

# Etude des attributs d'images pertinents de l'imagerie Pleïades

Marine Campedel Henri Maître Michel Roux Ivan Kyrgyzov Mihai Costache Houda Chaabouni



Mai 2005

Département Traitement du Signal et des Images Groupe Traitement et Interprétation des Images

Dépôt légal : 2005 – 4<sup>ème</sup> trimestre Imprimé à l'Ecole Nationale Supérieure des Télécommunications – Paris ISSN 0751-1345 ENST D (Paris) (France 1983-9999)

Ecole Nationale Supérieure des Télécommunications Groupe des Ecoles des Télécommunications - membre de ParisTech 46, rue Barrault - 75634 Paris Cedex 13 - Tél. + 33 (0)1 45 81 77 77 - www.enst.fr Département TSI



# Rapport à mi-parcours de l'étude DCT/SI/AP-04-521 effectuée à la demande du CNES

# Etude des attributs d'images pertinents de l'imagerie Pleïades

Marine Campedel, Henri Maître, Michel Roux, Ivan Kyrgyzov, Mihai Costache, Houda Chaabouni

> **11 mai 2005** Département Traitement du Signal et des Images GET - Télécom Paris

# Table des matières

In	trodu	ction		5								
1	Séle	ction de	e caractéristiques	9								
	1.1	Introd	uction	9								
	1.2	Métho	dologie	10								
	1.3	Sélecti	ions supervisées	11								
		1.3.1	Les données	11								
		1.3.2	Performances des différents ensembles de caractéristiques	11								
		1.3.3	Sélection sur l'ensemble des caractéristiques	13								
		1.3.4	La sélection de Fisher-FS	13								
		1.3.5	Conclusion	16								
	1.4	Sélecti	ions non supervisées	16								
		1.4.1	Expériences similaires	16								
		1.4.2	Stabilité des résultats	16								
		1.4.3	Conclusion	17								
	1.5	Conclu	usion et travaux futurs	17								
2	Hierarchical clustering of textures 1											
4	2 1	Introdu	uction	10								
	$\frac{2.1}{2.2}$	Hieror		19								
	2.2	221	Notations	20								
		2.2.1	Partitional clustering algorithm	20								
		2.2.2	Hierarchical agglomerative clustering algorithm	$\frac{21}{22}$								
		2.2.3	Estimation of the clustering solution	22								
	23	Z.Z.4		22								
	2.3	2 2 1	Clustering of Produtz textures	23								
		2.3.1	Clustering of textures of SPOT image	$\frac{25}{25}$								
		2.3.2	Clustering of urban textures of SPOT image	23								
	24	2.5.5 Conch	Clustering of urban-textures of 51 O1 image	20								
	2.4	Conch		50								
3	The	role of	the window size in the classification process	33								
	3.1	Object	tive and method	33								
		3.1.1	Objectives of the study	33								
		3.1.2	Methodology	33								
		3.1.3	Notations	34								

		3.1.4	Obtaining images with size as a power of 2	34
	3.2	Experi	ments and results	36
		3.2.1	Conclusions	36
4	Inde	exation ]	par primitives de région	41
	4.1	Introdu	action	41
		4.1.1	Objectifs	41
		4.1.2	Principes	41
	4.2	Etat de	e l'art des systèmes de fouille par régions	42
		4.2.1	Blobworld	42
		4.2.2	Netra	42
		4.2.3	Simplicity	43
		4.2.4	STRICT	44
	4.3	Attribu	Its de description de la forme	44
		4.3.1	Moments d'inertie	44
		4.3.2	Moments de Hilbert	44
		4.3.3	Décomposition	45
		4.3.4	Facteur de forme	45
		4.3.5	Conclusions	46
	4.4	Notre e	expérimentation	46
		4.4.1	Pertinence des attributs	46
		4.4.2	Description de la base de données	47
		4.4.3	Les moments de Hilbert	48
		1.1.5		10
Co	onclus	sion-1		57
Bi	bliogr	aphie		57

# Avertissement

Ce document est le rapport à mi-parcours de l'étude : **DCT/SI/AP-04-521 « Etude des attributs image pertinents de l'imagerie PLEIADES »**. Il accompagne un rapport concernant l'étude : **DCT/SI/AP-04-522 « Interpolation Multi-échelle des attributs des images »**. Ces deux études sont exécutées à la demande du CNES par le département TSI de Télécom-Paris, et dans le cadre du Centre de Compétence CNES-DLR-ENST.

Ces deux rapports sont très complémentaires, les travaux qu'ils relatent étant intimement imbriqués, mais présentés ici de façon séparée pour des raisons contractuelles. Ces travaux font suite à deux rapports [5], [6], aux conclusions regroupés dans un document public : [4].

Ces travaux ont également bénéficié de l'assistance de Luo Bin et Mihai Costache pour la constitution des bases de données et l'expérimentation, des conseils d'Eric Moulines pour l'optimisation des systèmes d'apprentissage et des discussions avec Jean-François Aujol et Yann Gousseau sur la détection des primitives.

Alain Giros (CNES) et Mihai Datcu (DLR) ont contribué au choix des méthodes présentées, ainsi qu'aux discussions des résultats.

# Introduction

Nous présentons ici les premiers résultats concernant l'étude des primitives appropriées à l'indexation des images des satellites à haute résolution Pléïades.

Nos travaux antérieurs (présentés dans [4]) ont essentiellement porté sur les images du satellite SPOT-5, à résolution de 5 m. Elles ont conclu à l'intérêt d'une indexation à partir de primitives de texture. Nous devions vérifier dans quelle mesure ces résultats s'étendaient à des images à résolution submétrique comme les images du futur programme Pléïades.

Nos travaux récents (rapportés dans [3]), contribuent de façon importante à ce débat. Ils ont montré que l'évolution des images, lorsque l'on fait croître la résolution, conduit leur apparence d'un domaine où la géométrie est dominante à un domaine où la texture est dominante. La zone de transition dépend beaucoup de l'homogénéité des structures présentes et - bien sûr - de la taille des objets constituant la scène. Dans le rapport cité, nous avons proposé des outils pour caractériser ces échelles. Nous ne les présentons pas ici, mais renvoyons le lecteur à leur consultation pour plus de détails. Ce ne sont que des résultats préliminaires qui s'appuient sur une démarche théorique et n'ont été vérifiés expérimentalement que sur un seul exemple. Nous ne disposons en effet que d'un seul jeu d'images à la résolution des images de Pleïades. Ces travaux doivent donc encore être étayés.

Il est donc confirmé (mais cela était bien sûr intuitif) que le passage à la résolution de Pleïades doit nous amener à introduire plus de composantes géométriques dans les primitives d'indexation.

## **Retour sur la classification**

Nous avons donc tout d'abord porté notre attention sur cet aspect en étudiant le comportement des algorithmes d'apprentissage et de classification en présence d'un mélange de primitives géométriques et de primitives texturales (Chapitre 1). Ces travaux avaient été briêvement mentionnés dans notre rapport [5]. Nous avions alors pressenti la complémentarité des primitives géométriques associées aux primitives texturales. Ces premiers résultats sont confirmés et approfondis dans le chapitre 1. Mais ce chapitre 1 va au delà de ces conclusions puisqu'il présente également nos dernières conclusions sur cette étape d'apprentissage et de sélection des primitives : des méthodes adaptées de sélection des primitives permettent de faire progresser notablement les performances du système de classification. En particulier, dans le cas de sélections supervisées, les performances deviennent excellentes et les erreurs extrêmement faibles (de l'ordre de 2 %). en outre, même dans le cas non supervisé, nos résultats de classification sont assez bons, même si l'on ne bénéficie pas d'un filtrage du bruit lors de l'étape de sélection des primitives.

### La taille des fenêtres d'analyse

Nous nous sommes ensuite un peu éloignés de l'aspect Pléïades pour aborder un point qui avait été laissé de côté dans nos études précédentes et qui concerne la taille de la fenêtre d'analyse lors de la mesure des primitives d'indexation. Nous avions choisi de façon assez arbitraire de mesurer les primitives sur des fenêtres de 64 x 64 pixels. Des fenêtres plus grandes contiennent plus d'information, et ce, d'autant plus que la résolution est forte comme dans Pléïades. On pourrait donc en attendre une meilleure qualité. Mais on peut s'interroger sur leur capacité de bien caractériser une zone homogène puisque pour SPOT-5, une fenêtre de 100 x 100 pixels couvre une étendue de 500 x 500 m, rarement très homogène. Au contraire, une fenêtre plus petite sera très homogène, mais probablement mal capable de refléter les propriétés d'objets de grande taille.

Afin de contrôler ce point nous avons donc entrepris une étude pour déterminer la taille de la fenêtre la plus intéressante pour extraire des primitives (chapitre 3). Nous montrons qu'une fenêtre de l'ordre de 72 pixels garantit un taux de reconnaissance quasiment optimal. Nous étendons ces travaux à une analyse classe de paysage par classe de paysages. Nous montrons que, si de petites fenêtre (35 x 35) sont suffisantes pour les classes de mer, de nuages ou de forêts, il faudrait des fenêtres nettement plus grandes pour des zones de champs.

## **Indexation structurelle**

Nous nous sommes ensuite intéressés aux approches d'indexation à base d'information structurelle (Chapitre 4), tout d'abord en faisant une étude bibliographique de l'existant, ce qui nous a permis de choisir quelques voies d'exploration, puis en conduisant quelques expérimentations où nous avons privilégié une approche par segmentation suivie d'une description des régions, soit par des moments invariants, soit par des descripteurs de forme. Des premiers résultats sont présentés à titre d'information, mais ils ne permettent pas encore de conclure, ni sur le choix des caractéristiques, ni sur les performances que l'on est en droit d'attendre.

Nous en sommes là de ces travaux.

# Chapitre 1

# Sélection de caractéristiques

# **1.1 Introduction**

Les rapports d'études précédents [5, 6] ont montré l'intérêt d'utiliser des procédures de sélection automatique de sélection d'attributs pour comparer l'efficacité de différents ensembles de caractéristiques produits par des experts. Nous avons alors étudié différents algorithmes supervisés de la littérature et mis en évidence que :

- une procédure de sélection efficace permet d'améliorer les performances d'une classification effectuée sur un même ensemble de données, en réduisant la redondance et en éliminant les attributs non pertinents; en particulier, sur une base de 600 imagettes satellitaires issues de 6 classes différentes, nous avons pu démontrer l'intérêt (en termes de taux de classification et de mesure de redondance à travers le critère appelé "entropie de représentation") de réduire l'ensemble initial des 78 coefficients d'Haralick à une dizaine.
- les procédures de sélection et les algorithmes de classification doivent être couplées intelligemment pour fournir les meilleurs résultats. Par exemple une sélection de Fisher suivie par un classificateur de Fisher donne toujours de moins bons résultats que si elle est suivie par un classificateur SVM; en effet, ce dernier permet de réduire encore le poids des attributs redondants qui ont tendance à être conservés par la procédure de sélection de Fisher.

Dans le cadre de cette étude, nous nous intéressons à la comparaison des caractéristiques utiles pour l'indexation des images satellitaires. Il apparaît clairement que se limiter à travailler sur des bases supervisées n'est pas suffisant pour assurer l'efficacité des attributs sélectionnés lors de requêtes non préalablement définies. Nous nous sommes donc également intéressés à des procédures de sélection non supervisées. L'approche usuelle consiste à effectuer une classification non supervisée (clusterisation) des données sur différents sousensembles d'attributs et de sélectionner le sous-ensemble le plus efficace à l'aide d'heuristiques. L'exploration des sous-ensembles possibles exploite des algorithmes gourmands (*greedy*) ou stochastiques (algorithmes génétiques); ils sont donc très coûteux en temps de calcul. Nous avons choisi de limiter notre intérêt à des algorithmes simples et efficaces qui ne sont capables que de réduire la redondance (et non le bruit). Ces algorithmes reposent tous sur une clusterisation des attributs et choisissent des représentants de clusters comme résultat de la sélection. Nous avons montré que ces algorithmes fonctionnent très efficace-

ment sur des ensembles redondants (tel que celui des 78 coefficients d'Haralick) mais ne sont pas robustes en présence de bruit (par exemple si on ajoute des attributs géométriques, dont l'efficacité n'est pas a priori prouvée).

Nous effectuons tout d'abord, dans ce chapitre, une petite correction méthodologique. Nous présentons ensuite un prolongement des travaux effectués auparavant, en effectuant une sélection sur l'ensemble des caractéristiques à notre disposition, en incluant les GMRF (Gaussian Markov Random Fields) [11]. Nous montrons enfin que les procédures non supervisées ne donnent pas d'aussi bons résultats sur l'ensemble concaténé des caractéristiques mais nous évaluons la stabilité des résultats de sélection obtenus sur une base d'imagettes choisies aléatoirement. Nous terminons en concluant sur ces travaux et en présentant les axes de recherche concernant la sélection automatique.

# 1.2 Méthodologie

La méthodologie que nous avons présentée dans les précédents rapports consiste à appliquer des algorithmes de sélection de caractéristiques afin de comparer différents ensembles. Les critères de comparaison utilisés sont des heuristiques (séparabilité, entropie, index de flou et surtout entropie de représentation dont nous avons montré l'efficacité) et des performances de classification en validation croisée (moyenne et variance de l'erreur de classification sur chaque boucle). L'ensemble des algorithmes retenus dans ce chapitre est présenté dans la table 1.1. Il est à noter que, contrairement à ce que nous avions écrit dans [7], il est important que la sélection de caractéristiques exploitée lors de l'apprentissage du classificateur se fasse dans la boucle de validation croisée. En effet, l'auteur de [31] illustre parfaitement le biais induit par une sélection effectuée sur l'ensemble des données, biais qui ne permet finalement pas de comparer les différentes méthodes de sélection. Les heuristiques s'appliquent à tout le jeu de données après une sélection globale de caractéristiques. Cette procédure d'évaluation est présentée figure 1.1

Nom	Туре	Description		
kNN	С	<i>k</i> -plus proches voisins [16]		
SVM	С	Support Vector Machine [10, 8]		
Fisher	С	Classification de Fisher [17]		
Fisher-FS	S-W	Sélection de Fisher [33]		
ReliefF	S-F	[23, 32]		
SVM-RFE	S-W	[13]		
$\mathcal{L}_2$ -AROM	S - Wrapper	[34]		
MIC	S-F	Non supervisé [28]		
<i>k</i> Means-FS	S-F	Non supervisé [6]		
SVC-FS	S-F	Non supervisé [6]		

TAB. 1.1 – Algorithmes utilisés dans cette étude. S designe un algorithme de sélection (F pour *filter* et W pour *wrapper*), C un algorithme de classification. La seule heuristique d'évaluation exploitée est l'entropie de représentation notée H.



FIG. 1.1 – Evaluation d'un sous-ensemble d'attributs sélectionné automatiquement. La procédure de sélection peut être supervisée ou non mais l'ensemble de données utilisé pour évaluer l'efficacité par le taux de classification doit être actuellement étiqueté.

# **1.3** Sélections supervisées

Nous prolongeons les travaux effectués essentiellement sur les coefficients d'Haralick à la comparaison des différents ensembles d'attributs à notre disposition.

#### 1.3.1 Les données

Les données exploitées sont toujours les 600 imagettes (64x64) réparties en 6 classes illustrées sur la figure 1.2, extraites de scènes SPOT 5 (5m/pixel, données fournies par le CNES). Le détail des ensembles de caractéristiques étudiés est présenté brièvement dans la table 1.2. Les caractéristiques dénommées GMRF sont issues d'une modélisation (sur une fenêtre de taille 64 dans notre étude), paramétrique, à partir de modèles de champs de Gibbs-Markov. La procédure de choix du modèle et d'estimation de ses paramètres est présentée dans[11]. Les trois attributs extraits sont : la norme des paramètres du modèle optimal, la variance de l'erreur de modélisation ainsi que l'évidence du modèle.

#### **1.3.2** Performances des différents ensembles de caractéristiques

La première comparaison possible consiste à mesure le pouvoir discriminant de ces différents ensembles en mesurant le taux moyen de classification en validation croisée. Les résultats obtenus sont présentés dans la table 1.3. Ils correspondent à la moyenne et à l'écarttype des taux d'erreurs de trois classificateurs différents appliqués aux données exprimées selon les différents ensembles de caractéristiques, ainsi qu'à l'ensemble concaténé. Les observations sont les suivantes :



FIG. 1.2 – Illustration des 6 classes de textures, de gauche à droite et de haut en bas : ville, forêt, champs, mer, désert et nuages (@CNES).

Name	Nb	Ref.	Description
m	1		gray level mean
$\sigma$	1		gray level standard deviation
Haralick	78	[14]	Statistics on cooccurrence matrices.
Gabor 24 [3		[30]	Means and variances of gabor filtered
			outputs, 3 scales and 4 orientations.
Qmf	18	[25]	Means and variances of QMF
			sub-bands decomposition (3 scales)
GMRF	3	[11]	Gibbs Markov random fields
			w=64
Geometric	15	[4]	Statistics based on linear
			approximation of detected edges.

TAB. 1.2 – Caractéristiques extraites des imagettes 64x64. Les références bibliographiques correspondantes sont précisées entre crochets.

- La concaténation des différentes caractéristiques disponibles n'obtient pas les meilleures performances de classification, ce qui démontre une nouvelle fois la nécessité de sélectionner les attributs pertinents et non redondants.
- Les moyennes et écart-types des niveaux de gris, avec seulement deux paramètres, obtiennent des performances plus intéressantes que les Gabor ou les QMF.
- Les trois paramètres GMRF, auxquels ont été ajoutés les moyennes et écart-types des niveaux de gris obtienent, très significativement, deux des meilleurs performances de classification, avec seulement 5 paramètres.
- Les caractéristiques géométriques seules ne sont pas significativement plus intéressantes que les Gabor ou les QMF.

Ces premières observations permettent de conclure que l'ensemble  $m+\sigma+GMRF$  est le plus compact et le plus performant à ce stade de l'étude. Il semble intéressant de tester les procédures supervisées de sélection automatique sur l'ensemble concaténé.

Satelli		llite	600 exemples - 6 classes			]
				$\mathbf{F} =$		
		dim.	,	kNN(k=8)	Fisher	SVM
Concaténa	tion	140		$11.3 \pm 1.8$	$17.8 \pm 4.2$	$6.5 \pm 2.0$
m+\sigma		2		$11.2 \pm 3.2$	$15.2 \pm 2.0$	-
Haralic	k	78		$13.8 \pm 2.2$	$30.8 \pm 4.3$	$7.7 \pm 2.0$
m+ $\sigma$ +Hara	alick	80		$10.8 \pm 1.6$	$22.7 \pm 2.6$	6.2±1.4
Gabor		24		$22.2 \pm 2.5$	$40.7 \pm 3.8$	$16.2 \pm 5.3$
m+ $\sigma$ +Gabor		26		17.7±3.3	$33.7 \pm 5.1$	$12.2 \pm 2.8$
QMF		18		$24.8 \pm 1.9$	38.5±4.0	$21.2 \pm 5.4$
m+ $\sigma$ +QN	МF	20		$17.2 \pm 1.7$	28.3±3.9	-
GMRF	7	3		$20.6 \pm 5.2$	26.7±3.2	-
m+ $\sigma$ +GMRF		5		5.7±2.3	$10.8{\pm}2.2$	$8.3 \pm 1.8$
Geo		15		33.7±3.9	$38.8 \pm 3.7$	-
m+\sigma+G	eo	17		$16.5 \pm 2.2$	$28.3 \pm 3.9$	11.8±1.9

TAB. 1.3 - % Moyenne $\pm\%$  Ecart-type de l'erreur de classification, pour différents ensembles de caractéristiques. Les classificateur SVM n'exploite pas de noyau particulier, il est employé dans le cadre linéaire. Les valeurs manquantes correspondent à une non-convergence de l'algorithme, en préservant exactement les mêmes paramètres (C = 1000, implantation libsvm [8]).

#### **1.3.3** Sélection sur l'ensemble des caractéristiques

Les meilleurs résultats obtenus correspondent à une sélection de 14 attributs par l'algorithme de Fisher-FS. Le choix de 14 correspond à la meilleure performance de classification obtenue en comparant des sélections de 6 à 40 attributs (cf figure 1.3). Cette performance atteint une erreur moyenne d'environ 2%, à comparer aux 6% précédemment obtenus. Notons également que la performance de l'algorithme des *k*-plus-proches-voisins est alors équivalente à celle obtenue par l'ensemble m+ $\sigma$ +GMRF mais que celle de l'algorithme de Fisher est dégradée. Cette dernière est cependant nettement moins intéressante (autour de 10%).

Nous observons également que les autres méthodes de sélection obtiennent des performance de classification similaires à celles de l'ensemble concaténé avec une réduction d'un facteur 10 du nombre de caractéristiques. Elles sont donc efficaces mais ne nous ont pas permis d'atteindre la performance de la méthode de Fisher-FS; nous détaillons donc maintenant cette méthode et ses résultats.

#### 1.3.4 La sélection de Fisher-FS

La sélection de Fisher est une méthode classique et surprenamment efficace sur notre ensemble de données. Nous exploitons l'implantation faite par l'outil Spider [33]. Cet algorithme ordonne les attributs à l'aide des poids estimés par analyse discriminante. Dans le cas d'un problème à deux classes, le principe de cette analyse consiste à déterminer l'hyperplan pour lequel le rapport des variances projetées inter vs intra classes est maximisé.

Sat	tellite 600 e	exemples - 6	classes	
		D = 140, F =	= 5	
	kNN(k=8)	Fisher	SVM	Н
d = 140	$11.3 \pm 1.8$	$17.8 \pm 4.2$	$6.5 \pm 2.0$	
d = 14				
Relieff	$18.5 \pm 2.5$	31.1±3.7	17.7±3.4	0.14
Fisher-FS	5.7±1.9	15.2±2.9	<b>1.8</b> ±1.7	1.36
SVM-RFE	$10.5 \pm 2.8$	$16.7 \pm 3.6$	$5.5 \pm 2.2$	1.54
$l_2$ -AROM	9.5±3.6	15.8±4.6	$6.8 \pm 2.8$	1.75
MIC	$20.5 \pm 5.0$	$28.5 \pm 7.8$	9.2±2.1	2.18
kMeans-FS	$20.5 \pm 1.6$	$28.5 \pm 1.4$	$10.2 \pm 0.7$	2.20
SVC-FS	$19.7 \pm 2.8$	$26.8 \pm 5.5$	$8.5 \pm 2.1$	2.01

TAB. 1.4 – Evaluation des 14 attributs sélectionnés automatiquement par 4 méthodes supervisées.



FIG. 1.3 –

Lorsque l'on fait l'hypothèse que les attributs suivent une loi gaussienne  $N(\mu^{(+)}, \sigma^{(+)})$  et  $N(\mu^{(-)}, \sigma^{(-)})$  sur chaque classe, les poids de chaque attribut d se calculent explicitement par la formule :

$$w_d = \frac{\mu_d^{(+)} - \mu_d^{(-)}}{\sigma_d^{(+)} + \sigma_d^{(-)}} \tag{1.1}$$

Dans le cas de problèmes à plus de deux classes, on se ramène à des problèmes à deux classes par une stratégie une-contre-une. Les poids permettant de discriminer une classe de chacune des autres sont moyennés afin de fournir un poids par attribut et par classe. Les d meilleurs attributs finalement retenus font partie des quelques meilleurs de chaque classe. Il

apparaît donc que l'ordre des classes n'est pas innocent dans le résultat obtenu. N'importe quel nombre de caractéristiques peut être choisi, en respectant  $K-1 \le d$ , avec K le nombre de classes.

La sélection, ainsi que l'avons souligné dans la section 1.2, est effectuée sur chaque ensemble d'apprentissage. Les résultats sont présentés dans la table 1.5. Nous observons que :

- La moyenne des niveaux de gris est une caractéristiques dominante. Nous observons effectivement que cet attribut permet de discriminer trois classes, par un simple effet de seuil : (nuages), (mer,champs), (désert, ville, forêt).
- Quatre attributs géométriques sont sélectionnés. Geo1 correspond au nombre de segments de droites extraits pour approximer les contours obtenus. Geo5 (resp. Geo6) est relié à la moyenne de la taille (resp. seuillée) de ces mêmes segments. Geo13 est obtenue à l'aide des matrices de co-occurrence estimées sur les images binaires de contour.
- Cinq coefficients d'Haralick sont sélectionnés. Le fait d'observer des échanges de caractéristiques au même rang, pour des ensembles d'apprentissages différents, suggère une relation de redondance entre ces caractéristiques. En effet, Hara6, Hara19, Hara58 and Hara71 correspondent au même calcul de statistiques mais sur des matrices de co-occurrence différentes (obtenues respectivement pour des orientations de 0°, 45° ainsi que pour la moyenne et l'écart-type de la statistique sur toutes les orientations). Cette statistiques est appelée Sum Average <sup>1</sup>. Les quatre derniers coefficients correspondent à l'écart-type calculé sur les 4 orientations, de quatre autres statistiques (appelées Inverse Difference Moment, Sum Entropy, Entropy and Information Measures of Correlation ).
- L'évidence de la modélisation par des champs de Markov est également un paramètre bien classé dans la sélection, ce qui confirme l'intérêt d'un tel modèle dans le cadre de l'imagerie satellitaire.

1		2	3		4	5	6
m		Hara6	Gabor6		Geo1	Geo5	GMRF3
7		8	9		10	11	12
Hara19,58		Hara71,74	Geo13		QMF5,1	Geo6	σ
13 14							
Hara19 Hara70,71,73,77			1				

TAB. 1.5 – Les 14 attributs ordonnés par la méthode de Fisher. Les attributs séparés par un	e
virgule sont obtenus au même rang, pour des ensembles d'apprentissages différents.	

 $<sup>^{1}</sup>f_{6} = \sum_{i=2}^{2N_{g}} ip_{x+y}(i)$  avec  $N_{g}$  le nombre de niveaux de gris et  $p_{x+y}(k) = \sum_{i=1}^{N_{g}} \sum_{j=1,i+j=k}^{N_{g}} p(i,j), k = 2,3,...,2N_{g}, p(i,j)$  correspondant à une donnée de la matrice de co-occurrence.

#### 1.3.5 Conclusion

Les méthodes de sélection supervisées ont confirmé leur efficacité sur l'ensemble de tous les attributs disponibles à ce jour. La méthode de Fisher a permis d'atteindre la meilleure performance (aux alentours de 2% d'erreur avec un classificateur SVM linéaire). D'autre part, les GMRF ont démontré une bonne aptitude à discriminer les 6 classes de l'ensemble de données. Enfin le sous-ensemble sélectionné et permettant la meilleure classification contient essentiellement des attributs de texture mais également quelques attributs géométriques.

## 1.4 Sélections non supervisées

#### **1.4.1** Expériences similaires

Nous avons également effectué une sélection non supervisée sur l'ensemble de toutes les caractéristiques concaténées. Les résultats sont présentés dans la table 1.6. Nous observons, ainsi que nous nous y attendions que les méthodes non supervisées ont des performances très nettement dégradées face aux méthodes supervisées. Nous estimons que cela est dû à la présence de caractéristiques bruitées, que ne savent pas traiter les méthodes non supervisées. Cette conclusion est renforcée par le fait que l'entropie de représentation associée à ces algorithmes est plus élevée que pour les méthodes supervisées, ce qui est significatif d'une plus faible redondance (donc à nombre égal de composantes, et étant donné les faibles performances de classification, de plus de bruit).

Sa	tellite 600	exemples - 6 D = 140, F =	classes = 5	
	<i>k</i> NN(k=8)	Fisher	SVM	Н
d = 140	11.3±1.8	17.8±4.2	6.5±2.0	
d = 14				
Fisher-FS	5.7±1.9	15.2±2.9	1.8±1.7	1.36
MIC	20.5±5.0	28.5±7.8	9.2±2.1	2.18
<i>k</i> Means-FS	$20.5 \pm 1.6$	28.5±1.4	$10.2 \pm 0.7$	2.20
SVC-FS	19.7±2.8	26.8±5.5	8.5±2.1	2.01

TAB. 1.6 – Evaluation des 14 attributs sélectionnés automatiquement par 4 méthodes supervisées.

#### 1.4.2 Stabilité des résultats

Comme souligné en introduction, Les procédures de sélection de caractéristiques non supervisées sont faites pour réduire la redondance. Nous avons observé que les performances obtenues sur l'ensemble des 78 coefficients d'Haralick étaient équivalentes à celles obtenues par des méthodes supervisées, sur les mêmes ensemble de données. Nous souhaitons maintenant savoir si les sélections effectuées sur la base étiquetée sont semblables à celles obtenues sur une base quelconque, aléatoire. Ce test ne peut pas ce faire avec des algorithmes supervisés (puisqu'ils nécessitent la connaissance des vraies classes); nous le faisons à l'aide de nos procédures non supervisées afin de valider la stabilité des sélections effectuées. Nous observons (cf table 1.7) que les résultats de classification obtenus sur la base des coefficients d'Haralick issus des 600 imagettes, après sélection sur cette même base ou sur une base aléatoire, sont équivalents. Ceci semble signifier que :

- Les exemples choisis dans la base étiquetée sont suffisamment représentatifs pour faire de la sélection;
- Les algorithmes non supervisés de sélection permettent d'obtenir des sélections équivalentes en terme de potentiel de classification lorsque la sélection est effectuée sur des imagettes sélectionnées aléatoirement. Ceci nous conforte dans le fait que ces méthodes donnent un résultat suffisamment fiable et stable en fonction des exemples utilisés.

		Sate	llite 600 e	exemples - 6	classes		
			$\mathbf{D} = \mathbf{T}$	78 (Haralick	), $F = 5$		
		Base 600			Base aléa.		
	d	kNN(k=8)	Fisher	SVM	kNN(k=8)	Fisher	SVM
Pas de sel.	78	$13.8 \pm 2.2$	30.8±4.3	$7.7 \pm 2.0$	-	-	-
kMeans-FS	10	$12.8 \pm 2.3$	$28.7 \pm 5.8$	$9.8 {\pm} 2.5$	$13.2 \pm 2.2$	$29.7 \pm 4.3$	$9.0{\pm}2.2$
SVC-FS	10	$14.7 \pm 2.3$	31.0±3.4	$11.7 \pm 1.3$	$14.7 \pm 2.7$	$32.0{\pm}2.7$	$10.8 {\pm} 1.7$
kMeans-FS	20	$13.5 \pm 2.8$	30.3±5.2	$6.7 \pm 1.0$	13.8±2.9	31.1±3.6	8.3±1.0
SVC-FS	20	$14.8 {\pm} 2.6$	$31.7 \pm 4.2$	$11.0 \pm 2.3$	$15.2 \pm 3.6$	$30.8 \pm 5.0$	$10.0{\pm}2.1$
kMeans-FS	39	$13.5 \pm 2.2$	31.0±3.7	8.5±1.6	$14.2 \pm 2.8$	31.2±4.6	$8.0{\pm}0.7$
SVC-FS	39	$13.7 \pm 2.7$	29.2±4.9	8.5±1.5	13.8±2.9	29.8±5.3	8.7±1.7

TAB. 1.7 – Evaluation de la stabilité de nos deux méthodes de sélection non supervisées.

#### 1.4.3 Conclusion

Les méthodes de sélection non supervisées permettent effectivement de réduire la redondance d'un ensemble d'attributs; par contre elles sont sensibles à la présence de caractéristiques non pertinentes qu'elles ne distinguent pas des autres.

Nous avons en outre montré que les sélections effectuées sur un ensemble aléatoire et sur notre base étiquetée produisent des performances de classification similaires sur la base étiquetée. Ceci laisse à penser que ces procédures sont stables et que le nombre de données utilisées (600 pour la base étiquetée et 2000 pour la base aléatoire) est suffisant pour obtenir des performances stables.

## **1.5** Conclusion et travaux futurs

Nous avons effectué différentes expériences afin de mieux évaluer nos algorithmes de sélection et de comparer les ensembles d'attributs à notre disposition.

Il vient d'une part que les méthodes supervisées sont particulièrement efficaces, notamment

la méthode de Fisher qui produit une sélection de 14 attributs (ce qui correspond à une réduction d'un facteur 10) permettant de passer de 6% à 2% d'erreur moyenne (avec un classificateur SVM linéaire). Cette méthode est stable mais ne produit pas exactement un ordonnancement des attributs; dans nos travaux futurs nous nous intéresserons à l'approfondissement de cette méthode et aux extensions apparues récemment dans la littérature (Weighted Fisher).

Le sous-ensemble le meilleur, sélectionné automatiquement, a mis en évidence l'intérêt de combiner attributs géométriques et texturaux. Cette étude a également permis d'observer l'efficacité des GMRF, associés à la moyenne et variance des niveaux de gris des images, pour classifier notre ensemble de données.

Enfin, nous avons étudié la stabilité des méthodes non supervisées en effectuant une sélection sur une base non étiquetée et en comparant les sélections produites sur la base étiquetée. Nous avons insisté sur le fait que ces méthodes ne savent actuellement par gérer les attributs non pertinents (dits bruités). Nous nous proposons de travailler sur ce point dans la suite de l'étude, d'une part en analysant les clusters produits par les différentes méthodes (taille, forme) et d'autre part en partant du principe que les attributs de bruit ne doivent pas être systématiquement sélectionnés. Nous espérons donc mettre en évidence les attributs fluctuants par une validation croisée, effectuée sur chaque base d'apprentissage, suivie d'une fusion des différentes sélection obtenues.

# Chapitre 2

# **Hierarchical clustering of textures**

## 2.1 Introduction

Image clustering and categorization is a mean for high-level description of image content. They are usually performed in the early stages of the mining process. Clustering is a division of data into groups of similar objects. Each group, called cluster, consists of objects that are similar and dissimilar to objects of other groups.

The goal of image clustering is to find a mapping of the images into clusters such that the set of clusters provide information about the image database. The generated clusters provide a summarization and visualization of the image content that can be used for different tasks related to image database management.

In our previous work [27] classification of textures of SPOT image using different classifiers (SVM, Fisher, KNN (K nearest neighbors)) was considered. But classification task use category labels that tag objects with prior identifiers. The absence of category labels distinguishes cluster analysis from classification. The objective of cluster analysis is to find a convenient and valid organization of the data, not to establish rules for separating future data into categories. We are interested in applying clustering to find structure in the database of textures of SPOT image (e.g., we can say that textures of "city" have a group of density city, industrial zone ect.).

In this work we present clustering of textures of different database. The first database consists of Brodatz textures with 40 classes, the second database is made of the textures which were extracted from satellite images SPOT5 (data given by CNES) has 6 classes (forest, field, city, cloud, desert and sea) and the third database of textures is the textures also from images SPOT5 but extracted from urban areas of different cities and different countries. The hierarchical clustering algorithm and metrics for estimation clustering are presented in section 2, results of database clustering are shown in section 3.

## 2.2 Hierarchical clustering

Usually clustering techniques are divided in partitional and hierarchical. Partitional clustering for given n patterns in a d-dimensional metric space is a partition of the patterns into m groups, or clusters, such that the patterns in a cluster are more similar to each other that to

patterns in different clusters. Hierarchical clustering methods are categorized into divisive and agglomerative [20] also called bottom-up and top-down. A divisive clustering starts with one cluster which contains all data points and splits it into the most appropriate clusters. An agglomerative clustering starts with one-point (singleton) clusters and merges two or more most appropriate clusters. Results of partitional clustering can be used as an initial solution for agglomerative algorithm.

Clustering algorithms treat the clustering problem as an optimization process which tries to maximize or minimize a particular clustering criterion function.

Some of the criterion functions that can be used to drive both partitional and agglomerative clustering algorithms are described and analyzed in [20], [36], [35]. Most of these criterion functions have been shown to produce high quality clustering solutions in high dimensional datasets. Also some of the more traditional local criteria (e.g., single-link, complete-link, and UPGMA (Unweighted Pair Group Method with Arithmetic Mean)) can be used for agglomerative clustering.

The method used to optimize this criterion function uses a randomized incremental optimization algorithm that has low computational requirements, and shows high-quality clustering solutions [35]. We use clustering toolkit (CLUTO) for the graph-partitioning clustering algorithms that is well-suited for finding clusters that form contiguous regions and for agglomerative clustering. In this way, the desired *K*-clustering solution is computed by combining both the partitional and agglomerative methods.

First stage of the method is the clustering the dataset into m clusters using partitional clustering which computed by modeling the objects using a nearest-neighbor graph (each object becomes a vertex, and each object is connected to its most similar other objects), and then splitting the graph into m-clusters using a min-cut graph partitioning algorithm. After that K clustering solution is obtained by merging some of these clusters using an agglomerative algorithm (m > K).

This approach was motivated by the two-phase clustering approach of the CHAME-LEON algorithm [21] and gives the opportunity to compute a clustering solution that uses a different clustering criterion function for the agglomeration phase. An application of such approach is to allow the clustering algorithm to find non-globular clusters.

#### 2.2.1 Notations

Graph-partitional clustering works with data wich are presented as a similarity graph. Let n denote the total number of data in the set S and  $S_r$  be the set of objects assigned to the r-th cluster of size  $n_r$ , r = 1...K, K is the number of clusters. v and u represent two objects of S, and sim(v, u) is their similarity. The similarity graph  $G_s$  is constructed by modeling each object as a vertex, and there is an edge between two objects u and v if and only if both of them are in the nearest-neighbor lists of each other. The weight of this edge is set equal to the similarity of the objects. There are some possible similarity measures between two objects :

- cosine function

$$sim_{cos}(u,v) = \frac{u^t v}{\| u \| \| v \|}$$
 (2.1)

- correlation coefficient

$$sim_{corr}(u,v) = \frac{W_{uv}}{\sqrt{W_{uu}W_{vv}}}$$

$$where \quad W_{uv} = \sum (u-\bar{u})(v-\bar{v}),$$

$$W_{uu} = \sum (u-\bar{u})^2, \quad W_{vv} = \sum (v-\bar{v})^2$$

- similarity based on Euclidean distance || u - v ||

$$sim_{eucl}(u,v) = 1 - \frac{\|u - v\|}{1 + max\{\|u - v\|\}}$$
(2.3)

- similarity between the objects is computed using the extended Jaccard coefficient of their vectors

$$sim_{jacc}(u,v) = \frac{u^{t}v}{\|u\| + \|v\| - u^{t}v}$$
(2.4)

To visualize results of clustering we use images of confusion matrix and dendrograms. Columns of the confusion matrix represent original classes of a database and rows represent found clusters. Rows are rearranged so that a position of a row correspond to a position of maximal element of row. In this case we will have the best clustering solution when the confusion matrix is diagonal matrix, it means that we found all clusters and each of them correspond to original class with the same number of the objects. A dendrogram of the agglomerative algorithm is a special type of tree structure that provides a convenient picture of hierarchical clustering. The dendrogram consist of layers of nodes, each node represents the cluster. Lines connect nodes representing clusters which are nested into one another.

#### 2.2.2 Partitional clustering algorithm

Partitional clustering of the graph splits the vertices of the graph into m roughly equal parts, such that the objective function defined over the edges is optimized. We use the criterion function called MinMaxCut, that was proposed recently [2]. MinMaxCut falls under the category of criterion functions that combine both the internal and external views of the clustering process and is defined as.

$$minimize \sum_{r=1}^{K} \frac{\sum_{v \in S_i, u \in S} sim(v, u)}{\sum_{v, u \in S_i} sim(v, u)}$$
(2.5)

Clustering process using this criterion function devides the vertices of the set S into clusters  $S_i$  by minimizing the sum of edges connecting the vertices of  $S_i$  to the vertices of the graph  $S - S_i$ .

#### 2.2.3 Hierarchical agglomerative clustering algorithm

Agglomerative algorithms build the solution by initially assigning each object to its own cluster and then repeatedly selecting and merging pairs of clusters, to obtain a single all-inclusive cluster. Thus, agglomerative algorithms build the tree from bottom (leaves) to top (root). The key parameter in agglomerative algorithms is the method used to determine the pairs of clusters to be merged at each step. Usually agglomerative algorithms select the most similar pair of clusters, and numerous approaches have been developed for computing the similarity between two clusters (single-link, complete-link, and UPGMA schemes) [21] [?] [12] [22] [20]. The single-link scheme measures the similarity of two clusters by the maximum similarity between the objects from each cluster. That is, the similarity between two clusters  $S_i$  and  $S_j$  is given by

$$sim_{single-link}(S_i, S_j) = max_{u \in S_i, v \in S_j} \{sim(u, v)\}$$

$$(2.6)$$

In contrast, the complete-link scheme uses the minimum similarity between a pair of objects to measure the same similarity.

$$sim_{complete-link}(S_i, S_j) = min_{u \in S_i, v \in S_j} \{sim(u, v)\}$$

$$(2.7)$$

In general, both the single- and the complete-link approaches do not work very well because they either base their decisions on limited amount of information (single-link), or they assume that all objects in the cluster are very similar to each other (complete-link approach). The UPGMA scheme (also known as group average) overcomes these problems by measuring the similarity of two clusters as the average of the pairwise similarity of the objects from each cluster. That is,

$$sim_{UPGMA}(S_i, S_j) = \frac{1}{n_i n_j} \sum_{u \in S_i, v \in S_j} sim(u, v)$$
(2.8)

#### 2.2.4 Estimation of the clustering solution

The quality of a clustering solution was measured by using two different metrics that look at the class labels of the objects assigned to each cluster. The first metric is the widely used *entropy* measure that looks are how the various classes of objects are distributed within each cluster, and the second measure is the *purity* that measures the extend to which each cluster contained objects from primarily one class. Given a particular cluster  $S_r$  of size  $n_r$ , the entropy of this cluster is defined to be

$$E(S_r) = -\frac{1}{\log Q} \sum_{i=1}^{Q} \frac{n_r^i}{n_r} \log \frac{n_r^i}{n_r}$$
(2.9)

where Q is the number of classes in the dataset,  $n_r^i$  is the number of objects of the *i*th class that were assigned to the *r*-th cluster. The entropy of the entire clustering solution is then defined to be the sum of the individual cluster entropies weighted according to the cluster size. That is,

$$Entropy = \sum_{r=1}^{K} \frac{n_r}{n} E(S_r)$$
(2.10)

A perfect clustering solution will be the one that leads to clusters that contain objects from only a single class, in which case the entropy will be zero. In general, the smaller the entropy values, the better the clustering solution is. The purity of the cluster is the fraction of the overall cluster size that the largest cluster of objects assigned to that cluster represents and defined as

$$P(S_r) = \frac{1}{n_r} max_i(n_r^i) \tag{2.11}$$

The overall purity of the clustering solution is obtained as a weighted sum of the individual cluster purities and is given by

$$Purity = \sum_{i=1}^{k} \frac{n_r}{n} P(S_r)$$
(2.12)

The larger the values of purity, the better the clustering solution is.

## 2.3 Results

#### 2.3.1 Clustering of Brodatz textures

We made first experiment with 40 textures of Brodatz [1]. From each original texture that has size of 640x640 pixels we extract small textures with size of 64x64 pixels and a sliding step 30 pixels, thus we have 400 small textures from each original texture of Brodatz and for all 40 classes we have 16000 textures. Examples of small textures are presented on Figure 2.1. Haralick features [15] are extracted for each texture. This kind of features based on co-occurrence matrixes which are the gray-level spatial dependence matrixes and computed for different direction [27].

On the first step we want to select parametres of hierarchical clustering algorithm. We cluster 16000 textures depending on the similarity functions of two objects  $(sim_{cos}, sim_{corr}, sim_{eucl}, sim_{jacc})$  and the similarity functions between two clusters for agglomerative part of the algorithm (Single – link, UPGMA, Complete – link). To control the quality of a clustering solution Entropy and Purity are used.

Results of evaluation presented in Table 2.1 and Table 2.2 show that the values of similarities of two objects have not important deviation within the similarities of clusters and the similarity UPGMA in comparison with others gives better performances of clustering with lower *Entropy* and higher *Purity*.

We use standard K-means algorithm to compare the clustering results with those of the hierarchical algorithm. For the hierarchical algorithm the similarity functions of two objects and the similarity functions between two clusters are chosen as  $sim_{eucl}$  and UPGMA respectively. For clustering results of K-means the values of quality Entropy = 0.1877 and



FIG. 2.1 - the 40 textures from Brodatz used in our experiment

Purity = 0.6935 are worse than for the hierarchical algorithm (see Table 2.1, and Table 2.2 ).

The confusion matrixes Figure 2.2 are presented as images with gray levels proportional to the rate of clustering : black is equal to 100% and white to 0%. The columns of this image

	$sim_{cos}$	$sim_{corr}$	$sim_{eucl}$	$sim_{jacc}$
Single - link	0.074	0.062	0.079	0.068
Complete – link	0.033	0.043	0.036	0.039
UPGMA	0.034	0.033	0.033	0.034

TAB. 2.1 – Entropy of clustering of 16000 textures into 40 clusters

TAB. 2.2 – Purity of clustering of 16000 textures into 40 clusters

	$sim_{cos}$	$sim_{corr}$	$sim_{eucl}$	$sim_{jacc}$
Single - link	0.844	0.878	0.842	0.864
Complete - link	0.966	0.939	0.960	0.953
UPGMA	0.960	0.964	0.965	0.963

correspond to actual classes and the rows correspond to found clusters. We can see that hierarchical algorithm gives much pure result than K-means moreover it found 30 clusters with their error of size < 0.3%.

#### 2.3.2 Clustering of textures of SPOT image

We are interested in applying the hierarchical algorithm for clustering a database of real textures. These textures were extracted from images SPOT5 which have resolution 5m and size 12000x12000 pixels (data given by CNES). We form 6 classes : city, field, sea, desert, forest and cloud. Each class has 100 textures thus we have 600 samples. Examples of textures are presented on Figure 2.3.

The size of new database of textures is lower than for the textures of Brodatz therefore we want to see whether parametres have to be changed or not. To chose these parametres for the hierarchical algorithm we made the same experiments that in Section 2.3.1 and build new tables for similarities Table 2.3 and Table 2.4. These tables show that as in Section 2.3.1 using the UPGMA we obtain better clustering solution (the lowest *Entropy* and the highest *Purity*) than using others criteria.

	$sim_{cos}$	$sim_{corr}$	$sim_{eucl}$	$sim_{jacc}$
Single – link	0.397	0.400	0.397	0.443
Complete – link	0.545	0.613	0.560	0.593
UPGMA	0.349	0.349	0.349	0.371

TAB. 2.3 – Entropy of clustering of 16000 textures into 40 clusters



FIG. 2.2 – Confusion matrix for K-means (top) and the hierarchical algorithm (bottom). The figures have been replaced with grey levels : white is zero %, black is 100 %

ib. 2.1 Tunity of clustering of 10000 tentales into 10 clust									
	$sim_{cos}$	$sim_{corr}$	$sim_{eucl}$	$sim_{jacc}$					
Single – link	0.675	0.672	0.675	0.635					
Complete - link	0.600	0.541	0.529	0.518					
UPGMA	0.732	0.732	0.732	0.720					

TAB. 2.4 – Purity of clustering of 16000 textures into 40 clusters

Unfortunately the result of clustering presented by confusion matrix even for UPGMA is unsatisfactory since the clusters are mixed. An ideal situation would have been as depicted in Section 2.3.1. One cluster would have found to group all samples of a class, and only the samples of this class. Therefore by rearranging lines of Table 2.5, it could be possible to



FIG. 2.3 – Some samples of the 6 classes of textures from SPOT image (©CNES)

obtain a diagonal matrix with all diagonal values equal to 100.

The confusion matrix ( the columns are the actual classes, the rows - the found clusters, Table 2.5 ) shows that the class "desert" mixed with the class "forest" and the class "city" mixed with the class "field". Originally each class has 100 textures but after clustering class "Cloud" has 98 textures and class "Field" has 99 textures. This 3 objects are the outliers which were obtained on the stage of building graph and which have not been taken into account on the stage of clustering. On the dendrogram of this clustering solution Figure 2.4

Clusters	Cloud	Sea	Desert	City	Forest	Field	Cluster size
1	89	0	0	0	0	23	112
2	0	89	1	0	0	0	90
3	6	0	59	0	0	5	70
4	0	0	0	39	0	16	55
5	0	10	39	0	100	4	153
6	3	1	1	61	0	51	117
Class size	98	100	100	100	100	99	

TAB. 2.5 - Confusion matrix for 6 classes

we can see that class "Sea" (it corresponds to cluster 2) is detected better from other clusters and situated much farther from other data. Cluster 1 contains data from class "Cloud" (89 objects) and "Field" (23 objects) because the class "Field" has homogeneous parts as the class "Cloud". Cluster 6 and cluster 4 have mixing of classes "City" and "Field" because some images of field have small fields like big buildings on images of class "City".

Perhaps, one of the reasons of cluster mixing is, that for this algorithm the number of textures for each class is not enough and classes are closely to each other. In order to improve the discernibility between classes we may increase the number of looked-for classes so that we obtain several pure-classes for each initial classes : for instance "forest" class could split in two classes "deciduous" and "coniferous". This will be done in the next section for one specific class, the one of "city".

	Cloud	Sea	Desert	City	Forest	Field
 11						
2  -10	U	89	T	U	U	U
-9    5	0	10	39	Ο	100	4
-8    3	6	0	59	Ο	0	5
1  7	89	0	0	0	0	23
6  4	3 0	1 0	1 0	61 39	0 0	51 16

FIG. 2.4 – Dendrogram for 6 clusters

#### 2.3.3 Clustering of urban-textures of SPOT image

We extract about 600 urban textures from different cities and different countries. Those textures include a dense part of city, isolated buildings, tanks, industrial zone, suburb, villages with fields. Examples of urban textures are on Figure 2.5.

We estimate the quality of clustering using *Entropy* and *Purity* depending on the number of clusters. Figure 2.6 shows the behavior of these two measures.

On this graph we see that after choosing the number of clusters equal 8 the *Purity* does not change and the *Entropy* smoothly changes. For clustering we take the number of clusters 8. Result is presented as the confusion matrix in Table 2.6.

The 2.5 Confusion matrix for clustering o classes									
Clusters	Buildings	City	Suburb	Tank	Village	IndustrZone	Cluster size		
1	6	37	0	0	0	5	48		
2	4	7	26	4	0	5	46		
3	1	0	0	2	1	58	62		
4	20	53	17	0	0	6	96		
5	1	1	1	14	0	46	63		
6	11	82	5	0	0	9	107		
7	0	0	0	0	92	4	96		
8	8	4	3	5	7	72	99		
Class size	51	184	52	25	100	205			

TAB. 2.6 – Confusion matrix for clustering 6 classes

Confusion matrix shows that the classes "Suburb", "Village" and "Industrial Zone" were found as clusters, but the class "Industrial Zone" is divided into 3 clusters and the cluster 5 has the class "Tank" with part of class "Industrial Zone".

It is interesting that clusters which were found (cluster 1, cluster 4, cluster 6) correspond to different cities (Los-Angeles 37 textures, Bellegarde 53 textures, Istanbul 82 textures). Examples of each city are on Figure 2.7.

The dendrogram of clustering urban textures into 8 clusters Figure 2.8 shows that cluster 4 contains 53 objects of city Bellegarde as a main part and also 20 objects of class "Buil-

![](_page_31_Picture_0.jpeg)

FIG. 2.5 – Some instances of the urban textures of SPOT5 image (©CNES)

dings" and 17 objects of class "Suburb". The cluster 2 which represent class "Suburb" (26 objects) is "nearest" to cluster 4. The cluster 1 and the cluster 6 represent the cities of Los-Angeles and Istanbul. They are very close to each other. Both these cities have long and straight streets and also both have high density buildings and constructions.

![](_page_32_Figure_0.jpeg)

FIG. 2.6 – Quality of clustering

![](_page_32_Figure_2.jpeg)

FIG. 2.7 - Urban textures of SPOT image (Los-Angeles (top), Bellegarde (middle), Istanbul

# 2.4 Conclusion

Clustering of textures was made by a hierarchical clustering algorithm which consists of partitional and agglomerative stages. Different kinds of database of textures were used for the experiments (textures of Brodatz and textures extracted from images SPOT5). It was shown, that the best similarity functions between two clusters for the agglomerative part of

	Buildings	City	Suburb	Tank	Village	IndustrZone
15						
12						
5	1	1	1	14	0	46
-11						
4	20	53	17	0	0	6
2	4	7	26	4	0	5
-14						
-13						
9						
1	6	37	0	0	0	5
6	11	82	5	0	0	9
8	8	4	3	5	7	72
10						
3	1	0	0	2	1	58
7	0	0	0	0	92	4

FIG. 2.8 – Dendrogram for 8 clusters

#### the algorithm is

*UPGMA* and the choice of the similarity functions of two objects has no great influence on the quality of the clustering solution. The hierarchical algorithm gives good performances for large databases (several thousands of examples). The hierarchical clustering approach allows to discover the structure of the data (e.g. how far one cluster is from the others) and to help interpret results of clustering (e.g. which clusters are grouped and what groups of clusters exist in data). But hierarchical trees can not always be used for interpreting the results of image clustering because satellite images of the different domains may be grouped into similar clusters, especially when groups are not well separated.

# **Chapitre 3**

# The role of the window size in the classification process

## 3.1 Objective and method

#### **3.1.1** Objectives of the study

The objective of this study is to determine the influence of the size N of the area where the measure of the vector of features is done, on the performance of the classification process.

The reference performances are those obtained in [4] for areas of size 64x64. This dimension was chosen empirically and for computational conveniences ( $N_0 = 64$  is a power of 2 which makes easy to compute FFT needed in the case of QMF and Gabor).

Smaller areas (N < 64) will probably provide more homogeneous textures and therefor could be better for fine textures like forest, fields and sea.

Larger areas (N > 64) allows to have better estimates when they cover homogeneous fields, they may also capture textures made of large grains (cities). But they will seldom guarantee a good homogeneity (if the size of the area N is equal to 100, the window covers a surface on the ground of 500x500 meters).

#### 3.1.2 Methodology

We fix the training and testing protocol all along the experiment as defined in Section 6.1 of [4]. We only consider as a variable the size N of the area for which the measure of the vector of features is done. Based on this we determine experimentally the recognition rate (or accuracy prediction) of classification process and determine the size of windows which provides the best performances in the classification process.

Having a data base of 600 images (100 for each of the 6 classes : cloud, sea, desert, city, forest and fields), the characteristics of these images are extracted by Gabor, Haralick and QMF primitive selection methods, providing 18 parameters for QMF, 24 for Gabor and 78 for Haralick.

The images in the data base are randomly distributed using a random sequence. Then using a cross validation method (the information of only three quarters of the data base images the corresponding SVM models are constructed and the classification rate measured on the rest of the images) the rate of accuracy of classification is obtained.

The range of variation of the image sizes is from 16x16 to 128x128 pixels with a step of 8 pixels. For each size the accuracy rate is as the mean of the five cross validation sets.

#### 3.1.3 Notations

The reference samples are issued from a SPOT-5 image and refereed from their upper left point. We denote as  $S_k^0$  the original sample number k, with size of the area  $N_0 = 64$ . The area is deduced from the SPOT-5 image by the following relation :

$$S_k^0(i,j) = I(i_k + i, j_k + j)$$
(3.1)

with  $i=1..N_0$ ,  $j=1..N_0$  and where  $(i_k, j_k)$  is the position of the upper left corner of the sample k in the SPOT5 image.

When taking a sample of dimension  $N_l$ , the expression becomes

$$S_k^l(i,j) = I(i_k + i, j_k + j)$$
(3.2)

with  $i=1..N_l$ ,  $j=1..N_l$ .

The vector of features measured from one original sample is denoted by  $v_i^0$ , and the one issued from the sample of size  $N_k$  as  $v_i^k$ .

For Haralick and Gabor features, the measure  $v_i^0$  is obtained by averaging local measures on the whole window  $N_0 x N_0$  as it is depicted in Section 3 of [4]. Therefore  $v_k^i$  may be derived by a similar process, but averaging on window with size  $N_k$ , whatever the value of  $N_k$  might be.

This however is not the case of the QMF and Gabor. As these two methods used FFT transform in order to extract the information, we must use windows as power of 2 only and not any size  $N_k$ . In order to deal with this we propose below 2 ways to deduce  $v_i^k$  in the case of QMF features when the size of area is not a power of two.

#### **3.1.4** Obtaining images with size as a power of 2

The first way to do this is described below.

The upper size power of two is determined and then the original image is put in the center of the new one. The empty space left are filled by a process of mirroring as shown in Fig.3.1. For instance an image with size 48 is expanded by mirroring to an image with size equal to 64.

Another way to treat the images which have not a dimension power of two is to determine the lower size window power of two and then to cut four images of this size from the original window, where each of the four images has a corner identical with the one of the original image. The process is illustrated in Fig.3.2 In this way the data base consists in four times more images than the original data base. For instance, starting with an image of size 80 x 80 pixels, we end with 4 images of sizes 64 x 64 pixels.

Remark that this second way is not giving a correct answer to our problem since each new image which was expected to carry the information contained in the 80 x 80 pixels

![](_page_37_Figure_0.jpeg)

FIG. 3.1 – Mirroring process. The mirroring process is made in two steps. The doted arrows represent the first step, followed by the second represented by the continuous arrows.

![](_page_37_Figure_2.jpeg)

FIG. 3.2 – Cutting four smaller size images. The central image as a size which is not a power of 2, All the 4 new images have a size a power of two, but they represent only one part of the original image.

image, is not containing more information that the level 64 x 64.

In the case of Gabor, the used parameters for the bank filters are as follows :

```
- scale filter = 3
```

- orientations = 4
- size of the filter 65
- For Haralick the parameters are :
- distance = 3
- For QMF the parameters are :
- number of levels = 3

Image size [pixel]	Gabor [%]	Haralick [%]	QMF I[%]	QMF II [%]
16	73.33	70.33	70.33	70.49
24	74.16	81.33	80.49	79.08
32	70.83	85.86	81.16	80.32
40	78.33	90.16	84.33	86.12
48	76.66	89.49	85.93	86.12
56	80.83	90.32	84.83	85.11
64	81.66	91.33	86.03	85.19
72	84.16	91.99	87.58	91.83
80	81.66	94.76	85.93	90.93
88	84.16	94.83	88.83	89.83
96	79.16	94.49	88.32	91.31
104	79.16	94.16	88.83	90.48
112	80.00	93.33	89.49	91.00
120	82.50	95.33	89.66	91.40
128	81.66	95.16	89.33	89.33

TAB. 3.1 – Recognition rate for different images sizes. The first line represents the sizes of the images which compose the data base. The other lines represent the accuracy of prediction expressed in percents. QMF I corresponds to the case of the windows obtained by mirroring effect, while QMF II is the cropping case.

## **3.2** Experiments and results

The results are presented in Table 3.2.

The accuracy of prediction process is defined as the percent of the images from the test set which are correctly classified.

A second experiment was carried out with the same experiment made separately on each class, to see whether we may expect different behaviours from different areas in the image. The results are given in Table 3.4 to 3.3.

Based on the data presented in Table. 3.2 the recognition rate was plotted against image size and represented in Fig.3.2

The graphical representation of the data given in Table 3.2 to 3.4 are plotted in Fig.3.4 to 3.6

#### 3.2.1 Conclusions

As expected the primitive detection based on Haralick method gives the best results when the global score (on the 6 classes) is considered. The overall recognition rate is almost constant for window sizes greater than 72, a little bit above the value proposed when starting the study (i.e. 64).

There is almost no variations above this value and their is no need to enlarge the window size.

Both overall QMF and Gabor curves remain below the Haralick curve and never improve

Image size [pixel]	Clouds	Sea	Desert	Town	Forest	Fields
16	70	88	90	67	65	44
24	81	90	96	70	60	45
32	76	85	96	73	64	58
40	87	86	99	73	63	61
48	85	85	96	77	60	60
56	89	84	99	83	66	51
64	96	84	98	81	64	59
72	95	84	97	86	64	61
80	95	86	96	88	62	64
88	97	89	99	90	66	65
96	97	86	97	92	75	66
104	97	86	97	91	67	67
112	97	85	97	91	69	58
120	99	87	96	88	68	65
128	98	86	97	89	69	72

TAB. 3.2 - Recognition rate for 6 classes, in the case of characteristics issued from Gabor

Image size [pixel]	Clouds	Sea	Desert	Town	Forest	Fields
16	92	87	54	80	69	42
24	93	91	78	84	87	55
32	94	93	80	88	90	70
40	94	91	90	96	94	75
48	96	94	88	89	94	75
56	97	91	89	94	94	77
64	93	95	89	92	96	83
72	95	97	91	93	95	81
80	97	96	93	96	98	85
88	97	95	93	96	98	90
96	97	96	95	94	95	90
104	97	95	96	93	94	90
112	97	94	95	92	92	90
120	99	93	96	95	93	90
128	100	92	96	95	96	93

TAB. 3.3 – Recognition rate for 6 classes, in the case of characteristics issued from Haralick

the rate of Haralick.

For QMF case when mirroring the signal to 128, it appeared necessary to add one more level of QMF decomposition in order to preserve results for comparison. Therefore the number of levels used is 4 instead of 3. If we maintain the number of levels to 3 then the recognition rate would drop to a value of 50 % only. In the same time when employing QMF for images of size 16 the level value is decreased by one unit becoming 2.

Image size [pixel]	Clouds	Sea	Desert	Town	Forest	Fields
16	73	90	92	70	56	41
24	84	95	96	77	78	53
32	80	95	97	80	78	56
40	90	95	95	87	81	58
48	88	96	97	82	89	60
56	88	95	97	90	85	53
64	87	96	98	79	90	62
72	94	96	97	81	93	55
80	88	95	98	79	94	58
88	92	95	99	86	95	66
96	94	94	99	82	92	69
104	92	95	99	82	94	76
112	87	94	98	89	94	75
120	93	92	100	91	91	71
128	95	92	97	89	90	75

TAB. 3.4 - Recognition rate for 6 classes, in the case of characteristics issued from QMFI

![](_page_40_Figure_2.jpeg)

FIG. 3.3 - Accuracy of prediction process for different window sizes

Comparing the curves with the separated recognition rates per classes, it can be seen that Haralick not only provides the best results in terms of recognition rate for different window sizes, but also provides the lowest the variation of recognition rate for each class. This is an important benefit, since it guarantees a stable error of recognition.

QMF I presents the largest variation with some classes very well recognized, even for very small windows qnd some other poorly.

![](_page_41_Figure_0.jpeg)

FIG. 3.4 - Recognition rate for all six classes in the case of Gabor

![](_page_41_Figure_2.jpeg)

FIG. 3.5 – Recognition rate for all six classes in the case of Haralick

In terms of how window size influences the accuracy of prediction for each class, Figures 3.4, 3.5 and 3.6 illustrate the obtained results. It can be seen that in all three cases (Gabor, Haralick and QMF I) the evolution class by class is not similar to the one in Fig.3.2.

The class of fields has the recognition rate while the classes of desert and sea are always stable with small variations and high recognition rate, while the recognition rate of the class of cloud increases with window size. The poor results obtained in the case of fields class can be explained by the diversity of images which represent this class and which are depending

![](_page_42_Figure_0.jpeg)

FIG. 3.6 – Recognition rate for all six classes in the case of QMF I

on the type of crop the region and the season.

It can be seen in Figure 3.4 and 3.6, (the case of Haralick and QMF I), that a very small image size determines a recognition rate especially for classes of fields and forest. However recognition rate improves with the window size for both classes.

A good accuracy is obtained for classes of sea, desert and cloud. The high and almost constant recognition rate can be explained by the uniformity of these textures.

# **Chapitre 4**

# Indexation par primitives de région

## 4.1 Introduction

#### 4.1.1 Objectifs

Dans le cas des images satellitaires, les méthodes d'indexation que nous avons développées jusqu'à maintenant se basent sur une extraction des caractéristiques de texture les plus efficaces pour définir une base la plus réduite possible. Cette base fournira une bonne description des textures des différentes images à classifier. Ces notions de texture sont particulièrement efficaces pour représenter les images à moyenne résolution, mais elles perdent de leur importance à très haute résolution, en particulier chaque fois que les structures géométriques s'imposent fortement. C'est très probablement le cas des images du futur satellite Pleïades.

Dans un premier travail nous avons abordé l'approche par primitives structurelles en détectant des contours [4]. Nous avons en particulier porté l'accent sur la densité, l'orientation mais également la longueur moyenne des contours. Une autre démarche consiste à s'intéresser aux régions qui composent l'image. C'est ce type d'approche qui fait l'objet de ce chapitre. Dans cette démarche, on procède tout d'abord à la segmentation de l'image en régions, généralement sur un critère d'homogénéité des niveaux de gris. On caractérise alors les régions ainsi déterminées et ce sont les descripteurs des régions qui servent à l'indexation.

L'approche que nous présentons ici consiste donc en une extraction des caractéristiques de forme et de topologie des régions déjà extraites par un outil de segmentation.

Nous présentons tout d'abord un état de l'art de la fouille d'images se basant sur des attributs régionaux. Vient ensuite le description des attributs régionaux que nous avons adoptés pour la description de la forme des régions.

#### 4.1.2 Principes

L'indexation d'images est un thème de recherche en pleine expansion, du fait de l'explosion ultra-rapide du multimédia et de la demande croissante d'accès à des bases d'images de plus en plus nombreuses et de tailles de plus en plus importantes. C'est donc souvent à ce domaine du multimédia ou de la vision par ordinateur, plutôt qu'à celui de l'image satellitaire, que nous nous référons dans ce travail. Les attributs couramment utilisés dans ce cadre sont la couleur, la texture ou encore la forme des régions. Nous écartons la couleur et la texture et nous concentrons ici sur la forme des régions qui nous semble bien susceptible de représenter l'information de régions.

Afin de capturer l'information visuelle en terme d'objets, plusieurs algorithmes de segmentation automatiques ont été introduits dans la recherche par le contenu. Une partition de l'image est construite, à partir de critères d'homogénéité locale. Le résultat de ces algorithmes est un ensemble de régions représentatif des objets contenus dans l'image.

Ensuite, en établissant des mesures sur les régions, une indexation de ces régions est faite à partir de la base d'images originale. Des systèmes récents s'appuient sur cette approche et proposent à l'utilisateur des requêtes par morceaux d'images obtenus par segmentation.

Nous présentons ici les quelques systèmes et approches récentes qui se basent sur une représentation régionale de l'image pour indexer et retrouver des régions.

## 4.2 Etat de l'art des systèmes de fouille par régions

#### 4.2.1 Blobworld

Blobworld [18] est un système de recherche visuelle qui a été développé à l'université de Californie à Berkeley. Il a été l'un des premiers à exploiter les régions comme moteur de comparaison entre images.

La ségmentation Blobworld consiste à attribuer à chaque région des caractéristques de couleur et d'autres de texture. La couleur est décrite par l'histogramme ralatif à la région. Quant à la texture, elle est définie par le contraste moyen et l'anisotropie moyenne calculés sur la région.

Ensuite, pour l'étape de recherche ou encore de comparaison, l'utilisateur doit commencer par spécifier sa région d'intérêt. Le système lui envoie alors une liste de résultats, choisie en faisant des comparaisons par couple de régions : les images retenues étant celles qui ont obtenues les meilleurs scores lors de l'évaluation des similarités région à région.

- La correspondance des deux attributs se fait de façon différente.
- couleur : la comparaison de deux histogrammes  $h_1$  et  $h_2$  se fait à l'aide de la distance quadtratique :

$$d(h_1, h_2) = (h_1 - h_2)^T A(h_1 - h_2)$$

avec  $A = (a_{ij})$  est une matrice symétrique et  $a_{ij}$  désigne le poids de similarité entre les couleurs i et j.

 texture : la distance choisie pour évaluer la ressemblance entre deux textures est la distance euclidienne entre leurs coordonnées spatiales.

#### 4.2.2 Netra

Le système Netra [24] a été développé à l'université de Californie à Santa-Barbara. Il propose lui aussi des requêtes par simples références régionales.

Chaque région Netra est caractérisée par trois attributs : la couleur, la texture et la forme. La caractéristique de couleur est définie par l'histogramme de couleur associé à la région calculé à partir d'une table de codage déjà établie dans la phase de l'apprentissage. L'index de texture est donné par l'application de filtres de Gabor sur les pixels de la région. Enfin, le descripteur de forme est obtenu suite à une décomposition de Fourier des lacets du contour de la région traitée.

Aussi, le système propose de retrouver les régions selon des contraintes spatiales particulières. Plus précisement, il est possible de préciser la zone de l'image où la région de requête doit être cherchée ou trouvée.

L'uilisateur Netra peut constituer des requêtes assez intéressantes. il pourrait, par exemple, retrouver toutes les images contenant des régions ayant la couleur de l'objet A, la texture de l'objet B et la forme de l'objet C et qui s'étale sur la partie supérieure de l'image. Les objets peuvent correspondre à des régions appartenant à des images différentes.

La correspondance des trois attributs entre deux régions à comparer se fait de façon différente.

- couleur : on désigne par  $f_C^A$  et  $f_C^B$  les vecteurs de l'attribut de couleur respectifs à deux régions A et B. Pour chaque couleur  $c_i^A$  de  $f_C^A$ , on cherche la couleur  $c_k^B$  de  $f_C^B$  la plus proche calculée selon la distance euclidienne pondérée  $d(c_i^A, f_C^B) = |p_i^A - p_i^B| d(c_i^A, c_k^B)$  où  $p_i^A$  est la contribution de la couleur  $c_i$  dans la région A. Ensuite, La distance entre deux vecteurs de couleur est définie par :

$$\sum_{i=0}^{n_A} d(c_i^A, f_C^B) + \sum_{k=0}^{n_B} d(c_i^B, f_C^A)$$

- texture : la distance choisie pour comparer deux vecteurs de texture Netra est la distance L 1.
- forme : la distance adoptée pour comparer deux vecteurs de forme dans le système Netra est la distance euclidienne.

#### 4.2.3 Simplicity

Simplicity [9] a été développé à l'université de Stanford. Il propose lui aussi une représentation régionale.

La région est aussi décrite par trois modalités : la couleur, la texture et la forme. Le descripteur de forme est dans ce cas donné par les trois premiers moments d'inertie. il est à noter qu'il s'agit de descripteurs invariants par rotation et par changement d'échelle.

L'originalité du système Simplicity tient à son algorithme de correspondance entre régions. Cet algorithme prend en compte la sur-segmentation des images : une région peut, par exemple, correspondre à plusieurs autres.

Ce schéma original permet d'envisager la comparaison d'images entières par le biais de leurs régions. L'utilisateur commence par choisir une requête globale parmi une base d'images et le système lui procure alors les images y ressemblant.

#### **4.2.4 STRICT**

Le système STRICT [29] (Système Témoin de Recherche d'Images par le ConTenu) est un moteur de recherche d'images par le contenu qui a été récemment construit.

Ce moteur repose sur une description de l'image structurée en régions. Un algorithme de segmentation automatique extrait les objets ou les parties d'objets de chaque images. Ces régions sont ensuites indexées par des descripteurs classiques du domaine (couleur, forme, position). L'ensemble de ces descripteurs constitue une base de données dans lequel le moteur va rechercher des images par similarité.

Par rapport aux systèmes précédents, STRICT inclut la configuration spatiale dans son schéma de comparaison et de correspondance. En effet, celle-ci porte généralement une information essentielle sur la sémantique de l'image.

Les régions sont proposées à l'utilisateur comme des "briques de base" pour composer ses requêtes selon ses besoins (retrouver les différents objets d'une même scène, retrouver les différentes déclinaisons visuelles d'un même objet, etc).

Afin d'évaluer la ressemblance entre la requête ainsi formée et les images de la banque de données, le moteur met en œuvre des mesures de similarité floues. Le formalisme souple et intuitif de ces mesures allié à des opérateurs de fusion adaptés à chacun des besoins de l'utilisateur permet de mettre en œuvre les requêtes de façon générique et efficace.

## 4.3 Attributs de description de la forme

Notre approche étant de trouver des attributs non texturaux et non colorimétriques, nous nous sommes donc intéressés à la forme des régions découpées de l'image.

Afin de permettre la recherche par la forme, de nombreux outils ont été développés.

#### 4.3.1 Moments d'inertie

A partir d'un objet de fonction caractéristique f, une représentation classique de sa forme consiste à en mesurer les divers moments centrés :

$$M_{mn} = \frac{1}{J^m K^n} \sum_{j=1}^{J} \sum_{k=1}^{K} (x_j - X_g)^m (y_k - Y_g)^n f(x_j, y_k)$$

Ces moments ont l'avantage d'être invariants par translation.

Les moments d'inertie (les  $M_{mn}$  pour m + n = 2) sont de plus invariants par rotation. Pour les rendre également invariants par changement d'échelle, on peut les normaliser par rapport au plus grand moment.

Les moments d'inertie décrivent bien l'allongement des formes régulières comme les ellipses ou des distributions gaussiennes.

#### 4.3.2 Moments de Hilbert

Dans [19], on propose une série de moments normalisés, dérivés des moments d'inertie. Il s'agit des moments de Hilbert utilisés en reconnaissance des formes par Hu. Ils se construisent à partir des moments normés donnés par la formule suivante :

$$N_{nm} = \frac{J^m K^n}{M_{00}^{\alpha}} M_{mn}$$

avec  $\alpha = \frac{1}{2}(m+n) + 1.$ 

Les moments de Hilbert s'écrivent comme suit :

$$\begin{split} h_1 &= N_{20} + N_{02} \\ h_2 &= (N_{20} - N_{02})^2 + 4N_{11}^2 \\ h_3 &= (N_{30} - 3N_{12})^2 + (3N_{21} - N_{03})^2 \\ h_4 &= (N_{30} + N_{12})^2 + (N_{21} + N_{02})^2 \\ h_5 &= (N_{30} - 3N_{12})(N_{30} + N_{12})[(N_{30} + N_{12})^2 - 3(N_{21} + N_{03})^2] \\ &\quad + (3N_{21} - N_{03})(N_{21} + N_{03})[3(N_{30} + N_{12})^2 - (N_{21} + N_{03})^2] \\ h_6 &= (N_{20} - N_{02})[(N_{30} + N_{12})^2 - (N_{21} + N_{03})^2] \\ &\quad + 4N_{11}(N_{30} + N_{12})(N_{21} + N_{03}) \\ h_7 &= (3N_{21} - N_{03})(N_{30} + N_{12})[(N_{30} + N_{12})^2 - 3(N_{21} + N_{03})^2] \\ &\quad - (N_{30} - 3N_{12})(N_{21} + N_{03})[3(N_{30} + N_{12})^2 - (N_{21} + N_{03})^2] \\ \end{split}$$

Ces moments ont l'avantage d'être invariants par translation, rotation et changement d'échelle.

Ainsi, avec une telle particularité, on peut caractériser, de façon assez satisfaisante les formes géométriques des régions obtenues après segmentation de l'image.

Les moments de Hilbert, calculés sur chaque région isolément, peuvent former un vecteur de caractéristiques pertinentes utile pour notre travail de classification régionale.

#### 4.3.3 Décomposition

Le contour fermé d'une région peut être considéré comme un lacet dans l'espace 2D de l'image. Cette fonction de [0, 1] dans  $\Re^2$  peut être assimilée à un signal à deux dimensions.

Une décomposition en Fourier ou en ondelettes de ce signal peut nous fournir des information sur les caractéristiques fréquentielles de la région en question.

Pour caractériser la forme d'une région particulière, on peut donc appliquer une transformation de Fourier ou d'ondelettes sur les pixels appartenant à la région choisie en considérant que les autres pixels extérieurs ont la même couleur que l'on éliminera par la suite dans la description spectrale.

#### 4.3.4 Facteur de forme

Un autre attribut est souvent utilisé pour nous informer quant à l'isotropie de l'objet. Il s'agit du facteur de forme (ou rapport iso-périmétrique), défini comme le rapport du carré du périmètre sur l'aire de l'objet, et décrit par la formule suivante :

$$R=\frac{P^2}{S}$$

où P et S désigent respectivement le périmètre et l'aire de l'objet à traiter.

Pour un cercle, ce rapport est maximum et égal à  $4\pi$ .

Cet attribut présente une certaine ambiguité quand il s'agit de contours irréguliers.

Une étape de prétraitement (lissage des contours) serait alors nécessaire pour pouvoir inclure cet attribut parmi ceux selectionnés pour classifier les images par régions.

#### 4.3.5 Conclusions

Suite à notre étude bibliographique, nous avons choisi de calculer, pour chaque région, les paramètres suivants :

- les 7 moments de Hilbert;
- le facteur de forme ; et
- la transformée de Fourier ou en ondelettes du contour.

Afin de réduire la taille du vecteur de caractéristiques relatif à chaque image, nous pouvons procéder à des calculs statistiques de chacun des attributs décrits ci-dessus.

Par exemple, si  $x_i$  désigne un attribut sur la région *i*, avec *i* variant entre 1 et N (N : nombre de régions de l'image), notre étude statistique consistera en le calcul, pour chaque image, des quantités suivantes :

- la moyenne : 
$$m = \frac{1}{N} \sum_{j=1}^{N} x_i$$

- la variance : 
$$v = \frac{1}{N} \sum_{j=1}^{N} (x_i - m)$$

– l'aplatissement skewness : 
$$\frac{\alpha}{v^{3/2}}$$
 avec :  $\alpha = \frac{1}{N} \sum_{j=1}^{N} (x_i - m)^3$ 

- le moment de Kurtosis : 
$$\frac{\beta}{v^2}$$
 avec :  $\beta = \frac{1}{N} \sum_{j=1}^{N} (x_i - m)^4$ 

## 4.4 Notre expérimentation

#### 4.4.1 Pertinence des attributs

Il nous faut maintenant procéder au choix des primitives à adopter pour décrire de façon, à la fois pertinente et parcimonieuse les images à indexer.

L'étude bibliographique nous a permis de restreindre notre choix à certaines primitives à caractère régional. Il s'agit maintenant d'extraire celles qui sont les plus pertinentes. Pour ce faire, une étude sur l'efficacité de classification de ces attributs s'avère nécessaire.

Nous décrivons ici les différentes étapes de notre étude. A l'issue de cette étude, nous établirons notre choix concernant les attributs que nous utiliserons pour évaluer le pouvoir de classification de notre méthode, ainsi que sa capacité de distinction entre les différentes classes de notre banque d'images.

#### 4.4.2 Description de la base de données

La base d'images satellitaires se compose de 33 scènes SPOT 5 HMA panchromatiques de résolution 5m par pixel. Chaque image comporte  $12000 \times 12000$  pixels. Pour permettre une visualisation aisée, chaque scène est subdivisée en 16 images  $3000 \times 3000$  pixels conduisant ainsi à une base de 528 images.

De cet ensemble de 528 images, une base d'imagettes de taille  $64 \times 64$  pixels a été extraite manuellement illustrant sans ambiguité six classes à savoir : nuages, mer, déserts, villes, forêts et champs.

Pour chaque classe, nous avons choisi de travailler sur 100 imagettes, quelques unes sont présentées sur la figure 4.1.

![](_page_49_Picture_4.jpeg)

FIG. 4.1 – Quelques exemples de la base d'images satellitaires. De haut en bas : nuages, mer, déserts, villes, forêts, champs ©CNES

Avant de passer à l'évaluation de la capacité des attributs pour distinguer les six classes, nous avons commencé par segmenter les images par la méthode de segmentation *split and merge* [26]. Nous avons choisi de travailler sur des paramètres de contraste unitaires et avec une taille minimale des régions égale à 50 pixels. En effet, ces paramètres ont permis d'obtenir une ségmentation satisfaisante des images, c'est-à-dire les régions obtenues

correpondent à peu près aux objets formant l'image.

#### 4.4.3 Les moments de Hilbert

Une première façon d'évaluer l'éfficacité des attributs est d'identifier les différents intervalles occupés par ces attributs. Ensuite, il s'agit de chercher des éventuelles disjonctions entre ces intervalles. Celles-ci pourraient bien caractériser certaines classes.

Nous avons alors commencé par tracer les différents moments statistiques correspondants aux sept moments de Hu (moyenne, variance, moment de Skewness, moment de Kurtosis) en fonction des imagettes. Ceci pour toutes les classes.

Notre objectif étant de trouver si parmi ces 28 attributs (4 moments statistiques  $\times$  7 moments de Hilbert), il existe certains avec lesquelles nous arrivons à distinguer les classes. Dans ce rapport, nous avons choisi de présenter les courbes relatives aux statistiques suivantes :

- aplatissement et kurtosis de h1 (4.2, 4.3)
- aplatissement et kurtosis de h2 (4.4, 4.5)
- aplatissement et kurtosis de h3 (4.6, 4.7)
- aplatissement et kurtosis de h4 (4.8, 4.9)
- aplatissement et kurtosis de h5 (4.10, 4.11)
- aplatissement et kurtosis de h6 (4.12, 4.13)
- aplatissement et kurtosis de h7 (4.14, ??.)

![](_page_50_Figure_12.jpeg)

FIG. 4.2 – Aplatissement de h1 calculé pour les différentes classes.

![](_page_51_Figure_0.jpeg)

FIG. 4.3 – Kurtosis de h1 calculé pour les différentes classes.

![](_page_51_Figure_2.jpeg)

FIG. 4.4 – Aplatissement de h2 calculé pour les différentes classes.

![](_page_52_Figure_0.jpeg)

FIG. 4.5 – Kurtosis de h2 calculé pour les différentes classes.

![](_page_52_Figure_2.jpeg)

 $FIG.\ 4.6-A platissement\ de\ h3\ calculé\ pour\ les\ différentes\ classes.$ 

![](_page_53_Figure_0.jpeg)

FIG. 4.7 – Kurtosis de h3 calculé pour les différentes classes.

![](_page_53_Figure_2.jpeg)

 $FIG.\ 4.8-A platissement \ de \ h4 \ calculé \ pour \ les \ différentes \ classes.$ 

![](_page_54_Figure_0.jpeg)

FIG. 4.9 – Kurtosis de h4 calculé pour les différentes classes.

![](_page_54_Figure_2.jpeg)

FIG. 4.10 – Aplatissement de h5 calculé pour les différentes classes.

![](_page_55_Figure_0.jpeg)

FIG. 4.11 – Kurtosis de h5 calculé pour les différentes classes.

![](_page_55_Figure_2.jpeg)

FIG. 4.12 – Aplatissement de h6 calculé pour les différentes classes.

![](_page_56_Figure_0.jpeg)

FIG. 4.13 – Kurtosis de h6 calculé pour les différentes classes.

![](_page_56_Figure_2.jpeg)

FIG. 4.14 – Aplatissement de h7 calculé pour les différentes classes.

![](_page_57_Figure_0.jpeg)

FIG. 4.15 – Kurtosis de h7 calculé pour les différentes classes.

# **Bibliographie**

- [1] P. Brodatz. *Textures : A Photographic Album for Artists and Designers*. Dover Publications, New York, 1966.
- [2] H. Zha C. Ding, X. He, M. Gu, and H. Simon. Spectral min-max cut for graph partitioning and data clustering. Technical report, Lawrence Berkeley National Laboratory, University of California, Berkeley, CA, 2001.
- [3] M. Campedel, J.F. Aujol, Y. Gousseau, H. Maître, and B. Luo. CNES DCT/SI/AP-04-522 - interpolation multi-échelles des attributs des images. Technical report, Télécom-Paris, Dept TSI, Paris, 2005.
- [4] M. Campedel, B. Luo, H. Maître, E. Moulines, and M. Roux et I. Kyrgyzov. Indexation des images satellitaires : Détection et évaluation des caractéristiques de classification. Technical Report 2004-D-008, Télécom Paris, Paris, december 2004.
- [5] M. Campedel, B. Luo, H. Maître, E. Moulines, M. Roux, and I. Kyrgyzov. CNES
   DU 402-01 indexation des images satellitaires : Détection et évaluation des caractéristiques. Technical report, Télécom-Paris, Dept TSI, Paris, 2004.
- [6] M. Campedel, H. Maître, and E. Moulines. CNES DU 402-02 indexation des images satellitaires : Comparaison et évaluation des classifications. Technical report, Télécom-Paris, Dept TSI, Paris, 2004.
- [7] M. Campedel and E. Moulines. Classification et sélection automatique de caractéristiques de textures. *RNTI*, C-1 :25–37, 2004.
- [8] C.C. Chang and C.J. Lin. Libsvm 2.5, 2004.
- [9] Y. Chen and J.Z. Wang. A region-based fuzzy feature matching approach to contentbased image indexing and retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(9), September 2002.
- [10] C. Cortes and V. Vapnik. Support-vector networks. *Mach. Learn.*, 20(3) :273–297, 1995.
- [11] M. Datcu, K. Seidel, and M. Walessa. Spatial information retrieval from remotesensing images. i. information theoretical perspective. *Geoscience and Remote Sen*sing, IEEE Transactions on, 36(5):1431–1445, 1998.
- [12] S. Guha, R. Rastogi, and K. Shim. Rock : A robust clustering algorithm for categorical attributes. *In Proc. of the 15th Int l Conf. on Data Eng.*, 1999.
- [13] I. Guyon. Gene selection for cancer classification using support vector machines. *Mach. Learn.*, 46:389–422, jan 2002.

- [14] R. M. Haralick, K. Shanmugan, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3(6):610–621, November 1973.
- [15] R. M. Haralick, K. Shanmugan, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, *3*(6) :610 621, 1973.
- [16] P.E. Hart. The condensed nearest neighbor rule. *ieeeinfth*, IT(14):515–516, 1968.
- [17] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning*. Springer Series in Statistics. Springer-Verlag, New York, 2001. Data mining, inference, and prediction.
- [18] C. Carson, S. Belongie, J.M Hellerstein and J.Malik. Blobworld : A system for regionbased image indexing and retrieval. In *Third Int. Conf. on Visual Information Systems*, Amesterdam, 1999.
- [19] M.K Hu. Visual pattern recognition by moment invariants. *IRE Transactions on information theory*, 1962.
- [20] A. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Englewood Cliffs, NJ, 1988.
- [21] G. Karypis, E.H. Han, and V. Kumar. Chameleon : A hierarchical clustering algorithm using dynamic modeling. *IEEE Computer*, *32*(8) :68-75, 1999.
- [22] B. King. Step-wise clustering procedures. *Journal of the American Statistical Association*, 69:86101, 1967.
- [23] K. Kira and L.A. Rendell. A practical approach to feature selection. In *International Conference on Machine Learning*, pages 249–256, 1992.
- [24] W.Y Ma and B.S. Manjunath. Netra : A toolbox for navigating large image databases. *Multimedia Systems*, 7(3) :184-198, 1999.
- [25] Stephane G. Mallat. A Wavelet tour of signal processing. Elsevier, second edition, 2003.
- [26] H. Maître. Echantillonnage. In *Le Traitement des Images*, pages 15–45. Hermes, IC2, Paris, 2003.
- [27] H.Maître M.Campedel, B.Luo, E.Moulines, M.Roux, and I.Kyrgyzov. Indexation des images satellitaires. détection et évaluation des caractéristiques de classification http://www.tsi.enst.fr/ campedel/contribution/tsireport\_2004d008.pdf. Technical report, École Nationale Supérieure des Télécommunications, Département Traitement du Signal et des Images, 2004.
- [28] P. Mitra, C.A. Murthy, and S.K. Pal. Unsupervised feature selection using feature similarity. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(3):301–312, 2002.
- [29] J.F Omhover and M. Detyniecki. Strict : an image retrieval platform for queries based on regional content. In *Proc of the International Conference on Image and Video Retrieva-CIVR2004l*, Dublin, Ireland, Aout 2004.
- [30] J. Ontrup and H. Ritter. Perceptual grouping in a neural model : Reproduced human texture perception. Technical Report SFB360-TR-98/6, Neuroinformatics group University of Bielefeld, Germany, 1998.

- [31] J. Reunanen. Overfitting in making comparisons between variable selection methods. *Journal of Machine Learning Research*, 3 :1371–1382, march 2003.
- [32] M. Robnik-Sikonja and I. Kononenko. Theoretical and empirical analysis of relieff and rrelieff. *Mach. Learn.*, 53(1-2):23–69, 2003.
- [33] J. Weston, A. Elisseeff, G. Bakir, and Fabian Sinz. The spider for matlab v1.4, 2004.
- [34] J. Weston, A. Elisseeff, B. Schoelkopf, and M. Tipping. Use of the zero-norm with linear models and kernel methods. *Journal of Machine Learning Research*, 3:1439– 1461, 2003.
- [35] Y. Zhao and G. Karypis. Evaluation of hierarchical clustering algorithms for document datasets. *CIKM*, 2002.
- [36] Ying Zhao and George Karypis. Criterion functions for document clustering : Experiments and analysis. Technical report, Department of Computer Science, University of Minnesota, Minneapolis, MN, 2001.