



**HAL**  
open science

# CARACTERISATION DE TEXTURES ET SEGMENTATION POUR LA RECHERCHE D'IMAGES PAR LE CONTENU

Adel Hafiane

► **To cite this version:**

Adel Hafiane. CARACTERISATION DE TEXTURES ET SEGMENTATION POUR LA RECHERCHE D'IMAGES PAR LE CONTENU. Interface homme-machine [cs.HC]. Université Paris Sud - Paris XI, 2005. Français. NNT: . tel-00097977

**HAL Id: tel-00097977**

**<https://theses.hal.science/tel-00097977v1>**

Submitted on 22 Sep 2006

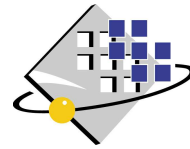
**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° D'ORDRE :



UNIVERSITE DE PARIS-SUD XI  
Faculté des Sciences d'Orsay



THESE DE DOCTORAT

*Ecole Doctorale «Sciences et Technologies de l'Information des  
Télécommunications et des Systèmes»*

Présentée par :

**Adel Hafiane**

Sujet :

**CARACTERISATION DE TEXTURES ET SEGMENTATION  
POUR LA RECHERCHE D'IMAGES PAR LE CONTENU**

Soutenue le 12 Décembre 2005 devant les membres du jury :

M. Subhasis CHAUDHURI	Professeur
M. Jean DEVARIS	Professeur
M. Nicolas LOMENIE	Maître de conférences
M. Roger MOHR	Professeur
M. Georges QUENOT	Chargé de recherche
M. Bertrand ZAVIDOVIQUE	Professeur



*À mes parents*





# Remerciements

Mes premiers remerciements s'adressent à Bertrand Zavidovique qui m'a encadré tout au long de ma thèse. Il a su me faire partager ses réflexions, tout en laissant une importante place à l'initiative personnelle. Je lui suis très reconnaissant d'avoir été disponible et attentif au déroulement de ce travail.

Je remercie Roger Reynaud de m'avoir accueilli au sein dans son équipe et pour tous les moyens mis à ma disposition pour mener à bien ce travail de thèse. Je salue également tous les membres du département AXIS.

Je remercie Subhasis Chaudhuri de m'avoir reçu dans son laboratoire pendant 3 mois et pour l'aide qui ma apporté sur ce travail.

Je tiens à remercier Roger Mohr d'avoir accepté de participer et présider mon jury de thèse. Je suis également reconnaissant à Jean Devars et Georges Quénot d'avoir apporter leur point de vu sur mon travail à travers leurs rapports sur la thèse. Je remercie aussi Nicolas Loménie d'avoir accepté d'examiner ce travail. J'ai apprécié particulièrement les remarques et les discussions des membres du jury qui mon permis d'améliorer la version finale de ce manuscrit.

Merci à tout les personnes qui m'ont aidé et entouré durant ma thèse.

Enfin merci à Nadia, oncles Ahmed et Ibrahim et à toute la famille.

---



# Résumé

Dans cette thèse nous avons élaboré puis automatisé une chaîne complète de recherche d'image par le contenu. Ceci nous a permis de définir une *sémantique limitée* relative à la satisfaction de l'utilisateur quant à la réponse du système. Notre approche est locale c'est-à-dire basée sur les régions de l'image. La décomposition en entités visuelles permet d'exhiber des interactions entre celles-ci et du coup faciliter l'accès à un niveau d'abstraction plus élevé. Nous avons considéré plus particulièrement trois points de la chaîne : l'extraction de régions fiables, leur caractérisation puis la mesure de similarité. Nous avons mis au point une méthode de type C-moyennes floues avec double contrainte spatiale et pyramidale. La classification d'un pixel donné est contrainte à suivre le comportement de ses voisins dans le plan de l'image et de ses ancêtres dans la pyramide. Pour la caractérisation des régions deux méthodes ont été proposées basées sur les courbes de Peano. La première repose sur un principe grammatical et la deuxième manipule le spectre par l'utilisation des filtres de Gabor. La signature de l'image requête ou cible consiste en une liste d'entités visuelles. La mesure de similarité entre entités guide l'appariement. Nous avons élaboré une méthode basée sur la mise en correspondance dans les deux sens, requête vers cible et vice versa, afin de donner indépendamment une grande priorité aux éléments qui se préfèrent mutuellement. Chaque partie du système a été testée et évaluée séparément puis ramenée à l'application CBIR. Notre technique a été évaluée sur des images aériennes (et ou satellitaires). Les résultats en terme de "rappel×précision" sont satisfaisants comparé notamment aux méthodes classiques type matrice de co-occurrence des niveaux de gris et Gabor standard. Pour ouvrir sur de futures extensions et montrer la généralité de notre méthode, la conclusion explique sa transposition à la recherche de situations en conduite automobile, au prix d'une adaptation limitée des paramètres.

---



# Abstract

This thesis describes the design and realization of a complete processing chain for content based image retrieval (CBIR). The study allows to define some *limited semantics* with respect to the user's satisfaction from the system response. The image is decomposed on visual entities to obtain interactions between them, allowing to reach higher levels of abstraction. We have addressed three points in the chain : reliable region-detection, region characterization and then similarity measure. We have modified a Fuzzy C-means by incorporating the spatial and multiresolution information into the objective function. Therefore, the classification of a given point is forced to follow both neighbors and ancestors in a pyramidal representation. Two methods are proposed which exploit Peano scans to coding region features. The first one is based on a grammatical representation of the pixels neighbourhood called motif. The second method modifies the spectrum before to apply Gabor filters. The image signature consists of a list of visual entities containing features. The similarity measure between two images turns into a graph matching problem. We have elaborated a technique that allows a bidirectional matching from query to target and vice versa. A high priority is assigned to those elements which prefer mutually. Each part of the system is evaluated and tested independently then incorporated into the CBIR application. The evaluation of CBIR in terms of "recall $\times$ precision" shows that the proposed methods perform better than classical ones, such as grey level co-occurrence matrix and Gabor filters. To open on further extensions and suggest the generality of our method, the conclusion deals with extending it to the situation assessment in car driving, with limited tuning of parameters.

---



---

# Table des matières

<b>Introduction</b>	<b>1</b>
<b>1 Etat de l'art</b>	<b>5</b>
1.1 Introduction	5
1.2 L'information visuelle	5
1.3 Le RIC standard	6
1.4 Les paradigmes de la recherche d'information visuelle	7
1.5 Extraction des caractéristiques	8
1.5.1 La couleur	8
1.5.2 La texture	10
1.5.3 La forme	14
1.5.4 Raisonnement spatial	15
1.5.5 Caractérisation de haut niveau	16
1.6 Mesure de similarité entre attributs	17
1.7 Techniques de mise en correspondance	21
1.8 Interaction homme machine	24
1.9 Les Systèmes RIC	24
1.10 Techniques de segmentation d'image	27
1.10.1 Les méthodes de regroupement	28
1.10.2 Techniques hiérarchiques	29
1.10.3 Techniques de partitionnement	31
1.10.4 Techniques de mélange de distributions	33
1.10.5 Techniques floues	34
1.10.6 D'autres méthodes, d'autres modèles	37
1.10.7 Mesure de validité	37
1.11 Bilan	40
<b>2 Caractérisation locale de la texture par des parcours récursifs</b>	<b>43</b>
2.1 Introduction	43
2.2 Analyse de la texture	43
2.3 Schéma général	44
2.4 Description de la texture	45
2.4.1 L'approche par mesures statistiques	45
2.4.2 L'approche fréquentielle	47
2.5 Les courbes de Peano	48

---



---

2.6	Les parcours locaux . . . . .	49
2.7	Le parcours optimal . . . . .	51
2.8	Description par les motifs . . . . .	52
2.8.1	Matrice de co-occurrence des motifs (MCM) . . . . .	52
2.8.2	Problème de translation . . . . .	53
2.8.3	Propriétés de MCM . . . . .	53
2.9	Description par valeur des parcours optimaux . . . . .	56
2.9.1	Matrice de co-occurrence des VPO . . . . .	57
2.9.2	Filtres de Gabor des VPO . . . . .	58
2.10	Evaluation des performances . . . . .	58
2.10.1	La base Outex . . . . .	59
2.10.2	Le classifieur des "K plus proches voisins" . . . . .	60
2.10.3	Paramétrage . . . . .	62
2.11	Résultats et discussion . . . . .	62
2.11.1	Classes de texture générale . . . . .	62
2.11.2	L'effet des transformations géométriques . . . . .	63
2.11.3	L'effet de la résolution . . . . .	65
2.11.4	L'effet de l'éclairage . . . . .	65
2.11.5	MCM sans recouvrement . . . . .	65
2.11.6	L'effet du nombre de KPP . . . . .	66
2.12	Conclusion . . . . .	66
<b>3</b>	<b>Extraction d'entités visuelles par décomposition en régions</b> . . . . .	<b>69</b>
3.1	Introduction . . . . .	69
3.2	Description par régions . . . . .	70
3.3	La segmentation d'image . . . . .	71
3.4	Particularité de la segmentation pour CBIR . . . . .	72
3.4.1	Algorithme de C-Moyennes . . . . .	72
3.4.2	Classification floue . . . . .	73
3.4.3	Algorithme de C-Moyennes Floues . . . . .	73
3.4.4	Corrélation Spatiale . . . . .	75
3.5	Le choix des caractéristiques . . . . .	76
3.5.1	Contribution du voisinage . . . . .	76
3.6	FCM avec contrainte spatiale . . . . .	77
3.7	La contrainte spatio-pyramidale . . . . .	79
3.7.1	La multirésolution . . . . .	79
3.7.2	FCM avec contrainte spatio-pyramidale . . . . .	80
3.8	Détermination automatique du nombre de classes . . . . .	83
3.9	Résultats et discussion . . . . .	85
3.9.1	Critères d'évaluation . . . . .	85
3.9.2	Evaluation et comparaison . . . . .	87
3.9.3	Mesure de validité . . . . .	93
3.9.4	Conclusion . . . . .	94

---

---

<b>4</b>	<b>Recherche d'image par composantes visuelles</b>	<b>99</b>
4.1	Introduction	99
4.2	Description du contenu	99
4.3	Extraction de descripteurs visuels locaux	100
4.4	Concept de la sémantique	101
4.4.1	Niveaux sémantiques	102
4.5	Les relations spatiales	102
4.6	Extraction de descripteurs spatiaux	104
4.7	Techniques de recherche	107
4.7.1	Requête par région	107
4.7.2	Requête par multiples régions	108
4.7.3	Recherche par multiples régions avec relations spatiales	113
4.8	Application	114
4.8.1	Implantation	114
4.8.2	Signature visuelle	114
4.8.3	Signature spatiale	115
4.8.4	Traitement hors-ligne	115
4.8.5	Traitement en ligne	115
4.8.6	Interface utilisateur	116
4.9	Evaluation des performances	116
4.10	Application au SIG	119
4.10.1	Base d'image	120
4.10.2	Recherche partielle	120
4.10.3	Recherche par multiples régions	122
4.10.4	Comparaison entre segmentation humaine et automatique	130
4.10.5	Recherche avec contraintes spatiales	132
	<b>Conclusion et perspectives</b>	<b>135</b>
	<b>Annex : Interface homme machine</b>	<b>143</b>
A.1	Notion	143
A.2	Schéma de principe	143
A.2.1	Barre de menu	144
A.2.2	Paramétrage et recherche	147
	<b>Bibliographie</b>	<b>151</b>

---



# Table des figures

1.1	Images d'une base généraliste . . . . .	8
1.2	Espaces de couleur RGB et HSI . . . . .	9
1.3	Des textures différentes . . . . .	11
1.4	Exemple d'un arrangement spatial avec 2D string . . . . .	15
1.5	Orientation spatiale entres objets : a gauche l'image requête et à droite image de la base . . . . .	17
1.6	Problème de la mise en correspondance . . . . .	21
1.7	Exemple d'appariement IRM . . . . .	23
1.8	Exemple d'un nuage de point dans un espace 3D . . . . .	29
1.9	Exemple d'un clustering hiérarchique . . . . .	30
2.1	Système de classification . . . . .	44
2.2	Matrice de co-occurrence des niveauuw de gris pour deux textures différentes . . . . .	47
2.3	De gauche à droite un banc des filtres de Gabor orientés respectivement en $0^\circ$ , $45^\circ$ , $90^\circ$ , $135^\circ$ . . . . .	48
2.4	(a) Balayage de Peano en motif "Z" (b) Balayage horizontal classique . . . . .	49
2.5	Tous les motifs possibles pour 4 points . . . . .	50
2.6	Motifs représentant tous les cas possibles . . . . .	51
2.7	<b>a)</b> l'image motif originale <b>b)</b> l'image motif après un décalage horizontal à gauche de l'image originale . . . . .	53
2.8	<b>a)</b> Exemple de texture "Brodatz" <b>b)</b> A gauche l'analyse en composantes principales; les numéros sur la figure correspondent aux classes ci-dessus. A droite les valeurs propres de la matrice de covariance . . . . .	55
2.9	<b>a)</b> Différentes classes d'images aériennes <b>b)</b> Les points des classes représentés suivant les deux axes principaux . . . . .	56
2.10	Les valeurs de MCM (6 motifs) en fonction des colonnes et des lignes correspondant aux images ci-dessus . . . . .	56
2.11	<b>a)</b> image originale <b>b)</b> image motifs <b>c)</b> image VPO . . . . .	57
2.12	<b>a)</b> <b>b)</b> image originale/image vpo avec egalisation de l'histogramme . . . . .	57
2.13	<b>a)</b> <b>b)</b> FFT d'image original et image VPO . . . . .	58
2.14	Système d'acquisiton Outex . . . . .	60
2.15	Echantillons de classes de texture Outex . . . . .	61
3.1	Image de synthèse représentant une scène urbaine . . . . .	70
3.2	Decomposition du voisinage 3x3 en un ensemble de 4 groupes . . . . .	77

3.3	Représentation pyramidale . . . . .	80
3.4	A gauche un Z arbre, à droite l'arbre binaire associé . . . . .	80
3.5	Contrainte en multirésolution par fonctions d'appartenance . . . . .	81
3.6	Mesure de la qualité de segmentation pour 4 valeur de bruit gaussien(a) contrainte spatiale (b) contrainte pyramidale (c) les deux contraintes avec les même valeurs . . . . .	88
3.7	Mesure de qualité de segmentation pour un bruit de grenaille (a) contrainte spatiale (b) contrainte pyramidale (c) les deux contraintes avec les même valeurs . . . . .	89
3.8	Qualité de segmentation pour un bruit uniforme (a) contrainte spatiale (b) contrainte pyramidale (c) les deux contraintes avec les même valeurs . . . . .	90
3.9	Comparaison entre 3 contraintes : spatiale, pyramidale et spatio-pyramidale pour des images satellitaires . . . . .	91
3.10	Comparaison entre 3 contraintes : spatiale, pyramidale et spatio-pyramidale pour des images de la base Berkley . . . . .	92
3.11	Résultat de segmentation d'images synthétiques . . . . .	95
3.12	Segmentation d'images satellitaires et de télédétection . . . . .	96
3.13	Résultat de segmentation pour des images naturelles . . . . .	97
3.14	Mesure de validité pour images synthétiques . . . . .	97
3.15	Mesure de validité pour images naturelles . . . . .	98
4.1	Decomposition d'une image en entités visuelles . . . . .	101
4.2	Les niveaux d'abstraction de la sémantique du contenu . . . . .	103
4.3	Relations topologiques . . . . .	103
4.4	Orientation entre deux objets dans le plan . . . . .	104
4.5	Représentation des quatres situations basiques . . . . .	106
4.6	Un exemple d'une situation réelle . . . . .	106
4.7	Comparaison exclusive des regions . . . . .	109
4.8	Deux exemples d'extraction des minimums lignes et colonnes . . . . .	111
4.9	L'architecture du système de recherche . . . . .	114
4.10	Interface utilisateur . . . . .	117
4.11	A ensemble d'éléments trouvé et B ensemble d'éléments pertinents . . . . .	118
4.12	Exemple de requête par région (segment marqué) . . . . .	121
4.13	Comparaison des performances pour une recherche de région . . . . .	122
4.14	Un exemple de requête avec ABIP . . . . .	123
4.15	Comparaison des résolutions. Légendes U : une seule résolution, D : deux résolutions. (a) MCM (b) GLCM (c) FGVPO (d) GABOR . . . . .	125
4.16	Comparaison des méthodes d'appariement : (a) MCM (b) GLCM (c) FGVPO (d) GABOR . . . . .	127
4.17	Exemple de rotation. (a) image originale (b) rotation d'un angle de 90° . . . . .	128
4.18	Comparaison des performances pour la rotation. La courbe NR : requête sans rotation, la courbe R : requête avec rotation. (a) MCM (b) GLCM (c) FGVPO (d) GABOR . . . . .	129
4.19	Différence entre deux courbes rappel/précision . . . . .	130
4.20	Comparaison entre l'homme et la machine en terme de segmentation . . . . .	131

---

4.21	Un exemple de requête avec relations spatiales . . . . .	133
4.22	Entrée dans un espace clos . . . . .	138
4.23	Sortie d'un espace clos . . . . .	139
4.24	Galerie d'images contenant des passages piéton avec une segmentation par SPFCM . . . . .	139
4.25	Un exemple d'une requête contenant un passage piéton en utilisant la technique ABIP. Les résultats sont rangés du droite à gauche et du haut en bas. . . . .	140
4.26	Deuxième exemple de requête choisie des résultats précédents . . . . .	141
A.1	Schema de principe de l'interface graphique . . . . .	144
A.2	Menu de l'interface . . . . .	144
A.3	Selecteur de répertoire . . . . .	145
A.4	Selecteur de fichier . . . . .	146
A.5	Définition du nombre d'images à afficher en lignes/colonnes . . . . .	146
A.6	Choix des images pertinentes . . . . .	147
A.7	La courbe rappel/précision . . . . .	148
A.8	Exemple de différents paramètres du système . . . . .	149

---



---

# Liste des tableaux

1.1	Un exemple de liste de préférence . . . . .	22
2.1	Comparaison des caractéristiques pour chaque classe de la catégorie Outex_TC_00000 . . . . .	64
2.2	Comparaison des caractéristiques pour chaque catégorie de problème . . . . .	65
2.3	Comparaison des deux méthodes de calcul des MCM, avec recouvrement et sans recouvrement . . . . .	66
2.4	Variation du nombre de KPP . . . . .	66
4.1	Un exemple de relations spatiales . . . . .	107
4.2	Mesure des performances pour requêtes en région . . . . .	122
4.3	Résultat avec une seule résolution pour les images pertinentes . . . . .	124
4.4	Résultat avec deux résolutions pour les images pertinentes . . . . .	124
4.5	Mesures des performances avec différentes méthodes de matching . . . . .	126
4.6	Evaluation de l'efficacité du système sous l'influence de rotation . . . . .	128
4.7	Comparaison des performances de segmentation manuelle et automatique . . . . .	131





# Introduction

Avec la démocratisation et le développement technologique des moyens d'acquisition, de calcul et de stockage l'utilisation de l'image a connu un essor considérable durant ces deux dernières décennies. Aujourd'hui le stockage de l'image sous une forme de document numérique concerne aussi bien les professionnels que le grand public. Il a de multiples applications : les systèmes géographiques, l'imagerie médicale, la video surveillance, la robotique, l'archivage personnel. Généralement les images sont compressées puis archivées dans des bases de données, généralistes ou spécialisées. Ainsi les bases de données contiennent des quantités d'images de plus en plus grandes. La problématique nouvelle est donc apparue d'extraire une information pertinente, selon un besoin qui varie dynamiquement et qui peut être spécifié de manière peu précise, hors d'une telle quantité croissante de documents.

La recherche d'information dans de telles bases nécessite une organisation adéquate et des outils efficaces. L'approche classique consiste en l'annotation par le texte : les images sont étiquetées par des mots clefs décrivant leurs contenu. L'extraction d'une image est donc ramenée à la manipulation de chaînes de caractères. Les méthodes textuelles disponibles sont jugées suffisamment abouties pour offrir des réponses satisfaisantes dans de nombreuses situations, c'est ce que réalisent les moteurs de recherche bien connus sur le web. Cependant rien ne permet d'affirmer que deux personnes décrivent une image avec les mêmes mots clefs. Il est difficile d'extraire des règles d'annotation textuelle d'image : la difficulté est en réalité équivalente à celle de définir des catégories d'images, lesquelles dépendent de l'utilisation envisagée pour les images extraites. Le passage d'une représentation visuelle à une représentation textuelle engendre donc des pertes d'information. De plus, la tâche d'étiquetage est pénible en pratique quant il s'agit d'annoter un grand nombre d'images.

La recherche d'images par le contenu (*Content Based Image Retrieval* CBIR ou encore RIC en Français) est une alternative intéressante. Le CBIR cherche à représenter le contenu d'une manière automatique, souvent à l'aide d'attributs visuels tels la couleur, la texture, ou la forme. Des caractéristiques codant ces attributs sont calculées à bas niveau pour en définir une signature. Le but est d'identifier les classes d'images cibles similaires à celle de la requête au sens du procédé de codage. La mesure de ressemblance est souvent fondée sur le calcul d'une distance dans l'espace des caractéristiques. Même si le contenu

---

image est complexe et difficile à définir (*une image vaut mille mots*). Les descripteurs visuels devraient en extraire des propriétés à la fois communes à des images similaires et discriminantes pour celles qui sont visuellement différentes. A l'usage le procédé est apparu suffisamment efficace pour retrouver des documents pertinents dans de nombreux cas. Il n'en reste pas moins que la sémantique adressée est faible, rendant par exemple ces cas aussi imprédictibles que nombreux. Dans le but d'améliorer la recherche certaines techniques mobilisent des descripteurs de niveau d'abstraction plus élevé dont les possibles relations a priori entre objets projetés dans l'image. Cette tendance favorise un degré de discrimination supplémentaire mais sans encore résoudre le problème. La question devient donc : "pour atteindre une sémantique plus élevée, comme celle du but poursuivi par l'utilisateur, faut il un traitement de haut niveau (compréhension, apprentissage) ?" Nous nous sommes inscrits dans cette problématique en essayant de définir une *sémantique limitée* susceptible de satisfaire l'utilisateur par une réponse statistiquement proche de la sienne dans une application bien circonscrite. Pour preuve de sa validité, nous nous attachons à automatiser une chaîne complète de traitements dont la réponse reste satisfaisante hors intervention humaine.

L'analyse automatique à plusieurs niveaux d'abstraction nécessite par exemple une décomposition de l'image en entités visuelles, la caractérisation de ces entités, l'extraction d'une propriété relationnelle entre entité, la définition d'une mesure de ressemblance puis le test du processus complet dans des cas variés et représentatifs. La première difficulté est dans la détermination d'un algorithme capable d'extraire les composantes image en respectant un critère de cohérence visuelle suffisante. La deuxième réside dans le choix d'attributs puis dans la définition de caractéristiques pour coder les composantes précédentes. On peut ensuite doter le système d'une caractéristique plus abstraite, par exemple captant l'interaction spatiale entre entités. Le dernier point avant expérimentation porte sur la mise en oeuvre d'une mesure de ressemblance entre requête et cible qui exploite les éléments précédents. L'automatisation de la chaîne implique donc de traiter successivement plusieurs problèmes de natures différentes.

L'image est d'abord considérée comme un ensemble d'objets définis par des régions distinctes. Les régions correspondent généralement à des zones visuellement homogènes. Elle sont isolées par les frontières qui les séparent. Le but de la segmentation d'image est d'identifier ces frontières en même temps qu'attribuer chaque pixel à une classe. La classe se résume à ce stade en l'étiquette d'un groupe de pixels vérifiant un critère de ressemblance. Ce critère est maximisé intra groupe et maximisé inter groupes. Dans les images du monde réel les régions que seraient les objets projetés ne sont pas assez contrastées pour être isolées sans erreur. En même temps on dispose rarement d'information précise sur les bords. Le résultat est donc un ensemble de régions imprécises et peu sûres à cause des plausibles échecs de la classification. On rencontre deux type d'erreurs : sous (regroupements intempestifs de régions) ou sur (éclatement de région) segmentation. On cherchera donc à minimiser l'incertitude de sous/sur segmentation.

Il faut ensuite déterminer un attribut pertinent accentuant encore la séparation entre classes mais robuste d'une image à l'autre du même objet. Une telle séparation dépend

---

du type d'image ou d'application visé, deux régions peuvent avoir par exemple une distribution de couleur proche mais une texture différente. La texture est omniprésente dans les images naturelles ou satellitaires dont les surfaces présentent une certaine rugosité. La texture a des propriétés discriminatoires élevées mais elle est difficile à caractériser. Elle peut être considérée comme distribution stochastique, arrangement ou répétition d'un ou plusieurs motifs conduisant à des modèles structurels ou fréquentiels. La résolution de l'analyse est donc importante pour l'appréciation de la texture dans la mesure où elle influe sur le choix de l'éventuelle primitive autant que sur l'arrangement spatial. Par exemple à une échelle grossière les fenêtres d'un bâtiment en constituent une texture répétitive en damier marquant la façade, et à une échelle plus fine les veines du bois ou les pores du ciment caractérisent la porte et le mur conduisant à retrouver de même l'immeuble mais selon un modèle différent. Ce problème est également rencontré dans l'imagerie satellitaire où de surcroît l'image peut avoir été acquise à différents niveaux de résolution. Prendre en compte l'application dans la caractérisation de la texture conduit donc à des descripteurs discriminants à différentes échelles des différentes classes, ce qui pourrait également diminuer la complexité calculatoire tout en contraignant la précision de reconnaissance.

Après la décomposition en régions et leur caractérisation il faut mesurer une similarité. Lorsque les deux entités sont représentées par des vecteurs la solution est le calcul d'une distance dans l'espace caractéristique. Cependant un objet au sens humain du terme recouvre généralement plusieurs régions différentes. Pour comparer deux objets il est donc primordial de prendre en considération les liens entre leurs régions. Et le calcul de ressemblance devra impliquer toutes les régions de l'objet. On a donc un ensemble de composantes d'une part côté requête et d'autre part du côté cible un ensemble de même type. Si à nouveau un immeuble se décompose en fenêtres et murs, pour trouver un bâtiment similaire les entités de même nature doivent se correspondre. On cherchera les couples de régions (requête, cible) qui ont des caractéristiques semblables. Un coût total de cette mise en correspondance est calculé pour évaluer la ressemblance. Ceci relève de l'appariement de graphes. Il existe plusieurs méthodes connues pour produire des résultats satisfaisants sur entités bien déterminées et précises. Dans le cas présent la segmentation automatique est comme déjà dit source d'erreurs et d'imprécision. La mise en correspondance classique de graphes, notamment biparties, supporte mal une telle approximation qui résulte en une piètre estimation du coût de l'appariement global. Ce dernier devra prendre en considération ces mauvaises conditions.

Notre approche a suivi les trois étapes précédentes, d'où le plan du mémoire

Le premier chapitre est consacré à la présentation de différentes notions et méthodes existantes liées au contexte de notre étude. Il est décomposé en deux parties : la première concerne les principes fondamentaux de la recherche d'image par le contenu et la deuxième étudie les méthodes de segmentation. Nous présentons les attributs usuels décrivant le contenu. Nous exposons alors les différentes techniques utilisées pour caractériser ces attributs. Puis nous présentons le raisonnement spatial comme supportant une représentation du contenu d'un niveau plus élevé. La mesure de similarité est analysée selon deux points de vue : les possibles distances dans l'espace caractéristique puis trois mé-

---

thodes clefs de la mise en correspondance. La fin de la première partie présente quelques systèmes CBIR populaires. La deuxième partie trace un état de l'art sur les méthodes de segmentation et plus précisément les méthodes de classification par regroupement qui nous ont intéressés. Nous décrivons les principales méthodes et leur principe fondateur dans le but de mieux expliciter les différences entre ces méthodes.

Le deuxième chapitre est dédié à l'étude et la présentation de méthodes de caractérisation de la texture. Nous nous intéressons aux techniques statistiques et fréquentielles. En premier lieu nous rappelons quelques fondements théoriques et nous explicitons les différents choix adoptés. La deuxième partie de ce chapitre est consacrée à l'étude expérimentale des techniques que nous prenons pour les évaluer et les comparer avec des méthodes plus classiques dans différentes conditions. Nous sommes attachés à travailler sur des images de bases connues pour la texture et avec un Benchmark pour la comparaison avec d'autres caractéristiques.

Dans le troisième chapitre, nous présentons deux méthodes de détection de régions. Ces méthodes doivent permettre une détection précise en minimisant le plus possible sous et sur segmentation. Nous justifions d'abord la nécessité selon nous d'une décomposition en régions. Puis nous exploitons quelques algorithmes classiques de regroupement, notamment les C-moyennes floues, que nous améliorons. Nous leur rajoutons deux types de contraintes pour adapter l'algorithme à notre application CBIR tout en améliorant la segmentation. Ce type d'algorithme nécessite un nombre de classes prédéterminé, nous proposons une mesure pour déterminer ce nombre d'une manière entièrement automatique. La fin du chapitre est consacrée également aux expérimentations sur différents types d'images.

Le quatrième chapitre concerne la technique de recherche d'image. Les images sont supposées avoir été segmentées par les techniques présentées dans le chapitre 3. Nous définissons deux types de descripteurs, visuel et spatial. Les caractéristiques visuelles retenues sont celles proposées au deuxième chapitre. La description spatiale est à nouveau du type relations floues basées sur des variables linguistiques. Un résumé de l'image est donc obtenu. Dans ce chapitre nous proposons une méthode de mise en correspondance entre régions susceptible de compenser des erreurs de segmentation. Deux applications sont visées, images satellitaires et images de route. L'évaluation et la comparaison sont effectuées selon la méthode "Trec Eval".

En conclusion, nous dressons un bilan de ce travail et amorçons un nouveau développement autour de l'utilisation du CBIR dans un véhicule intelligent afin d'aider à la décision dans des situations particulièrement ambiguës.

---

# Chapitre 1

## Etat de l'art

### 1.1 Introduction

L'objectif de ce chapitre est de dresser un état de l'art sur les méthodes de recherche d'image par le contenu (RIC) et la segmentation qui en est souvent une étape préliminaire. Etant donné la riche littérature de ces deux domaines la liste des références n'est pas exhaustive. Nous avons cité les méthodes qui nous semblaient pertinentes et qui permettent d'étayer le travail de cette thèse.

Le chapitre est donc explicitement divisé en deux parties. La première est consacrée aux principales approches utilisées dans les systèmes RIC. Nous étudions successivement la caractérisation visuelle et spatiale, la mesure de similarité, la mise en correspondance de régions et terminons par quelques systèmes populaires.

La deuxième partie s'intéresse à la segmentation de l'image en régions. Après un bref rappel des principales approches nous détaillons les méthodes de regroupement utilisées en segmentation.

### 1.2 L'information visuelle

Aujourd'hui avec le développement des systèmes multimédia ou 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 évoquer plusieurs significations en passant d'un humain à l'autre. Elle peut également présenter différentes significations à différents niveaux : analyse, description, reconnaissance et

---

interprétation[Sha86]. La recherche d'information (RI) couvre le traitement de documents numériques impliquant la structure, l'analyse, le stockage et l'organisation des données [Sal68]. Dans le passé le terme "recherche d'information" était lié au concept de l'information textuelle [JW77]. Actuellement RI pourra être de type textuel, visuel ou autre. 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 aux systèmes de RIC (Recherche d'image par le contenu) ces dernières années. Contrairement aux méthodes de recherche basées sur le texte, la RIC est souvent confrontée au problème de la sémantique car l'image est plus compliquée à interpréter que le texte. Certains systèmes de RIC associent le texte à l'image pour mieux réduire le fossé sémantique.

### 1.3 Le RIC standard

L'objectif de n'importe quel système RIC est de satisfaire la requête d'un utilisateur par la pertinence des résultats. Comme l'accès direct à un document via sa pure sémantique est impossible, les systèmes RIC traditionnels s'appuient sur un paradigme de représentation de bas niveau du contenu de l'image, par la couleur, la texture, la forme,...etc. La recherche d'image se fait ainsi par comparaison de caractéristiques. Les principales fonctions d'un RIC selon Smeulders et al [SWS+00] sont les suivantes :

- L'analyse et la représentation du contenu des données source 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 parce que les images sont traitées d'une manière séquentielle dans la base de données. Mais elle est exécutée en temps mort (*off-line*)
  - L'indexation et le stockage des caractéristiques. Elle devra être "scalable" pour répondre à la variation en taille des bases et rapide pour les nécessités de type temps réel. Ces contraintes requièrent une stratégie de recherche pour naviguer efficacement en bases de données. Les techniques modernes de gestion de base multimédia incorporent de tels concepts.
  - L'analyse et représentation de la requête utilisateur sous une forme compatible avec celle de la base. Cette opération est analogue à celle de la première étape mais appliquée seulement à l'image requête.
  - La mesure de similarité entre les image requête et source, mesure généralement basée sur une distance.
  - L'interface utilisateur. C'est la vitrine du système permettant la présentation des résultats ainsi que l'interaction avec l'humain. Elle doit être intuitive et simple pour offrir à l'utilisateur le confort nécessaire à une utilisation souple et efficace. Cette partie sert également lors du réglage des paramètres système ou de l'évaluation par retour de la pertinence.
-

## 1.4 Les paradigmes de la recherche d'information visuelle

Le traitement d'une requête dans un système de recherche dépend 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 types les plus utilisés :

### Requête par 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 connaissances approfondies pour manipuler le système. Elle est donc bien adaptée à un utilisateur non spécialiste.

### Requête par crayonnage (*Sketch*)

L'utilisateur 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 entraîne plus d'interaction avec l'utilisateur.

### Requête par caractéristique

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

### 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 indexées en groupes de pertinence. Chaque groupe définit un type ou un domaine dont l'image relève, par exemple végétation ou animal,...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 tâche à fort caractère subjectif ou expert.

---



## 1.5 Extraction des caractéristiques

L'extraction des attributs est l'opération la plus élémentaire dans un système de recherche d'information [RHC99]. Les caractéristiques peuvent être de haut ou bas niveau ; les attributs intuitifs tels que la couleur, la texture et la forme sont les plus employés dans le RIC à cause de leur fort pouvoir discriminant. En fait, il n'existe pas une caractéristique meilleure qu'une autre, l'optimalité dépend de la nature de l'image et du domaine d'application. Les caractéristiques d'une image peuvent être décrites comme [GR95] :

- *Caractéristiques physiques ou primitives* : sont des primitives de nature tangible qui peuvent être exprimées d'une manière quantitative. Elles sont obtenues souvent par des opérations de bas niveau. La couleur, la texture, la forme et les surfaces de régions sont autant d'exemples de caractéristiques physiques.
- *Caractéristiques logiques* : ces caractéristiques impliquent une opération de haut niveau à sémantique plus élevée. Par exemple l'association de l'étiquette "Ciel" à une région de l'image est considéré comme une caractérisation logique, même si elle ne résulte que de l'association d'une couleur et d'une position relative.

### 1.5.1 La couleur

La "couleur" est devenue un attribut largement utilisé dans les systèmes opérationnels de recherche d'image par le contenu. Elle facilite l'extraction et l'identification d'un objet dans une scène [SCZ01]. Typiquement les images sont codées sur trois canaux contre un seul pour les monochrome. Il semble que son efficacité à ce stade soit liée au fait que l'être humain peut distinguer des milliers de couleurs et seulement 24 niveaux de gris [GW02].



FIG. 1.1 – Images d'une base généraliste

Plusieurs études ont été réalisées sur l'identification d'espaces colorimétriques plus discriminants [SO95, WG01]. 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 [GMD+97, GS99, GvdBSG01]. D'autres espaces également fréquents dans le domaine revendiquent d'être perceptuellement uniformes et indépendants de l'intensité tels que CIE XYZ et CIE LUV [GS97, STC97]. Là encore ce sont des modèles de représentation et il n'existe pas un espace de couleur idéal, on trouve une comparaison entre les

espaces de couleur ainsi que leurs caractéristiques et une analyse avantages/inconvénient dans [KSGA96]

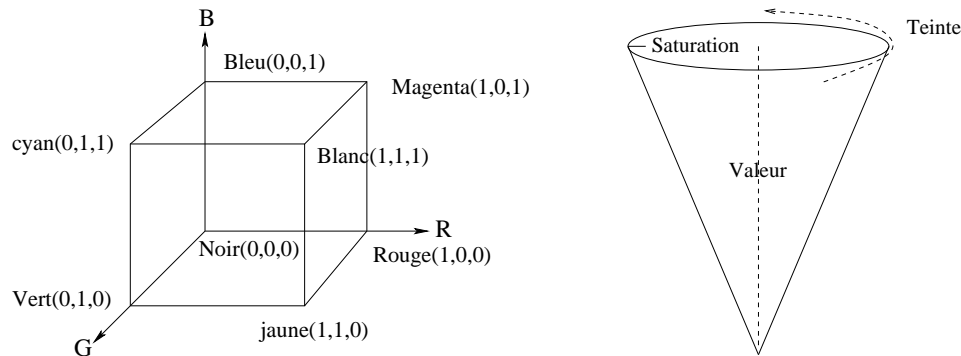


FIG. 1.2 – Espaces de couleur RGB et HSI

## L'espace RGB

Les composantes rouge, vert et bleu sont les trois couleurs primaires dont les autres dérivent. Correspondant à la perception physiologique initiale, le format RGB est également le plus adapté techniquement au traitement et surtout à la visualisation d'image parce que les dispositifs CRT (affichage numérique) utilisent ces trois mêmes composantes. L'espace RGB, considéré comme vectoriel, est souvent représenté sous forme d'un cube figure 1.2. Une image en format RGB est composée de trois plans images de couleurs rouge, vert, bleu. Si chaque couleur est codée sur huit bits, la profondeur est de 24 bits soit  $(2^8)^3$  couleurs possible. Les systèmes RIC n'utilisent pas toute cette plage, l'espace couleur est quantifié pour un nombre beaucoup plus limité de valeurs mais impliquant tous les axes (eg. 216 ( $6^3$ ) pour 6 valeurs par axe).

## L'espace HSI

Cet espace a été introduit pour une représentation intuitive de la couleur et pour se rapprocher de la perception humaine. Trois caractéristiques de couleurs sont définies : la teinte, la saturation et l'intensité. Le cône de la figure 1.2 illustre un modèle de l'espace HSI. La teinte représente la couleur pure (i.e. la longueur d'onde), sur le cône elle constitue l'angle entre une ligne de référence et la couleur considérée. La saturation indique la quantité de blanc dans la couleur pure. L'intensité est l'axe joignant le point d'intensité la plus faible (noir) au point d'intensité la plus forte (blanc), et représente la quantité de lumière contenue dans une couleur. L'espace HSI est facilement quantifiable, la teinte est l'information la plus signifiante pour la couleur, sa quantification sera donc en général plus fine que celle des autres composantes. Typiquement 162 couleurs sont retenues avec 18 cases pour la teinte et 3 pour saturation et intensité ( $18 \times 3 \times 3$ ).

## Les espaces XYZ et LUV de la CIE

La CIE (Commission Internationale de l'Eclairage) a défini un espace de couleur nommé XYZ. Y contient la luminance, X et Z l'information de chrominance. Cet espace n'est pas uniforme pour la perception humaine. Chaque axe est quantifié avec la même précision. Un autre espace dérivé de ce dernier par transformation projective est nommé LUV. La composante L représente l'information d'intensité et U et V représentent la chrominance.

## Les espaces YUV, YIQ et UCbCr

Les espaces YUV, YIQ et YCbCr sont trois espaces semblables. YUV est utilisé pour le codage des couleurs dans le système de télévision PAL, YIQ pour le système NTSC. Dans ces espaces Y indique l'intensité, UV et IQ sont les deux composantes de chrominance. Cb et Cr correspondent aux complémentaires du rouge et du bleu.

## Histogramme de couleur

L'histogramme des couleurs exprime la distribution statistique de celles ci dans l'image. Ce type d'histogramme est calculé typiquement sur un espace caractéristique quantifié. Chaque valeur de caractère dans l'histogramme représente donc un rang de couleur dans la palette. L'histogramme a été introduit pour la première fois en RIC par [SB91], depuis il est très utilisé à cause de sa simplicité de calcul, son invariance aux changements d'échelle et aux transformations géométriques. Les inconvénients majeurs de l'histogramme sont la perte de toute information spatiale dont la texture ou la forme. Par exemple un histogramme d'un tapis rouge peut être très proche de celui d'une porte rouge ou d'une voiture rouge. Des méthodes alternatives ont été proposées pour augmenter l'efficacité de l'histogramme. Pour en citer quelques unes : les moments de la couleur [SS94, RGB99], les constantes de couleur [FSN<sup>+</sup>95, WG01], la signature couleur [KY98], les blobs [CCM<sup>+</sup>97] et le vecteur cohérent de couleur [PZM97].

### 1.5.2 La texture

La texture est une caractéristique intuitive facile à reconnaître mais difficile à définir. Des modèles de la perception humaine ont été proposés tels que celui de Tamura et al [TMY78]. On s'accorde sur six propriétés basiques : *coarseness*, *contrast*, *directionality*, *linelikeliness*, *regularity* et *roughness*. D'un point de vue sémantique la définition de Tamura semble plus fiable que d'autres mesures plus classiques issues du Traitement de signal ou inspirés par la théorie de l'information comme par exemple l'énergie du spectre de la

---

texture ou l'entropie..., ceci se vérifie notamment dans les systèmes RIC. Pour d'encore meilleurs résultats cependant les systèmes RIC associent différentes méthodes d'analyse texturale. On les classe en quatre principales catégories[TJ93] : statistique, basée sur un modèle, fréquentielle et géométrique.



FIG. 1.3 – Des textures différentes

### Méthodes statistiques

Ce sont les méthodes basées sur des évaluations quantitatives de la distribution de niveaux de gris ou d'autres primitives, comme les micro-contours ou les maxima locaux. La matrice de co-occurrence des niveaux de gris GLCM (*Grey Level Co-occurrence Matrix*) [HSD73] est une distribution de ce genre. Souvent on en extrait des paramètres statistiques tels que la moyenne, la variance, la corrélation, l'entropie, des moments d'ordre supérieur... (voir chapitre 2).

Dans la littérature on trouve beaucoup de travaux qui traitent des matrices de co-occurrence. Davis et al [DJA79, DCA81] proposent une version généralisée impliquant des prototypes structurels comme les contours et les lignes. Sun et al [SW83] calculent une GLCM sur un voisinage normalisé. Gotlieb et Kreyszig [GK90] ont réalisé une étude statistique sur la GLCM en concluant que le contraste, l'entropie et l'inverse du moment des différences ont un grand pouvoir discriminatoire. D'autres travaux exploitent des relations entre GLCM et les champs de Gibbs, par exemple démontrant que l'histogramme du spectre est équivalent aux champs de Markov mais avec moins de paramètres à calculer [PE92, EP94, ZLW00]. Kovalev et Petrou [KP96] présentent une méthode de mesure des relations entre structures élémentaires de l'image pour obtenir des caractéristiques invariantes aux rotations et symétries. Une nouvelle approche par Al-Janobi [AJ01] associe les propriétés de GLCM et le spectre de texture, l'avantage en est l'indépendance par rapport à la dynamique des niveaux de gris. La statistique d'ordre supérieur a été également utilisée dans la caractérisation de la texture pour diverses applications. Par exemple Murino et al. [MOP98] l'emploient pour obtenir une invariance au bruit Gaussien. Johnson et al. calculent les statistiques de l'image en utilisant la réponse d'un filtre "biologique" dans des propriétés du premier et second ordre[JB04].

## Méthodes Géométriques

Elles sont dédiées à l'analyse structurelle basée 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 placement appliquées 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 idéal. 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 différentes 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 considèrerait 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 chaîne dans un langage dont les symboles sont les primitives.

## 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 multirésolution .

Les modèles MRF traitent l'image comme une réalisation d'un champs 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 entres 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 autorégressive models* (SAR)) est l'approche le plus utilisée dans la modélisation et la reconnaissance de la texture [CK85, MJ92]. L'un des avantage des modèles AR est leur faible complexité en terme de paramétrage par rapport au MRF.

Plusieurs surfaces naturelles ont un aspect à la fois statistique 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 [TJ93]. Parmi les travaux dont le but est de caractériser les textures par la dimension fractale citons [PNHA84, Pen84, KBT01]

## 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 des 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, DH95, 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 multirésolution. 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] 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. Thyagarajan et al. [TNP94] et Kundu et al. [KC92] lui associent la matrice de co-occurrence pour extraire à la fois les propriétés statistiques et fréquentielles. Pour ce qui est de filtrage plus classique, Jernigan 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.

## 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. Connors et Harlow [CH80] présentent une étude comparative de quatre différentes méthodes SGLDM (*Spatial Grey Level Dependence Method*), GLDM (*Grey Level Difference Method*), GLRLM (*Grey Level Run Length Method*)

---

et PSM (*Power Spectral Method*). Les auteurs trouvent SGLDM la plus performante. Du-Buf et al [BKS90] évaluent des techniques de types GLCM, extrema des niveaux de gris, dimension fractale et décomposition spectrale. Les trois premières affichent les meilleurs résultats. Ohanian et Dubes [OD92] comparent et évaluent quatre types de méthode basées sur les matrices de co-occurrence, les champs de Markov, les filtres de Gabor et la représentation fractale. Dans leurs tests, menés sur un groupe d'images synthétiques et naturelles, les matrices de co-occurrence exhibent les meilleures performances. Ojala et al. [OPH96] étudient les performances des techniques de GLDM, CSCM (*center-symmetric covariance measures*), LTM (*Laws texture method*) et LBP (*local binary patterns*). Testés sur des images de Brodatz <sup>1</sup> LBP a obtenu le meilleur score en terme de classification de texture. D'autres évaluations ont été consacrées aux méthodes fréquentielles. Chen et Chen [CC99] comparent les caractéristiques fondées sur des approches par transformée de Fourier, filtres spatiaux, filtres de Gabor et ondelettes. Les résultats obtenus montrent que les filtres de Gabor et les ondelettes sont meilleurs. Reste que ces études comparatives indépendantes ne dégagent pas un processus d'évaluation et de classification standard. Leurs modes d'analyse des résultats, souvent différents, ont un impact direct sur les résultats. Une étude intéressante réalisée par Smith et Burns [SB97] évalue et compare la précision des méthodes de classification de la texture. Les auteurs proposent un Benchmark libre d'utilisation comportant les techniques de texture les plus courantes avec différentes méthodes de classification. Dans le présent travail, les techniques que nous proposons pour la caractérisation de textures sont évaluées sur cette plate forme.

### 1.5.3 La forme

Si l'être humain est particulièrement sensible à l'attribut de couleur pour distinguer les objets, pour certain types d'ambiguïté cela n'est pas suffisant et l'on a besoin de l'attribut de forme. L'objectif de la description par la forme est d'extraire une caractéristique géométrique pour coder un profil, estimer la ressemblance et reconnaître un objet [GW02]. Pour la recherche d'image par le contenu la forme d'un objet devra être invariante aux transformations affines classiques translation, rotation et facteur d'échelle [MKL97]. Généralement les descriptions du profil d'un objet se divisent en deux catégories : les contours et les régions. Les méthodes basées sur les contours comprennent les descripteurs de Fourier [ZR72, PF77], et les chaînes de Freeman [FD77]. Les descripteurs de Fourier traditionnels sont obtenus par l'application de la transformée de Fourier (DFT) à un codage du contour résultant en une signature 1D de la forme. La DFT transforme les variations spatiales le long des contours d'un objet en représentation fréquentielle. Les caractéristiques sont donc obtenues à partir de la réponse fréquentielle, les détails correspondant aux hautes fréquences et les traits grossiers aux basses fréquences. Le codage de Freeman transforme également le contour en une chaîne 1D. Chaque point dans la chaîne est codé en fonction de son orientation par rapport au point précédent. Le code est donc basé sur quatre ou huit directions selon que l'angle de l'élément est codé modulo  $\pi/4$  ou  $\pi/8$ . Il existe d'autres expressions des contours telles que codes de Freeman dérivés,

---

<sup>1</sup>www.sipi.usc.edu

signature polaire [Ott91, Dav97], modèles autorégressifs [KC81, DG86]. Pour la caractérisation des régions, divers moments invariants [Hu62, TC88, JB91, Tau92] sont souvent utilisés comme descripteurs. Les moments invariants offrent une description robuste aux transformations affines, propriété appréciable pour les systèmes RIC.

### 1.5.4 Raisonnement spatial

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. Par exemple le système IRIS [HKKZ95] associe les relations spatiales aux caractéristiques de bas niveau en vue d'une interprétation plus évoluée de la scène. Le plus souvent on introduit l'arrangement spatial dans la mesure de similarité. La recherche d'image par similarité spatiale (RISS) est appliquée sur des images plus symboliques ou logiques constituant une version abstraite de la représentation physique.

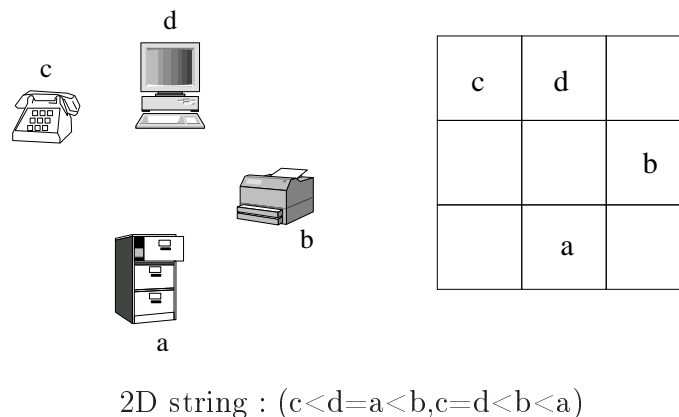


FIG. 1.4 – Exemple d'un arrangement spatial avec 2D string

#### 2D string et ses variantes

Chang et al. [CSY87] proposent les *2D string*, pour une description spatiale du contenu par des chaînes de symboles. Les objets dans l'image sont englobés par des rectangles dénommés *minimum bounding rectangle* (MBR). Les 2D string résultent de la projection des objets de l'image sur les axes  $x$  et  $y$  pour former deux chaînes relatant leurs position relatives. Les relations spatiales sont exprimées par des opérateurs logiques tels que ( $;$ ,  $<$ ,  $=$ ). La figure 1.4 illustre un exemple de ce type de relations. Et par conséquent la mesure de ressemblance entre deux relations est reportée sur la similarité de chaînes de caractère.

2D G-string [CJL89] est une extension de 2D string rajoutant plusieurs opérateurs spatiaux ainsi que de nouvelles relations acceptant mieux le chevauchement entre objets. 2D



G-string offre plus d'information spatiale ce qui facilite le filtrage des situations confuses.

Dans le même contexte Lee et Hsu [LH90] ont proposé 2D C-string basé sur un mécanisme de troncature des objets par MBR. Cette technique minimise le nombre des parties impliquées dans le calcul des relations spatiales. Ceci réduit significativement la taille des chaînes sans affecter les relations spatiales.

2D B-string [LYC92] n'utilise pas le mécanisme de troncature, mais représente l'objet par deux symboles. L'un indique le début du bord d'un objet et l'autre la fin. 2D B-string réduit également le nombre de relations.

Les méthodes de type 2D-string sont très sensibles aux rotations. Les chaînes générées pour une image qui a subi une rotation sont complètement différentes de celles calculées pour l'originale. Dans la section suivante nous présentons quelques méthodes reconnues invariantes aux transformations géométriques.

### $\theta R$ -string et ses variantes

Gudivada [Gud98] propose une description fondée sur les orientations relatives des objets dans l'image.  $\theta R$ -string ordonne les objets dans une chaîne selon l'angle formé par la ligne qui relie les deux centres de gravité et l'axe horizontal. La distance au centre de gravité de l'image est considérée en cas d'égalité (ordre spiral en polaire). La figure 1.5 représente deux images requête et cible contenant 4 objets. Dans la première l'ordre est :  $(d, c, a, b)$  en considérant l'ordinateur comme un élément d'ancrage. La deuxième image a subi une rotation en gardant le même point d'ancrage l'ordre des objet dans la chaîne reste le même  $(d, c, a, b)$ . La similarité est évaluée par mise en correspondance de graphes de type SOG (*spatial oriented graph*) dont les noeuds correspondent aux objets et les arcs retracent les relations spatiales.

Une extension de cette méthode est proposée par El-Kwae et Kabuka [EKK99] pour inclure les contraintes spatiales topologiques en sus des directionnelles. L'extension topologique offre des relations qui ne peuvent pas être traduites par les orientations. La technique proposée est invariante aux multiples rotations. La mesure de ressemblance est basée sur trois facteurs : nombres d'objets communs, distance entre relations directionnelles et distance entre relations topologiques.

### 1.5.5 Caractérisation de haut niveau

Les techniques d'apprentissage, supervisé ou non, peuvent servir à piloter un système RIC. 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 [MM96, MM97,

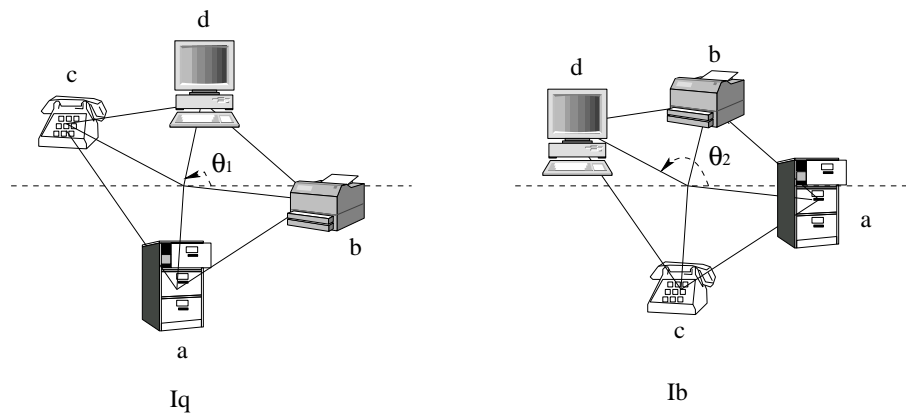


FIG. 1.5 – Orientation spatiale entres objets : a gauche l’image requête et à droite image de la base

[PM95, MP96] Ces algorithmes fournissent une approche plus évoluée sauf qu’il n’est pas toujours possible d’en interpréter les résultats, a fortiori de manière automatique, au delà du fait par exemple qu’ils sont conformes à ceux de tels ou tels humains sur un ensemble d’exemples donnés. La différence entre la recherche d’image par le contenu et la vision par ordinateur c’est que dans la première l’humain fait partie intégrante du système. Une tendance actuelle insiste donc sur l’interaction avec l’utilisateur pour l’élaboration d’une stratégie de recherche adaptée au comportement humain qui boucle sur les résultats affichés par le système. Jain et al. [Jai93, JPP95] indiquent qu’il y a un besoin d’explorer l’énergie humaine avec celle de l’ordinateur. La perception humaine est centrale dans les études de plusieurs équipes [MP96, CMOY96, PCC+98, RHOM98] avec l’idée d’intégrer à terme un modèle humain dans le RIC. Ces équipes recommandent donc les méthodes de retour de pertinence (*feedback*) pour compenser le fossé sémantique. Elles consistent à associer l’usager dans une boucle de retroaction où il privilégie les images qui correspondes à ses préférences. Ces quatre dernières années le concept du retour de pertinence s’est trouvé particulièrement renforcé [CS01, MBKB02, KJ03, KC03, GR04]. De plus en plus de systèmes intègrent des attributs de raisonnement haut niveau pour atteindre une sémantique satisfaisante.

## 1.6 Mesure de similarité entre attributs

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 RIC cette tâche s’exécute en temps réel. D’autres paramètres entrent en jeu tels la dimension de l’espace caractéristique, la taille de la base... La méthode naïve de recherche calcule la distance entre la requête et toutes les images de la base puis les ordonne selon leurs score. 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 comparé à la recherche séquentielle. Pour résumer, la mesure de similarité vérifie généralement les propriétés :

- *La perception* : Une faible distance dans l'espace de caractéristique indique deux images semblables.
- *Le calcul* : La mesure de distance se calcule rapidement pour une faible latence.
- *La scalabilité* : 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

Nous listons ci dessous les distances les plus utilisées pour comparer des images considérées comme vecteurs ou comme distributions statistiques.

## Distances 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 :

$$d^p(f_1, f_2) = \left( \sum_{i=1}^n |f_1(i) - f_2(i)|^p \right)^{1/p} \quad (1.1)$$

$p$  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 dite de Manhattan ( $p = 1$ ).

## 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 ressemblant, sa forme générale est donné par

$$D_Q = \sqrt{(f_1 - f_2)^T A (f_1 - f_2)} \quad (1.2)$$

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  $A$

$$a_{ij} = 1 - \frac{d_{ij}}{\max(d_{ij})} \quad (1.3)$$

Les propriétés de cette distance la rendraient proche de la perception humaine de la couleur, ce qui en fait une métrique attractive pour l'application RIC [FSN+95, CTB+99].

## Distance de Mahalanobis

Cette distance prend en compte la corrélation entre les distributions des classes. Elle est ainsi définie par :

$$D_M = \sqrt{(f_1 - f_2)^T C^{-1} (f_1 - f_2)} \quad (1.4)$$

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

$$D_M = \frac{\sum (f_1(i) - f_2(i))^2}{c_i} \quad (1.5)$$

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  $h_2$

$$D_{Intersec} = \frac{\sum_i \min(h_1(i), h_2(i))}{\sum_i h_2(i)} \quad (1.6)$$

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 d'attributs. Elle est exploitée notamment par Rubner et al. [RTC98]. EMD requiert une optimisation linéaire.

$$D_{EMD} = \frac{\sum_{ij} g_{ij} d_{ij}}{\sum_{ij} g_{ij}} \quad (1.7)$$

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

$$D_{EMD} = \sum_{ij} g_{ij} d_{ij} \quad (1.8)$$

Le coût est minimisé sous les contraintes suivantes :

$$\begin{aligned} g_{ij} &\geq 0, \forall i, j \\ \sum_i g_{ij} &\leq f_2(j), \forall j \\ \sum_j g_{ij} &\leq f_1(i), \forall i \\ \sum_i \sum_j g_{ij} &= \min(f_1(i), f_2(j)) \end{aligned}$$

EMD prétend également mimer la vision humaine.

### Divergence de Kullback-Leiber (KL)

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

$$D_{KL} = \sum_i f_1(i) \log \frac{f_1(i)}{f_2(i)} \quad (1.9)$$

### Divergence de Jeffrey (JD)

La divergence de Jeffrey est définie par :

$$D_{JD} = \sum_i (f_1(i) \log \frac{f_1(i)}{\hat{f}_i} + f_2(i) \log \frac{f_2(i)}{\hat{f}_i}) \quad (1.10)$$

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  $f^c(i)$  :

$$D_{KS} = \max_i |f_1^c(i) - f_2^c(i)| \quad (1.11)$$

### Distance de Cramer-Von Mises

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

$$D_{CVM} = \sum_i (f_1^c(i) - f_2^c(i))^2 \quad (1.12)$$

## Distance de Bhattacharyya

La distance de Bhattacharyya exploite la séparabilité entre deux distributions gaussiennes représentées par leur covariance  $\Sigma$  :

$$D_B = \frac{1}{8}(\mu_1 - \mu_2)^T \Sigma^{-1}(\mu_1 - \mu_2) + \frac{1}{2} \ln \frac{\det(\Sigma)}{\sqrt{\det(\Sigma_1)\det(\Sigma_2)}} \quad (1.13)$$

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.

## 1.7 Techniques de mise en correspondance

Il s'agit d'apparier deux ensembles de régions appartenant à deux images différentes afin d'élaborer une mesure de similarité globale entre images. Soit  $X$  et  $Y$  deux groupes de régions respectivement de l'image requête et cible figure 1.6. L'ensemble de régions des deux côtés peut être considéré comme un graphe bipartite dont les noeuds correspondent aux régions et les arcs attribués (pondérés par des distances) relient ces noeuds en couples. La solution optimale consiste en le graphe qui minimise le coût 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). (*Assignment Problem*)
2. Attribuer plusieurs éléments de la requête à plusieurs éléments de la cible ( $N : N$ )

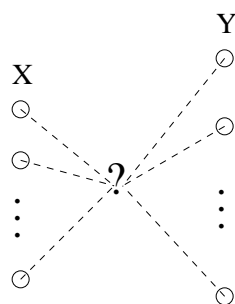


FIG. 1.6 – Problème de la mise en correspondance

Nous présentons maintenant quelques méthodes utilisées dans la recherche d'image par régions avec appariement 1 : 1 et  $N : N$ .

## L'algorithme Hongrois

Proposée par Kuhn [Kuh55] cette méthode est la plus communément utilisée pour résoudre l'AP *Assignment 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 noeud  $x_i \in X$  un noeud  $y_i \in Y$  qui contribue à minimiser la distance :

$$D_H = \sum_{i=1}^{|X|} w(x_i, y_j) \quad (1.14)$$

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 affecte le coût de la mise en correspondance.

## 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 cible (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 avec  $y_j$  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 génère des paires 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 indifférences ou encore des paires ex aequo. Ces types de MS peuvent être résolus de la même manière que MS standard.

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

TAB. 1.1 – Un exemple de liste de préférence

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

## Graphes attribués

Dans ce cas le modèle objet/scène est représenté par un graphe attribué (ARG) (*attributed relational graph*). Les noeuds codent les caractéristiques visuelles et les arcs spécifient les relations contextuelles entre noeuds. 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 ramène à une mise en correspondance d'ARG. L'appariement 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 de l'ARG suivie par l'algorithme hongrois puis 2D string.

## Appariement par région intégrée

L'appariement par région intégrée IRM (*Integrated region matching*) introduit par Li et al. [LWW00] est une méthode de type ( $N : N$ ), la figure 1.7 en représente un exemple. Le but ici est de compenser les imperfections de segmentation ce qui est souhaitable pour un 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.

$$D_W(x_i, y_j) = \sum_{ij} s(x_i, y_j) d(x_i, y_j) \quad (1.15)$$

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

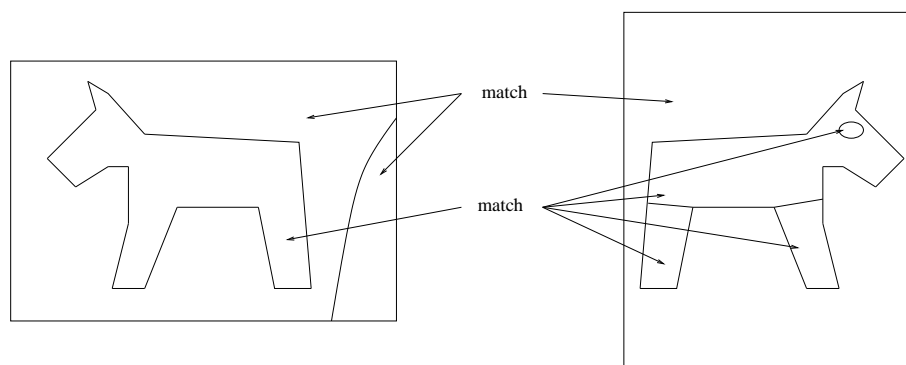


FIG. 1.7 – Exemple d'appariement IRM



## 1.8 Interaction homme machine

La recherche d'image implique une interaction avec l'utilisateur qui est difficile à formaliser. Elle dépend de plusieurs facteurs : type de recherche, fonctions disponibles, environnement mis à disposition, etc. Les premiers travaux sur la RIC étaient consacrés à l'aspect algorithmique plus qu'à la conception d'environnement facilitant l'interaction. Ces dernières années l'interaction homme machine est devenue partie intégrante des systèmes RIC. En effet la multiplication des choix d'actions que propose l'environnement à l'utilisateur aide à augmenter les performances d'un système.

### Espace requête

L'espace de la requête est la première étape que l'utilisateur saisit. Smeulders et al. [SWS<sup>+</sup>00] décompose l'espace requête  $Q$  en quatre entités :

$$\{I_Q, F_Q, S_Q, Z_Q\}$$

La première recouvre la sélection d'une image  $I_Q$  dans la base  $I$ .  $F_Q \subset F$  consiste en le choix des caractéristiques de l'image  $I_Q$ . La troisième entité est le choix de la fonction de ressemblance. La dernière  $Z_Q \subset Z$  est un ensemble d'étiquettes caractérisant un objet dans l'image avec un langage de haut niveau pour accéder à une sémantique. Ces trois premières composantes constituent la base de toute requête, la quatrième est optionnelle et dépend des procédures adoptées. En examinant cette instantiation  $Q$  au plan interactif on constate que  $F_Q$  est la composante la plus susceptible d'actions de l'utilisateur, et d'autant plus si le système propose de multiples choix d'attribut. La mesure de ressemblance étant estimée dans l'espace  $F_Q$ , l'exploitation de cette interaction pourrait s'en trouver limitée lorsqu'on veut améliorer le résultat parcequ'elle est figée indépendamment de l'utilisateur. En revanche, si le système intègre une boucle sur la pertinence l'interaction devient une variable prépondérante dans le calcul du prochain résultat.

## 1.9 Les Systèmes RIC

Dans cette section nous décrivons quelques systèmes de recherche d'image par le contenu qui sont opérationnels. Nous avons sélectionné les systèmes les plus réputés ou ceux qui sont proches de notre travail. Parmi les articles de synthèse répertoriant les systèmes existants on pourra se référer par exemple à Velkamp et al. [VT02] Ils exposent un état de l'art avec une taxonomie selon les méthodes de requête, les caractéristiques utilisées, la mesure de similarité, la présentation des résultats et le domaine d'application.

---

## QBIC

QBIC [FSN+95] est le premier système commercialisé dans le domaine du RIC. QBIC supporte différents types de requête : par exemple crayonnage (sketch), choix de couleur et texture. Il intègre trois types de caractéristiques couleur, texture et forme. La première est une moyenne de la couleur dans les espaces RGB, YIQ, Lab et Munsell. La deuxième est représentée par une version modifiée des caractéristiques de Tamura [TMY78]. La troisième comprend la surface de la forme, sa circularité, son excentricité, l'orientation des principaux axes et les moments invariants. Ce système utilise principalement la distance Euclidienne pour comparer les images et la distance quadratique pour comparer les histogrammes. QBIC est le premier système RIC à pratiquer l'indexation multidimensionnelle ( $R^* - tree$ ) pour accélérer la recherche.

## Virage

Virage est un moteur de recherche RIC qui emploie également la couleur, la texture et la forme comme attributs. Il offre à l'utilisateur une interface interactive souple pour formuler sa requête avec insertion d'image, fonction de dessin, réglage des poids,... La fonction de calcul de similarité est laissée au choix du développeur. L'ajustement des poids par l'utilisateur pour augmenter la pertinence des résultats est une sorte de retour de pertinence (feedback).

## Photobook

Photobook [PPS96] intègre des ensembles d'images comprenant forme ou texture et ultérieurement visages. Une signature de l'image est extraite différente selon l'étiquette et l'utilisateur peut interroger chacune des bases séparément avec des attributs correspondants. Pour la description de la texture, caractère prépondérant, Photobook considère l'image comme un champs aléatoire dont il identifie la périodicité, l'orientation et la distribution stochastique. Une modélisation par éléments finis dont on dérive des attributs pertinents est utilisée pour décrire la forme. La requête s'effectue classiquement par choix d'une image candidate. La ressemblance est estimée par une distance euclidienne.

## Netra

Netra [MM99] emploie une description par région segmentée dont chacune est caractérisée par couleur, texture, forme et localisation spatiale. La couleur est représentée par un dictionnaire avec quantification sur 256 couleurs. Les filtres de Gabor avec ondelettes est l'approche adoptée pour les textures. Le système utilise courbures et centroïde de la forme comme descripteurs. Cette représentation permet à l'utilisateur de formuler sa requête en

---

sélectionnant une région d'intérêt à laquelle il peut adjoindre l'attribut qui lui convient. Pour la mesure de similarité Netra utilise une distance Euclidienne.

## **Blobworld**

Au même titre que Netra, Blobworld [CTB<sup>+</sup>99] repose sur les régions caractérisées par la couleur, la texture, la forme et la localisation spatiale. La couleur est décrite par un histogramme de 218 cases dans l'espace Lab. Pour représenter la texture Blobworld utilise le contraste et l'anisotropie. La surface, l'excentricité, et l'orientation sont les caractéristiques de forme. La requête consiste à sélectionner une région et caractériser son importance selon une variable linguistique à deux valeurs ('somewhat', 'very'). Puis l'utilisateur indique de même l'importance de chaque attribut selon trois valeurs ('not', 'somewhat', 'very'). La distance employée combine la distance quadratique (pour couleur) avec la distance Euclidienne (pour texture et forme). L'indexation du contenu est similaire à celle du système QBIC. Blobworld est appliqué à 10000 images de la base Corel.

## **VisualSEEK**

VisualSEEK [SC96] supporte des requêtes basées sur des descripteurs visuels et spatiaux. Les caractéristiques sont un ensemble de couleurs définies dans l'espace HSV, les ondelettes pour la texture ainsi que les relations spatiales entre régions. Le système permet à l'utilisateur de crayonner les régions d'intérêt et sélectionner la couleur appropriée puis paramétrer les attributs de type spatial. Les distances quadratique et Euclidienne sont les deux mesures de similarité. VisualSEEK utilise la technique *R - tree* pour l'indexation.

## **PicHunter**

L'histogramme et la distribution spatiale de la couleur constituent la signature de l'image. La distance utilisée est de type Minkowski  $L_1$ . Le système PicHunter [CMOY96] incorpore une boucle de retour de pertinence probabiliste qui prédit les images cible d'après l'interaction système/utilisateur. Ce système a été testé sur des images de la base Corel.

## **SIMPLIcity**

Le système SIMPLIcity [WLW00] vise à augmenter la sémantique dans la RIC. Les images de la base sont classifiées automatiquement en catégories par une méthode sémantique. La recherche s'effectue dans l'une de ces catégories. Les images sont segmentées en

---

région, la caractérisation de chaque région étant basée sur des ondelettes. L'utilisateur formule sa requête en présentant un exemple au système. Pour la mesure de similarité SIMPLicity propose une méthode de mise en correspondance (IRM) entre régions des deux images permettant à une région requête d'avoir plusieurs correspondants cible. Le système est testé avec des images généralistes de type photothèque.

## MARS

Mars [ORC<sup>+</sup>97] est un système inter disciplinaire qui implique plusieurs domaines de recherche : traitement d'image, gestion de base de données et recherche d'information. Pour la caractérisation visuelle, l'image est découpée en blocs  $5 \times 5$ . Des indices de texture et couleur sont calculés pour chaque imagerie. La couleur est représentée par un histogramme 2D (coordonnées HS), les coefficients d'ondelettes décrivent la texture. La segmentation des images se déroule en deux procédures. La première est l'algorithme des k-moyennes sur l'espace couleur/texture, la deuxième une détection de régions par regroupement selon un modèle d'attraction. Le système est paramétrable : par exemple on peut choisir la palette de couleurs. De plus MARS offre un nombre d'opérateurs logiques pour formuler la requête. La ressemblance entre couleurs est mesurée par l'intersection d'histogrammes. Pour les textures le système applique une distance euclidienne.

## 1.10 Techniques de segmentation d'image

Les méthodes de segmentation peuvent être classées en trois catégories [PP93, GW02]

- Approche région (domaine spatial).
- Approche contour (domaine spatial).
- Approche classification (espace de caractéristique).

Le seuillage d'après histogramme est un élément fondamental de la segmentation [Ots79, SSWC88]. La technique de base pour un choix automatique de seuil consiste à séparer un objet contrasté sur un fond, chacun obéissant à une distribution Gaussienne. Le choix du seuil est critique car le résultat de segmentation en dépend. Pour des images contenant plusieurs objets ou régions la méthode s'étend en un multi seuillage. [PG87, WS89, Abu89].

La croissance de région peut être considérée déjà comme une sorte de classification [RK82]. Elle consiste à faire s'élargir une cellule à partir d'un pixel initial selon un critère d'homogénéité (couleur, texture,...). Les problèmes inhérents à cette technique sont la détermination des pixels de départ et le choix du critère d'arrêt. Pour pallier la complexité des premières étapes où l'on traite un grand nombre de petites régions cette technique peut

---

être associée avec la précédente (multiseuil sur histogramme). De nombreuses stratégies de croissance de région ont été publiées [FM81, HS85, GW02].

La technique de division/fusion (*split-and-merge*) procède d'abord au partitionnement récursif de l'image. La décision de diviser une région est déterminée par un critère du type moyenne, variance, étendue etc . Le processus de fusion permet dans un second temps de regrouper les régions voisines qui satisfont un même critère d'homogénéité. Cette technique est volontiers supportée par une structure de données hiérarchique (arbre quaternaire)[JKS95].

La détection des bords [FM81, PP93, GW02] permet de segmenter l'image en générant des frontières entre les régions. Un point de contour est défini comme le siège d'une discontinuité dans l'homogénéité de région. Le modèle de contour est une transition rapide entre niveaux de gris dans une direction associée à une faible variation dans la direction orthogonale. La forte variation est donc communément détectée par un opérateur type gradient tel que le filtre de Sobel ou celui de Canny-Deriche par exemple. On trouve dans littérature des méthodes hybrides qui combinent les approches région et contour [TA97, ZL98] pour allier la robustesse des premières avec la précision des secondes.

### 1.10.1 Les méthodes de regroupement

La segmentation sera donc vue comme un problème de regroupement où l'on cherche à mettre les points de la même région dans le même ensemble. Le clustering, en tant que méthode d'apprentissage non supervisé, consiste à organiser des objets dans des groupes dont les membres partagent des propriétés communes et vérifient donc un critère de similarité souvent explicité au sein du même groupe. De nombreux articles témoignent de l'intérêt essentiel du regroupement dans de nombreux domaines [Fuk90, KR90, JMF99, DK97, BB99, JDM00]. Les méthodes proposées dans la littérature peuvent être divisées en deux groupes : méthodes hiérarchiques et méthodes de partitionnement [JMF99]. Le principal avantage des méthodes hiérarchiques par rapport aux méthodes de partitionnement (fonction objectif), c'est qu'elles n'ont pas besoin d'une connaissance *a priori* du nombre de groupes. En revanche elles présentent des anomalies croissant avec la taille des données. Bien que les techniques de partitionnement nécessitent un certain paramétrage elles sont largement utilisées car elles délivrent des résultats cohérents avec la perception humaine.

Les algorithmes de partitionnement les plus populaires sont fondés sur les centres, ou prototypes, des classes. Ils consistent généralement en la minimisation d'une fonction objectif qui implique la distance entre le point de données et le prototype du cluster. On peut distinguer deux types de tels algorithmes :

- Le regroupement exact (*Hard Clustering*) dont dérive la famille des K-Moyennes et où chaque point doit être assigné à une seule classe à chaque itération.
  - Le regroupement flou (*Fuzzy Clustering*) qui exploite le concept d'appartenance floue
-

[Zad65] et tolère pour un point d'appartenir à plusieurs classes.

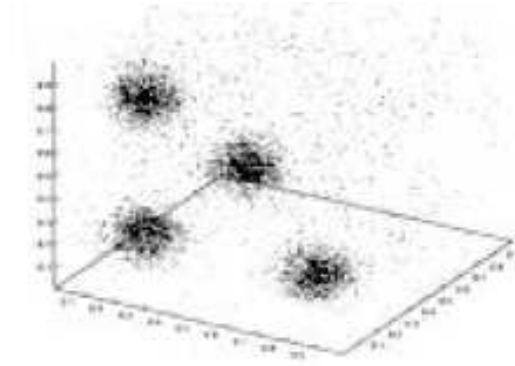


FIG. 1.8 – Exemple d'un nuage de point dans un espace 3D

### 1.10.2 Techniques hiérarchiques

Les techniques hiérarchiques consistent à produire un graphe (dendrogramme figure 1.9) dans lequel les régions sont imbriquées à plusieurs niveaux d'abstraction. Ces techniques permettent d'examiner un objet à plusieurs échelles en évaluant sa similarité aux différents niveaux du dendrogramme [JMF99, dS01]. Les méthodes proposées dans la littérature [JD88, Eve93, DHS00] sont généralement divisées en deux catégories qui rappellent les tendances en détection de région : aglomérative vs. fractionnelle. La première procède par approche ascendante, en commençant d'assigner chaque entité à un cluster pour fusionner itérativement les couples de clusters similaires. La deuxième procède d'une manière descendante. Au départ l'ensemble des points est considéré comme un seul groupe puis d'une manière hiérarchique il est divisé jusqu' au dernier cluster. On obtient de même des classes dans chaque niveau d'abstraction. La première méthode est la plus répandue, elle se résume en les étapes suivantes :

1. Etant donné  $n$  entités, initialise  $p$  à  $n$
2. Détermine les deux plus proches clusters  $c_i$  et  $c_j$  selon une mesure de similarité appropriée
3. Fusionne  $c_i$  et  $c_j$  et décrémente  $p$  de 1
4. Répète 2 et 3 tant que  $p > 1$ .

Comme un cluster contient plusieurs entités la mesure de similarité entre deux groupes ne dépend pas seulement de la métrique mais également de la technique utilisée. C'est pour cela que la principale différence entre les méthodes hiérarchiques réside dans la technique de mesure. Soit  $A$  et  $B$  deux clusters contenant respectivement  $n_1$  et  $n_2$  éléments. Notons par  $d(A, B)$  la distance entre deux groupes. Nous distinguerons typiquement quatre concepts :

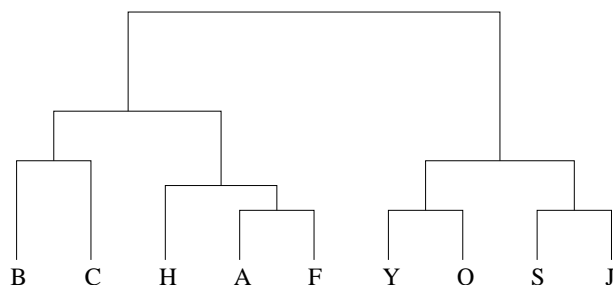


FIG. 1.9 – Exemple d'un clustering hiérarchique

### Liaison simple

La méthode dite *single link* [Joh67, JD88, Eve93, DHS00, dS01] fusionne deux clusters si la distance entre leurs plus proches membres est faible. Elle se calcule par la formule :

$$d(A, B) = \min_{x \in A, y \in B} d(x, y) \quad (1.16)$$

Cette méthode semble bien adaptée aux images car les régions peuvent présenter une variance importante, dans un tel cas la présente fusion n'exclut pas les points qui dans les deux clusters sont loin de la moyenne.

### Liaison complète

La méthode dite *complete link* [JMF99, dS01] définit la similarité entre deux groupes comme la distance entre leurs membres les plus distants. Ils sont fusionnés si cette distance est faible.

$$d(A, B) = \max_{x \in A, y \in B} d(x, y) \quad (1.17)$$

Cette méthode est efficace sur des données formant des groupes compacts.

Les deux précédentes techniques représentent deux extrêmes dans l'acceptation de la similarité et tendent à être sensibles aux intrus, puisque dépendant seulement de plus proches ou plus lointains voisins. La technique suivante mélange les deux informations dans une sorte de moyenne qui la rend moins sensible aux éléments atypiques.

### Liaison moyenne

Le *group average link* est connu également comme UWGMA (*un-weighted within-group method using arithmetic averages*) [LW66, Eve93]. Elle consiste à calculer la distance

moyenne entre toutes les paires du produit des deux groupes.

$$d(A, B) = \frac{1}{\text{card}(A)\text{card}(B)} \sum_{x \in A, y \in B} d(x, y) \quad (1.18)$$

Il est bien évident que cette méthode est plus coûteuse en temps de calcul que les deux précédentes.

### Méthode de Ward

La méthode de Ward [War63] mesure la distance en calculant la somme des écarts types de chaque point par rapport à la moyenne du futur groupe.

$$d(A, B) = \frac{1}{\text{card}(A) + \text{card}(B)} \sum_{x \in A, B} \|x - m\|^2 \quad (1.19)$$

où  $m$  est le centroïde du cluster fusionné. A chaque étape deux clusters sont réunis si leur erreur quadratique est la plus petite.

Le coût de calcul est un inconvénient majeur des méthodes hiérarchiques. Kurita [Kur91] estime ce temps pour les méthode agglomératives en  $O(n^2 \log n)$ . Cela ne convient pas pour les données volumineuses. Dans le but de réduire la complexité du calcul Amadasun et King [AK88] combinent l'approche région et la technique agglomérative. La technique consiste à extraire des blocs uniformes, puis obtenir un vecteur moyen pour chaque bloc. Ces vecteurs seront regroupés selon la méthode agglomérative, le processus s'arrêtant à l'obtention d'un nombre de clusters prédéterminé.

### 1.10.3 Techniques de partitionnement

Les techniques de partitionnement semblent mieux adaptées à l'approche hiérarchique pour les applications impliquant un nombre important de points. Dans les méthodes hiérarchiques la classification des point n'est pas flexible, si à une étape donnée un élément est classé il ne peut être reclassé dans les prochaines étapes. Le partitionnement permet alors à un point de se déplacer d'une classe à une autre pendant le processus de clustering. La difficulté pour les techniques de partitionnement réside dans la détermination du nombre de classe que le nuage de données comporte [Dub87]. Idéalement les méthodes de partitionnement optimisent une fonction objectif qui favorise la similarité à l'intérieur d'un groupe et la dissimilarité inter groupes. Si  $X = x_1, x_2, \dots, x_N$  est l'ensemble des points et  $v_i$  est un centre de cluster, les techniques de partitionnement tentent de former des amas de points  $x_j$  autour de chaque centre  $v_i$ . Le résultat consiste en  $C$  groupes séparés.

---



## K-moyennes

Les K-moyennes [Mac67, TG74] est l'un des algorithmes de partitionnement fréquemment utilisé. Il est basé sur la minimisation d'une erreur quadratique entre tous les éléments et les centres des clusters.

$$J_m = \sum_{i=1}^C \sum_{j=1}^N \|x_j - v_i\|^2 \quad (1.20)$$

Nous détaillons cet algorithme dans le chapitre 3. Il existe d'autres versions des K-moyennes visant à améliorer le résultat de la segmentation en introduisant des critères supplémentaires [JMF99].

## K-moyennes adaptatif

Pappas [Pap92] propose une version généralisée de K-moyennes appliquée sur des images en niveaux de gris. Deux caractéristiques interviennent : l'intensité et la position spatiale. Le K-means généralisé est donc un algorithme adaptatif permettant l'incorporation d'une contrainte spatiale. Il inclut une fonction de densité de la probabilité *a posteriori* qui modélise la contrainte spatiale et celle de l'intensité d'une région.

## ISODATA

Proposé par Ball et Hall [BH67] ISODATA (*Iterative Self-Organizing Data Analysis*) est une variante de K-means qui vise à contruire une partition optimale à partir d'une partition initiale. Les clusters sont fractionnés ou fusionnés durant le processus de classification selon un critère bien déterminé par exemple la variance. L'ISODATA requiert un paramétrage traduit par différents seuils imposés par l'utilisateur, rendant le résultat sensible aux choix des paramètres.

## DYNOC

Une autre variante de K-moyennes et voisin de l'ISODATA a été introduit par Diday [Did73], DYNOC (*DYNAmique Optimal Cluster-seeking*) ou Nuées Dynamiques utilise le principe division/fusion pour atteindre à un nombre optimal de classes. On trouve encore d'autres extensions ou variantes de cet algorithme par exemple exploitant simplement des critères différents tel que maximum-vraisemblance [Sym81].

### 1.10.4 Techniques de mélange de distributions

Ces techniques supposent que l'objet à classifier correspond à une distribution identifiable, le but est alors d'extraire les paramètres de chaque distribution constituant la distribution globale. En version paramétrique cela conduit au mélange de gaussiennes [JMF99] et les paramètres à estimer sont donc la collection des couples "moyenne  $\mu_i$  - matrice de covariance  $\Sigma_i$ ". L'algorithme dit Expectation Maximisation (EM) [DLR76, MK96] a été employé dans cette perspective pour l'estimation des paramètres, où les densités des composantes et les paramètres dérivent d'un modèle d'objet par identification itérative. Le mélange des distributions est contrôlé par les proportions  $\gamma_i$  (avec la contrainte  $\sum_{i=1}^C \gamma_i = 1$ ). La densité de probabilité de la résultante s'exprime par :

$$\begin{aligned} p(x, \theta) &= \sum_{i=1}^C \gamma_i p_i(x|i, \theta_i) \\ &= \sum_{i=1}^C \gamma_i \frac{1}{2\pi^{p/2} |\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)\right) \end{aligned} \quad (1.21)$$

Où  $\theta_i = \{\mu_i, \Sigma_i, \gamma_i\}$  est le paramètre global de mélange.  $p$  est la distribution des données  $x$ . Le critère d'optimalité est par exemple le maximum de vraisemblance.

$$L(\theta) = \sum_{j=1}^N \ln \sum_{i=1}^C \gamma_i p(x_j|i) \quad (1.22)$$

La procédure considère

*Expectation* : estimation de la probabilité d'appartenance.

$$p^k(i|x_j) = \frac{\gamma_i^k p^k(x_j|i)}{p^k(x_j)} \quad (1.23)$$

*Maximisation* : maximisation du log de vraisemblance.

$$\mu_i^{k+1} = \frac{\sum_{j=1}^N p^k(i|x_j) x_j}{\sum_{j=1}^N p^k(i|x_j)} \quad (1.24)$$

$$\Sigma_i^{k+1} = \frac{\sum_{j=1}^N p^k(i|x_j) (x_j - \mu_i^k)(x_j - \mu_i^k)^T}{\sum_{j=1}^N p^k(i|x_j)} \quad (1.25)$$

$$\gamma_i^{k+1} = \frac{1}{N} \sum_{j=1}^N p^k(i|x_j) \quad (1.26)$$

où  $k$  est le nombre d'itération

La première étape de la procédure EM consiste à estimer un vecteur initial de paramètres, puis on applique un processus itératif en deux étapes (expectation et maximisation). Pendant ces deux étapes les entités (ou régions de l'image) sont évaluées par leur score en regard du mélange de densités généré à partir de ces paramètres. Les nouvelles entités sont attribuées à des clusters selon leurs scores. Les paramètres sont alors recalculés pour chaque nouvelle composante impliquant ces entités. Le choix initial des paramètres affecte le déroulement de la classification jusqu'à engendrer des clusters incohérents dans le résultat final.

Certaines méthodes sont des versions non paramétriques [JD88] de la précédente. D'autres combinent EM avec des techniques de décision différentes exploitant par exemple une structure d'arbre [FD97]. Caillol et al. [CPH97] segmentent l'image avec une estimation floue des distributions gaussiennes. Delingnon et al. [DMP97] ont introduit une estimation généralisée appliquée à des images SAR.

### 1.10.5 Techniques floues

Les données organisées en groupes compacts et bien séparées sont statistiquement rares. Dans les zones où les groupes se chevauchent les points appartiennent à la fois aux différents groupes. Cette situation d'appartenance partielle est fréquente dans les images. Les techniques du clustering flou ont justement étaient pensées en vue de cette situation. Elles déterminent pour chaque point un degré d'appartenance à une classe. La matrice de partition floue  $u_{ij}$  code la probabilité qu'un point  $x_j$  soit dans la classe  $v_i$ . Cette matrice doit satisfaire certaines conditions, précisées plus loin dans le chapitre 3.

#### C-moyennes floues

Parmi les algorithmes adaptés avec succès du déterministe au flou figure celui des C-moyennes floues FCM (*fuzzy C-means*). Le procédé FCM [Bez81] est donc une version floue des k-moyennes où l'on introduit le degré d'appartenance d'un point à une classe. La fonction objectif de FCM est :

$$J_{FCM} = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m \|x_j - v_i\|^2 \quad (1.27)$$

Où  $u_{ij}$  est la matrice de partition, Nous détaillons cette algorithme dans le chapitre 3.

## K-moyennes possibilistes

Krishnapuram et Meller [KK93, KK96] introduit dans le clustering la notion de possibilité [DP88] très voisine du flou ou d'autres extensions des probabilités comme l'évidence [Sha76]. Les auteurs considèrent le degré d'appartenance d'un point à une classe comme une possibilité d'appartenir à une classe. Contrairement à FCM la somme des degrés d'appartenance de toute les classes n'est pas 1. La fonction objectif modifiée est :

$$J_P = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m d_{ij}^2 + \sum_{i=1}^C \eta_i \sum_{j=1}^N (1 - u_{ij})^m \quad (1.28)$$

Où  $\eta_i$  est un nombre positif qui détermine la distance à laquelle le degré d'appartenance égale 0,5.  $d_{ij}$  est la distance d'un échantillon  $j$  au centre de cluster  $i$ . Le premier terme est identique à celui de FCM, le second force  $u_{ij}$  à la plus grande valeur possible. Cet algorithme est robuste au bruit mais il souffre du même inconvénient que FCM : le nombre de classe doit être spécifié.

## Algorithme d'agrégation compétitive

Frigui et Krishnapuram [FK97] proposent un algorithme CA (*Competitive Agglomeration Algorithm*) qui combine les approches hiérarchique et de partitionnement. CA est un algorithme itératif. Il partitionne les données dans un premier temps en un grand nombre de cluster qui entrent en compétition ; progressivement les clusters qui perdent sont fusionnés. CA minimise la fonction objectif suivante :

$$J_{CA} = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m d_{ij}^2 - \alpha \sum_{i=1}^C \left[ \sum_{j=1}^N u_{ij}^m \right]^2 \quad (1.29)$$

Initialement le nombre de groupes est surestimé puis à chaque itération ce nombre décroît jusqu'à atteindre le nombre optimal. L'un des attraits de cette méthode est que le nombre exact de cluster n'est pas requis a priori.

## FCM sous contrainte spatiale

Les techniques standard de type FCM ou K-moyennes sont sensibles aux bruit, aux intrus, etc. Les considérations spatiales visent à rendre la segmentation plus consistante. Nous décrivons quelque méthodes clefs qui imposent la contrainte spatiale sur la fonction objectif.

Tolias et Panas [TP98b] proposent une méthode basée sur un vote des voisins du pixel donné. La contrainte spatiale est appliquée à la fonction de partition par le bais

d'un automate appelé (*Sugeno-type fuzzy system*) [TS85] à trois entrées, une sortie et 11 règles. Les entrées sont des variables linguistiques, HIN et LON, traduisant le nombre des pixels d'une classe présentant respectivement un fort et faible degré d'appartenance. La variable DUMAX définit les connexions relatives entre un pixel et ses voisins. Ce système génère une matrice de correction  $DU$  et la nouvelle fonction de partition est obtenu par :

$$U^{new} = U + DU \quad (1.30)$$

Le comportement des pixels voisins influence la nouvelle fonction  $U$ . Cette méthode est générique car elle s'applique comme post traitement sur les résultats des algorithmes de type FCM.

Ahmed et al. [AYMF99] modifient la fonction objectif de FCM en rajoutant un deuxième terme Eq. 1.31. Ce terme représente la contrainte spatiale exercée par les voisins en chaque pixel d'après leurs distances au même cluster. Si un pixel est proche d'un centre de groupe et ses voisins affichent également une proximité pour ce même centre, alors le degré d'appartenance à ce cluster augmente. Donc un point est contraint de suivre son voisinage ce qui produit un effet de régularisation

$$J_m = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m \|x_j - v_i\|^2 + \frac{\alpha}{card(N_j)} \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m \left( \sum_{x_k \in N_j} \|x_k - v_i\|^2 \right) \quad (1.31)$$

où  $N_j$  est l'ensemble des voisins "spatiaux" et  $\alpha$  contrôle l'influence du deuxième terme. La fonction objectif est optimisée avec la méthode de Lagrange sur les centres et la fonction de partition. L'algorithme est similaire à celui de FCM. Cette méthode a été appliquée à des image de résonance magnétique et s'est montrée plus d'efficace en présence de bruit que EM et FCM. Chen et Zhang [CZ04] proposent une extension de la technique précédente en calculant la distance par la méthode du Noyau. Il en résulte une certaine amélioration.

Pham [Pha01] opte pour un critère différent qui consiste à contraindre la classification par les appartenances des voisins. Il propose la fonction objectif suivante :

$$J_{RFCM} = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m \|x_j - v_i\|^2 + \frac{\beta}{2} \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m \sum_{l \in N_j} \sum_{k \in M_k} u_{lk}^m \quad (1.32)$$

Où  $N_j$  représente aussi un ensemble de voisins et  $M_k = \{1, \dots, C\} \setminus \{k\}$  représente l'ensemble des clusters à l'exception du cluster  $k$ . Ici l'appartenance d'un pixel à une classe déterminée est conditionnée par l'appartenance de ses voisins aux autres classes, ce qui revient à une corrélation inverse. En d'autres termes si l'appartenance des voisins aux autres classes est faible alors cela renforce la confiance dans la classe en question. Cette méthode dépend donc des valeurs des autres classes et non pas uniquement de la classe envisagée. Comme il n'y a pas de contrainte sur les centres des clusters l'équation reste identique à celle de FCM. En revanche la fonction de partition est différente mais obtenue également par optimisation lagrangienne. L'algorithme procède selon les mêmes étapes que FCM.

### 1.10.6 D'autres méthodes, d'autres modèles

Selon Bensaid et al. [BHC96] les algorithmes de clustering sont sensibles au nombre de clusters de même que l'attribution aux classes après terminaison du processus (*defuzzification*). Ils proposent un FCM semi supervisé dont chaque classe comporte des prototypes de données étiquetés.

Liew et al. [LL00] proposent un FCM avec contrainte spatiale adaptative selon l'information locale et l'espace de caractéristique. L'adaptation est fonction de l'homogénéité région. Cette fonction contrôle la contrainte du voisinage permettant l'annulation de l'influence des pixels voisins dans des régions non homogènes. Un processus de fusion est adopté permettant à deux clusters proches dans l'espace de caractéristique et qui partagent le même contexte spatial d'être fusionnés.

Une autre technique originale proposée par Yung et Lai [YL98] adopte un principe de contrainte gravitationnelle. Deux points  $x_i$  et  $x_j$  dans un espace de couleur RGB sont considérés comme deux particules  $p_i$  et  $p_j$  avec les masses respectives  $m_i$  et  $m_j$ . Ces particules sont soumises à la loi d'interaction gravitationnelle.  $F = -Gm_i m_j (x_i - x_j) / |x_i - x_j|^3$ .  $G$  est la constante de gravitation. Chaque particule détermine la recombinaison d'un point dans un cluster dont le cardinal est régi par une force.

Shi et Malik [SM97] et Shi et al. [SBLM98] traitent la segmentation via le clustering comme un problème de partitionnement de graphe. Ils considèrent les points d'un espace de caractéristiques arbitraire comme un graphe pondéré  $G = (V, E)$ , où les points sont les noeuds et les arcs représentent les liens entre ces points. Le poids  $w(i, j)$  de chaque arc indique la similarité entre les noeuds  $i$  et  $j$ . Le but est de partager l'ensemble des vertex  $V$  en groupes disjoints  $V_1, \dots, V_C$  respectant la contrainte d'une forte similarité à l'intérieur du groupe et faible similarité inter groupes. La séparation entre deux groupes se fait par la suppression de tout les arcs reliant leurs vertex. Le coût total des suppressions (*cut*) détermine la ressemblance :  $cut(V_1, V_2) = \sum_{v_1 \in V_1, v_2 \in V_2} w(v_1, v_2)$ . La valeur optimale du partitionnement est celle qui minimise le *cut*.

### 1.10.7 Mesure de validité

L'objectif des méthodes de clustering est de déterminer les groupes présents dans les données, au sens où ces groupes ont des membres qui se ressemblent à l'intérieur et sont aussi différents que possible des membres d'un autre groupe. Cette définition n'impose pas un nombre de groupes dans le cas général. La difficulté qui se pose alors est la détermination du nombre optimal de clusters. La recherche de la solution optimale est appelée mesure de validité CVI (*cluster validity indices*) [Bez74]. Les méthodes CVI exploitent les critères suivant :

1. *Compacité* : Il s'agit de la mesure de dispersion des éléments d'un groupe. La variance peut être un indice de cette mesure, plus elle est petite plus grande est la densité .
2. *Séparation* : Indique le degré de séparation entre deux groupes en calculant une distance généralement entre les prototypes.

D'après ces deux critères la mesure de validité réalise un compromis entre les distances intra-cluster et inter-cluster. Une grande variété de méthodes a été proposée dans ce contexte [JMF99]. Nous focalisons ici sur les principales techniques floues.

## Indices de Bezdek

Le premier indice associé avec FCM est le coefficient de partition [Bez81] défini par :

$$PC(c) = \frac{1}{N} \sum_{i=1}^c \sum_{j=1}^N u_{ij}^2 \quad (1.33)$$

L'indice  $PC$  prend ses valeurs dans l'intervalle  $[\frac{1}{c}, 1]$ . Dans le cas particulier où toutes les fonctions de partition sont égales,  $u_{ij} = \frac{1}{c}$ ,  $PC$  atteint donc le minimum ce qui signifie qu'il n'y a pas de groupes potentiels. Le nombre optimal de cluster est obtenu par la recherche du maximum ( $\max_{2 \leq c \leq N-1} PC(c)$ ).

Le deuxième indice consiste à mesurer une entropie des fonctions de partition [Bez74].

$$PE(c) = -\frac{1}{N} \sum_{i=1}^c \sum_{j=1}^N u_{ij} \log_2 u_{ij} \quad (1.34)$$

L'indice est calculé pour  $c > 1$  et  $PE(c) \in [0, \log_2 c]$ . Les valeurs de  $c$  proches des bornes indiquent l'absence d'une structure en groupes. La solution consiste à minimiser  $PE$ , ( $\min_{2 \leq c \leq N-1} PE(c)$ ).

L'inconvénient des deux indices c'est leur tendance monotone en fonction de  $c$ . Dave [Dav96] propose une mesure pour réduire cette tendance.

$$MPC(c) = 1 - \frac{c}{c-1}(1 - PC(c)) \quad (1.35)$$

Où  $MPC(c) \in [0, 1]$ . L'optimum est obtenu par une maximisation comme pour  $PC$ .

---

### L'indices de Fukuyama et Sugeno

La mesure de validité de Fukuyama et Sugeno [FS89b] est définie par :

$$FS(c) = \sum_{i=1}^c \sum_{j=1}^N u_{ij}^m \|x_j - v_i\|^2 - \sum_{i=1}^c \sum_{j=1}^N u_{ij}^m \|v_i - \bar{v}\|^2 \quad (1.36)$$

Où  $\bar{v}$  est la moyenne des centres des groupes ou la moyenne des données. Le premier terme mesure la compacité des clusters et le deuxième mesure la séparation. Le minimum exprime une bonne densité et séparation ( $\min_{2 \leq c \leq N-1} FS(c)$ ).

### L'indice de Gath et Geva

Cette indice est basé sur le concept de l'hypermétrie et sur la densité. Gath et Geva [GG89] proposent une fonction de validité dénommée *fuzzy hypervolume validity function* définie par :

$$FHV(c) = \sum_{i=1}^c [\det(F_i)]^{1/2} \quad (1.37)$$

Où  $F_i$  est la matrice floue de covariance du cluster  $i$

$$F_i = \frac{\sum_{j=1}^N (u_{ij})^m (x_j - v_i)(x_j - v_i)^T}{\sum_{j=1}^N (u_{ij})^m} \quad (1.38)$$

Une petite valeur de  $FHV(c)$  indique l'existence d'un cluster compact.

### L'indice de Xie et Beni

Xie et Beni [XB91] définissent la mesure de validité comme le rapport de la compacité à la séparation.

$$XB(c) = \frac{\sum_{i=1}^c \sum_{j=1}^N u_{ij}^2 \|x_j - v_i\|^2}{N * \min_{i,j} \|v_i - v_j\|^2} \quad (1.39)$$

La minimisation de cette fonction produit le nombre optimal de cluster que peuvent comporter les données. La fonction  $XB(c)$  a tendance à décroître avec le nombre de cluster pour réduire cette tendance une borne maximale  $c_{max}$  est introduite. La recherche de la valeur minimale se limite dans un intervalle inférieur au nombre de points ( $\min_{2 \leq c \leq c_{max}} XB(c)$ ).

### L'indice de Zahid et al.

La mesure de validité proposée par Zahide et al. [ZLE99] est définie par :

$$SC(c) = SC_1(c) - SC_2(c) \quad (1.40)$$



Où

$$SC_1(c) = \frac{\sum_{i=1}^c \|v_i - \bar{v}\|^2 / c}{\sum_{i=1}^c (\sum_{j=1}^N u_{ij}^m \|x_j - v_i\|^2 / \sum_{j=1}^N u_{ij})} \quad (1.41)$$

et

$$SC_2(c) = \frac{\sum_{i=1}^{c-1} \sum_{l=i+1}^c (\sum_{j=1}^N (\min(u_{ij}, u_{lj}))^2) / \sum_{j=1}^N \min(u_{ij}, u_{lj})}{\sum_{j=1}^N (\max_{1 \leq i \leq c} u_{ij})^2 / \sum_{j=1}^N \max_{1 \leq i \leq c} u_{ij}} \quad (1.42)$$

les deux composantes mesurant la séparation et la compacité des groupes.  $SC_1$  prend en compte les propriétés géométriques de la structure de données et les fonctions de partitions.  $SC_2$  ne considère que le degré d'appartenance. Le nombre optimal est obtenu par maximisation du résultat ( $\max_{2 \leq c \leq N-1} SC(c)$ ).

## 1.11 Bilan

Dans ce chapitre nous avons cerné l'état de l'art des techniques de traitement d'image ou de vision utiles dans le cadre de la recherche d'image par le contenu. Dans la première partie nous avons parlé de la description de l'image par la couleur, la texture et la forme. Ces attributs constituent les descripteurs fondamentaux des systèmes RIC. Le raisonnement spatial est appliqué au contenu pour obtenir une description d'un niveau d'abstraction plus élevé. Le résumé de l'image contient donc toute information nécessaire au calcul de la similarité. La mesure de cette dernière repose souvent sur un calcul de distance dans un espace de caractéristique. Le choix de la distance est lié également à l'attribut par exemple pour l'histogramme on utilise généralement l'intersection d'histogrammes ou la distance quadratique. Cependant la mesure de ressemblance devient plus complexe pour l'approche par région : pour faire correspondre deux images elle nécessite un traitement particulier qui se ramène à l'appariement sur graphe bipartite. La recherche d'image basée sur les régions requiert donc une détection préalable des objets ou régions significatives, c'est le but de la segmentation. La deuxième partie aborde les principales approches de segmentation par regroupement. Les techniques de partitionnement sont les mieux adaptées aux systèmes RIC. Leur inconvénient majeur est la spécification du nombre de groupes. Il existe des techniques qui permettent de déterminer ce nombre, techniques dites de mesure de validité. De l'exposé des techniques précédentes nous déduisons nos propres choix successivement détaillés dans les trois chapitres principaux de ce mémoire.

- Les objets supportant la recherche seront chez nous des régions caractérisées par leur texture. Cette texture sera décrite par des indices de type statistique mais qui captent des propriétés fréquentielles grâce au parcours d'image optimal. Ce sont les co-occurrences de motifs que nous étudions au chapitre 2 et dont nous montrons chapitre 4 qu'elles supportent une recherche comparable à celle d'un humain au moins par le taux de réussite

- 
- Pour extraire les régions automatiquement nous tablons sur des techniques FCM qui sont apparues parmi les plus efficaces et s'accrochent particulièrement des parcours de Peano, récursifs et locaux. En effet nous les exploitons pour proposer deux extensions nouvelles de FCM, par contraintes de voisinage et en multi-résolution. L'indice de validité est de même adapté en multi-résolution et nous avons réalisé une étude de la qualité de segmentation, comparative pour plusieurs techniques classiques et celle que nous proposons.
  - Enfin la recherche proprement dite exploite une distance  $L_2$  entre caractéristiques de régions associée à des relations spatiales floues et un appariement bidirectionnel de type XOR flou sur matrice de coût que nous avons développé. L'ensemble est mis à portée de l'utilisateur via une interface que nous avons spécialement conçue et qui a permis notamment une évaluation circonstanciée de performances allant de la robustesse aux transformations affines jusqu'à la compétition avec l'opérateur humain.

**Il est important de souligner pour conclure cette étude que la justification ultime de tous nos choix ne se conçoit que dans la performance mesurée sur une collection de requêtes estimées suffisamment représentatives.**

---



## Chapitre 2

# Caractérisation locale de la texture par des parcours récursifs

### 2.1 Introduction

La texture est un attribut visuel majeur quasi omniprésent dans les images naturelles. Pourtant la reconnaissance et la classification automatique de formes par leur texture apparaît difficile, notamment en vue d'une recherche d'image par le contenu. Bien qu'on recense de nombreuses techniques pour caractériser la texture aucune ne ressort comme un détecteur universel : chacune présente des avantages et désavantages qui dépendent notamment du domaine d'application. Dans ce chapitre nous décrivons une méthode utilisant les courbes de Peano pour extraire localement (voisinage du pixel) puis à un niveau plus global (régions) des attributs caractérisant la texture.

### 2.2 Analyse de la texture

L'analyse de la texture a été abordée d'une manière intensive ces dernières années en vue d'un système intelligent comparable à celui de l'être humain. Les capacités de ce dernier à discerner les textures restent très supérieures à celles de la machine. Cependant plusieurs méthodes permettent de caractériser automatiquement des textures et résoudre ainsi certains problèmes liés à ce phénomène tels que extraction ou reconnaissance d'objet, reconstruction 3-D ou perception de la déformation. Les performances de la caractéristique codant la texture sont alors souvent jugées via celles de la perception elle-même fondée sur la classification de ces textures et la segmentation. Une caractéristique efficace présente des propriétés discriminantes élevées. Il n'existe pas une définition précise de la texture au-delà du fait qu'elle restitue l'état de surface, mais certaines propriétés en ont été à

---

l'usage vérifiées pertinentes. On retiendra deux d'entre elles :

- répétition régulière ou aléatoire d'une (famille de) primitive(s) de base.
- distribution spatiale des niveaux de gris, incluant donc des notions fréquentielles.

Elles conduisent à des définitions trop générales dans le cas pratique et doivent encore être déclinées en des paramètres qui prennent des formes variées et des techniques d'analyse qui obéissent à des modèles et des interprétations diverses. L'une des difficultés réside ensuite dans l'adaptation d'une méthode non universelle à la variété des problèmes visés pour l'application. Par exemple les données peuvent être multispectrales, la résolution peut être variée, la texture peut participer à une vision plus ou moins attentive etc. mais aussi la puissance de calcul disponible, le temps de réaction, la précision spatiale sont des facteurs d'importance variable.

## 2.3 Schéma général

Un système de classification se décompose généralement en trois tâches principales :

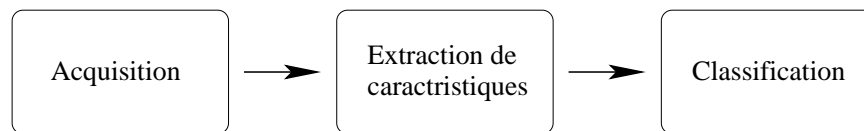


FIG. 2.1 – Système de classification

A l'acquisition d'image, plusieurs paramètres physiques entrent en jeu tels que l'éclairage, la position de la caméra, la résolution. Dans la plupart des applications réelles il est difficile de contrôler ces paramètres, les méthodes de caractérisation sont donc supposées robustes à leurs effets et variations.

L'extraction des caractéristiques est la phase du traitement qui quantifie les descripteurs. Le choix de ces variables et de la méthode de mesure a un impact direct sur le résultat et la performance de classification

La classification est l'étape finale dont on attend des résultats cohérents et répondant aux critères prédéterminés par l'application. La mesure de similarité et le type du classifieur déterminent la construction des classes.

---

## 2.4 Description de la texture

Qu'il s'agisse de modèle (description a priori) ou de codage du résultat (description a posteriori) la segmentation des images par la texture implique la description des entités de l'image texturées (régions, facettes, objets etc. selon le degré d'interprétation immédiate) avant classification. Une description efficace améliorera les performances d'analyse d'image. Le rôle du descripteur est d'exhiber les indices pertinents communs au rendu d'un même phénomène. Par exemple, si nous considérons la texture comme une distribution de primitives de base les descripteurs seront plutôt de nature statistique. En revanche si on essaye de mimer le comportement visuel humain plutôt sensible aux fréquences spatiales, des paramètres (amplitude ou énergie) liés à des transformations orthogonales de l'image seront favorisés. Nous décrivons ci-dessous des méthodes illustrant les deux approches.

### 2.4.1 L'approche par mesures statistiques

Ces approches consistent à évaluer des propriétés statistiques d'une région ou d'un certain voisinage autour d'un pixel comme information texturale. Il est évident que ces mesures dépendent de la taille de la région ou fenêtre d'analyse. Déjà une statistique valide sinon robuste requiert une taille qui comprend au moins un "motif" de base. La taille dans ce cas dépend de facteurs interdépendants tels que la résolution, la dynamique des niveaux de gris, la granularité de la texture, le niveau de détail souhaité... dont certains sont inconnus sauf à supposer la texture connue ! Par exemple plus la fenêtre est réduite plus la dynamique doit l'être pour garder significative l'occurrence d'un niveau de gris donné.

#### Statistiques du premier ordre

Elles prennent en compte un pixel à la fois pour construire une distribution, par exemple l'histogramme qui est celle des niveaux de gris dans l'image. Certaines applications utilisent ce genre de distribution (vecteur) directement comme attribut. Dans d'autres on en extrait des propriétés statistiques comme moyenne, variance ou kurtosis. Bien que les statistiques de premier ordre prouvent des propriétés discriminantes dans des cas simples, elles demeurent une représentation limitée des textures comparé aux statistiques d'ordre supérieur.

#### Statistiques du deuxième ordre

Ce sont celles qui exploitent les paires de pixels en incluant donc une certaine dépendance spatiale. Parmi ces méthodes on trouve la fonction d'autocorrection et la matrice de

---

co-occurrence des niveaux de gris (GLCM). Cette dernière a fait l'objet de plusieurs études et s'avère souvent efficace. Mais comme elle conduit en réalité à un vecteur de matrices, la complexité de calcul et de stockage d'un tel descripteur augmente considérablement si bien qu'on est conduit souvent à en dériver des mesures (à nouveau d'inspiration statistique ou informationnelle) qui résument le phénomène en un attribut plus dense. Un tel attribut capte des facettes de la texture qui peuvent s'avérer plus difficiles à interpréter.

Notons que deux textures perçues comme différentes peuvent avoir des attributs statistiques du premier ordre identiques et des attributs du deuxième ordre différents, notamment par le caractère directionnel introduit : ceci permet de relativiser les rôles respectifs dans la discrimination entre classes de textures.

### Matrice de co-occurrence des niveaux de gris

La construction de GLCM consiste à mesurer une probabilité conjointe dans une paire de pixels en calculant la fréquence d'apparition simultanée de deux niveaux de gris aux extrémités d'un vecteur donné  $U = (d, \theta)$ . Supposant que  $g(x)$  est la valeur du pixel  $x$  appartenant à un voisinage déterminé du pixel  $(i, j)$ , GLCM correspond à l'histogramme 2D des paires  $[g(x), g(x + U)]$  sur ce voisinage. La normalisation de cette matrice produit une distribution de probabilité jointe  $pU(i, j)$ . Dans la pratique la GLCM peut être moyennée sur toutes les orientations possibles pour  $d$  fixé. La taille de la matrice dépend de la dynamique des niveaux de gris : pour une image en 256 niveaux les dimensions de GLCM seraient  $256 \times 256$  ; la figure 2.2 illustre les valeurs GLCM calculées pour deux textures. Les voisinages considérés étant de taille bornée, le simple bon sens conduit à limiter le nombre d'éléments a priori nuls de GLCM en limitant la dynamique. De plus on réduit encore les données de description en associant des mesures aux matrices. 14 mesures statistiques ont été initialement proposées dans [HSD73], seulement une partie de ces mesures est utilisée [WDR76, CH80, CTH84]. Parmi ces caractéristiques :

Contrast	$\sum_{i,j} (i - j)^2 p(i, j)$
Energy	$\sum_{i,j} p^2(i, j)$
Entropie	$-\sum_{i,j} p(i, j) \log(p(i, j))$
Homogénéité	$\sum_{i,j} \frac{1}{1+(i-j)^2} p(i, j)$
Corrélation	$\frac{\sum_{i,j} (i-\mu_i)(j-\mu_j)p(i,j)}{\sigma_i \sigma_j}$
Moment inverse	$\sum_{i,j} \frac{1}{1+(i-j)^2} p(i, j)$
Shade	$\sum_{i,j} ((i - \mu_i) + (j - \mu_j))^3 p(i, j)$
Prominence	$\sum_{i,j} ((i - \mu_i) + (j - \mu_j))^4 p(i, j)$

Où  $\mu_i = \sum_{i,j} i p(i, j)$ ,  $\mu_j = \sum_{i,j} j p(i, j)$   
 $\sigma_i = \sum_{i,j} (i - \mu_i)^2 p(i, j)$  et  $\sigma_j = \sum_{i,j} (j - \mu_j)^2 p(i, j)$

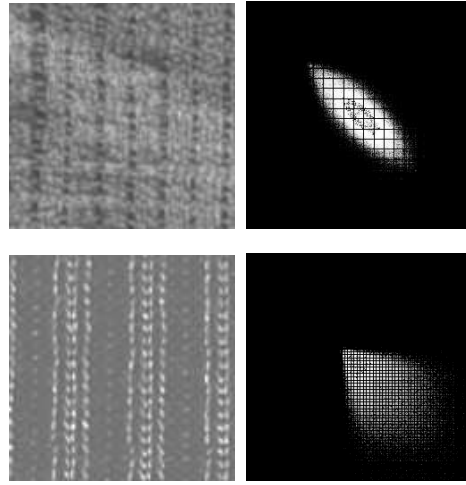


FIG. 2.2 – Matrice de co-occurrence des niveaux de gris pour deux textures différentes

### 2.4.2 L'approche fréquentielle

Ces approches consistent à extraire l'énergie portée par le signal dans diverses bandes de fréquence. Parmi les méthodes largement utilisées en description de texture les filtres de Gabor ont démontré de bonnes performances. Ce banc de filtres réalise la décomposition spectrale du signal en fréquence et en orientation. Des études ont montré que le processus de ce filtrage est similaire aux champs récepteurs des cellules du cortex visuel. Nous nous contenterons donc de décrire cette technique pour résumer les principes fréquentiels de la texture.

#### Filtres de Gabor

Les filtres de Gabor sont des filtres passe bande. Leur forme générale résulte de la multiplication d'une fonction de forme d'enveloppe Gaussienne avec une fonction sinusoïdale complexe. Nous utilisons la famille de filtres suivante :

$$G_{\lambda,\theta,\varphi}(x, y) = e^{-\frac{1}{2}\left(\frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2}\right)} \cos\left(2\pi \frac{x'}{\lambda} + \varphi\right) \quad (2.1)$$

$$\begin{aligned} x' &= x \cos(\theta) + y \sin(\theta) \\ y' &= -x \sin(\theta) + y \cos(\theta) \end{aligned}$$

L'écart type  $\sigma$  de la gaussienne détermine la largeur de la fenêtre autour du pixel centre de la convolution.  $\theta$  est l'orientation du lobe gaussien. Le paramètre  $\lambda$  est la longueur d'onde où  $\frac{1}{\lambda}$  est la fréquence spatiale dans l'image.  $\varphi$  représente le décalage en phase. La détection effective de texture consiste à convoluer l'image avec un banc de filtres de Gabor



paramétrés par  $\lambda, \theta$  et  $\phi$ . Bien que le résultat de la convolution puisse être directement utilisé comme un attribut de la texture, on préfère souvent la fonction d'énergie comme caractéristique [FS89a]. Cette fonction peut être définie en chaque pixel comme suit :

$$E_{\lambda, \theta}^2(x, y) = \left( \sum_{l, k \in \Omega} G_{\lambda, \theta, 0}(x - l, y - k) I(l, k) \right)^2 \quad (2.2)$$

Le vecteur de caractéristique sera constitué des énergies calculées pour un nombre déterminé de paramètre. La figure 2.3 montre un échantillon de filtres de Gabor.

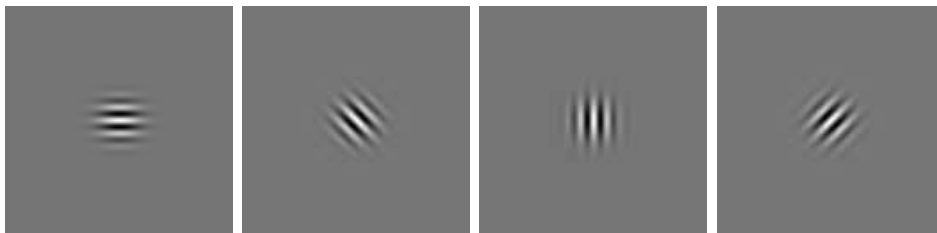


FIG. 2.3 – De gauche à droite un banc des filtres de Gabor orientés respectivement en  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ .

## 2.5 Les courbes de Peano

Les courbes de remplissage de l'espace "Space Filling Curves" (SFCs) appliquent d'une manière récursive un espace à plusieurs dimensions sur un espace monodimensionnel. Les points sont ainsi ordonnés en une séquence où l'ordre dépend d'une permutation locale (initiale) et de la récursion. Les parcours ainsi définis trouvent application dans l'indexation, le stockage et la manipulation de données en plusieurs dimensions [FR89, ARR<sup>+</sup>97]. Souvent en effet, la nature physique des supports de stockage rend nécessaire une vectorisation des données. Par exemple le transfert et l'accès aux données dans un disque dur est une opération séquentielle que le parcours peut accélérer [SZ97, PW00]. Ces courbes présentent également des propriétés intéressantes dans le traitement d'image. Elles matérialisent autant de balayages et les différentes relations d'ordre entre pixels engendrées par les différentes séquences modulent les fréquences présentes dans l'image. On sait par exemple qu'un balayage type Peano de base (parcours dit en  $Z$ ) exhibe plus de régularité en moyenne que le classique balayage ligne à ligne. On peut ainsi défendre l'intuition que certains parcours résonnent plus ou moins avec certaines textures jusqu'à permettre leur caractérisation. Un exemple de tel parcours d'image est illustré figure 2.4.

**Définition 2.5.1** *Etant donné un entier  $x \in \mathbf{Z}$ , son code de Peano pour un facteur d'étalement de 2 (i.e. dans le plan) "en Z" est :*

$$p_2(x) = 2^{2^{\lfloor \log_2 x \rfloor}} + p_2(x - 2^{\lfloor \log_2 x \rfloor}) \quad (2.3)$$

$$p_2(0) := 0 \quad (2.4)$$

**Définition 2.5.2** *pour un point de l'image  $(x, y) \in \mathbf{Z}^2$ , son indice dans le parcours de Peano en Z est :*

$$p(x, y) = p_2(x) + 2p_2(y) \quad (2.5)$$

Une opération très simple permet d'obtenir l'indice de Peano d'une position donnée  $(x, y)$ . La fonction  $p_2(x)$  s'obtient par insertion d'un 0 entre chaque bit du nombre. Exemple :  $y = 1 : 1 \rightarrow \underline{0}1$   $x = 5 : 101 \rightarrow \underline{0}1\underline{0}0\underline{0}1$ . Le pixel  $(5,1)$  sera donc le 19<sup>eme</sup> élément visité (voir figure 2.4 (a)). Cette opération fournit une implantation matérielle efficace.

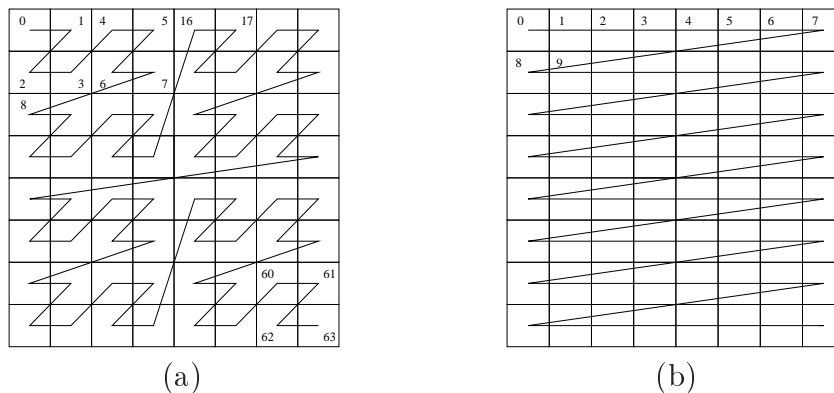


FIG. 2.4 – (a) Balayage de Peano en motif "Z" (b) Balayage horizontal classique

Avantage supplémentaire la séquence de pixels obtenue par balayage de Peano sur une image, supporte une représentation naturelle en arbre binaire. Seetharaman et Zavidovique [SZ98, ZS01] en ont dérivé des propriétés algébriques intéressantes pour manipuler les données image. Les travaux plus récents de [JCSZ02] portent sur la manipulation du parcours de Peano et de l'arbre associé. Ils ont démontré que la permutation des noeuds de l'arbre (pixels ou un groupes de pixels) selon un critère d'optimalité permet de réduire les transitions rapides entre pixels ou bloc de pixel. Cela se traduit par l'élimination des hautes fréquences spatiales dans l'image. Ce résultat a été appliqué pour la recherche d'image par le contenu avec le spectre de puissance comme signature des images.

## 2.6 Les parcours locaux

Les relations entre pixels spatialement proches révèlent des caractéristiques intéressantes telle que la dépendance entre points voisins e.g. en terme de variation de l'intensité

lumineuse. Les parcours locaux résultent de l'ordonnancement des pixels selon des courbes de Peano définies sur un voisinage limité. L'opération associée est donc locale. Notre but est d'optimiser des propriétés choisies de ces courbes pour extraire des caractéristiques proches des attributs visuels. En effet les parcours locaux supportent un certain codage par un attribut local nommé "motif" [JCSZ04]. Pour des raisons de complexité on se limite à un voisinage de 4 points. L'image est donc divisée en blocs de 4 pixels, soit 24 parcours (permutations) possibles et un codage exhaustif par 24 motifs comme illustré figure 2.5. Un critère d'optimalité étant choisi, chaque bloc élémentaire engendre le motif particulier approprié qui va le représenter.

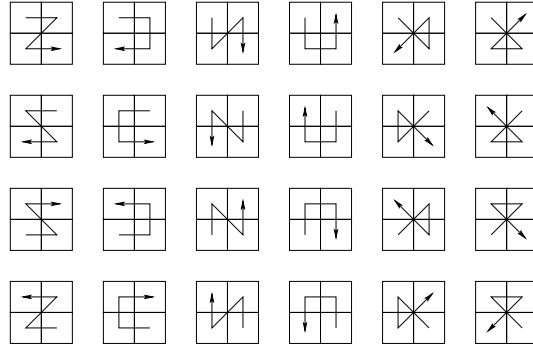


FIG. 2.5 – Tous les motifs possibles pour 4 points

**Définition 2.6.1** *Etant donné un ensemble de 4 points voisins  $g = \{a, b, c, d\} \in$  au plan  $\pi$ , un motif  $M$  est défini comme le chemin parcouru par une courbe  $\zeta$  passant par chaque point une seul fois. En fixant l'origine du parcours on obtient 6 motifs possibles voir figure 2.5.*

Le point de départ détermine un sous ensemble de 6 motifs. La symétrie (abcd/dcba) fait que les motifs s'apparient par sens de parcours. Etant donné que le sens n'a pas d'incidence physique sur les propriétés concernées dans notre cas, le nombre de motifs peut être réduit de moitié. On remarquera la redondance sur la figure 2.5.

**Définition 2.6.2** *Pour tous les points  $\in g$  les 24 différents motifs sont réduits à 12.*

$$M_t = \{\forall x \in g \exists \zeta / \zeta \in \{Z, C, N, U, \alpha, \gamma, Z^{-1}, C^{-1}, N^{-1}, U^{-1}, \alpha^{-1}, \gamma^{-1}\}\} \quad (2.6)$$

$M_t$  est le nouvel ensemble de parcours locaux qui engendre tous les sous ensembles de motifs figure 2.5.

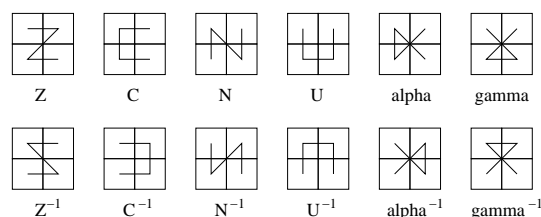


FIG. 2.6 – Motifs représentant tous les cas possibles

## 2.7 Le parcours optimal

Dans cette section nous décrivons le critère d'optimalité utilisé pour obtenir les motifs appropriés. Nous exploitons les variations de l'intensité dans le voisinage pour choisir entre les différentes courbes possible. On considère que l'intensité lumineuse d'un pixel représente un niveau d'énergie. Supposons qu'une particule veut visiter 4 niveaux une seule fois pendant le parcours en fournissant le moindre effort. Le changement de niveau provoque une perte d'énergie pour la particule, cette perte est liée aux différences entre niveaux. Une grande différence se traduit par une grande perte d'énergie. Le comportement de la particule consiste à chercher le chemin optimal de manière à réduire les transitions rapide des niveaux énergétiques. Le parcours doit engendrer la courbe la plus régulière parmi celles de l'ensemble et ainsi on obtient le motifs correspondant.

**Définition 2.7.1** *Un parcours optimal est défini comme le chemin qui minimise la variation d'intensité lumineuse le long de la courbe. Soit quatre pixels adjacents  $\begin{matrix} p_1 & p_2 \\ p_3 & p_4 \end{matrix}$  le scan optimal suit la permutation  $r^*$  qui correspond à :*

$$\delta = \min_r \{|p_{r1} - p_{r2}| + |p_{r2} - p_{r3}| + |p_{r3} - p_{r4}|\} \quad (2.7)$$

La valeur minimale de  $\delta$  détermine un parcours optimal. Un des 12 motifs représentera donc le bloc de 2x2 pixels. L'application de ces parcours sur toute l'image produit l'image de motifs associée. Ce codage produit toujours la meilleure disposition entre les pixels voisins en défavorisant les transitions rapides. Il constitue localement un attribut de texture. En effet les motifs sélectionnés sont déterminés par les relations entre des pixels voisins.

**Remarque 1** *Plusieurs motifs peuvent parcourir la grille d'une manière optimale. Par exemple dans le cas particulier où les pixels du bloc ont la même intensité lumineuse tous les parcours sont optimaux.*

## Image de motifs

Il existe a priori deux variantes pour obtenir une image de motifs. Dans la première méthode l'image est divisée en grille  $2 \times 2$  pixels où chaque élément de la grille (bloc) est traversé par un parcours optimal et remplacé ainsi par un motif sans recouvrement. La taille de l'image résultat est réduite d'un facteur 2 en largeur et en hauteur par rapport à l'image originale. La deuxième méthode consiste à extraire pour chaque pixel et ses 3 voisins immédiats le motif correspondant. Ceci résulte en un opérateur local dont le résultat peut être par exemple un code du motif retenu. Par convention on commence par le point le plus haut à gauche.

## 2.8 Description par les motifs

L'approche que nous allons décrire par la suite peut être considérée comme à la fois structurelle et statistique. Après avoir obtenu un élément de description locale de la texture par le biais des parcours locaux, nous cherchons maintenant à extraire une information au niveau global pour caractériser la texture.

### 2.8.1 Matrice de co-occurrence des motifs (MCM)

Les parcours locaux nous offrent une primitive locale de texture en déterminant des relations entre les pixels parcourus. Cependant pour caractériser la texture de manière plus robuste on procédera d'une manière globale en exploitant une distribution spatiale des motifs. Pour cela nous utilisons les matrices de co-occurrence. Nous déterminons les relations spatiales entre paires de motifs en mesurant la probabilité qu'un motif soit associé à un autre. Un terme de la matrice de co-occurrence des motifs MCM est défini comme la probabilité de trouver un "*motif i*" à une distance donnée  $\bar{d}$  du "*motif j*". En cela la MCM généralise la matrice de co-occurrence de niveaux de gris. La distance  $\bar{d}$  est définie selon les coordonnées polaires  $(d, \alpha)$  discrètes. En pratique  $\alpha$  prend les valeurs de  $0^\circ$  à  $315^\circ$  par pas de  $45^\circ$ . A cause de la symétrie on peut n'utiliser que les 4 premiers codes de Freeman en 8-connexité. Comme le nombre de motifs est de 12, le vecteur de caractéristiques est représenté dans un espace de dimension  $12 \times 12$  ajoutant un avantage de compression par rapport à GLCM dont la taille peut aller jusqu'à de  $256 \times 256$ . Le fait de transformer d'abord les intensités lumineuses en une description plus symbolique engendre un ordre statistique supérieur au niveau des données et par conséquent MCM code le troisième ordre statistique de l'intensité lumineuse alors que GLCM ne code que le deuxième ordre.

---

## 2.8.2 Problème de translation

La MCM calculée sans recouvrement est très sensible aux translations. Une image décalée d'un nombre impair de pixels peut engendrer deux MCM différentes. En effet le décalage change les positions relatives des pixels voisins considérés sur la grille. Les parcours adressent alors les points dans un contexte et donc un ordre différents de ceux de l'image initiale. Cela produit des motifs différents voir figure 2.7 et par conséquent leur distribution statistique conjointe en sera affectée. Afin d'éviter ce problème deux solutions sont testées :

1) L'image originale est décalée d'un pixel horizontalement, verticalement et diagonalement afin de cerner tout les cas possibles. Pour chaque translation on calcule une MCM. On obtient 4 MCM dont, en cas de décalage, l'une correspond à l'originale.

2) Le parcours avec recouvrement est invariant aux translations car tous les pixels sont visités et la relation entre voisins n'est pas affectée.

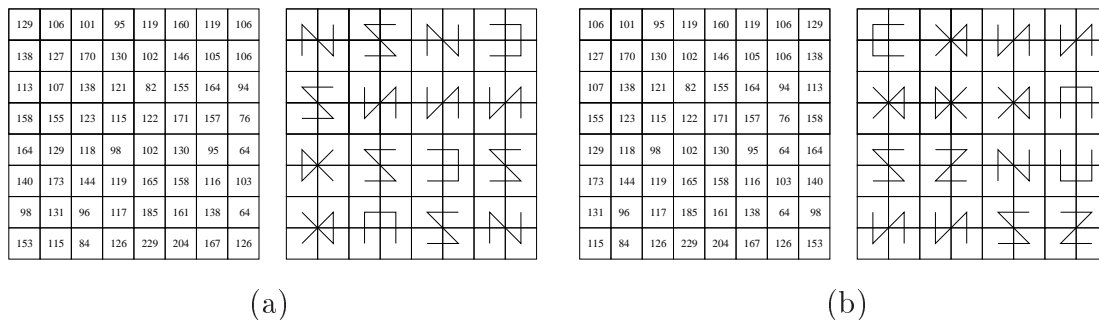


FIG. 2.7 – a) l'image motif originale b) l'image motif après un décalage horizontal à gauche de l'image originale

## 2.8.3 Propriétés de MCM

### Invariance au changement de luminance

Le changement des conditions d'illumination influence l'extraction de caractéristiques image, y compris des mesures de textures comme GLCM. Cela peut engendrer certaines incohérences dans l'information extraite (contours artificiels, mirages etc.). Il est intéressant de garantir une certaine consistance du phénomène. Supposons que la lumière incidente est uniforme et que le changement dans les conditions d'éclairage est monotone. On peut constater que dans un tel cas l'histogramme de l'image ou GLCM seront affectés par ce changement. L'avantage de MCM est d'être invariante par rapport à ces conditions. En effet supposant que  $c$  est une constante additive de l'intensité lumineuse, pour calculer le parcours optimal on se réfère à l'équation 2.7; on a donc :

$$\delta = \min_r \{ |(p_{r1} + c) - (p_{r2} + c)| + |(p_{r2} + c) - (p_{r3} + c)| + |(p_{r3} + c) - (p_{r4} + c)| \} \quad (2.8)$$

Il est évident que l'équation 2.8 est équivalente à l'équation 2.7. Dans les deux cas ils en résulte le même motif.

### Séparabilité entre classes

Dans cette section nous allons étudier les propriétés de la MCM quant à la séparation de différentes classes de texture. La dimension importante du vecteur MCM (144) suggère d'utiliser une technique de regroupement optimal comme l'analyse en composantes principales (ACP) pour réduire la dimension de l'espace caractéristique en le projetant sur les (deux) premiers axes principaux.

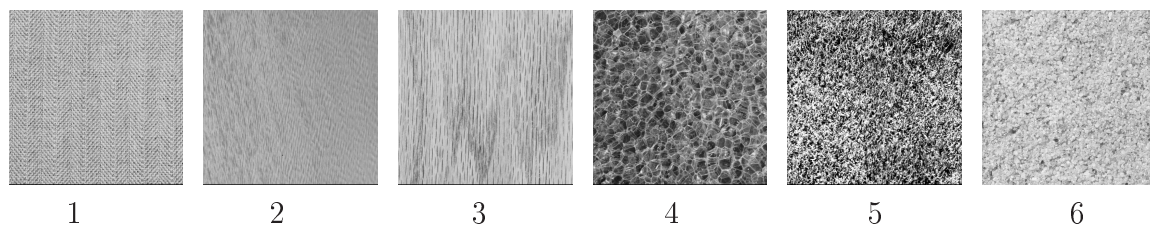
### Analyse en composantes principales

La technique de l'ACP a pour but de condenser les données originelles en une nouvelle représentation qui conserve au mieux leur information intrinsèque. La réduction de dimensionnalité est obtenue dans l'espace caractéristique par projection sur la base des vecteurs propres de la matrice d'autocorrélation et troncature. En effet si l'information (énergie) est une fonction de la trace de cette matrice, donc invariante par rapport à la base, on minimise l'erreur de projection en ne retenant que les vecteurs propres (directions principales) associés aux valeurs propres (variance dans la direction considérée) les plus grandes.

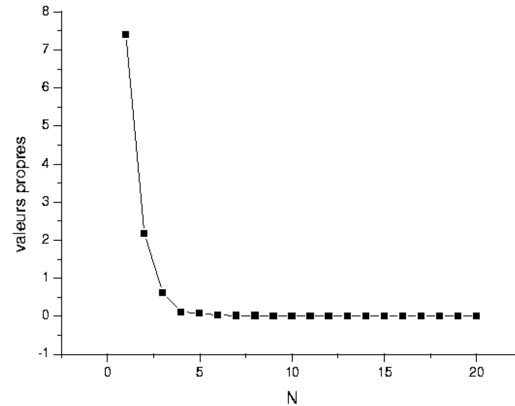
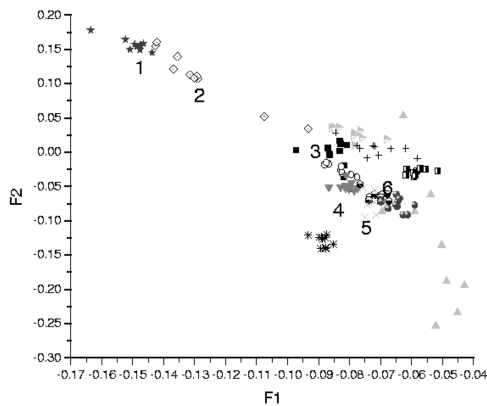
### Expérimentation sur des classes de textures

Pour étudier les propriétés discriminantes de MCM on traite 13 classes de textures issues de la base Brodatz (un exemple figure 2.8(a)) chaque classe est représentée par deux images en niveaux de gris ; la deuxième image est le résultat d'une égalisation d'histogramme sur la première. On extrait aléatoirement de chacune des deux images 5 blocs de taille  $w$  soit 10 blocs par classe de texture. Chaque bloc est caractérisé par sa MCM et on applique l'ACP sur l'espace caractéristique des MCM pour ne garder que les deux axes principaux. Ce choix de 2 permet notamment de visualiser la distribution des points de chaque classe dans un plan. La figure 2.9(a) représente des échantillons d'images aériennes prises avec une résolution de 16 mètres au sol. Ces images échantillons contiennent différents types ou classes de reliefs (champs cultivés, chaîne montagneuse, zones urbaines,...). La figure 2.9(b) montre la dispersion des points de chaque classe après projection. On remarquera dans cette figure que les points de la même classe ont une tendance à former des groupes séparés dans le plan.

---



(a)



(b)

FIG. 2.8 – a) Exemple de texture "Brodatz" b) A gauche l'analyse en composantes principales ; les numéros sur la figure correspondent aux classes ci-dessus. A droite les valeurs propres de la matrice de covariance

### Effet de la multirésolution

La résolution constitue intuitivement un problème majeur pour la caractérisation et l'analyse de texture. Le problème d'échelle est confirmé expérimentalement et souvent cité dans les image réelles. Dans cette section nous examinons l'effet de la résolution sur la MCM. Nous testons encore une série d'images satellite parcequ'elles constituent dans la suite notre application privilégiée. Dans la figure 2.10 les images au dessus des courbes sont obtenues du site *teraserver*. Ce serveur nous permet de naviguer sur des images satellites en plusieurs résolutions, de gauche à droite 4m, 2m, 1m, 0,5m et 0,25m au sol. La zone d'intérêt est un terrain de type agricole. Les MCM correspondantes sont normalisées pour compenser le facteur d'échelle. Pour les deux premières images, la MCM est calculée à l'intérieur du cadre blanc. On remarquera sur les courbes la stabilité en valeur des MCM malgré la variation de résolution. Les paramètres de la matrice de co-occurrence  $(d, \alpha)$  sont maintenus constants. L'écart entre MCM est fonction croissante de l'écart en résolution. Dans la figure 2.10 la première courbe à droite qui correspond à une résolution de 4m a des valeurs plus élevées que celle de 0,25m (extrême gauche). On peut l'expliquer par la distribution spatiale des niveaux de gris qui induit une distorsion de la texture entre images. La déminution de la résolution peut avoir un effet de lissage sur les images ainsi qu'une influence sur la finesse ou le caractère grossier de la texture. A cela



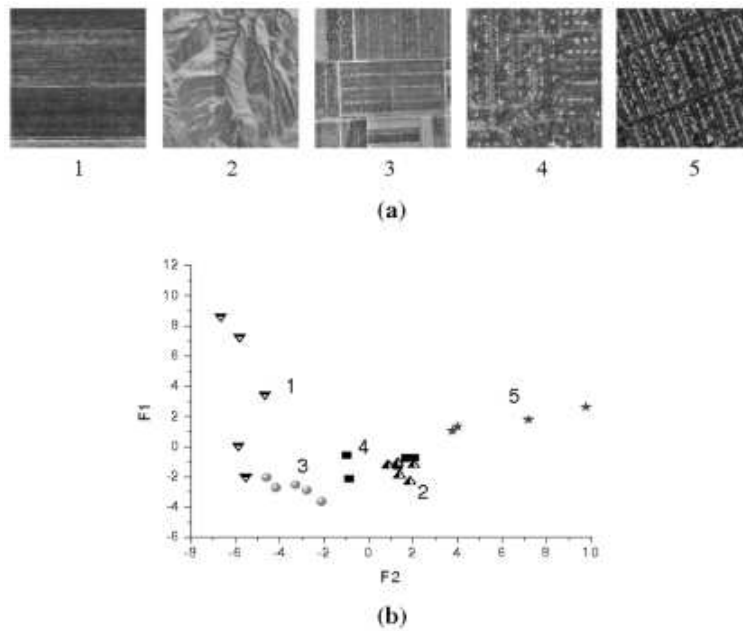


FIG. 2.9 – a) Différentes classes d’images aériennes b) Les points des classes représentés suivant les deux axes principaux

s’ajoutent les facteurs de distorsion classiques tels que les conditions d’acquisition, l’effet de l’optique, des capteurs, etc.

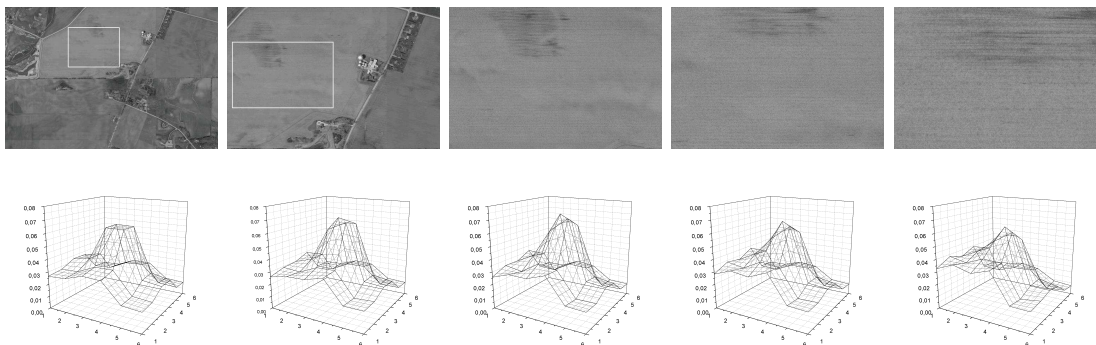


FIG. 2.10 – Les valeurs de MCM (6 motifs) en fonction des colonnes et des lignes correspondant aux images ci-dessus

## 2.9 Description par valeur des parcours optimaux

Dans cette section nous allons introduire une nouvelle approche qui est basée également sur les parcours locaux mais qui diffère en termes de représentation. En effet chaque parcours appliqué sur un voisinage de 4 pixels engendre une valeur de parcours optimal

(VPO) désignée selon  $\delta$  (voir section 2.7). Cette valeur peut être considérée comme indice de variation des niveaux de gris dans un voisinage. Une grande valeur de  $\delta$  traduit une forte variation des niveaux de gris, si elle est plutôt faible le voisinage est plutôt lisse. Dans le cas particulier où  $\delta$  est nul tous les pixels parcourus ont la même valeur. Cette représentation offre aussi l'avantage d'être robuste au changement monotone de l'intensité comme décrit dans la section 2.8.3. Il est donc intéressant de représenter un bloc de pixels par la valeur de  $\delta$  optimale, la transformation d'image étant déterminée par l'optimalité choisie. Cette transformation peut également se concevoir comme un filtrage obtenu par convolution avec un ensemble de 6 masques organisés selon 12 combinaisons (12 parcours). Les figures 2.11 et 2.12 illustrent un exemple de ce filtrage.

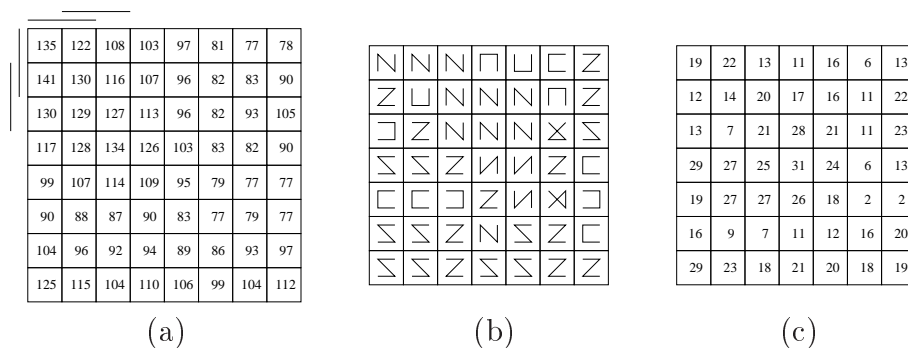


FIG. 2.11 – a) image originale b) image motifs c) image VPO

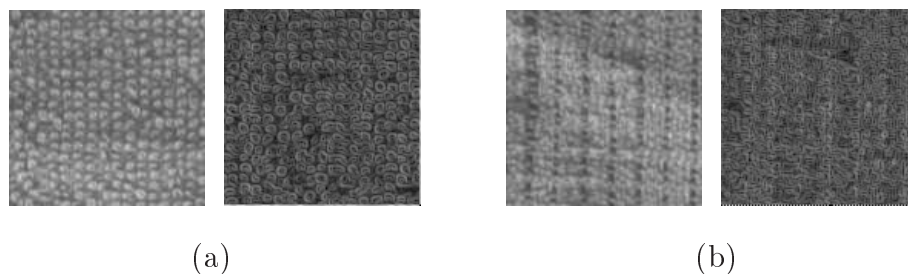


FIG. 2.12 – a) b) image originale/image vpo avec egalisation de l'histogramme

### 2.9.1 Matrice de co-occurrence des VPO

Une fois encore nous mesurons une probabilité jointe pour décrire la texture. Cette fois ci la matrice de co-occurrence, notée "MCVPO", est calculée pour les VPO ; les calculs sont les mêmes que pour GLCM. En revanche le principe résultant est différent puisqu'on détermine en fait la relation spatiale entre des indices de la variation de niveau de gris. L'occurrence de  $\delta$  est indépendante de l'occurrence des niveaux de gris de même que la valeur d'une fonction en un point ne préjuge pas de celle de sa dérivée. Par exemple, trivialement pour un groupe de pixels tels que  $p_1 = p_2 = p_3 = p_4 = c$  alors  $\delta = 0$  quelle que soit la valeur de  $c$ . Dans ce cas les valeurs de GLCM dépendent de la valeur de  $c$ .

### 2.9.2 Filtres de Gabor des VPO

Pour mieux apprécier le changement dans la répartition spectrale, nous appliquons les filtres de Gabor sur les images de valeurs optimales (FGVPO). On constate que la transformation en VPO implique un changement de fréquence spatiale, comme on peut le remarquer sur la figure 2.13. L'analyse dans le domaine fréquentiel montre que le spectre de puissance de l'image originale exhibe plusieurs pics de lobe d'harmonique au croisement des lignes verticales et horizontales : ceci est dû à la forme générale de la texture qui est un arrangement de petits objets carrés. Après transformation en VPO le spectre de puissance de l'image résultat devient celui de la figure 2.13 (b) on constate un étalement du spectre et plus de hautes fréquences. Ce n'est pas contradictoire avec le critère d'optimalité qui a pour but de régulariser en réduisant l'effet des transitions rapides dans un voisinage. En effet la transformation en VPO remplace globalement les niveaux par des gradients (Eq. 2.7) donc privilégie les hautes fréquences spatiales. Une description efficace par filtres de Gabor nécessite une bonne résolution en fréquence et une bonne préservation des contours. Ces deux objectifs sont contradictoires, il peut s'avérer difficile de conjuguer les deux propriétés. L'image VPO réalise un certain compromis entre les deux critères. Une fois l'image filtrée, on lui applique une fonction non linéaire pour calculer l'énergie de chaque pixel (Eq. 2.2) et le résultat est pris pour caractéristique de la texture.

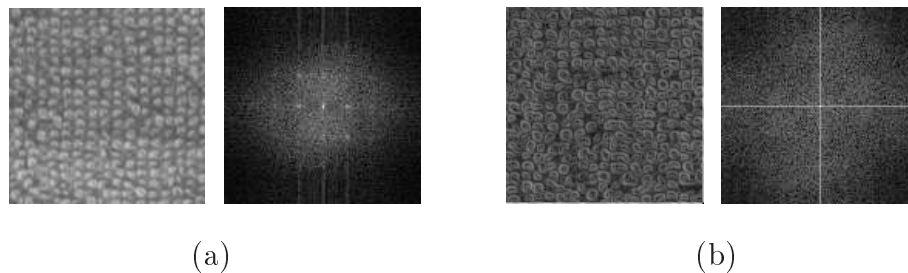


FIG. 2.13 – a) b) FFT d'image original et image VPO

## 2.10 Evaluation des performances

Reste à vérifier la capacité des attributs proposés à capter les propriétés de surface des régions identifiées dans l'image. L'évaluation des méthodes de description de texture consiste à vérifier les propriétés suivantes :

- L'invariance aux variations d'illumination : le procédé doit fournir la même valeur d'indice malgré des variations de niveaux de gris, par exemple dans les images aériennes dues à un changement des conditions météorologiques qui peut modifier localement l'intensité lumineuse dans l'image (e.g. ombre d'un nuage).
- L'invariance aux distorsions géométriques telles que par exemple la position relative de la camera peut en induire.

- L'invariance en résolution ou facteur d'échelle : l'algorithme doit permettre la reconnaissance d'une texture en cas d'éloignement ou rapprochement, et plus généralement tout décalage en fréquence
- La robustesse : la méthode doit être consistante vis à vis du bruit ou d'autres artefacts.
- La taille minimale de la fenêtre nécessaire pour capturer l'information de texture.
- La base de textures : le choix des images de test est un facteur des performances d'un algorithme
- Le temps d'exécution ou la complexité.

L'évaluation supervisée d'une mesure de similarité entre textures constitue le test de la pertinence des caractéristiques proposées. Le caractère discriminant de ces dernières supportera la détermination des classes identifiées comme étant de texture différente. La classification dépend en effet du choix des attributs (espace des variables) et de la distance ou du modèle de décision choisi. Les mesures des performances sont basées sur des techniques statistiques de classification qui peuvent être de type paramétrique (e.g. théorie bayésienne) ou non paramétrique telles que les K plus proches voisins (KPP) ou les réseaux de neurones et dérivés. L'évaluation des performances dépend donc du choix du classifieur, mais encore de la plateforme expérimentale comprenant des algorithmes de référence, ainsi que des images de test. Le caractère très subjectif de ces choix a engendré des polémiques autour de la validité de telles ou telles mesures expérimentales, notamment le choix des tests de référence [HS92]. Pour remédier à ça des équipes de recherche [SB97] ont construit des plateformes expérimentales accessibles sur le Web et qui sont devenues des "benchmark" pour la mesure des performances. Pour la confrontation avec des algorithmes classiques nous avons choisi le logiciel (open source) de Meastex<sup>1</sup> qui comprend des procédures de référence codées en langage C. Cette plateforme récemment créée est devenue d'après sa fréquence d'emploi dans la littérature la meilleure base de référence pour les tests d'opérateurs de textures. Cependant les bases d'images de test offertes dans Meastex sont pauvres qualitativement et quantitativement comparé à d'autres comme Outex<sup>2</sup> [OMP<sup>+</sup>02]. Son inconvénient majeur est sa pauvreté en images naturelles ou de scène réelles.

### 2.10.1 La base Outex

Outex contient une large variété de textures organisées en plusieurs catégories selon des critères liés aux problèmes à résoudre. On y trouve par exemple des textures prises sous différentes conditions d'illumination pour tester l'invariance au changement d'éclairage, des prises de vue avec rotation ou translation en vue d'invariance au changement géométrique etc. Le système d'acquisition d'image est montré figure 2.14. Il réunit une source de lumière (Macbeth SpectralLight II) et une caméra CCD (Sony DXC-755P) suspendue à un bras de robot. La station de travail pilote les composantes du système. Elle

---

<sup>1</sup><http://www.cssip.elec.uq.edu.au/guy/meastex/meastex.html>

<sup>2</sup><http://www.outex.oulu.fi>

---

permet de contrôler la source pour obtenir l'illumination désirée, la caméra pour obtenir la résolution souhaitée, le bras de robot pour opérer des mouvements de caméra et la carte d'acquisition pour le format et la taille d'image (24-bit RGB). Les positions relatives entre source de lumière, caméra et échantillon texturé sont illustrées par la figure 2.14. L'acquisition des images est réalisée avec trois types d'éclairage simulés : lumière naturelle horizontale, source incandescente (Inc) ou fluorescente (tl84). Ce système fournit des images référencées de texture à partir des surfaces de matériaux tels que le textile, le bois, le plastique... La figure 2.15 montre quelques exemples de textures produites par ces surfaces.

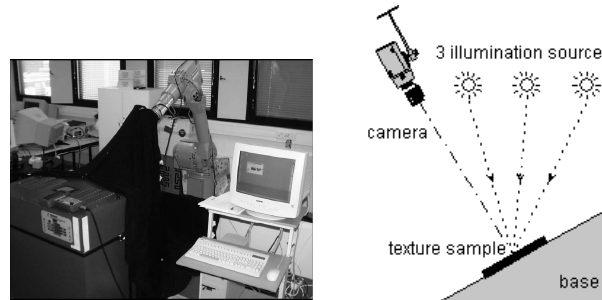


FIG. 2.14 – Système d'acquisition Outex

### 2.10.2 Le classifieur des "K plus proches voisins"

La méthode dite des K Plus Proches voisins, KPP, a engendré un ensemble d'algorithmes de base pour la classification et la reconnaissance de forme : sa simplicité et son efficacité ont conduit à l'utiliser communément dans l'évaluation de performance de caractéristiques textuelles. Les KPP présentent l'avantage par rapport aux méthodes paramétriques de ne pas demander une connaissance *a priori* ou hypothèse initiale sur les données. En revanche le paramètre principal que constitue K est souvent défini de manière empirique, le nombre optimal de voisins reste une notion subjective que le choix de la distance vient encore relativiser. Un algorithme "des k plus proches voisins" mesure une similarité entre des échantillons de test et des exemples d'apprentissage dont la classe est déjà bien déterminée. Les k exemples les plus semblables d'une épreuve sont ses k plus proches voisins. La mesure de similarité est le plus souvent la distance Euclidienne entre vecteurs de caractéristiques, c'est la distance que nous utiliserons également. La distance entre deux MCM sera donc donnée par l'équation suivante :

$$D(MCM^1, MCM^2) = \sum_{p=1}^n \sum_{i=1}^{12} \sum_{j=1}^{12} (MCM_{ij}^{1p} - MCM_{ij}^{2p})^2 \quad (2.9)$$

où  $n$  est le nombre des plans dans l'image.

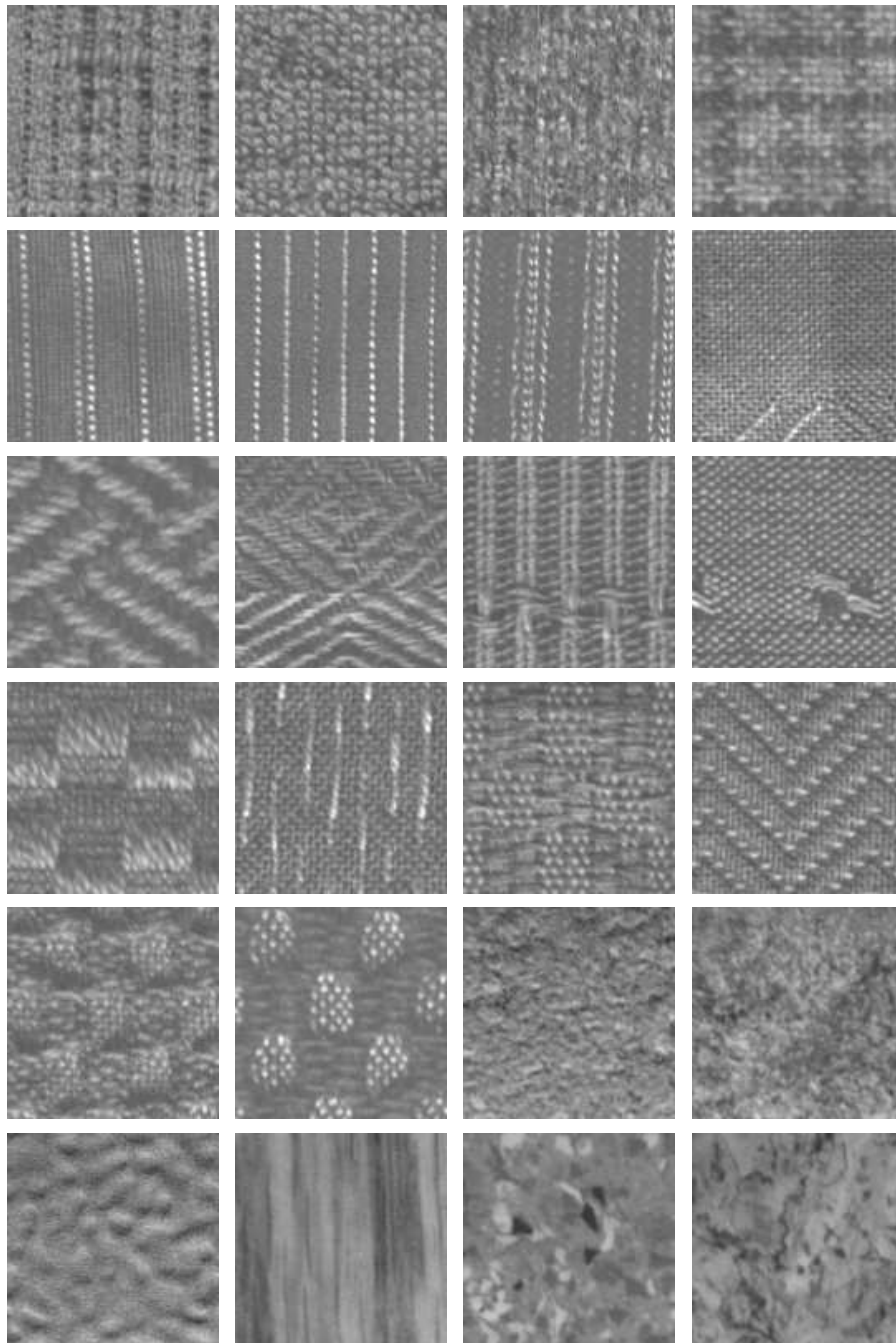


FIG. 2.15 – Echantillons de classes de texture Outex

### 2.10.3 Paramétrage

Pour évaluer notre approche il est naturel de comparer avec la matrice de co-occurrence des niveaux de gris (GLCM) ainsi qu'avec d'autres méthodes classiquement admises comme efficaces pour la caractérisation des textures telles que les filtres de Gabor. Le paramétrage des algorithmes est comme suit :

- GLCM :  $\theta = 0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 270^\circ, 315^\circ$ . La distance  $d = 1$ , la dynamique est de 256 niveaux de gris. Caractéristiques : énergie, inertie, entropie, inverse des différences des moment, différence d'entropie, différence de variance, somme des moyennes, somme des variances, somme d'entropie, corrélation d'Haralick.
- Filtre de Gabor :  $\lambda = 2, 4, 8, \theta = 0^\circ, 45^\circ, 90^\circ, 135^\circ$ , taille de la fenêtre =  $17 \times 17$  pixels.

## 2.11 Résultats et discussion

Nous conduisons une évaluation quantitative des performances en vérifiant des critères mentionnés ci-dessus tels que l'effet du changement d'éclairage, de la résolution, ... Nos expériences sont réalisées sur des images de la base Outex. Les performances des caractéristiques sont mesurées d'après la discrimination entre classes de texture par minimisation d'une distance euclidienne. Le taux de succès est calculé par l'association des KPP avec l'algorithme de "winner take all". La stratégie de ce dernier consiste à mettre à 1 le taux de succès si les plus proches voisins appartenant à la même classe que l'échantillon sont absolument majoritaires, ce taux vaut 0 s'ils sont minoritaires. En cas de majorité relative le taux de réussite appartient à l'intervalle  $[0,1]$  et sa valeur dépend de la participation des autres classes. Nous présentons ici des résultats détaillés obtenus pour plusieurs catégories de textures. Soulignons encore que toutes les méthodes ont été testées sous les mêmes conditions. Les MCM et GLCM sont calculées pour une distance fixe et toutes les orientations : la notation MCM6 représente MCM à 6 motifs [JCSZ04] et MCM12 dénote la MCM à 12 motifs décrite dans ce chapitre.

**Remarque 2** *Les MCM utilisées dans les expérimentations sont calculées avec recouvrement. Nous complétons les résultats d'expériences par une comparaison séparée entre MCM avec et sans recouvrement.*

### 2.11.1 Classes de texture générale

Dans ces expériences nous comparons les performances des caractéristiques pour chaque classe de texture. Les tests ont été effectués sur les images de la catégorie Outex\_TC\_00000 contenant 24 classes de texture ; chaque classe comprend 20 images monochromes de taille

---

128×128 avec une illumination incandescente constante et une résolution de 100dpi (figure 2.15). Ces images sont divisées en deux groupes test et apprentissage, chaque groupe contient 240 images indexées par leurs classes d'appartenance. Nous testons 10 images par classe en utilisant les KPP (section 2.10.2). Chaque image de test est comparée avec tous les exemples d'apprentissage. Le score de chaque classe est obtenu par le calcul de la moyenne sur les 10 images. Le tableau 2.1 illustre une comparaison entre 6 différentes méthodes. Le nombre  $k$  des plus proches voisins est fixé à 3. On remarque dans tous les cas l'écart important entre GLCM et les MCM, FGVP ainsi que le filtre de Gabor. Pour la GLCM nous avons testé systématiquement plusieurs valeurs de paramètres : les résultats présentés ici sont les meilleurs obtenus. On constate également que MCM12 produit un taux de réussite supérieur à MCM6 et se compare favorablement aux filtres de Gabor. La transformation d'image en VPO augmente les performances d'une manière consistante que ce soit pour GABOR ou GLCM. La combinaison entre le filtrage de Gabor et les VPO (FGVPO) produit le meilleur résultat par rapport aux autres techniques. Certaines textures posent des problèmes pour Gabor et GLCM, en revanche elle sont correctement détectées par MCM6 et MCM12 et réciproquement pour certaines autres textures.

### 2.11.2 L'effet des transformations géométriques

Le système visuel humain arrive rapidement à se recalculer en cas de transformations géométriques d'une scène ou d'une image, il n'en va pas de même pour les systèmes de vision artificielle. La complexité du recalage est naturellement fonction du nombre de paramètres définissant la transformation géométrique. Donc les rotations, plus complexes, posent plus de problèmes que les translations. Nous avons effectué les tests sur des images ayant subi différentes rotations selon plusieurs angle. La catégorie Outex\_TC\_00010 répond à ce besoin où chaque classe comporte 20 images avec les rotations suivantes : 0°,05°,10°,15°,30°,45°,60°, 75°,90°. Il en résulte 180 images par classe Cela augmente le nombre d'échantillons et d'exemples à tester par rapport à la catégorie Outex\_TC\_00000. Cependant les autres conditions et paramètres restent inchangés par rapport à la section précédente. Les exemples d'apprentissage correspondent à un jeu de 480 (20/classe) images originales, les échantillons de tests comprennent 3840 élément toutes classes confondues. On peut apprécier les résultats sur la catégorie Outex\_TC\_00010. Le tableau 2.2 montre la dégradation du score pour les MCM, FGVPO et les filtres de Gabor par rapport aux résultats obtenus pour Outex\_TC\_00000. Cela est dû au déplacement important des pixels et vérifié d'autant plus sur les grandes rotations. Comme les parcours sont fortement sensibles aux relations de voisinage, une fois cette relation modifiée par le fait que les pixels prennent des positions différentes dans le quadrant  $2 \times 2$  il est difficile de récupérer les positions originales. Dans ce cas les parcours optimaux peuvent changer et du coup les MCM et les VPO correspondantes. Bien que la GLCM demeure invariante à ces transformations géométriques elle reste moins efficace. On remarque une dégradation importante du taux de réussite pour les filtre de Gabor. Les performance FGVPO restent supérieures à celles d'autres méthodes.

---



classe	MCM6	MCM12	MCVPO	GLCM	FGVPO	Gabor
canvas001	1.000	1.000	0.267	0.900	1.000	1.000
canvas002	1.000	1.000	0.400	0.200	1.000	1.000
canvas003	0.900	1.000	0.067	0.033	1.000	1.000
canvas005	1.000	1.000	0.233	0.100	1.000	1.000
canvas006	0.900	1.000	0.067	0.800	1.000	1.000
canvas009	1.000	1.000	0.533	1.000	1.000	1.000
canvas011	1.000	1.000	0.400	0.567	1.000	1.000
canvas021	1.000	1.000	0.600	0.033	1.000	1.000
canvas022	0.733	0.867	0.467	0.133	1.000	0.900
canvas023	0.833	0.933	0.067	0.167	0.600	0.733
canvas025	1.000	1.000	0.100	0.367	0.800	1.000
canvas026	1.000	1.000	0.700	0.133	1.000	1.000
canvas031	1.000	0.900	0.167	0.033	1.000	0.900
canvas032	0.533	0.900	0.200	0.433	1.000	1.000
canvas033	0.533	0.633	0.033	0.100	0.600	0.800
canvas035	1.000	1.000	0.233	0.067	1.000	1.000
canvas038	0.700	0.767	0.200	0.067	1.000	0.800
canvas039	0.500	0.733	0.033	0.067	0.900	0.800
carpet002	0.767	0.700	0.300	0.233	1.000	0.900
carpet004	0.733	0.900	0.067	0.000	0.900	0.667
carpet005	1.000	1.000	0.600	1.000	1.000	1.000
carpet009	1.000	1.000	0.800	0.067	1.000	1.000
tile005	1.000	0.900	0.200	0.067	0.900	0.533
tile006	0.600	0.733	0.333	0.067	0.700	0.701
Total :	0.864	0.915	0.294	0.276	0.933	0.906

TAB. 2.1 – Comparaison des caractéristiques pour chaque classe de la catégorie Outex\_TC\_00000

Catégorie	MCM6	MCM12	MCVPO	GLCM	FGVPO	Gabor
Outex_TC_00010	0.454	0.484	0.311	0.282	0.514	0.349
Outex_TC_00011	0.803	0.828	0.284	0.294	0.747	0.623
Outex_TC_00014	0.724	0.714	0.153	0.113	0.490	0.454

TAB. 2.2 – Comparaison des caractéristiques pour chaque catégorie de problème

### 2.11.3 L’effet de la résolution

Le but ici est d’étudier l’effet du changement multi échelles en testant des textures prises à différentes résolutions. Pour cette fin on utilise la catégorie Outex\_TC\_00011 qui comporte des images prises à deux résolution 100dpi et 120dpi. Dans une classe il existe 20 images pour chaque résolution. Le nombre de classes, la taille des images et les conditions d’éclairage ainsi que les paramètres des algorithmes sont identiques à ceux indiqués précédemment. Les résultats sont résumés dans le tableau 2.2. On peut remarquer que les MCM sont moins sensibles au facteur d’échelle que les autres méthodes. Bien que robustes les MCM finissent par produire des résultats incohérents si le changement d’échelle est très important et conduit à un changement catastrophique d’apparence visuelle.

### 2.11.4 L’effet de l’éclairage

Dans cette section nous vérifions l’impact du changement d’éclairage sur le calcul des attributs. Les expériences sont menées sur la catégorie Outex\_TC\_00014 dédiée à ce problème. Cette dernière contient 68 classes de textures. Pour rester conforme aux expériences précédentes on ne teste que les 24 classes présentées dans ce chapitre. Chaque classe contient des images de la même texture, mais prises sous trois éclairages différents (voir section 2.10.1). Au total il y a 20 images pour chaque type de lumière. Les résultats obtenus pour la catégorie en question sont montrés dans le tableau 2.2. On constate une fois de plus que les MCM produisent des meilleurs résultats.

### 2.11.5 MCM sans recouvrement

Une comparaison entre les deux méthodes de calcul de MCM a montré que les MCM avec recouvrement supportent une meilleure discrimination que ceux calculés sans recouvrement. Cette comparaison implique les différentes catégories évoquées précédemment. Le tableau 2.3 illustre les résultats obtenus pour MCM6 et MCM12. On constate que le seul cas où MCM sans recouvrement dépasse MCM avec recouvrement est le test en multi résolution. Cela est dû au sous échantillonnage associé au calcul de MCM sans recouvrement qui engendre un équivalent de niveau de résolution plus élevé et donc accroît la robustesse.

classe	MCM6		MCM12	
	sans recouv	avec recouv	sans recouv	avec recouv
Outex_TC_00000	0.882	0.864	0.907	0.915
Outex_TC_00010	0.422	0.454	0.458	0.484
Outex_TC_00011	0.826	0.803	0.855	0.828
Outex_TC_00014	0.720	0.724	0.690	0.714

TAB. 2.3 – Comparaison des deux méthodes de calcul des MCM, avec recouvrement et sans recouvrement

	k=1	k=3	k=5	k=7	k=9
MCM6	0.900	0.864	0.822	0.795	0.747
MCM12	0.954	0.915	0.873	0.849	0.820
MCVPO	0.321	0.294	0.304	0.282	0.299
GLCM	0.283	0.276	0.293	0.288	0.311
FGVPO	0.933	0.933	0.917	0.902	0.889
Gabor	0.900	0.906	0.844	0.856	0.824

TAB. 2.4 – Variation du nombre de KPP

### 2.11.6 L'effet du nombre de KPP

Nous avons testé le comportement des différentes méthodes fonction du nombre de KPP. Le tableau 2.4 résume l'évaluation pour des variations croissantes de KPP. Pour  $k = 1$  MCM12 produit le meilleur résultat, pour les autres valeurs de  $k$  c'est toujours FGVPO qui exhibe les plus hautes performances. On remarque en outre que FGVPO présente une bonne stabilité par rapport à ces changements de  $k$ . La valeur optimale de  $k$  pour les filtres de Gabor est 3, pour les autres méthodes elle est de 1.

## 2.12 Conclusion

On a vérifié expérimentalement, sur un échantillon de textures considéré comme représentatif dans la littérature, que les parcours locaux associés à des mesures statistiques ou des méthodes fréquentielles décrivent ces textures d'une manière efficace. À des fins de comparaison nous avons proposé une description détaillée de ces méthodes ainsi que des caractéristiques résultantes. Les expériences effectuées sur les bases d'images dédiées "Benchmark" ont montré l'amélioration des performances des approches proposées comparé à des approches plus classiques et reconnues comme techniques de références. L'évaluation a pris en compte des différents problèmes de robustesse aux perturbations associés à l'identification de textures.





## Chapitre 3

# Extraction d'entités visuelles par décomposition en régions

### 3.1 Introduction

Le contenu d'une image peut être décrit par un ensemble (organisé) de composantes visuelles. Les relations entre elles traduisent donc une sémantique limitée. Leur détection est une tâche ardue et un problème fondamental du Traitement d'Image. Une solution élémentaire mais naturelle fait partie de la segmentation et consiste à partitionner l'image en régions homogènes selon certains critères tels que la couleur, la texture ou la forme. Pour décomposer l'image en entités visuelles nous utilisons des techniques de regroupement (Clustering) améliorées afin d'obtenir une segmentation à la fois plus efficace et plus cohérente. L'image est donc considérée comme un nuage de points représentés dans un espace de caractéristiques. Les points similaires sont regroupés sous contraintes en amas (Cluster) eux-mêmes décomposés naturellement en parties connexes de l'image i.e. régions séparées. Les méthodes que nous proposons reposent sur la technique de C-Moyennes floues "Fuzzy C-Means" (FCM). Bien que ce type de technique soit considéré comme méthode d'apprentissage non supervisé le nombre de classes est ici introduit manuellement. Ceci rend le résultat subjectif en même temps que la tâche difficile pour une grande base d'images. Afin de surmonter ce dernier problème nous proposons une technique qui permet d'extraire automatiquement le nombre de groupes que peut éventuellement contenir le nuage de points en se basant sur leur densité et leur séparation.

---



FIG. 3.1 – Image de synthèse représentant une scène urbaine

## 3.2 Description par régions

Une région de l'image correspond à un ensemble de pixels qui présentent une certaine homogénéité visuelle par exemple de couleur, de texture, de forme globale émergente, etc. Une région peut contenir un objet ou une partie de celui-ci. La notion d'objet est étroitement liée à la perception humaine. C'est ainsi que les régions et les relations entre elles accèdent à une certaine sémantique basique et limitée du contenu image. Les techniques de CBIR fondées sur les régions (approche locale) annoncent des résultats prometteurs par rapport aux méthodes optant pour un traitement global sur l'image [CTB+99] [MM99]. En effet la décomposition de l'image en plusieurs entités permet un plus grand degré de liberté dans le processus de recherche justement favorisé par cette sémantique de bas niveau. Le but des systèmes CBIR basés sur les régions est d'obtenir une description la plus complète possible d'un ou plusieurs objets dans l'image. Une fois ces derniers détectés, il est plus facile au système d'identifier les images cibles. D'où un gain en performance [WLW01].

Les méthodes locales sont donc conditionnées par la détection automatique d'objets qui nécessite une segmentation précise. Si on note une région par " $R$ " et l'objet physique à détecter par " $O$ ". Une segmentation idéale regrouperait donc ici les points connexes d'une image dans  $R$  telle que  $R = O$ . Cette description demeure très difficile à obtenir avec les méthodes actuelles et à ce niveau de segmentation, en particulier pour les objets constitués de plusieurs entités visuellement différentes (par construction, occlusion, ombrage etc.). Shi et Malik [SM97] supposent que la segmentation ne pourra pas et ne devra pas fournir un résultat final objet complet, mais qu'elle engendre une séquence de partitionnement hiérarchique basée sur les attribut de bas niveau et assurant une cohérence de couleur, texture ou mouvement. Le rôle de la segmentation serait alors limité à regrouper en régions homogènes des points qui partagent les mêmes propriétés. Ce d'autant plus que les techniques de CBIR montrent une certaine tolérance à la segmentation imparfaite parce que le processus de recherche de régions similaires s'accommode d'une mise en correspondance partielle entre composantes de l'image. Une détection de région sera donc considérée pertinente si elle satisfait un critère moins exigeant de type  $R \in O$ .

Notre but est de capter des zones d'intérêt qui seront significantes pour l'utilisateur, afin d'apparier les images d'une manière plus sûre. Par exemple l'image de la figure 3.1 représente du ciel, des bâtiments et des voitures... si l'utilisateur associe les objets bâtiments et voitures cela permet de guider la recherche dans une direction plus précise. Il augmentera encore la sémantique en ajoutant des contraintes spatiales sur les objets dans l'image, nous abordons cette notion dans le prochain chapitre.

### 3.3 La segmentation d'image

La segmentation est :

- Essentielle parce qu'elle constitue l'étape préliminaire à l'analyse d'image et la reconnaissance de formes.
- Critique parce que du résultat final dépend notamment la recherche d'image par le contenu.

Une grande activité de recherche lui a été consacrée durant ces trois dernières décennies mais elle reste un problème difficile dans son aspect universel. Comme décrit ci-dessus on admet donc que le rôle de la segmentation consiste à partitionner l'image en régions respectant une certaine cohérence visuelle et une continuité spatiale. Les éventuelles imperfections de segmentation pourront être surmontées par le processus de mise en correspondance fondant la recherche en base de données. Nous retenons donc la définition donnée par Horowitz et Pavlidis [HP76].

**Définition 3.3.1** Soit  $I(x, y)$  la valeur du pixel en  $(x, y)$  et  $P()$  un prédicat qui mesure l'uniformité sur un groupe de pixels connexes : la segmentation est définie comme une partition de  $I$  en  $n$  sous-ensembles ou régions  $\{R_1, \dots, R_n\}$  tels que :

$$\begin{aligned} \cup_{i=1}^n R_i &= I \\ R_i \cap R_j &= \Phi \quad (i \neq j) \\ P(R_i) &= \text{vrai} \quad \forall i \in \{1, \dots, n\} \\ P(R_i \cup R_j) &= \text{faux} \quad \text{si } i \neq j \end{aligned}$$

Une grande variété de méthodes de segmentation a été proposée dans la littérature. Les techniques de segmentation sont généralement catégorisées en 4 groupes : fondé sur les contours, sur les régions, par regroupement et les hybrides. On trouve des articles de synthèse [HS85, FM81, PP93] portant sur l'état de l'art antérieur aux années 1990. Ces techniques ont le plus souvent été mises au point sur des images mono canal à cause des contraintes de calcul et de la capacité mémoire. Cela n'a pas été trop gênant parce que selon plusieurs modèles de la vision humaine, nous serions plus sensibles aux variations de luminosité qu'à la sensation de couleur. Cette hypothèse trouve une application par



exemple en compression d'images couleur (JPEG) où on sépare la chrominance de la luminance pour coder la couleur sur moins de bits. Les avancées technologiques des moyens de calcul et de stockage ont permis la banalisation des images en couleur, d'où l'utilisation plus récente de cette dernière comme primitive dans le processus de segmentation. Pendant cette dernière décennie on note une augmentation remarquable des techniques de segmentation traitant la couleur (voir notamment [SK94, LM01, CJS<sup>+</sup>01] qui en tracent un état de l'art).

## 3.4 Particularité de la segmentation pour CBIR

Dans le contexte du CBIR relatif à des bases généralistes la segmentation est confrontée à des problèmes plus larges. Comme déjà dit en 3.2 la recherche d'image par le contenu est en partie subjective et la segmentation vise à réduire le fossé sémantique en créant des régions significatives. Nous listons à nouveau ici les principaux systèmes de CBIR qui utilisent la segmentation pour fonder une recherche dans les bases d'image. Cette liste n'est évidemment pas exhaustive. Le système "BlobWorld" [CBGM02] repose sur l'algorithme Expectation Maximisation (EM) [DLR76] pour extraire des régions. L'algorithme est employé pour le regroupement des pixels en exploitant la couleur, la texture et la position comme primitives avant un post traitement supposé éliminer le bruit de la segmentation. "Netra" [MM99] est un autre système segmentant les images en régions homogènes grâce à un algorithme basé sur le modèle prédictif nommé "edge flow" [MM97] avec les mêmes primitives. "SIMPLIcity" adopte une stratégie de recherche différente : la segmentation exploite les C-Moyennes. L'image est divisée en blocs 4x4, chacun représenté par 6 attributs de couleur et texture. Citons encore un couple de travaux qui adressent la même problématique et diffèrent par les méthodes de segmentation [MANL03] [FB04]. La plupart des méthodes citées ci-dessus emploient des algorithmes fondés sur le regroupement partant du principe que la perception humaine tend à regrouper les points homogènes. Notre travail est motivé par ce même principe. Nous développons des algorithmes encore optimisés pour notre type d'application.

### 3.4.1 Algorithme de C-Moyennes

Le C-Moyennes standard est un algorithme général de regroupement qui tend à rassembler  $N$  points ou primitives dans  $C$  groupes dont le nombre est prédéterminé. Étant donné un ensemble de points  $X = x_1, x_2, \dots, x_N$ . Chaque point  $x_j$  est caractérisé par  $n$  variables (attributs) et donc représenté dans un espace à  $n$  dimensions  $x_j \in R^n$ . L'algorithme de C-Moyennes minimise la fonction objectif suivante :

---

$$J(X, V) = \sum_{i=1}^C \sum_{j=1}^N \|x_j - v_i\|^2 \quad (3.1)$$

où  $v_i$  est le centre du cluster  $i$  et  $\|\cdot\|$  représente la norme, ici Euclidienne, qui mesure la dissimilarité entre un point et un centre. L'algorithme se résume en les étapes suivantes :

---

**Algorithm 1:** Algorithme de C-Moyennes
 

---

Step 1. Choisir aléatoirement les  $C$  centres initiaux  $v_1, v_2, \dots, v_C$  à partir des points  $\{x_1, x_2, \dots, x_N\}$

**repeat**

Step 2. Assigner  $x_j, j = 1, 2, \dots, N$  au clusters  $Z_i, i = 1, 2, \dots, C$

Si

$\|x_j - v_i\| \leq \|x_j - v_p\|, p = 1, 2, \dots, C$  avec  $j \neq p$

Choix aléatoire si conflit

Step 3. Calculer les nouveaux centres :

$v_i^* = \frac{1}{n_i} \sum_{x_j \in Z_i} x_j, i = 1, 2, \dots, C$

$n_i$  est le nombre d'éléments dans le cluster  $Z_i$

**until**  $v_i^* = v_i, \forall i = 1, 2, \dots, C$

---

### 3.4.2 Classification floue

Il s'agit de partitionner un nuage de points selon les ensembles flous tels qu'ils ont été définis par Zadeh [Zad65]. Dans notre cas, les ensembles flous sont les nuages de points et les objets sont les pixels.

**Définition 3.4.1** *Un ensemble flou  $A$  est caractérisé par une fonction  $f_A$  qui varie continuellement sur  $[0, 1]$  et qui associe à chaque objet  $x$  d'un ensemble  $\xi$  la valeur  $f_A(x) \in R$  considérée comme le degré d'appartenance de  $x$  à  $\xi$ .*

### 3.4.3 Algorithme de C-Moyennes Floues

Comme nous l'avons décrit ci-dessus l'algorithme de C-Moyennes compte parmi les algorithmes classiques de regroupement qui affectent chaque point à une seule classe à chaque étape. Pour pallier ce qui apparaît comme une limitation dans de nombreux cas, le C-Moyennes flou (FCM) établit un degré d'appartenance à une classe donnée. Cette représentation correspond mieux au cas réel que le "Hard Clustering" : par exemple les points des bords d'un objet dans une image naturelle sont généralement imprécis et

---

peuvent appartenir à plusieurs classes à la fois jusqu'à décision finale, voire au delà. FCM détermine une partition floue d'un ensemble de données  $X = \{x_j \mid j = 1, 2, \dots, N\}$  sur  $C$  classes et cela également par la minimisation d'une fonction objectif :

$$J(X, U, V) = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m \|x_j - v_i\|^2 \quad (3.2)$$

$U = \{[u_{ij}] \mid u_{ij} \in [0, 1]\}$  est la matrice de partition floue qui détermine le degré d'appartenance d'un point  $x_j$  à un cluster  $i$ . Elle satisfait les conditions suivantes :

$$0 < \sum_j^N u_{ij} < N, \quad \forall i \in \{1, 2, \dots, C\} \quad (3.3)$$

$$\sum_i^C u_{ij} = 1, \quad \forall j \in \{1, 2, \dots, N\} \quad (3.4)$$

Le paramètre  $m \in [1, +\infty[$  est appelé pondérant exponentiel ou "Fuzzifier". Il détermine le degré de chevauchement entre les bords des classes. Pour une grande valeur de ce paramètre les frontières deviendront plus lisses, en revanche une petite valeur de  $m$  produit des bords en pente rapide. Une valeur de  $m$  typique est 2.

La contrainte Eq. 3.3 garantit qu'il n'y a aucun groupe vide. La deuxième contrainte Eq. 3.4 est une propriété de type stochastique et contrôle la manière dont un point influence à la fois plusieurs groupes. Certain auteurs [DHS00] considèrent  $u_{ij}$  comme la probabilité conditionnelle  $\hat{P}(\omega_i/x_j)$ .

On résout l'équation 3.2 par un algorithme itératif. Il a deux principales étapes : calcul de la fonction d'appartenance, puis définition des centres de chaque groupe. Les formules de calcul sont obtenues classiquement par annulation d'une dérivée partielle de  $J$  par rapport au terme qu'on veut optimiser [Bez81]. Le degré d'appartenance à une classe est donné par la formule :

$$u_{ij} = \frac{1}{\sum_{p=1}^C \left( \frac{\|x_j - v_i\|^2}{\|x_j - v_p\|^2} \right)^{\frac{1}{m-1}}} \quad (3.5)$$

On remarque que  $u_{ij}$  dépend de la distance des points aux centres des classes. Elle atteint en pratique une valeur maximale quand cette distance est globalement minimale :

$$u_{ij} = \begin{cases} u_{ij} & \text{si } x_j \neq v_i \\ 1 & \text{si } x_j = v_i \end{cases} \quad (3.6)$$

Les centres sont calculés selon la formule barycentrique :

$$v_i = \frac{\sum_{j=1}^N u_{ij}^m x_j}{\sum_{j=1}^N u_{ij}^m} \quad (3.7)$$

$v_i$  est ainsi une moyenne pondérée où les éléments participent avec un poids plus ou moins important : les points qui ont une appartenance forte au cluster  $i$  influencent plus la valeur du centre.

L'algorithme FCM standard est donc le suivant :

---

**Algorithm 2:** Algorithme de FCM

---

```

Step 1. Initialisation :
    Choisir le nombre de classes
     $u_{ij} \leftarrow$  valeur aléatoire  $\in [0, 1]$ 
repeat
    | Step 2. Calculer les centres en utilisant (3.7)
    | Step 3. Calculer la distance  $\|x_j - v_i\|^2$ 
    | Step 4. Mise à jour de la Matrice  $U$  par (3.5)
until  $\|U^k - U^{k+1}\| < \epsilon$ 
Step 5. Déffuzification

```

---

La convergence est atteinte quand le changement dans la matrice floue entre les itérations  $k$  et  $k+1$  est inférieur à  $\epsilon$  (e.g 0.001). La "défuzzification" consiste à affecter chaque pixel à la classe pour laquelle il présente une valeur maximale d'appartenance. En cas d'égalité le choix est aléatoire.

### 3.4.4 Corrélation Spatiale

Nous avons déjà rappelé que les images réelles du monde contiennent généralement du bruit et autres artefacts. La qualité d'image dépend des conditions d'acquisition, de stockage,...etc. Par exemple dans les images satellitaires, conditions météorologiques et capteurs constituent une source de bruit. De même la compression JPEG, que la plupart des bases de données adopte, introduit une perte d'information et une certaine distorsion au niveau des contours. Beaucoup d'algorithmes traitent les points individuellement et ne prennent pas ces conditions en compte, omettant toute dépendance spatiale inter pixel. Cela peut engendrer des ambiguïtés lors de la classification et par conséquent un partitionnement peu cohérent avec multiplication de petites régions non significantes. Ceci affecte alors les performances du CBIR car des régions de taille suffisante et donc mieux significantes sont nécessaires le plus souvent à l'extraction de caractéristiques pertinentes. A cause de ces problèmes certains systèmes ont recours à un post traitement pour éliminer le bruit et régulariser les régions [CBGM02]. En revanche rares sont les systèmes CBIR qui introduisent l'information spatiale dès le processus de clustering. Or, les travaux récents de Tolia [TP98a, TP98b, AYM99, Pha01] ont montré l'amélioration que peut apporter l'incorporation de la contrainte spatiale à ce stade du clustering. Les champs de Markov

---

(MRF) [Li95] supportent également une approche fondée sur les contraintes spatiales. Ils sont largement utilisés dans le contexte de la segmentation et fournissent des résultats intéressants, au prix toutefois d'une complexité de calcul significative due au nombre de paramètres à estimer. Il nous a semblé intéressant d'exploiter l'information spatiale et de l'introduire dans le processus de regroupement afin d'améliorer la segmentation et augmenter les performances de CBIR. Dans la suite de ce chapitre nous proposons une approche fondée sur ces algorithmes FCM mais avec une contrainte spatiale selon Peano.

## 3.5 Le choix des caractéristiques

Le processus de segmentation suppose l'image transformée pour être codée selon des primitives caractéristiques à partir desquelles on saura définir une homogénéité sur un ensemble de points. Nous avons vu que traditionnellement la couleur et la texture constituent les principaux attributs pour segmenter l'image en région. Elles sont utilisées indépendamment ou fusionnées. La couleur dépend initialement des 3 valeurs RVB, en revanche la texture peut concerner l'intensité lumineuse seule. S'il est facile d'obtenir dès l'acquisition l'information de couleur ou d'intensité en un pixel, la texture a besoin d'une certaine extension spatiale autour du pixel pour capter suffisamment d'information (voir chapitre 2). Nous allons voir plus loin qu'avec les contraintes spatiales nous pourrions obtenir une certaine homogénéité sur les textures.

### 3.5.1 Contribution du voisinage

Il s'agit donc de définir une caractéristique locale pour capter de l'information spatiale. Comme nous l'avons vu chapitre 3.2 le parcours optimal définit une relation entre les points qu'il visite. Nous exploitons particulièrement cette propriété afin d'introduire une contrainte supplémentaire au niveau des primitives.

Considérons un voisinage de 8 pixels dont le centre constitue un point de départ de la courbe de Peano. Nous formons ainsi 4 groupes à partir de ce point (voir figure 3.2). Soit  $g_1 = \{o, a, b, c\}$ ,  $g_2 = \{o, c, d, e\}$ ,  $g_3 = \{o, e, f, g\}$  et  $g_4 = \{o, g, h, a\}$  les 4 groupes issus de ce voisinage. En utilisant cette configuration on peut engendrer une valeur du scan optimal ( $\delta$ ) pour chaque groupe. Cette valeur est considérée comme une caractéristique représentant l'interaction d'un point avec ses voisins. Un poids  $w$  est attribué à cette caractéristique pour contrôler son influence. Finalement le vecteur de caractéristique s'établit comme suit :

$$x = \{a^1, \dots, a^p, w(\delta_1^1, \delta_2^1, \delta_3^1, \delta_4^1, \dots, \delta_1^p, \delta_2^p, \delta_3^p, \delta_4^p)\}$$

où  $a^p$  représente la valeur de l'intensité lumineuse dans le plan  $p$  de l'image et  $\delta_k^p$  est la VPO (valeur du parcours optimal) du groupe  $g_k / k \in \{1, 2, 3, 4\}$  dans le  $p^{eme}$  plan. Cette

méthodes nous permet de capter les basses fréquences dans le voisinage en utilisant le critère de minimisation de l'intensité lumineuse le long du parcours.

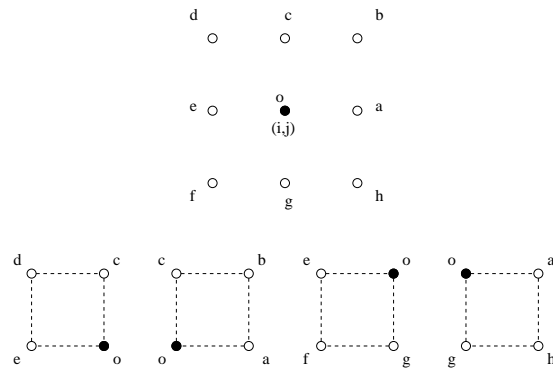


FIG. 3.2 – Decomposition du voisinage 3x3 en un ensemble de 4 groupes

### 3.6 FCM avec contrainte spatiale

Nous avons évoqué précédemment le peu de robustesse des algorithmes de regroupement tels que FCM vis à vis du bruit et autres artéfacts. La fonction objectif (3.2) est minimisée pour répartir les points dans les groupes en fonction de leur distance par rapport au centre du groupe sans prendre en compte aucune consistance locale. Pour améliorer la segmentation nous proposons de modifier FCM en rajoutant un deuxième terme à l'équation générale. Ce terme contraint la classification d'un point par ses voisins; la fonction objectif se présente donc sous la forme suivante :

$$\begin{aligned}
 J_M(U, V) &= \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m \|x_j - v_i\|^2 \\
 &+ \alpha \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m e^{-\sum_{k \in \Omega} u_{ik}^m}
 \end{aligned} \tag{3.8}$$

où  $u_{ik}$  représente toujours la fonction d'appartenance de l'élément  $k$  à la classe  $i$  et  $\Omega$  un voisinage tel que  $k \neq j$ . Le deuxième membre se comporte comme un terme de pénalité dont le paramètre  $\alpha$  contrôle l'influence. Ce terme de contrainte spatiale atteint son minimum quand la fonction d'appartenance des points du voisinage prend des valeurs maximales pour la classe  $i$ . L'optimisation de (3.8) sous la contrainte  $U$  sera résolue par la technique du multiplicateur de Lagrange sachant 3.4.

$$\begin{aligned}
 J_M(U, V) &= \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m (\|x_j - v_i\|^2 + \alpha e^{-\sum_{k \in \Omega} u_{ik}^m}) \\
 &+ \sum_{j=1}^N \lambda_j (1 - \sum_{i=1}^C u_{ij})
 \end{aligned} \tag{3.9}$$

La dérivée de (3.9) par rapport à  $u_{ij}$

$$\frac{\partial J_M}{\partial u_{ij}} = m u_{ij}^{m-1} (\|x_j - v_i\|^2 + \alpha e^{-\sum_{k \in \Omega} u_{ik}^m}) - \lambda_j \quad (3.10)$$

Pour calculer  $u_{ij}$  on annule la dérivée, soit :

$$u_{ij} = \left( \frac{\lambda_j}{m(\|x_j - v_i\|^2 + \alpha e^{-\sum_{k \in \Omega} u_{ik}^m})} \right)^{\frac{1}{m-1}} \quad (3.11)$$

En remplaçant  $u_{ij}$  dans l'équation (3.4) on obtient :

$$\sum_{i=1}^C \left( \frac{\lambda_j}{m(\|x_j - v_i\|^2 + \alpha e^{-\sum_{k \in \Omega} u_{ik}^m})} \right)^{\frac{1}{m-1}} = 1 \quad (3.12)$$

$\lambda_j$  ne dépend pas de  $i$ , donc après mise en facteur :

$$\lambda_j^{\frac{-1}{m-1}} = \sum_{i=1}^C (m\|x_j - v_i\|^2 + \alpha e^{-\sum_{k \in \Omega} u_{ik}^m})^{\frac{-1}{m-1}} \quad (3.13)$$

En substituant  $\lambda_j$  dans (3.11) nous obtenons l'équation de la fonction d'appartenance :

$$u_{ij} = \frac{1}{\sum_{p=1}^C \left( \frac{\|x_j - v_i\|^2 + \alpha e^{-\sum_{k \in \Omega} u_{ik}^m}}{\|x_j - v_p\|^2 + \alpha e^{-\sum_{k \in \Omega} u_{pk}^m}} \right)^{\frac{1}{m-1}}} \quad (3.14)$$

On remarque l'apparition du terme de contrainte spatiale dans l'équation (3.14). La valeur d'appartenance d'un point  $j$  à une classe  $i$  est désormais bien influencée par la fonction d'appartenance des voisins. Par exemple si un point a une préférence pour une classe particulière et ses voisins pour une autre classe alors le deuxième terme pénalise ce point en l'incitant à suivre le comportement du voisinage. Ceci produit un effet de régularisation. Le poids  $\alpha$  contrôle l'importance de cette régularisation : pour la valeur particulière 0 de  $\alpha$  on se retrouve dans le cas du FCM standard. Si  $\alpha$  est trop grand un lissage important se produira dans le résultat de la segmentation qui pourra affecter la forme de la région. Donc le choix du poids est important mais n'est pas critique, il doit réaliser un compromis entre les deux termes.

Pour obtenir l'équation de mise à jour des prototypes ou centres on minimise l'équation (3.8) par rapport au centre  $v_i$ .

$$\frac{\partial J_M}{\partial v_i} = 0 \Rightarrow \sum_{j=1}^N u_{ij}^m (x_j - v_i) = 0 \quad (3.15)$$

La résolution de cette équation ne dépend pas du deuxième terme ce qui nous ramène au cas du FCM : la mise à jour de  $v_i$  suit donc la même equation (3.7). Le déroulement de l'algorithme du FCM avec contrainte spatiale, qu'on note SCFCM, est résumé dans la procédure (Algorithm 3)

---

**Algorithm 3:** Algorithme de SCFCM
 

---

Step 1. Initialisation :

- Choisir le nombre de classes
- Choisir le poids  $\alpha$   $u_{ij} \leftarrow$  valeur aléatoire  $\in [0, 1]$

**repeat**

- Step 2. Calculer les centres en utilisant (3.7)
- Step 3. Calculer la distance  $\|x_j - v_i\|^2$
- Step 4. Mise à jour de la Matrice  $U$  par (3.14)

**until**  $\|U^k - U^{k+1}\| < \epsilon$

Step 5. Déffuzification

---

On remarquera que la seule différence entre cet algorithme et le FCM classique est la fonction d'appartenance  $u_{ij}$ . Pour le reste on a les mêmes étapes dans les deux cas.

## 3.7 La contrainte spatio-pyramidale

Dans cette section nous continuons de développer la méthode précédente, dans le même but, en testant d'autres contraintes pour augmenter les performances. Ici nous contraignons FCM non seulement par influence spatiale mais également par influence multirésolution permettant à l'ancêtre d'un point de conditionner sa classification.

### 3.7.1 La multirésolution

La multirésolution permet une analyse plus efficace de l'image en l'examinant dans différentes bandes de fréquence. En effet la décomposition de l'image à plusieurs niveaux de résolutions exhibe différents types de détail à chacun des niveaux. La relation hiérarchique entre éléments des couches maintient une consistance dans les propriétés de l'image. Par exemple un contour qui se manifeste à tous niveaux augmente la confiance dans sa classification. La décomposition pyramidale se fait souvent par la combinaison d'un filtrage linéaire, une convolution, avec un sous échantillonnage. Etant donné une image  $I$  de taille  $N \times N$ , et en utilisant un filtre approprié  $h$  de taille  $L \times K$ , on obtient l'image de plus

---



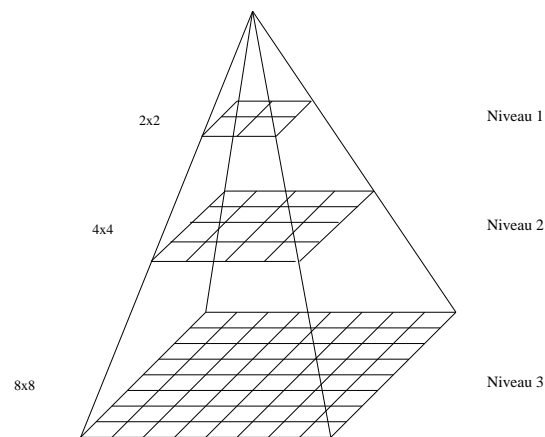


FIG. 3.3 – Representation pyramidale

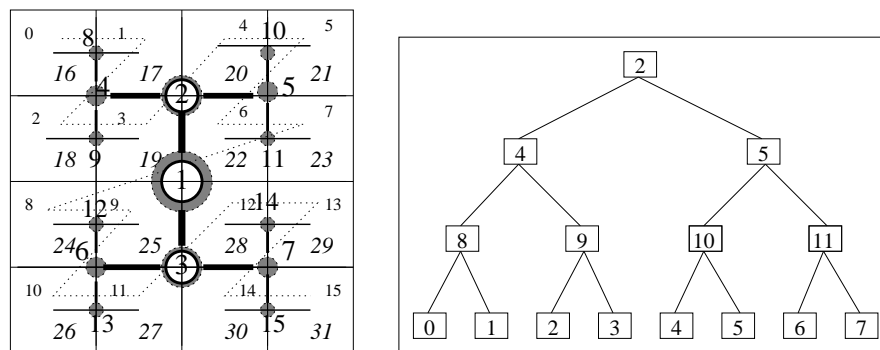


FIG. 3.4 – A gauche un Z arbre, à droite l'arbre binaire associé

faible résolution ( $n - 1$ ) selon la formule suivante :

$$I^{n-1}(x, y) = \sum_{i=1}^L \sum_{j=1}^K h(i, j) I^n(2x - i, 2y - j) \quad (3.16)$$

Parmi les filtres les plus populaires est le passe-bas Gaussien. D'autres filtres pourront être utilisés suivant le type d'application et les propriétés qu'on veut mettre en avant. La structure pyramidale est obtenue par l'application successive de la procédure Eq. 3.16 en réduisant la taille de l'image d'un niveau à l'autre dans le sens ascendant (figure 3.3). Cette structure est naturellement supportée par un arbre quaternaire ou binaire. Les parcours de Peano bénéficient d'une représentation efficace par de tels arbres, voir figure 3.4.

### 3.7.2 FCM avec contrainte spatio-pyramidale

Nous avons vu dans la section 3.6 l'intégration de l'information spatiale dans le processus de segmentation par une contrainte sur les fonctions d'appartenance. La représen-

tation pyramidale fournit une information supplémentaire consistante et robuste. Il est donc avantageux d'analyser la classification des pixels à plusieurs niveaux de résolution. Supposons qu'un pixel appartient à une classe bien déterminée avec une forte vraisemblance ; si son ancêtre dans la pyramide appartient à la classe correspondante par sous échantillonnage et avec une degré d'appartenance aussi important, la confiance que le pixel soit bien classé augmente dans la mesure où l'ancêtre résume par filtrage un point avec ses voisins. L'idée est donc d'influencer la classification d'un point dans chaque niveau de résolution avec les deux contraintes : spatiale et pyramidale. Notons de plus que les voisinages respectivement impliqués dans ces dernières peuvent être différents. La contrainte par multirésolution est donc introduite par le biais d'un troisième terme rajouté à la fonction objectif 3.8 qui se présente alors sous la forme :

$$\begin{aligned}
 J_{SPC}(U, V) &= \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m \|x_j - v_i\|^2 + \frac{n}{2} \alpha \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m e^{-\sum_{k \in \Omega} u_{ik}^m} \\
 &+ \beta \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m e^{-\tilde{u}_{ij}^m}
 \end{aligned} \tag{3.17}$$

où  $n$  est l'indice du niveau de résolution dans la pyramide.  $\tilde{u}_{ij}$  est la fonction d'appartenance à une classe  $i$  de l'ancêtre immédiat (niveau  $n-1$ ) d'un point  $j$  (niveau  $n$ ). La figure 3.5 illustre la nature de la contrainte pyramidale. Le paramètre  $\beta$  permet de contrôler l'influence du troisième terme. La construction de la pyramide engendre un lissage qui croît en montant d'un niveau à l'autre. Cela augmente l'effet de la régularisation dans les niveaux plus haut et peut mener à une sous segmentation. Intuitivement, la contrainte spatiale devra donc être moins importante dans les niveaux de faible résolution. Le paramètre  $\frac{n}{2}$  a pour but de limiter la régularisation aux niveaux plus hauts où le lissage devient trop fort.

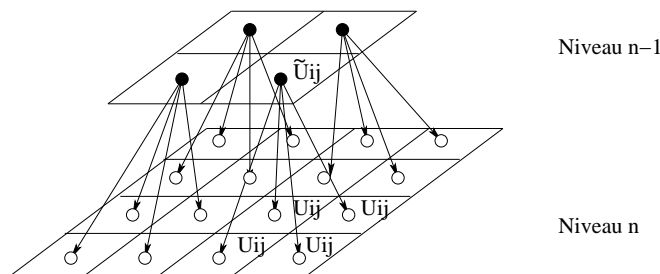


FIG. 3.5 – Contrainte en multirésolution par fonctions d'appartenance

A l'instar de la fonction objectif (3.8) l'optimisation de (3.17) s'opère par la méthode

du lagrangien.

$$\begin{aligned}
J_{SCM}(U, V) &= \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m \left( \|x_j - v_i\|^2 + \frac{n}{2} \alpha e^{-\sum_{k \in \Omega} u_{ik}^m} + \beta e^{-\tilde{u}_{ij}^m} \right) \\
&+ \sum_{j=1}^N \lambda_j \left( 1 - \sum_{i=1}^C u_{ij} \right)
\end{aligned} \tag{3.18}$$

L'équation de la matrice d'appartenance est résolue de la même manière que dans la section 3.6, on obtient donc :

$$u_{ij} = \frac{1}{\sum_{p=1}^C \left( \frac{\|x_j - v_i\|^2 + \frac{n}{2} \alpha e^{-\sum_{k \in \Omega} u_{ik}^m} + \beta e^{-\tilde{u}_{ij}^m}}{\|x_j - v_p\|^2 + \frac{n}{2} \alpha e^{-\sum_{k \in \Omega} u_{pk}^m} + \beta e^{-\tilde{u}_{pj}^m}} \right)^{\frac{1}{m-1}}} \tag{3.19}$$

La conjugaison des deux termes de contrainte dans  $u_{ij}$  unifie les contraintes spatiale et de multirésolution imposées à la fonction objectif. On constate que la valeur de la fonction d'appartenance est doublement conditionnée par influence *horizontale* et *verticale*. Le dosage de cette influence est contrôlable à partir des poids associés à chaque terme de contrainte. Dans le cas particulier où  $\alpha = \beta = 0$  on retrouve le cas du FCM standard. Pour  $\beta = 0$  il y a que la contrainte spatiale qui est prise en considération (SCFCM).

La contrainte pyramidale se propage de la racine à l'image, où on obtient la segmentation finale. Si le père appartient à une classe particulière avec une grande probabilité et de même pour ses fils cela renforce la confiance locale en cette classe. Sinon on pénalise les fils pour les rapprocher du comportement de leur père.

L'algorithme FCM avec contrainte spatio-pyramidale (SPFCM). est résumé dans la procédure (Algorithm 4)

**Algorithm 4:** Algorithme de SPCFCM

---

Step 1. Initialisation :

- Choisir le nombre de classes
- Choisir les poids  $\alpha$  et  $\beta$
- $u_{ij} \leftarrow$  valeur aléatoire  $\in [0, 1]$  pour tout niveau de résolution
- Forcer  $\beta = 0$  pour le premier niveau

**repeat**

- foreach** *niveau de résolution* **do**
  - Step 2. Calculer les centres en utilisant (3.7)
  - Step 3. Calculer la distance  $\|x_j - v_i\|^2$
  - Step 4. Mise à jour de la Matrice  $U$  par (3.19)
- end**

**until**  $\|U^k - U^{k+1}\| < \epsilon$

Step 5. Défuzzification dans le dernier niveau

---

Le premier niveau ne possède pas d'ancêtre ce qui fait que  $\tilde{u}_{ij}$  ne peut pas y être calculé et par conséquent ce terme est forcé à 0 via le poids  $\beta$ . Le critère de convergence est par ailleurs respecté pour toutes les couches de résolution. L'avantage de cet algorithme est qu'à chaque itération on a une information sur l'état de classification pour les nouvelles fréquences impliquées. Ceci permet d'augmenter progressivement les performances du clustering. Le choix de résolution dépend du taux de régularisation qu'on souhaite incorporer dans le processus. Il reste expérimental dans notre étude. S'agissant d'un paramètre d'ordre plus élevé, il n'a pas semblé intéressant de l'automatiser à ce stade, par exemple d'après une analyse grossière du spectre.

### 3.8 Détermination automatique du nombre de classes

Les algorithmes de type partitionnement tels que FCM et ses variantes nécessitent une connaissance préalable du nombre de classes. Le choix de ce nombre est d'autant plus délicat que son impact est direct sur le résultat final. D'où l'intérêt de sa détermination automatique. La recherche du nombre optimal de groupes est appelé mesure de validité. Il existe plusieurs travaux qui ont abordé ce problème [FS89b, GG89, XB91, PB95, ZLE99, HBV01]. Ces travaux sont généralement basés sur des méthodes itératives qui mesurent la densité et la séparation des groupes. Afin d'obtenir une mesure de validité efficace pour notre algorithme de SPCFCM nous proposons de même une méthode adaptée mesurant la densité et la séparation des groupes pour chaque niveau de résolution. L'indice de validité se présente sous la forme suivante :

$$S_k(c) = \frac{\sum_{i=1}^C \sum_{j=1}^N u_{ij}^2(k) \|x_j(k) - v_i(k)\|^2 / N}{\sum_{p=1}^C \|v_p(k) - \bar{v}(k)\|^2 / C} + e^{-\min_{i,j} \|v_i(k) - v_j(k)\|^2} \quad (3.20)$$

on peut écrire :

$$S_k(c) = \frac{D_k}{sep1_k} + sep2_k \quad (3.21)$$


---

où  $c$  est une variable entière qui correspond au nombre de groupes,  $\bar{v}$  correspond à la moyenne des centres de clusters.  $k$  est l'indice de la couche dans la représentation pyramidale.  $D_k$  est basé sur la fonction d'appartenance et les distances, indique la densité des données au sein d'un groupe au niveau de résolution  $k$ . Le terme  $sep1_k$  intervient quant à lui dans la mesure de séparation. Il est basé sur la variance des centres normalisés par leurs somme. Une grande valeur de celui ci indique que les centres ont une dispersion suffisante pour que les groupes soient distants.  $sep2_k$  obtient l'information concernant la séparation inter groupe et cela par le calcul de la distance minimale entre les centres. Une petite valeur de cette distance signifie que les centres sont proches et les groupes ne sont pas bien séparés.

Une valeur optimale  $c^*$  du nombre de groupe est inversement proportionnelle à  $sep1_k$  et proportionnelle à  $sep2_k$  donc décroissante en  $\|v_i(k) - v_j(k)\|^2$ . La valeur minimale de  $S_k(c)$  est obtenue pour  $c^*$  : cela traduit une division de l'espace des points en groupes qui ont une tendance à être compacts à l'intérieur et séparés par rapport aux autres groupes.

$$\min_{2 \leq c \leq N-1} \left\{ \min_k S_k(c) \right\} \quad (3.22)$$

La mesure de validité est obtenue par une minimisation globale de  $S_k(c)$  impliquant tous les niveaux de résolution. En d'autre termes la valeur de  $c$  qui sera retenue est celle qui est minimale tous niveaux confondus. L'idée c'est qu'un objet peut être mieux identifié à une résolution qu'à une autre. Grâce à la multirésolution nous retraçons ainsi une configuration plus complète de l'état des groupes en examinant leur densité et leur séparation à diverses fréquences. Cela nous a permis d'évaluer d'une manière plus sûre le nombre optimal de classes que peuvent contenir les données brutes.

L'algorithme de mesure de validité est de type itératif : il consiste à accroître la valeur  $c$  jusqu'à un maximum ( $C_{max}$ ). Théoriquement  $N-1$  est la valeur d'arrêt de  $c$ . La complexité serait trop grande, mais en pratique  $C_{max} \ll N-1$  [XB91]. A la fin des itérations le min de  $S_k(c) \forall k$  représente  $c^*$  (nombre optimal). La procédure est illustrée par l'algorithme 5

---

**Algorithm 5:** Algorithme de mesure de validité

---

```

Step 1.  $c \leftarrow 2, S_k^*(c) \leftarrow \infty, c^* \leftarrow 1$ 
repeat
  Step 2. Utiliser l'algorithme 4 (SPCFM)
  pour calculer  $v_i(k)$  et  $u_{ij}(k)$ 
  Step 3. Calculer la fonction  $S_k(c)$  en utilisant (3.21)
  if  $S_k(c) < S_k^*(c) \forall k = 1, \dots, n$  then
    |  $S_k^*(c) \leftarrow S_k(c), c^* \leftarrow c$ 
  end
  Step 4.  $c \leftarrow c + 1$ 
until  $c = C_{max}$ 

```

---

$v_i(k)$  et  $u_{ij}(k)$  représentent respectivement le centre d'une classe et la fonction d'appartenance au niveau de résolution  $k$

## 3.9 Résultats et discussion

Les expériences que nous avons effectuées se divisent en deux parties. La première consiste à comparer les algorithmes de regroupement proposés avec les méthodes standard. Dans ce cas le nombre de classes est choisi manuellement et nous détaillons rubrique après rubrique comment les classes sont choisies. La deuxième partie est consacrée à tester les performances de l'algorithme avec détermination automatique du nombre de classes.

### 3.9.1 Critères d'évaluation

La mesure de qualité de la segmentation d'images du monde réel ne peut être que subjective et souvent experte. On a vu qu'il n'existe pas de banc test commun ni a fortiori une méthode commune. Pour évaluer nos algorithmes qualitativement il nous a semblé pertinent de respecter les critères suivants :

1. Choix des images : la selection doit être aléatoire avec un échantillon d'environ 10% du nombre d'images dans la base.
2. Robustesse au changement de contexte : les algorithmes doivent avoir un comportement consistant indépendant du type d'image ou d'application.
3. Mesure de performance : l'évaluation du résultat doit inclure une comparaison des segmentations de l'homme et de la machine.

Les tests ont été menés sur trois types d'image différents. La première série est synthétique avec cinq classes parfaitement identifiées. On rajoute trois types de bruit de distributions différentes : Gaussien, grenaille (shot noise) et uniforme. Nous en donnons un exemple figure 3.11. Le choix d'une image de synthèse permet de contrôler le bruitage artificiel. La deuxième concerne notre application dans la recherche d'image par le contenu. Elle contient dix images aériennes et satellitaires ; le but ici est de tester l'influence des paramètres de pondération sur le résultat de la segmentation afin de mieux choisir les valeurs des poids lors de la segmentation effective avant traitement de la base des données. Enfin le troisième type d'images comporte également dix photographies naturelles de différents paysages extraites de la base de Berkeley<sup>1</sup> la figure 3.13 est un exemple de telles images. Cette base est considérée comme un Benchmark dont on dispose d'une vérité terrain. Martin et al. [MFTM01] font segmenter chaque image par plusieurs personnes puis un critère de rappel/précision est utilisé pour déterminer les régions les plus

---

<sup>1</sup><http://www.cs.berkeley.edu/projects/vision/grouping/segbench/>

---

pertinentes. L'intérêt ici est de comparer les méthodes à une segmentation de référence appliquée à des photographies dans différents contextes.

### Indice de qualité de la segmentation

Toute région retrouvée correspondant à l'image de référence est considérée comme pertinente. Notre mesure de qualité adresse les trois points suivant :

1. La localisation : les régions détectées doivent être en cohérence spatiale (e.g. position, taille, forme etc.) avec les régions de références.
2. La sur segmentation : elle, est considérée comme un bruit et pénalisée dans l'indice de qualité
3. La sous segmentation : elle exprime un trop fort lissage et elle est également pénalisée

Soit  $A_i$  et  $B_i$  deux régions respectivement référence et résultat. L'indice de qualité  $Q$  est ainsi donné par :

$$Q = \sum_{i,j/A_i \cap B_j \max} \frac{A_i \cap B_j}{A_i \cup B_j} \eta \quad (3.23)$$

La mesure de qualité implique les segments "résultat" en recouvrement localement maximal avec les segments "référence" dans le plan de l'image. Le rapport intersection/union est un indice morphologique exprimant la cohérence spatiale entre deux régions. En effet, en cas de superposition parfaite le rapport vaut 1, en cas de correspondance partielle le rapport est toujours inférieur à 1 et à surfaces égales l'inclusion stricte (intersection totale) est moins pénalisée que l'intersection partielle. Le facteur  $\eta$  mesure la sur/sous segmentation : il est exprimé par :

$$\eta = \begin{cases} N_A/N_B & \text{si } N_B \geq N_A \\ \frac{1}{\log(N_A/N_B)} & \text{autrement} \end{cases} \quad (3.24)$$

où  $N_A$  et  $N_B$  sont les nombres de régions respectifs dans l'image référence et l'image résultat. Le logarithme sert à pénaliser moins la sous que la sur segmentation, à la fois parcequ'elle est plus rare et surtout parcequ'elle porte logiquement sur des sous ensembles de pixels plus larges.

---

### 3.9.2 Evaluation et comparaison

Dans cette section nous étudions les performances des méthodes comparant les algorithmes de C-Moyennes, FCM, SCFCM et SPCFCM. Tous les algorithmes de type FCM sont expérimentés dans les mêmes conditions : le "Fuzzifier"  $m = 2$  et  $\epsilon = 0.0001$ . La pondération des caractéristiques du voisinage est  $w = 0.3$  (choisie heuristiquement). Notre méthode repose sur les contraintes spatiale et pyramidale. Pour étudier l'influence de ces dernières on en varie les poids  $\alpha$  et  $\beta$ . Sachant que  $\alpha = \beta = 0$  annule les contraintes et on se retrouve dans le cas de FCM, et que pour  $\beta = 0$  c'est le cas SCFCM. La mesure de performance est déterminée par le calcul de l'indice de qualité  $Q$ .

**Remarque 3** *Trois niveaux de résolution sont impliqués pour l'algorithme SPCFCM. Ces niveaux sont obtenus par l'équation 3.16 en choisissant un filtre gaussien de taille  $5 \times 5$ . Nous considérons que trois couches de résolution sont suffisantes pour exercer la contrainte pyramidale.*

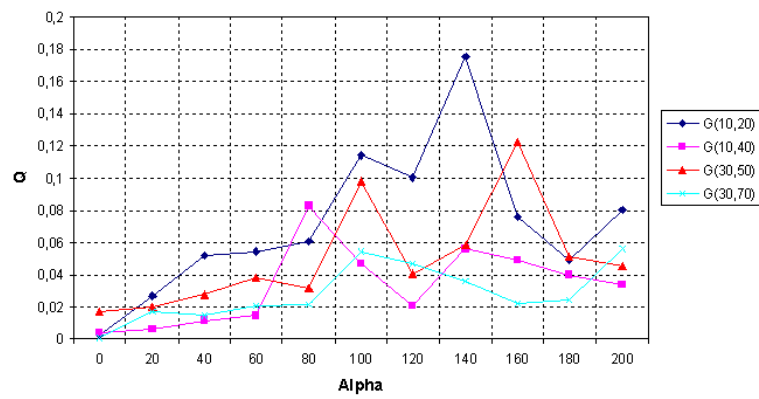
#### Test 1

Le but de ces expériences est d'étudier l'influence du bruit sur la détection des régions. Pour cela nous avons testé 4 différentes valeurs pour chaque bruit : le choix de ces valeurs est en fonction du rapport signal/bruit. On commence par une valeur faible puis on augmente la proportion du bruit. Ceci permet de tester les algorithmes selon la variation du bruit. Le bruit Gaussien est défini par le couple moyenne, variance  $(\mu, \sigma)$  dont les valeurs sont successivement : (10,20), (10,40), (30,50) et (30,70). Le bruit de grenaille est exprimé en taux de corruption on a : 5%, 8%, 10% et 12%. Le dernier bruit est de type uniforme caractérisé par un minimum et un maximum, (0,10), (0,20), (0,40) et (0,50). Le nombre de classes est initialisé à 5 pour tous ces tests car l'image comporte 5 régions et la classe bruit est ignorée. On obtient donc 12 images synthétiques bruitées. La figure 3.6 (a) représente la mesure de qualité de segmentation en fonction de  $\alpha$ . Nous remarquons que la contrainte spatiale améliore le résultat d'une manière croissante jusqu'à une valeur limite où les performances commencent à se dégrader. En fixant  $\alpha = 0$  et variant  $\beta$  on constate sur la figure 3.6 (b) que la contrainte pyramidale est moins efficace. Cependant la combinaison entre les deux contraintes donne des résultats meilleurs que ceux des contraintes séparées comme illustré sur la figure 3.6 (c). Il est à noter que  $\alpha$  et  $\beta$  varient avec les mêmes valeurs. Chaque courbe de la figure 3.6 (d) est obtenue à partir de la moyenne du bruit sur les 4 valeurs.

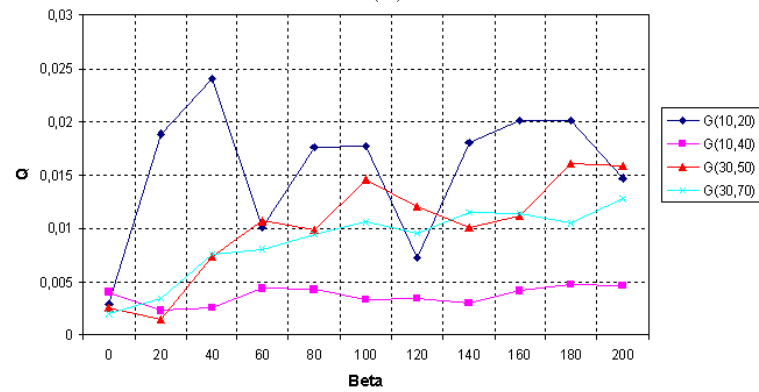
La figure 3.7 concerne les résultats du bruit de grenaille on retrouve pratiquement le même comportement de FCM, SCFCM et SPCFCM pour ce type de bruit. Ce dernier a un impact plus important sur la qualité de segmentation que le bruit gaussien.

En ce qui concerne le bruit uniforme (figure 3.8), on remarque que la contrainte spatiale

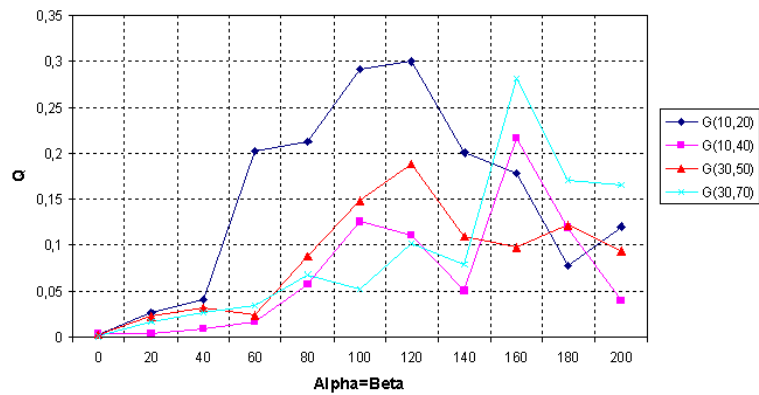




(a)



(b)



(c)

FIG. 3.6 – Mesure de la qualité de segmentation pour 4 valeur de bruit gaussien(a) contrainte spatiale (b) contrainte pyramidale (c) les deux contraintes avec les même valeurs

augmente l'indice de qualité, en revanche la contrainte pyramidale le diminue.

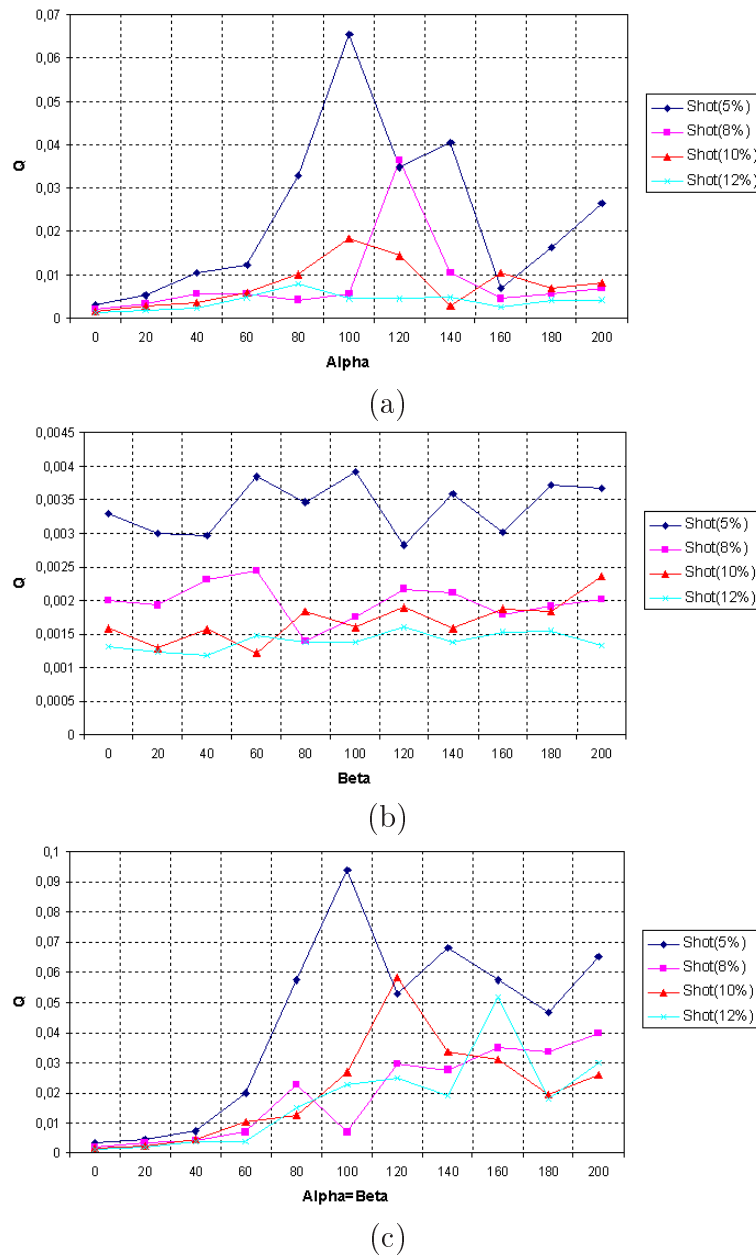
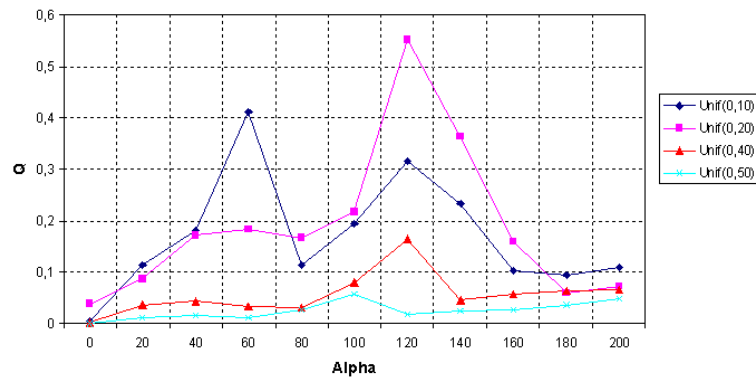
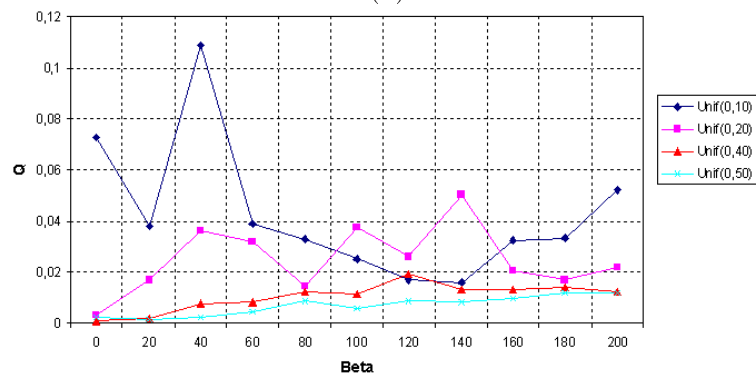


FIG. 3.7 – Mesure de qualité de segmentation pour un bruit de grenaille (a) contrainte spatiale (b) contrainte pyramidale (c) les deux contraintes avec les mêmes valeurs

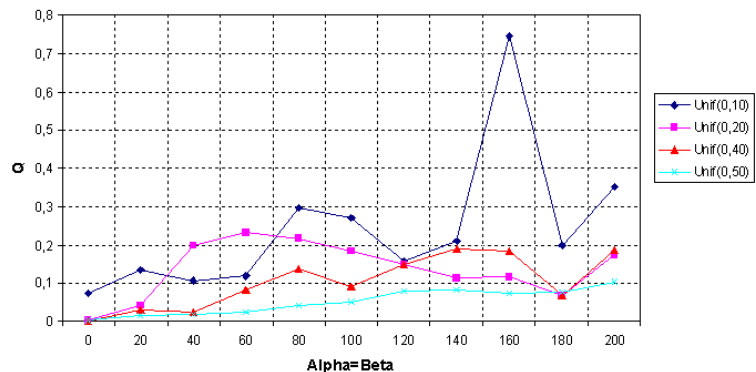
Sur ces trois expériences on constate l'apport des algorithmes SCFCM et SPFCM. Une forte régularisation peut conduire à une dégradation de la segmentation. S'il est difficile de déterminer une valeur optimale des poids, on peut définir une plage de valeurs : par exemple pour ces exemples les valeurs de l'intervalle  $[40, 100]$  améliorent systématiquement le résultat. Ces expériences indiquent que la régularisation spatiale ainsi que la contrainte pyramidale sont efficaces pour rendre la classification mieux résistante au bruit. D'une manière générale l'indice de qualité est fonction décroissante du rapport signal/bruit.



(a)



(b)



(c)

FIG. 3.8 – Qualité de segmentation pour un bruit uniforme (a) contrainte spatiale (b) contrainte pyramidale (c) les deux contraintes avec les même valeurs

### Exemple 1

Ici on présente un exemple du type d'image utilisé pour ces expériences. Les images du haut de la figure 3.11 représentent les échantillons à segmenter. De gauche à droite la première image est l'original, la deuxième contient un bruit gaussien de moyenne 10 et variance 30, la troisième contient 10 % de bruit et dans la quatrième un bruit de distri-

bution uniforme ( $\min=0$  ,  $\max=50$ ) a été rajouté. Le résultat de segmentation de chaque image est illustré dans les images de la même colonne en dessous. On remarque que toutes les méthodes donnent le même résultat pour l'image sans bruit et les classes sont parfaitement détectées. Par ailleurs dans le cas où l'image comporte du bruit les résultats sont contrastés. On constate que K-moyenne et FCM produisent une segmentation erronée, le bruit a affecté significativement la classification. En revanche SFCM et SPCFCM ont réussi à éliminer une grande partie du bruit, on peut remarquer que SPCFCM produit la meilleure classification.

## Test 2

Les expériences dans cette section portent sur 10 images aériennes et satellitaires choisies aléatoirement de notre base de donnée (voir chapitre 4). Nous avons segmenté manuellement ces images selon l'homogénéité des régions en texture ou en couleur. La figure 3.9 montre les courbes des trois contraintes. On constate que la contrainte spatiale est légèrement meilleure que la pyramidale. L'utilisation des deux contraintes ensemble augmente l'indice de qualité d'une manière linéaire, en tout cas jusqu'à la valeur 200. Comme pour les images synthétiques l'influence de ces contraintes dégrade le résultat après une valeur limite. Expérimentalement nous avons trouvé que les valeurs entre (40,100) réalisent un bon compromis pour ces images et choisissons les poids dans cette plage.

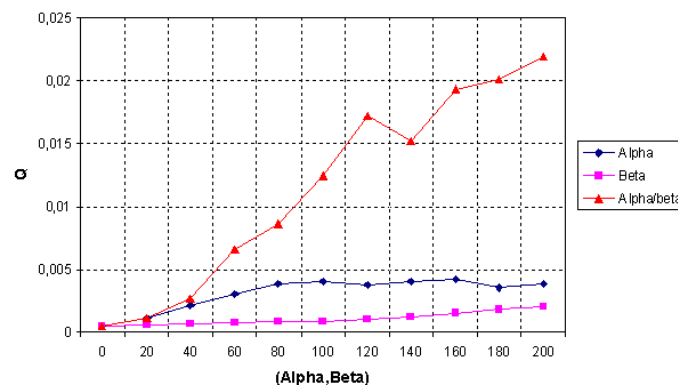


FIG. 3.9 – Comparaison entre 3 contraintes : spatiale, pyramidale et spatio-pyramidale pour des images satellitaires

## Exemple 2

Dans cet exemple nous traitons des images aériennes de différentes natures. Nous avons vu dans la section précédente l'impact du bruit artificiel sur la segmentation. Cependant dans le monde réel les images contiennent un bruit plus complexe et de type différent. Souvent il est difficile de l'associer à un modèle prédéfini. Nos expériences sont motivées par ce fait et les images de télédétection sont un bon exemple de telle situation.

Le bruit présent dans ces images est généralement dû aux capteurs, systèmes électroniques, conditions atmosphériques, etc. La première ligne de la figure 3.12 présente des images prises avec différents systèmes. De gauche à droite la première et la deuxième image proviennent d'un radar à ouverture synthétique (SAR bande C/X) du CCRS<sup>2</sup>. La première image est un champ forestier et la seconde représente des terrains agricoles, chaque pixel est codé sur 3 bandes spectrales. La troisième image est une image satellite de USGS<sup>3</sup> prise par Landsat 7 avec une échelle de 2,9 km, elle représente une région du désert de Namibie. La quatrième image qui représente des champs cultivés de la région Kozloduy en Bulgarie est produite par le satellite multi-spectral FORMASAT-2<sup>4</sup> avec une résolution de 8 mètres. Les résultats de segmentation qui sont au dessous de chaque image montrent que les algorithmes classiques de C-Moyenne et FCM produisent beaucoup de petites régions insignifiantes. Les méthodes de FCM avec contraintes spatiales et pyramidales favorisent comme attendu la cohérence spatiale au sein d'une région ce qui se traduit par un découpage plus homogène. On remarque aussi que malgré la régularisation les bords sont préservés

### Test 3

Les expériences sont menées à présent sur 10 images naturelles de la base Berkley. Ces images de type naturel et généraliste comportent différentes scènes et paysages ; la couleur et la texture y sont pratiquement omni présentes. Le nombre de classes varie d'une image à une autre. Il est donc choisi en fonction des images de références. La courbe de la figure 3.10 montre la variation de l'indice de qualité selon la pondération de la contrainte de régularisation employée. On remarque que  $\alpha$  et  $\beta$  ont la même tendance pour ces images cependant Il est bien évident que l'association de la contrainte spatiale et pyramidale améliore le résultat de segmentation.

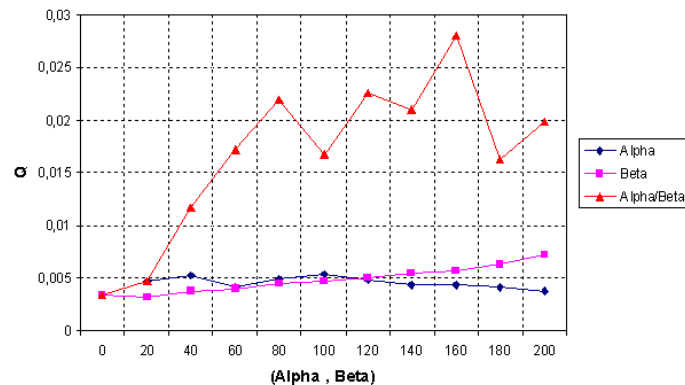


FIG. 3.10 – Comparaison entre 3 contraintes : spatiale, pyramidale et spatio-pyramidale pour des images de la base Berkley

<sup>2</sup>Canadian Center for Remote Sensing (<http://www.ccrs.nrcan.gc.ca>)

<sup>3</sup>US Geological Survey (<http://edc.usgs.gov>)

<sup>4</sup>Spot Image (<http://www.spotimage.fr>)

### Exemple 3

Dans ces expériences nous appliquons les algorithmes sur les images de Berkeley. La figure 3.13 illustre des exemples de segmentation de ces dernières. On constate que SCFCM et SPCFCM produisent des régions cohérentes et significatives ce qui est important pour un système CBIR. La raison, en est probablement que ces deux méthodes supportent la texture sans l'introduire explicitement dans le processus de segmentation. Ce n'est pas le cas pour C-Moyennes et FCM.

### 3.9.3 Mesure de validité

Comme nous l'avons décrit auparavant, la détection automatique du nombre  $c^*$  dépend de la nature des données. Pour cette raison nous essayons de tester l'algorithme avec différents types d'images où les données n'ont pas toujours la même structure de regroupement. Pour ces expériences on pose  $C_{max} = 15$ . Dans cette mesure nous cherchons le nombre de classes produisant la valeur minimale de  $S_k$ .

### Exemple 1

Cet exemple vise à mesurer la robustesse de la détection dans un milieu bruité. Pour cette fin nous utilisons des images synthétiques dont on connaît parfaitement le nombre de classes ainsi que la quantité de bruit qu'elles contiennent. On reprend les images de test de la section 3.9.2 exemple 1 où le nombre de groupes vaut 5. Dans la figure 3.14 au dessous de chacune des images originales on trouve des courbes de mesure de validité avec  $S_k$  en fonction du nombre de classes  $c$ . La première mesure en partant de la gauche sur la figure 3.14 affiche un minimum sur  $c^* = 6$  ce qui est proche du nombre exact des classes. La deuxième mesure est appliquée sur la même image bruitée (gaussienne  $\mu = 10$ ,  $\sigma = 30$ ) elle affiche également le même nombre de clusters. L'image d'après (bruit de grenaille) engendre deux minimums à 6 et 11 classes, ceci reflète la difficulté rencontrée par l'algorithme pour segmenter cette image. Pour la dernière courbe où les valeurs minimales sont 3 et 5. On peut constater que les valeurs obtenues par la mesure de validité sont généralement proches de la valeur optimale. Ceci permet de segmenter l'image d'une manière entièrement automatique indépendamment du bruit ou d'autres artefacts.

### Exemple 2

Dans cette partie la détection automatique du nombre de classes est appliquée à des images contenant des scènes naturelles. Il est difficile d'avoir un préjugé sur le nombre exact de classes que comporte ce genre d'image parce qu'il est subjectif. On considère

---

que la pertinence du  $c^*$  est liée au résultat de la segmentation. La cohérence des régions atteste si le nombre été bien choisi. La figure 3.15 illustre des exemples de segmentation par SPCFCM après la détection de  $c^*$ . Sur la figure 3.15 de gauche la première courbe affiche un nombre optimal de 6 groupes, la suivante  $c^*$  vaut également 6, celle d'après détecte 4 classes et dans la dernière le nombre optimal est 5. Les résultats de segmentation en dessous des courbes indiquent une détection correcte des régions avec une possible sur segmentation de ces dernières.

### 3.9.4 Conclusion

Dans ce chapitre, nous avons proposé deux méthodes de segmentation qui reposent sur le regroupement flou ainsi qu'une technique de détection automatique du nombre de classes. La première méthode exploite une contrainte spatiale et la deuxième utilise deux contraintes, spatiale et pyramidale. Cette dernière est particulièrement intéressante car elle est associée à une représentation robuste qui conduit à examiner l'évolution de l'état du processus à différents niveaux de résolution. Les tests effectués supportent l'idée du départ et confirment l'amélioration apportée à la segmentation par rapport aux méthodes standard. La contrainte spatio-pyramidale affiche de bonnes performances et un comportement plus stable et plus robuste comparé à une régularisation spatiale seule. Les contraintes sont contrôlées par des poids qui déterminent l'équilibre de la régularisation qu'on veut obtenir. Notons que ce type de contraintes peut s'appliquer à d'autres algorithmes. En tenant compte de l'application que nous visons, la détection automatique du nombre de classes est nécessaire. L'indice de validité que nous proposons répond à ce besoin et permet d'obtenir un nombre adéquat. D'autres exemples de segmentation de base d'images seront donnés dans les chapitres suivants.

---

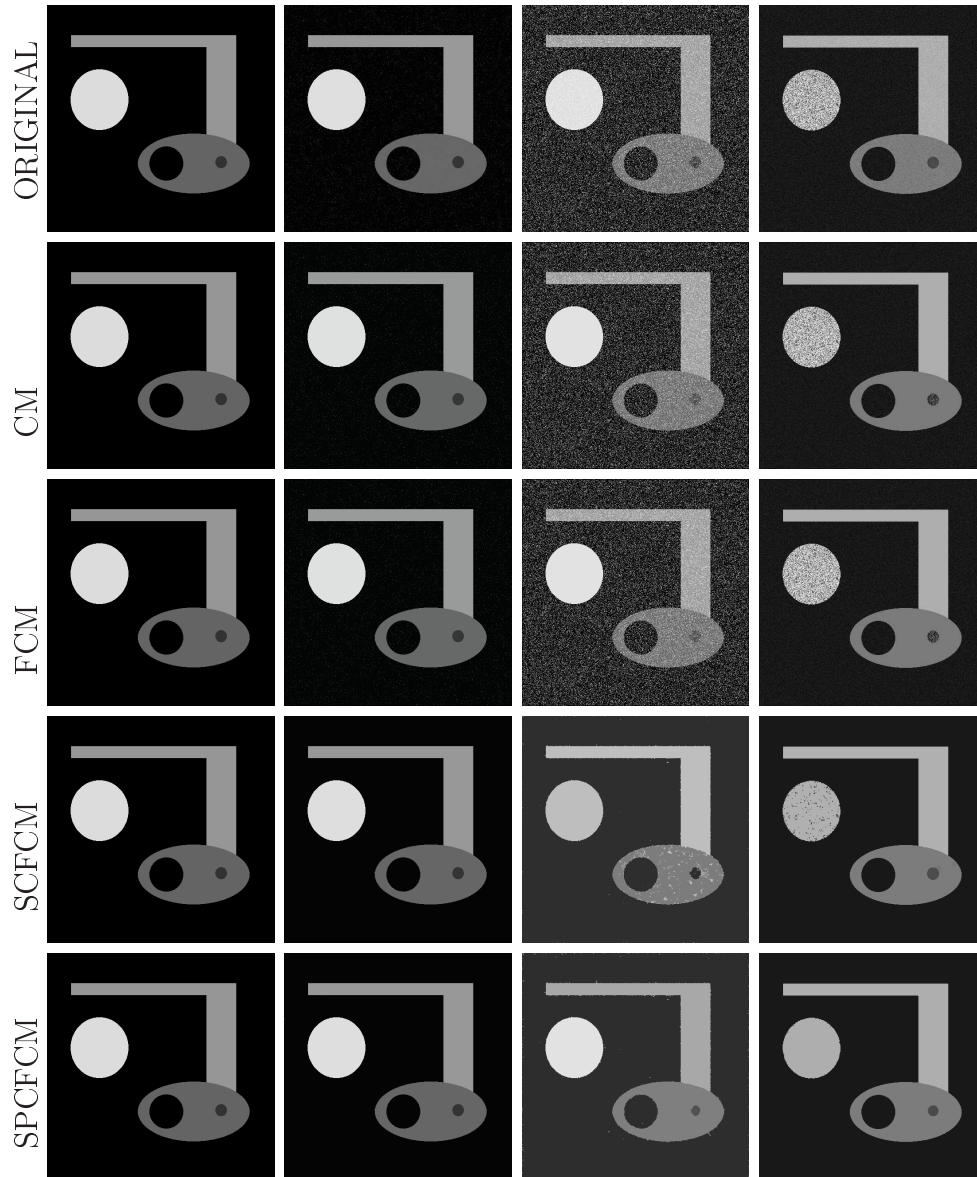


FIG. 3.11 – Résultat de segmentation d'images synthétiques



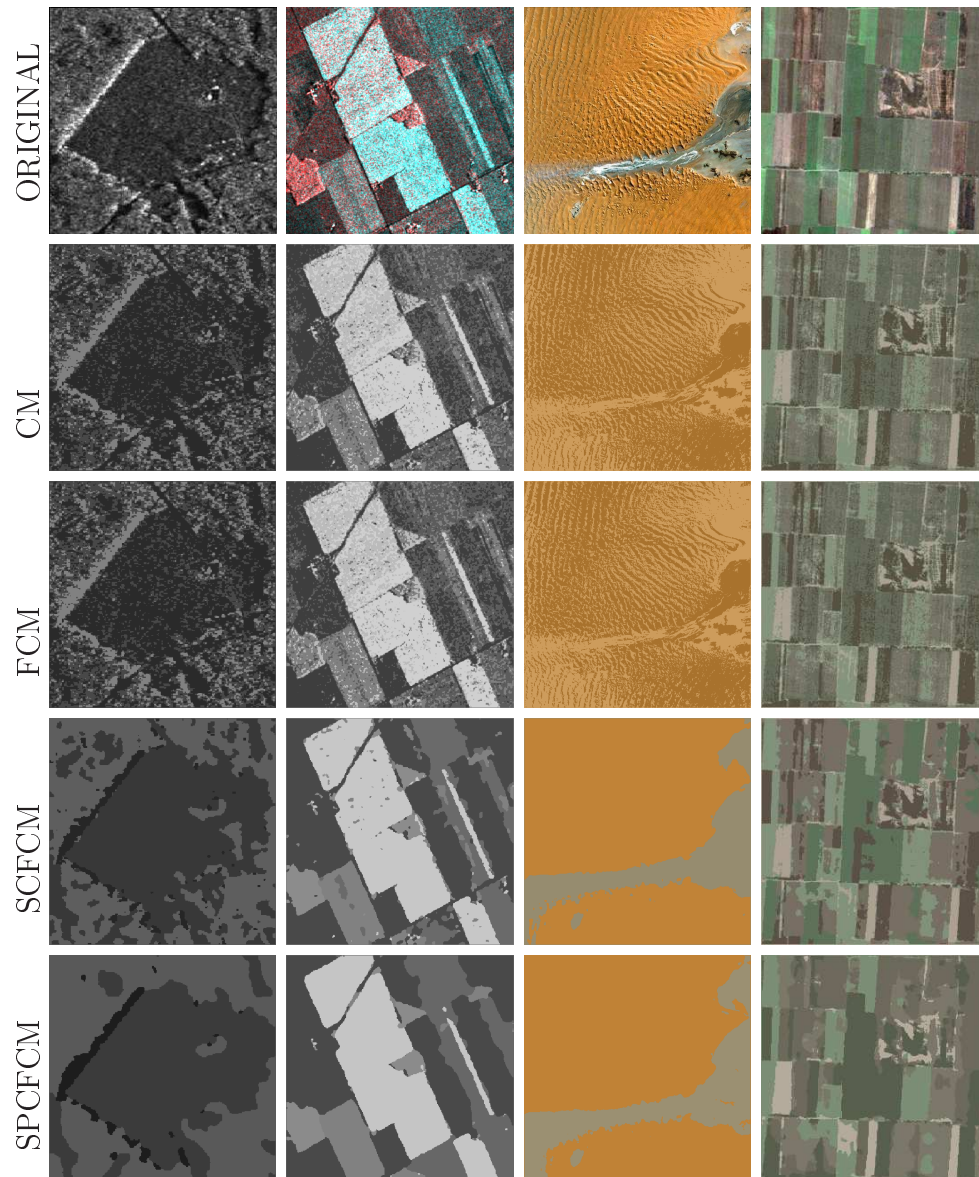


FIG. 3.12 – Segmentation d'images satellitaires et de télédétection

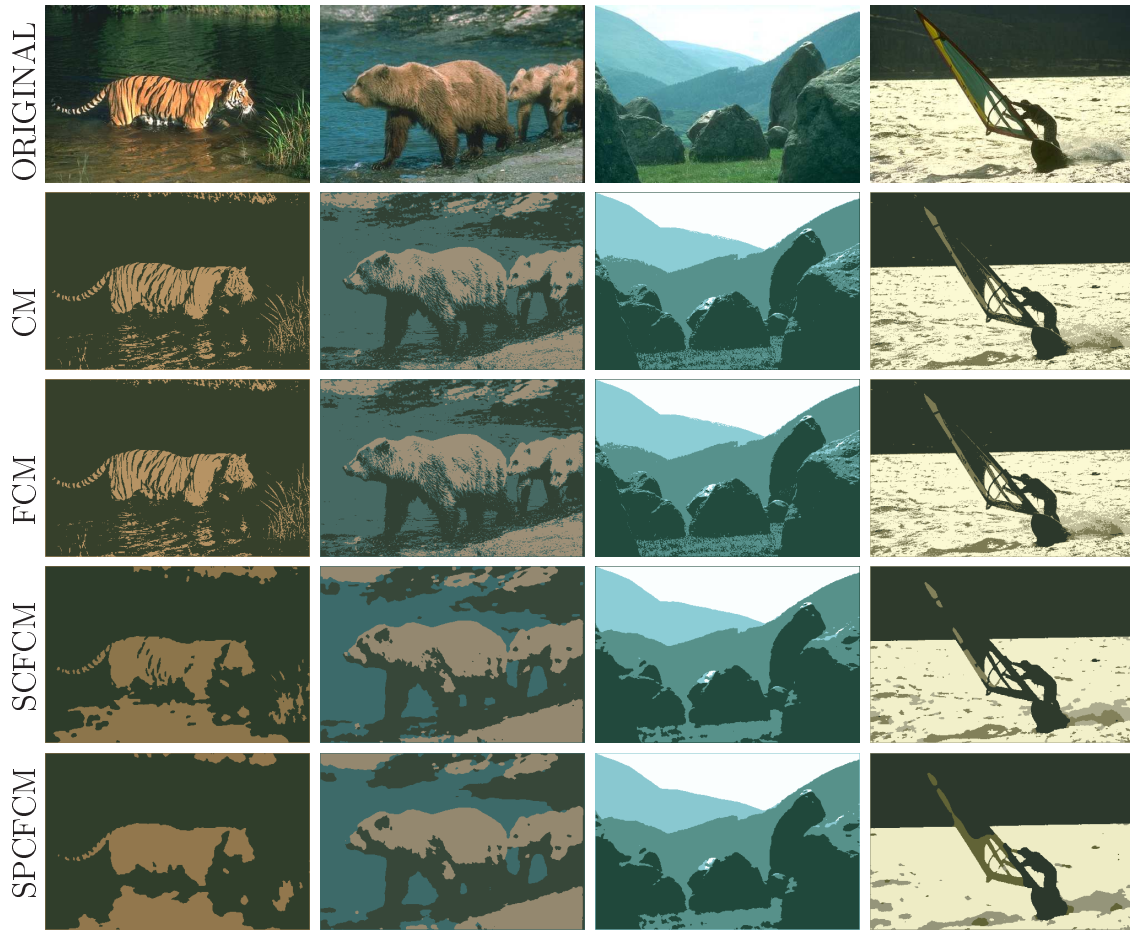


FIG. 3.13 – Résultat de segmentation pour des images naturelles

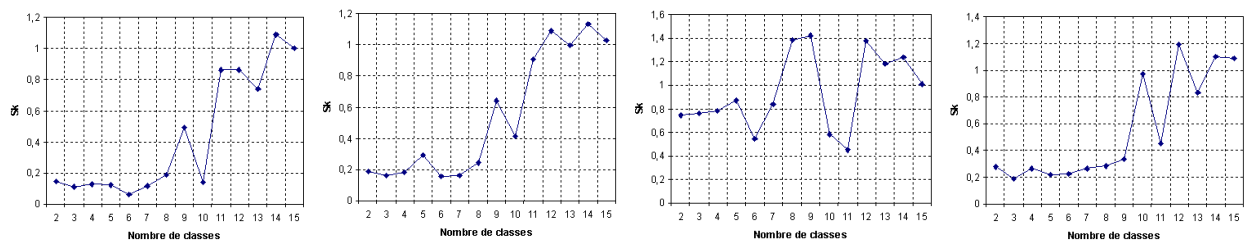
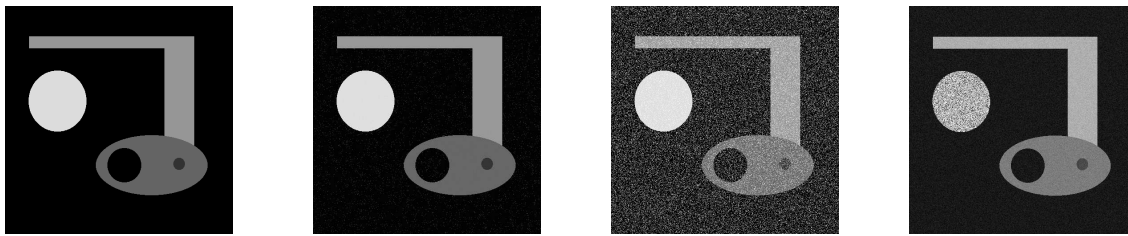


FIG. 3.14 – Mesure de validité pour images synthétiques

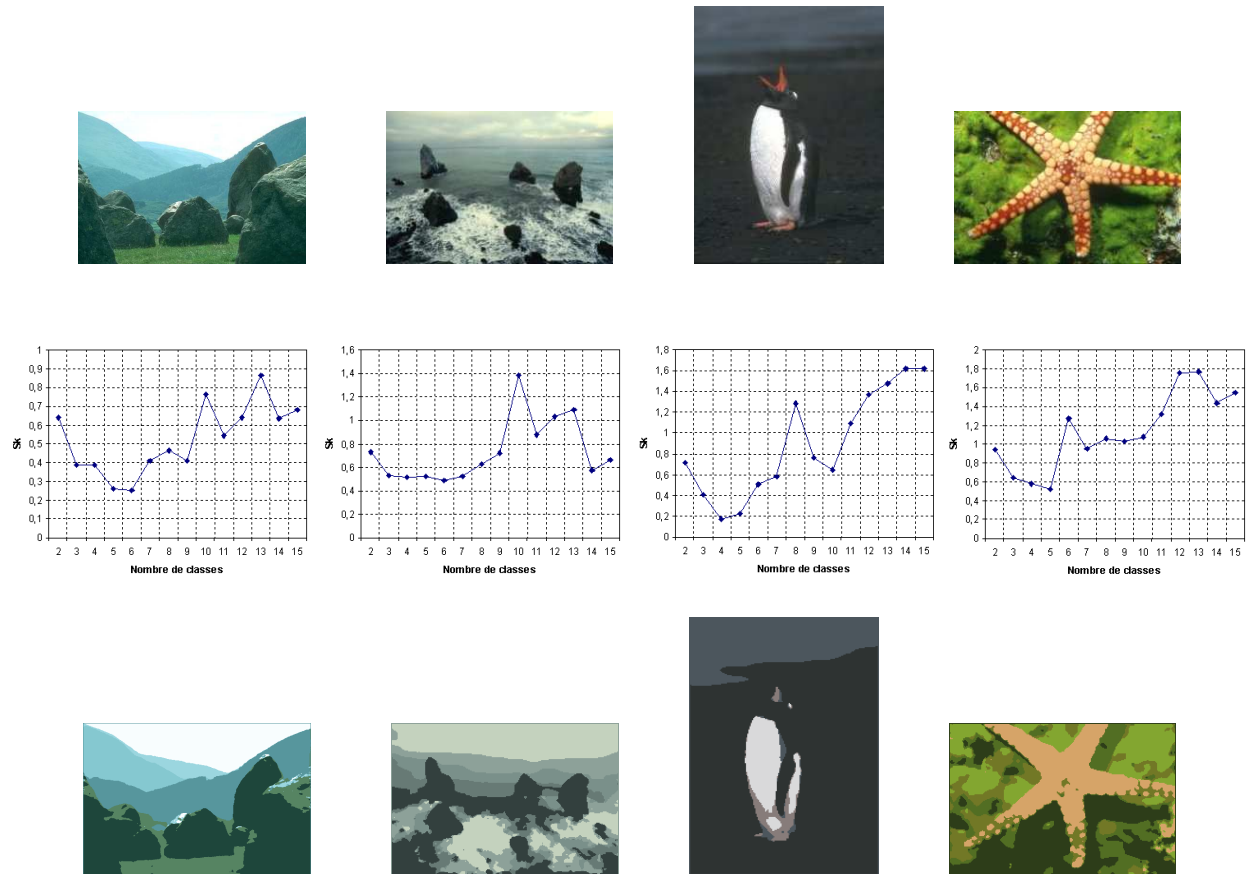


FIG. 3.15 – Mesure de validité pour images naturelles

# Chapitre 4

## Recherche d'image par composantes visuelles

### 4.1 Introduction

Dans ce chapitre nous nous focalisons sur la recherche d'image en base de données. Notre méthode est locale basée sur des régions munies d'une signature attribuée dans chaque image. Des descripteurs de type visuel et spatial servent à caractériser les régions à cet effet. La mesure de similarité entre deux images est alors ramenée à un problème d'appariement entre leurs régions. Nous proposons une nouvelle méthode de mise en correspondance fondée sur un double appariement et permettant à un élément d'être associé avec une ou plusieurs entités de la deuxième image. La similarité entre deux images est jugée en fonction de la distance entre descripteurs et le nombre d'entités visuelles appariées. La validité de notre méthode est testée dans deux applications différentes. La première relève du domaine des systèmes géographiques utilisant des images aériennes et satellitaires. La deuxième est destinée à la conduite routière où les images proviennent d'une caméra placée sur un véhicule intelligent. Les résultats de la recherche et les performances supportent la comparaison de différentes méthodes.

### 4.2 Description du contenu

Le contenu d'une image décrit par un aspect visuel i.e. des entités munies d'attributs de forme, d'intensité ou de fréquence supporte une certaine sémantique. Le contenu visuel implique la couleur, la texture ou la forme de zones définies mais encore des caractéristiques qui peuvent être plus élaborées telles que points d'intérêts, géométrie ou mesure des contours,...etc. La sémantique image, quant à elle, va beaucoup plus loin que ces simples

---

attributs visuels de régions et par exemple comporte aussi des relations entre entités. Mais il est difficile de la définir exactement et notamment de produire des règles pour son expression. Plusieurs méthodes automatiques sont proposées revendiquant une introduction de la sémantique ou son exploitation accrue dans la reconnaissance d'image via des relations d'interférence entre entités visuelles au sein de l'image. Les méthodes manuelles se contentent de lui associer du texte exploitable lors d'une requête. En l'absence d'une caractérisation rigoureuse de la sémantique image on parlera d'une sémantique limitée, codée via des attributs associés aux entités extraites de l'image dont leurs interrelations. Ces attributs réfèrent de manière plus ou moins exacte et directe aux intentions ou à l'action portées par la requête. Le contenu visuel sera donc pour nous la caractéristique prépondérante dans la mesure de ressemblance entre images car il offre un pouvoir discriminant élevé. Les attributs de sémantique limitée sont un apport supplémentaire pour filtrer les indésirables et augmenter la précision.

Les performances d'un système CBIR dépendent évidemment de la nature des descripteurs et attributs visuels ainsi que de leur méthode de calcul qui peut être de type global ou local. L'approche globale consiste à extraire des caractéristiques représentant l'image entière par traitement impliquant tous les pixels de l'image : l'histogramme ou corrélogramme de l'image est par exemple fréquemment utilisé pour cette fin. Une description globale s'accommode mal de requêtes partielles où l'on fait se correspondre des parties d'images. L'approche locale au contraire permet une description partielle offrant ainsi une représentation plus fine et plus naturelle. L'approche locale opère sur des partitions de l'image. La manière la plus simple pour obtenir ces partitions est la division de l'image en blocs égaux dont on extrait des descripteurs ; on peut calculer ces descripteurs par bloc simplement avec les mêmes procédures que celles mises en oeuvre globalement (e.g. histogramme). Dans ce type de partitionnement les régions ne sont pas significatives. Il peut donc s'avérer plus judicieux d'utiliser un partitionnement en régions homogènes, qui relève de la segmentation.

### 4.3 Extraction de descripteurs visuels locaux

La première étape de notre procédure de recherche consiste à décomposer l'image en entités visuelles *via* une segmentation en régions (voir chapitre 3). Les différentes parties de l'image issues de cette décomposition sont chacune représentées par des descripteurs visuels. La figure 4.1 montre un exemple de décomposition de l'image en plusieurs régions après segmentation par l'algorithme SPCFCM. On remarque que chaque entité présente une certaine homogénéité visuelle correcte du point de vue de l'interprétation géographique élémentaire, sémantique qu'il faut ensuite traduire en paramètres.

Les applications que nous visons, images satellite ou paysages automobile, concernent des images où la texture est une caractéristique dominante. Pour représenter cette dernière nous utilisons des descripteurs de type MCM ou FGVPO introduits dans le chapitre 2.

---

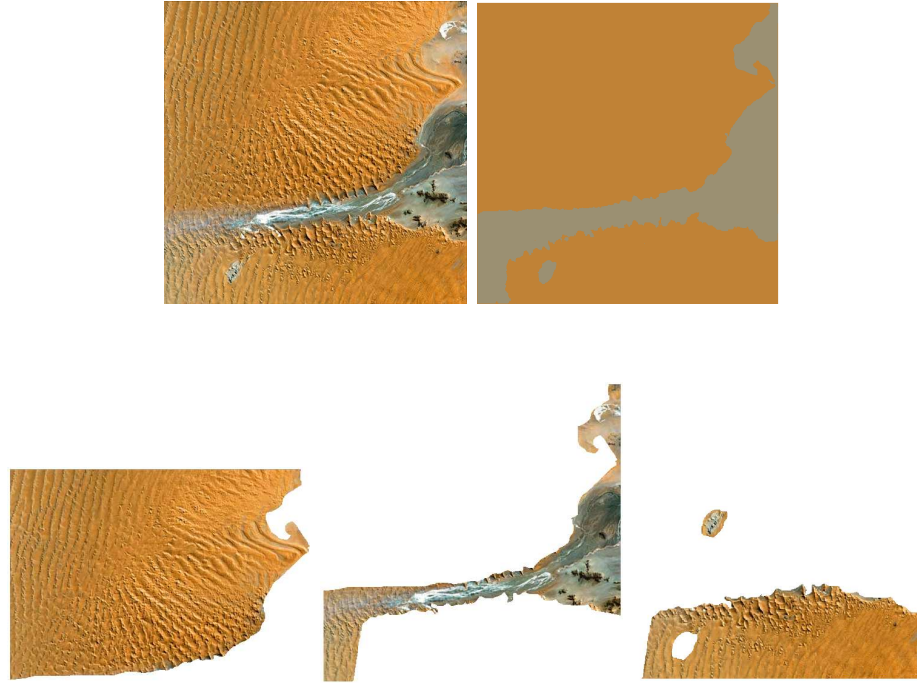


FIG. 4.1 – Décomposition d’une image en entités visuelles

Notre choix est motivé par leur puissance de discrimination des textures dans l’absolu. Dans la partie expérimentale nous comparons à nouveau les performances de ces caractéristiques avec les méthodes classiques d’analyse de texture, mais dans le cadre du CBIR.

**Définition 4.3.1** *Soit une image  $I$  décomposée en  $M$  régions, chaque région  $R$  caractérisée par son vecteur d’indices  $F$ , l’ensemble des descripteurs visuels  $Edv$  est défini par :*

$$\begin{aligned} Edv &= \{Dv_1, Dv_2, \dots, Dv_M\} \\ Dv_i &= (R_i, F_i) \\ F_i &= h(R_i) \end{aligned}$$

$h$  est un opérateur plutôt de bas niveau qui calcule des primitives visuelles ou métriques (e.g. filtre pour la texture ou opérateur morphologique pour la taille). Un descripteur est donc un couple (région, vecteur caractéristique associé).

## 4.4 Concept de la sémantique

Une description visuelle n’est pas suffisante pour accéder à une interprétation robuste de l’image. Deux régions peuvent avoir la même couleur ou la même texture ou tout autre



attribut identique, et leurs sémantiques être complètement différentes. Par exemple le soleil couchant et une voiture rouge, un fleuve et une autoroute vus du ciel dans certaines longueurs d'onde sont semblables. Ou encore une requête d'image contenant un ours blanc peut donner un résultat contenant une personne portant un manteau de fourrure blanche voire un divan recouvert de même. D'un point de vue sémantique ces résultats sont "non signifiant" mais eu égard aux seules caractéristiques visuelles, couleur ou texture, ils sont très proches. L'élaboration d'une description contextuelle de l'image par des machines est un problème ouvert qui sort du cadre de la présente thèse. La machine aujourd'hui dans l'état actuel des connaissances ne sait pas répondre à tout coup avec exactitude à une requête intelligente telle qu'un utilisateur humain le souhaite. En ce sens l'interprétation d'une image ou d'une scène reste subjective. La différence entre ce que le système produit par traitement automatique et le résultat que l'utilisateur humain juge pertinent est définie comme le fossé sémantique.

#### 4.4.1 Niveaux sémantiques

L'objectif principal d'un système CBIR est de satisfaire une requête dont le contexte n'est pas absent. Dans le but d'améliorer l'interprétation automatique du contenu, des traitements de niveau plus élevé sont utilisés après extraction des descripteurs visuels. Le schéma général de description de la sémantique comprend alors plusieurs niveaux d'abstraction comme illustré sur la figure 4.2. Chaque niveau correspond à un traitement élaboré en vue d'une information additionnelle par exemple extraction des positions relatives entre objets de l'image. En montant dans les couches d'abstraction la nature des descripteurs devient de plus en plus contextuelle. Par exemple la description visuelle ou fonctionnelle d'un objet sont deux concepts très différents. On remarquera seulement que la fonctionnalité occupe un niveau plutôt haut sur l'axe de la sémantique tandis que le caractère visuel serait plus volontiers situé en bas d'échelle. Cela ne veut pas dire que l'aspect visuel est pauvre en information contextuelle. En fait il est la base de tout traitement visant à réduire le fossé sémantique : dans certains cas le pouvoir discriminant d'un descripteur visuel pourra restituer un degré sémantique élevé i.e. il est directement interprétable. Il est à noter également qu'une information additionnelle n'implique pas toujours une augmentation de performance. Plusieurs facteurs entrent en jeu tels que le type d'application, la stratégie de recherche, la mesure de similarité utilisée, les caractéristiques contextuelles choisies...etc.

### 4.5 Les relations spatiales

L'analyse spatiale est couramment exploitée dans plusieurs applications dont la recherche d'image [CL91, Sam90]. Elle a pour but d'extraire et représenter des relations entre objets dans le plan. Généralement la description linguistique de l'agencement spatial s'exprime par des prépositions de lieu, par exemple "à côté, dessous, dessus, dans,...".

---

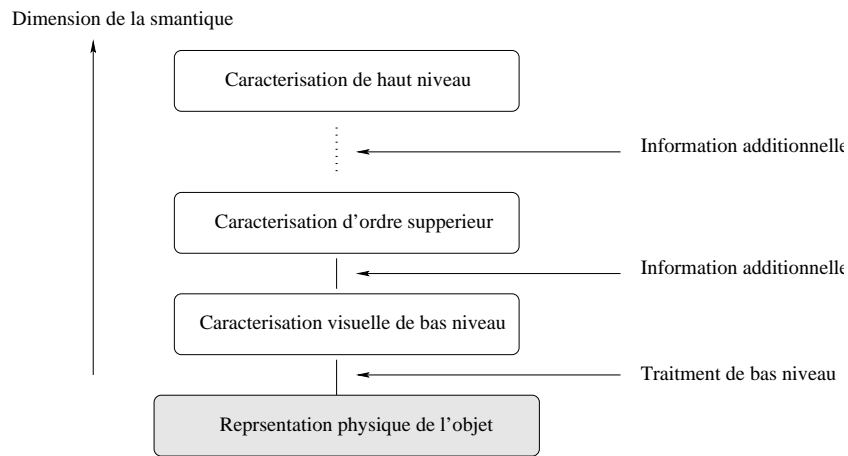


FIG. 4.2 – Les niveaux d’abstraction de la sémantique du contenu

Avec la richesse du vocabulaire des langues naturelles cette description varie d’un langage à un autre. Les méthodes de raisonnement spatial cependant exploitent un nombre limité de telles relations écrites dans un langage universel. Les relations spatiales peuvent être divisées en deux principales catégories (topologiques vs. métriques et directionnelles). La recherche d’image par similarité spatiale est notamment utilisée dans les Systèmes d’Information Géographique (SIG) qui constituent une de nos applications. La requête peut être par exemple du style "extraire les situation où un musée est en face d’une station service". L’analyse spatiale est considérée comme relevant d’un concept lié au langage humain. La mesure de similarité sur les caractéristiques spatiales indique quelles sont les relations spatiales des images de la base qui sont en conformité avec les relations de celle de la requête.

## Relations topologiques

Généralement huit attributs fondamentaux sont employés pour déterminer la relation entre deux objets dans le plan [EF91]. Il s’agit des caractères intérieur, contient, disjoint, adjacent, chevauchement, recouvre, recouvert par, égal fig. 4.3. Entre deux objets il existe toujours une relation topologique mutuellement exclusive [LH92]. C’est à dire qu’il y a une et une seule relation valide entre deux objets.

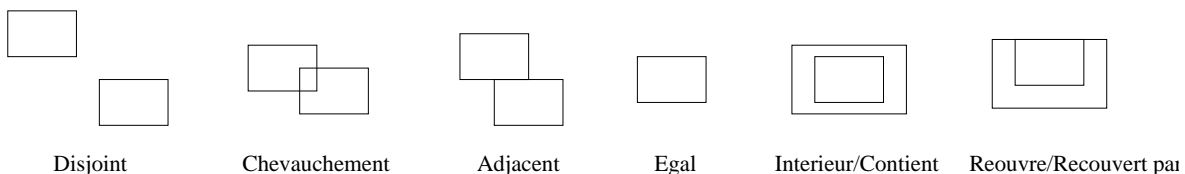


FIG. 4.3 – Relations topologiques



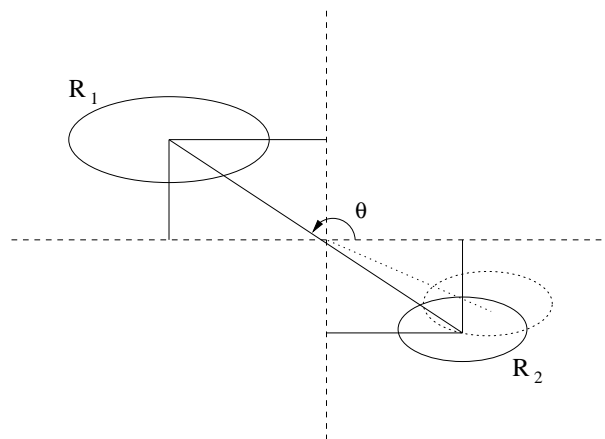


FIG. 4.4 – Orientation entre deux objets dans le plan

### Relations directionnelles

Ces relations expriment les positions relatives entre objets. En d'autres termes elles déterminent comment un objet se situe par rapport à un autre. Ces relations peuvent être de type direction conventionnelle tel que Est, Nord, Ouest, Sud traitée comme variable floue, ou de type métrique représenté par des angles. Dans la figure 4.4 la position relative des deux objets est mesurée par l'angle  $\theta$  ( $R_1$ ,  $R_2$ ) formé par l'axe reliant les deux centroïdes et la ligne horizontale. Les relations directionnelles peuvent également avoir une description floue relative telle que à gauche, en bas, à droite,....

## 4.6 Extraction de descripteurs spatiaux

En montant dans l'axe de la sémantique la caractérisation du contenu a tendance à suivre un modèle de raisonnement logique exprimé en langage humain. Dans cette section nous définissons l'ensemble de relations entre régions qui a été effectivement implémenté pour supporter une description d'ordre supérieur du contenu. Une information directionnelle exprime plus précisément une situation spatiale qu'une information topologique. Nous avons testé d'abord ce genre d'expression. Mais certaines situations importantes comme intérieur ou extérieur sont difficiles voire impossibles à exprimer par de simples relations d'orientation. La combinaison de types de relations permet d'obtenir une description spatiale plus complète du contenu.

La figure 4.4 illustre une configuration spatiale par exemple  $R_2$  est en bas à droite par rapport à  $R_1$ . On remarque que l'objet  $R_2$  peut subir un déplacement significatif sans que cela affecte cette relation spatiale du point de vue humain. Une approche naturelle pour représenter les relations spatiales est la théorie du flou [Zad65] qui est largement utilisée dans ce contexte et que nous avons finalement retenue. On peut se référer à un état de

l'art récent dressé par Bloch [Blo05] sur les relations spatiales floues.

Une expression linguistique d'une position spatiale entre deux objets ou régions peut se traduire par une mesure floue de l'angle  $\theta$ . Le degré d'appartenance d'un angle  $\theta$  à un ensemble  $\xi$  déterminé par une variable linguistique  $L$  est mesuré par la fonction  $\mu(\theta)$ . Cette fonction permet de compenser les imprécisions du langage humain en définissant pour chaque situation son degré d'appartenance à une classe bien déterminée. Un intervalle  $[\theta^-, \theta^+]$  est associé à chaque classe dont les bornes cernent toutes les valeurs possibles de  $\theta$  pour une relation donnée. Plus on se rapproche des bornes plus faible est la probabilité d'appartenir à la classe.

**Définition 4.6.1** Une relation directionnelle entre deux objets  $(R_1, R_2)$  de l'image est définie par l'angle  $\theta$  associé à la fonction  $\mu$  tel que

$$\mu_q(\theta) = \begin{cases} f(\theta) & \theta \in [\theta^-, \theta^+] \\ 0 & \text{autrement} \end{cases} \quad (4.1)$$

$$q \in \xi, \{\xi = \{S_1, S_2, \dots, S_m\} / S_i \in L\}$$

$f$  peut être une fonction trapèze ou trigonométrique [MR94].  $S_i$  est une relation spatiale définie selon une variable linguistique  $L$ .  $m$  est le nombre de relations choisies. Dans notre cas nous optons pour les quatre situations basiques ( $\xi = \{haut, bas, droite, gauche\}$ ) définies selon une variable linguistique de position plus générale et respectivement associées aux fonctions  $\mu_d, \mu_g, \mu_h, \mu_b$ . Ce petit nombre de fonctions d'appartenance a l'avantage de réduire la complexité des calculs. Toutefois la mesure du degré d'appartenance induit implicitement d'autres relations. La valeur de  $\mu$  permet en effet de distinguer si par exemple un objet est plutôt à droite ou plutôt en bas. La relation intermédiaire "en bas à droite" n'existe pas dans  $\xi$ . En effet  $\mu_d(\theta) = \mu_b(\theta) = 0,5$  signifie que l'objet n'est ni tout à fait à droite ni tout à fait en bas. Dans la mesure de similarité deux situations correspondant à  $\mu_d(\theta) = 0,5$  ou  $\mu_b(\theta) = 0,5$  sont considérées identiques : en bas à droite, valeur linguistique qui les caractérise conjointement n'est supportée ici que par l'expression numérique. Pour le calcul des fonctions d'appartenance nous utilisons les fonctions trigonométriques  $\sin^2$  et  $\cos^2$  parcequ'elles apparaissent les mieux adaptées pour supporter les relations directionnelles [MR94].

$$\mu_{\{d,g\}}(\theta) = \begin{cases} \cos^2(\theta) & \theta \in [\theta^-, \theta^+] \\ 0 & \text{autrement} \end{cases} \quad \mu_{\{h,b\}}(\theta) = \begin{cases} \sin^2(\theta) & \theta \in [\theta^-, \theta^+] \\ 0 & \text{autrement} \end{cases} \quad (4.2)$$

Une relation entre deux objets quelconques dans l'image se résume en les cas suivants :

- $\mu_d : \theta^- = -\pi/2$  et  $\theta^+ = \pi/2 \Rightarrow \mu_g = 0$ .
- $\mu_g : \theta^- = \pi/2$  et  $\theta^+ = -\pi/2 \Rightarrow \mu_d = 0$ .

- $\mu_h : \theta^- = 0$  et  $\theta^+ = \pi \Rightarrow \mu_b = 0$ .
- $\mu_b : \theta^- = \pi$  et  $\theta^+ = 0 \Rightarrow \mu_h = 0$ .

$\mu_d$  et  $\mu_g$  sont mutuellement exclusives, de même pour  $\mu_h$  et  $\mu_b$ . Le cas particulier du strict droit, gauche, haut et bas correspond respectivement aux valeurs  $\theta = 0$ ,  $\theta = \pi$ ,  $\theta = \pi/2$  et  $\theta = -\pi/2$ . Remarquons que ces choix de mesure et d'exclusion impliquent  $\mu_b(\theta) + \mu_g(\theta) + \mu_h(\theta) + \mu_b(\theta) = 1$  fig. 4.5.

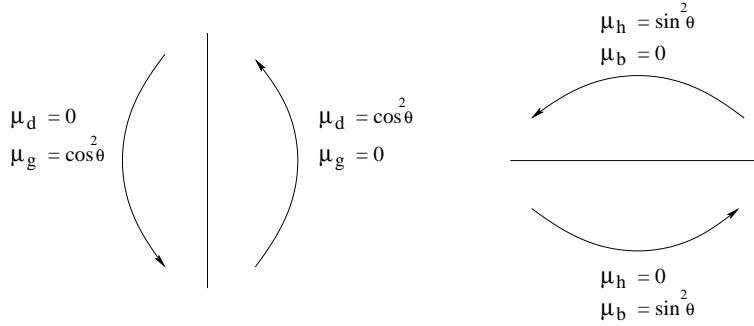


FIG. 4.5 – Représentation des quatres situations basiques

**Définition 4.6.2** *L'ensemble des descripteurs spatiaux  $E_{ds}$  effectivement mis en oeuvre dans notre système est défini par le couple  $(\mu_q(\theta), \tau)$  :*

$$E_{ds} = \{Ds_1, Ds_2, \dots, Ds_N\}$$

$$Ds_i = \{(\mu_q(\theta_{ij}), \tau)\}$$

$$q \in (g, d, h, b); \theta_{ij} = \theta(R_i, R_j)$$

$$\tau \in \{interieur, exterieur\}$$

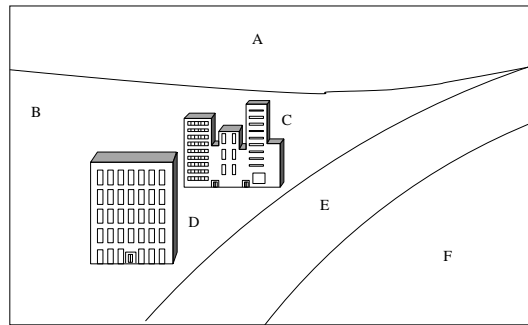


FIG. 4.6 – Un exemple d'une situation réelle

La figure 4.6 illustre une situation d'une scène réelle. Les relations intérieur et extérieur sont notées par  $\tau_{int}$  et  $\tau_{ext}$  elle prennent les valeurs binaires *vrai* ou *faux*. Le tableau 4.1 résume l'ensemble d'arrangements spatiaux entre entités de l'image figure 4.6.

	A	B	C	D	E	F
A	-	$(\mu_h, \tau_{ext})$	$(\mu_h, \tau_{ext})$	$(\mu_h, \tau_{ext})$	$(\mu_h, \tau_{ext})$	$(\mu_h, \tau_{ext})$
B	$(\mu_b, \tau_{ext})$	-	$(\mu_g, \tau_{int})$	$(\mu_h, \tau_{int})$	$(\mu_g, \tau_{ext})$	$(\mu_g, \tau_{ext})$
C	$(\mu_b, \tau_{ext})$	$(\mu_d, \tau_{int})$	-	$(\mu_h, \tau_{ext})$	$(\mu_g, \tau_{ext})$	$(\mu_g, \tau_{ext})$
D	$(\mu_b, \tau_{ext})$	$(\mu_b, \tau_{int})$	$(\mu_b, \tau_{ext})$	-	$(\mu_g, \tau_{ext})$	$(\mu_g, \tau_{ext})$
E	$(\mu_b, \tau_{ext})$	$(\mu_d, \tau_{ext})$	$(\mu_d, \tau_{ext})$	$(\mu_d, \tau_{ext})$	-	$(\mu_g, \tau_{ext})$
F	$(\mu_b, \tau_{ext})$	$(\mu_d, \tau_{ext})$	$(\mu_d, \tau_{ext})$	$(\mu_d, \tau_{ext})$	$(\mu_d, \tau_{ext})$	-

TAB. 4.1 – Un exemple de relations spatiales

## 4.7 Techniques de recherche

Dans cette section nous détaillons les différentes stratégies de recherche employées pour traiter une requête. Quand un usager présente une image requête au système on ne sait pas s'il cherche une région particulière (par exemple un champs de blé), des régions multiples (champs de céréales) ou des régions spatialement organisées (champs de blé à l'est de champs de maïs). Un système fiable devra prendre en considération les différents critères. Par analogie avec les systèmes de recherche textuelle on peut autoriser plusieurs options pour composer sa requête : par exemple un mot seul, plusieurs mots dans le texte ou plusieurs mots associés qui devront être l'un à coté de l'autre. Une telle stratégie offre une marge de manoeuvre supplémentaire à l'utilisateur pour raffiner le résultat. Dans ce contexte notre système supporte trois modes de CBIR, recherche par région singulière, par plusieurs régions et par régions avec relations spatiales. Sur le plan technique la recherche par région est facile à implanter et les performances ne dépendent que des descripteurs visuels. En revanche dans le cas de plusieurs régions la similarité est contrainte par les caractéristiques visuelles mais aussi par la méthode de mise en correspondance entre composantes visuelles respectives de la requête et de la cible. Nous aborderons ce problème dans la suite du chapitre. Soulignons qu'en tous cas ce n'est pas l'utilisateur qui segmente l'image requête, il est obligé d'utiliser la segmentation fournie par la machine.

### 4.7.1 Requête par région

La stratégie de recherche adoptée à ce niveau consiste à extraire toutes les images qui comportent des régions similaires à celles distinguées dans l'image requête. La région requête est donc comparée à toutes les régions de l'image cible de la base, celle qui affiche une distance faible entre attributs sera retenue. Les attributs utilisés dans ce cas sont de type visuel.

## Mesure de similarité entre deux régions

La similarité entre deux régions requête ( $R_i$ ) et cible ( $R'_j$ ) est déterminée par une distance pondérée de type norme  $L_2$  appliquée aux vecteurs de caractéristiques.

$$D_r^2 = \alpha \sum_{k=1}^n (F_k - F'_k)^2 \quad (4.3)$$

Où  $F_k$  et  $F'_k$  représentent respectivement les composantes des vecteurs de caractéristiques de  $R_i$  et  $R'_j$ .  $n$  est donc la dimension de l'espace de caractéristiques. Le poids  $\alpha$  favorise les régions de grande taille dans l'image.

$$\alpha = \frac{1}{\min\{\rho_i, \rho'_j\}} \quad (4.4)$$

$\rho$  est la proportion de surface (en nombre de pixels) d'une région dans l'image. Soit  $w_i$  et  $w'_j$  les surfaces respectives de ( $R_i$ ) et ( $R'_j$ ) le ratio d'une région est égal à la surface de celle ci divisée par la surface totale des régions (taille de l'image) :

$$\rho_i = \frac{w_i}{\sum_k^M w_k} \quad \rho'_j = \frac{w'_j}{\sum_k^N w'_k} \quad (4.5)$$

On a donc

$$w_k \rightarrow 0 \Rightarrow D_r \rightarrow \infty \quad (4.6)$$

On notera que si deux régions très différentes par la taille sont comparées, la plus petite pénalise  $D_r$  en augmentant  $\alpha$ . De même comparer deux régions de surface  $S$  pixels engendre un  $D_r$   $k$  fois plus petit que comparer deux régions de caractéristiques respectivement identiques aux deux premières mais de surface  $S/k$  pixels.

### 4.7.2 Requête par multiples régions

La recherche par multiples régions implique tous les segments de l'image fig. 4.7. Le but de la recherche ici est d'obtenir le nombre maximum de régions requêtes correspondant à celles de la cible. Quand il s'agit d'une recherche par région singulière ou d'une approche globale la mise en correspondance des images ne pose pas de problème pour estimer la similarité. En revanche le problème devient plus complexe avec une requête à plusieurs régions car il faut satisfaire un critère global impliquant toutes les composantes, ce critère réalisant un compromis entre par exemple le nombre de régions impliquées dans l'appariement (global) et le degré de ressemblance dans chaque couple (local). Le mécanisme de la mise en correspondance est illustré par la figure 4.7 où chaque noeud du graphe correspond à une région dans l'image. L'appariement de deux images avec régions en nombre différent pose la difficulté supplémentaire du choix des intervenant dans le calcul de distance. Une solution est de garder les meilleurs couples, un sous graphe est

donc considéré. D'autres solutions consistent à rajouter des noeuds fictifs pour compenser la différence en nombre de régions telle la méthode hongroise (chapitre 1). De plus la segmentation ne fournit pas toujours un résultat fiable. Il devient illusoire de viser une optimalité de la solution. Les éventuelles erreurs de segmentation conduisent également à une mise en correspondance de sous graphes qui favorise des recoupements. D'autres méthodes reposant sur des principes différents visent à compenser les erreurs de segmentation en testant plusieurs appariements simultanés ou en fusionnant des noeuds d'après les premiers résultats d'appariement. La plus populaire est IRM (Integrate Region Matching) [LWW00] qui consiste à donner une grande priorité aux paires de régions présentant une distance minimale i.e. ressemblance maximale. IRM ne met pas l'accent sur une stratégie d'appariement efficace : elle s'appuie principalement sur les distances entre attributs, deux régions adjacentes ou non seront par exemple traitées à l'identique. Dans notre méthode on tient compte de deux notions principales : les distances et les préférences. Nous la décrivons dans les sections suivantes.

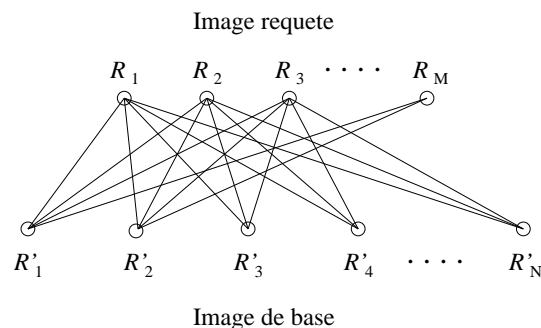


FIG. 4.7 – Comparaison exclusive des regions

### Appariement bipartite imparfait

L'appariement bipartite imparfait (ABI) est une technique qui repose sur l'idée d'un appariement à double sens, c'est à dire de mesurer la ressemblance des régions requête vers les régions de la cible et *vice versa*. Cette méthode permet d'estimer la préférence de la requête entre les cibles d'une part, et d'autre part à la cible d'exprimer son taux d'attrance selon une formule identique. A cause de l'incertitude sur la segmentation automatique on laisse une région correspondre à plusieurs homologues suivant le principe IRM. Mais un plus grand poids est accordé aux couples qui satisfont les deux parties requête et cible i.e. qui se préfèrent mutuellement. Il nous a semblé judicieux d'introduire cette notion dans le calcul de similarité pour réduire les erreurs systématiques que peut produire une requête avec régions incomplètes. L'estimation de l'indice de similarité est consolidée par la mesure bidirectionnelle. Par exemple si une région appartenant à un objet dans l'image requête correspond à plusieurs régions dans l'image cible qui appartiennent à des objets différents, l'objet le plus vraisemblable est celui dont la région a comme premier choix la région requête. On vérifie d'ailleurs expérimentalement sur des groupes d'une vingtaine d'images que la satisfaction d'une requête après mise en correspondance bidirectionnelle est plus pertinente que chacune des deux après appariement monodirectionnel.

Soit  $Q = \{Q = \{R_i\}_1^M / Q = R_1 \cup R_2 \cup \dots \cup R_M\}$  l'image requête et  $B$  l'image de la base  $B = \{B = \{R'_i\}_1^N / B = R'_1 \cup R'_2 \cup \dots \cup R'_N\}$ . La distance entre  $Q$  et  $B$  est basée sur l'appariement des régions. En comparant chaque segment de  $Q$  avec tous les segments de  $B$  nous obtenons une matrice de similarité  $S$  de dimensions  $M \times N$ . Chaque élément  $s_{ij}$  représente la distance  $D_r(R_i, R'_j)$  indiquant le degré de similarité entre la région  $i$  de  $Q$  et la région  $j$  de  $B$ . On écrit la matrice  $S$  sous la forme suivante :

$$S = \begin{bmatrix} s_{1,1} & s_{1,2} & \dots & s_{1,N} \\ s_{2,1} & s_{2,2} & \dots & s_{2,N} \\ \dots & \dots & \dots & \dots \\ s_{M,1} & s_{M,2} & \dots & s_{M,N} \end{bmatrix} \quad (4.7)$$

La matrice  $S$  offre une représentation adéquate de la correspondance entre régions. Les lignes indiquent la correspondance entre chaque région de  $Q$  et toutes les régions de  $B$ . Et les colonnes représentent la correspondance des régions cible  $B$  avec celles de l'image  $Q$ . Si on regarde le long d'une ligne (resp. colonne), le minimum donne forcément la région cible (resp. requête) la plus ressemblante à la région requête (resp. cible) considérée. A partir de cette représentation on peut extraire une liste de préférence pour chaque région dans la quelle on range les régions correspondantes de l'autre image par distance croissante. Une liste de préférence est un indicateur traduisant l'interaction entre les éléments de la requête et de la cible; elle fournit une information complète sur l'état de similarité. Supposons que la matrice de similarité engendre les listes de préférence suivantes :

$$\begin{aligned} R_1 & : R'_2, R'_N, \dots, R'_3 \\ R_2 & : R'_N, R'_2, \dots, R'_1 \\ & \vdots \\ R_M & : R'_2, R'_4, \dots, R'_N \end{aligned} \quad (4.8)$$

$$\begin{aligned} R'_1 & : R_2, R_M, \dots, R_3 \\ R'_2 & : R_4, R_5 \dots, R_M \\ & \vdots \\ R'_N & : R_M, R_3 \dots, R_1 \end{aligned} \quad (4.9)$$

Une telle représentation est la même qui supporte naturellement une optimisation de type mariage stable avec listes incomplètes. Le mariage stable (stable matching) et ses variantes a engendré plusieurs algorithmes [GI89], notamment celui de Gale et Shapley (chapitre 1). Le but d'un algorithme de mariage stable est d'exhiber des couples tels que deux éléments dans deux couples différents ne se préfèrent pas mutuellement à leurs conjoints respectifs actuels. Outre que ce type d'algorithme permet difficilement une mise en correspondance à plusieurs éléments, fondamentale ici, la contrainte de stabilité

n'est pas majeure dans les appariements requête/image-base. En revanche, la satisfaction globale qui elle est importante peut être gravement pénalisée par l'imperfection de la segmentation. En effet pour Gale et Shapley, si un élément n'est pas préféré par le partenaire visé alors il est obligé d'en choisir un autre à distance plus élevée donc le coût augmente. Nous choisissons d'agir au contraire sur la réduction du coût en favorisant les segments les plus proches, sans contraindre un élément à changer ses préférences. Nous ne retenons des mariages que la représentation par liste de préférence qui sert à pondérer les minimums trouvés dans  $S$ . Non seulement un élément peut être apparié avec plusieurs autres mais encore la complexité de calcul est en  $N$  au lieu de  $N^2$

### Mise en correspondance par lignes

La mesure consiste à extraire de chaque ligne de la matrice  $S$  un minimum. Dans ce cas là nous obtenons la région cible  $j$  la plus semblable à la région requête  $i$ . Le couple  $(i, j)$  traduit la mise en correspondance entre ces deux régions avec le coût minimal. La formule suivante exprime l'appariement en question :

$$\min_i = \min_j \{s_{ij} \mid j = 1, 2, \dots, N\} \quad (4.10)$$

### Mise en correspondance par colonnes

A l'instar de la mise en correspondance par lignes les minimums sont obtenus cette fois ci sur les colonnes de la matrice  $S$ . Ici la région cible  $j$  cherche la meilleure région requête qui lui correspond :

$$\min_j = \min_i \{s_{ij} \mid i = 1, 2, \dots, M\} \quad (4.11)$$

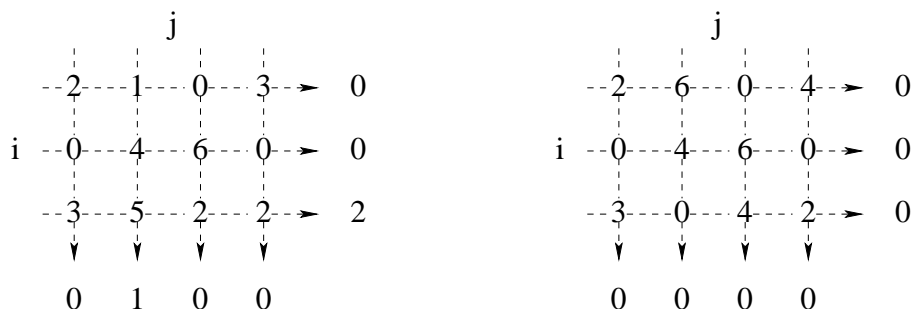


FIG. 4.8 – Deux exemples d'extraction des minimums lignes et colonnes



### Appariement bipartite imparfait pondéré (ABIP)

ABIP consiste à calculer la fonction coût de la mise en correspondance ABI avec une contrainte supplémentaire qui est la pondération par ordre de préférence. Il semble intéressant de favoriser les régions proches dans l'espace caractéristique en ce que leur association est mieux porteuse de sémantique. Quand nous construisons la matrice de similarité  $S$  une région peut correspondre à un minimum dans un sens (lignes ou colonnes) mais pas dans l'autre. Cela ne signifie pas que cette région est véritablement ressemblante : il se peut qu'elle soit la plus proche parce que les autres régions de la même partie sont trop loin. Pour augmenter donc la confiance dans cette région il faut examiner si elle est choisie par l'autre partie. Ceci rappelle à nouveau les mariages stables si ce n'est qu'on ne marie pas les couples. La technique se limite à donner, via une pondération, plus d'importance aux éléments qui préfèrent un partenaire d'une partie et qui sont préférés de cette autre partie.

Supposant que  $(R_i, R'_k)$  et  $(R'_j, R_l)$  sont deux couples appariés respectivement sur la ligne  $i$  et la colonne  $j$  de la matrice  $S$ . Soit  $p_i$  l'ordre de  $R_i$  dans la liste de  $R'_k$  et  $p_j$  celui de  $R'_j$  dans la liste de préférence de  $R_l$ .  $p_i$  ou  $p_j$  est l'indice de préférence, décroissant avec cette préférence.

$$1 \leq p_i \leq M \quad , \quad 1 \leq p_j \leq N \quad (4.12)$$

Nous accordons plus d'importance aux éléments issus de la mise en correspondance par lignes ou colonnes et qui présentent un indice de préférence minimal, cet indice devra également minimiser une distance. La pondération de chaque appariement est exprimée sous la forme suivante :

$$\gamma_q = e^{K \frac{p_q}{2}} \quad (4.13)$$

Remarquons que la pondération  $\gamma_q$  est exponentielle croissante avec l'indice de préférence  $p_q$ .  $\gamma_q$  est donc minimale quand  $p_q = 1$ .  $K$  est une constante qui détermine l'influence des préférences dans le calcul de la distance. En variant  $K$  de 0,01 à 1, une évaluation heuristique sur une centaines d'images a indiqué que  $K = 0,1$  réalise un bon compromis entre ABI et ABIP.

### Calcul de la fonction coût

Le calcul de la fonction coût est la phase finale qui permet de déterminer le degré de similarité entre deux images. La mesure doit inclure le coût global de la mise en correspondance, pour cela nous utilisons la distance de Chamfer qui exploite justement les minimums par ligne et colonne. Elle est pondérée par l'ordre de préférence.

$$Dv = \frac{1}{M} \sum_{i=1}^M \gamma_i \min_i + \frac{1}{N} \sum_{j=1}^N \gamma_j \min_j \quad (4.14)$$

Nous constatons ici que la distance entre  $Q$  et  $B$  est le résultat de la double mise en correspondance qui exprime leur interaction. Dans le cas particulier où  $M = N$  et on a le même minimum ligne et colonne ( $\min_i = \min_j \Rightarrow \gamma_i = \gamma_j = 1$ ) l'appariement est exact. C'est à dire que chaque région de  $Q$  correspond à une seule région de l'image base  $B$ . Pour retrouver la fonction coût de ABI il suffit d'imposer  $\gamma_i = \gamma_j = 1$ .

### 4.7.3 Recherche par multiples régions avec relations spatiales

Une deuxième contrainte s'impose maintenant au système de recherche. Les images sont jugées similaires non seulement par le critère visuel mais aussi parceque satisfaisant une certaine configuration spatiale. La mesure de similarité spatiale est appliquée après celle de similarité visuelle sur les régions candidates ayant un bon score au premier test.

#### Mesure de similarité spatiale

Dans notre approche la similarité spatiale s'applique aux régions issues de l'appariement visuel. En effet deux régions de  $Q$  engendrent une relation spatiale de même que leurs correspondants respectifs dans  $B$ . La distance entre ces deux relations mesure la similarité spatiale. Cependant une telle distance ne peut pas être calculée sur des régions à multiple correspondant parce que cela mène à une situation de conflit. Un exemple extrême est que si deux régions de  $Q$  correspondent à une seule région de  $B$  il n'existe pas de relation dans  $B$  donc pas de distance entre relations. Compte tenu d'une possibilité de mise en correspondance multiple il est alors nécessaire de revoir l'appariement. Nous optons donc pour un filtrage des éléments sources de conflit en ne gardant qu'un seul élément à la fois, le plus proche. Notons que l'on pourrait le cas échéant opérer des regroupements type "croissance de région".

La distance spatiale est calculée de  $Q$  vers  $B$ . Soit les deux couples  $(R_i, R'_j)$  et  $(R_l, R'_k)$  la distance spatiale peut être formulée comme suit :

$$ds_i = \beta \sum_{q=1}^n (\mu_q(R_i, R_l) - \mu_q(R'_j, R'_k))^2 + \sum_{q=1}^2 (\tau_q(R_i, R_l) \text{ XOR } \tau_q(R'_j, R'_k)) \quad (4.15)$$

La distance spatiale globale est alors ainsi :

$$Ds = \sum_{i=1}^C ds_i \quad (4.16)$$

$C$  est le nombre de relations possibles entre couples appariés  $C \leq \min\{C_Q, C_B\}$  ( $C_Q$  et  $C_B$  nombre de relations spatiale dans  $Q$  et  $B$  respectivement),  $\mu_q$  et  $\tau_q$  définis tel qu'en 4.6.

## Calcul de la fonction coût

Dans le cas de recherche d'image avec multiples régions et relations spatiales nous définissons la distance entre deux images par la somme des distances visuelle et spatiale.

$$D = Dv + Ds \quad (4.17)$$

## 4.8 Application

### 4.8.1 Implantation

Le schéma général du système de recherche est constitué de plusieurs modules dont chacun prend en charge une tâche spécifique. De la segmentation à la recherche la chaîne du traitement est entièrement automatisée et l'intervention humaine ne porte que sur quelques paramètres. On distingue deux phases de traitement, hors-ligne et en ligne, qui facilitent respectivement la recherche et la navigation de l'utilisateur.

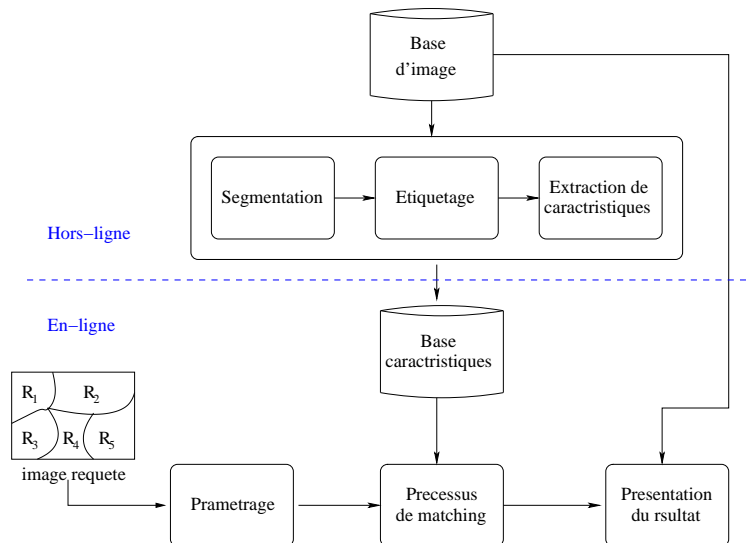


FIG. 4.9 – L'architecture du système de recherche

### 4.8.2 Signature visuelle

Dans notre système le choix de l'attribut visuel est paramétrable. Outre les nôtres, nous avons implanté principalement des méthodes classiques de texture et pour la couleur nous utilisons l'histogramme. Les caractéristiques de texture sont donc celles déjà étudiées dans le chapitre 2 (GLCM, MCM, GABOR, FGVPO). Ces descripteurs sont régis par le paramétrage suivant :

- GLCM : distance = 1 , moyennage sur les orientations ( $0^\circ, 45^\circ, 90^\circ, 135^\circ$ ) avec symétrie, taille de la matrice  $256 \times 256$ . Les caractéristiques sont celles déterminées par Meastex (voir chapitre 2). Dimension de l'espace de caractéristique :  $n = 12$ .
- MCM : 12 motifs avec recouvrement, mêmes orientations que GLCM. Taille de la matrice :  $12 \times 12$ .
- Filtres de Gabor (GABOR) :  $\lambda = 2, 4, 8$  et  $\theta = 0^\circ, 45^\circ, 90^\circ, 135^\circ$  , taille de la fenêtre =  $17 \times 17$ ,  $n = 12$
- Filtre de Gabor pour valeur du parcours optimal (FGVPO) : 12 motifs avec recouvrement, même paramétrage que les filtres de Gabor précédents.  $n = 12$
- Histogramme : RGB avec 256 niveaux pour chaque plan.

### 4.8.3 Signature spatiale

Pour la signature spatiale nous avons considéré les informations suivantes :

- L'importance relative de la région dans l'image (rapport surface région / surface image).
- Coordonnées du centre de gravité de chaque région mesurées en pixel.
- Orientation entre régions  $\mu_d, \mu_g, \mu_h, \mu_b$
- Relations de topologie intérieur / extérieur ( $\tau_{int}, \tau_{ext}$ ).
- Distance entre centres de gravité.

### 4.8.4 Traitement hors-ligne

Le traitement hors-ligne est une tâche qui s'effectue sans contrainte temporelle véritable parce qu'indépendante du processus de recherche. Il produit un indice signature de chaque image sur une base d'attributs (couleur, texture, forme, taille, relations spatiales,...). La phase hors-ligne de notre système se construit en trois étapes principales : la segmentation pour extraire les composantes visuelles, l'étiquetage en composantes connexes pour exhiber les régions et l'extraction de caractéristique de chaque région puis leur stockage dans un fichier index (voir figure 4.9). Ce fichier est utilisé par la suite comme base de caractéristiques image. Toute modification de la base nécessite une mise à jour du fichier index.

### 4.8.5 Traitement en ligne

Lors de cette phase l'utilisateur fournit un ou plusieurs exemples sous forme d'image ou de parties d'image et le système explore la base pour trouver les images les plus ressemblantes. Le traitement en-ligne de notre système se décompose en deux parties :

une comparaison avec les images de la base et l'affichage des résultats. La procédure de comparaison exploite l'une des techniques de recherche décrites précédemment et opère sur la base des caractéristiques. L'affichage dispose simplement les images selon le score engendré par la procédure de comparaison.

### 4.8.6 Interface utilisateur

Nous avons construit une interface graphique pour faciliter l'interaction usager-système. Cette interface est basée sur la bibliothèque GTK+<sup>1</sup> sous un environnement linux. L'utilisation de GTK+ pour la création de l'interface est avantageuse sur plusieurs points : logiciel multi plateformes libre et utilisable avec plusieurs langages de programmation. L'interface se veut intuitive avec des fonctionnalités simplifiées pour permettre à un utilisateur non expert d'effectuer des recherches et naviguer sur les images sans avoir des connaissances approfondies en traitement d'image ou en informatique.

L'interface graphique est composée de deux parties principales ainsi qu'une barre de menu voir figure 4.10. La fenêtre de droite est celle qui contient l'affichage du résultat : l'utilisateur peut choisir le nombre d'images en spécifiant la taille de la matrice d'affichage. La gauche de l'interface contient l'image requête et les options de la recherche "globale", "locale", "texture", "couleur" ainsi qu'un bouton pour la lancer. Pour l'option locale trois types de recherche sont autorisés : simple région, multiple régions et relations spatiales. Les entités de la barre de menu sont "Fichier" "Base", "Image" "Edit", "Outils", "?" supportant les actions :

- Fichier : sauver ou quitter une session
- Base : charger une base
- Image : charger une image requête, sachant qu'on peut le faire aussi à partir de la partie droite en cliquant sur l'une des images.
- Edition : sélectionner la taille d'affichage
- Outils : afficher la courbe rappel-précision (la sélection des images pertinentes se faisant dans un fichier éditable).
- ? : aide à l'utilisation

## 4.9 Evaluation des performances

La détermination de la pertinence d'un résultat reste l'une des grandes problématique dans la recherche d'image par le contenu. On manque même encore de sens commun en la matière [MMS+01, NKW97]. Par exemple la notion de test standard en CBIR et donc a fortiori l'élaboration d'une plateforme d'évaluation d'algorithmes, se heurtent au fait que

---

<sup>1</sup>[www.gtk.org](http://www.gtk.org)

---

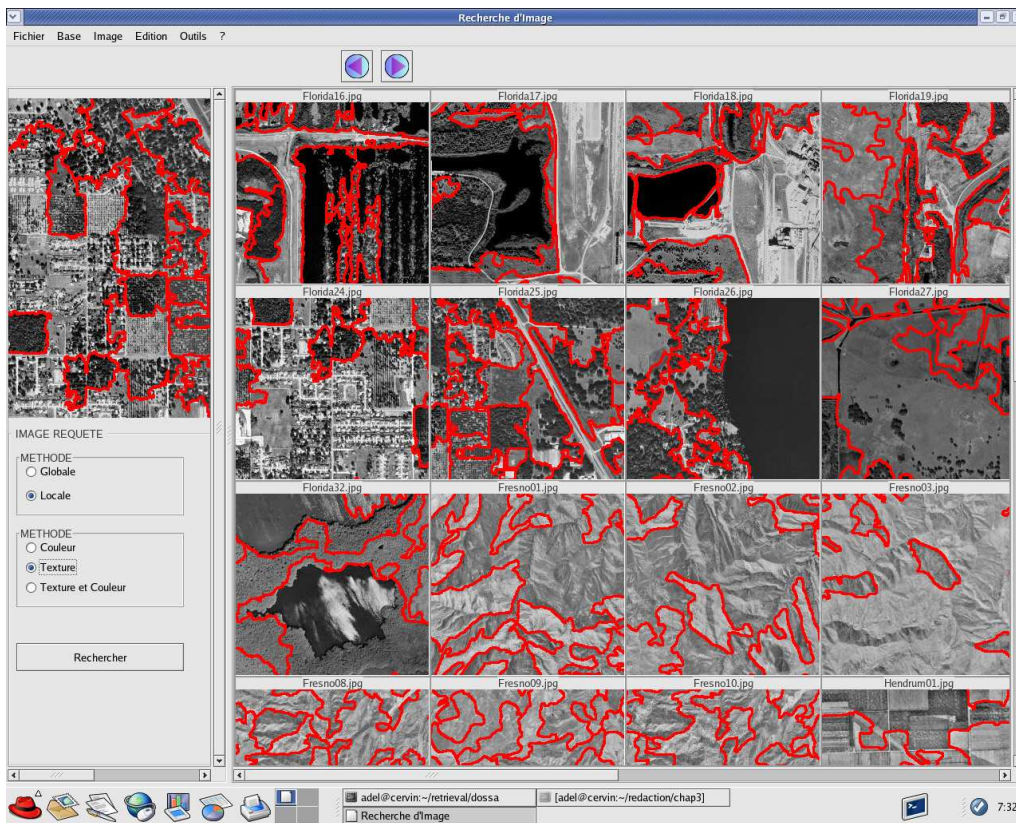


FIG. 4.10 – Interface utilisateur

les humains ne sont pas forcément d'accord. Il y a une difficulté ne serait ce qu'à établir une base d'image commune avec des classes sur la pertinence desquelles le jugement serait commun. Ce manque vient de la diversité des applications visées qui implique des images spécialisées mais encore de la subjectivité liée à l'action envisagée ou au but poursuivi qui motivent la recherche, tout ce que nous avons évoqué comme faisant la sémantique. Dans tous les cas on ne peut s'attendre à ce qu'un être humain examine des grandes bases d'image de l'ordre du Méga images ou Téra octets. Le jugement humain quant à la pertinence portera au mieux sur des portions séparées de la base.

Dans ce contexte, une difficulté d'ordre plus pratique réside dans la création de collections assez diverses pour répondre au différents domaines de spécialisation comme par exemple les images satellite, médicales, anthropométriques, généralistes,... Malgré cela plusieurs efforts ont été déployés pour répondre à ce besoin, par exemple la base généraliste ou photothèque Corel<sup>2</sup> où l'on peut distinguer une centaine de groupes contenant des images admises comme similaires [CTB<sup>+</sup>99]. Ces groupes facilitent la mesure de pertinence et autorisent une certaine confiance sur des classes qui contiennent des images de sujets différents (eg. autobus, bâtiment,...). On constate que plusieurs groupes de recherche ont utilisé cet outil pour juger leurs résultats. Il Reste que des applications spécifiques, où l'expertise est diffuse, génèrent plus de difficulté que les images dites générales où cette

<sup>2</sup>[www.corel.com](http://www.corel.com)

expertise peut se matérialiser comme du simple bon sens. Notre travail vise justement de tels domaines spécifiques comme celui des images satellite ou aériennes. Il n'existe pas des bases communes organisées (Benchmark) pour tester la recherche de ces images. En revanche il existe des serveurs Web qui mettent à disposition des images du genre (voir section 4.10) pour que les chercheurs tentent progressivement de se comparer. La méthode la plus commune ici est de définir des classes de consensus comme par exemple "zone urbaine". Toutefois ceci reste général et ne répond pas forcément aux besoins de l'utilisateur visé.

## Rappel et précision

Le Rappel et la Précision  $Precision \times Recall$  sont les jauges communément utilisées pour l'évaluation des performances en recherche d'information et notamment en CBIR. Le rappel est le rapport du nombre des documents pertinents trouvés au nombre de documents pertinents stockés dans la base (si tant est donc qu'on le connaisse), souvent exprimé en pourcentage Eq. 4.18. Si nous admettons qu'il y a par exemple 100 images pertinentes dans la base et le système en trouve 10 alors le rappel vaut 10%. Du paragraphe précédent sur la relativité de la pertinence on comprend qu'il soit difficile de mesurer le rappel. La précision est le rapport du nombre des documents pertinents trouvés sur le nombre total des documents trouvés Eq. 4.19 (eg. 5 images pertinentes sur 10 trouvées, précision = 50%). Elle souffre des mêmes manques.

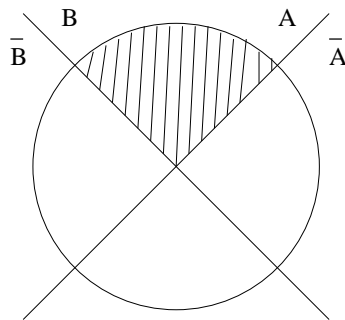


FIG. 4.11 – A ensemble d'éléments trouvé et B ensemble d'éléments pertinents

Si on suppose que  $A$  est le nombre total de documents trouvés et  $B$  le nombre total des documents pertinents dans la base (voir figures 4.11) le rappel et la précision peut se traduire par les formules suivantes :

$$Rappel = \frac{\#(A \cap B)}{\#(B)} \quad (4.18)$$

$$Precision = \frac{\#(A \cap B)}{\#(A)} \quad (4.19)$$

Pour une mesure plus relative donc plus sûre le rappel et la précision devront être utilisés ensemble. Une grande précision et un grand rappel témoignent de performances

élevées. La représentation graphique du rappel en fonction de la précision est la méthode standard pour évaluer et comparer les performances.

### *La méthode d'évaluation Trec Eval*

Cette méthode s'affirme comme un des standards d'évaluation de performances des systèmes de recherche d'information. Elle opère selon la norme NIST (National Institute of Standards and Technology). Trec-Eval<sup>3</sup> est un logiciel qui permet, à partir d'un résultat de requête, de générer les valeurs du rappel et précision ainsi que des mesures statistiques. La méthode de calcul du rappel et précision repose sur le moyennage des valeurs sur un ensemble de requêtes. La moyenne est calculée après une interpolation de la précision sur des valeurs du rappel bien déterminées et cela pour chaque résultat de recherche. Dans nos expériences nous avons utilisé une version récente du Trec-Eval (trec\_eval.8.0). Les différentes mesures prises en compte dans l'évaluation de notre système sont :

- Statistiques générales : comprennent le nombre total d'images requêtes, documents pertinents, documents extraits et images pertinentes trouvées.
- MAP : (*Mean Average Precision*) Précision moyenne.
- GMAP : (*Geometric Mean Average Precision*) Moyenne géométrique du MAP.
- R-Prec : Précision après un nombre R de documents pertinents trouvés.
- Courbe rappel-précision

## 4.10 Application au SIG

Les systèmes d'information géographique ou SIG sont des outils constitués d'un ensemble de logiciels et matériels qui permet d'analyser, gérer et représenter des données géographiques. La richesse des SIG en matière d'information en a fait des outils indispensables dans plusieurs domaines tels que la planification urbaine, la prévention des catastrophes naturelles, les guides touristiques, les outils pédagogiques,...etc. Les données d'un système géographique proviennent de plusieurs sources de types différents cartes, image,...stockés conjointement dans des bases de données. Parmi les sources d'information importantes dans un système SIG sont les images satellite. Le développement de la technologie spatiale a facilité la mise en orbite de plus en plus de satellites de technologie récente pour des application SIG. Historiquement les images du satellite "Landsat" ont été utilisées pour l'étude des changements climatiques ou écologiques et de leurs conséquences, entre autres application la déforestation du bassin Amazone. Plus récemment un satellite tel que ERS-1 équipé d'un radar à ouverture synthétique a été utilisé pour surveiller la zone polaire nord et le réchauffement de la planète. Parmi les applications les plus récentes utilisant des images satellitaires on trouve le système de navigation "Googleearth" lancé par la société "Google".

---

<sup>3</sup>[http://trec.nist.gov/trec\\_eval/](http://trec.nist.gov/trec_eval/)

---



Les bases d'images satellite actuelles déjà gigantesques ( $10^6 < \cdot < 10^{12}$  octets), augmentent d'une manière exponentielle. Parmi les objectifs visés est par exemple celui d'étudier le phénomène des inondations dans le monde. L'indexation manuelle pour une telle tâche est difficile à cause du volume des données; de plus l'indexation textuelle même réalisée par un expert reste une procédure subjective qui varie d'un spécialiste à l'autre. Il y a donc un réel besoin d'automatiser pour une utilisation plus rapide et efficace.

### 4.10.1 Base d'image

Les images que nous traitons ici proviennent de la base du serveur Terraserver<sup>4</sup> [Bar98, BGC04]. Avec la collaboration de USGS (*U.S Geological Survey*) ce serveur contient l'une des plus grande base de données géographique mise en ligne permettant une navigation sur des images aériennes et satellitaires en plusieurs résolutions (512m/pixel - 0,25m/pixel). La base d'image que nous avons construite compte environ 400 images dont 300 avec une résolution de 8m et 4m et 100 à résolution de 1m et 0,5m. Nous avons choisi des sites géographiques diversifiés avec différents reliefs pour enrichir le contenu de la base. On trouvera notamment des chaînes montagneuses, des terres agricoles, des zones urbaines,.... Dans le but de favoriser une meilleure mesure de pertinence, les images similaires sont triées en groupes de 6 à 10 éléments, soit 2 à 2,5 % de la base chacun. Généralement les images d'un groupe sont choisies dans des zones connues pour voisines géographiques et où les propriétés du terrain entraînent le même aspect visuel. Le choix de zones avec approximativement le même relief facilite le jugement de la pertinence pour un humain non expert.

Les images utilisées sont de  $600 \times 400$  pixels en format jpeg monocal. Il est à noter que ces images acquises avec une ancienne technologie ont une qualité plus ou moins bonne par rapport aux plus récentes comme celles de Googlearth. Toutefois Terraserver commence à proposer des image de haute qualité sur certains lieux, et nous continuons d'enrichir notre base avec de nouvelles images.

Pour mettre en évidence l'apport de la caractérisation par les MCM (Matrice de Co-occurrence des Motifs) et FGVPO (Filtre de Gabor des Valeurs de Parcours Optimaux) étudiées dans le chapitre 2, nous les avons comparées avec les méthodes classiques (GLCM, filtre de Gabor,...). Les tests sont effectués dans les mêmes conditions pour toutes les caractéristiques afin de garantir une cohérence des mesures.

### 4.10.2 Recherche partielle

La recherche partielle consiste à sélectionner une partie de l'image (région) dans le but de trouver toute les images qui contiennent cette partie. Ce type de recherche est

---

<sup>4</sup><http://terraserver.microsoft.com/>

---

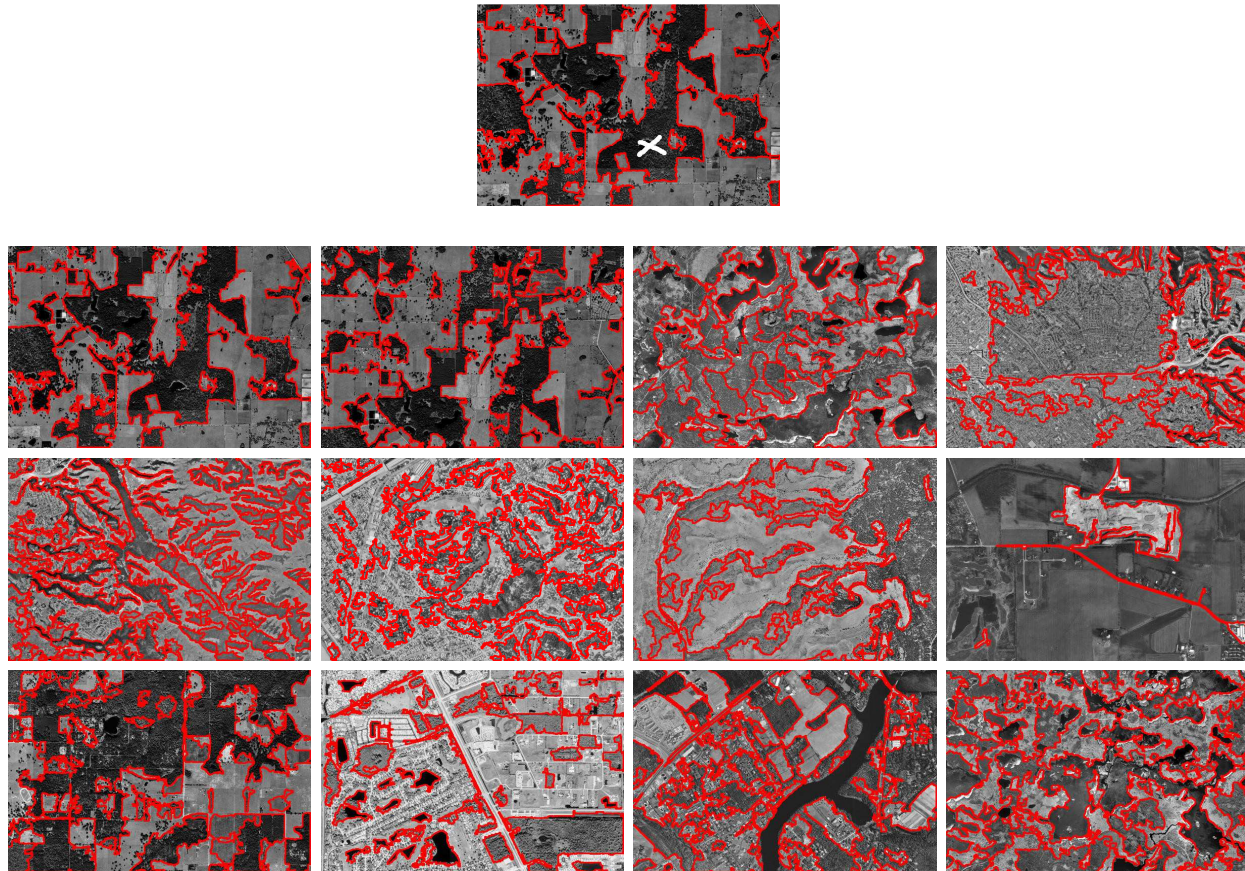


FIG. 4.12 – Exemple de requête par région (segment marqué)

facile à implanter et ne rencontre pas le problème d'appariement de régions. L'interface utilisateur propose une image segmentée et l'utilisateur sélectionne une région de son choix en cliquant dessus. Cette technique est similaire à celle du système Blobworld [CTB+99]. La figure 4.12 illustre un exemple de recherche en régions avec MCM<sup>5</sup> comme descripteur visuel. Les images résultats sont triées, selon le score, de gauche à droite et de haut en bas.

La table 4.2 résume les mesures statistiques effectuées pour un ensemble de 51 images requêtes. Les régions sont choisies aléatoirement dans chaque image. Ces mesures indiquent que MCM présente les meilleures performances avec un taux MAP supérieur d'environ 10% par rapport à la deuxième meilleure méthode (FGVPO). Suivent en classement par les filtres de Gabor et enfin la GLCM qui présente un MAP, GMAP et RPrez plus faibles. Le rappel et la précision moyenne sont représentés par la figure 4.13. Nous remarquons que la courbe MCM est supérieure à celles des autres caractéristiques. On peut également constater que GABOR et FGVPO ont pratiquement le même comportement avec un léger apport de FGVPO.

<sup>5</sup>MCM à 12 motifs avec recouvrement

Statistiques générales	Méthodes	MAP	GMAP	RPrec
Nombre de requêtes : 51	MCM	0.2902	0.1005	0.2710
Documents extraits : 21726	GLCM	0.1301	0.0544	0.1119
Documents pertinents : 768	FGVPO	0.1939	0.0739	0.1727
Documents pertinents trouvés : 528	GABOR	0.1890	0.0717	0.1647

TAB. 4.2 – Mesure des performances pour requêtes en région

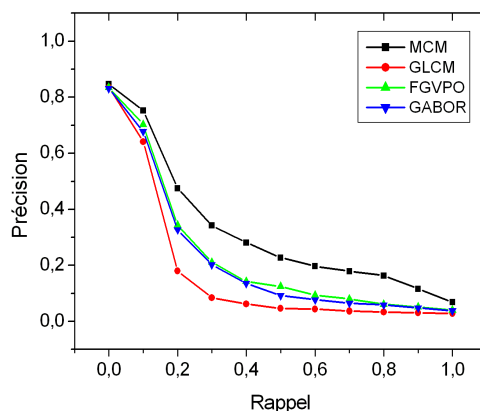


FIG. 4.13 – Comparaison des performances pour une recherche de région

### 4.10.3 Recherche par multiples régions

Nous testons à présent la recherche par multiples régions en comparant les différentes méthodes. Nous étudions le comportement du système vis à vis de la résolution de l'image, la méthode de mise en correspondance et des transformations géométriques. La figure 4.14 montre un exemple de résultat de recherche selon la méthode de mise en correspondance ABIP en utilisant MCM comme caractéristique. La résolution de l'image requête est de 1 mètre le groupe de pertinence pour ces images contient deux résolutions, 1m et 50cm, avec 15 images considérées pertinentes pour chaque résolution. L'image requête (en milieu haut) est composée de trois principales régions, zone urbaine, plage et mer. Nous remarquons sur le résultat que la plupart des images retrouvées comprend ces régions. Toutefois d'autres éléments pertinents ne sont pas détectés par le système. Les premières images trouvées dans cet exemple sont d'une résolution de 1m. Les autres images que nous considérons non pertinentes contiennent au moins une région d'aspect visuel similaire à la requête.



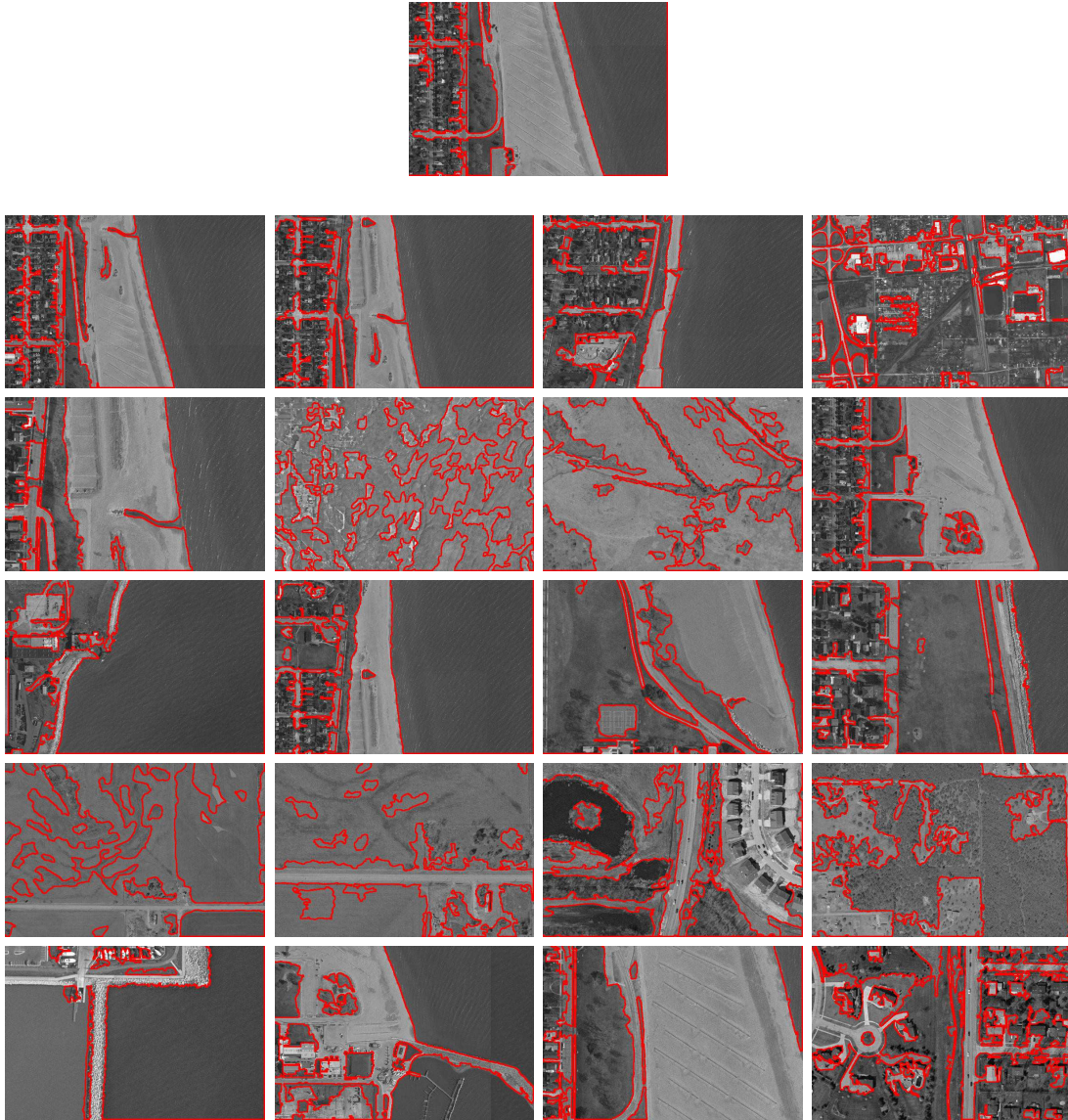


FIG. 4.14 – Un exemple de requête avec ABIP

## Influence de la résolution

Nous avons testé une dizaine de requêtes par la méthode ABIP à différentes résolutions. En premier lieu une seule résolution est considérée, les images pertinentes sont donc celles qui appartiennent au même groupe de pertinence et ont la même résolution. En second lieu nous rajoutons des images du même endroit géographique prises à une résolution différente mais avec une petite différence du facteur d'échelle, parce que la dissemblance croît avec cette différence. Une zone urbaine à 8m de résolution n'a pas le même arrangement structural quand elle est vue à 0,5m. Ce problème d'échelle est général pour les textures. Pour éviter cela nous avons couplé les résolutions (4m avec 8m) et (1m avec 0,5m).

Le tableau 4.3 représente les mesures de performance du système en considérant une seule résolution pour les images pertinentes. Le tableau 4.4 contient les mesures effectuées pour les mêmes requêtes du tableau précédent, en revanche la pertinence des images comprend deux résolutions confondues. On remarque que le nombre de documents pertinents est supérieur à celui pour une seule résolution. On peut vérifier l'évidence que les résultats pour une seule résolution sont meilleurs que ceux de deux résolutions.

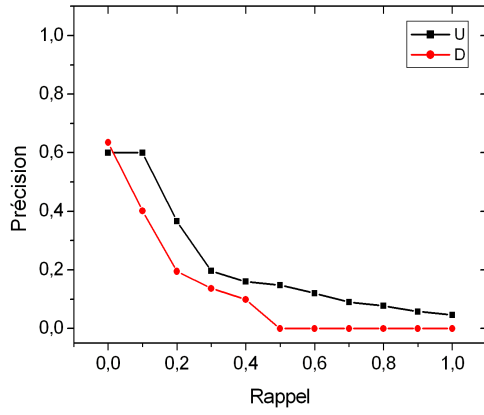
Les courbes du rappel et précision obtenues pour les mêmes images par Trec Eval (figure 4.15) montrent que, indépendamment du type de caractéristique, les performances diminuent quand on implique plusieurs résolutions dans le processus de recherche. En introduisant une résolution supplémentaire sur les images pertinentes nous avons contraint le système à trouver plus d'images pertinentes. Comme la caractérisation ne peut pas supporter parfaitement les variations du facteur d'échelle alors cela mène logiquement à une dégradation des performances.

Statistiques générales	Méthodes	MAP	GMAP	RPrec
Nombre de requêtes : 10	MCM	0.2074	0.0052	0.1722
Documents extraits : 4260	GLCM	0.0671	0.0027	0.0389
Documents pertinents : 188	FGVPO	0.1229	0.0037	0.1167
Documents pertinents trouvés : 108	GABOR	0.1001	0.0033	0.0778

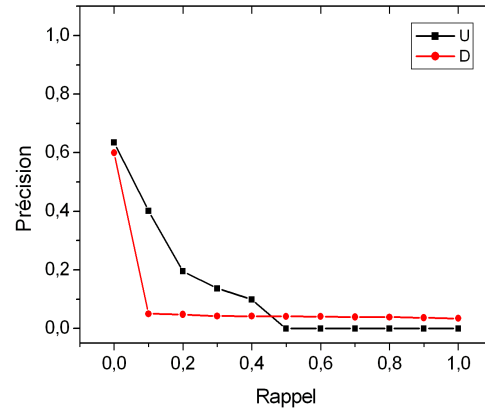
TAB. 4.3 – Résultat avec une seule résolution pour les images pertinentes

Statistiques générales	Méthodes	MAP	GMAP	RPrec
Nombre de requêtes : 10	MCM	0.1107	0.0827	0.1395
Documents extraits : 4260	GLCM	0.0441	0.0424	0.0450
Documents pertinents : 377	FGVPO	0.0723	0.0622	0.0978
Documents pertinents trouvés : 177	GABOR	0.0544	0.0422	0.0605

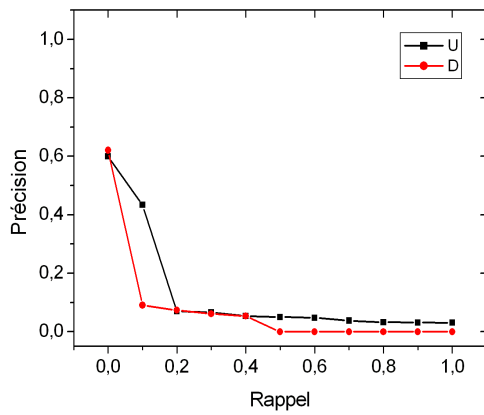
TAB. 4.4 – Résultat avec deux résolutions pour les images pertinentes



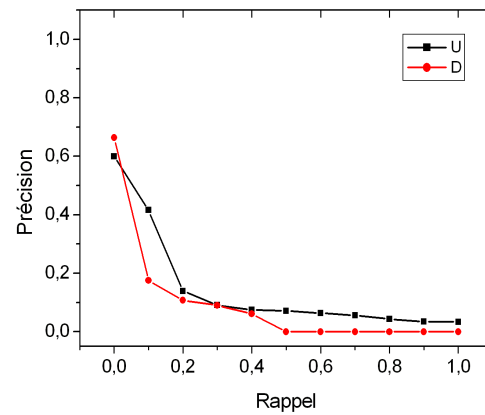
(a)



(b)



(c)



(d)

FIG. 4.15 – Comparaison des résolutions. Légendes U : une seule résolution, D : deux résolutions. (a) MCM (b) GLCM (c) FGVPO (d) GABOR

## Influence de la méthode de "matching"

Dans cette section nous allons comparer trois méthodes de mise en correspondance des régions : ABIP, ABI et mariage stable (MS). Nous avons évalué les performances de chaque méthode sur les différentes caractéristiques.

Dans le tableau 4.5 nous avons les résultats des différentes méthodes selon les trois procédures de mise en correspondance. En comparant les MAP, GMAP et RPreC sur le plan de la caractérisation on remarque que MCM produit les meilleures performances pour toutes les méthodes du matching. Sur le plan de la mise en correspondance, ABIP est supérieure aux autres méthodes.

La figure 4.16 montre la réponse que notre système fournit pour ces 3 méthodes de matching. Les courbes du rappel et précision indiquent que ABI et ABIP ont le même comportement avec une amélioration des performances pour ABIP. Cependant MS donne des résultats moins intéressants.

Statistiques générales
Nombre de requêtes : 51
Documents extraits : 21726
Documents pertinents : 768
Documents pertinents trouvés : 528

Méthodes	MAP			GMAP			RPreC		
	ABI	ABIP	MS	ABI	ABIP	MS	ABI	ABIP	MS
MCM	0.3556	0.3667	0.1926	0.1266	0.1331	0.0685	0.3202	0.3290	0.1946
GLCM	0.1955	0.1962	0.1206	0.0717	0.0730	0.0494	0.1670	0.1690	0.0905
FGVPO	0.2723	0.2852	0.1395	0.0952	0.0970	0.0543	0.2452	0.2611	0.1098
GABOR	0.2651	0.2780	0.1294	0.0876	0.0938	0.0538	0.2309	0.2559	0.1094

TAB. 4.5 – Mesures des performances avec différentes méthodes de matching

## Influence des transformations géométriques

Les transformations géométriques induisent un changement de position des pixels et des régions. La rotation est celle qui pose généralement le plus de problèmes. Pour tester son effet sur la recherche nous avons effectué des rotations sur des images requêtes. Le but est de voir quelle est la réponse du système si l'image requête a subi une variation géométrique. Une collection de 10 images a été choisie pour subir des transformations géométriques. Nous avons créé 5 images par rotation de la photo originale de 90° et 5 autres avec une rotation de 180° (figure 4.17). Ces deux angles représentent des valeurs extrêmes permettant de tester les fortes rotations. Les résultats de recherche sont résumés dans le tableau 4.6. Selon les valeurs des MAP, GMAP et RPreC : MCM, GABOR et

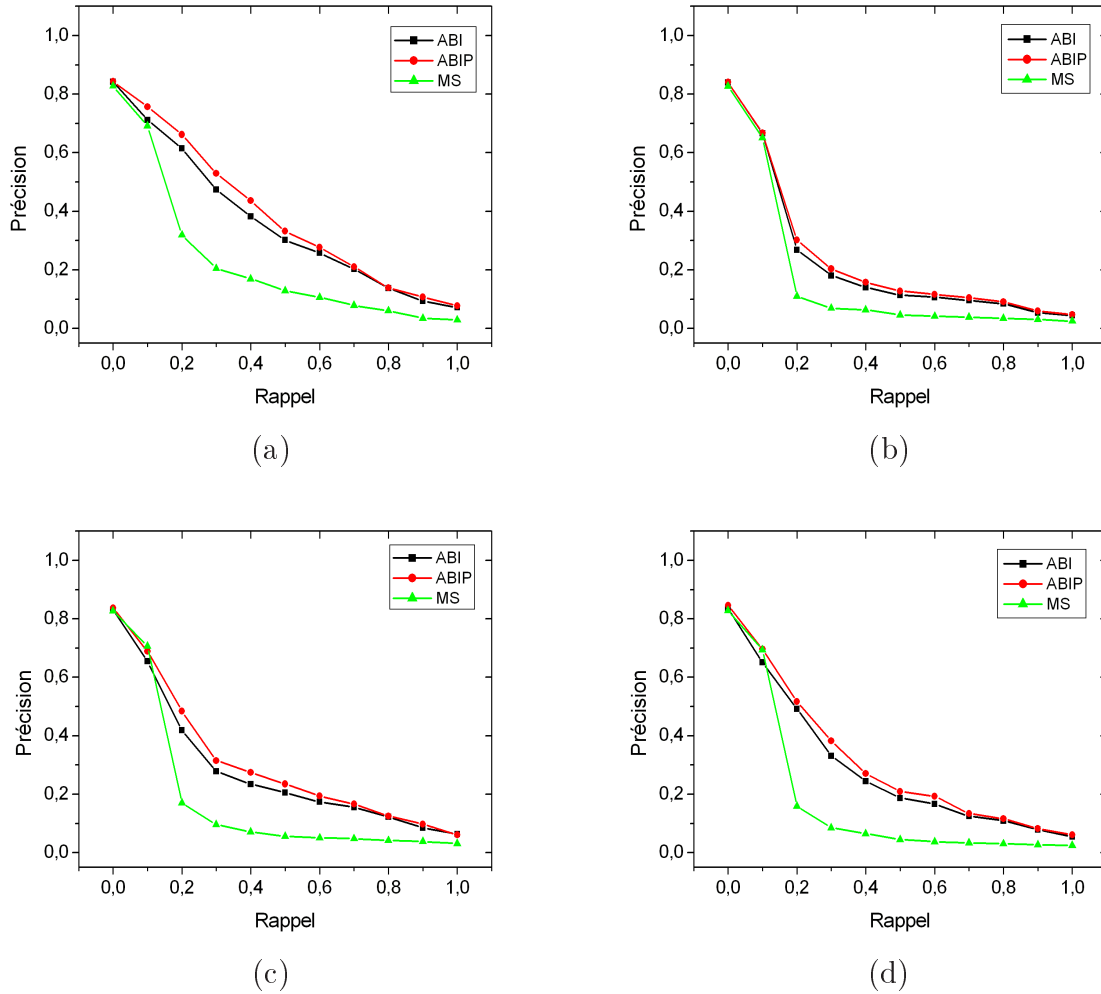


FIG. 4.16 – Comparaison des méthodes d'appariement : (a) MCM (b) GLCM (c) FGVPO (d) GABOR



FGVPO sont les méthodes qui affichent une certaine dégradation des performances dans le cas d'une rotation. En revanche on peut constater une certaine stabilité dans le cas de GLCM. Les courbes rappel-précision de la figure 4.18 indiquent également l'influence de la rotation sur les caractéristiques. Le même comportement a été également constaté pour les textures (chapitre 2). Cette dégradation est dû au changement des relations de voisinage dans les blocs de pixels, ce qui affecte les fréquences spatiales dans l'image.

Statistiques générales	
Nombre de requêtes : 10	
Documents extraits : 4260	
Documents pertinents : 155	
Documents pertinents trouvés : 105	

Méthodes	Avec rotation			Sans rotation		
	MAP	GMAP	RPrec	MAP	GMAP	RPrec
MCM	0.2272	0.0573	0.2260	0.3585	0.1028	0.3149
GLCM	0.1798	0.0594	0.1584	0.1842	0.0606	0.1584
FGVPO	0.2064	0.0429	0.1783	0.2673	0.0743	0.2388
GABOR	0.1633	0.0454	0.1505	0.2595	0.0812	0.2310

TAB. 4.6 – Evaluation de l'efficacité du système sous l'influence de rotation

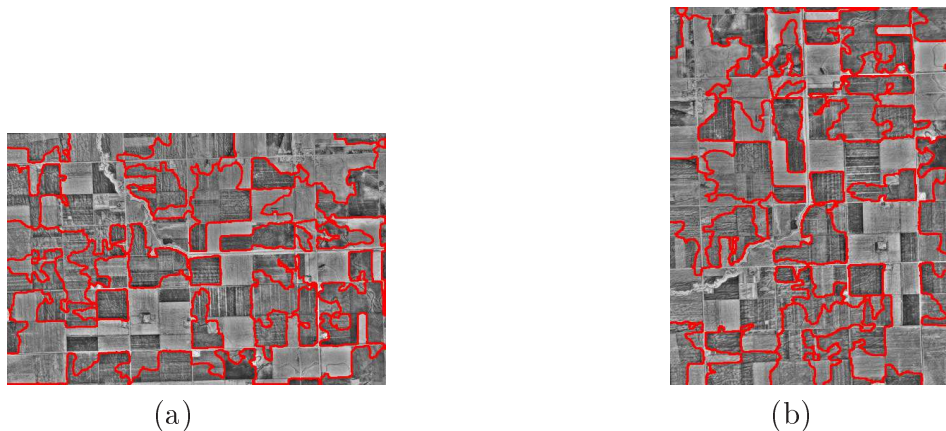
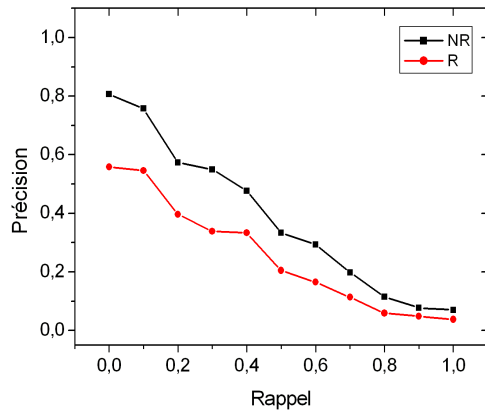
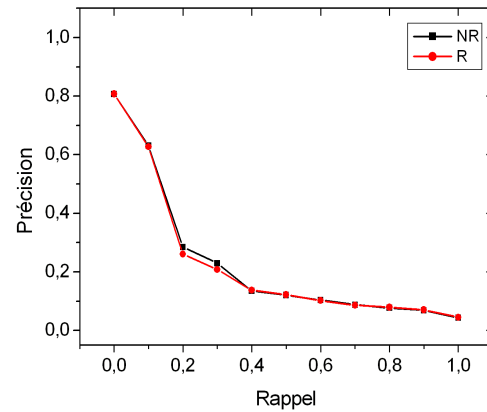


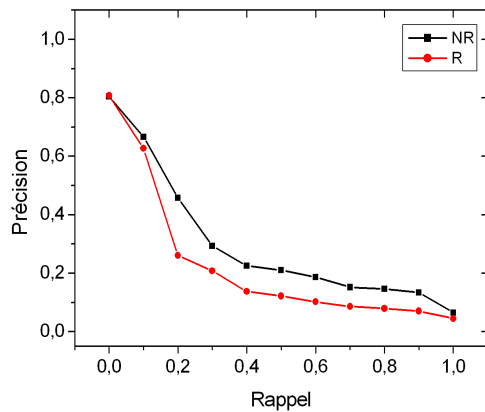
FIG. 4.17 – Exemple de rotation. (a) image originale (b) rotation d'un angle de 90°



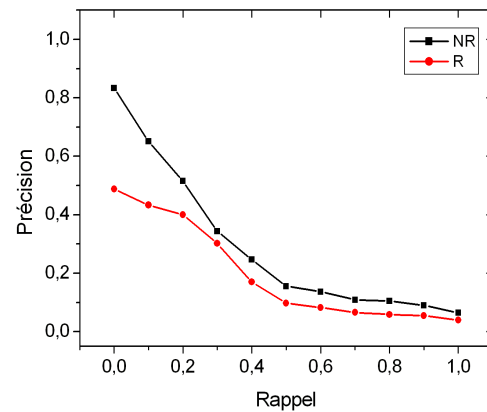
(a)



(b)



(c)



(d)

FIG. 4.18 – Comparaison des performances pour la rotation. La courbe NR : requête sans rotation, la courbe R : requête avec rotation. (a) MCM (b) GLCM (c) FGVPO (d) GABOR

#### 4.10.4 Comparaison entre segmentation humaine et automatique

##### Méthode de différence des surfaces

Le but de ces expériences est d'étudier le fossé sémantique que peut induire la segmentation automatique. Les images de la base ont été segmentées par un humain non expert, déployant un effort raisonnable pour détecter des régions homogènes. La comparaison des performances entre les deux types de segmentation est basée sur la courbe rappel-précision. Nous considérons qu'un résultat de recherche est d'autant meilleur par rapport à un autre que sa courbe rappel-précision est supérieure. La surface entre les deux courbes indique l'écart entre les courbes comme illustré figure 4.19. Plus cette surface est grande plus l'écart des performances est important.

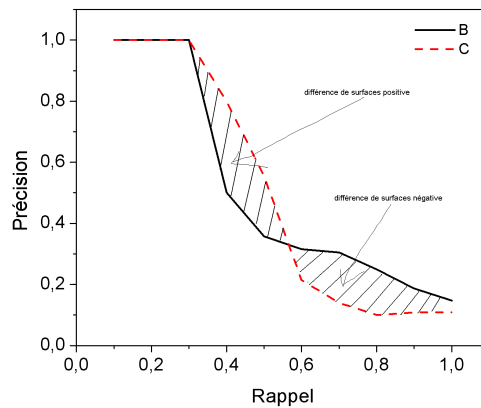


FIG. 4.19 – Différence entre deux courbes rappel/précision

Soit  $C_h$  et  $C_a$  les courbes du rappel-précision produites respectivement par une segmentation humaine et automatique. La différence entre les courbes est donnée par la formule suivante :

$$S_f = \sum_i C_a(i) - C_h(i) \quad (4.20)$$

où  $S_f$  est une surface qui détermine le taux de supériorité d'une courbe. Le signe positif signifie que la segmentation automatique est meilleure et c'est la machine qui gagne. En revanche si le signe est négatif alors c'est l'humain qui gagne.

Nous avons effectué des tests sur une centaine d'image de la base pour à la fois la segmentation automatique et humaine (la caractéristique utilisée dans ce cas est MCM). On a établi un histogramme des surfaces entre courbes. La figure 4.20 est le résultat de cette expérience : l'histogramme a une allure gaussienne. On constate que les performances entre l'homme et la machine sont comparables, l'humain gardant une légère supériorité.

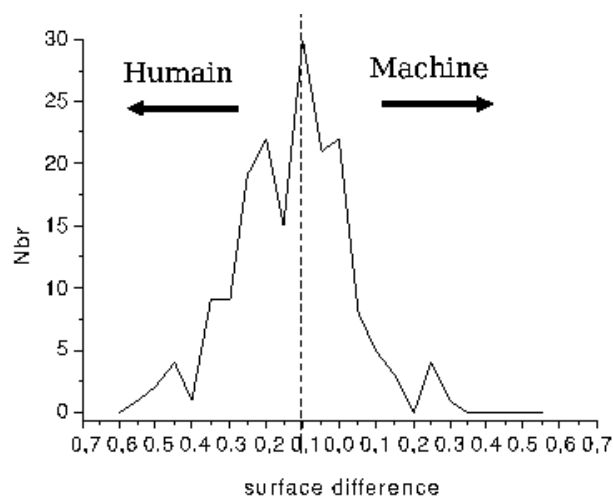


FIG. 4.20 – Comparaison entre l’homme et la machine en terme de segmentation

Statistiques générales	
Nombre de requêtes :	35
Documents extraits :	8260
Documents pertinents :	342
Documents pertinents trouvés :	301

Méthodes	Humain			Machine		
	MAP	GMAP	RPrec	MAP	GMAP	RPrec
MCM	0.5603	0.1682	0.4994	0.3869	0.1163	0.3403
GLCM	0.3047	0.0967	0.2612	0.2404	0.0735	0.2105
FGVPO	0.4939	0.1514	0.4382	0.3237	0.1025	0.2824
GABOR	0.4658	0.1413	0.4311	0.3179	0.0944	0.2794

TAB. 4.7 – Comparaison des performances de segmentation manuelle et automatique

### Méthode Trec Eval

Une deuxième mesure de performances est effectuée à l’aide du logiciel Trec Eval pour obtenir les précisions moyennes du CBIR des deux segmentations. Les mesures obtenues (tableau 4.7) nous montrent que le comportement des méthodes de caractérisation est consistant indépendamment du type de segmentation. Par ailleurs nous remarquons que MAP, GMAP et RPrec sont supérieurs pour une détection de régions par un humain. La dégradation constatée pour une segmentation automatique est dû en fait aux erreurs produites au niveau des régions. La machine considère la région comme une zone homogène en terme de couleur ou texture alors qu’en images satellitaires on trouve souvent des situations plus complexes. L’humain arrive à mieux segmenter car il est opportuniste et utilise à la fois la texture, la couleur, la forme ainsi que des connaissances *à priori*.

Cependant on peut admettre que les résultats fournis par la machine sont acceptables et réalisent un bon compromis.

#### 4.10.5 Recherche avec contraintes spatiales

La requête de l'utilisateur peut être soumise à des contraintes spatiales pour filtrer des situations indésirables. En effet pour plus de précision on peut indiquer au système d'inclure les relations spatiales entre les régions. Les images pertinentes doivent donc répondre aux critères de ressemblance visuelle ainsi qu'à l'arrangement spatial. Prenons l'exemple de la figure 4.14 avec MCM comme caractéristique. Nous imposons au système de rechercher des images qui contiennent zone urbaine, plage et mer dans l'ordre spatial suivant : la ville est à gauche de la plage et de la mer, la plage est à gauche de la mer, la symétrie est également prise en considération. Nous remarquons sur la figure 4.21 l'amélioration des résultats par rapport à la requête sans contrainte spatiale. Nous avons également comparé les performances sur les autres caractéristiques. La figure 4.21 montre l'amélioration apportée par cette contrainte supplémentaire. On remarque également que l'amélioration diffère d'une caractéristique à une autre. En effet dans notre environnement la détection de similarité visuelle est un préalable indispensable pour une bonne détection de similarité spatiale.

Cependant nous avons constaté en multipliant les tests que l'information spatiale n'améliore pas systématiquement le résultat de la recherche. Cela est dû aux imperfections de la segmentation et de la mesure de similarité visuelle. Il est à noter que l'utilisation des relations spatiales dépend de la nature des images de la base ainsi que du choix de l'utilisateur. Si par exemple la base comprend un nombre important d'images contenant les mêmes catégories de régions, l'utilisation des relations spatiales permet de guider la recherche avec davantage de précision.

---

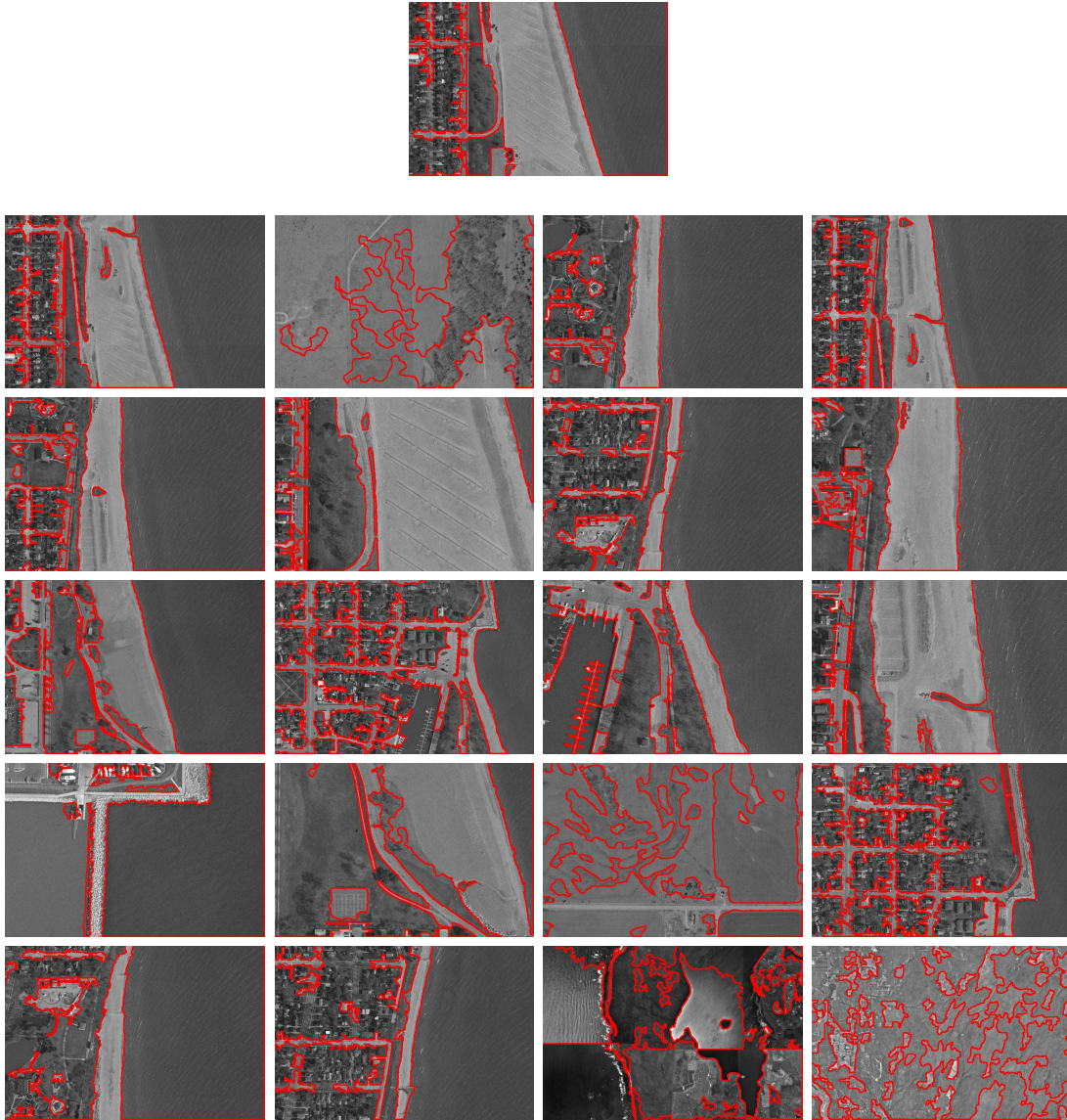


FIG. 4.21 – Un exemple de requête avec relations spatiales



## Conclusion et perspectives

Dans ce travail, nous avons distingué trois aspects fondamentaux de la recherche d'image par le contenu : description, extraction et reconnaissance d'objets. Notre étude s'est concrétisée en la réalisation d'une chaîne de traitement entièrement automatisée mais dédiée à une application relativement restreinte : les images satellitaires. Nous avons ainsi développé de nouvelles méthodes de segmentation ou de reconnaissance d'image en veillant à exploiter et respecter la spécificité des contraintes RIC. Dans cette conclusion, nous présentons d'abord la synthèse des travaux effectués et les leçons tirées, puis nous esquissons en guise d'ouverture sur l'avenir le passage à une application différente : l'aide au recalage d'un véhicule autonome.

Si on doit retrouver une image par son contenu, les caractéristiques extraites par traitement de cette image doivent garantir une discrimination élevée pour des surfaces jugées d'aspect visuel différent par des humains. En effet, la mesure de ressemblance entre deux images dépend directement de ces caractéristiques. Nous avons choisi la texture pour indice discriminant, à cause de sa constante pertinence dans les images aériennes. Pour la modéliser ou la paramétrer nous avons considéré deux de ses tendances reconnues qui prouvent une certaine efficacité en RIC, les caractères statistique et fréquentiel. Nous nous sommes attachés à développer des techniques invariantes aux changements d'éclairage, d'échelle et aux transformations affines. Nous exploitons pour ce faire les parcours de Peano. Ils exhibent des propriétés intéressantes en projetant l'image dans l'espace des variances minimales le long du parcours adapté. Deux types de codage sont proposés : grammatical et numérique. Le premier décrit l'allure du parcours optimal traversant un bloc de 4 pixels par référence à un "motif". La texture est alors mesurée de manière statistique par matrices de co-occurrence des dits motifs. Le deuxième exploite ce même critère d'optimalité mais ici c'est directement la variation totale le long du parcours des quatre voisins que nous retenons. Cette valeur remplace l'intensité en chaque pixel et le spectre de fréquence s'en trouve atténué, même s'il s'agit en réalité d'un gradient, puisque le parcours favorise les plus basses parmi les hautes fréquences. Les filtres de Gabor sont alors appliqués à l'image transformée pour une description fréquentielle de ce compromis. Nous avons montré expérimentalement par des comparaisons menées sur des textures standards l'apport de notre méthode par rapport aux mêmes techniques appliquées directement sur images non transformées. Cette première étude a constitué l'étape d'analyse vers une description des régions d'images de scène réelles. Nous avons également évalué à ce stade les performances de notre description de texture en rappel/précision sur une base

---



d'images segmentée par un humain et les résultats ont confirmé la tendance observée sur les textures isolées.

Le choix de la texture pourrait conduire à considérer l'image dans sa globalité, par exemple la découper en bloc ou lui faire subir une transformée orthogonale globale (e.g. Fourier ou Hadamard) ou sectorielle (e.g. Gabor ou ondelettes). Nous avons choisi de considérer cette image comme une composition d'entités visuelles organisées. La détection de ces entités nous semble en effet un préalable nécessaire à la construction progressive d'une sémantique. Texture impliquant logiquement régions, les descripteurs de bas niveau supporteront la mesure d'une cohérence visuelle i.e. l'homogénéité ou quasi constance, en un sens à définir, des attributs dans une partie de l'image. Nous avons alors eu recours à des algorithmes de segmentation par classification et nous avons choisi le type C-moyennes, très classique et dont les mérites sont bien cernés. Compte tenu de l'incertitude et de l'indétermination des objets dans une image satellitaire (il arrive que des experts différents, e.g. géographes ou climatologues, ne redessinent pas l'image de la même manière!) nous en essayons les versions "flou" (*FCM pour fuzzy c-means*). La cohérence spatiale des régions nous semble critique : nous avons donc particulièrement travaillé la contrainte spatiale imposée à FCM. En effet FCM divise l'espace des points en classes sans tenir compte d'une quelconque corrélation entre pixels voisins. Nous avons donc modifié la fonction objectif de FCM en y introduisant la continuité spatiale. La classification d'un pixel est contrainte par l'appartenance des voisins dans l'image. Cette influence du voisinage permet de réduire les petites régions isolées ou le bruit pour les intégrer dans d'autres plus représentatives. Cependant nous avons constaté que la contrainte spatiale n'est pas suffisante pour détecter les régions d'une manière assez sûre. Nous avons introduit une deuxième contrainte de consistance par le biais d'une analyse en multi échelles. En effet nous construisons une pyramide sur l'image originale par un filtrage itéré qui réduit récursivement la résolution. L'idée est d'influencer la classification de manière descendante : lors de la classification d'un pixel on regarde non seulement ses voisins dans le plan mais encore ses ancêtres dans la pyramide en incorporant leur préférence quant aux classes. La fonction objectif de FCM contient désormais deux termes supplémentaire exerçant deux contraintes, spatiale et pyramidale. Les tests que nous avons effectués sur de nombreuses image de contextes différents ont montré l'efficacité de la méthode comparé aux méthodes FCM plus classiques ou à la segmentation humaine.

Cependant les algorithmes de type FCM nécessitent une intervention humaine pour déterminer préalablement le nombre de classes. Ceci devient une tâche fastidieuse lorsque la base d'images est volumineuse. Pour éviter ce problème nous avons automatisé l'assignation de ce nombre grâce à une mesure adaptée à la technique des C moyennes. Cette mesure exprime la densité et la séparation entre les groupes de points constituant l'image. Les expériences effectuées nous ont permis ici encore de valider l'ensemble de la méthode pour une décomposition de l'image en régions.

Une fois l'image décomposée en régions et celles-ci caractérisées vient la mesure de ressemblance. Nous distinguons trois principes de recherche laissés au choix de l'utilisateur.

---

- Région singulière
- Multiples régions
- Régions et relations spatiales

En ce qui concerne les régions singulières la mesure de similarité se résume en un calcul de distance euclidienne entre les vecteurs caractéristiques respectifs des deux régions. L'image cible comportant un segment proche est considérée similaire. Le deuxième principe suppose que les segments de l'image ne sont pas indépendants et participent collectivement au contexte de la scène. Nous incluons la collection des régions de l'image dans un processus de mise en correspondance qui capte de ce contexte. Nous sommes donc ramenés à un appariement de graphe bipartite dont nous voulons qu'il tienne compte de l'imperfection de la segmentation et la particularité de la recherche d'image par le contenu. Nous avons élaboré une technique de mise en correspondance basée sur un principe de préférence bidirectionnelle. Nous avons considéré que deux images requête et cible sont similaires si leurs entités exhibent globalement une ressemblance mutuelle suffisante. La technique consiste en la construction préalable de la matrice de similarité dont chaque élément représente la distance entre deux régions respectivement requête et cible. On associe par ailleurs à chaque élément d'une partie une liste de préférences où sont triées les régions de la partie opposée par distance croissante. Notre processus de choix mutuel basé sur les minima par ligne et par colonne dans la matrice de similarité permet l'association de un vers plusieurs et bidirectionnel, requête/cible et vis versa. La mesure de ressemblance finale incorpore toutes les distances issues de l'appariement pondérées par l'ordre de préférence. Au delà du groupe de régions que constitue l'image, leur organisation spatiale à l'intérieur de celle-ci peut être déterminante : à notre idée, la recherche implique alors avec profit des relations directionnelles floues entre régions. Dans ce cas deux images sont considérées similaires si leurs composantes sont visuellement proches et montrent deux arrangements spatiaux cohérents l'un avec l'autre. Cela conduit à rajouter dans l'expression de la distance un terme représentatif des différences entre images dans les orientations relatives et les recouvrements de leurs régions.

L'application visée concerne les systèmes d'information géographique contenant des images aériennes et/ou satellitaires. Nous avons testé les performances de notre système en étudiant l'influence de chaque élément de la chaîne sur le processus de recherche. Les résultats obtenus sont satisfaisants au sens de la mesure classique dans le domaine qu'est la dépendance rappel/précision. Pour les améliorer on peut penser incorporer des caractéristiques complémentaires comme les contours ou des points d'intérêt (amers). Cela ne semble pas très prometteur pour ce qui est de l'application qui nous a fortement guidés jusque là, fournissant suffisamment de contraintes pour que nous construisions un système quasi opérationnel. Nous envisageons plutôt de changer d'application dans le but de tester plusieurs limitations du même coup. Nous avons commencé la transposition de notre système à l'extraction de la base de données d'images représentatives de situations ou phases particulières dans la conduite automobile. Côté applicatif le but serait de recalibrer la commande du véhicule lorsqu'il est complètement perdu, soit par suite d'une défaillance de capteur soit à cause d'une ambiguïté particulièrement complexe de sa situation en regard de son intelligence. La référence à des archétypes de scènes introduit alors

---

une salubre expertise. Comme exemple de deux situations différenciées regardons deux groupes d'images correspondant à deux actions distinctes : entrée et sortie d'un espace clos. La série d'images de la figure 4.22 montre une scène prise par le véhicule PICAR. PICAR est une plate forme expérimentale pour la conduite assistée par ordinateur d'un véhicule routier [BFL+03]. Cette plate forme est installée sur un véhicule électrique, elle utilise trois caméras CCD (deux pour vue avant et une sert comme rétroviseur) pour explorer l'environnement extérieur. La vérité terrain indique qu'on se dirige vers un espace clos. En revanche les images de la figure 4.23 représentent des échantillons du chemin de sortie de ce même espace. Nous donnons à titre d'illustration les résultats de segmentation correspondant. Pour obtenir un résultat acceptable il a suffi de changer les valeurs de  $\alpha$  et  $\beta$  dans les expressions de  $u_{ij}$  et de la distance. La segmentation doit encore être améliorée en équilibrant mieux les influences de la texture, contrainte de voisinage et en multirésolution, par rapport à celle du niveau ou de la couleur. Plutôt qu'approfondir cet aspect de l'application déjà étudié dans la thèse, nous nous sommes concentrés sur l'identification de situation.



FIG. 4.22 – Entrée dans un espace clos

On remarque que certains objets sont volatiles et peuvent disparaître de la scène à tout moment, comme les voitures. Mais elles peuvent également passer de gauche à droite, contribuant alors à la vérification du sens de progression. D'autres éléments sont constants et constituent les invariants de la scène. Un but ici pourrait être de collecter plusieurs images prises dans différentes conditions et extraire ces invariants ainsi que leurs propriétés pour construire des images types.

Nous présentons en guise de conclusion expérimentale la recherche d'images comprenant un passage piéton. La galerie des images de ce type contenues dans la base est affichée figure 4.24 associée à leur segmentation. On constate que le passage clouté est détecté dans la majorité des cas même si très partiellement parfois. Nous affichons ensuite les résultats de deux requêtes correspondant à la situation cherchée figure 4.25. Dans chacune les résultats apparaissent très bons, avec toutefois l'apparition d'un étranger (sans



FIG. 4.23 – Sortie d’un espace clos

passage piéton) dans les dix meilleurs, si ce n’est que cet étranger est en fait une image de la même rue de l’Université! On en déduit donc que la requête n’est pas absolument satisfaite grâce au passage piéton, plus ou moins correctement segmenté d’ailleurs, et ceci peut apparaître inquiétant à bien des égards. En revanche on est fondé à conclure des expériences que, dans la réponse, une importance significative est accordée aux relations entre régions attribuées, comme un immeuble en haut gauche de la chaussée. Et ceci est très encourageant. Nous allons donc poursuivre l’analyse des résultats sur plus d’images et de situations pour améliorer notamment l’équilibre entre segmentation et extraction d’image dans le problème considéré.

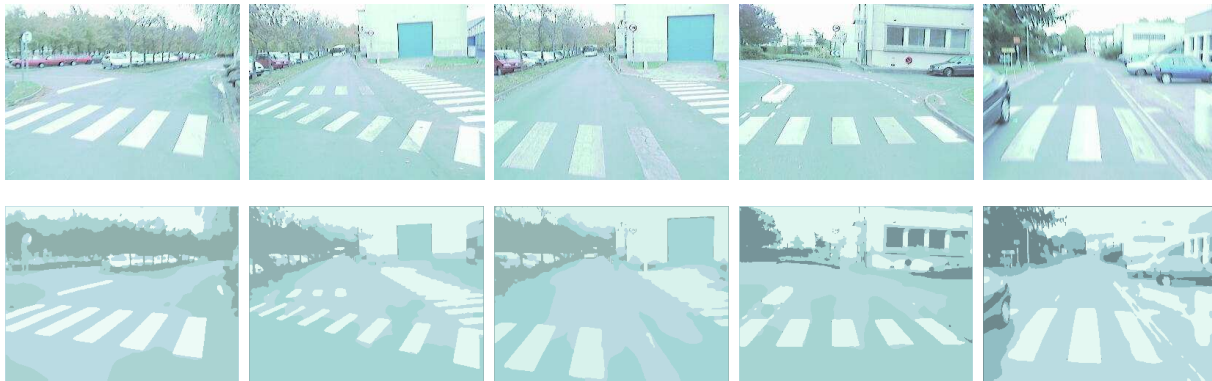


FIG. 4.24 – Galerie d’images contenant des passages piéton avec une segmentation par SPFCM

Plus généralement, côté recherche, le changement radical d’application devrait d’abord contribuer à mesurer et confirmer la généralité de notre approche. Ensuite l’application robotique justifie mieux un essai de nouvelles caractéristiques plus propices au recalage comme les amers mais encore le mouvement ou la profondeur. Cela conduit à rechercher non plus des images isolées mais des ensembles, couples ou séquences, d’images introdui-

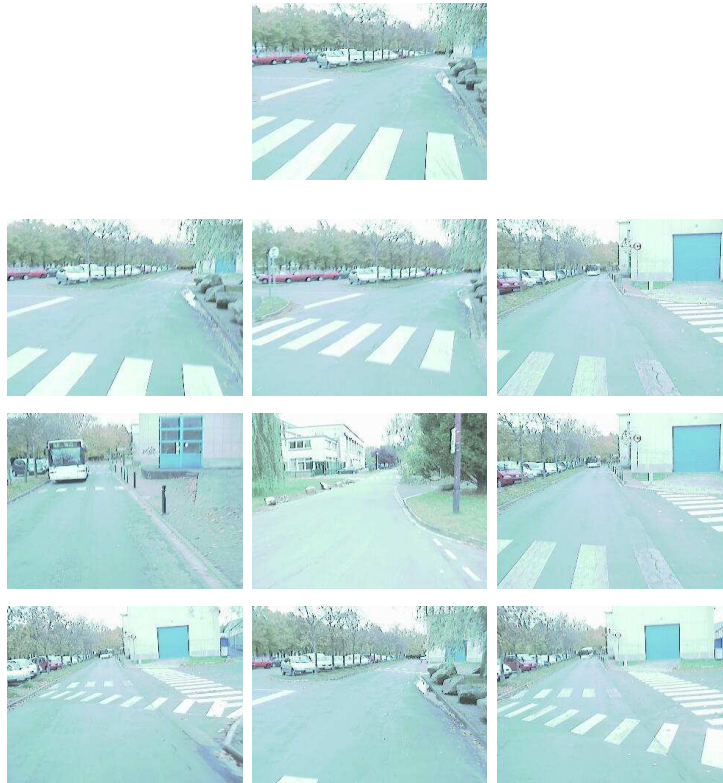


FIG. 4.25 – Un exemple d’une requête contenant un passage piéton en utilisant la technique ABIP. Les résultats sont rangés du droite à gauche et du haut en bas.

sant de nouvelles contraintes propres à rendre le problème traitable. Enfin, même si d’un point de vue automatique la commande par solution stockée n’est pas la plus attractive, elle introduit dans le cas présent une expertise intéressante à étudier sur le plan système. D’autant que l’expertise en question ne doit plus être très éloignée du niveau sémantique de la plupart des requêtes courantes dans l’utilisation banalisée actuelle de la Recherche d’Images par le Contenu. On peut imaginer réussir en quelques années, dans un processus restreint comme la conduite sur route, à associer des commandes perceptives ou actives bien définies à des phases d’une tâche identifiées voire explicitées et donc nommées.

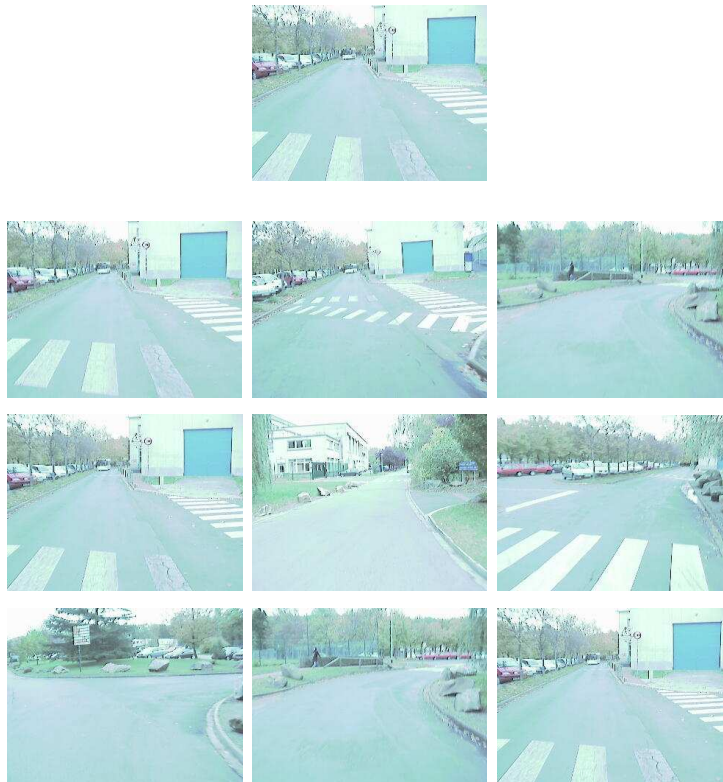


FIG. 4.26 – Deuxième exemple de requête choisie des résultats précédents





# Annex : Interface homme machine

## A.1 Notion

L'interface que nous présentons a été conçue puis réalisée dans le cadre d'un stage d'ingénieur que nous avons encadré. Le but de cette interface est de transformer les différentes tâches de notre système en une application graphique intuitive permettant à un non spécialiste de le manipuler. La conception d'une interface graphique doit répondre à plusieurs critères :

- Le système doit être sous contrôle de l'utilisateur et non l'inverse pour rendre l'application plus attractive et assurer un certain confort.
- L'accès à toutes les fonctionnalités du système doit rester simple pour faciliter l'utilisation et l'interaction homme/machine.
- L'uniformité de la présentation de l'information est essentielle. En ce qui concerne les routines classiques, l'interface doit être conforme aux applications graphiques standard ainsi qu'à leur logique iconographique (par exemple le X pour fermer un fichier).
- L'exploration permet à l'utilisateur de découvrir les différentes fonctions offertes par le système dans un esprit interactif. Cela favorise l'apprentissage au détriment des fausses manoeuvres qui peuvent détruire le système.
- Le confort visuel de l'environnement graphique facilite l'interaction avec le système et contribue à la compréhension des fonctionnalités.

## A.2 Schéma de principe

Afin de répondre aux critères cités ci-dessus l'interface se structure en trois principales parties illustrées par la figure [A.1](#) : le menu, partie gauche requête et commande, zone d'affichage du résultat. Le fond englobe toutes les composantes de l'interface.

Comme nous l'avons indiqué au chapitre 4 l'application graphique a été conçue à l'aide des fonctions GTK de base. Nous détaillons par la suite les différentes parties de l'interface.

---



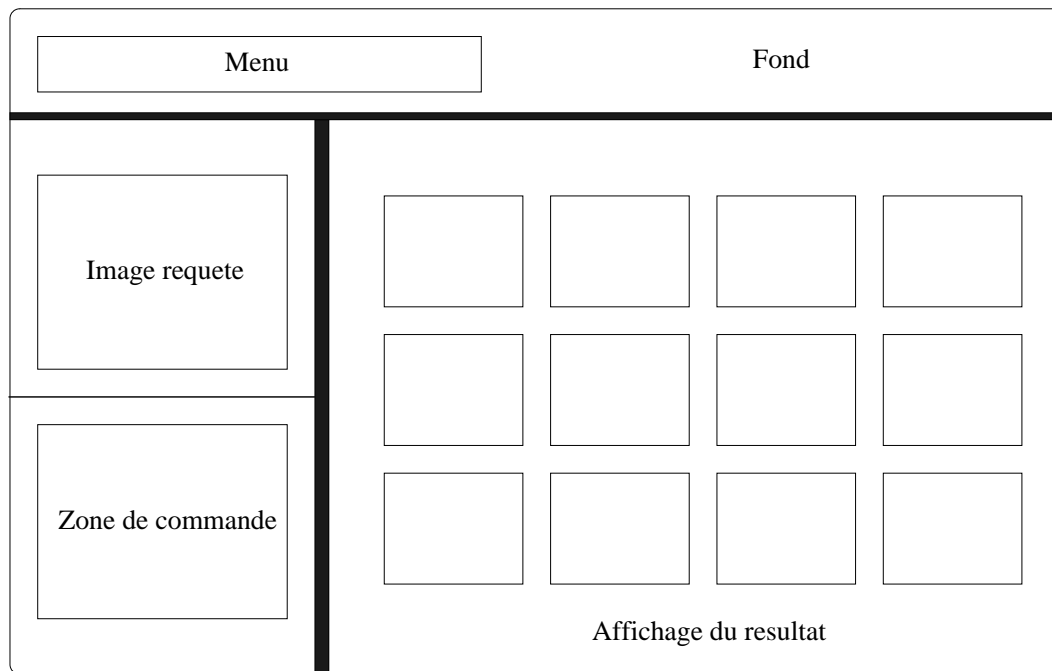


FIG. A.1 – Schema de principe de l'interface graphique

### A.2.1 Barre de menu

Elle contient six rubriques avec un sous-menu pour chacune. Leurs fonctions sont principalement de type organisation de fichiers, préparation des données,... fig. A.2 Ces rubriques ne sont pas impliquées dans le processus de recherche d'image.

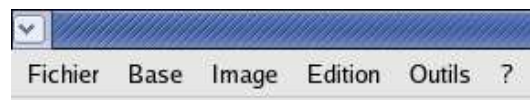


FIG. A.2 – Menu de l'interface

#### Fichier

Vu notre besoin actuel, cette commande est réduite à une simple action qui est la fermeture de l'application. Il est possible de rajouter facilement d'autres champs d'actions.

## Base

Cette rubrique permet à l'utilisateur de charger une base d'images existant sur le disque dur. Un sous menu apparaît en cliquant sur cette rubrique comme le montre la figure A.4. Les images acceptées sont de type jpg ou bmp. Le chargement d'une base se déroule en 3 étapes :

- Activation du sélecteur de répertoire
- Sélection du répertoire contenant les images soit en parcourant l'arborescence soit en saisissant le chemin absolu vers la base dans la case d'en dessous.
- Une fois la base choisie les premières images de celle ci seront affichées dans la zone résultat permettant ainsi à l'utilisateur de naviguer sur toute la base.

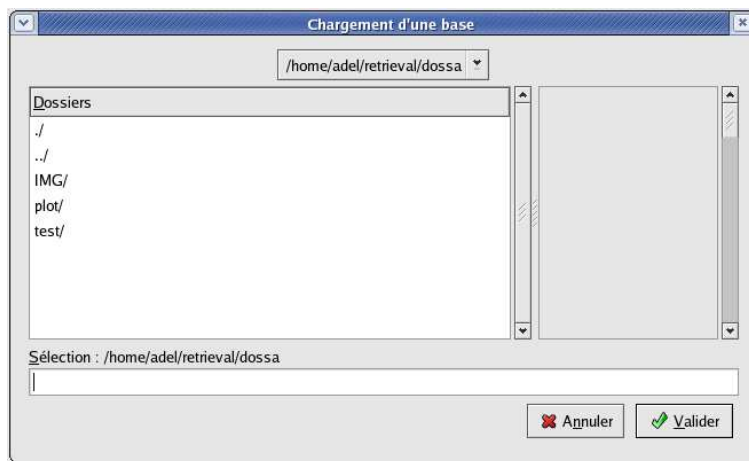


FIG. A.3 – Selecteur de répertoire

**Remarque 4** *Chaque base contient les images originales et segmentées.*

## Image

A travers cette commande on peut choisir une image requête et la visualiser dans la partie gauche de l'interface. Un selecteur de fichier apparaîtra en cliquant sur cette commande, puis une navigation classique sur l'arborescence.

## Edit

Ici on dispose de deux sous menus espace de couleur (non utilisé dans cette version) et taille d'affichage. Cette dernière détermine la taille de la matrice d'affichage dans la

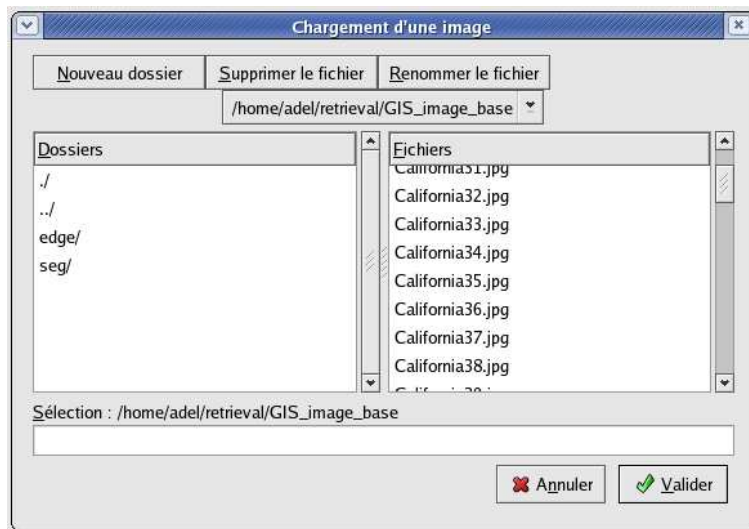


FIG. A.4 – Selecteur de fichier

partie droite voir figure A.4. Les images sont ordonnées de gauche à droite sur les lignes et du haut en bas sur les colonnes. Pour une matrice  $8 \times 8$  la première image est l'élément (1,1) et la 64<sup>ème</sup> se trouve dans (8,8).



FIG. A.5 – Définition du nombre d'images à afficher en lignes/colonnes

## Outils

Les taches du menu "Outils" sont consacrées à l'étude des performances du système en utilisant le critère du rappel/précision. Pour cela l'utilisateur doit introduire les images pertinentes. Il y a deux façons de le faire, en cliquant sur le sous menu "Choix des images pertinentes" nous obtenons un fichier editable. On peut choisir les documents pertinents en cliquant directement sur les images dans la zone d'affichage du résultat ou tout simplement on les saisit dans le fichier. A chaque action sur l'image un chemin absolu vers celle ci est introduit dans le fichier des images pertinentes fig. A.6.

Une fois que les images pertinentes sont chargées dans le fichier approprié, l'exécution de la procédure de recherche produit des résultats qui sont bien sûr affichés mais aussi triés et stockés dans un fichier. Le rappel et la précision sont alors obtenus en comparant les

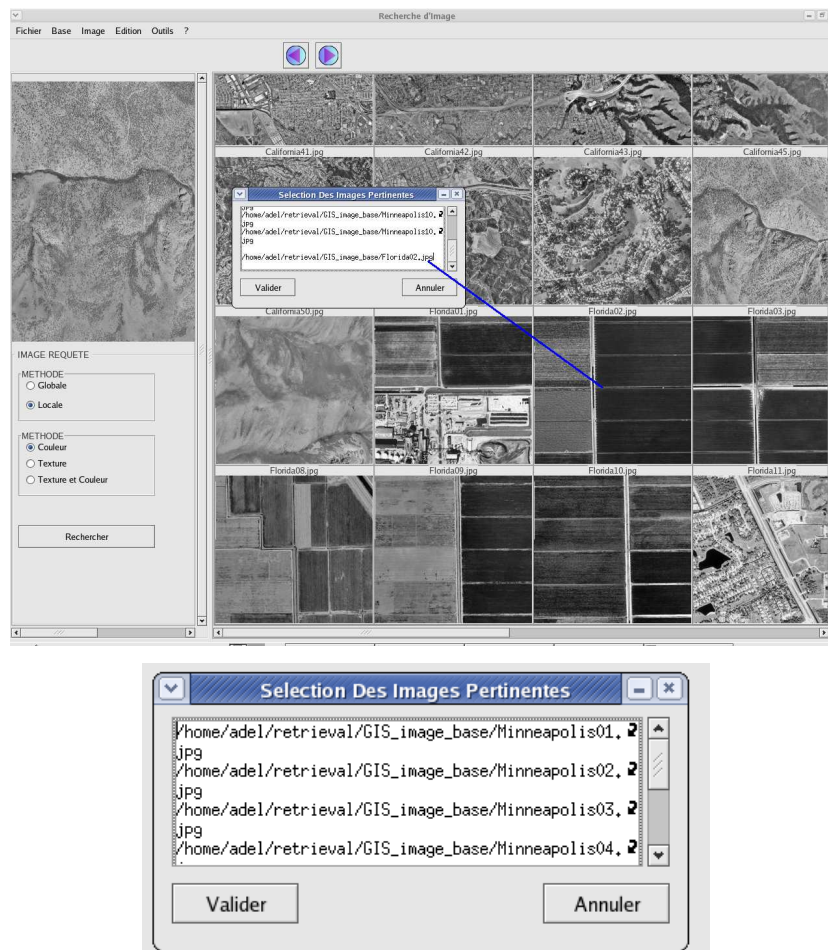


FIG. A.6 – Choix des images pertinentes

résultats et les pertinences des deux fichiers selon la formule décrite au chapitre 4. Enfin le rappel/précision est affiché sous la forme d'une courbe dans une fenêtre séparée d'où la possibilité de sauvegarder le résultat en format Post Script (illustration en figure A.7).

### A.2.2 Paramétrage et recherche

Dans la partie gauche de l'interface on trouve les commandes qui gèrent et agissent d'une manière directe sur le déroulement de la recherche. On dispose de multiples choix sur les attributs, les caractéristiques, la mesure de similarité (voir figure A.8. Nous résumons les différentes actions dans les points suivants :

- Approche Globale : dans ce cas les caractéristiques sont calculées sur l'image entière et par conséquent ça ne nécessite pas une segmentation
- Approche Locale : basée sur les régions on y trouve les trois possibilités déjà expliquées : simple région, multiple régions et relations spatiales. En ce qui concerne la recherche par

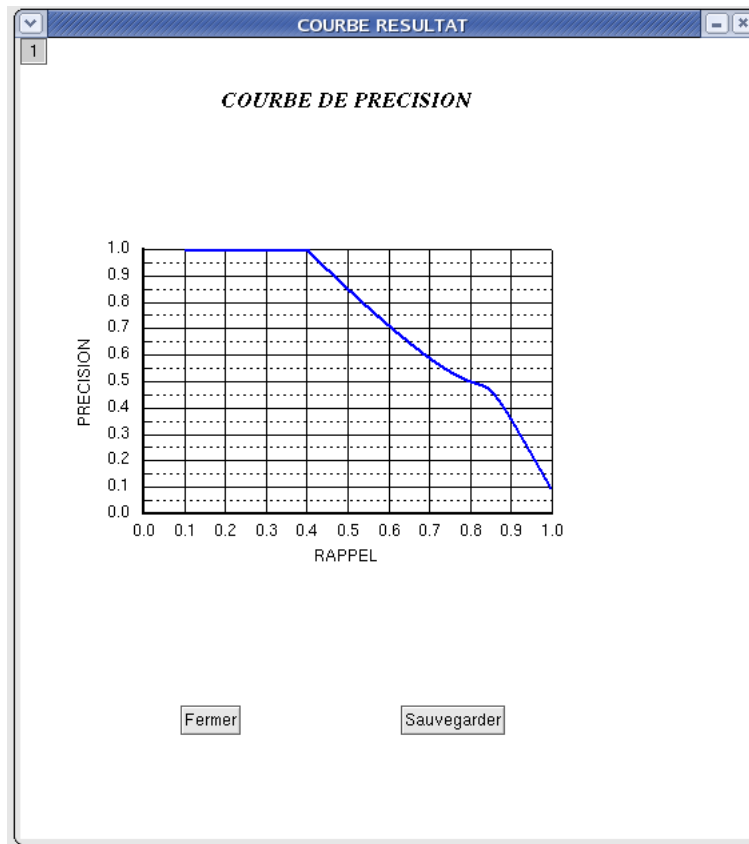


FIG. A.7 – La courbe rappel/precision

simple région on peut sélectionner une région dans l'image requête en cliquant dessus.

- Methodes : Consiste en le choix des attributs couleur, texture ou une mixture entre les deux. Les méthodes de couleur ne sont pas encore implantées.
- Texture : la possibilité de choisir une méthode parmi les quatre étudiées dans cette thèse.

Cette plate forme a été destinée essentiellement à faciliter notre études mais elle supporte l'intégration de plusieurs autres algorithmes.

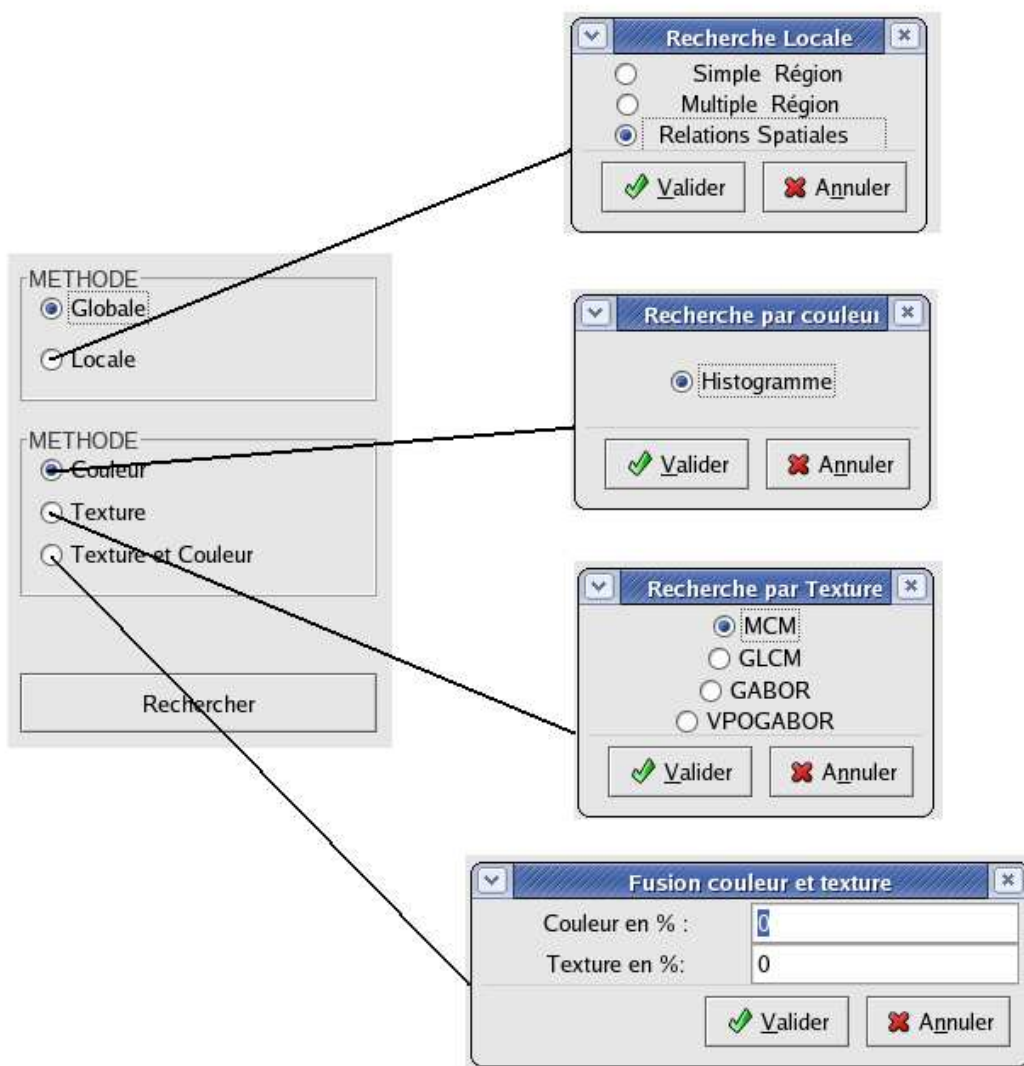


FIG. A.8 – Exemple de différents paramètres du système



---

# Bibliographie

- [Abu89] A. S. Abutableb. Automatic thresholding of gray-level pictures using two-dimensional entropy. *Comput. Vision Graph. Image Process*, 47(1) :22–32, 1989. [27](#)
- [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. [23](#)
- [AJ01] A. Al-Janobi. Performance evaluation of cross-diagonal texture matrix method of texture analysis. *Pattern Recognition*, 34(1) :171–180, 2001. [11](#)
- [AK88] M. Amadasun and R. A. King. Low-level segmentation of multispectral images via agglomerative clustering of uniform neighbourhoods. *Pattern Recogn.*, 21(3) :261–268, 1988. [31](#)
- [ARR<sup>+</sup>97] T. Asano, D. Ranjan, T. Roos, E. Weizl, and P. Windmayer. Space filling curves and their use in the design of geometric data structures. *Journal of Theoretical Computer Science*, 181(1) :3–15, 1997. [48](#)
- [AYMF99] M. N. Ahmed, S. M. Yamany, N. A. Mohamed, and A. A. Farag. A modified fuzzy c-means algorithm for mri bias field estimation and adaptive segmentation. In *MICCAI '99 : Proceedings of the Second International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 72–81. Springer-Verlag, 1999. [36](#), [75](#)
- [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. [12](#)
- [Bar98] T. Barclay. The microsoft terraserver tm. Technical report, Technical Report MSR-TR-98-17, Microsoft Research Microsoft Corporation One Microsoft Way Redmond, WA 98052, 1998. [120](#)
- [BB99] A. Baraldi and P. Blonda. A survey of fuzzy clustering algorithms for pattern recognition. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 29(6) :786–801, 1999. [28](#)
- [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. [13](#)
-



- 
- [Bez74] J. C. Bezdek. Cluster validity with fuzzy sets. *J. Cybernet*, 3 :58–73, 1974. [37](#), [38](#)
- [Bez81] J.C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981. [34](#), [38](#), [74](#)
- [BF84] B. Bhanu and O. Faugeras. Shape matching of two-dimensional objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(2) :137–156, 1984. [23](#)
- [BFL<sup>+</sup>03] S. Bouaziz, M. Fan, A. Lambert, T. Maurin, and R. Reynaud. Picar : experimental palatform for road tracking applications. In *IEEE Intelligent Vehicles Symposium*, Columbus, OH, USA, 2003. [138](#)
- [BGC04] T. Barclay, J. Gray, and W. Chong. Terraserver bricks - a high availability cluster alternative. Technical report, Technical Report MSR-TR-2004-107, Microsoft Research Microsoft Corporation One Microsoft Way Redmond, WA 98052, 2004. [120](#)
- [BH67] G. Ball and D. Hall. A clustering technique for summarizing multivariate data. *Behavioral Science*, 12(2) :153–155, 1967. [32](#)
- [BHBC96] A. M. Bensaid, L. O. Hall, J. C. Bezdek, and L. P. Clarke. Partially supervised clustering for image segmentation. *Pattern Recognition*, 29(5) :859–871, May 1996. [37](#)
- [BKS90] J. M. H. Buf, M. Kardan, and M. Spann. Texture feature performance for image segmentation. *Pattern Recognition*, 23(3) :291–309, 1990. [14](#)
- [Blo05] I. Bloch. Fuzzy spatial relationships for image processing and interpretation : a review. *Image Vision Comput*, 23(2) :89–110, 2005. [105](#)
- [CB87] M. Clark and A. C. Bovik. Texture segmentation using gabor modulation/demodulation. *Pattern Recognition Letters*, 6(4) :261–267, 1987. [13](#)
- [CBGM02] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld : Image segmentation using expectation-maximization and its application to image querying. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(8) :1026–1038, 2002. [72](#), [75](#)
- [CC85] R. Chellappa and S. Chatterjee. Classification of textures using gaussian markov random fields. *IEEE Trans. Acoustics Speech Signal Process*, 33 :959–963, August 1985. [12](#)
- [CC96] C. C. Chen and D. C. Chen. Multi-resolutional gabor filter in texture analysis. *Pattern Recognition Letters*, 17(10) :1069–1076, 1996. [13](#)
- [CC99] C. C. Chen and D. C. Chen. Filtering methods for texture discrimination. *Pattern Recognition Letters*, 20(8) :783–790, 1999. [14](#)
- [CCM<sup>+</sup>97] S. F. Chang, W. Chen, H. J. Meng, H. Sundaram, and D. Zhong. Videoq : An automated content based video search system using visual cues. In *The Fifth ACM International Multimedia Conference*, pages 313–324, 1997. [10](#)
- [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. [12](#)
-

- 
- [CH80] R. W. Connors and C. A. Harlow. A theoretical comparison of texture algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(3) :204–222, 1980. [13](#), [46](#)
- [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. [12](#)
- [CJ93] R. Chellappa and A. K. Jain. *Markov Random Fields : Theory and Applications*. Academic Press, 1993. [12](#)
- [CJL89] S. K. Chang, E. Jungert, and Y. Li. Representation and retrieval of symbolic pictures using generalized 2d string. In *In : SPIE Proceedings on Visual Communications and Image Processing*, pages 1360–1372, 1989. [15](#)
- [CJS<sup>+</sup>01] H.D. Cheng, X. H. Jiang, Y. Sun, , and J. Wang. Color image segmentation : advances and prospects. *Pattern Recognition*, 34(12) :2259–2281, 2001. [72](#)
- [CK85] R. Chellappa and R.L. Kashyap. Texture synthesis using 2-d noncausal autoregressive models. *IEEE Trans. Acoustics Speech Signal Process*, 33 :194–203, February 1985. [12](#)
- [CL91] C. C. Chang and S. Y. Lee. Retrieval of similar pictures on pictorial databases. *Pattern Recogn.*, 24(7) :675–681, 1991. [102](#)
- [CLOP01] L. Cinque, S. Levialdi, K. A. Olsen, and A. Pellicano. Color-based image retrieval using spatial-chromatic histograms. *Image Vision Comput.*, 19(13) :979–986, 2001. [8](#)
- [CMOY96] I. J. Cox, M. L. Miller, S. M. Omohundro, and P. N. Yianilos. Pichunter : Bayesian relevance feedback for image retrieval. In *In Proceedings of International Conference on Pattern Recognition*, pages 361–369, 1996. [17](#), [26](#)
- [CPH97] H. Caillol, W. Pieczynski, and A. Hillon. Estimation of fuzzy gaussian mixture and unsupervised statistical image segmentation. *Transactions on Image Processing*, 6(3) :425–440, 1997. [34](#)
- [CS01] G. Ciocca and R. Schettini. Content-based similarity retrieval of trademarks using relevance feedback. *Pattern Recognition*, 34(8) :103–199, 2001. [17](#)
- [CSY87] S. K. Chang, Q. Y. Shi, and C. W. Yan. Iconic indexing by 2-d strings. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(3) :413–428, 1987. [15](#)
- [CTB<sup>+</sup>99] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik. Blobworld : A system for region-based image indexing and retrieval. In *VISUAL '99 : Proceedings of the Third International Conference on Visual Information and Information Systems*, pages 509–516, London, UK, 1999. Springer-Verlag. [18](#), [26](#), [70](#), [117](#), [121](#)
- [CTH84] R. W. Connors, M. M. Trivedi, and C. A Harlow. Segmentation of a high-resolution urban scene using texture operators. *Computer Vision, Graphics and Image Processing*, 25(3) :273–310, March 1984. [46](#)
-

- 
- [CZ04] S. Chen and D. Zhang. Robust image segmentation using fcm with spatial constraints based on new kernel-induced distance measure. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 34(4) :1907–1916, 2004. [36](#)
- [Dav96] R. N. Dave. Validating fuzzy partitions obtained through c-shells clustering. *Pattern Recogn. Lett.*, 17(6) :613–623, 1996. [38](#)
- [Dav97] E. R. Davies. *Machine Vision : Theory, Algorithms, Practicalities*. Academic Press, San Diego, 1997. [15](#)
- [DCA81] L. S. Davis, M. Clearman, and J. K. Aggarwal. An empirical evaluation of generalized cooccurrence matrices. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 3(2) :214–221, 1981. [11](#)
- [DG86] S. R. Dubois and F. H. Glanz. An autoregressive model approach to two-dimensional shape classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(1), 1986. [15](#)
- [DH95] D. Dunn and W. Higgins. Optimal gabor filters for texture segmentation. *IEEE Trans. on Image Processing*, 4(7) :947–964, 1995. [13](#)
- [DHS00] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000. [29](#), [30](#), [74](#)
- [DHW94] D. Dunn, W. Higgins, and J. Weakley. Texture segmentation using 2-d gabor elementary functions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(2) :130–149, 1994. [13](#)
- [Did73] E. Diday. The dynamic clusters method and optimization in non-hierarchical clustering. *Optimization Techniques*, pages 241–258, 1973. [32](#)
- [DJA79] L. S. Davis, S. A. Johns, and J. K. Aggarwal. Texture analysis using generalized co-occurrence matrices. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1(3) :251–259, 1979. [11](#)
- [DK97] R. N. Dave and R. Krishnapuram. Robust clustering methods : A unified view. *IEEE Trans. Fuzzy Systems*, 5(2) :270–293, 1997. [28](#)
- [DLR76] A. P. Dempster, N. M. Laird, , and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm(with discussion). *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1) :1–38, 1976. [33](#), [72](#)
- [DMP97] Y. Delignon, A. Marzouki, and W. Pieczynski. Estimation of generalized mixtures and its application in image segmentation. *IEEE Trans. On Image Processing*, 6(10) :1364–1375, 1997. [34](#)
- [DP88] D. Dubois and H. Prade. *Possibility theory :An Approach to Computerized Processing of Uncertainty*. Plenum Press, New York, 1988. [35](#)
- [dS01] J. P. Marques de Sa. *Pattern Recognition : Concepts, Methods and Applications*. Springer-Verlag, 2001. [29](#), [30](#)
- [Dub87] R. C. Dubes. How many clusters are best an experiment. *Pattern Recognition*, 20(6) :645–663, 1987. [31](#)
-

- 
- [EF91] M. Egenhofer and R. Franzosa. Point-set topological spatial relations. *International Journal of Geographical Information Systems*, 5(2) :161–174, 1991. [103](#)
- [EKK99] E. A. El-Kwae and M. R. Kabuka. A robust framework for content-based retrieval by spatial similarity in image databases. *ACM Trans. Inf. Syst.*, 17(2) :174–198, 1999. [16](#)
- [EP94] I. M. Elfadel and R. W. Picard. Gibbs random fields, cooccurrences, and texture modeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(1) :24–37, 1994. [11](#)
- [Eve93] B. S. Everitt. *Cluster Analysis*. Edward Arnold and Halsted Press, 1993. [29](#), [30](#)
- [FB04] J. Fauqueur and N. Boujemaa. Region-based image retrieval : fast coarse segmentation and fine color description. *Journal of Visual Languages & Computing*, 15(1) :69–95, 2004. [72](#)
- [FD77] H. Freeman and L. S. Davis. A corner-finding algorithm for chain-coded curves. *IEEE Trans. Computers*, 26(3) :297–303, 1977. [14](#)
- [FD97] J. K. Fwu and P. M. Djuric. Em algorithm for image segmentation initialized by a tree structure. *IEEE Trans. Image Processing*, 6(2) :349–352, 1997. [34](#)
- [FK97] H. Frigui and R. Krishnapuram. Clustering by competitive agglomeration. *Pattern Recognition*, 30(7) :1109–1119, 1997. [35](#)
- [FM81] K. S. Fu and J. K. Mui. A survey on image segmentation. *Pattern Recognition*, 13(1) :3–16, 1981. [28](#), [71](#)
- [FP80] O. D. Faugeras and W. K. Pratt. Decorrelation methods of texture feature extraction. *Transactions on Pattern Analysis and Machine Intelligence*, 2(4) :323–333, 1980. [13](#)
- [FR89] C. Faluostos and S. Roseman. Fractals for secondary key retrieval. In *Proceeding of the ACM Conference on principles of database system*, pages 247–252, 1989. [48](#)
- [FS89a] I. Fogel and D. Sagi. Gabor filters as texture discriminator. *Journal of Biological Cybernetics*, 61 :102–113, 1989. [48](#)
- [FS89b] Y. Fukuyama and M. Sugeno. A new method of choosing the number of clusters for the fuzzy c-means method. In *Proceedings of 5th Fuzzy Systems Symposium*, pages 247–250, 1989. [39](#), [83](#)
- [FSN<sup>+</sup>95] M. Flickner, 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*, 28(9) :23–32, 1995. [10](#), [18](#), [25](#)
- [Fu82] K. S. Fu. *Syntactic Pattern Recognition and Applications*. Prentice-Hall, New Jersey, 1982. [12](#)
- [Fuk90] K. Fukunaga. *Introduction to statistical pattern recognition (2nd ed.)*. Academic Press Professional, Inc., San Diego, CA, USA, 1990. [28](#)
-

- 
- [GG89] I. Gath and A. B. Geva. Unsupervised optimal fuzzy clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(7) :773–780, 1989. [39](#), [83](#)
- [GI89] D. Gusfield and R. W. Irving. *The stable marriage problem : structure and algorithms*. MIT Press, Cambridge, MA, USA, 1989. [110](#)
- [GK90] C. C. Gotlieb and H. E. Kreyzig. Texture descriptors based on co-occurrence matrices. *Comput. Vision Graph. Image Process.*, 51(1) :70–86, 1990. [11](#)
- [GMD<sup>+</sup>97] P. Gros, G. Mclean, R. Delon, R. Mohr, C. Schmid, and G. Mistler. Utilisation de la couleur pour l'appariement et l'indexation d'images. Technical Report RR-3269, Technical report, INRIA, 1997. [8](#)
- [GR95] V. N. Gudivada and V. V. Raghavan. Design and evaluation of algorithms for image retrieval by spatial similarity. *ACM Trans. Inf. Syst.*, 13(2) :115–144, 1995. [8](#)
- [GR04] G. Giacinto and F. Roli. Bayesian relevance feedback for content-based image retrieval. *Pattern Recognition*, 37(7) :1499–1508, 2004. [17](#)
- [GS62] D. Gale and L.S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69 :9–15, 1962. [22](#)
- [GS97] I. Gagliardi and R. Schettini. A method for the automatic indexing of color images for effective image retrieval. *The New Review of Hypermedia and Multimedia*, 3 :201–224, 1997. [8](#)
- [GS99] T. Gevers and A. W. M. Smeulders. Color based object recognition. *Pattern Recognition*, 32(3) :453–464, 1999. [8](#)
- [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. [16](#)
- [GvdBSG01] J. M. Geusebroek, R. van den Boomgaard, A. W. M. Smeulders, and H. Geerts. Color invariance. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(12) :1338–1350, 2001. [8](#)
- [GW02] R. C. Gonzales and R. E. Woods. *Digital image processing*. Prentice-Hall, New Jersey, 2002. [8](#), [14](#), [27](#), [28](#)
- [HBV01] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *J. Intell. Inf. Syst.*, 17(2-3) :107–145, 2001. [83](#)
- [HKKZ95] T. Hermes, C. Klauck, J. Kreyß, and J. Zhang. Image retrieval for information systems. In *SPIE, Conference on Storage and Retrieval for Image and Video Databases*, 1995. [15](#)
- [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. [13](#)
- [HP76] S. L. Horowitz and T. Pavlidis. Picture segmentation by a tree traversal algorithm. *J. ACM*, 23(2) :368–388, 1976. [71](#)
-

- 
- [HS85] R. M. Haralick and L. G. Shapiro. Image segmentation techniques. *CVGIP*, 29(1) :100–132, 1985. [28](#), [71](#)
- [HS92] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision*. Addison-Wesley, 1992. [59](#)
- [HSD73] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Trans. Sys. Man. Cybern*, 3(6) :610–621, 1973. [11](#), [46](#)
- [HSE95] J. Hafner, H. S. Sawhney, and W. Equitz. Efficient color histogram indexing for quadratic form distance functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7) :729–736, 1995. [18](#)
- [Hu62] M. K. Hu. Visual pattern recognition by moment invariants, computer methods in image analysis. *Transactions on Information Theory*, 8(2) :179–187, 1962. [15](#)
- [HW92] D. C. He and L. Wang. Unsupervised textural classification of images using the texture spectrum. *Pattern Recognition*, 25(3) :247–255, 1992. [13](#)
- [Jai93] R. Jain. Nsf workshop on visual information management systems. In *SPIE Storage and Retrieval for Image and Video Databases*, 1993. [17](#)
- [JB91] X. Y. Jiang and H. Bunke. Simple and fast computation of moments. *Pattern recognition*, 24(8) :801–806, 1991. [15](#)
- [JB04] A. P. Johnson and Jr. C. L. Baker. First- and second-order information in natural images : A filter-based approach to image statistics. *Journal of the Optical Society of America*, 21(6) :613–925, 2004. [11](#)
- [JCSZ02] N. Jhanwar, S. Chaudhuri, G. Seetharaman, and B. Zavidovique. Content based image retrieval using optimal peano scans. In *The Sixteenth International Conference on Pattern Recognition. ICPR 2002*, 2002. [49](#)
- [JCSZ04] N. Jhanwar, S. Chaudhuri, G. Seetharaman, and B. Zavidovique. Content based image retrieval using motif cooccurrence matrix. *Image and Vision Computing*, 22(14) :1211–1220, 2004. [50](#), [62](#)
- [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. [13](#)
- [JD88] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, 1988. [29](#), [30](#), [34](#)
- [JDM00] A. K. Jain, Robert P. W. Duin, and J. Mao. Statistical pattern recognition : A review. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(1) :4–37, 2000. [28](#)
- [JKS95] R. Jain, R. Kasturi, and B. G. Schunck. *Machine vision*. McGraw-Hill, Inc., New York, NY, USA, 1995. [28](#)
- [JMF99] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering : a review. *ACM Comput. Surv*, 31(3) :264–323, 1999. [28](#), [29](#), [30](#), [32](#), [33](#), [38](#)
- [Joh67] S. C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 2 :241–254, 1967. [30](#)
-

- 
- [JPP95] R. Jain, A. Pentland, and D. Petkovic. Nsf-arpa workshop on visual information management systems. In *Cambridge, MA*, 1995. 17
- [JW77] K. S. Jones and P. Willett. *Readings in Information Retrieval*. Morgan Kaufmann Publishers, 1977. 6
- [KBT01] L. Kam and J. Blanc-Talon. *Multifractal multipermuted multinomial measures for texture characterization and segmentation*. Nova Science Publishers, Inc., Commack, NY, USA, 2001. 13
- [KC81] R. L. Kasyap and R. Chellappa. Stochastic models for closed boundary analysis : Representation and reconstruction. *IEEE Trans. Inform. Theory*, 27(5) :627–637, 1981. 15
- [KC92] A. Kundu and J. L. Chen. Texture classification using qmf bank-based subband decomposition. *Graphical Models and Image Processing*, 54(5) :369–384, 1992. 13
- [KC97] S. Krishnamachari and R. Chellappa. Multiresolution gauss-markov random field models for texture segmentation. *IEEE Trans. Image Process*, 6(2) :251–267, 1997. 12
- [KC03] J. W. Kwak and N. I. Cho. Relevance feedback in content-based image retrieval system by selective region growing in the feature space. *Signal Processing : Image Communication*, 18(9) :787–799, 2003. 17
- [KJ03] I. King and Z. Jin. Integrated probability function and its application to content-based image retrieval by relevance feedback. *Pattern Recognition*, 36(9) :2177–2186, 2003. 17
- [KK93] R. Krishnapuram and J. M. Keller. A possibilistic approach to clustering. *IEEE Transactions on Fuzzy Systems*, 1(2) :98–110, 1993. 35
- [KK96] R. Krishnapuram and J. M. Keller. The possibilistic c-means algorithm : insights and recommendations. *IEEE Transactions on Fuzzy Systems*, 4(3) :385–393, 1996. 35
- [KP96] V. A. Kovalev and M. Petrou. Multidimensional cooccurrence matrices for object recognition and matching. *Graphical Models and Image Processing*, 58(3) :187–197, 1996. 11
- [KR90] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data : An Introduction to Cluster Analysis*. John Wiley & Sons, 1990. 28
- [KSGA96] R. Kasturi, S. H. Strayer, U. Gargi, and S. Antani. An evaluation of color histogram based methods in video indexing. Technical report, Technical Report CSE-96-053, Penn State University, Department of Computer Science and Engineering, 1996. 9
- [Kuh55] H. W. Kuhn. The hungarian method for the assignment problem. *Nay. Res. Log. Quart*, 2 :83–97, 1955. 22
- [Kur91] T. Kurita. An efficient agglomerative clustering algorithm using a heap. *Pattern Recogn.*, 24(3) :205–209, 1991. 31
-

- 
- [KY98] J. Kender and B. Yeo. Video scene segmentation via continuous video coherence. In *IEEE Computer Vision and Pattern Recognition*, pages 367–373, 1998. [10](#)
- [LH90] S. Y. Lee and F. J. Hsu. 2d c-string : a new spatial knowledge representation for image database systems. *Pattern Recogn.*, 23(10) :1077–1087, 1990. [16](#)
- [LH92] S. Y. Lee and F. J. Hsu. Spatial reasoning and similarity retrieval of images using 2d c-string knowledge representation. *Pattern Recogn.*, 25(3) :305–318, 1992. [103](#)
- [Li95] S. Z. Li. *Markov random field in computer vision*. Springer-Verlag, 1995. [76](#)
- [LLL00] A. W. C Liew, S. H. Leung, and W. H. Lau. Fuzzy image clustering incorporating spatial continuity. *Vision, Image and Signal Processing*, 147(2) :185–192, 2000. [37](#)
- [LM01] L. Lucchese and S. K. Mitra. Color image segmentation : A state-of-the-art survey. In *Proceedings of the Indian National Science Academy (INSA-A)*, 2001. [72](#)
- [LW66] G. N. Lance and W. T. Williams. Computer programs for hierarchical polythetic classification. *Comput. J*, 9 :60–64, 1966. [30](#)
- [LWW00] J. Li, J. Z Wang, and G. Wiederhold. IRM : integrated region matching for image retrieval. In *ACM Multimedia*, pages 147–156, 2000. [23](#), [109](#)
- [LYC92] S. Y. Lee, M. C. Yang, and J. W. Chen. 2d b-string knowledge representation and picture retrieval for image database. In *2nd International Computer Science Conference Data and Knowledge Engineering : Theory and Applications*, pages 13–16, 1992. [16](#)
- [Mac67] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967. [32](#)
- [Man82] B. Mandelbrot. *Fractal Geometry of Nature*. W. H. Freeman, 1982. [13](#)
- [MANL03] V. Sridhar M. A. Nascimento and X. Li. Effective and efficient region-based image retrieval. *Journal of Visual Languages & Computing*, 14(2) :151–179, 2003. [72](#)
- [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. [23](#)
- [MBKB02] S. D. MacArthur, C. E. Brodley, A. C. Kak, and L. S. Broderick. Interactive content-based image retrieval using relevance feedback. *Computer Vision and Image Understanding*, 88(2) :55–75, 2002. [17](#)
- [MFTM01] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001. [85](#)
-



- 
- [MJ92] J. Mao and A. K. Jain. Texture classification and segmentation using multi-resolution simultaneous autoregressive models. *Pattern Recogn.*, 25(2) :173–188, 1992. [12](#)
- [MK96] G. J. McLachlan and T. Krishnan. *The EM algorithm and extensions*. Wiley, New York, 1996. [33](#)
- [MKL97] B. M. Mehtre, M. S. Kankanhalli, and W. F Lee. Shape measures for content based image retrieval : a comparison. *Inf. Process. Manage.*, 33(3) :319–337, 1997. [14](#)
- [MM96] W. Y. Ma and B. S. Manjunath. Texture features and learning similarity. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 425–430, 1996. [17](#)
- [MM97] W. Y. Ma and B. S. Manjunath. Edge flow : a framework for boundary detection and image segmentation. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, San Juan, Puerto Rico, June 1997. [17](#), [72](#)
- [MM99] W. Ma and B. S. Manjunath. Netra : A toolbox for navigating large image databases. *Multimedia Syst.*, 7(3) :184–198, 1999. [25](#), [70](#), [72](#)
- [MMS<sup>+</sup>01] H. Muller, W. Muller, D. McG. Squire, S. Marchand-Maillet, and T. Pun. Performance evaluation in content-based image retrieval : overview and proposals. *Pattern Recognition Letters*, 22(5) :593–601, 2001. [116](#)
- [MOP98] V. Murino, C. Ottonello, and S. Pagnan. Noisy texture classification : A higher-order statistics approach. *Pattern Recognition*, 31(4) :383–393, 1998. [11](#)
- [MP90] J. Malik and P. Perona. Preattentive texture discrimination with early vision mechanisms. *Journal of the Optical Society of America*, 7 :923–932, 1990. [13](#)
- [MP96] T. P. Minka and R. W. Picard. Interactive learning using a "society of models". In *IEEE, Conference on Computer Vision and Pattern Recognition*, pages 447–452, 1996. [17](#)
- [MR94] K. Miyajima and A. Ralescu. Spatial organization in 2d images. In *In Proceedings of the Third IEEE Conference on Fuzzy Systems*, 1994. [105](#)
- [NKW97] A. D. Narasimhalu, M. S. Kankanhalli, and J. Wu. Benchmarking multimedia databases. *Multimedia Tools Appl.*, 4(3) :333–356, 1997. [116](#)
- [OD92] P. P. Ohanian and R. C. Dubes. Performance evaluation for four classes of textural features. *Pattern Recognition*, 25(8) :819–833, 1992. [14](#)
- [OMP<sup>+</sup>02] T. Ojala, T. Maenpaa, M. Pietikainen, J. Viertola, J. Kyllonen, and S. Huovinene. Outex - a new framework for empirical evaluation of texture analysis algorithms. In *Proc. 16th Intl. Conf. Pattern Recognition 2002*, 2002. [59](#)
- [OPH96] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29(1) :51–59, 1996. [14](#)
-

- [ORC<sup>+</sup>97] M. Ortega, Y. Rui, K. Chakrabarti, S. Mehrotra, and T. S. Huang. Supporting similarity queries in mars. In *MULTIMEDIA '97 : Proceedings of the fifth ACM international conference on Multimedia*, pages 403–413, 1997. 27
- [Ots79] N. Otsu. A threshold selection method from gray-level histogram. *IEEE Trans. Systems, Man, and Cybernetics*, 6(1) :62–66, 1979. 27
- [Ott91] P. J. Van Otterloo. *A contour-oriented approach to shape analysis*. Prentice Hall International (UK) Ltd., Hertfordshire, UK, UK, 1991. 15
- [Pap92] T. N. Pappas. An adaptive clustering algorithm for image segmentation. *IEEE Trans. on Signal Processing*, 40(4) :901–914, 1992. 32
- [PB95] N. R. Pal and J. C. Bezdek. On cluster validity for the fuzzy c-means model. *IEEE Transactions on Fuzzy Systems*, 3(3) :370–379, 1995. 83
- [PCC<sup>+</sup>98] T. V. Pappathomas, T. E. Conway, I. J. Cox, J. Ghosn, M. L. Miller, T. P. Minka, and P. N. Yianilos. Psychophysical studies of the performance of an image database retrieval system. In *IS&T/SPIE Conf. on Human Vision and Electronic Imaging III*, pages 591–602, 1998. 17
- [PE92] R. W. Picard and I. M. Elfadel. Structure of the aura and co-occurrence matrices for the gibbs texture model. *Journal of Mathematical Imaging and Vision*, 2 :5–25, 1992. 11
- [Pen84] A. P. Pentland. Fractal-based description of natural scenes. *IEEE Trans. Pattern Anal. Machine Intell*, 6 :661–675, 1984. 13
- [Pet02] E. G. M. Petrakis. Design and evaluation of spatial similarity approaches for image retrieval. *Image Vision Comput.*, 20(1) :56–76, 2002. 23
- [PF77] E. Persoon and K. S. Fu. Shape discrimination using fourier descriptors. *IEEE Trans. Sys. Man. Cyb*, 7 :170–179, 1977. 14
- [PF97] E. G. M. Petrakis and C. Faloutsos. Similarity searching in medical image databases. *Knowledge and Data Engineering*, 9(3) :435–447, 1997. 23
- [PG87] A. Pérez and R. C. Gonzalez. An iterative thresholding algorithm for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(6) :742–751, 1987. 27
- [Pha01] D. L. Pham. Spatial models for fuzzy clustering. *Computer Vision and Image Understanding*, 84(2) :285–297, 2001. 36, 75
- [PM95] R. W. Picard and T. P. Minka. Vision texture for annotation. *Multimedia Systems : Special Issue on Contentbased Retrieval*, 3(1) :3–14, 1995. 17
- [PNHA84] S. Peleg, J. Naor, R. Hartley, and D. Avnir. Multiple resolution texture analysis and classification. *IEEE Trans. Pattern Anal. Machine Intell*, 6 :518–523, 1984. 13
- [PP93] N. R. Pal and S. K. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26(9) :1277–1294, 1993. 27, 28, 71
- [PPS96] A. Pentland, R. W. Picard, and S. Sclaroff. Photobook : content-based manipulation of image databases. *Int. J. Comput. Vision*, 18(3) :233–254, 1996. 25
-

- 
- [PW00] R. Pajarola and P. Widmayer. An image compression method for spatial search. *IEEE Trans. Image Processing*, 9(3) :357–365, 2000. 48
- [PZM97] G. Pass, R. Zabih, and J. Miller. Comparing images using color coherence vectors. In *Fourth ACM Conference on Multimedia*, pages 65–73, 1997. 10
- [RHC99] Y. Rui, T. S. Huang, and S. F. Chang. Image retrieval : Current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation*, 10 :39–62, 1999. 8
- [RHOM98] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback : A power tool for interactive content-based image retrieval. *IEEE Transactions on circuits and systems for video technology*, 8 :644–655, 1998. 17
- [RK82] A. Rosenfield and A. C. Kak. *Digital Picture Processing*. Academic Press, Inc., Orlando, FL, USA, 1982. 27
- [RPG99] K. Ravishankar, B. Prasad, S. Gupta, and K. Biswas. Dominant color region based indexing for cbir. In *Proceedings of the International Conference on Image Analysis and Processing*, pages 887–892, 1999. 10
- [RTG98] Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In *ICCV '98 : Proceedings of the Sixth International Conference on Computer Vision*, 1998. 19
- [Sal68] G. Salton. *Automatic Information Organization and Retrieval*. McGraw-Hill, 1968. 6
- [Sam90] H. Samet. *The design and analysis of spatial data structures*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1990. 102
- [SB91] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1) :11–32, 1991. 10, 19
- [SB97] G. Smith and I. Burns. Measuring texture classification algorithms. *Pattern Recognition Letters*, 18(14) :1495–1501, 1997. 14, 59
- [SBLM98] J. Shi, S. Belongie, T. K. Leung, and J. Malik. Image and video segmentation : The normalized cut framework. In *IEEE International Conference on Image Processing*, pages 943–947, 1998. 37
- [SC94] J. R. Smith and S. F. Chang. Transform features for texture classification and discrimination in large image databases. In *IEEE Int. Conf. on Image Proc*, 1994. 13
- [SC96] J. R. Smith and S. F. Chang. Visualseek : a fully automated contentbased image query system. In *ACM Multimedia Conference*, pages 87–98, 1996. 8, 26
- [SCZ01] R. Schettini, G. Ciocca, and S. Zuffi. *Color Imaging Science : Exploiting Digital Media*, chapter Indexing and retrieval in colour image databases. R. Luo, L. MacDonald eds., J. Wiley, 2001. 8
- [Sha76] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, New Jersey, 1976. 35
-

- 
- [Sha86] S. Shatford. Analyzing the subject of a picture : A theoretical approach. *Cataloguing and Classification Quaterly*, 6(3) :39–62, 1986. 6
- [SK94] W. Skarbek and A. Koschan. Colour image segmentation - a survey. Technical report, Technical University of Berlin, 1994. 72
- [SM97] J. Shi and J. Malik. Normalized cuts and image segmentation. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, pages 731–737, 1997. 37, 70
- [Smi97] J. R. Smith. *Integrated spatial and feature image systems : retrieval, analysis and compression*. PhD Thesis, Columbia University, 1997. 19
- [SO95] M. A. Stricker and M. Orengo. Similarity of color images. In *SPIE, Storage and Retrieval for Image Video Databases*, pages 381–392, 1995. 8
- [SS94] M. A. Stricker and M. J. Swain. The capacity of color histogram indexing. In *IEEE, Conference on Computer Vision and Pattern Recognition*, pages 704–708, 1994. 10
- [SSWC88] P. K. Sahoo, S. Soltani, A. K.C. Wong, and Y. C. Chen. A survey of thresholding techniques. *Comput. Vision Graph. Image Process.*, 41(2) :233–260, 1988. 27
- [STC97] S. Sclaroff, L. Taycher, and L. La Cascia. Imagerover : A content based image browser for the world wide web. In *IEEE Workshop on Content-based Access of Image and Video Libraries*, 1997. 8
- [SW83] C. Sun and W. G. Wee. Neighboring gray level dependence matrix for texture classification. *Computer Vision, Graphics and Image Processing*, 23(3) :341–352, 1983. 11
- [SWS<sup>+</sup>00] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12) :1349–1380, 2000. 6, 24
- [Sym81] M. J. Symons. Clustering criteria and multivariate normal mixtures. *Biometrics*, 37 :35–43, 1981. 32
- [SZ97] G. Seetharaman and B. Zavidovique. Image processing in the tree of peano traversed images. In *IEEE Workshop on Computer Architecture for Machine Perception*, 1997. 48
- [SZ98] G. Seetharaman and B. Zavidovique. Z-trees : Adaptive pyramid algorithms for segmentation. In *IEEE International Conference on Image Processing, Chicago, IL*, 1998. 49
- [TA97] M. Tabb and N. Ahuja. Multiscale image segmentation by integrated edge and region detection. *IEEE Trans. Image Processing*, 6(5) :642–655, 1997. 28
- [Tau92] G. Taubin. *Recognition and positioning of rigid objects using algebraic and moment invariants*. PhD thesis, Providence, RI, USA, 1992. 15
- [TC88] C. H. Teh and R. T. Chin. On image analysis by the methods of moments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(4) :496–513, 1988. 15
-

- [TG74] J. T. Tou and R. C. Gonzalez. *Pattern Recognition Principles*. Addison-Wesley, Massachusetts, 1974. 32
- [TJ93] M. Tuceryan and A. K. Jain. Texture analysis. *Handbook of pattern recognition & computer vision*, pages 235–276, 1993. 11, 13
- [TMY78] H. Tamura, S. Mori, and T. Yamawaki. Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man and Cybernetics*, 8(6) :460–473, 1978. 10, 25
- [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. 13
- [TP98a] Y.A. Tolias and S.M. Panas. Image segmentation by a fuzzy clustering algorithm using adaptive spatially constrained membership functions. *IEEE Transactions on Systems, Man, and Cybernetic Part A*, 28(3) :359–369, 1998. 75
- [TP98b] Y.A. Tolias and S.M. Panas. On applying spatial constraints in fuzzy image segmentation using a fuzzy rule-based system. *IEEE Signal Processing Letters*, 5(10) :245–247, 1998. 35, 75
- [TS85] T. Takagi and M. Sugeno. Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics*, 15(1) :116–132, 1985. 36
- [Tur86] M. R. Turner. Texture discrimination by gabor functions. *Biological Cybernetics*, 55 :71–82, 1986. 13
- [Uns95] M. Unser. Texture classification and segmentation using wavelet frames. *IEEE Trans. on Image Processing*, 4(11) :1549–1560, 1995. 13
- [VP87] H. Voorhees and T. Poggio. Detecting textons and texture boundaries in natural images. In *In Proceedings of the First International Conference on Computer Vision*, pages 250–258, 1987. 12
- [VT02] R. C. Veltkamp and M. Tanase. Content-based image retrieval systems : A survey. Technical report, Technical report UU-CS-2000-34, Utrecht University, 2002. 24
- [War63] J. H. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(2) :236–245, 1963. 31
- [WDR76] J. S. Weszka, C. R. Dyer, and A. Rosenfeld. A comparative study of texture measures for terrain classification. *IEEE Transactions on Systems, Man and Cybernetics*, 6(4) :269–285, 1976. 13, 46
- [WG01] M. Worring and T. Gevers. Interactive retrieval of color images. *International Journal of Image and Graphics*, 1(3) :387–414, 2001. 8, 10
- [WLW00] J. Z. Wang, J. Li, and G. Wiederhold. Simplicity : Semantics-sensitive integrated matching for picture libraries. In *VISUAL '00 : Proceedings of the 4th International Conference on Advances in Visual Information Systems*, pages 360–371. Springer-Verlag, 2000. 26
-

- 
- [WLW01] J. Z. Wang, J. Li, and G. Wiederhold. Simplicity : Semantics-sensitive integrated matching for picture libraries. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(9) :947–963, 2001. 70
- [WS89] A. K. C. Wong and P. K. Sahoo. A gray-level threshold selection method based on maximum entropy principle. *IEEE Trans. Systems, Man, and Cybernetics*, 19(4) :866–871, 1989. 27
- [XB91] X. L. Xie and G. Beni. A validity measure for fuzzy clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(8) :841–847, 1991. 39, 83, 84
- [YL98] H. C. Yung and H. S. Lai. Segmentation of color images based on the gravitational clustering concept. *Optical Engineering*, 37(3) :989–1000, 1998. 37
- [Zad65] L. Zadeh. Fuzzy sets. *Information and Control*, 8(3) :338–353, 1965. 29, 73, 104
- [ZL98] D. Zugaj and V. Lattuati. A new approach of color images segmentation based on fusing region and edge segmentations outputs. *Pattern Recognition*, 31(2) :105–113, 1998. 28
- [ZLE99] N. Zahid, M. Limouri, and A. Essaid. A new cluster-validity for fuzzy clustering. *Pattern Recognition*, 32(7) :1089–1097, 1999. 39, 83
- [ZLW00] S. C. Zhu, X. W. Liu, and Y. N. Wu. Exploring texture ensembles by efficient markov chain monte carlo-toward a 'trichromacy' theory of texture. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(6) :554–569, 2000. 11
- [ZR72] C. T. Zahn and R. Z. Roskies. Fourier descriptors for plane closed curves. *IEEE Trans. on Computers*, pages 269–281, 1972. 14
- [ZS01] B. Zavidovique and G. Seetharamann. \*-trees for peano based image processing. Technical report, CACS Technical Report, and Working paper, University of Louisiana, 2001. 49
- [Zuc76] S. W. Zucker. Toward a model of texture. *Computer Graphics and Image Processing*, 5 :190–202, 1976. 12
-