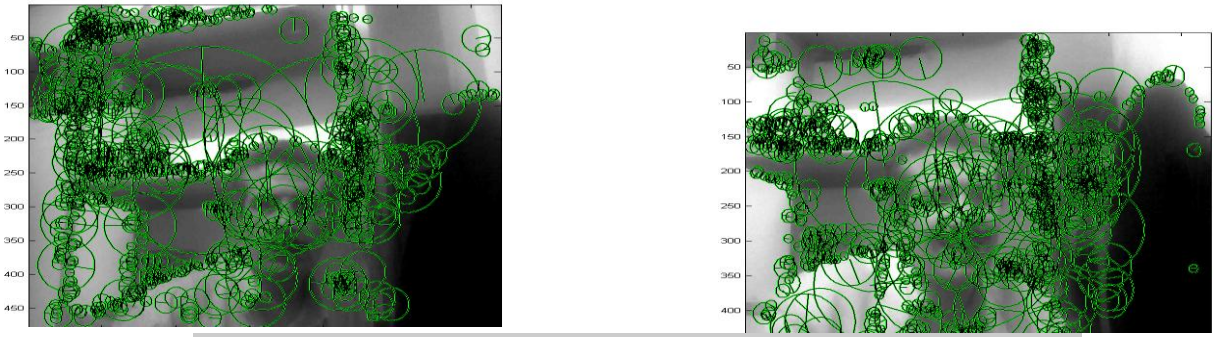
Pour l'implémentation de SIFT, nous avons utilisé les codes « SIFT++ » de Andrea Vedaldi de l'Université de California en 2006, qui est une implémentation de l'algorithme des détecteurs et descripteurs SIFT en C++ basée sur l'article [BLOW02+]. Nous l'avons utilisé, à la fois pour détecter des points d'intérêt au sein des images de visages et des images quelconques et pour appliquer le descripteur SIFT aux zones détectées.

Le premier essai est de configurer différents paramètres (échelle de premier octave, seuil de contraste, seuil de bord, etc.) sur des images de visages (*Figure 25*), et sur d'autres images *(Figure 26).* Le nombre de points détectés est différent selon les paramètres configurés.

Exemple 1 :



(a) (b)



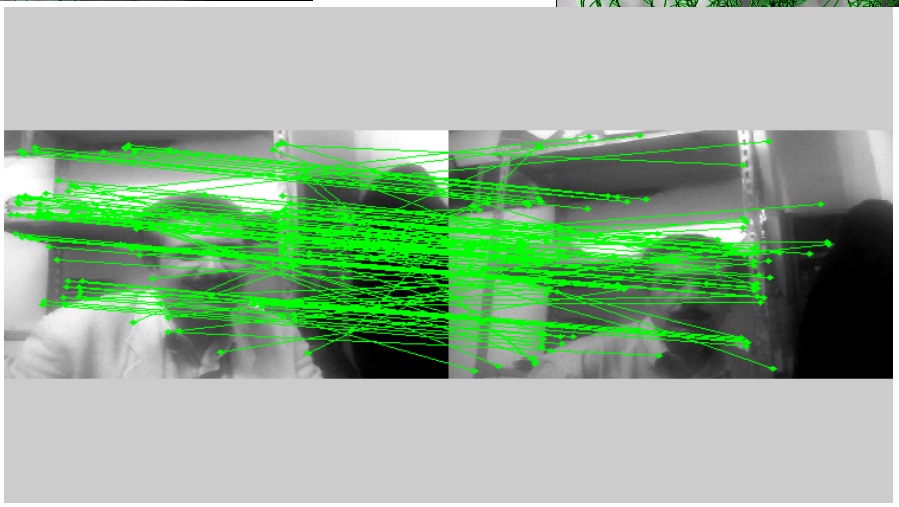


Fig. 25 : Détection des points d’intérêt par SIFT en différents paramètres

(a) fist-octave = -1, threshold = 0.007, edgeThreshold = 10
(b) fist-octave = 0, threshold = 0.02, edgeThreshold = 5
(c) fist-octave = 0, threshold = 0.03, edgeThreshold = 5

Exemple 2 :

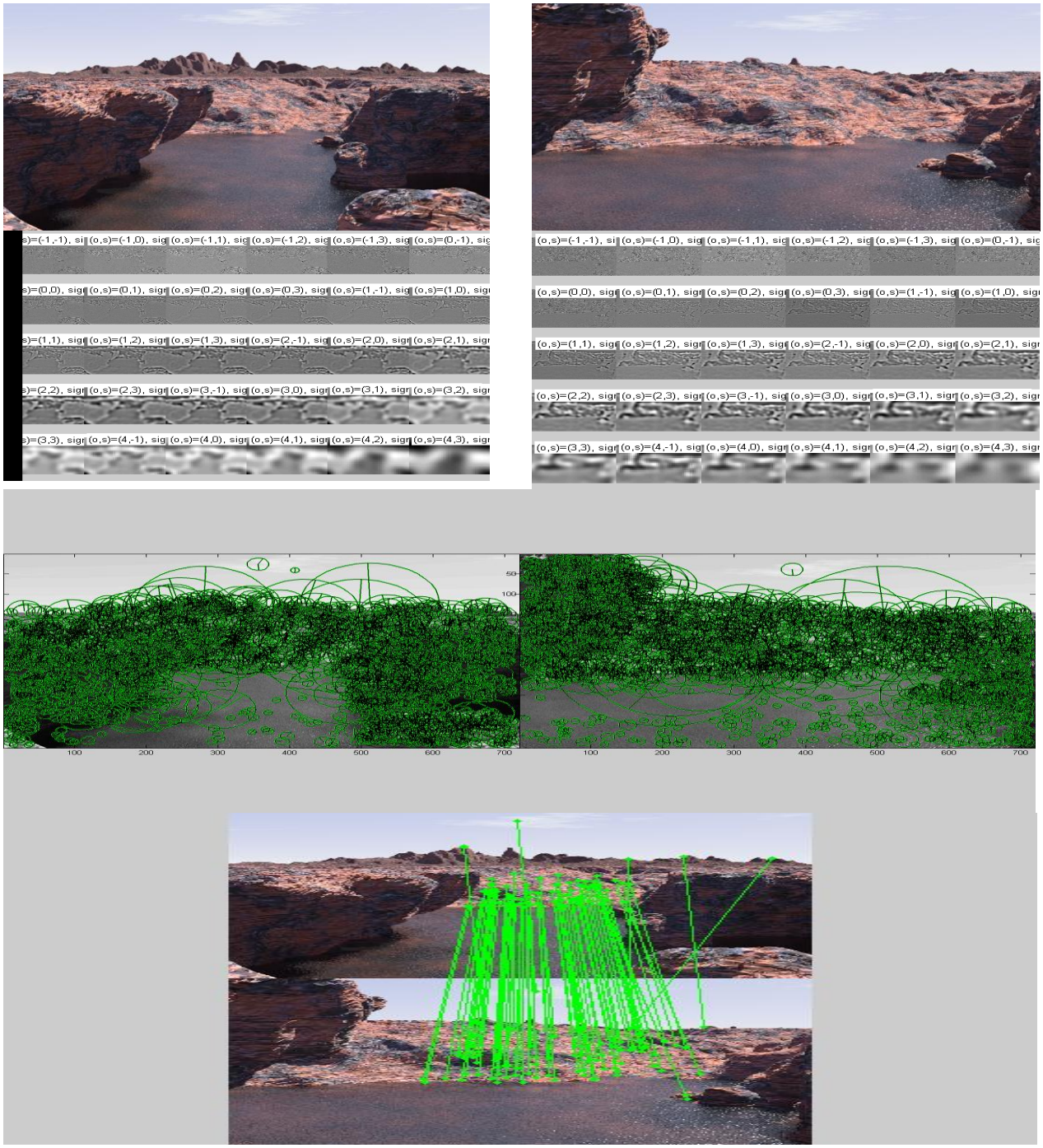
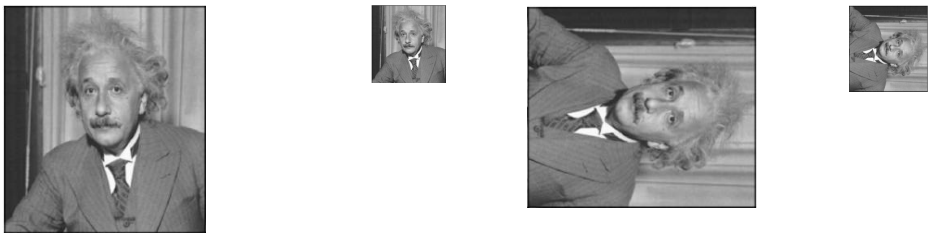


Fig. 26 : Détection des points d'intérêt par SIFT en différents paramètres

(a) fist-octave = -1, threshold = 0.007, edgeThreshold = 10
(b) fist-octave = 0, threshold = 0.02, edgeThreshold = 5
(c) fist-octave = 0, threshold = 0.03, edgeThreshold = 5

Avec les mêmes paramètres comme *Figure 25*, nous avons détecté des points d'intérêt des images de visages de différentes échelles et de différentes positions (orientations) voir la *Figure 26.*

Pour l’extraction des caractéristiques sur les points d’intérêt, on obtient un fichier *.key qui contient les vecteurs des descripteurs SIFT pour chaque point détecté, représentant le coordonnées x et y, échelle, orientation, et les 128 composants des descripteurs SIFT. Pour évaluer les vecteurs extraits par SIFT, quatre images de même visage en différentes échelles et différentes rotations sont calculées, comme montrées sur la *Figure 27*, les points d’intérêt détectés sont relativement stables, robustes, sur le même visage en différentes échelles et différentes rotations.



(a) (b) (c) (d)

Fig.27 : Détection des points d’intérêt par SIFT en différentes échelles et différentes rotations

Image	PI(x)	PI(y)	Echelle	Orientation (de 0 à 8)	128 composants des descripteurs SIFT
(a)	54.52	73.08	5.39	3.259	0 12 43 31 3 000 23 62 17 00000 155 17 0…
(b)	26.98	36.26	2.65	3.251	0 13 45 29 2 000 22 64 16 00000 154 17 0…
(c)	154.00	54.58	5.41	4.812	0 12 43 33 3 000 21 63 18 00000 153 17 0…
(d)	76.72	26.95	2.66	4.815	0 13 46 29 2 000 21 65 16 00000 153 17 0…

Tableau :4 Comparaison des descripteurs SIFT

Les descripteurs SIFT pourraient être discriminants pour la rechercher d’images couleur dans des bases généralistes, ce qui justifie leur utilisation dans notre système.

2.5. Descripteurs de formes.

La forme est un descripteur très important dans l'indexation des images. La forme désigne l’aspect général d’un objet, son contour. Nous présentons dans ce qui suit la méthode utilisé permettant de reconnaître une forme donnée dans une image. Itérée pour toutes les formes d’une image, cette méthode permet finalement de relever toutes les formes communes à deux images.

2.5.1. Les descripteurs géométriques de région.

Les descripteurs géométriques de forme permettent de distinguer les différents types de forme que peuvent prendre les objets d'une scène. Ils nécessitent une segmentation en région préalable de l'image. Ils sont ensuite calculés sur les différentes régions de l'image [LAF05, LAF06].

La surface relative (ou normalisée) d'une région $\Re_k$ de l'image *I* est le nombre de pixels contenus dans cette région par rapport au nombre total de pixels de l'image :

$$S_k = \frac{card(R_k)}{hauteur(I) * largeur(I)} \tag{2.49}$$

Le centre de masse des pixels de la région est définie par :

$$P = (P_i, P_j) = \left(\frac{\sum_{i=R_k} i/card(R_k)}{largeur(I)}, \frac{\sum_{j=R_k} j/card(R_k)}{hauteur(I)}\right) \tag{2.50}$$

La longueur du contour de la région est le nombre de pixels en bordure de la région:

$$l_k = card(contour(R_k)) \tag{2.51}$$

La compacité traduit le regroupement des pixels de la région en zones homogènes et non trouées:

$$C_k = \frac{l_k^2}{S_k} \tag{2.52}$$

Ces descripteurs très simples permettent d'obtenir des informations sur la géométrie des régions de l'image. Il existe d'autres descripteurs de forme, basés sur des statistiques sur les pixels des régions de l'image.

2.5.2. Les moments géométriques.

Les moments géométriques [SHA78] permettent de décrire une forme à l'aide de propriétés statistiques. Ils sont simples à manipuler mais leur temps de calcul est très long.

Formule générale des moments :

$$m_{p,q} = \sum_{p=0}^{m} \sum_{q=0}^{n} x^p y^q f(x,y) \tag{2.53}$$

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

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

$$x_c = \frac{m_{1,0}}{m_{0,0}} \quad \text{et} \quad y_c = \frac{m_{0,1}}{m_{0,0}} \tag{2.54}$$

Il est possible de calculer à partir de ces moments l'ellipse équivalente à l'objet.

Afin de calculer les axes de l'ellipse, il faut ramener les moments d'ordre 2 au centre de gravité :

$$m^g_{2,0} = m_{2,0} - m_{0,0}x_c^2$$
$$m^g_{1,1} = m_{1,1} - m_{0,0}x_c y_c \tag{2.55}$$
$$m^g_{0,2} = m_{0,2} - m_{0,0}y_c^2$$

Puis on détermine l'angle d'inclinaison de l'ellipse α.

$$\alpha = \frac{1}{2} arctan \frac{2\, m^g_{1,1}}{m^g_{2,0} - m^g_{0,2}} \tag{2.56}$$

À partir des moments géométriques, Hu [HU62] a introduit sept invariants aux translations, rotations et changement d'échelle, appelés moments de Hu.

$$M_1 = \mu_{20} + \mu_{02} \tag{2.57}$$

$$M_2 = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \tag{2.58}$$

$$M_3 = (\mu_{30} - 3\mu_{21})^2 + (3\mu_{21} - \mu_{03})^2 \tag{2.59}$$

$$M_4 = (\mu_{30} - \mu_{21})^2 + (\mu_{21} + \mu_{03})^2 \tag{2.60}$$

$$M_5 = (\mu_{30} - 3\mu_{21})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{21})^2 - 3(\mu_{21} + \mu_{03})^2] + (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2]. \tag{2.61}$$

$$M_6 = (\mu_{20} - \mu_{02})[(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{030} + \mu_{21}), \tag{2.62}$$

$$M_7 = (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})] - (\mu_{30} - 3\mu_{21})(\mu_{12} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{12} + \mu_{03})^2 \tag{2.63}$$

.

Les moments de Hu offrent d'excellents descripteurs invariants en translation, rotation et changement d'échelle pour décrire une image. Cependant leur calcul est relativement long et ils sont très sensibles au bruit, ce qui peut s'avérer être un gros inconvénient dans un système de recherche d'images.

2.5.3. Transformée de Hough.

Cette transformation a été proposée par P.V.C, Hough dans un brevet déposé en 1960 [HU62] afin de détecter des alignements à l'aide d'un oscilloscope et de deux caméras vidéo. Inaperçue pendant plusieurs années, elle a reçu quelque publicité après les travaux de Rosenfeld [ROS69] et Duda et Hart [HAR72] au début des années 70, et fait l'objet depuis cette date d'une attention soutenue de la communauté scientifique.

Depuis les années 80 elle a quitté les laboratoires de recherche et trouvé des champs d'applications dans de nombreux domaines industriels.

On trouvera deux études de synthèse des travaux sur la transformée de Hough dans [BAL78] et [SHA78].

Très peu d'attention a été portée dans la littérature à une définition convenable de la transformée de Hough (TH). Il s'ensuit une confusion fréquente sur le type d'opération effectivement réalisé [COH77, THI81], et une prolifération de termes imprécis pour qualifier des modifications mineures (TH étendue, ou généralisée, ou modifiée, etc.).

On regroupe usuellement sous le nom de 'TH' des transformations qui permettent de détecter dans des images la présence de courbes paramétriques appartenant à une famille connue, à partir d'un ensemble de points sélectionnés, appelés points caractéristiques. La TH utilise essentiellement l'information spatiale des points caractéristiques (leur position dans l'image), mais, parfois, tient compte également de l'information contenue dans le signal d'image lui-même (la valeur de la luminance en un point donné). Nous supposerons que ce signal est une fonction scalaire, mais rien ne s'oppose à ce qu'il soit vectoriel; c'est le cas des images en couleur, ou des images multispectrales. Enfin, bien que la plupart des images soient bidimensionnelles, nous pourrons appliquer la TH à des images à trois dimensions (en imagerie médicale par exemple, ou sur des séquences d'images animées), voir à quatre dimensions. Nous désignerons par n la dimension de l'espace de définition de l'image.

Définition.

Soit $\Re^n$ l'espace image, et ξ un ensemble de N points sélectionnés par un prétraitement : $\xi = \{M_i, i = 1, \dots, \bar{N}\} \in \Re^n$ (2.64)

Un point M de $\Re^n$ est repéré par ses coordonnées x.

Soit $\Omega \subset \Re^P$ un espace de paramètres et *F* une famille de courbes dans $\Re^n$ paramétrée par a :

$$F = \{\{x : f(x,a) = 0, x \in \Re^n\}, a \in \Omega \quad (2.65)$$

On appelle transformation de Hough associée à la famille *F* une transformation qui fait correspondre à l'ensemble ξ une fonction *g* définie sur Ω.

Il existe donc de nombreuses transformations de Hough, les deux principales sont les suivantes:

- **Transformation de m à 1 :**

Soit *m* le nombre minimal de points de $\Re^n$ définissant une courbe de *F*. Soit $\xi^{(m)}$ l'ensemble de tous les $m - uples\ issus\ de\ \xi$

$$:\xi^{(m)} = \{M^{(m)}{}_{i1} = \{M_{i2}, \dots \dots \dots, M_{im} : M_{ik} \in \xi\}\} \quad (2.66)$$

$$avec : Card(\xi^{(m)}) = C_N^m \quad (2.67)$$

A tout $m - uples\ M_i^{(m)}\ de\ \xi^{(m)}$ est associée une courbe de *F* de paramètre a_i. Soit C(a) la fonction caractéristique de $\mathbb{R}^P$. La transformation de Hough de m à 1 est définie par : $g(a) = \sum_{M_i^{(m)} c \xi^{(m)}} c(a - a_i)$ (2.68)

- **Transformation de 1 à m :**

Par tout point $M_i\ de\ \mathcal{R}^n$ passent m courbes de *F*, soit A_i l'ensemble des valeurs de ***a*** telles que $f(x_i, a) = 0$: $A_i = \{a_k = f(x_i, a_k\}, A_i = \{a_k = f(x_i, a_k) = 0\}$

:

La Transformation de Hough de 1 à m est définie par :

$$g(a) = \sum_{M_i \in \xi} \sum_{a_k \in A_i} c(a - a_{k)}), \quad (2.69)$$

La TH (Transformation de Hough) de 1 à m conduit, en pratique, à des calculs moins nombreux que la TH de m à 1, car elle évite une recherche combinatoire parmi les points. D'autre part, elle se prête bien à des implantations rapides, par sa structure parallélisable. Pour ces raisons c'est actuellement la TH la plus utilisée, et on lui réserve souvent l'usage du nom générique TH

Algorithme: Transformée de Hough Standard.

1- Choisir une espace discret (ρ, θ).
2. Initialiser un accumulateur $A(\dim(\rho), \dim(\theta))$à $zéro$.
3. Pour chaque pixel :
• Si I (u, v) = 1
– Calculer $\theta = u * \cos(\theta) + v * \sin(\theta)$,
– Incrémenter A $(roun(\rho), round(\theta))$.
4. Recherche le ou les maximums dans la matrice A.
5. Les couples (ρ, θ) sélectionnés caractérisent les lignes de l'image.

Remarque: Transformée de Hough généralisée

L'algorithme présenté ci-dessus permet de détecter les droites dans une image, nous rappelons que le paramétrage cartésien d'une droite $(y = ax + b)$ est généralement maladroit [HAR72], car l'espace $a = \{a, b\}$ est très inhomogène. Un paramétrage polaire$a = \{\rho, \theta\}$ est plus satisfaisant.

A partir de ce principe on peut généraliser avec d'autres types de courbes :

1. Cercles : $(x - x_0)^2 + (y - y_0)^2 = r^2$,
2. Ellipses : $\lceil (x - x_0)^2 / a^2 \rceil + \lceil (y - y_0)^2 / b^2 \rceil = 1$,
3. Quelconques : $y = f(x, a, b, c, \ldots\ldots)$,

Exemple 2-2: Application de la transformée de Hough sur notre image test (grenouille.gif)

Fig.28 : Application de la tranformée de Hough sur une image couleur.

2.5.4. Résumé.

Les formes représentent un descripteur puissant pour décrire les objets contenu dans l'image. Nous avons utilisé La transformée de Hough pour l'extraction des différentes formes dans l'image, les résultats était satisfaisants comme le montre la figure 28.

Chapitre 3.

3. Les Systèmes de Recherche d'Images par le Contenu.

3.1. Introduction.

L'expression « recherche d'images par le contenu » (« Content-Based Image Retrieval, CBIR, en Anglais) remonte aux travaux de Kato en 1992. Son système, ART MUSEUM, permet de retrouver des images d'art par couleurs et contours. Le terme s'est étendu par la suite à tout procédé permettant de rechercher des images selon des descripteurs, pouvant être de type « signal », comme la couleur et la forme, mais également symboliques. Comme le remarquent les auteurs d'un rapport important sur les systèmes de recherche par le contenu [JPEMG99], retrouver des images indexées manuellement par des mots clefs n'est pas de la recherche par le contenu au sens où le terme est généralement compris, même si ces mots clefs décrivent le contenu effectif de l'image.

Les **applications** des systèmes de recherche d'images existants (et donc les collections d'images) sont variées. Elles incluent des applications judiciaires : les services de police possèdent de grandes collections d'indices visuels (visages, empreintes) exploitables par des systèmes de recherche d'images.

Les applications militaires, bien que peu connues du grand public, sont sans doute les plus développées [JPEMG99] : reconnaissance d'engins ennemis *via* images radars, systèmes de guidage, identification de cibles *via* images satellites en sont des exemples connus. Le journalisme et la publicité sont également d'excellentes applications. Les agences de journalisme ou de publicité maintiennent en effet de grosses bases d'images afin d'illustrer leurs articles ou supports publicitaires. Cette communauté rassemble le plus grand nombre d'utilisateurs de recherche par le contenu (davantage pour les vidéos) mais l'aide apportée par ces systèmes n'est absolument pas à la hauteur des espoirs initiaux ([JPEMG99]).

D'autres applications incluent: le diagnostic médical, les systèmes d'information géographique, la gestion d'œuvres d'art, les moteurs de recherche d'images sur Internet et la gestion de photos personnelles. C'est sur ce dernier thème que nous concentrons notre travail. Nous focalisons donc notre étude de l'état de l'art sur les approches dont la cible (utilisateurs, contenu des images) est peu spécialisée. Ces approches, devant nécessairement faire face aux problèmes dont nous avons parlé en introduction (problématique du temps de recherche dans des bases de grandes dimensions et pertinence de la recherche par le contenu), sont par conséquent les plus pertinentes dans notre cadre.

Concevoir un système permettant d'assister des utilisateurs dans leurs tâches de recherche d'images pose des **problèmes** variés. Dans [JPEMG99] les difficultés suivantes sont identifiées :

1. Comprendre les utilisateurs d'images et leurs comportements : de quoi les utilisateurs ont-ils besoin ?

2. Identifier une manière « convenable » de décrire le contenu d'une image.

C'est une tâche rendue difficile par la subjectivité intrinsèque aux images.

3. Extraire des « descripteurs » des images brutes.

4. Pouvoir stocker de manière compacte un grand nombre d'images

5. Comparer requêtes et images stockées de manière à refléter les jugements de similarité humains

6. Accéder efficacement aux images par leur contenu

7. Fournir des interfaces utilisables

A cela, ajoutons la difficulté majeure, dont nous avons parlé dans l'introduction, connue sous le nom de « **Malédiction de la dimension**» [SMEU00].

Il convient donc d'abord de montrer quelles sont les approches existantes ainsi que leurs limitations. Nous commençons par rappeler les composants d'un système de recherche d'images par le contenu, puis nous présentons une taxonomie des systèmes selon leur niveau d'abstraction en donnant systématiquement des exemples.

3.2. Composants d'un CBIR.

Nous décrivons brièvement ici les caractéristiques communes à la plupart des approches : le traitement de la base d'images, les requêtes puis la mise en correspondance et la présentation des résultats. La Figure 29 illustre l'ordonnancement de ces étapes :

Dans un premier temps (2), des descripteurs sont calculés à partir de chaque image de la collection (1), ils peuvent être de type signal ou/et symbolique (le vocabulaire d'indexation).Les données extraites (à présent représentatives du contenu de l'image du point de vue du système) constituent la base d'index (3). Les requêtes de l'utilisateur (4) sont alors transformées afin d'être comparables avec la base d'index (5) ; une mise en correspondance (6) entre la requête transformée et la base d'index permet ensuite de produire le résultat de la requête (7). Il se peut également que le système possède des composantes liées à la personnalisation, comme par exemple l'extraction, le stockage et l'utilisation d'un profil d'utilisateur.

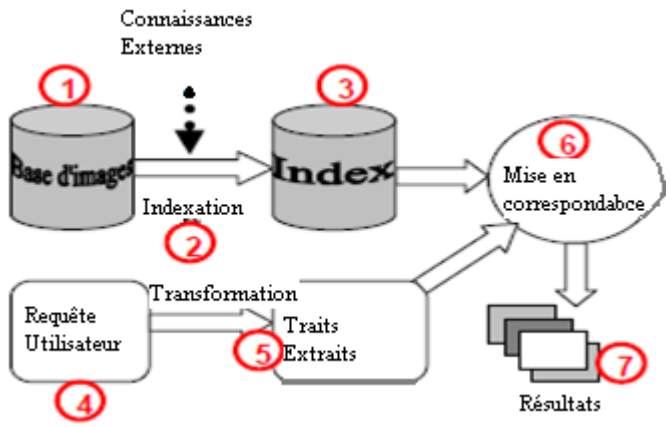


Fig.29 : Principaux composants d'un Système de Recherche par le Contenu.

(1) La base d'image.

La collection (ou base) d'images est la donnée principale du système. Les bases d'images varient d'abord par leur **taille** : la majorité des systèmes est conçue pour des bases de quelques centaines ou milliers d'images ([VAI01, MIN96, LI04A]). Ce nombre peut cependant s'approcher du milliard (880 millions d'images) lorsque la base est constituée par les images collectées par des robots sur Internet. La taille de la base d'image impose des contraintes sur la complexité des traitements effectués sur chaque image. Il en résulte que la comparaison qualitative entre des systèmes travaillant sur des bases de tailles très différentes est peu pertinente.

Le **type d'image** composant la base varie également : des portraits en noir et blanc [LEW96], des peintures chinoises anciennes [LI04A], des images personnelles [VAI01], des images de tissus humains [FBM03], etc. Le type d'image influe fortement sur la conception globale du système, particulièrement sur les descripteurs de bas niveau calculés. D'une manière générale, plus la variabilité intra et inter images est importante, plus le système doit être riche et précis (et plus le problème d'indexer/rechercher ces images est difficile).

Finalement, les collections diffèrent par leur **stabilité**, c'est-à-dire le taux de changements (ajouts d'images, retraits, etc.) en fonction du temps. Faible pour une collection d'images représentant les œuvres d'un peintre ne créant plus, elle peut être très forte lorsque, par exemple, on s'intéresse aux images de la Toile ou à l'actualité.

(2) L'indexation.

L'indexation est l'ensemble des processus aboutissant à la construction d'un index de l'image. Contrairement à d'autres types de données, comme le texte, il n'est pas possible d'utiliser les images *directement* dans un CBIR. Il faut caractériser les images par des informations à la fois discriminantes et invariables à certains paramètres (comme la taille de l'image, l'angle de la prise de vue, etc.). L'indexation peut être **fixe** ([FLI95, SMI96]) : les descripteurs calculés sont toujours les mêmes. L'indexation peut aussi être **évolutive** ([MIN96]) : les descripteurs s'adaptent à l'utilisateur ou au contexte dans le temps, ce qui permet de renforcer l'adéquation système/utilisateur.

L'indexation peut être **générique** (indexation de photographies diverses dans [FLI95]), pouvant caractériser des collections hétérogènes, ou **spécifique** (indexation de peintures chinoises dans [LI04A]), adaptée à un type d'image particulier. Une collection hétérogène est par exemple constituée de photographies personnelles, mettant en scène diverses entités physiques dans des conditions de prise de vue variables. Indexer une telle collection impose l'usage de descripteurs suffisamment génériques (la couleur par exemple), c'est-à-dire qui caractérisent une propriété discriminante applicable à la plupart des entités physiques. A l'inverse, indexer une collection d'images très spécifiques (des empreintes digitales par exemple) requiert l'utilisation de descripteurs également très spécifiques qui, par ailleurs, ne conviendraient probablement pas à une collection hétérogène.

La phase d'indexation peut inclure une étape de **segmentation**, afin de caractériser des régions homogènes de l'image ([SMI96]) ou bien indexer l'image dans sa globalité [VAI01]. La segmentation de l'image précède généralement l'indexation individuelle des régions de l'image et cela permet, outre le fait d'accéder à des parties de l'image, de calculer des descripteurs de 'forme'.

Enfin, l'indexation varie d'un système à l'autre par son **niveau d'abstraction** : extraire des histogrammes de couleurs est une opération directe, alors que reconnaître des personnes ou des objets est beaucoup plus complexe et requiert un apprentissage préalable.

(3) La gestion des index.

Elle concerne la manière dont sont gérés les index des images : stockage et accès. La gestion des index, anecdotique pour une collection de taille modeste, devient une préoccupation essentielle lorsque l'on travaille sur une base de taille conséquente. La manière la plus basique de stocker les index est la liste séquentielle, que ce soit en mémoire ou dans un fichier. Cependant, lorsque le nombre d'images augmente, le temps d'accès à une image augmente linéairement et il est souvent nécessaire d'organiser les index de manière hiérarchique, sous forme d'arbres (organisés selon les descripteurs), ou de tables de 'hash-code' par exemple, afin d'accélérer l'accès à l'information.

(4) Les requêtes.

Le type de requête proposé découle de choix fait en amont, au niveau de l'indexation. Dans des systèmes où seuls des descripteurs de bas niveau sont extraits ([FLI95, SMI96]), les requêtes ne peuvent être que de bas niveau : requête par « image exemple », par croquis ou par manipulation directe des traits de bas niveau. Dans ces systèmes, des descripteurs sont extraits à partir de la requête (une image, un croquis…) et sont comparés aux descripteurs calculés à partir des images de la base (les index des images).

A l'opposé, dans des systèmes proposant plus d'abstraction ([VAI01, TSLA00]), les requêtes peuvent être sémantiques (textuelles par exemple). Par exemple dans [TSLA00], les images sont indexées par des « catégories sémantiques visuelles », ce qui permet à un utilisateur de formuler des requêtes sémantiques (« Je veux des images prises à l'extérieur. »).

(5) Analyse de la requête.

Cette étape a pour but de transformer la requête utilisateur pour la rendre comparable avec les index de la base d'images ; elle consiste donc généralement à extraire les mêmes types de descripteurs que ceux extraits de la base d'image lors de l'indexation.

(6) Mise en correspondance requête / base.

Il s'agit d'estimer dans quelle mesure une image (son index) satisfait une requête donnée. Dans le contexte de la recherche d'images, cela se ramène souvent à calculer la similarité entre les caractéristiques extraites de la requête et les caractéristiques de chaque image dans la base. Cela aboutit généralement à une valeur de correspondance qui caractérise la pertinence (du point de vue du système) d'une image par rapport à la requête. Cette mise en correspondance peut être simple

(comparaison d'histogrammes) ou complexe (comme dans [SMI96] par exemple, avec une mise en correspondance qui tient compte de l'arrangement spatial des régions).

La phase de mise en correspondance peut également inclure une pondération des descripteurs (comme dans [FLI95] où chaque descripteur est pondéré par rapport à son pouvoir discriminant dans la base). Pondérer les descripteurs permet d'éliminer une partie du bruit dans la mesure où les descripteurs les moins pertinents voient leur influence diminuer dans l'évaluation de la similarité requête/image.

La mise en correspondance peut également inclure un bouclage de pertinence. Le but est également d'éliminer le bruit (augmenter la précision) en tentant de converger vers une précision maximale.

(7) La présentation des résultats.

Dans la grande majorité des systèmes disponibles [REM01], le résultat d'une requête est présenté sous la forme d'une liste d'images (réduites à des vignettes) ordonnées par pertinence décroissante. Parfois cette présentation prend d'autres formes, comme par exemple l'œil de poisson (FishEye View) [GUE03]. L'avantage des images par rapport aux documents textuels est qu'il est possible de visionner d'un coup d'œil l'intégralité du document, ce qui permet de visualiser un grand nombre de résultats et de les comparer plus rapidement. Comme indiqué plus haut, la présentation des résultats est souvent couplée avec une possibilité d'interaction, qui permet par exemple de raffiner une requête en indiquant au système les résultats pertinents et ceux qui ne le sont pas (bouclage de pertinence), et de permettre ainsi une reformulation automatique de la requête.

3.3. Représentation des images dans un CBIR.

Dans la majeure partie des systèmes existant, les images sont représentées avec des descripteurs de bas niveau, i.e., en termes de couleur, texture, formes, points d'intérêt ou par des descripteurs hybrides.

3.3.1. Descripteurs de bas niveau.

Dans la vie de tous les jours, nous sommes soumis à une quantité astronomique d'informations brutes, sous forme d'images, de sons, etc. Pour pouvoir utiliser ces informations, il faut réduire de manière drastique cette quantité. C'est là le rôle de la perception : extraire d'une quantité énorme d'information brute un « résumé » utilisable ; et cela consiste essentiellement à trouver des régularités dans les données. Les régularités sont intéressantes car elles permettent de coder, représenter une information brute, de manière concise (particulièrement lorsqu'on ignore les détails). Prenons l'exemple de notre système visuel. Il existe toutes sortes de régularités visuelles et nous possédons un certain nombre de 'détecteurs' pour ces régularités. Il se trouve que les régularités que nous sommes capables de détecter nous permettent d'avoir suffisamment d'information pour identifier la plupart des objets physiques. Dans ce chapitre, nous nous intéressons aux techniques utilisées en informatique pour reproduire notre processus de perception. Nous appellerons ces régularités dans l'information visuelle des descripteurs de bas niveau.

Un descripteur de bas niveau est un ensemble de valeurs extraites directement et qui caractérisent l'image. L'extraction de descripteurs de bas niveau représente une première

abstraction par rapport à l'image brute. Elle constitue la *perception* du système, dans la mesure où les descripteurs extraits sont la seule information conservée. Par opposition aux documents textuels par exemple, où l'extraction de mots porteurs de sens (même s'il est souvent ambigu) est directe. Décrire une image par des descripteurs de bas niveau est un problème difficile. La sémantique « contenue dans les pixels de l'image » n'est absolument pas accessible directement par une machine alors quelle apparaît de manière évidente à un humain qui voit simultanément l'ensemble des pixels. Associer une description à un ensemble de pixels requiert en effet de nombreux processus et surtout, une quantité énorme de connaissances. Par conséquent, les descripteurs extraits d'une image ne sont pas porteurs de sémantique explicite mais tentent de capturer des propriétés visuelles intéressantes de l'image qui constitueront des indices suffisamment discriminants et invariables pour inférer de la sémantique.

Dans [RUI99], les auteurs distinguent deux types de descripteurs de bas niveau : les descripteurs génériques et ceux relatifs à un domaine particulier. Le premier type inclut la couleur, la texture et la forme alors que le second dépend du domaine investigué et peut inclure par exemple les visages, empreintes digitales ou une caractérisation du « coup de pinceau » (comme dans [LI04A]).

Un certain nombre de critères sont à prendre en compte lors de la sélection d'un descripteur. Le but est que ce descripteur soit **discriminant** par rapport aux entités visuelles que l'on cherche à caractériser. Les critères sont les suivants :

- Un descripteur de bas niveau doit être **pertinent** par rapport à un contexte donné. Le contexte repose principalement sur le type d'image traité : un descripteur pertinent pour la classification de visages ne l'est pas nécessairement pour la classification de photographies de peintures.

- Un descripteur de bas niveau pertinent doit être **cohérent** par rapport à l'entité visuelle qu'il caractérise. L'apparence visuelle d'une entité physique varie de manière plus ou moins continue dans un intervalle donné selon son état (par exemple les différentes phases d'un coucher de soleil, les couleurs que revêt un arbre, etc.) et selon plusieurs axes. Un descripteur cohérent est un descripteur qui suit ces variations. Par exemple, un descripteur extrait à partir d'un espace de couleurs organisé en « Luminosité – teinte – saturation » est plus cohérent pour décrire l'apparence visuelle d'un arbre qu'un descripteur extrait d'un espace « Rouge – Vert – Bleu ». La raison est que l'axe « luminosité » suit plus ou moins les variations de luminosité au cours d'une journée, les axes « teinte » et « saturation » suivent à peu près les variations de couleurs au cours des saisons. Par conséquent, de faibles variations au niveau physique se traduisent par de faibles distances dans les caractéristiques calculées.

- Un descripteur de bas niveau, sauf cas particulier, devrait être **invariant** aux variations géométriques et aux variations d'illumination. La raison est que le but de ce descripteur est généralement de caractériser une entité visuelle parmi d'autres, et que celle-ci doit être identifiée quelle que soit la manière dont l'entité apparaît.

- Finalement, l'extraction d'un descripteur à partir d'une image doit être suffisamment **rapide** pour envisager son utilisation à grande échelle.

En résumé nous pouvons dire que dans un contexte donné, pour qu'un descripteur soit **discriminant**, il doit être à la fois **pertinent**, **cohérent** et **invariant**. De plus, dans un cadre interactif, ou si le nombre d'images à traiter est très grand, l'extraction de ces descripteurs devrait

être **rapide**. Nous présentons maintenant les trois grandes catégories de descripteurs de bas niveau, tout en gardant à l'esprit que notre contexte est celui des photos personnelles, donc un contexte général.

3.3.2. Représentation par les couleurs.

La couleur est le descripteur le plus utilisé en CBIR. Un descripteur couleur est généralement indépendant de la taille et l'orientation de la région caractérisée [RUI99].

Cependant, une collection d'images ou même une image seule contiennent généralement énormément de couleurs18 distinctes et il n'est pas envisageable de toutes les considérer indépendamment. Un descripteur couleur repose sur deux choix : la sélection d'un espace de couleur et le choix d'une représentation.

Voyons quels sont les moyens utilisés dans la littérature pour représenter la couleur des images.

Une première possibilité, qui semble naturelle, est de caractériser les couleurs comme nous le faisons chaque jour, par une palette de couleurs (rouge, bleu, jaune, etc.). En plus de réduire drastiquement le nombre de couleurs, cela permet de donner au système une représentation des couleurs similaire à celle des humains. Le système CIRES ([IQB02]) illustre cette méthode. CIRES est basé sur la couleur, l'orientation d'objet et la position relative. Chacune de ces combinaisons a des avantages et des inconvénients et chacune peut être bien adaptée pour certaines bases images, mais ne fonctionnera pas correctement avec d'autres, il se base sur l'espace CIE LAB pour analyser les couleurs. Chaque couleur de l'image est alors associée à l'une des quinze couleurs d'une palette, l'information colorimétrique d'une image est donc représentée par un vecteur de quinze valeurs. Les auteurs justifient ce choix par le fait que la représentation habituelle par histogrammes ne tient pas compte du fait que peu de couleurs sont en fait nécessaires pour la discrimination visuelle et que des valeurs adjacentes dans un histogramme peuvent représenter finalement la « même » couleur.

Une généralisation de l'idée ci-dessus est de représenter l'information couleur sous la forme d'histogrammes. L'histogramme est une représentation efficace du contenu couleur de l'image, particulièrement lorsqu'un motif couleur (image ou région) est unique comparé au reste de la collection ([LON02]). Un histogramme de couleurs est facile à calculer et efficace pour caractériser la distribution locale ou globale des couleurs. De plus, c'est une représentation insensible aux translations et rotations et peu sensible à l'échelle, aux occlusions, ou à l'angle de vue ([LON02]). Ces qualités en font la manière de 8 Sur 50 images (2 272 × 1 704 × 24 BPP) choisies aléatoirement parmi une collection de l'auteur, une image contient en moyenne 250 832 couleurs distinctes.

Les représentations dont nous venons de parler décrivent efficacement la présence des couleurs mais ne tiennent pas compte de leurs corrélations. Les '*color correlograms*' ([HUA97]) ont été proposés pour qualifier non seulement les couleurs mais aussi les corrélations spatiales de chaque paire de couleurs. Ces informations sont représentées sous la forme d'un histogramme à trois dimensions : les deux premières dimensions représentent les combinaisons possibles de paires de pixels, la troisième dimension représente leurs distances spatiales. Par rapport aux autres représentations, les '*color correlograms*' aboutissent aux meilleurs résultats en terme de pouvoir de discrimination, toutefois, leur calcul s'avère coûteux dû au nombre élevé de dimensions ([LON02]).

L'information relative aux couleurs est particulièrement importante dans la caractérisation d'une image. Avant de sélectionner un type de description du contenu couleur, il convient de choisir un espace de couleurs. Si un objet physique évolue (déplacement, différence d'illumination, modifications liées au temps, etc.) alors cette évolution devrait se refléter proportionnellement dans un espace de couleur. C'est entre autres pour cela que nous choisissons de représenter les couleurs dans un espace dont les dimensions sont la teinte, la saturation et la luminosité : l'espace HSV.

3.3.3. Représentation par les textures.

Au même titre que la couleur, la texture est une caractéristique fondamentale des images car elle concerne un élément important de la vision humaine. De nombreuses recherches ont été menées à la fois dans les domaines de l'analyse et de la synthèse de texture. Mais d'après [DEL99] une définition formelle de la texture est quasiment impossible. D'une manière générale, la texture se traduit par un arrangement spatial des pixels que l'intensité ou la couleur seules ne suffisent pas à décrire. Elles peuvent consister en un placement structuré d'éléments mais peuvent aussi n'avoir aucun élément répétitif.

De nombreuses définitions ont été proposées, mais aucune ne convient parfaitement aux différents types de textures rencontrées. Dans une définition couramment citée [POL98], la texture est présentée comme une structure disposant de certaines propriétés spatiales homogènes et invariantes par translation. Cette définition stipule que la texture donne la même impression à l'observateur quelle que soit la position spatiale de la fenêtre à travers laquelle il observe cette texture. Par contre l'échelle d'observation doit être précisée. On peut le faire par exemple en précisant la taille de la fenêtre d'observation.

La notion de texture est liée à trois concepts principaux:

1. un certain ordre local qui se répète dans une région de taille assez grande,
2. cet ordre est défini par un arrangement structuré de ses constituants élémentaires,
3. ces constituants élémentaires représentent des entités uniformes qui se caractérisent par des dimensions semblables dans toute la région considérée.

Il existe un grand nombre de textures. On peut les séparer en deux classes: les textures structurées (macro textures) et les textures aléatoires (micro texturés).

Une texture qualifiée de structurée est constituée par la répétition d'une primitive à intervalle régulier. On peut différencier dans cette classe les textures parfaitement périodiques (carrelage, damier,...), les textures dont la primitives subit des déformations ou des changements d'orientation (mur de briques, grains de café,...). Les textures qualifiées d'aléatoires se distinguent en général par un aspect plus fin (sable, herbe,...). Contrairement aux textures de type structurel, les textures aléatoires ne comportent ni primitive isolable, ni fréquence de répétition. On ne peut donc pas extraire de ces textures une primitive qui se répète dans l'image mais plutôt un vecteur de paramètres statistiques homogènes à chaque texture.

Voyons maintenant à travers quelques exemples comment l'information relative à la texture est utilisée dans les systèmes de recherche d'images.
Les méthodes de caractérisation de la texture peuvent être classées selon la taxonomie suivante ([MIR98]) *:*

Les **méthodes statistiques** cherchent à caractériser des propriétés statistiques basées sur les occurrences de niveaux de gris de l'image. Parmi ces méthodes, la méthode des *matrices de cooccurrences* est sans doute la plus connue. Il s'agit d'une approche statistique de l'étude de la texture. Une matrice de cooccurrence P(i,j) est une matrice dans laquelle l'élément (i,j) décrit la fréquence d'occurrence de deux pixels (pour une distance d et une orientation o). Cette matrice décrit les régularités observables dans les niveaux de gris des pixels d'une région. Généralement, on ne se sert pas directement de la matrice de cooccurrence mais plutôt de valeurs calculées à partir de celle-ci (i.e. la moyenne, le contraste, l'homogénéité, l'entropie ou l'énergie). L'auteur précise que la méthode des matrices de cooccurrence est extrêmement coûteuse en temps de calcul.

Une approche différente est de considérer qu'une texture est un agencement de composants élémentaires. Cette approche, qualifiée de **géométrique** dans [MIR98] n'est pas adaptée au domaine large, particulièrement les scènes naturelles, qui ont des textures desquelles il est difficile d'extraire des composants élémentaires. Nous ne rentrerons donc pas dans des détails supplémentaires. De la même manière, les **méthodes basées sur des modèles**, qui tentent de calculer les paramètres d'un modèle de texture prédéfini, ne semblent pas adaptées aux textures naturelles [MIR98]. Elles sont par contre utilisées pour la génération de textures.

Les **méthodes issues du traitement du signal** semblent plus adaptées pour caractériser des textures naturelles. Les *filtres spatiaux* constituent probablement la technique la plus directe pour capturer les propriétés relatives aux textures de l'image. Les premières approches consistèrent à quantifier la densité de contour (« edge density per unit area ») : les textures fines ont une densité élevée en comparaison aux textures plus grossières. Ces filtres ont pour but d'extraire la fréquence spatiale, qui fait référence à la fréquence de variation des différents tons qui apparaissent dans une image. Les régions d'une image où la texture est « rugueuse » sont les régions où les changements dans les tons sont abrupts; ces régions ont une fréquence spatiale élevée. Les régions « lisses » ont une variation des tons qui est plus graduelle sur plusieurs pixels; ces régions ont une fréquence spatiale faible. La méthode de filtrage spatial consiste à déplacer un filtre d'une dimension de quelques pixels (ex. : 3 sur 3, 5 sur 5, etc.) au-dessus de chaque pixel de l'image. Un filtre peut être uniforme ou réagir à des caractéristiques précises comme les coins, les jonctions, etc. Il est ainsi possible de caractériser grâce à cette méthode de nombreuses caractéristiques de la texture comme sa rugosité ou sa direction alité.

Toujours dans le domaine du traitement du signal, les méthodes issues des *transformées de Fourier* sont sans doute les plus courantes en analyse de texture. L'analyse de Fourier est un outil largement utilisé en physique et en mathématiques. Le principe de la transformée de Fourier repose sur le fait que toute fonction périodique peut être représentée comme la somme d'une série de sinus et de cosinus dont on fait varier d'une part les amplitudes en les multipliant par des coefficients, et d'autre part les phases en les décalant de manière à ce qu'elles s'additionnent ou se compensent. Le problème est que cette représentation fréquentielle de l'image est globale. La caractérisation locale des textures utilise donc le principe des fenêtres de Fourier, qui « glissent » sur l'image en capturant ses propriétés locales. Les techniques dites « de Gabor » utilisent cette méthode avec une fenêtre Gaussienne ([GOS99]). Il est commun d'utiliser un banc de *filtres de Gabor* pour caractériser plus précisément l'information fréquentielle des régions de l'image selon des directions ou des distances particulières. Par leur nature fréquentielle, les méthodes issues des transformées de Fourier sont adaptées pour caractériser les textures régulière et uniformes (comme un grillage, un mur de brique ou un damier) et peu adaptées pour caractériser les textures

aléatoires (un feuillage, une foule, un visage, une écorce, etc.) ou les textures régulières projetées sur des objets non plats ou vues sous un angle particulier.

Une étude très intéressante est menée dans [SIN02] sur les algorithmes de segmentation et les algorithmes d'extraction de texture. Au-delà des expérimentations menées dans leur travail, des faits remarquables sont rapportés.

Le premier de ces faits est qu'il n'existe aucun consensus sur la meilleure technique d'extraction de texture. De plus, les comparaisons entre techniques ne sont guère viables dans la mesure où les conditions expérimentales sont très différentes. Les résultats obtenus dépendent beaucoup des données utilisées pour les tests et celle-ci étant de faible taille, les résultats ne sont pas généralisables.

Un autre point très important est que les techniques d'extraction de texture sont très souvent évaluées sur des collections spécialisées telles que Brodatz, qui n'ont aucun rapport avec les photographies traditionnelles. Les auteurs rapportent qu'ils n'ont rencontré que très rarement des évaluations portant sur de « vraies scènes ». Or, les résultats obtenus sur des collections de textures ne sont pas reportables sur des images naturelles. Par exemple, toujours dans [SIN02], la méthode d'extraction "*Edge Frequency*" se révèle, en dépit de sa simplicité, parmi les plus efficaces pour la description de la texture de scènes naturelles alors que cette même méthode n'obtient que des résultats très modestes dans d'autres travaux validés sur des collections de textures. L'article passe également en revue onze articles qui testent diverses techniques d'extraction de textures. Il en ressort, outre le fait qu'il n'y a pas de consensus, le fait étonnant que les techniques basées sur les filtres de Gabor (très utilisées en CBIR) ne figurent jamais en tête.

En fait il ressort de cette étude que les méthodes simples semblent les plus efficaces pour les images naturelles (édge Frequency', matrices de cooccurrences, filtres spatiaux). Il ressort également que l'ajustement des paramètres est une tâche très difficile et qu'il est très rarement possible de l'automatiser. Enfin, un des points principaux est que l'efficacité d'un extracteur de texture est indissociable de l'algorithme de segmentation qui génère les régions à partir desquelles sont extraites les textures.

Chapitre 4.

4. Mesure de similarité entre descripteurs.

La mesure de similarité quantifie la proximité des images dans l'espace des caractéristiques. Elle est souvent métrique, les images sont considérées ressemblantes si la distance est faible. La complexité de calcul d'une distance doit être raisonnable parce que dans un système CBIR cette tache s'exécute en temps réel. D'autres paramètres entrent en jeu tel la dimension de l'espace caractéristique, la taille de la base, etc.… . La méthode naïve de recherche calcule la distance entre la requête et toutes les images de la base puis les ordonne selon leurs scores. Ceci par conséquent rend le temps de réponse proportionnel au nombre d'images (O(N)). Les méthodes d'indexation du contenu permettent par ailleurs de réduire cette complexité comparée à la recherche séquentielle. Pour résumer, la mesure de similarité vérifie généralement les propriétés :

- La **perception** : Une faible distance dans l'espace caractéristique indique deux images semblables.
- Le **calcul** : La mesure de distance se calcule rapidement pour une faible latence.
- La **scabilité** : Le calcul de distance ne doit pas être affecté par une modification de taille de la base
- La **robustesse** : la mesure devra être robuste aux changements des conditions d'acquisition d'image.

Ci-après les distances les plus utilisées pour comparer des images considérées comme vecteurs ou comme distributions statistiques.

- **Distance de Mahalanobis** :

Cette distance prend en compte la corrélation entre les distributions des classes. Elle est ainsi définie par :$D_M = \sqrt{(f_1 - f_2)^T C^{-1} (f_1 - f_2)}$.

Où *C* est la matrice de covariance. Dans les cas où les dimensions des caractéristiques sont indépendantes, *C* ne comporte que des les variances et la distance de Mahalanobis se simplifie sous la forme :

$$D_M = \frac{\sum (f_1(i) - f_2(i))^2}{c_i} \qquad (4.1)$$

Si *C* est la matrice identité, D_M est la distance euclidienne.

- **Intersection d'histogrammes :**

Cette mesure est l'une des premières distances utilisée dans la recherche d'image par le contenu. Elle a été proposée par Swain et Ballard [SB91] mesurant la partie commune entre deux histogrammes. Etant donné deux histogrammes h_1 et h_2.

$$D_{Intersec} = \frac{\sum_i min(h_1(i), h_2(i))}{\sum_i h_2(i)} \quad (4.2)$$

Deux images présentant une intersection normalisée d'histogrammes proche de 1 sont considérées comme similaires. Cette mesure n'est pas une métrique parce que non symétrique. Cependant il en existe des versions symétriques telles que celle proposée par Smith [SMI97].

- **Earth Mover Distance (EMD) :**

EMD consiste à minimiser le coût de transformation d'une distribution en une autre sous certaines contraintes de déplacement des classes de descripteurs. Elle est exploitée notamment par Rubner et al [RTG98]. EMD requiert une optimisation linéaire.

$$D_{EMD} = \frac{\sum_{ij} g_{ij} d_{ij}}{\sum_{ij} g_{ij}}$$

Où d_{ij} représente la disimilarité entre deux indices (i,j) et g_{ij} est le flot optimal entre deux distributions dont le coût total est :

$$D_{EMD} = \sum_{ij} g_{ij} d_{ij}$$

Le coût est minimisé sous les contraintes suivantes :

$$g_{ij} \geq 0, \forall i, j$$

$$\sum_i g_{ij} \leq f_2(j), \forall j$$

$$\sum_j g_{ij} \leq f_1(i), \forall i$$

$$\sum_i \sum_j g_{ij} = \min(f_1(i), f_2(j))$$

EMD prétend également mimer la vision humaine.

- **Distance de Minkowski :**

La distance de Minkowski est une famille de distances vectorielles. Soit f_1, f_2 deux vecteurs de caractéristiques, elle s'exprime par :

$$d^p(f_1, f_2) = \left(\sum_{i=1}^{n} |f_1(i) - f_2(i)|^p\right)^{1/p} \quad (4.3)$$

p est le facteur de Minkowski et n la dimension de l'espace caractéristique. La distance Euclidienne est un cas particulier de cette distance où p=2, de même que la distance de Manhatan (p=1).

- **Distance quadratique :**

La distance de Minkowski traite les éléments du vecteur de caractéristique d'une manière équitable.

La distance quadratique en revanche favorise les éléments les plus ressemblants. Sa forme générale est donné par :$D_Q = \sqrt{(f_1 - f_2)^T A (f_1 - f_2)}$ où $A = [a_{ij}]$ est la matrice de similarité. a_{ij} Représente la distance entre deux éléments des vecteurs$f_1 et\ f_2$. Hafner et al [HSE95] propose la formule suivante pour construire la matrice A.

$$a_{ij} = 1 - \frac{d_{ij}}{\max(d_{ij})} \tag{4.4}$$

Les propriétés de cette distance la rendraient proche de la perception humaine de la couleur, ce qui en fait une métrique attractive pour les systèmes de Recherche d'images couleur par le contenu .

- **Distance de Bhattacharya :**

La distance de Bhattacharya exploite la séparabilité entre deux distributions gaussiennes représentées par leur covariance Σ:

$$D_B = \frac{1}{8}(\mu_1 - \mu_2)^T \Sigma^{-1} (\mu_1 - \mu_2) + \frac{1}{2} ln \frac{\det(\Sigma)}{\sqrt{\det(\Sigma_1) det(\Sigma_2)}} \tag{4.5}$$

où $\Sigma = 0.5 \times (\Sigma_1 + \Sigma_2)$ La séparabilité entre classes est estimée par la distance des moyennes et des matrices de covariance de chaque classe.

- **Distance de Kullback Leiber (KL) :**

La divergence de Kullback Leiber exprime l'entropie relative de deux distributions :

$$D_{KL} = \sum_i f_1(i) log \frac{f_1(i)}{f_2(i)} \tag{4.6}$$

- **Divergence de Jeffrey (JD) :**

La divergence de Jeffrey est défini par :

$$D_{JD} = \sum_i f_1(i) log \frac{f_1(i)}{\hat{f}_i} + f_2(i) log \frac{f_2(i)}{\hat{f}_i} \quad (4.7)$$

Où $\hat{f}_i = (f_1(i) + f_2(i))/2$. A la différence de la mesure KL. JD est symétrique et plus stable.

- **Distance de Kolmogorov Smirnov** :

Cette distance est appliquée aux distributions cumulées $f^c(i)$:

$$D_{KS} = \max_i |f_1{}^c(i) - f_2{}^c(i)| \quad (4.8)$$

- **Distance de Cramer Von Mises :**

La distance de Cramer Von Mises s'applique également sur des distributions cumulées, elle est définie par :

$$D_{CVM} = \sum_i (f_1^c(i) - f_2^c(i))^2 \quad (4.9)$$

4.1. Techniques de mise en correspondance.

Il s'agit d'apparier deux ensembles de régions appartenant à deux images différentes afin d'elaborer une mesure de similarité entre images. Soit X et Y deux groupes de régions repectivement de l'image requete et cible figure 30. L'ensemble de régions des deux cotés peut etre consideré comme un graphe bipartite dont les nœuds correspondent aux régions et les arcs attribués (pondérés par les distances) relient ces nœuds en couples.

La solution optimale consiste en le graphe qui minimise le cout global de mise en correspondance. Généralement on distingue deux possibilités d'appariement :

1- attribuer un élément de la requête à un seul élément de la cible (1 :1). (Assignement Problem).
2- Attribuer plusieurs éléments de la requête à plusieurs éléments de la cible (N : N)

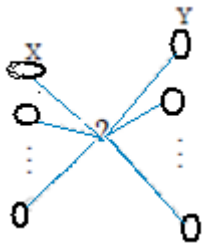


Fig.30 : Problème de mise en correspondance.

Nous présentons dans ce qui suit quelques méthodes utilisées dans la recherche d'images par le contenu par régions avec appariement 1 :1 et N : N.

4.1.1. L'algorithme Hongrois.

Proposée par Kuhn [MEWK99] cette méthode est la plus communément utilisée pour résoudre l'AP (Assignement Problem). Les poids $w(x_i, y_j)$sont réunis dans la matrice D appelée matrice de distance. Soit $G(X, Y, X \times Y)$ un graphe pondéré par les $w(x_i, y_j)$. Supposant que $|X| \leq |Y|$. Trouver pour chaque nœud $x_i \in X$ un nœud $y_j \in Y$ qui contribue à minimiser la distance :

$$D_H = \sum_{i=1}^{|X|} w(x_i, y_j)$$

L'appariement est complet ou incomplet selon que $|X| = |Y|\ ou\ |X| \neq |Y|$. L'algorithme Hongrois autorise d'ajouter une constante aux colonnes ou aux lignes sans que cela n'affecte le coût de la mise en correspondance.

4.1.2. Mariages stables.

L'appariement entre plusieurs entités image peut se ramener à un problème de mariages stables (MS). La méthode également de type 1 :1 est fondée sur les préférences entre régions requête (hommes) et régions (femmes). Chaque entité range par ordre de préférence tous les membres de l'autre image (sexe opposé). La mise en correspondance est dite instable si x_i est marié à y_j et x_k avec $y_{p,}$ alors que x_i préfère y_p à y_j et y_p préfère x_i à x_k. Ceci génére des pairs bloquantes. Gale et Shapley [GS62] ont proposé un algorithme linéaire en le nombre de couples. Ils ont montré qu'il existe toujours une mise en correspondance stable dans n'importe quelle instance. Dans de nombreux cas les listes de préférence peuvent être incomplètes soit à cause d'incompatibilités ou parce que les populations sont de taille différente. Une autre extension dite SMTI (Stable Marriage with Ties and Incomplete lists) accepte des différences encore des paires ex aequo. Ces types de MS peuvent être résolus de la même manière que MS standard.

Liste. Homme	Liste. Femme
$x_1 : y_3, y_2, y_6, \dots$	$y_1 : x_4, x_1, x_3, \dots$
$x_2 : y_2, y_4, y_1, \dots$	$y_2 : x_2, x_9, x_7, \dots$
$x_3 : y_3, y_7, y_5, \dots$	$y_3 : x_2, x_8, x_4, \dots$
$x_4 : y_9, y_1, y_6, \dots$	$y_4 : x_3, x_1, x_5, \dots$

Tableau 5 : Exemple de liste de préférence.

MS a eu de nombreux developpements théoriques et s'emploie dans de nombreuses applications, de l'elaboration de l'emploi du temps à la gestion de stock en passant par le problème du voyageur de commerce. La contrainte de stabilité ajoutée aux préférences peut faire aboutir à ce que tous les couples soient malheureux d'où une satisfaction globale faible.

4.1.3. Graphes attribués (Attributed Relationnal Graph ARG).

Dans ce cas le modele objet / scène est representé par un graphe attribué (ARG). Les nœuds codent les caracteristiques visuelles et les arcs spécifient les relations contextuelles entre nœuds. Un exemple de relation est le couple orientation-distance entre objets déjà mentionné. Le contenu est donc résumé dans cette structure et la mesure de similarité se ramene à une mise en correspondance d'ARG. L'appariemnet des graphes attribués est bien cerné dans la littérature avec différentes approches [BF84, AD93, MB98, PF97], mais reste un problème ouvert. Petrakis [PET02] compare et évalue plusieurs méthodes de similarité spatiale par ARG, algorithme hongrois et 2D string. L'auteur conclue que la meilleure méthode est celle d'ARG suivie par l'algorithme hongrois puis 2D string.

4.1.4. Appariement par région integrée (Integrated Region Matching IRM).

L'appariement par région integré IRM, a été introduit par Li et al [LWW00] est une méthode de type (N :N), la figure 26 en represente un exemple. Le but ici est de compenser les imperfections de segmentation ce qui est souhaitable pour tout système CBIR. IRM définit une matrice de signification de taille $|X| \times |Y|$dont chaque élément est une pondération du couple de régions requête et cible $s(x_i, y_j)$. Cette pondération contrôle le coût d'appariement de ces régions dans la distance globale. Elle dépend de la distance; plus celle-ci est petite plus la contribution est grande.

$$D_W(x_i, y_j) = \sum_{ij} s(x_i, y_j) d(x_i, y_j)$$

où $d(x_i, y_j)$ est la distance entre deux segments de l'image.

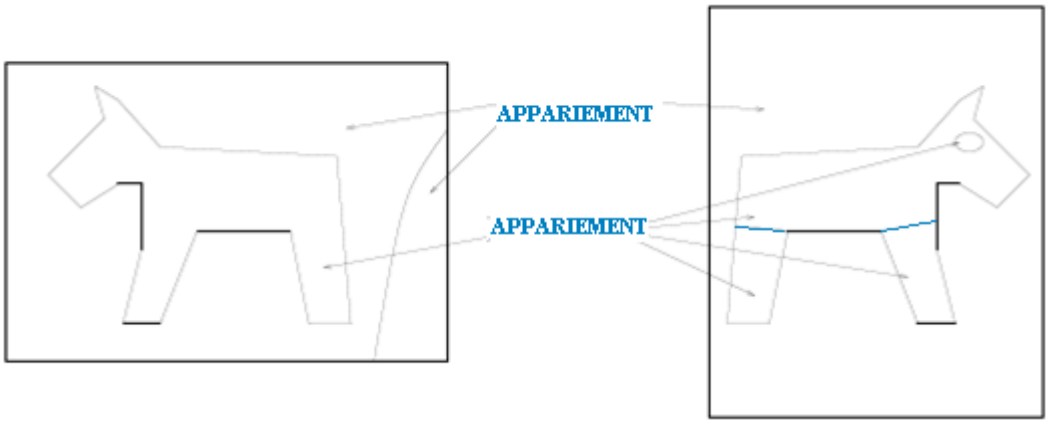


Fig.31 : Exemple d'appariement IRM.

4.2. Conclusion.

Le choix des descripteurs pour un système de recherche d'images par contenu est important, dans le sens où, ce choix influe sur les résultats attendus. Cependant, d'une part il n'y a pas de descripteurs universels, et d'autre part le choix du descripteur SIFT combiné à un bon descripteur de texture semble donner une meilleure description de l'image qui est invariant aux transformations d'images qu'on peut avoir sur la base.

Une autre problématique, est que ces descripteurs de bas niveaux (décrivent des caractéristiques de bas niveaux) sont très couteux en termes de temps de calcul et rend leurs utilisations très fastidieuses.

Nous pensons qu'un système se basant uniquement sur ces descripteurs, n'a aucune chance d'être performant, dans le sens que le calcul de ces derniers reste très gourmand en termes de temps processeur.

Nous pensons qu'un système de recherche d’image adapté aux besoins des utilisateurs doit être très rapide, c’est-à-dire capable d’extraire des images similaires avec une forte précision dans des temps relativement acceptable dans le cadre des normes régissant l’IHM.

Dans le prochain chapitre, nous présenterons Les techniques d'appariement des graphes ainsi que les mesures de similarités sur graphes. Nous pensons en effet, que la représentation des images par des graphes serait un moyen efficace pour permettre aux systèmes d'indexation des images un degré d’appariement satisfaisant et un moyen d’intégration de l’information spatiale dans la modélisation.

Chapitre 5.

5. Méthodes d'appariement des graphes.

5.1. Introduction.

En effet, la représentation sous forme de graphe est utilisée dans différentes applications d'analyse d'images. Dans ces applications, le graphe est utilisé pour représenter les objets et les relations spatiales entre objets. Cette représentation structurelle a prouvé sa flexibilité dans plusieurs domaines [DBLO05].

Nous présentons dans cette section des travaux sur la représentation sous forme de graphe. Ensuite nous détaillerons quelques travaux sur les mesures de similarité entre les graphes.

5.2. Notions de graphes.

5.2.1. Définitions.

Un graphe fini G = (V, E) est défini par l'ensemble fini $V = \{v_1, v_2, \dots, v_n\}$ $(|V| = n)$ dont les éléments sont appelés sommets, et par l'ensemble fini $E = \{e_1, e_2, \dots, e_m\}$ $(|E| = m)$ dont les éléments sont appelés arcs.

Un arc e de l'ensemble E est défini par une paire non-ordonnée de sommets, appelés les extrémités de e. Si un arc e relie les sommets a et b, on dira que ces sommets sont adjacents, ou incidents avec e, ou encore que l'arc e est incident avec les sommets a et b. On appelle ordre d'un graphe le nombre de sommets (n) de ce graphe.

5.2.2. Appariements de graphes.

Afin de comparer deux graphes, il est nécessaire de mettre en correspondance leurs sommets, c'est à dire de définir un appariement (« matching »).

Étant donné deux graphes G = (V, E) et G'= (V', E'), un appariement *m* entre G et G' est une relation entre V et V', c'est à dire $m \subseteq V \times V'$. Sans perte de généralité, il est supposé que les ensembles des sommets de G et de G' sont disjoints ($V \cap V' = \emptyset$). Quand aucune contrainte n'est ajoutée à l'appariement, c'est à dire que chaque sommet peut être apparié à zéro, un, ou plusieurs sommet(s) de l'autre graphe, l'appariement est dit multivoque. Cependant, il est possible d'ajouter des contraintes sur le nombre de sommets avec lesquels il peut être apparié. Cela permet de définir les appariements fonctionnels, applicatifs, univoques, injectifs et bijectifs.

Étant donnés deux graphes G = (V,E) et G'=(V',E'), un appariement $m \in V \times V'$ est :

- Fonctionnel de G dans G' si *m* apparie chaque sommet du graphe G avec au plus un sommet du graphe G' : $\forall v \in V, \| m(v) \| \leq 1$

- Applicatif de G dans G' si *m* apparie chaque sommet du graphe G avec exactement un sommet du graphe G' : $\forall$ v$\in$V, $\| m(v) \|=1$

- Univoque entre G et G' si *m* apparie chaque sommet du graphe G et G' avec au plus un sommet : $\forall$ v$\in$V, $\| m(v) \| \leq 1$ $\wedge$ $\forall$ v'$\in$V', $\| m (v') \| \leq 1$

- Injectif de G dans G' si *m* apparie chaque sommet du graphe G avec un sommet différent de G' : $\forall$ v$\in$V, $\| m(v) \| =1$ $\wedge$ $\forall$ v'$\in$V', $\| m (v') \| \leq 1$

- Bijectif entre G et G' si *m* apparie chaque sommet du graphe G (respectivement de G') avec un sommet différent de G'(respectivement de G) : $\forall$ v$\in$V, $\| m(v) \| =1$ $\wedge$ $\forall$ v'$\in$V', $\| m (v') \| = 1$

Arcs appariés : Soit *m* un appariement des sommets de deux graphes G =(V,E) et G'=(V',E'), un arc (u,v) $\in$ E est dit apparié à un autre arc (u',v') $\in$ E' si et seulement si {(u,u'), (v,v')} $\subseteq$ *m*.

5.2.3. Représentation par des graphes.

Les images sont généralement segmentées en régions. Chaque sommet du graphe représente alors une région de l'image et les arcs représentent les différentes relations possibles entre les régions. L'intérêt principal de l'utilisation des graphes pour la représentation d'images est l'intégration de l'information spatiale dans la modélisation. En effet les représentations classiques telles que les histogrammes de couleurs, les descripteurs de textures, etc. ne donnent aucune information sur la façon dont les régions d'intérêt de l'image sont agencées. Au contraire, la représentation par les graphes permet de décrire la structure de l'image comme la façon dont les régions sont disposées les unes par rapport aux autres (*Figure 32*). En outre, selon les types de relations choisies, cette représentation en graphe peut être invariante à certaines transformations telles que les rotations de l'image ou les translations de certaines parties de l'image.

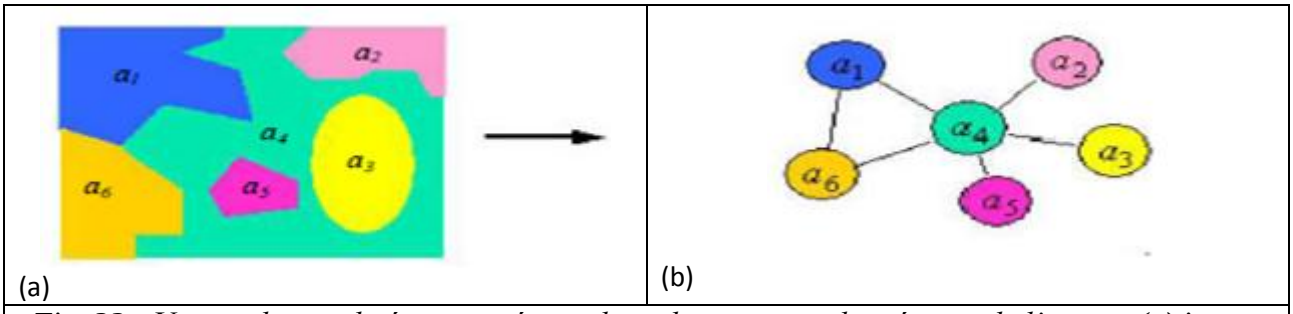


Fig. 32 : Un graphe attribué qui représente les relations entre les régions de l'image. (a) image qui contient un groupe de régions (b) le graphe qui représente (a)

La plupart des travaux qui utilisent les graphes comme modèle de représentation, ont pour objectif la définition de la mesure de similarité entre les documents pour les comparer. La section suivante présente des mesures de similarités pour comparer deux graphes.

5.3. Mesure de similarité des graphes.

La structure des graphes est caractérisée principalement par la flexibilité, l'universalité et l'utilisation dans des domaines d'applications variés. Ce fait a conduit au développement de

plusieurs mesures de similarité pour les graphes qui sont optimisées pour différentes applications. La plupart de ces mesures ont une caractéristique commune qui consiste en quelque sorte à utiliser des opérations d'édition (*edit operations*). Le principe est de définir l'effort nécessaire pour rendre les graphes identiques. Dans la suite, nous détaillerons les mesures les plus connues dans la littérature.

5.3.1. Distance d'éditions.

La distance d'édition pour les graphes est l'extension de la distance d'édition pour les chaînes des caractères (*ou distance de Levenshtein*) [RWMF74]. C'est une mesure très commune pour évaluer la similarité entre les graphes. La distance d'édition entre deux graphes est le nombre minimum des opérations d'édition qui sont nécessaires pour transformer un graphe en un autre. Les principales opérations d'édition sont, la suppression ou l'insertion des sommets ou arcs, et le changement des attributs des sommets ou des arcs.

Pour calculer la distance entre deux graphes Robles-Kelly [AREH05] propose une méthode de conversion d'un graphe à une séquence de chaînes de caractères afin d'appliquer des techniques de mesure de la distance d'édition de chaîne de caractères. La problématique principale de la distance d'édition est la détermination du coût minimal des opérations d'éditions effectuées. Les solutions proposées à cette problématique sont dans la majorité probabiliste [AREH05], [MNHB04].

5.3.2. Mesure de Papadopoulos et Manolopoulos.

Papadopoulos et Manolopoulos [NPYM99] présentent une mesure de similarité pour les graphes, qui est aussi basée sur le concept des opérations d'édition. Ils proposent trois opérations primitives différentes, qui sont l'insertion de sommet, la suppression de sommet et la mise à jour de sommet. Pendant que les opérations de suppression et d'insertion ont des significations évidentes, l'opération de mise à jour est nécessaire pour insérer ou supprimer les arcs pour un sommet. En plus, Papadopoulos et Manolopoulos introduisent la notion de la séquence de degrés du sommet d'un graphe ordonnée en ordre décroissant. La distance de similarité entre deux graphes est définie comme le nombre minimum des opérations primitives qui sont nécessaires pour que les deux graphes possèdent la même séquence de degrés des sommets.

Pour calculer la mesure de similarité, un histogramme trié de graphe est introduit dans [NPYM99]. Soit un graphe G = (V, E), son histogramme est construit par le calcul des degrés de chaque sommet de G. Chaque sommet correspond à une case différente de l'histogramme. En triant ces cases, on obtient un histogramme trié de graphe G (sorted graph histogram). Les auteurs montrent que la L1-distance entre deux histogrammes triés de graphes est égale, à la distance de similarité entre les deux graphes correspondants. En plus, cette distance satisfait les trois propriétés fondamentales de l'espace métrique :

- la positivité : $0 \leq$ distance (G1, G2)

- la symétrie : distance (G1, G2) = distance (G2, G1)

- l'inégalité triangulaire : distance (G1, G3) $\leq$ distance (G1,G2) + distance (G2,G3)

Évidemment, si deux graphes n'ont pas le même ordre, les histogrammes triés des ces graphes auront des dimensions différentes. Pour mesurer la similarité dans ce cas, une technique de pliage a été introduite pour achever à une dimension constante des histogrammes de chaque graphe. Pour créer un histogramme plié à partir d'un histogramme trié, l'ordre maximum de tous les graphes doit être connu à l'avance. Au début, la dimension désirée *n* est fixée, et pour les graphes avec un nombre de sommets inférieur à *n*, la dimension de leurs histogrammes triés est ramenée à la dimension *n*. Dans tout les autres cas, un nouvel histogramme plié Hp sera construit par échantillonnage de la somme des n-premières valeurs de Ht. Après la résolution du problème de la dimension, les auteurs utilisent la technique d'indexation R-tree pour exploiter leur mesure.

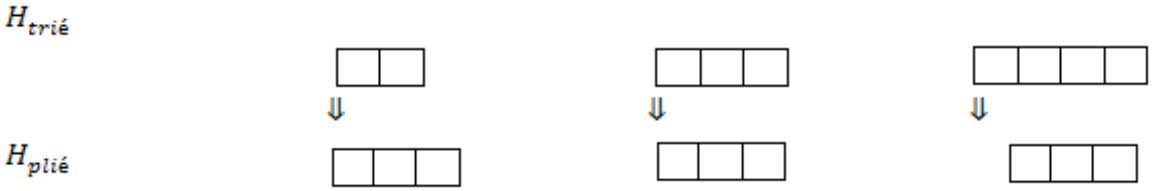


Fig.33 : Technique de pliage d'histogramme de Papadopoulos et Manolopoulos

5.3.3. Mesure de similarité basée sur le plus grand sous-graphe commun.

Cette méthode de mesure de similarité entre deux graphes proposée par Bunke et Shearer dans [HBKS98] est basée sur le plus grand sous-graphe commun de deux graphes. Un graphe G est appelé un sous-graphe commun aux deux graphes G1 et G2, s'il est un sous graphe de G1 et G2. Un sous-graphe commun G de deux graphes G1 et G2 est dit maximal (le plus grand) s'il n'existe aucun autre sous-graphe commun de G1 et G2 avec un ordre (taille, en terme de nombre de sommets) plus grand que celui de G. Dans [HBKS98], le plus grand sous-graphe commun est noté « *mcs* » (*maximal common subgraph*). La mesure de similarité entre deux graphes non-vides G1 et G2 est définit par :

$$d_{mcs}(G1,G2) = 1 - \frac{|mcs(G1,G2)|}{\max\{|G1|,|G2|\}} \qquad (5.1)$$

Où |mcs(G1,G2)| est le nombre des sommets du *mcs* et | G1 | et | G2| sont respectivement le nombre de sommets de G1 et G2. Cette mesure selon [NPYM99] respecte les propriétés suivantes :

* $0 \leq d_{mcs}(G1,G2) \leq 1.$
* $d_{mcs}(G1,G2) = 0 \leftrightarrow G1\ et\ G2\ sont\ isomorphes\ l'unde\ l'autre.$
* $d_{mcs}(G1,G2) = d_{mcs}(G2,G1)$
* $d_{mcs}(G1,G3) \leq d_{mcs}(G1,G2) + d_{mcs}(G2,G23$

5.3.4. Mesure de similarité Ó distance.

Une autre mesure de similarité pour les graphes est proposée par Chartrand, Kubicki et Schultz dans [CKS98]. Cette mesure est basée sur des fonctions entre les ensembles des sommets

des graphes. Elle est définit pour les graphes connexes de même ordre. Soient deux graphes connexes $G1 = (V1, E1)$ et $G2 = (V2, E2)$ ayant le même ordre n, et l'application linéaire ∅: V1 ↦V2.

La ∅-distance entre G1 et G2 est définie par :

$$D_{\phi}(G1, G2) = \sum_{u;v} |lp(u, v) - lp(\phi u, \phi v| \quad (5.2)$$

Où *lp(u,v)* est la longueur du plus court chemin entre u et v dans G1, et la somme est prise pour toutes les paires des sommets u et v distincts dans G1. La mesure de similarité entre deux graphes connexes G1 et G2 avec le même ordre est :

$$D(G1, G2) = \min \{D_{\phi}(G1, G2)\} \quad (5.3)$$

Le minimum est pris parmi toutes les applications ∅ possibles de V1 à V2.

Les auteurs Chartrand et al. Montrent que leur mesure de similarité respectent les propriétés des espaces métriques, mais ils ne proposent aucun algorithme pour calculer cette mesure.

5.3.5. Notion de graphe Médian.

Étant donné un ensemble d'éléments, le médian peut être un concept très utile pour avoir une représentation qui accumule une information globale de l'ensemble. Dans le domaine de l'analyse d'image par approches structurelles, le graphe médian, introduit par Jiang et Bunke dans [XJMB01, HBUN01], a pour objectif l'extraction de l'information essentielle à partir d'un ensemble de graphes dans un seul prototype (c'est à dire modèle).

Soit S un ensemble de graphes, le médian est défini comme le graphe dérivé de S qui a la plus petite somme de distances à tous les graphes dans l'ensemble S. Si la condition que le médian appartienne à l'ensemble des graphes est requise, le médian de l'ensemble est obtenu. Sinon, un graphe médian généralisant est obtenu.

Soit U l'ensemble contenant tout les graphes qui peuvent être construit à partir d'un ensemble $S = \{g_1, g_2, \dots, g_n\}$ (c'est à dire tout graphe avec un nombre de sommet entre 1 et la somme de tous les sommets de g_i. Le graphe médian généralisant et le graphe médian de sont respectivement définis par :

$$\bar{g} = argmin_{g \in U} \sum_{i=1}^{n} d(g, g_i) \quad (5.4)$$

$$g = argmin_{g \in S} \sum_{i=1}^{n} d(g, g_i) \quad (5.5)$$

Avec $d(.,.)$ une fonction de distance qui mesure la dissimilarité entre deux graphes. Le concept de graphe médian généralisant est plus intéressant parce qu'il a un grand potentiel de capter l'information essentielle de l'ensemble des graphes. Mais il est aussi plus complexe à calculer, en fait sa complexité grandie exponentiellement en fonction du nombre de graphes et de leurs tailles. Par contre, le graphe médian peut être calculé en$O(n^2)$ étapes (pour chaque étape, nous aurons besoin de calculer la distance entre deux graphes).

Même si le problème de graphe médian généralisant est récent, la littérature dans le domaine de l'analyse d'image et de la reconnaissance de forme par approches structurelles, présente quelques travaux qui proposent différentes méthodes pour calculer le graphe médian généralisant. Par exemple, Jiang [XJMB99] utilise un algorithme génétique avec représentation chromosomique. Hlaoui [AHSW03] propose une solution basée sur la décomposition de problème de la minimisation de la somme des distances en deux parties ; une pour la somme des distances dédiée aux sommets et l'autre pour les arcs. Encore, la théorie de graphes spectrale [FSS05] ainsi que la programmation linéaire [MSPXZB07] ont été utilisées pour cette problématique.

5.4. Conclusion.

Dans cette section nous avons passé en revue des différentes méthodes d'appariement de graphes. Différentes mesures de similarité ont été proposées dont nous avons cité une liste non-exhaustive. Ces travaux ont été choisis puisqu’ils gardent la structure générale des graphes. Mais les mesures de similarité sont très générales et très complexes. Il y a d’autres travaux essayant de faire l’appariement en transformant un graphe en arbre [GZJ06, AA07, CZ08]. Ces travaux ne sont pas présentés dans notre état de l’art puisque la transformation d’un graphe en arbre implique une grande perte d’information et donc une perte de pertinence dans notre systeme.

Chapitre 6.

6. Les modèles de recherche d'information visuelle.

Le traitement d'une requête dans un système de recherche de la manière dont on lui présente l'information. Un modèle de recherche spécifie le mode de représentation de la requête. Nous listons ci-dessous les plus utilisés.

6.1. Les différents types de requêtes.

6.1.1. Requête par l'exemple.

Dans ce cas le système a besoin de comparer un exemple de même type (image) avec la base pour produire les documents similaires. Cette méthode est simple naturelle et ne nécessite pas de connaissance approfondies pour manipuler le système. Elle est donc bien adaptée à un utilisateur non spécialiste.

6.1.2. Requête par crayonnage (Sketch).

L'usager spécifie un exemple par un dessin pour lequel il cherche des documents similaires. Outre la forme des contours le dessin peut inclure la couleur et la texture. Cette modalité entraine plus d'interaction avec l'utilisateur.

6.1.3. Requête par caractéristique.

L'utilisateur indique la ou les caractéristiques qu'il veut utiliser pour trouver les images similaires, par exemple trouver les images contenant 25% de rouge et 30% de jaune. Ces caractéristiques sont répertoriées dans un vocabulaire compilé en outils de traitement.

6.1.4. Requête exemple et texte.

Cette méthode consiste à renforcer l'image requête en lui associant du texte afin d'accéder à un niveau d'abstraction supérieur. Les images sont organisées et indexés en groupes de pertinence. Chaque groupe définit un type ou un domaine dont l'image relève, par exemple végétation ou animale, etc. Cette approche permet une sémantique accrue mais elle est plus biaisée voire moins générale que les précédentes en ce que la désignation des groupes de pertinence est une tache à fort caractère subjectif ou expert.

6.2. Méthodes Géométriques.

Elles sont dédiées à l'analyse structurelle sur l'identification d'un ou plusieurs motifs et de leur répartition spatiale. Le résultat dépend donc des propriétés géométriques des primitives et des contraintes spatiales sur elles. Le modèle de texture s'exprime volontiers alors sous forme de règles de placements appliqués à ces primitives. Parmi les méthodes les plus connues dans ce contexte on trouve le maillage de voronoi. Chaque cellule de ce maillage produit une primitive et

le regroupement des cellules semblables permet de définir une texture. Zuker [ZUC76] suppose que la texture est une version distordue d'une forme ou d'un arrangement idéale. Pour cela la structure idéale est considérée comme maillage représenté par un graphe isomorphe dont on mesure la compatibilité avec la texture observée. Voorhees et Poggio [VP87] considèrent les régions élémentaires comme primitives liées à la perception humaine. Ils utilisent le laplacien d'une gaussienne à différente échelles pour extraire ces primitives. Avec le même principe Blostein et Ahuja [BA89] les intègrent dans l'estimation de surface des formes. Une méthode plus ancienne considérait la texture comme un arrangement régulier d'éléments suivant une loi spatiale donnée [FU82]. Cette loi est déterminée par des règles grammaticales où la texture est vue comme une chaine dans un langage dont les symboles sont les primitives.

6.3. Méthodes basées sur un modèle.

Le but de ces méthodes est d'obtenir un modèle générateur de la texture [CJ83]. Les paramètres de ce modèle permettent de caractériser ou synthétiser une texture. On peut citer trois types de méthodes : **Les champs de Markov (MRF**), les fractales et le modèle autorégressif éventuellement en multi résolution.

Les modèles MRF traitent l'image comme une réalisation d'un champ aléatoire dans un contexte de voisinage. Ils considèrent que l'intensité en chaque pixel de l'image dépend seulement des intensités des pixels voisins. Le modèle le plus souvent utilisé est le champ de Markov gaussien (Gaussian Markov Random Field (GMRF) [CC85, CJ93]. Krishnamachari et Chellappa [KC97] utilisent GMRF pour modéliser l'image à chaque résolution avec l'hypothèse que les variables aléatoires sont indépendantes par rapport à la résolution. Comer et al. [CD99] utilisent un modèle gaussien autorégressif avec le même principe, mais ils prennent en considération la corrélation entre deux niveaux adjacents. Les auteurs définissent deux modèles markoviens gaussien, l'estimation des paramètres de ces modèles est basée sur des matrices de probabilités conditionnelles.

Le modèle AR (Autoregressive models) ou 'Simultaneous Autoregressive Models (SAR) est l'approche la plus utilisée dans la modélisation de la reconnaissance de la texture [CK85, MJ92]. L'un des avantages des modèles AR est leur faible complexité en termes de paramétrage par rapport au MRF.

Plusieurs surfaces naturelles ont un aspect à la fois statique et génératif. Mandelbrot [MAN82] a introduit les propriétés fractales en se basant sur cet aspect. Pentland a montré que la plupart des surfaces naturelles peut être modélisée comme des fractales spatiales isotropes. La dimension fractale mesure la rugosité de la surface ce qui la rend pratique pour la modélisation et l'analyse de la texture [TJ74]. Parmi les travaux dont le but est de caractériser les textures par la dimension fractales citons [PNHA84, PEN84, KBT01].

6.4. Méthodes Fréquentielles.

Elles sont plus spécifiques du traitement de signal et consistent à analyser la texture dans le spectre où la répétition des motifs se traduit par les fréquences spécifiques. Nous distinguerons trois types de filtrage : Spatial, Gabor et Ondelettes. Ces deux derniers ont montré de bonnes performances en analyse et description de la texture [DHW94, DHW95, UNS95].Les filtres de Gabor sont particulièrement efficaces pour la classification [TUR86, CB87]. Chen et Chen [CC96] en proposent une version modifiée en vue de l'analyse multi résolution. Les travaux de Haley et Manjunath [HM99] portent également sur les filtres de Gabor associés à des ondelettes.

L'avantage de leur méthode est l'invariance en rotation. Smith et Chang [SC94, SC96] utilisent la moyenne et la variance des sous bandes d'ondelettes. La transformée en ondelettes a été combinée aussi avec d'autres caractéristiques pour obtenir de meilleures performances. Thyagarjan et al. [TNP94] et Kundu et al. [KC92] lui associent la matrice de cooccurrence pour extraire à la fois les propriétés statistiques et fréquentielles. Pour ce qui est de filtrage plus classique. Jenigan et D'Astous [JD84] proposent une mesure de l'entropie du spectre de puissance normalisé comme caractéristique d'une région texturée. Bovik et al. [BCG90] codent les images sur des canaux selon la fréquence spatiale et l'orientation. Malik et Perona [MP90] préfèrent le filtrage spatial, l'image étant convoluée avec un banc de filtres symétriques d'inspiration physiologique. Après élimination des réponses locales non pertinentes on peut détecter les frontières entre les différentes textures. Une autre méthode développée par He et Wang [HW92] considère la texture comme un ensemble d'unités. Les caractéristiques sont obtenues à partir du spectre de puissance de la distribution fréquentielle de ces unités.

6.5. Comparaison et évaluation.

L'abondante littérature sur la texture nécessite des méthodes de comparaison pour choisir parmi toutes ces techniques. Weszka et al [WDR76] s'intéressent aux images aériennes et comparent les performances sur les surfaces des terrains. Les caractéristiques utilisées sont de type : statistique en niveaux de gris et puissance spectrale dans le domaine de Fourier. Selon les expériences effectuées par ces auteurs cette dernière est la moins fiable, les autres méthodes sont comparables. Faugeras et Pratt [FP80] décrivent la plupart des techniques en vue de la synthèse aussi bien que de la classification et donnent des figures de mérite dans les deux cas. Conners et Harlow [CH80] présentent une étude comparative de quatre différentes méthodes SGLDM (Spatial Grey Level Dependance Method) GLDM (Grey Level Difference Method) GLRLM (Grey Level Run Length Method).

6.6. Raisonnement spatiale.

Les travaux sur la description d'image présentés jusque là se focalisent sur les caractéristiques de bas niveaux. Il en existe qui tentent une description de plus haut niveau afin de réduire le fossé sémantique entre l'humain et la machine. Nous décrivons ci-après les deux plus importants.

- **2D string et ses variantes.**

Chang et al [CSY87] proposent les 2d string, pour une description spatiale contenu par des chaines de symboles. Les objets dans l'image sont englobés par des rectangles dénommés minimum bounding rectangle (MBR).

- **$\theta R\ string$ et ses variantes.**

Guldivada [GUD98] propose une description fondée sur les orientations relatives des objets dans l'image, $\theta R\ string$ ordonne les objets dans une chaine selon l'angle formé par la ligne qui relie les deux centres de gravités et l'axe horizontal.

6.7. Caractérisation de haut niveau.

Les techniques d’apprentissage supervisé ou non peuvent servir à piloter un système CBIR. Elles ont pour tâche de favoriser des regroupements de valeurs de caractéristiques pour faire émerger de nouvelles variables plus abstraites et plus stables. On trouve parmi elles les réseaux de neurones, les algorithmes génétiques ou le clustering, et c’est d’ailleurs ces dernières i.e. le clustering que nous avons adopté dans notre nouveau système.

Chapitre 7.

7. Nouvelle approche spectrale pour CBIR.

Dans ce chapitre, nous présentons une nouvelle technique de recherche d'images par le contenu. Contrairement à la plupart des techniques existante, notre technique se propose de résoudre le problème des CBIR relatif à la taille des bases d'images sans cesse grandissante et donc au temps de réponse du moteur de recherche par l'application de la réduction non linéaire de la dimension par l'approche noyau (spectral clustering) sur graphe avec l'utilisation d'un descripteur à la fois robuste et invariant aux transformations.

Notre descripteur est hybride et fait associé à la fois le descripteur SIFT et un descripteur de texture à base de transformée d'ondelettes dont les caractéristiques de ces derniers ont été décrit précédemment.

7.1. Méthodes de clustering.

De nombreuses méthodes de regroupement ("clustering") ont été développées dans le passé, l'objectif étant de construire des classes d'objets "similaires" entre eux à partir des données.

On vient de définir la notion de distance qui caractérise la "similarité" entre deux objets, on va à présent développer les méthodes employées pour construire ces groupes d'objets, à savoir des familles d'images similaires.

Les méthodes employées ne prendront en compte aucune connaissance à priori ; par conséquent, on effectuera des classifications non supervisées des données.

Les questions qui restent en suspens sont :

- Existe-t-il des groupes de points parmi les champs de données ?
- Combien de groupes arrivons-nous à former ?
- Ont-ils une signification ?

7.2. Graphe de représentation minimale (MST).

La méthode qui est développée dans cette partie est une méthode de segmentation de données qui a fait ces preuves dans le passe dans le contexte astrophysique [OLIV05]. Le principal avantage de MST (Minimum Spanning Tree) est de passer d'un problème de segmentation multidimensionnelle de données à un problème de partitionnement de graphe.

Tout d'abord, on définit les notions de base nécessaires à la compréhension du problème de partitionnement de graphe.

Un arbre G est un graphe connecte acyclique (aucune boucle).

Un arbre de représentation est un sous-graphe de G qui a le même ensemble de nœuds que G et qui est un arbre. Un nœud correspond à un point de dimension P (c.à.d. le spectre d'une image). La branche séparant deux nœuds correspond à la distance existante entre deux images (le terme de distance jouera le rôle de pondération lors de la construction de l'arbre). L'algorithme tolère n'importe quelle matrice de distance, mais nous nous limiterons à l'utilisation des métriques définies dans la section précédente. Etant donne un graphe non dirige G, un arbre de représentation G, dont la somme des poids des branches est minimum parmi tous les arbres de représentation de G, correspond à l'arbre de représentation minimale.

Ce problème admet une solution de complexité polynomiale.

Il existe plusieurs algorithmes permettant de construire un MST de manière exacte, les plus efficaces étant ceux de Kruskal [JBK56] et de Prim [PRIM57]. Nous avons choisi d'utiliser l'algorithme de Prim, qui construit l'arbre de manière itérative. Ce sont les caractéristiques des itérations successives qui sont alors utilisées et non la structure résultante globale de MST (voir Figure 37).

L'efficacité a été démontrée dans le même contexte astrophysique dans [OJMP05] et dans [OLIV05].

On peut détailler brièvement les différentes étapes de l'algorithme de Prim pour former un MST :

- On commence avec n'importe quel sommet et on le joint au plus proche sommet du réseau.
- Les sommets résultants forment un ensemble connecte et les sommets restants un ensemble non connecte. On choisit par la suite un sommet parmi l'ensemble non connecte tel qu'il est le plus proche de n'importe quel sommet de l'ensemble connecte.
- Ce nouveau sommet s'ajoute à ceux précédents pour former un nouvel ensemble connecte. On répète cette procédure jusqu'à ce que l'ensemble de sommets non connectés soit vide (N-1 itérations).

L'algorithme de Prim construit un arbre minimal acyclique.

On peut visualiser sur un exemple simple les différentes étapes de la construction du MST par l'algorithme Prim.

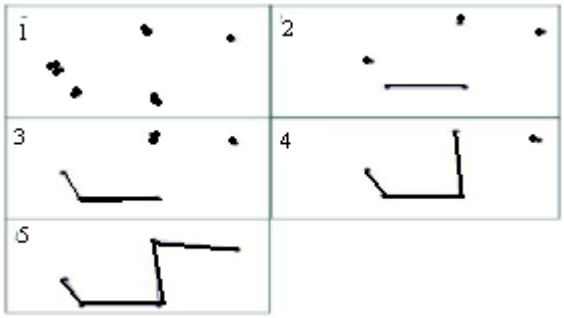


Fig. 34 – Exemple de construction d'un MST par l'algorithme PRIM

En l'absence de précision sur la distance voulue pour l'étude, le programme utilise une métrique euclidienne par défaut pour caractériser la notion de distance existante entre chaque point. On peut visualiser une courbe de distance en fonction de l'ordre d'itération dans la

construction du MST Figure 35. Cette courbe représente la distance séparant le nœud i au nœud suivant i+1, c'est-à-dire l'augmentation de la longueur totale de la structure connectée après l'itération i (i sommets sont alors connectes entre eux). On observe que des ensembles de sommets sont connectes par des segments de longueurs faible consécutivement.

La courbe (Figure.35) présente alors des zones "homogènes" de petite amplitude, ou des "vallées". Ces zones sont caractéristiques de la présence de groupes de points proches entre eux (toujours en termes de spectre). On constate également des distances importantes pour certaines itérations.

Cela sous-entend que les points sont éloignés et ne peuvent donc appartenir à un même groupe.

Fig.35 Représentation de la distance entre les différents points du graphe au cours des itérations.

On remarque bien l'importance da la valeur que l'on prend au seuil permettant de détecter les limites des vallées, qui correspondent aux surdensités de points (c.à.d. clusters). Pour pouvoir déterminer le seuil α qui permet de détecter le plus de surdensités de points, on le fait varier de manière plus ou moins fine. On doit également spécifier un autre paramètre : *nmin*, permettant de ne considérer comme "cluster d'intérêt" que des "vallées" impliquant un nombre minimum de points.

Soit χ l'ensemble des données $\in \mathbb{R}^P$, et G un arbre, la longueur d'un groupe à l'intérieur d'un arbre est la somme des longueurs de toutes les branches reliant les points d'un même groupe élevés à la puissance $\gamma \in (0, d)$.

$$\mathcal{L}(\chi) = \sum |e|^{\lambda}$$

.

avec $|e|$ la métrique utilisée (λ est fixe à 1 dans un premier temps). La métrique $|e|$ doit être euclidienne ou doit tendre vers une distance euclidienne [OLIV05]. Si on souhaite donner à $\mathcal{L}$ une interprétation en termes d'entropie : le MST constitue un estimateur de l'entropie de Rényi de la distribution des points dans l'espace (démonstration dans [OLIV05]).

On fait donc varier la valeur du seuil et on relève à chaque instant le nombre de surdensités de points présent sous le seuil. Suivant *nmin*, le nombre de groupes détectés varie.

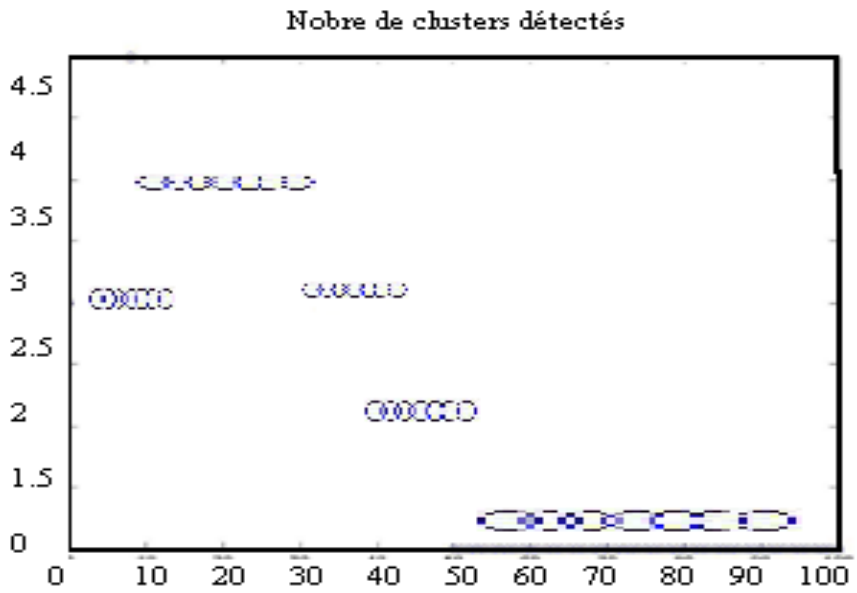


Fig. 36 – Nombre de classes détectés en fonction du seuil

K-Means est un algorithme de partitionnement basé sur le calcul de distances entre les points et les centres de gravité (barycentres) des classes et dont le but est la minimisation de l'inertie intra-classes.

Il faut initialiser ces points (les barycentres) au début de l'algorithme. On doit également spécifier le nombre de clusters à former.

Ces deux points étant les principaux inconvénients de K-Means, nous avons décidé d'utiliser les informations trouvées lors de la construction du graphe de représentation minimale par l'algorithme Prim. En effet, on a déterminé précédemment le nombre de classes maximal qu'on pouvait former en fonction du seuil choisi. Ce nombre constituera donc le nombre de clusters que devra former

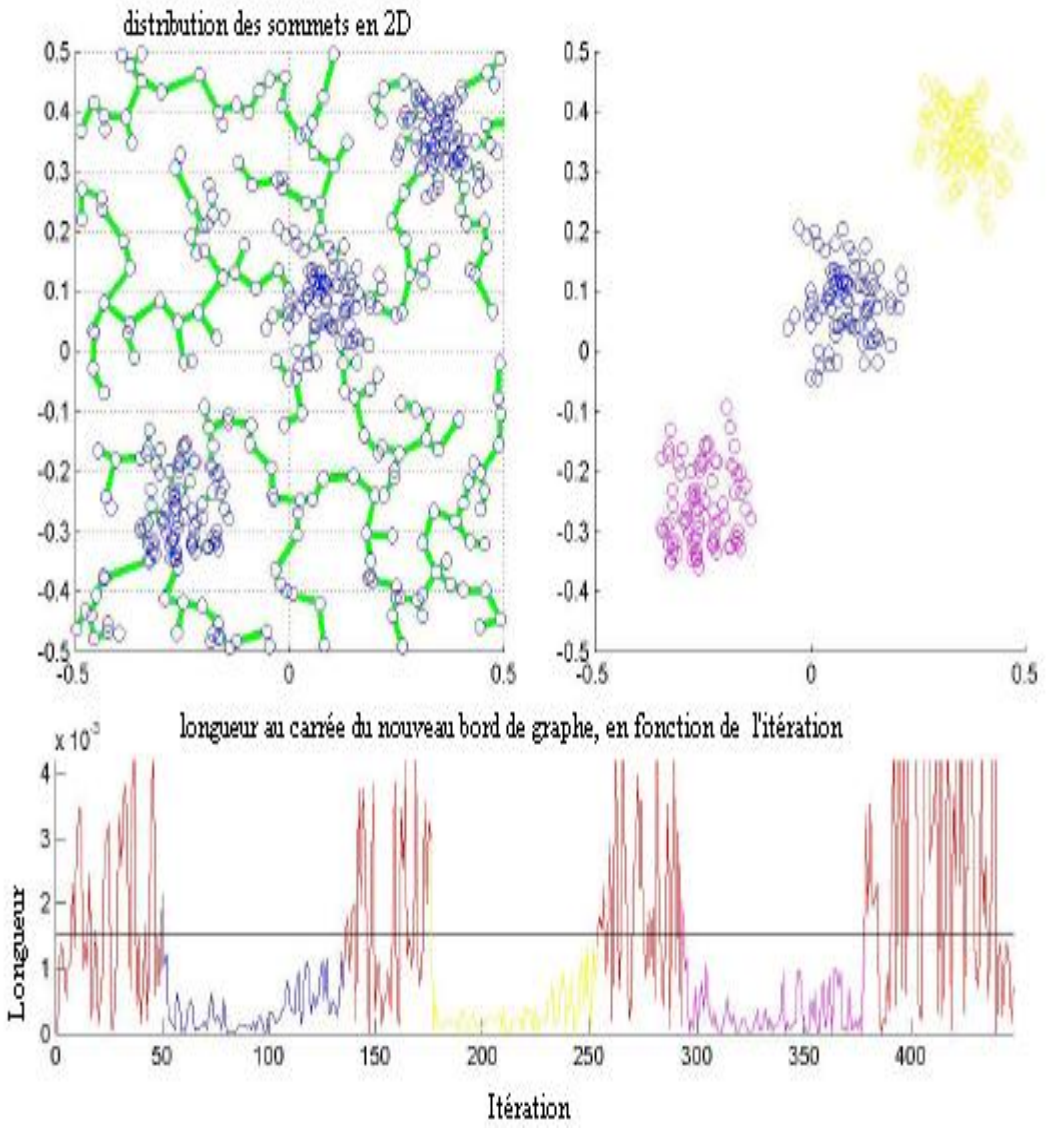


Fig. 37 – Exemple de détection de groupes effectuée par notre programme

L'algorithme K-Means. Si on effectue une initialisation aléatoire des centres de gravité des classes, le résultat obtenu peut ne pas être optimal (présence de minima locaux). Pour palier ce problème et donc obtenir de meilleurs résultats, les barycentres des K classes seront initialisés aux barycentres des classes déterminés précédemment lors de la détermination des groupes suite à la construction du MST.

Voici les différentes étapes que suit l'algorithme K-Means implémenté :

1. Théoriquement, l'algorithme choisit aléatoirement K points comme étant les centres des K classes. Pour augmenter la vitesse d'exécution du programme et pour avoir de meilleurs résultats, nous avons choisi de placer arbitrairement ces points aux barycentres de chaque classe définie auparavant. Les connaissances à priori que nous imposons dans cet algorithme améliore grandement les résultats.

2. Les points restants sont assignés à la classe dont le barycentre est le plus proche.
3. L’algorithme calcule ensuite le nouveau barycentre de chaque classe ainsi obtenue.
4. Les opérations 3 à 5 sont réitérées jusqu’à ce que le centre des classes ne varie plus.

On visualise les données à chaque étape du traitement en 3 dimensions, c.à.d. ”Suivant trois filtres” (Figure. 38, 39, 40).

On vient d’étudier une méthode de classification se basant sur la construction d’un graphe de représentation minimale et sur l’algorithme K-Means, on va à présent détailler la seconde méthode employée pour définir une classification non supervisée : le partitionnement spectral (spectral clustering).

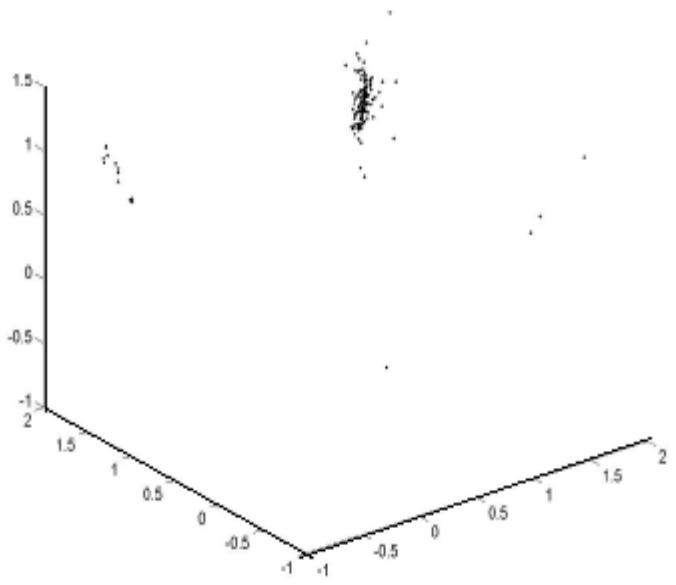


Fig. 38 – Visualisation en 3 dimensions des données

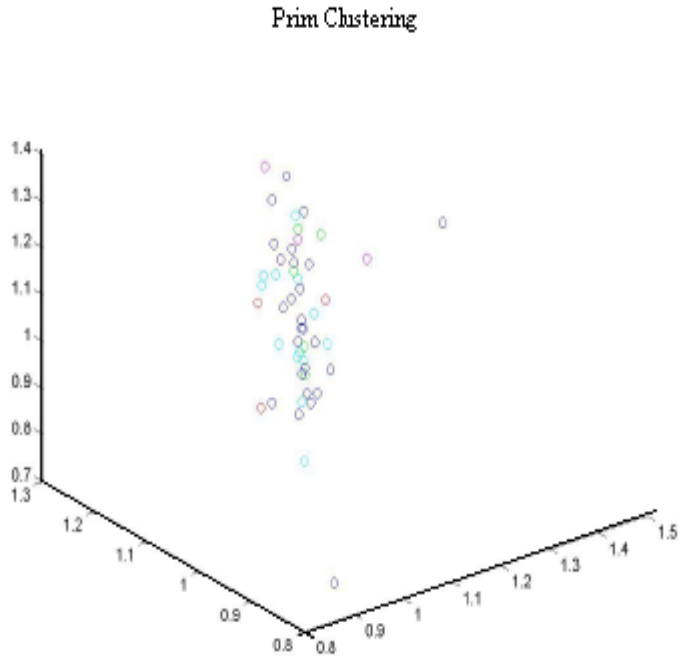


Fig. 39 – Mise en relief des groupes trouvés lors de la construction du MST par l'algorithme Prim

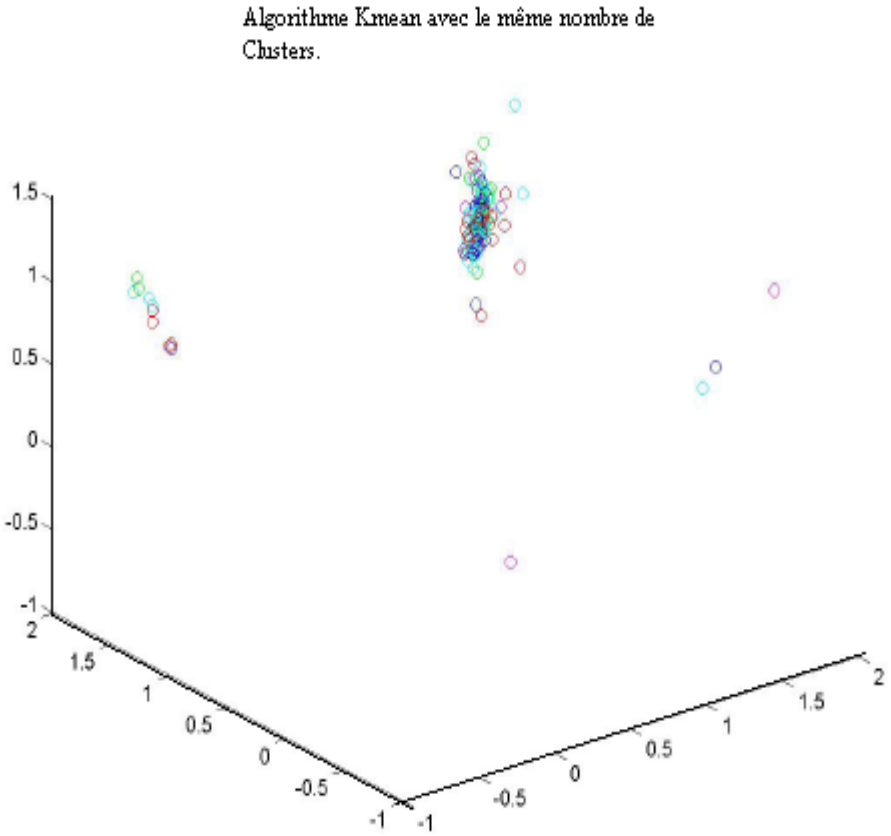


Fig. 40 – Groupes finaux après K-Means

7.3. Spectral Clustering.

Avant de rentrer dans le vif du sujet, rappelons brièvement les notions importantes à la compréhension du problème.

Hypothèse : Une matrice de "dissimilarité" entre chaque paire d'éléments d'un ensemble V de N objets est connue ou supposée comme telle. Soit D cette matrice, de terme général d_{ij}.

La dissimilarité la plus simple quand les objets considérés sont des points de $\mathbb{R}^n$, est la distance euclidienne. On supposera que toutes les mesures de similarité (ou de dissimilarité) sont des grandeurs réelles, positives ou nulles (si cette mesure de dissimilarité respecte toutes les propriétés énoncées dans la section 2, on appellera cette dissimilarité une distance).

Les limitations de l'approche 'single cut clustering' (exploitation des MST par exemple), et son instabilité ou son manque de robustesse en présence de bruit conduisent à introduire une alternative consistant à considérer cette fois un ensemble de coupures permettant de définir un groupe (cluster). S'il semble naturel de considérer un graphe totalement connecté (chaque élément de l'ensemble est connecté aux (N −1) autres éléments), les idées développées dans la suite s'appliquent encore dans la cas de graphes partiellement connectés.

L'ensemble des résultats développés ci après ont été proposés et étudiés par Shi et Malik [SHIMAL00], puis par Jordan et Bach [FRMJ03]. Notre contribution consiste en une présentation plus simple des calculs justifiant l'approche 'spectral clustering' d'une part, et en l'utilisation de notion de dissimilarité (ou de similarité) peu classiques dans le contexte de la segmentation non supervisée.

7.3.1. Définition de la notion de NCUT (Coupure Normalisée).

On introduit les notions de longueur de coupure (CUT), coupure normalisée (NCUT) et de longueur d'association (ASSOC) à partir d'un exemple mettant en jeu deux classes (deux clusters) et un ensemble de liens entres les éléments constitutifs de ces deux classes ; cet exemple est illustré sur la figure suivante (Figure 41). Si tous les liens existants entre les différents éléments sont pris en compte, la somme des mesures de ces liens n'est autre que $ASSOC(V,V)$ (premier graphe).

Si la mesure associée à chaque lien est sa longueur euclidienne, $ASSOC(V,V)$ est la somme des longueurs des segments reliant les points.

On définit $ASSOC(V,V)$ comme étant la somme des mesures des liens entre les éléments de la classe A et tous les autres éléments de V (les éléments de A inclus aussi par conséquent).

$$ASSOC(A,V) = \sum_{i\in A}\sum_{i\in A} d_{ij} \qquad (7.1)$$

La définition analogue équivaut pour $ASSOC(B,V)$.

On constate que $ASSOC(A,A)$ (et par analogie $ASSOC(B,B)$ correspond à la somme des mesures des liens (c.à.d. la distance si la métrique utilisée est euclidienne) reliant les éléments de (A et B) entre eux.

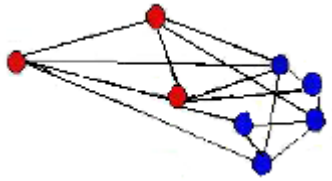
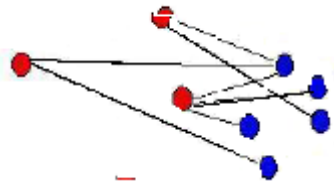
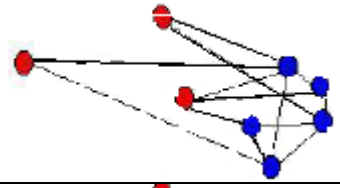
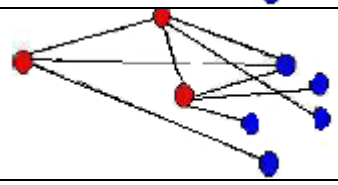
	ASSOC(V,V) = Graphe Complet
	CUT(A,V-A)
	ASOC(A,V)
	ASSOC(B,V)

Fig. 41 – Exemples pour définir les notions de CUT et d'ASSOC.

On définit ensuite la N-coupure entre les ensembles A (points bleus) et B (points rouges) comme étant la somme des mesures des liens reliant les éléments de A et de B, ou encore la somme des mesures des liens que l'on enlèverait si on devait séparer les deux classes A et B. Cette notion de coupure se définit par :

$$CUT(A,B) = \sum_{i \in A} \sum_{j \in V-A} d_{ij} \qquad (7.2)$$

Dans l'exemple proposé sur la figure, pour lequel B = V − A (A et B forment une partition de V, $A \cup B = V$, CUT(A,B) est représenté sur le deuxième graphe.

Avec les définitions citées ci-dessus et à l'aide des figures, on peut remarquer qu'il existe des relations entre ces diverses quantités :

CUT(A,B) = ASSOC(V, V) − ASSOC(A,A) − ASSOC(B,B)

ASSOC(A, V) = ASSOC(V, V) − ASSOC(B,B)

Et en associant les deux égalités précédentes :

CUT(A,B) = ASSOC(A, V) − ASSOC(A,A)

Par analogie, on peut définir également les notions suivantes :

ASSOC(B, V) = ASSOC(V, V) − ASSOC(A,A)

Donc, une autre définition du CUT :

CUT(A,B) = ASSOC(B, V) − ASSOC(B,B)

Les graphes en troisième et quatrième position sur la figure illustrent les mêmes définitions que ci-dessus, pour des arguments différents.

La coupure normalisée (NCUT), pour le problème de définition d'une partition de V en 2 clusters A et B, s'écrit :

$$NCUT(A,B) = \frac{CUT(A,B)}{ASSOC(A,V)} + \frac{CUT(A,B)}{ASSOC(B,V)} \quad (7.3)$$

L'intérêt de cette définition normalisée de la coupure a été largement discuté dans la littérature (voir réf. dans [SHIMAL00]).

Cette définition est étendue sans difficulté au cas de la recherche d'une partition de V en R classes notées $A_1, A_2, \ldots, A_R$:

$$NCUT\left(A_i|_{i=1,\ldots,R}\right) = \sum_{i=1}^{R} \frac{CUT(A_i, V-A_i)}{ASSOC(A_i,V)} \quad (7.4)$$

Remarque :

Jusqu'ici, l'ensemble des définitions présentées ne fait pas état de l'importance de la nature des mesures associées aux segments qui relient (quand ils existent) deux éléments de l'ensemble V. Si cette mesure est la longueur euclidienne, il apparait qu'une segmentation correcte (du point de vue intuitif au moins) est une segmentation pour laquelle ces longueurs sont petites entre éléments d'un même cluster, et importantes entre éléments de deux clusters différents. Le problème de segmentation (NP-Complet) est donc formulé comme étant la recherche de la partition de V en R classes Ai, qui maximise NCUT (Ai|i=1,...,R), de telle sorte à définir des clusters bien séparés les uns des autres.

Pour des raisons que nous développerons par la suite, nous allons (comme dans [SHIMAL00] et [FRMJ03]) utiliser une mesure d'affinité ou de ressemblance, qui prend des valeurs réelles positives, grandes entre deux éléments d'un même cluster, petite dans le cas contraire. Un choix souvent utilisé dans la littérature est :

$$w_{ij} = e^{-\alpha d_{ij}}$$

En fait, Shi et Malick [SHIMAL00] imposent que la matrice de similarité respecte leur hypothèse que le poids des liens existant entre deux points est inversement proportionnel à la distance (dissimilarité) séparant ces points. Par conséquent, n'importe quelle matrice W respectant cette condition est valable. On peut par exemple utiliser aussi $w_{ij} = e^{-\alpha d_{ij}^2}$, ou encore $w_{ij} = \mathrm{I} - d_{ij}$ mais la lenteur de ce dernier a été démontrée dans [SHIMAL00].

Dès lors, le problème de segmentation que l'on veut résoudre revient à chercher la partition de V en R $classes$ A_i , de sorte à minimiser $NCUT(A_i|_{i=1,\dots,R})$ (car minimiser le NCUT revient à maximiser la dissimilarité existante entre deux classes).

7.3.2. Propriétés de NCUT : notations et définitions.

Soit W la matrice de terme général $w_{ij}, où\ i \in \{1, \dots, N\}, j \in \{1, \dots, N\}\ \ et\ w_{ij} \geq 0. W$ est ici une matrice d'affinité ou de ressemblance. La matrice W est défini semi-positive et symétrique $w_{ij} = w_{ji}$ On introduit aussi $D = diagW1I$, matrice diagonale de terme général :

$$d_{ij} = \sum_{j \in A_i} w_{ij}$$

1I désignant le vecteur de $\mathbb{R}^N$ dont toutes les composantes sont égales à un. Chacun des termes qui apparaissent dans l'expression de $NCUT(A_i|_{i=1,\dots,R})$ N s'exprime, en utilisant les égalités présentées dans les sections précédentes

$$\frac{CUT(A_i, V - A_i)}{ASSOC(A_i, V)} = \frac{ASSOC(A_i, V) - ASSOC(A_i, A_i)}{ASSOC(A_i, V)} \tag{7.5}$$

$$= 1 - \frac{ASSOC(A_i, A_i)}{ASSOC(A_i, V)} \tag{7.6}$$

Finalement :

$$NCUT(A_i|_{i=1,\dots,R}) = \sum_{i=1}^{R} \frac{CUT(A_i, V - A_i)}{ASSOC(A_i, V)} = R - \sum_{i=1}^{R} \frac{ASSOC(A_i, A_i)}{ASSOC(A_i, V)} \tag{7.7}$$

Soit encore, en introduisant les w_{ij}

$$NCUT(A_i|_{i=1,\dots,R}) = R - \sum_{i=r}^{R} \left(\frac{\sum_{i,j \in A_r} w_{ij}}{\sum_{i \in A_r, j \in V} w_{ij}}\right) \tag{7.8}$$

On en déduit la propriété suivante : $0 \leq \ NCUT(A_i|_{i=1,\dots,R}) \ \leq R$

Cette propriété vient de l'inégalité trivialement vérifiée $\sum_{i,j\in A_r} w_{ij} \leq \sum_{i\in A_r, j\in V} w_{ij}$

et de la positivité des w_{ij}.

7.3.3. Reformulation du problème de segmentation par coupure normalisée (NCUT).

Soient $e_1, \dots, e_R$ *les vecteurs de* $\mathbb{R}^N$, définis comme les indicateurs de chacune des classes que l'on cherche à identifier

$$\begin{cases} e_i(j) = 1 & si\ j \in A_i \\ e_i(j) = 0 & sinon \end{cases}$$

Soit la matrice $E \in \mathbb{R}^N \times \mathbb{R}^R$dont chacune colonne est un vecteur$e_i, i = 1, \dots, R$:

$E = [e_1, \dots \dots, e_R]$

On établit facilement que $NCUT(A_i | i = 1, \dots, R)$ peut s'exprimer :

$$NCUT(A_i | i = 1, \dots, R) = R - Tr((E^T W E)(E^T D E)^{-1} \quad (7.9)$$

Où l'exposant T désigne la transposition est Tr(.) est l'opérateur 'Trace'.

En remarquant que la matrice $E^T DE \in \mathbb{R}^R \times \mathbb{R}^R$ est une matrice diagonale dont les termes sont tous positifs, $(E^T DE)^{-1})$partage ces propriétés, on peut donc introduire $\Lambda \in \mathbb{R}^R \times \mathbb{R}^R$ telle que :

$$(E^T DE)^{-1}) = \Lambda\Lambda^T \quad (7.10)$$

La commutativité (si les matrices le permettent) de l'opérateur Tr(.) conduit alors à :

$$\mathrm{Tr}((\mathrm{E}^{\mathrm{T}}\mathrm{WE})(E^T DE)^{-1}) = Tr((E\Lambda)^T W(E\Lambda)). \quad (7.11)$$

On observera facilement à partir de la définition de E que la matrice$E\Lambda$ est une matrice de $\mathbb{R}^N \times \mathbb{R}^R$ dont chaque colonne a la propriété suivante : les N composantes de chaque vecteur colonne ne prennent que R valeurs différentes, ces dernières étant constantes pour tous les éléments d'un même cluster.

Par ailleurs, en notant que :

$$Tr((E^TWE)(E^TDE)^{-1}) = Tr((E^TWE)(E^TD^{\frac{1}{2}}D^{\frac{1}{2}}E)^{-1}. \quad (7.12)$$

on est conduit à chercher la solution E sous la forme $D^{-\frac{1}{2}}Y$ où $Y \in \mathbb{R}^N \times \mathbb{R}^R$.

$D^{-\frac{1}{2}}$ étant une matrice diagonale, Y possède les mêmes propriétés que $E\wedge$ (chaque colonne ne comprend que R valeurs, constantes pour un même cluster).

On notera (P) cette propriété par la suite.

On montre alors (démonstration reprise de [FRMJ03]) que Y, défini par l'égalité

$$E\wedge = D^{-\frac{1}{2}}Y, vérifie \qquad Y^TY = \mathrm{I}_{\mathbb{R}^R\times\mathbb{R}^R} \quad (7.13)$$

En effet,

$$Y^TY = A^TE^TDE\wedge = \wedge^T (E^TDE)^{\frac{1}{2}}(E^TDE)^{\frac{1}{2}}\wedge \quad (7.14)$$

Car E^TDE est diagonale et positive. On peut donc exprimer :

$$Y^TY = M^TM \quad où\ M = (E^TDE)^{\frac{1}{2}}\wedge \quad (7.15)$$

M est une matrice carrée de $\mathbb{R}^R \times \mathbb{R}^R$.

On établit facilement que $MM^T = \mathrm{I}_{\mathbb{R}^R\times\mathbb{R}^R}$, ce qui implique que $M^TM = \mathrm{I}_{\mathbb{R}^R\times\mathbb{R}^R}$ et établit donc la propriété.

Finalement, en regroupant l'ensemble des résultats précédents, le problème de segmentation conduit `

$$\begin{cases} Minimiser\ \ NCUT(E) = R - Tr(Y^T.D^{-\frac{1}{2}}W.D^{-\frac{1}{2}}.Y) \\ \qquad avec\ Y^TY = \mathrm{I}_{\mathbb{R}^R\times\mathbb{R}^R} \\ \qquad colonne(Y)\ \ satisfait\ \ (P) \end{cases} \quad (7.16)$$

La résolution du problème ainsi formulé s'appuie sur un théorème, cité par Jordan et Bach, connu sous le nom de théorème de Ky Fan. Ce théorème permet de déterminer le maximum de la grandeur définie par $Tr(Y^T\psi Y)$ quand Ψ est une matrice définie positive. De plus, cela permet d'identifier le sous-espace principal de dimension R, dont les colonnes de Y sont une base :

– le maximum (par rapport aux fluctuations en Y) de $Tr(Y^T\psi Y)\ vaut\ \lambda_1 + \cdots .. + \lambda_R$ où les $\lambda_i, i = 1, \ldots., R$ sont les R valeurs propres principales de la matrice définie positive Ψ.

– La matrice Y solution n'est pas définie de façon unique : toute solution $Y = U.B$ où $U \in \mathbb{R}^N \times \mathbb{R}^R$ est une matrice donc les colonnes sont les vecteurs propres de Ψ, associées aux valeurs propres $\lambda_i, i = 1, \dots, R, B \in \mathbb{R}^R \times \mathbb{R}^R$ est une matrice de rotation arbitraire.

7.4. NCUT et méthode des K-means pondérés :

Il importe de souligner à nouveau qu'à l'issu du paragraphe précédent, le problème de segmentation est reformulé comme un problème de décomposition en éléments propres. Ce résultat obtenu par Bach et Jordan en s'appuyant sur le théorème de Ky Fan ne prend cependant pas du tout en compte la contrainte exprimée sur la matrice Y, qui doit vérifier la propriété (P).

Dans le cas général, la prise en compte de cette contrainte est difficile ; c'est pourquoi, elle est très souvent relaxée (abandonnée...), comme par exemple dans l'article de Ng ([ANJYW01]).

Les résultats obtenus sont alors satisfaisants, sous réserves que les clusters recherchés soient "assez séparés". Cette notion n'est pas précisée, en dehors d'un contexte asymptotique où les clusters sont infiniment distants les uns des autres.

On peut cependant assez facilement comprendre pourquoi les solutions proposées se développent toutes sur l'utilisation de l'algorithme des 'K-Means', plus souvent connu sous le nom d'algorithme de Lloyd pour la quantification vectorielle.

Supposons que la propriété (P) n'est qu'approximativement vérifiée, au sens où les éléments d'un même "cluster" au sein d'une colonne de la matrice Y prennent tous une valeur presque identique ((P) imposerait une valeur rigoureusement constante pour les éléments d'un "cluster").

Dans l'espace à R dimensions engendré par les colonnes de Y, les clusters sont bien séparés, tant que les fluctuations autour de la valeur idéale sont petites devant la séparation caractéristique entre deux clusters. Les éléments d'un même cluster forment donc dans cet espace, un 'paquet' dans RR. L'identification de ces paquets, ce qui revient à identifier les clusters cherchés, peut donc être résolue directement dans ce nouvel espace à l'aide de l'algorithme de Lloyd.

Il est important de souligner que les travaux de Bach et Jordan ont permis d'établir que la recherche d'une solution de segmentation dans laquelle la propriété (P) n'est pas relaxée, conduit à mettre en œuvre un algorithme de 'K-means' pondérés, décrit avec les notations introduites ci-avant par :

1. $\forall r, \quad C_r = \dfrac{\sum_{j \in A_r} d_j^{\frac{1}{2}} U_j}{\sum_{j \in A_r} d_j}$
2. $\forall j \in \{1, \dots, N\}$ *assigne l'élément* j *au cluster* r, *où* $r = Argmin_m \left\| U_j d_j^{-\frac{1}{2}} - C_m \right\|$

Les étapes '1' et '2' étant itérées jusqu' a l'obtention d'une partition stationnaire. On reconnait bien une forme pondérée de l'algorithme de Lloyd. La mise en place et les évaluations des performances et coûts de cette approche à priori plus performante feront l'objet d'un travail ultérieur.

Nous venons de détailler les techniques de classification non supervisée (spectral clustering) développées au cours de ma thèse sur les différentes métriques explicitées dans la section précédente.

7.5. Expérimentation.

Nous allons à présent présenter les résultats obtenus à travers les expérimentations réalisés par l'application de notre approche de clustering spectrale sur descripteur hybride (point SIFT associé avec un descripteur de texture WT (Transformée d'ondelettes) basé sur graphe bipartite.

Le schéma de notre système est comme suit :

1- Extraction des descripteurs des images (SIFT+WT) pour constituer la base d'index de notre CBIR.

2- Clustering spectrale : projection de l'espace multidimensionnelle (descripteurs d'images de la base) sur un plan à base de représentation un graphe bipartite.

3- Calcul des descripteurs (SIFT+WT) de l'image requête.

4- Mise en correspondance : c'est à ce niveau que le gain en temps de traitement est très significatif, du fait que notre système ne reprend pas les traitements précédents, mais seulement, nous avons appliqué le théorème de nystrom, et donc nous calculons la troisième valeur propre de la matrice d'affinité qui est déjà généré précédemment dans la constitution de la base d'index. Cette nouvelle valeur propre sera utilisée pour l'appariement du graphe et donc nous restituer tous les nœuds du graphe d'index auxquels cette image requête est similaire.

Les tests ont été réalisés suivant les différents traitements de données réalisés.

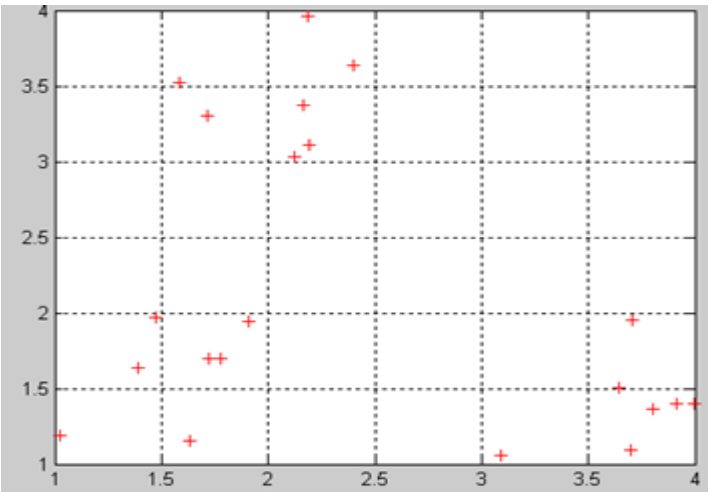


Fig. 42 : Exemple de base d'index d'origine.

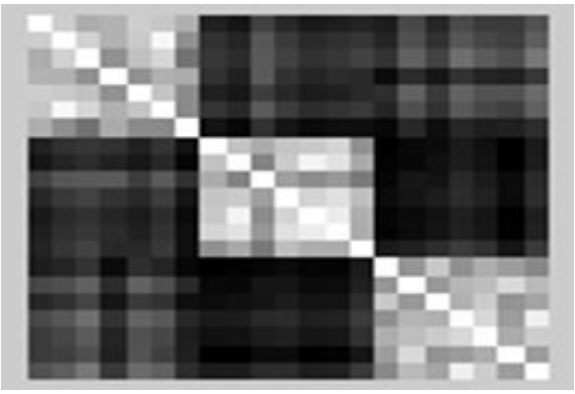
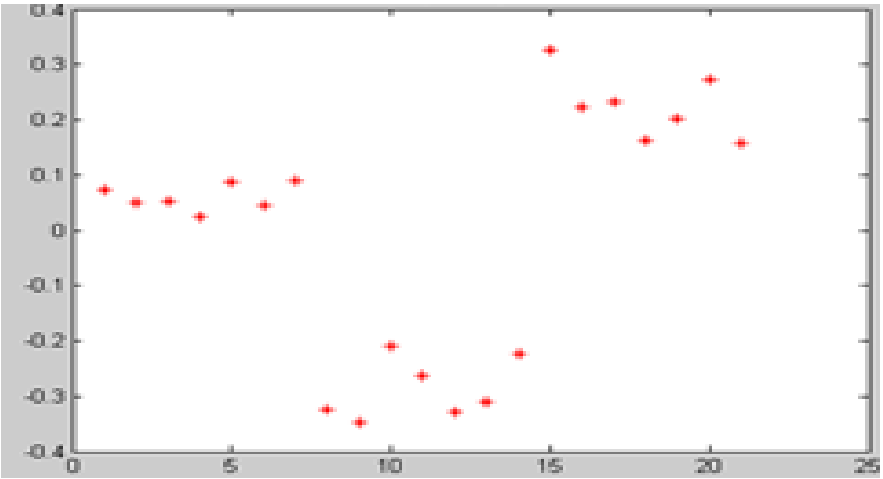
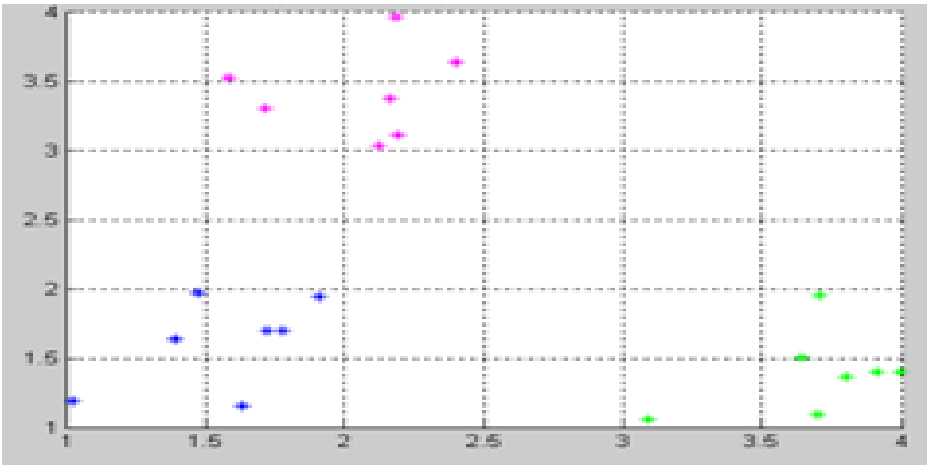


Fig.43 : La matrice d'affinité générée par clustering spectrale.



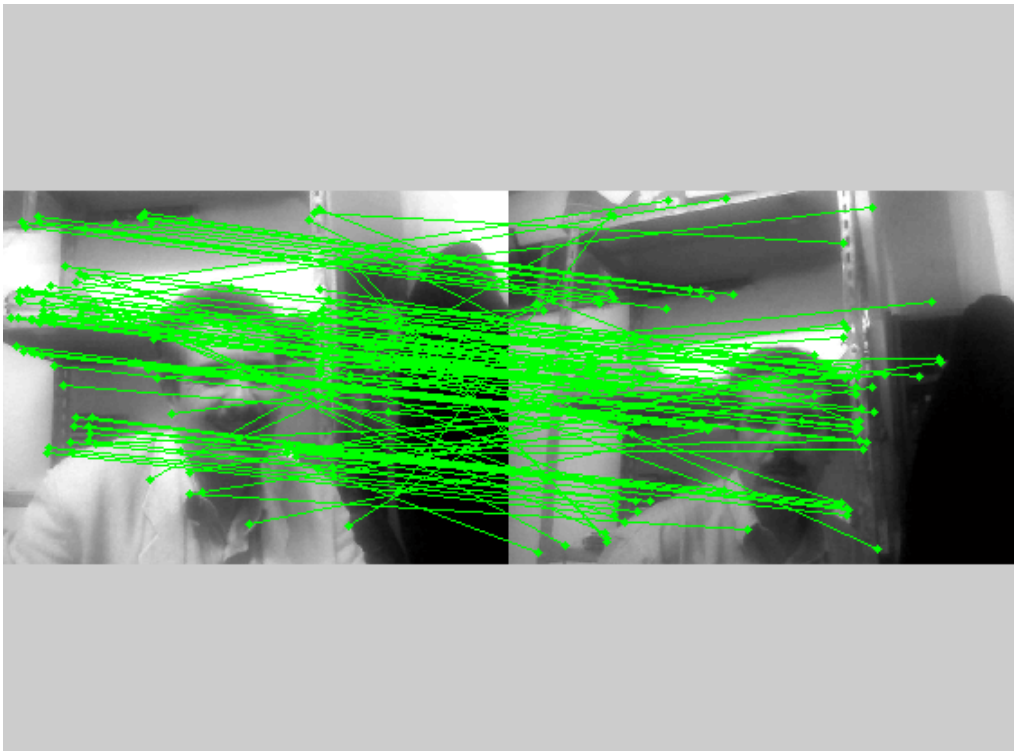
(a)



(b)

Fig. 44 : Résultats de Classification en utilisant seulement les deux premières valeurs propres de la matrice d'affinité : (a) avant classification,(b) après classification .

La comparaison de notre approche par rapport à d'autres méthodes de classification, montre que notre méthode de clustering spectrale sur graphe bipartite donne de bien meilleurs résultats.



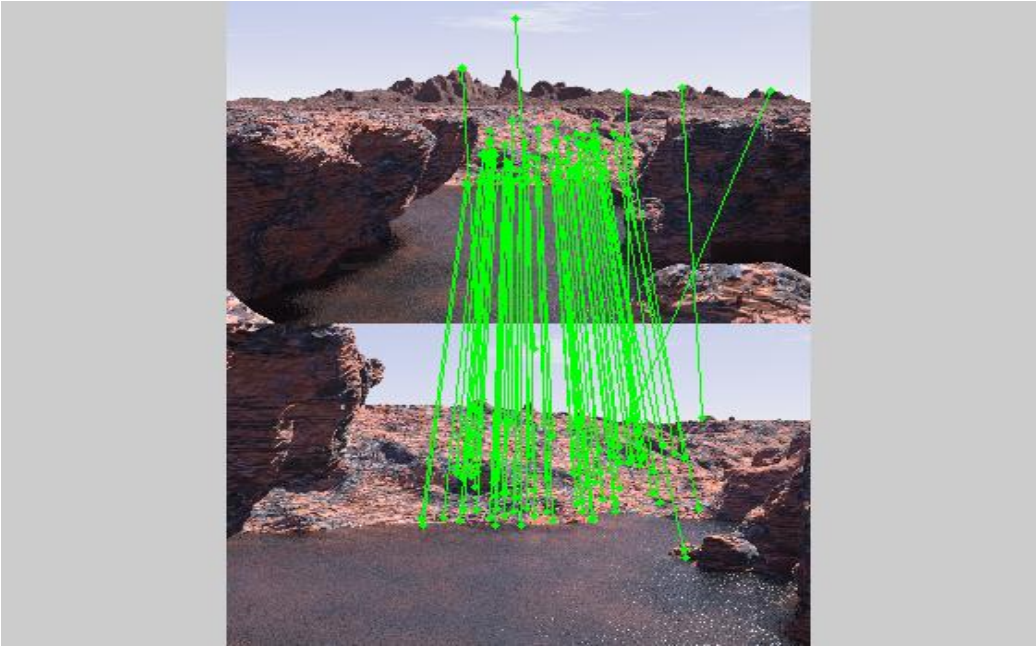


Fig. 45 : Résultats de Classification en utilisant seulement les deux premières valeurs propres de la matrice d’affinité

Conclusion générale et perspectives

Dans cette thèse, nous avons étudié tous les aspects théoriques de l'état de l'art de la problématique qui reste toujours ouverte à de nouvelles pistes de recherches. Nous avons aussi testés les différentes approches pour mieux cerner les difficultés et valider les résultats.

Dans un premier travail, nous avons proposé une solution qui permet d'indexer les images à travers l'identification de surfaces. Dans ce cadre, nous nous sommes intéressés plus particulièrement au problème de l'indexation d'une base de données cartographique par une recherche du contenu d'une image aérienne (on s'est intéressé à l'aspect extraction de routes), et dont les principales étapes sont:

- Première Phase: l'extraction non supervisée des réseaux linéiques à partir d'images aériennes. Nous avons utilisés dans cette étape une modélisation du réseau linéique par un processus dont les objets sont des segments interagissant entre eux. Le modèle à priori est construit de façon à exploiter au mieux la topologie du réseau recherché à travers des potentiels fondés sur la qualité de chaque interaction. Les propriétés radio métriques sont prises en compte dans un terme d'attache aux données fondé sur des mesures statistiques.
- Comme seconde phase, nous avons procédé à la reconstitution d'une grille des surfaces planes avec une méthode de classification qui respecte les relations spatiales entre les points (c.à.d. information de voisinage). Nous avons appliqué les trois méthodes suivantes :

    - La méthode du voisin le plus proche.
    - Analyse Discriminante Linéaire (ADL) et l'Analyse Discriminante Quadratique (ADQ).
    - Machines à vecteur de support (SVM).

Par l'application de ces trois méthodes, nous aurons une carte de classes labellisées des régions par contours actifs de notre image requête.

Pour déterminer les classes qui seront utilisé comme entrées dans la prochaine phase nous ne tiendrons en compte que les classes de régions connexes et éliminons les classes de régions discontinues.

- Troisième phase : Indexation de la base de données cartographique à travers l'appariement de la grille des surfaces connexes obtenues. Ceci en calculant les surfaces relatives de chaque classe de surface et qui seront introduits comme vecteurs de caractéristiques pour chaque surface (on considère que chaque surface ne peut pas avoir plus de 8 arêtes connexes). Et donc, Nous avons adopté l'approche de segmentation en régions qui fournit une approximation naturelle des objets. Pour satisfaire les besoins d'un système de recherche par régions, nous avons proposé une approche de segmentation rapide et non supervisée pour détecter des régions grossières et pertinentes pour l'utilisateur.
- Quatrième Phase : Restitution des emplacements possibles trouvés par notre moteur de recherche de l'image requête.

Récemment, l'intérêt pour la réduction et le traitement de données de haute dimension s'est tourné vers les méthodes de clustering spectrales, en raison de nombreux succès [TEN00, ROW00, ROW01, LAF05, LAF06]. Ces méthodes emploient le contenu spectral d'une matrice de similarité (distance entre chaque paire de données) pour réaliser la réduction et la partition d'un jeu de données. Plus spécifiquement, les vecteurs propres sont vus comme un outil fournissant une représentation des données dans un espace où il est bien séparé et peut facilement être groupé, et les valeurs propres ayant la possibilité de capturer les caractéristiques conceptuelles et perceptuelles d'un ensemble de données. Cette dernière propriété mathématique n'a pratiquement jamais été prise en compte pour l'évaluation des performances des algorithmes du clustering spectrales.

Nous avons présenté l'approche de la diffusion géométrique par marches aléatoire sur graphe où classification spectrale sur graphes qui constitue un cadre unifié à la réduction de dimensions, au traitement des données, leur organisation et leur visualisation. En effet, cette approche permet une réduction de dimension qui s'affranchit des limites inhérentes aux autres approches et qui permet d'approximer le mieux la norme euclidienne dans les espaces de grande dimension.

Nous avons également développé un modèle non linéaire basé sur la matrice de diffusion géométrique sur graphe adapté à la caractérisation de "textures visuelles". Ce modèle présente un avantage par rapport à d'autres solutions existantes basées uniquement sur le niveau de gris ([TAV03 et TAV06] notamment) puisqu'il permet de tenir compte de toutes les propriétés statistiques de la texture liées au graphe local pour caractériser au mieux l'information visuelle. Nous avons optimisé le temps de calcul en évitant la décomposition spectrale de chaque matrice de transition correspondant au graphe de diffusion locale en calculant seulement la trace.

Enfin, nous avons appliqué note approche de diffusion pour la segmentation d'objets d'intérêt. Cette proposition repose sur l'exploitation des informations portées par les maxima locaux qui représentent l'image. L'avantage de cette méthode est la réduction du coût de construction du graphe d'affinité et la décomposition spectrale de leur matrice de transition au sens du temps de calcul. En plus, chaque point maxima local peut être décrit par un ensemble de descripteurs représentant son voisinage (texture,...).

La gestion satisfaisante d'une vaste collection de documents multimédia est toujours un défi. Plusieurs domaines de recherche y ont contribués : le traitement numérique des signaux, les bases de données, l'intelligence artificielle ou encore l'ergonomie et l'interaction homme-machine. Les problèmes encore ouverts n'ont été que rarement considérés en leur intégrité. L'approche basée sur la diffusion géométrique par marches aléatoire sur graphe peut fournir une série de solutions nouvelles pour différents domaines :

- Catégorisation de documents (texte/image/son...)
- Analyse Vidéo :
  - Analyse d'événement dans une vidéo,
  - Analyse du graphe temporel et décomposition de la vidéo en segments (plans, scènes,.....)
  - Caractérisation d'Objets Visuels
  - Suivi d'objets en mouvement

- Extraction de connaissances dans une masse de données
- Fouille du Web : réseau sociaux,...
- Intelligence économique

Une application majeure de cette approche est la catégorisation d’expressions faciales et labiales en exploitant les informations visuelles, sonores et textuelles est en cours d’étude dans le cadre d’un projet agrée par le ministère ayant pour titre « système biométrique et SMA ».

Cette thèse fut pour moi une expérience très concluante puisqu’elle m’a permis d’une part de mettre en pratique des notions théoriques et d'appréhender un peu mieux les modèles de segmentation d’images, d’analyses d’images, d’analyses de données et de réduction de données et plus particulièrement l’intérêt des graphes en analyse d’image. D’autre part, j’ai également beaucoup appris au niveau technique et méthodologique et à travailler au sein d’une équipe de recherche et m’enrichir des discussions et des points de vue d’autres chercheurs.

C’était un travail passionnant et très riche et dont les applications sont multiples et multi disciplinaires. Il n’y a aucun doute que ce travail sera une très bonne référence dans mon cursus universitaire et ouvrira, je l’espère, de nouvelles pistes de recherche pour d’autres travaux.

Bibliographie Personnelle

Revues internationales :

[Houari et *al.* 2009] K**.** HOUARI, Y. CHAHIR et M.K. KHOLLADI. Spectral clustering and dimensionality reduction applied to Content based image retrieval with hybrid Descriptors**.** . *International Review on Computers and Software (**IRECOS**)*, 2(6): 209-219, novembre 2009.

[Houari et *al.* 2010] K. HOUARI, Y. CHAHIR, M.K. KHOLLADI. SIFT points for Matching Aerial Images in Reduced Space. *International Review on Computers and Software (**IRECOS**).*

Conférences internationales avec comité de lecture et publication des actes

[Houari et *al.* 2006] K. HOUARI, M. K.KHOLLADI et S.BRES. Seek by the content of an aerial images in cartographic database Dans *les actes du 8ème Colloque Africain sur la Recherche en Informatique (**CSIT'2006**)*, pages 377-384, Amman, Jordanie, mars 2006.

[Houari et *al.* 2007] A. HOUARI, Y.CHAHIR et M. K.KHOLLADI. SIFT points for Matching Aerial Images in Reduced Space. 4th International Conference on Computer Integrated Manufacturing **CIP'2007** 03-04 November 2007.

Conférences nationales

[Houari et *al.* 2006] A. HOUARI et M. K.KHOLLADI et Y.CHAHIR. Nouvelle approche de Recherche d'images aériennes par le contenu dans une base de données cartographique. *Journées Internationale d'Imagerie et Graphisme (**JIG'2006**)*, Tebessa, Algérie, Novembre 2006.

[Houari et *al.* 2007] A. HOUARI, M. K.KHOLLADI et Y.CHAHIR. Nouvelle approche de Recherche d'images aériennes par le contenu dans une base de données cartographique. *Journées Ecole Doctorale et Réseaux de Recherche en Informatique (**JED'2007**)*, Annaba, Algérie, mai 2007. http://www.univ-annaba.org/Jed2007/actes_jed2007.php

BIBLIOGRAPHIE.

[AA07] A. ALILAOUAR, "Contribution à l'interrogation flexible de données semi-structurées", thèse de doctorat, Université Paul Sabatier, 2007.

[ABMC90] A.C. Bovik, M. Clark, and W.S. Geisler, « Multichannel texture analysis using localized spatial filters », IEEE Trans. Pattern Analysis Mach. Intell, col 12, pp 55-73, 1990.

[AD93] H. A. Almohamad and S. O. Duffuaa. A linear programming approach for the weighted graph matching problem. IEEE Trans. on Pattern Analysis and Machine Intelligence, 15(5) :522-525, 1993.

[AHSW03] A. Hlaoui and S. Wang, "A new median graph algorithm", Workshop on graph-based representations in pattern recognition, pp. 225-234, 2003.

[AKRC88] A.K. Jain, R.C. Dubes, Algorithms for clustering Data. Prentice-Hall, Englewood Clifs, NJ, 1988.

[ANJYW01] Andrew Y. Ng, Michael I. Jordan & Yair Weiss. On Spectral Clustering :Analysis and an algorithm. Advances on Neural Information Processing Systems 14,MIT Pres., 2001.

[APCS27] P. Agouris, J. Carswell and A. Stefanidis, "An environment for content-based image retrieval from large spatial databases", Journal of photogrammetry and remote sensing, Vol. 54, No. 4, pp. 263-272, 1999.

[AREH05] A. Robles-Kelly and E. Hancock, "Graph edit distance from spectral seriation", IEEE Transactions on pattern analysis and machine intelligence, Vol. 27, No. 3, pp. 365-378, 2005.

[BA89] D. Blostein and N. Ahuja. Shape from texture : Integrating texture-element extraction and surface estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence 11(12) :1233-1251. 1989.

[BAL78] D. H . Ballard. Parameter networks : Towards a theory of low level vision. Proc 7th IJCAI, Vancouver, pp. 1068-1078, Août 1978.

[BCG90] A.C. Bovik, M. Clark, and W.S. Geisler. Multichannel texture analysis using localised spatial filters. IEEE Transactions on Pattern Recognition and Machine Intelligence, 12(1) :55-73, 1990.

[BF84] B. Bhanu and O. Faugeras. Shape matching of two dimentional objects, IEEE Trans. on Pattern Analysis and Machine Intelligence, 6(2) : 137-156, 1984.

[BLOW02] Brown, M. and Lowe, D.G. Invariant features from interest point groups. In The 13th British Machine Vision Conference, Cardiff University, UK, pp. 253–262, 2002.

[BLOW02+] Brown, M. and Lowe, D.G. Invariant features from interest point groups. In British Machine Vision Conference, BMVC 2002, Cardiff, Wales (September 2002), pp. 656-665. 2002.

[CB87] M. Clark and A. C. Bovik. Texture segmentation using gabor modulation/demodulation. Pattern Recognition Letters, 6(4) :261-267, 1987.

[CC85] R. Chellapa and S. Chatterjee. Classification of textures using gaussian markov random fields. IEEE Trans. Acoustics Speech Signal Process,33 : 959-963. August 1985.

[CC96] C. C. Chen and D. C. Chen. Multi-resolutionl gabor filter in texture analysis. Pattern Recognition Letters,, 17(10) : 1069-1076, 1996.

[CD99] M.L Comer and E.J. Delp. Segmentation of textured images using a multiresolution gaussian autoregressive model. IEEE Transactions on image Processing. 8(3) : 408-420. 1999.

[CJ83] G. C. Cross and A. K. Jain. Markov random field texture models. IEEE Transactions on Pattern Analysis and Machine Intelligence. 5(1) : 25-39. 1983.

[CH80] R.W. Conners and C. A. Harlow. A theoritical comparison of texture algorithms. IEEE Transaction on Pattern Analysis and Machine Intelligence. 2(3) :204-222, 1980.

[CHE02] Kan-Min Chen, Shu-Yuan Chen Color texture segmentation using feature distributions. Pattern Recognition Letters 23 :755–771, 2002.

[CHE03] Ya-Chun Cheng, Shu-Yuan Chen. Image classification using color, texture and regions. Image and Vision Computing 21 :759–776, 2003).

[CJ93] R. Chellapa AND A. K. Jain. Markov Random Fields : Theory and Applications. Academic Press. 1993.

[CK85] R. Chellapa and R.L. Kashyap. Texture synthesis using 2-d noncausal autogressive models. IEEE Trans. Acoustics Speech Signal Process. 33 : 194-203. February 1985.

[CKS98] G.Chartrand, G. Kubicki, and M. Schultz, "Graph similarity and distance in graphs". Journal of aequationes mathematicae, Vol. 55, No. 1/2, 129-145, 1998.

[CLOP01] L, Cinque, S. Levialdi, K. A. Olsen and A. Pellicano. Color based image retrieval using spatial chromatic histograms. Image Vision Compter. 19(13) : 979-986, 2001.

[COH77] M. CoHen et G. T. Toussaint. On the detection of structures in noisy pictures. Pattern Recognition, Pergamon Press, Vol. 9, pp. 95-98, 1977.

[CRJP84] Crowley, J. and Parker, A. A representation for shape based on peaks and ridges in the difference of low pass transform. IEEE Transactions on Pattern Analysis and Machine Intelligence, 6(2):156–170. 1984.

[CSY87] S.K. Chang, Q.Y. Shi and C.W. Yan. Iconic indexing by 2-d strings. IEEE Trans. Pattern Anl. Mach. Intell.,9(3) : 413-429, 1987.

[CZ08] C. ZAYANI, "Contribution à la définition et à la mise en oeuvre de mécanismes d'adaptation de documents semi-structurés", thèse de doctorat, Université Paul Sabatier, 2008.

[DEL99] M. de Andrade, G. Bertrand, et A. Araújo. An attribute-based image segmentation method. Materials Research, 2(3):145–151, 1999.

[DENMAN01] Y. Deng et B. Manjunath. Unsupervised segmentation of color texture regions in images and video. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23:800–810, 2001.

[DBLO05] D. Blostein, "Graph transformation in document image analysis: approaches and challenges", Workshop on graph-based representations in pattern recognition, pp. 23-34, 2005.

[DHW94] D. Dunn, W. Higgins. Optical gabor filters for texture segmentation. IEEE Trans. on Pattern Analysis and Machine intelligence, 16(2) :130-149, 1994.

[DHW95] D. Dunn, W. Higgins. Optical gabor filters for texture segmentation. IEEE Trans. on Image Processibng,4(7) :947-964, 1995.

[JPEMG99] John P Eakins and Margaret E Graham. Content-based Image Retrieval. A report to the JISC Technology Applications Programme. Institute for Image Data Research, University of Northumbria at Newcastle. January 1999.

[FLI95] M. Flickher, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D.Steele, and P. Yanker, "Query by Image and Video Content: The QBIC System," IEEE Computer, vol. 28, no. 9, pp. 23-32, Sept. 1995.

[FRMJ03] Francis R. Bach & Michael I. Jordan. Learning Spectral Clustering. EECS Department, University of California, Berkeley. Technical Report No. UCB/CSD-03-1249. June 2003.

[FBM03] Francis R. Bach & Michael I. Jordan. Learning Spectral Clustering. EECS Department, University of California, Berkeley. Technical Report No.UCB/CSD-03-1249. June 2003.

[FU82] K. S. Fu. Syntactic Pattern Recognition and Applications. Prentice-Hall. New Jersey. 1982.

[FSS05] M. Ferrer, F. Serratosa, and A. Sanfeliu, "Synthesis of median spectral graph", Conference on Pattern recognition and image analysis, pp. 139-146, 2005.

[FP80] O.D. Faugeras and W.K. Pratt. Decorrelation methods of texture feature extraction. IEEE Transaction on Pattern Analysis and Machine Intelligence. 2(4) :323-333,1980.

[GMD97] P. Gros, G. Mclean, R. Delon, R, Mahr, C. Schmid, and G. Mistler. Utilisation de la couleur pour l'indexation et l'appariement d'images. Technical Report RR-3269, INRIA, 1997.

[GS99] T. Gervers and A. W. M. Smeulders. Color based object recognition. Pattern Recognition, 32(3) :453-454. 1999.

[GVDBSG01] J.M. Geusebroek, R, van den Boomgaard, A. W. Smeulders, and H. Geerts. Color invariance. IEEE Trans, on Pattern Ananlysis and Machine intelligence , 23(12) : 1338 1350, 2001.

[GS97] I.Gagliari and R. Shettini. A method for the automatic indexing od color images for effective image retrieval. The New Review of Hypermedia an multimedia, 3 :201 224 , 1997.

[GW02] R.C. Gonzales and R. E. Woods. Digital image processing. Prentice-Hall. New Jersey, 2002.

[GZJ06] O. Grygorash, Y. Zhou, Z. Jorgensen, Minimum Spanning Tree Based Clustering Algorithms,18th IEEE International Conference on Tools with Artificial Intelligence, pp.73-81, 2006.

[GUÉ03] Guérin-Dugué A., Ayache S., Berrut C., Image retrieval : a first step for a human centered approach, in Fourth Pacific-Rim Conference on Multimedia (ICICS-PCM 2003), Singapore, 15-18 December 2003, 2003.

[GOS99] Jaideva C. Goswami et Andrew K. Chan. " Fundamentals of Wavelets: Theory,Algorithms, and Applications ". Wiley Series in Microwave and Optical Engineering. Wiley Interscience ISBN 0-471-19748-3, 1999.

[GS62] D. Gale and L.S. Shapley. College admissions and the stability of marriage. American Mathematical Monthly, 69 :9-15, 1962.

[GUD98] V.N. Gudivada. Thetar-string : A geometry-based representation for efficient and effective retrieval of images by spatial similarity. IEEE Trans. Knowl. Data Eng.,10(3) :504-512, 1998.

[HARA79] R. M. Haralick. Statistical and structural approaches to texture. Proceedings of the IEEE, 67:786–804, 1979.

[HU62] M. Hu. Visual pattern recognition by moment invariants. IRE Transactions on informatio theory, 8:351–326, 1962.

[HSP06] P.S. Hiremath, S. Shivashankar, et J. Pujari. Wavelet based features for color texture classification with application to CBIR. International Journal of Computer Science and Network Security (IJCSNS), 6(9) :124–133, 2006.

[HCS88] Harris, C. and Stephens, M. A combined corner and edge detector. In Alvey Vision Conference, pp. 147–151. 1988.

[HSTV90] Horaud, R., Skordas, T., and Veillon, F. Finding geometric and relational structures in an image. In Proceedings of the 1st European Conference on Computer Vision, Antibes, France, pp. 374–384. 1990.

[HUBBARD95] Barbara Burke Hubbard. Ondes et ondelettes. Sciences d'Avenir.Belin, 1995.

[HAR72] R. O. DUDA et P. E. HART. Use of the Hough Transformation te Detect Lines and Curves in Pictures. Communication of the ACM, Vol. 15, No. 1, pp. 11-15, janvier 1972.

[HBKS98] H. Bunke and K. Shearer, "A graph distance metric based on the maximal common subgraph", Journal of pattern recognition letters, Vol. 19. No. 3/4, pp. 255-259, 1998.

[HBUN01] H. Bunke, "Recent advances in structural pattern recognition with applications to visual form analysis", Workshop on visual form, pp. 11-23, 2001.

[HBME96] H. Bruns and M. Egenhofer, "Similarity of spatial scenes", Symposium on spatial data handling, pp. 31-42, 1996.

[HAR73] Haralik, R.M., Shanmugam, K., et Dinstein, I. Textural features for images Classification. IEEE Transaction on System , Man, Cybernetics, 3,610-621, 1973.

[HUA97] Huang, J., Kumar, S., Mitra, M., Zhu, W.-J., and Zabih, R. Image indexing using color correlogram. In Proc. of IEEE Conf. on Computer Vision and Pattern Recognition. , pages 762-768. 1997.

[HM99] G. M. Haley and B.S. Manjunath. Rotation-invariant texture classification using a complete space-frequency model. IEEE Transactions on Image Processing, 8(2) :255-269, 1999.

[HW92] D. C. He and L. Wang. Unsupervised textural classification of images using the texture spectrum. Pattern Recognition. 25(3) : 247-255, 1992.

[IQB02] Q. Iqbal and J. K. Aggarwal, CIRES: A System for Content-based Retrieval in Digital Image Libraries , Invited session on Content Based Image Retrieval: Techniques and Applications International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, pp. 205-210, December 2-5, 2002.

[JBK56] J. B. Kruskal. On the shortest spanning subtree and the traveling salesman problem. Proceedings of the American Mathematical Society. 7, pp. 48 50. 1956.

[JG07] G.W. Jiji et L. Ganesan. Comparative analysis of colour models for colour textures based on feature extraction. International Journal of Soft Computing, 2(3) :361–366, 2007.

[JD84] M. E. Jernigan and F. D'astous. Entropy-based texture analysis in the spatial frequency domain. IEEE Transactions on Pattern Analysis and Machine Intelligence, 6(2) :237-243, 1984.

[KSGA96] R.Kasturi, S. H. Strayer, U. Gargi, and S.Antani, An evaluation of color histogram based methods in video indexing. Technical report, CSE-96-053, Penn State University, Departement of Computer Science and Engineering, 1996.

[KBT01] L. Kam and J. Blanc-Talon. Multifractal multipermuted multinomial measures for texture characterization and segmentation. Nova Science Publishers, Inc. Commack. NY. USA. 2001.

[KC97] S. Krishnamachari and R. Chellappa. Multiesolution gauss-markov random field models for texture segmentation. IEEE Trans. Image Process. 6(2) : 251-267. 1997.

[KC92] A. Kundu and J. L. Chen. Texture classification using QMF-band-based and sub-band decomposition. Graphical Models and Image Processing, 54(5) : 369-384, 1992.

[LEW96] Michael S. Lew, D. P. Huijsmans, and Dee Denteneer. Content based image retrieval: KLT, projections, or templates. In A. W. M. Smeulders and R. Jain, editors. Image Databases and Multi-Media Search, proceedings of the First International Workshop IDB-MMS'96, Amsterdam, The Netherlands. Amsterdam University Press, pages 27-34. , August 1996.

[LI04A] Jia Li, James Z. Wang, "Studying digital imagery of ancient paintings by mixtures of stochastic models," IEEE Transactions on Image Processing, vol.

[LON02] Fuhui Long, Hongjiang Zhang, David D. Feng: Fundamentals of Content-based Image retrieval, in Multimedia Information Retrieval and Management – Technological Fundamentals and Applications, D. Feng, W.C. Siu, and H.J.Zhang. (ed.), Springer, 2002.

[LON03] Longbin Chen, Bao-Gang Hu, Lei Zhang, Mingjing Li, HongJiang Zhang: Face

Annotation for Family Photo Album Management. International Journal of Image and Graphics, Volume 3, pp 81-94. 2003.

[LIENHART02] Rainer Lienhart et Jochen Maydt. An Extended Set of Haar-like Features for Rapid Object Detection. Dans Proceedings of the IEEE International Conference on Image Processing, volume 1, pages 900{903, Rochester, New York, _Etats-Unis, septembre 2002.

[LASC03] Lazebnik, S., Schmid, C., and Ponce, J. Sparse texture representation using affine invariant neighbourhoods. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, USA, pp. 319–324. 2003.

[LOW99] Lowe, D.G. Object recognition from local scale-invariant features. In Proceedings of the 7th International Conference on Computer Vision, Kerkyra, Greece, pp. 1150–1157. 1999.

[LAF05] S. Lafon, R. R. Coifman, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S., W. Zucker.Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. 7426–7431,PNAS,May 24,. Vol. 102, no. 21, 2005.

[LAF06] Stéphane Lafon, Yosi Keller and Ronald R. Coifman. Data Fusion and Multi-Cue Data Matching by Diffusion Maps. In IEEE transactions on Pattern Analysis and Machine Intelligence, 2006.

[LADT01] R. Laurini and D. Thompson, "Fundamentals of spatial information systems", Academic press, 1992.

[LWW00] J. Li, J. Z. Wang, and G. Weiderhold. IRM : integrated region matching for image retrieval. in ACM Multimedia, pages 147-156, 2000.

[MIN96] T. P. Minka An Image Database Browser that Learns from User Interaction , Master of Engineering Thesis, 1996.

[MIR98] Mihran Tuceryan, Anil K. Jain, "Texture Analysis", The Handbook of Pattern Recognition and Computer Vision (2nd Edition), by C. H. Chen, L. F. Pau, P. S. P. Wang (eds.), pp. 207-248, World Scientific Publishing Co., 1998.

[MIT97] Machine Learning, Tom Mitchell, McGraw Hill, 1997.

[MALA89] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 11, pp. 674-693, July 1989.

[MI02] Mikolajczyk, K. Interest point detection invariant to affine transformations. PhD thesis,Institut National Polytechnique de Grenoble. 2002.

[MEWK99] M. Egenhofer and W. Kuhn, "Interacting with GIS", In P. Longley, M. Goodchild, D. Maguire, and D. Rhind, "Geographical information systems: principles, techniques, applications, and management", Wiley & Sons, pp. 401-412, 1999.

[MNHB04] M. Neuhaus and H. Bunke, "A probabilistic approach to learning costs for graph edit distance", Conference on pattern recognition, Vol. 3, pp. 389-393, 2004.

[MSPXZB07] L. Mukherjee, V. Singh, J. Peng, J. Xu, M. Zeitz, and R. Berezney, "Generalized median graphs: theory and applications", IEEE Conference on computer vision, pp. 1-8, 2007.

[MP90] J. Malik and P. Perona. Preattentive texture discrimination with early vision mechanisms. Journal of the Optical Society of America, 7 : 923-932, 1990.

[MJ92] J. Mao and A.K. Jain. Texture classification and segmentation using multiresolution simultaneous autoregressive models. Pattern Recognition. 25(2) : 173-188. 1992.

[MAN82] B. Mandelbrot. Fractal Geometry of Nature. W. H. Freeman. 1982.

[MB98] B.T. Messmer and H. Bunke. A new algorithm for error tolerant subgraph isomorphism detection. IEEE Trans. on Pattern Analysis and Machine Intelligence , 20(5) :493-504, May 1998.

[NAD94] J.-P. Nadal, Duality between learning machines: a bridge between supervised and unsupervised learning, Neural Computation Vol. 6, Issue 3 (pages 491-508), published by The MIT Press. May 1994.

[NPYM99] A. N. Papadopoulos and Y. Manolopoulos, "Structure-based similarity search with graph histograms", Workshop on database and expert systems applications, pp. 174-178, 1999.

[ORT97] Michael Ortega, Yong Rui, Kaushik Chakrabarti, Sharad Mehrotra, and Thomas S. Huang. Supporting similarity queries in MARS. In Proceedings of the 5th ACM International Multimedia Conference, Seattle, Washington, 8-14 Nov. '97, pages 403-413, 1997.

[OJMP05] Olivier J.J. Michel, Phillippe Bendjoya & Pablo RojoGuer Unsupervised Clustering with MST : Application to asteroıd data, PSIP, Toulouse, 2005.

[OLIV05] Olivier J.J. Michel, Phillippe Bendjoya & Pablo Rojo Guerra Détection de familles d'astéroides : un problème de segmentation non superviséee, GRETSI 2005.

[PMNHD00] P.Wu, B.S. Manjunath, S.Newman, and H.D. Shin, « A texture descriptor for browsing and similarity retrieval », Signal processing : Image communication, vol.16, no.1, 2, pp :33-43,2000.

[PEN94] A. Pentland, R. Picard, and S. Sclaroff . Photobook: Tools for Content-Based Manipulation of Image Databases, SPIE Storage and Retrieval of Image & Video Databases II, Feb 1994..

[POL98] Fabio Policarpo, The Computer Image, ACM Press. Pages 298-308. 1998.

[PRIM57] Prim, R. Shortest connection networks and some generalizations, Bell Syst. Tech. J. 36, 1389–1401. 1957.

[PNHA84] S. Peleg. J. Naor, R. Hartley. and D. Avnir. Multiple resolution texture analysis and classification. IEEE Trans. Pattern Anl. Machine Intell, 6:518-523, 1984.

[PAPAGE98] Constantine Papageorgiou, Michael Oren et Tomaso Poggio. A General Framework for Object Detection. Dans Proceedings of the 6th IEEE International Conference on Computer Vision, pages 555{562, Bombay, Inde, janvier 1998.

[PL02] C. Palm et T.M. Lehmann. Classification of color textures by gabor filtering. Machine Graphics and Vision International Journal, 11(2) :195–219, 2002.

[PEN84] A. P. Pentland. Fractal-based description of natural scenes. IEEE Trans. Pattern Anal. Machine Intell, 6 :661-675, 1984.

[PF97] E. G. M. Petrakis and C. Faloutsos. Similarity searching in medical image databases. Knowledge and data Enginneering, 9(3) :435-447, 1997.

[PET02] E. G. Petrakis. Design and evaluation of spatial similarity approaches for image retrieval. Image Vision Comput. 20(1) :56-76, 2002.

[RUBNER99] Y. Rubner. Perceptual metrics for image database navigation. Rapport Technique CS-TR-99-1621, Stanford University, 1999.

[REM01] Remco C. Veltkamp, Mirela Tanase, "Content-Based Image Retrieval Systems: A Survey", Department of Computing Science, Utrecht University, Technical Report UU-CS-2000-34, March 8, 2001.

[RWMF74] R. A. Wagner and M. J. Fisher, " The string-to-string correction problem". Journal of the ACM, Vol. 21, No. 1, pp. 168-173, 1974.

[ROW00] Sam Roweis and Lawrence Saul.*Nonlinear dimensionality reduction by locally linear embedding*.*Science*, v.290 no.5500, Dec.22, pp.2323-2326. 2000.

[ROW01] Lawrence Saul and Sam Roweis. *An Introduction to Locally Linear Embedding*.Janary, 2001.

[ROS69] A. Rosenreld. Picture processing by computer. Academie Press, New York, 1969.

[RUI99] Yong Rui, Thomas S. Huang, and Shih-Fu Chang, Image retrieval: current techniques, promising directions and open issues, Journal of Visual Communication and Image Representation, Vol. 10, no. 4, pp. 39-62, April 1999.

[RTG98] Y. Rubner, C. Tomasi, et L. Guibas. The earth mover distance as a metric for image retrieval. Stanford University Technical Report, 1998.

[SO95] M.A. Stricker and M. Orengo. Similarity of color images . In SPIE, Storage and Retrieval for image Video Databases, pages 381-392, 1995.

[SC94] J.R. Smith and S.F. Chang. Transform features for texture classification and discremination in large image Databases. in IEEE Int. Conf. on image Proc. 1994.

[SC96] J.R. Smith and S.F Chang. Visualssek : a fully automated content based image query system. in ACM Multimedia Conference , pages 87-98, 1996.

[STC97] S. Sclaroff, L. Taycher, and L. La Cascia, Imagerover : A content based image browser for word wide web. In IEEE Workshop on content based access of Image ad video Libraries, 1997.

[SB91] M. J. Swain et D. H. Ballard. Color indexing. International journal of computer vision, 7(1):11–32, 1991.

[SMEU00] A. Smeulders, M. Worring, S. Santini, and A. Gupta. Content based image retrieval at the end of the early years. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(12):1349-1380, 2000.

[SHIMAL00] Jianbo Shi and Jitendra Malik. Normalized Cuts and Image Segmentation. IEEE Transactions on pattern analysis and machine intelligence, Vol. 22, No. 8, August 2000.

[SCO90] G.L. Scott and H. C. Longuet-Higgins. Feature grouping by relocalisation of eigenvectors of the proxmity matrix. In Proc. British Machine Vision Conference, pages 103-108, 1990.

[SEB00] H. Sebastian Seung & Daniel D. Lee.*The Manifold Ways of Perception*. (Cognition Perspectives in same issue). *Science*, v.290, no.5500, Dec.22, pp.2268-2269. 2000.

[SOZ05] H.T. Shen, B.C. Ooi, and X. Zhou. Towards effective indexing for very large video sequence database. In proceedings of the ACM SIGMOD. International conference on Management of Data, pages 730-741, Baltimore Maryland, USA ACM, 2005.

[SHA78] A. Iannino et S. D. Shapiro. A survey of the Hough transform and its extension for curve detection. Proc. IEEE PRIP'78, Chicago, mai juin, p. 32-38. 1978.

[SMI97] J. R. Smith and S. F. Chang, "Querying by color regions using the VisualSEEk content-based visual query system", M. T. Maybury, "Intelligent multimedia information retrieval", pp. 23-42, 1997.

[SMI96] J. R. Smith and S.-F. Chang, Querying by color regions using the VisualSEEk content-based visual query system, In Intelligent Multimedia InformationRetrieval. IJCAI, pages 159-173, 1996.

[SIN02] S. Singh and M. Singh , Evaluation of Segmentation and Texture Algorithms Combinations for Scene Analysis, IEEE Transactions on SMC, 2002.

[TAV03] Alireza Targhi Tavakoli, A.Shademan: Clustering of singular value decomposition of image data with applications to texture classification. In: VCIP. 972–979, 2003.

[TAV06] Targhi Tavakoli, A.Hayman, E.Eklundh, J.Shahshahani: The eigen-transform and applications. In: ECCV (1). 70–79, 2006.

[TEN00] J. B. Tenenbaum, V. De Silva and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. Science, 290 (5500), 2319-2323. 2000.

[TSLA00] T. S. Lai, "CHROMA: a photographic image retrieval system", PhD thesis, School of computing, engineering and technology, University of Sunderland, UK, 2000.

[TJ74] J.T. Tou and R.C. Gonzalez. Pattern Recognition Principles. Addison Wesley. Massachusetts. 1974.

[THI81] V. S. Alagar et H . Thiel. Algorithms for Detecting M-Dimensional objects in N-Dimensional Spaces. IEEE Trans Pattern Anal. and Mach. Intel., PAMI-3, No. 3,p. 245-256, mai 1981.

[TUR86] M. R. Turner discrimination by gabor functions. Biological Cybernetics, 55 : 71-82, 1986.

[TNP94] K. S. Thyagarajan. T. Nguyen, and C. Persons. A maximum likelihood approach to texture classification using wavelet transform. in IEEE Int. Conf. on Image Proc, pages 640-644, 1994.

[UNS95] M. Unser, Texture classification and segmentation using wavelet frames. IEEE Trans. on Image Processibng, 4(11) : 1549-1560, 1995.

[VAI01] Aditya Vailaya, Mário Figueiredo, Anil Jain, HongJiang Zhang. A Bayesian Framework for Semantic Classification of Outdoor Vacation Images, IEEE Trans. Image Processing, Vol. 10, No. 1, pp. 157-172, Jan. 2001.

[VP87] H.Voorhees and T.Poggio. Detecting textons and texture boundaries in natural images. In Inter. Proceedings of the first International comference on computer Vision . pages 250-258. 1987.

[VIOLA01] Paul Viola et Michael Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. Dans Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 511-518, Etats-Unis, decembre 2001.

[VIOLA03] Paul Viola, Michael Jones et Daniel Snow. Detecting Pedestrians Using Patterns of Motion and Appearance. Dans Proceedings of the 9th IEEE International Conference on Computer Vision, volume 2, pages 734{741, Nice, France, octobre 2003.

[VETTER95] J. Kovacevic M. Vetterli. Wavelets and Subband Coding. Prentice Hall, 1995.

[WDR76] J.S. Weszka, C. R. Dyer, and A. Rosenfeld. A comparative study of texture measures for terrain classification. IEEE Transaction on Systems, Man and cybernetics, 6(4) : 269-285, 1976.

[XJMB01] X. Jiang, A. Munger, and H. Bunke, "On median graphs :properties, algorithms, and applications", IEEE Transactions on pattern analysis and machine intelligence, Vol. 23, No. 10, pp. 1144-1151, 2001.

[XJMB99] X. Jiang, A. Munger, and H. Bunke, "Computing the generalized median of a set of graphs", Workshop on graph based representations, pp. 115-124, 1999.

[YLFC25] Y. Lee and F. Chin, "An iconic query language for topological relationships in GIS", Journal of geographical information systems, Vol. 9, No. 1, pp. 24-46, 1995.

[ZHANG03] F. Long, H. Zhang and D. Feng, "Fundamentals of content-based image retrieval", In D. Feng and H. Zhang, "Multimedia information retrieval and management: technological fundamentals and applications", Springer, pp. 1-26, 2003.

[ZUC76] S.W. Zucker. Toward a model of texture. Computer Graphics and Image Processing. 5 :190 202, 1976.