

UNIVERSITÉ DE NANTES
ÉCOLE POLYTECHNIQUE DE L'UNIVERSITÉ DE NANTES

ÉCOLE DOCTORALE

SCIENCES ET TECHNOLOGIES
DE L'INFORMATION ET DES MATHÉMATIQUES

2008

Thèse de Doctorat de l'université de Nantes

Spécialité : Informatique

Présentée et soutenue publiquement par

Najlae IDRISI

le 18 octobre 2008

Faculté des sciences de Rabat

**La navigation dans les bases d'images :
prise en compte des attributs de texture**

Jury

Président	: Pr. Driss ABOUTAJDINE	Faculté des sciences, Rabat
Rapporteurs	: Saïd BENAOUICHA, Pr. Catherine BERRUT, Pr.	Faculté des sciences et techniques, Tanger Université Joseph Fourier, Grenoble
Examineurs	: Noureddine MOUADDIB, Pr. Béatrice DAILLE, Pr.	École polytechnique de l'université de Nantes Université de Nantes

Directeur de thèse : Pr. Driss ABOUTAJDINE & Pr. José MARTINEZ

Laboratoire : LABORATOIRE D'INFORMATIQUE DE NANTES ATLANTIQUE. 2, rue de la Houssinière,
F-44322 NANTES CEDEX 3

N° ED 0366-370

**LA NAVIGATION DANS LES BASES D'IMAGES : PRISE EN
COMPTE DES ATTRIBUTS DE TEXTURE**

Using Textures to Browse Image Databases

Najlae IDRISSI



favet neptunus eunti

Université de Nantes

Najlae IDRISSE

La navigation dans les bases d'images : prise en compte des attributs de texture

xxi+170 p.

Ce document a été préparé avec $\text{\LaTeX}2_{\epsilon}$ et la classe `these-LINA` version 0.92 de l'association de jeunes chercheurs en informatique LOGIN, Université de Nantes. La classe `these-LINA` est disponible à l'adresse :

<http://www.sciences.univ-nantes.fr/info/Login/>

Impression : memoire_idrissi2_final2008.tex - 1/12/2008 - 22:05

Révision pour la classe : \$Id: these-LINA.cls,v 1.3 2000/11/19 18:30:42 fred Exp

Et ma réussite ne dépend que d'Allah. En Lui je place ma confiance, et c'est vers Lui que je reviens repentant.

— Coran, Sourat-Houd (89).

Résumé

Ce travail de recherche entre dans le cadre des systèmes de recherche d'images par le contenu, en particulier la recherche par la texture. Le but de ce travail est de permettre à l'utilisateur de naviguer dans de grande base de données d'images sans formulation de requêtes en un langage d'interrogation spécifique. Pour atteindre cet objectif, nous avons réparti le travail en deux grands volets.

Le premier volet concerne l'extraction et l'identification d'un modèle de texture composé d'attributs pertinents. Pour atteindre cet objectif, nous avons proposé d'étudier deux modèles de texture : les matrices de co-occurrences et les attributs de Tamura. La sélection et la validation du modèle caractéristique ont été faites à partir de plusieurs applications que nous avons proposées dans le cadre de cette thèse après réduction de la dimension de l'espace de représentation des modèles de texture.

Ensuite, la navigation s'effectue à l'aide de treillis de Galois avec une interface HTML tout en passant par une phase d'interprétation du modèle de texture numérique en un modèle sémantique. Le problème de transcription du numérique au sémantique est considéré comme un problème de discrétisation des valeurs numériques continues. Un autre problème se manifeste lorsque la taille de la base des images augmente, les performances du système de navigation se dégradent. Pour pallier à ce problème, nous proposons de créer des résumés qui de plus permettent de focaliser la recherche et la navigation sur un ensemble d'images cibles et non pas sur toute la base.

Mots-clés : recherche d'image par le contenu, texture, analyse et extraction, fossé sémantique, navigation, résumés, logique floue.

Abstract

This work contributes to the field of Content-based Image Retrieval (CBIR) particularly texture-based retrieval. The main goal of this work is to enable the user to navigate through a large image database without making any query in specific language. To achieve this goal, we divided the work into two main parts.

The first part involves the extraction of a texture model made of relevant attributes. We proposed to study two models of texture: the co-occurrence matrices and Tamura's attributes. The selection and validation of the model features are based on several applications that we have proposed in this thesis after reducing the dimension of the representation's space.

Then, navigation is achieved using Galois' lattices with a simple HTML interface while passing through a phase of interpretation of numerical model texture into a semantic model. The problem of the transcription from the numerical values to the semantics is regarded as a problem of discretisation of continuous attributes. Another problem occurs when the size of the database of images increases: the performance of the navigation system are deteriorating. To overcome this problem, we propose to use techniques of summarisation to create summaries that help users to navigate through target collections instead of the whole database.

Keywords: Content-based image retrieval, CBIR, texture, analysis and extraction, semantical gap, browsing, summarisation, fuzzy logic.

Remerciements

Je tiens tout d'abord à préciser le cadre de cette thèse. Ce travail a été effectué dans le cadre d'une co-tutelle entre le laboratoire marocain GSCM-LRIT (laboratoire de Recherche en Informatique et Télécommunications) de la Faculté des Sciences de Rabat sous la direction du Professeur Driss Aboutajdine et le laboratoire français LINA (Laboratoire d'Informatique de Nantes Atlantique) sous la direction du Professeur José Martinez.

Ensuite, je tiens à remercier Mme Catherine Berrut et M. Said Benaouicha pour avoir consacré du temps à lecture de cette thèse ainsi pour avoir soumis leur précieux jugement sur la qualité et le contenu de ce travail.

Je remercie également M. Driss Aboutajdine pour son suivi et ses encouragements tout au long de ce travail de thèse de doctorat. Mes remerciements vont aussi à José Martinez pour son encadrement et son suivi rigoureux et patient durant ces années de thèse.

Je n'oublie pas de remercier M. Noureddine Mouaddib d'avoir accepté d'assister à cette commission d'examen et pour son soutien apporté depuis mon premier stage au laboratoire LINA. Je remercie aussi Mme Beatrice Daille d'avoir accepté de participer à ce jury.

Mes sincères remerciements et amitiés vont aussi à Guillaume Raschia avec qui j'ai eu tant de discussions riches.

Merci à mes parents et mes frères pour leur soutien durant toutes ces années d'études: je ne saurais être qu'infiniment reconnaissante quant aux sacrifices qu'ils ont consentis. Merci à tous les collègues du laboratoire GSCM-LRIT pour leur amitié et bonne humeur qui ont égayé ma vie au laboratoire.

Enfin merci à ceux et celles que je n'ai pas pu citer, mes sincères amitiés et remerciements.

Sommaire

1	Introduction	1
I	Recherche d'images par la texture	
	chapter2 État de l'art sur la recherche d'images par le contenu11chapter.2	
3	Processus de description de la texture	21
4	Applications	55
5	Expérimentations	73
II	Navigation sémantique dans des bases d'images	
	chapter6 Interprétation sémantique des textures117chapter.6	
7	La navigation dans une base d'images	145
8	Conclusions	157
	Bibliographie	161
	Liste des tableaux	183
	Table des figures	185
	Table des matières	187

CHAPITRE 1

Introduction

1.1 Recherche par contenu

De nos jours, l'image est devenue l'information multimédia la plus répandue et la plus utilisée dans différents domaines y compris dans notre vie quotidienne. Par exemple, lors d'un journal télévisé diffusé en une langue étrangère que vous ne maîtrisez pas, il est possible de deviner les différents événements survenus sans avoir recours au texte. Un autre exemple provient du domaine militaire où, pour analyser une zone, il est indispensable de disposer de cartes et d'images. Aussi, dans le domaine médical, il est fortement constaté qu'une description verbale est insuffisante pour établir un diagnostic exact et précis, il est toujours recommandé de disposer de radios ou d'autres types de clichés qui ne sont qu'un des types d'images numériques.

Le nombre d'exemples illustrant l'importance des images est bien plus large encore et cela peut être constaté par le volume des bases d'images indexées par les moteurs de recherche qui commencent à prendre de l'espace dans notre vie professionnelle et personnelle. Une estimation, datée de 1999 [176], répertorie 180 millions d'images indexées et réparties sur 800 millions de pages disponibles sur la Toile. Ce chiffre atteint, en 2007, selon le service scientifique de technologie, environ 15 milliards d'images sur la Toile. Le moteur de recherche Google Images lui seul a indexé 24 milliards d'images en novembre 2005, passé à 1.000 milliards d'images en juillet 2008. La banque d'images la plus populaire Flickr a disposé en mars 2005 de 5 millions d'images, actuellement en février 2008, ce chiffre atteint 2,2 milliards d'images. Un autre exemple d'agences de vente et d'achat d'images tels que Fotosearch et Fotolia qui disposent respectivement en 2008 de 3,1 et 4 millions d'images disponibles en ligne. Les images sont de types variés tels que sport, gastronomie, art, paysage, etc.

Ainsi, sous toutes ses formes, l'image attire l'attention des professionnels mais aussi du grand public. Parmi les causes de cette expansion, nous pouvons citer:

- la forte disponibilité des moyens d'acquisition et de numérisation, de moins en moins chers et de plus en plus performants – tels que les appareils photos et caméscopes numériques, les scanners, etc. – contribuent à l'augmentation du nombre d'images numérisées ;
- l'évolution et la diversité des méthodes et des médias de stockage et d'archivages des données favorisent la croissance des banques d'images;
- l'avènement des réseaux de transmission – la Toile, le système pairs à pairs, etc. – permet un large partage de cette information autant à l'échelle locale que mondiale ;
- les outils d'analyse et de traitement d'images – codage, compression, etc. – contribuent à une production continue et perpétuelle d'une masse importante d'images numériques.

Les progrès dans ces domaines contribuent énormément à l'augmentation du nombre, de la taille, de l'utilisation (plusieurs champs d'applications tels que médical, environnement, sécurité, etc.) et de la disponibilité de ces bases d'images. Pour cela, de nouveaux outils sont nécessaires à l'utilisateur pour la création, la gestion et la recherche des images à partir de ces bases.

Le problème ainsi posé est la recherche d'une ou plusieurs images cibles au sein de cette masse

considérable d'images numériques. Sans outils efficaces, cette tâche devient difficile à impossible. Aussi, le domaine de la recherche d'images est devenu un domaine très actif dans la communauté internationale à partir des années soixante-dix grâce au développement des systèmes de gestion des bases de données et les outils de vision par ordinateur. Deux approches sont reconnues dans la littérature. L'approche la plus ancienne, apparentée à la recherche documentaire et encore utilisée jusqu'à nos jours, est la recherche textuelle. Il s'agit d'annoter manuellement chaque image par un ensemble de mots clés tels que le titre, l'auteur, la date d'acquisition, le lieu, les objets visibles, etc., puis d'utiliser un système de gestion de bases de données basé sur ces annotations pour effectuer la recherche. Parmi les travaux les plus représentatifs dans ce domaine, nous trouvons [255, 35, 149, 248]. Malgré le grand succès de cette approche pour la recherche documentaire, l'annotation textuelle souffre de plusieurs inconvénients dans le cas des images, parmi lesquels, nous pouvons citer :

- non-exhaustive car la liste des mots clés ne peut pas couvrir la totalité ni du contenu ni de la sémantique de l'image ;
- fastidieuse lorsqu'elle est utilisée pour de grands volumes des bases d'images ;
- coûteuse en terme de temps de réponse à une requête (comparaison entre liste bases de données et liste requête) ;
- subjective, le même contenu d'une image peut être interpréter différemment par plusieurs personnes ou par la même personne avec un décalage dans le temps ;
- polysémie, un mot clé peut désigner divers objets différents sémantiquement et visuellement (ex. : le terme « avocat » peut signifier la personne exerçant la profession de plaider ou bien il peut s'agir du fruit comestible de l'avocatier).

C'est pourquoi, à partir de 1990, la recherche scientifique s'est penchée sur le développement d'un autre moyen de recherche d'images qui soit automatique et efficace. L'approche alternative qui est apparue depuis plus d'une dizaine d'années est la recherche d'images par le contenu visuel connue sous le nom anglais de CBIR pour *Content-based Image Retrieval*. Plusieurs travaux ont été publiés dans ce cadre [105, 321, 252, 59, 273]. Cette approche consiste à décrire le contenu visuel de l'image par un ensemble de paramètres extraits automatiquement de l'image. Chaque image étant identifiée non pas par une liste de mots clés mais par un vecteur de valeurs de ses paramètres (aussi appelé « signature » ou *feature vector* en anglais). Ces paramètres doivent être choisis avec précaution afin d'être discriminants entre les différentes images présentes dans la base de données. La recherche s'effectue alors en calculant une mesure de similarité entre la signature d'une requête et les signatures des images indexées dans la base. Les images les plus proches (images ayant de valeurs paramétriques similaires) à l'image requête sont alors retournées à l'utilisateur comme étant les images les plus proches visuellement à la requête.

Cette nouvelle approche, basée sur le contenu, pallie les problèmes posés par la recherche textuelle, permet d'améliorer des applications interrompues et contribue aussi à faire émerger de nouvelles applications dans divers domaines.

L'évaluation d'une réponse requête dans le cadre de la recherche d'images par le contenu est effectuée en terme de *pertinence* des images retournées. La pertinence est analysée sur le plan *visuel* et/ou *sémantique* selon le type de la requête posée :

1. Si la requête fait référence aux objets proprement visuels dans l'image, les images renvoyées doivent comporter les mêmes objets (couleurs, textures, etc.) que la requête ; ainsi l'aspect visuel l'emporte sur la sémantique. Par exemple, si la requête est : « Je cherche des zèbres », les images obtenues doivent contenir l'objet « zèbre » pris sur différentes vues et scènes (sémantiquement différentes). L'ensemble des images retournées est dit visuellement homogène. Cette recherche est dite « recherche primaire ».

2. Si la requête traduit un sens pour l'utilisateur tel que joyeux, triste, ensoleillé, contrasté, etc., les images retournées doivent correspondre à ce sens. Par exemple, si la requête formulée est: « Je cherche des images ensoleillées », le système doit être capable de traduire le terme « ensoleillé » en images qui reflètent cet aspect. On ne doit pas avoir des images de nuit. Il est fort probable que les images ainsi retournées soient visuellement hétérogènes mais sémantiquement homogènes. Si on prend le même exemple, on peut avoir des photos de plages, d'anniversaire, d'une randonnée, etc. mais il ne serait pas acceptable d'avoir des photos d'une soirée dansante. Il s'agit dans ce cas d'une « recherche sémantique ».

Les points principaux à prendre en compte pour l'évaluation d'une recherche sont donnés par ordre analytique :

1. Premièrement, il faut déterminer l'objectif de l'application (« qu'est ce qu'on cherche ? ») qui est lié au type des requêtes à formuler. Trois grandes catégories de recherche ont été proposées [273, 54]:
 - la recherche ciblée (*target search*) qui consiste à chercher une image particulière précisée par l'utilisateur ;
 - la recherche par catégorie (*category search*) a pour objectif de retourner un ensemble d'images appartenant à une même catégorie donnée ;
 - la recherche associative (*search by association*) concerne le cas où l'utilisateur n'a aucune idée sur l'image à chercher ; c'est en explorant la base d'images, en moyen d'une interaction système-utilisateur, que ce dernier peut affiner sa recherche et atteindre sa cible.
2. Deuxièmement, il faut choisir un contenu visuel pour la caractérisation des images. Parmi les contenus existants, les plus utilisés sont la couleur, la texture et la forme.
3. Enfin, il faut disposer d'une mesure de similarité pour la recherche des images les plus proches à la requête ; plusieurs distances ont été proposées et utilisées : distance Euclidienne, Mahalanobis, etc.

1.2 Le fossé sémantique

Mais avant tout processus de modélisation des attributs en concepts haut-niveau, la première question qui devra être posée est « *qu'est ce que la sémantique d'une image ?* » La réponse n'est pas unique et très vaste. L'image analysée au niveau sémantique peut se faire, d'une part, en terme d'objets constituant l'image et donc non seulement en terme d'attributs de bas niveau et, d'autre part, en terme de liens entre objets en faisant appel à des connaissances externes propres à l'utilisateur.

Prenons l'exemple de la figure 1.1(a), les objets composant cette image sont : éléphants, eau, arbres, ciel, etc. En tenant compte des liens entre objets, l'image pourra correspondre au « désert africain », « deux éléphants se livrant un combat », etc. De même pour la figure 1.1(b), l'image est composée d'oranges, tasses de cacao, gâteaux. En combinant la sémantique des différents objets, il peut s'agir d'un « petit déjeuner doux ».

De ces constats, nous supposons que la sémantique d'une image est une notion à trois vues :

- la première vue est liée aux connaissances et à la perception d'objets que l'utilisateur a pour l'image ;
- la deuxième vue correspond aux relations spatiales qui peuvent exister entre les objets de cette image ;
- la dernière vue vise l'objectif de l'utilisateur lorsqu'il regarde cette image. Par exemple, dans [6], selon la question posée à un observateur d'une image, le parcours, le regard posé et les informations



(a) Image d'éléphants



(b) Image d'un petit déjeuner

Figure 1.1 – Illustrations du fossé sémantique

recueillies dans l'image varient. Il en est de même en recherche d'images similaires où une requête peut signifier différents buts selon le contexte [257].

La définition de référence du fossé sémantique est celle fournie par Smeulders dans son état de l'art [273] :

*“The semantic gap is the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation.”*¹

Dans une tentative de définition de la sémantique des images, Eakins [71] classe les requêtes utilisateur en trois niveaux selon leurs degrés d'abstraction [189] :

1. les requêtes à ce niveau sont de types requête par image exemple (*trouver les images similaires à celle-ci*). La recherche est basée sur les attributs de bas niveau et une mesure de similarité entre images et la requête. La majorité des systèmes SRIC appartiennent à ce niveau.
2. les requêtes à ce niveau se focalisent sur les objets (*trouver les images avions*). La recherche dans ce cas consiste à identifier les objets de l'image par des attributs intégrant une certaine inférence logique. La sémantique des objets est considérée comme étant la classe des objets et les relations spatiales entre eux.
3. les requêtes à ce niveau reflètent un niveau haut d'abstraction (*trouver les images de fêtes de fin d'année*). La recherche ici traduit un degré élevé d'abstraction voire même un raisonnement sur le sens et le but des objets ou scènes identifiés dans l'image.

La classification faite par Eakins est très utile pour décrire les capacités et les limites des systèmes de recherche d'images. Notons que les niveaux 2 et 3 reflètent la recherche d'images purement sémantique [105]. La différence entre le niveau 1 et 2 correspond au fossé sémantique défini par Smeulders [273].

Par conséquent, pour supporter des requêtes de haut niveau, les systèmes de recherche d'images standard doivent disposer d'un module sémantique qui permet la réduction du fossé entre la description de bas niveau des images et la riche sémantique qu'elles contiennent [331, 273].

1.3 Définition du problème

Le problème auquel nous nous intéressons dans ce travail est la navigation dans des bases d'images généralistes. Ainsi, ce travail s'inscrit dans le cadre de la recherche par catégorie en se basant particulièrement sur la texture dont le but est de permettre à l'utilisateur de retrouver un ensemble d'images appartenant à la même classe cible d'une manière facile et efficace.

La clé de la réussite de ce travail réside :

1. en premier lieu dans le choix des attributs pertinents de la texture qui permettent de mieux caractériser les images sur les deux niveaux : primaire et sémantique ;
2. ensuite, le choix d'une mesure de similarité qui permet de regrouper les images les plus proches partageant les mêmes caractéristiques (primaires et sémantiques) et d'écarter celles qui diffèrent.

Le but de cette thèse est donc double :

1. d'une part, développer un modèle de texture répondant aux contraintes suivantes :

¹ « Le manque de concordance entre les informations qu'on peut extraire des données visuelles et l'interprétation qu'on en fait pour un utilisateur dans une situation déterminée. »

- (i) la taille du modèle de texture doit être faible afin d'éviter les problèmes de temps de calcul sans négliger la pertinence des paramètres extraits ;
 - (ii) le modèle texture doit être discriminant entre différents types de textures ;
 - (iii) il doit être interprétable en langage naturel et donc aisément compréhensible par l'humain (tenir compte de la sémantique de l'utilisateur) ;
 - (iv) les attributs du modèle de texture doivent être invariants ;
2. d'autre part, proposer un modèle sémantique qui permet de réduire le fossé sémantique entre la recherche primaire et la recherche sémantique en tenant compte de la sémantique de l'utilisateur.

L'évaluation de la performance du modèle de texture retenu sera faite en terme de mesure de la capacité du système à retourner des images similaires à la requête, à son efficacité à classer de nouvelles images et à regrouper les images similaires en même catégories.

1.4 Plan de la thèse

La première partie de la thèse est consacrée à la recherche d'images par texture.

Tout d'abord, dans le chapitre 2, nous proposons un état de l'art sur la recherche d'images par contenu en général, les contenus visuels par lesquels une image peut être décrite, les approches d'analyse correspondantes et les mesures de similarité les plus courantes, ainsi que les différents systèmes académiques et industriels existants. Ensuite, nous introduisons notre démarche pour la description de la texture dans le chapitre 3. La caractérisation de la texture est une étape « hors-ligne » réalisée automatiquement, qui comprend, dans notre cas, la sélection des bases d'apprentissage (base fiable et comprend une diversité de texture), la sélection des attributs pertinents de la texture par les méthodes de réduction de l'espace de représentation des attributs et nous terminons ce chapitre par présenter les applications de test des performances qui seront détaillées dans le chapitre 4.

Le chapitre 5 illustre les différentes expérimentations menées pour cette première partie d'identification d'un modèle compact et pertinent pour la caractérisation de la texture et ainsi évaluées dans le cadre de la recherche dans les bases d'images texturées (section 3.2).

La seconde partie de la thèse se focalise sur la recherche sémantique des images.

La notion du fossé sémantique est une notion encore nouvelle [273] et difficile à normaliser. Nous commençons cette seconde partie par introduire les différentes approches de réduction du fossé sémantique existantes dans la littérature de la recherche d'images à savoir le formalisme des ontologies, les systèmes de rétroaction et les approches d'apprentissage (supervisé et non-supervisé) (chapitre 6).

Ensuite, dans le chapitre 7, nous présentons notre proposition pour la réduction du fossé sémantique entre descripteurs bas-niveau et concepts haut-niveau en utilisant une approche de discrétisation de l'espace numérique des attributs de texture. La recherche sémantique dans ce travail s'effectue par le processus de navigation dans une hiérarchie préalablement construite à l'aide des treillis de Galois. L'interface offerte à l'utilisateur est une interface HTML facile à manipuler par la majorité des utilisateurs non-experts.

1.5 Liste des publications

Journal

N. Idrissi, J. Martinez, D. Aboutajdine , "Bridging the Semantic Gap for Texture-based Features for Image Retrieval and Navigation", in Special Issue on Multimodal Information Retrieval at the Journal of

MultiMedia (JMM)(second revision).

Chapitre de livres

N. Idrissi, J. Martinez, D. Aboutajdine, "Selecting a Discriminant Subset of Co-occurrence Matrix Features for Texture-based Image Retrieval", in LNCS 2005, editor Springer Berlin / Heidelberg, 3804(696-703), 2005.

Publication dans des actes de congrès internationaux avec comité de lecture

- N. Idrissi, José Martinez, Driss Aboutajdine, "Semantic Texture Interpretation for Image Retrieval and Navigation", 4th International Symposium on Image/Video Communications over fixed and mobile networks (ISIVC 2008), Espagne.
- N. Idrissi, J. Martinez, D. Aboutajdine. , " Selecting a Discriminant Subset of Co-occurrence Matrix Features for Texture-based Image Retrieval ", in: International Symposium on Visual Computing (ISVC 2005), Lake Tahoe, Nevada, USA, 2005.
- N. Idrissi, J. Martinez, D. Aboutajdine. , " Selection of a Discriminant Subset of Co-occurrence Matrix Features for Image Search-by-Content ", in: IEEE International Conference on Computer Systems and Information Technology (ICSIT 2005), Algiers, Algeria, 2005.
- N. Idrissi, J. Martinez, D. Aboutajdine. , " On the Use of Co-occurrence Matrices for Texture Description in Image Databases ", in: International Conference on Modeling and Simulation (ICMS 2005), Marrakech, Morocco, 2005.

PARTIE I

Recherche d'images par la texture

État de l'art sur la recherche d'images par le contenu

Dans ce chapitre, nous allons faire un tour d'horizon des principales approches de la recherche d'images par le contenu. Étant donné la diversité et le nombre important des travaux existants, nous n'allons rappeler que les références les plus pertinentes du domaine.

Ce chapitre est divisée en deux parties : les approches et les systèmes de recherche d'images par le contenu.

Dans la partie 2.2, nous verrons les différentes méthodes d'extraction de l'information visuelle nécessaire à la description des images, dénommés les « descripteurs ». Ensuite, vient la question de représentation de ces descripteurs sous la forme d'une signature (ou index) dans une base de données. Nous terminons cette partie par un ensemble de mesures de similarité entre les images et la requête les plus utilisées.

La seconde partie de ce chapitre sera consacrée aux différents systèmes de recherche d'images par contenu commerciaux et scientifiques les plus répandus.

2.1 Introduction

La figure 2.1 illustre l'architecture standard d'un système de recherche d'images par le contenu. Trois phases successives, non séparables, sur lesquelles reposent la recherche par le contenu sont :

- l'extraction ;
- l'indexation ;
- la recherche.

L'extraction et l'indexation se déroulent en mode « hors-ligne », transparent pour l'utilisateur. L'intervention effective de l'utilisateur se manifeste en mode « en-ligne » lors de la recherche proprement dite car il doit initier par la requête une description de la cible ou des indications – directes ou indirectes – sur les caractéristiques recherchées.

La performance d'une recherche d'images par le contenu est mesurée en terme de deux contraintes principales.

La première contrainte concerne le temps de réponse à une requête utilisateur qui doit être rapide. Cette contrainte est fortement liée au nombre de descripteurs utilisés pour indexer les images ainsi qu'à l'organisation interne des signatures (séquentielle, hiérarchique, etc.). Si le nombre de descripteurs est important alors une recherche séquentielle des signatures dans une base de données importante (plusieurs millions d'images) se montre gourmande en temps de réponse et donc inefficace.

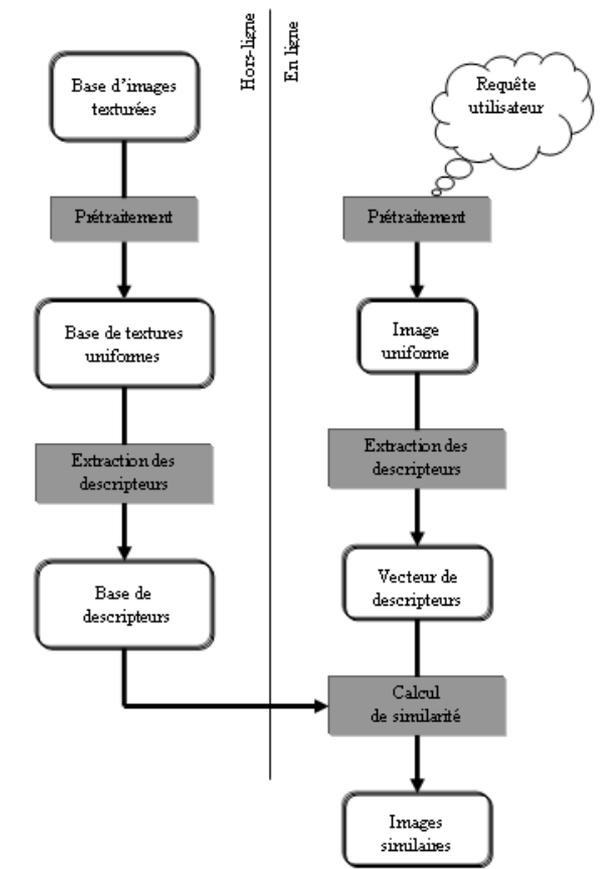


Figure 2.1 – Schéma standard d'un système de recherche d'images

La seconde contrainte est liée à la satisfaction de l'utilisateur. Un système de recherche est dit utile et performant s'il répond favorablement aux attentes des utilisateurs. Le choix des descripteurs pertinents décrivant au mieux le contenu visuel des images ainsi qu'une mesure de similarité adéquate entre la description de la requête et celles des images de la base conditionnent considérablement la qualité de la réponse d'un système de recherche d'images par contenu.

2.2 Modèles classiques de recherche d'information

Un modèle de représentation d'information définit dans le cadre de la recherche des images le modèle de représentation des descripteurs des images. Les modèles standards apparentés de la recherche documentaire sont très variés. Nous nous limitons ici à la présentation des trois modèles les plus connus :

- booléen ;
- vectoriel ;
- probabiliste.

2.2.1 Modèle booléen

Le modèle booléen représente les images et les requêtes par une liste de descripteurs reliés entre eux par des opérateurs logiques (« et », « ou », « et non »). Par exemple, une image I peut être représentée par une liste de descripteurs reliés par l'opérateur « et » (\wedge) :

$$I = di_1 \wedge di_2 \wedge \dots \wedge di_n \quad (2.1)$$

Plus généralement, une requête Q est représentée par une liste de descripteurs reliés par l'ensemble des opérateurs logiques :

$$Q = dq_1 \wedge dq_2 \vee \dots \neg dq_n \quad (2.2)$$

La correspondance entre les images et la requête se traduit par une implication logique :

$$I \rightarrow Q \quad (2.3)$$

Ainsi, le système effectue une classification binaire en deux classes, positive et négative, correspondant respectivement aux images qui satisfont la requête et à celles qui ne la satisfont pas.

Le modèle est simple, implémenté efficacement dans le monde des documents textuels. Il présente des difficultés importantes d'adaptation au monde des documents visuels. En particulier, les descripteurs sont généralement des valeurs numériques et une simple comparaison par égalité n'a plus grand sens. Par suite, la séparation rigide entre documents retrouvés et documents écartés est beaucoup trop stricte.

2.2.2 Modèle vectoriel

Dans le modèle vectoriel, les images I de la base et l'image requête Q sont représentées par un vecteur de descripteurs dans un espace d'attributs à n dimensions. Les éléments de ce vecteur représentent les pondérations des descripteurs utilisés :

$$I = (wi_1, wi_2, \dots, wi_n) \quad (2.4)$$

$$Q = (wq_1, wq_2, \dots, wq_n) \quad (2.5)$$

La correspondance entre les images de la base et la requête s'effectue en terme d'une fonction de similarité (ou de dissimilarité) entre leurs vecteurs.

Dans cette thèse, nous retiendrons ce modèle de représentation.

2.2.3 Modèle probabiliste

Dans le modèle probabiliste, une probabilité de pertinence de l'image, en réponse à la requête, est attribuée à chacun des descripteurs. Cela suppose qu'il existe un sous-ensemble d'images R pertinentes que l'utilisateur veut retrouver parmi celles disponibles, les autres images \bar{R} étant considérées comme non-pertinentes.

Si $P(R|\vec{Q})$ est la probabilité que l'image I soit pertinente pour la requête Q et si $P(\bar{R}|\vec{Q})$ est la probabilité que l'image I ne soit pas pertinente pour la requête Q , alors la similarité entre l'image I et la requête Q est exprimée par:

$$\text{sim}(I, Q) = \frac{P(R|\vec{Q})}{P(\bar{R}|\vec{Q})} \quad (2.6)$$

Dans la suite de ce chapitre, nous abordons les trois points fondamentaux pour la recherche d'images par le contenu faisant appel au modèle vectoriel.

2.3 Extraction des attributs

La phase d'extraction est une phase clé de la recherche d'images par le contenu, sur laquelle reposent les étapes suivantes du processus de la recherche. Selon le type d'application et les objectifs visés, le contenu peut représenter du texte (mots clés et annotations) ou bien des attributs visuels (couleur, forme, texture, etc.). D'autres attributs tels que les objets visuels, les croquis et les relations spatiales sont considérés dans le standard MPEG-7. Dans ce qui suit, nous nous limitons aux aspects visuels.

2.3.1 Attributs de couleur

La couleur est l'un des premiers descripteurs visuels utilisé pour la recherche d'images auquel les travaux de recherche s'y sont intéressés [310]. Cet intérêt provient de la robustesse que la couleur montre par rapport aux changements d'échelle et d'orientation, son indépendance au fond, à la taille et aux occlusions des images [200]. L'attribut couleur est primordiale dans notre quotidien pour décrire l'environnement et pour identifier les objets d'intérêt dans une scène [259] et, aussi dans divers domaines d'applications tels que les multimédias (indexation, classification, etc.), la vision par ordinateur, le traitement des signaux biomédicaux, etc. Les points clés du thème de la recherche par couleur sont le choix du meilleur espace colorimétrique, le mode de représentation de la couleur dans cet espace [286, 310] et l'étude de l'invariance aux conditions d'illumination et de prise de vue [102].

Plusieurs travaux ont été réalisés dont l'objectif est d'identifier les espaces colorimétriques les plus discriminants [286, 318]. L'abondance de la littérature sur cette thématique a montré qu'il n'existe pas d'espace couleur « idéal ». L'espace RGB est l'espace standard le plus utilisé. Néanmoins, cet espace est sensible aux variations de luminance. L'espace HSV par sa composante teinte permet de résister aux conditions d'éclairage et de prise de vue [102, 95, 94]. D'autres espaces définis comme perceptuellement uniformes tels que CIELab et CIELuv servent également à la caractérisation de la couleur.

2.3.1.1 Signatures de la couleur

Histogramme Dans le domaine de recherche d'images basé sur la couleur, l'histogramme de couleur est la représentation la plus populaire et largement utilisée. Statistiquement, il représente la probabilité jointe des intensités des trois canaux de couleur (RGB, HSV). A une constante près de normalisation, l'histogramme constitue une approximation de la densité de probabilité associée à l'image, considérée comme une variable aléatoire.

Swain et Ballard [288] ont été les premiers à utiliser l'histogramme de couleur pour la recherche d'images par le contenu. Depuis, plusieurs travaux se sont succédés et ont retenu cette représentation. Le succès de l'histogramme résulte de sa simplicité de calcul et d'utilisation, sa robustesse à la rotation, aux changements d'échelle et aux occlusions.

Cependant, l'utilisation de l'histogramme pose deux problèmes. Le premier concerne la perte de l'information spatiale (texture, forme), caractéristique importante pour décrire le contenu globale des images. Le second inconvénient est lié à la quantification de l'espace des attributs.

Alternatives de l'histogramme Pour augmenter l'efficacité de l'histogramme et pallier ses inconvénients, plusieurs alternatives ont été proposées dans la littérature parmi lesquelles: l'histogramme cumulé introduit par Stricker et Orengo [286] qui a montré une grande efficacité et performance quant au problème de quantification de l'espace des attributs. Biernacki et Mohr [20] modélisent la distribution de la couleur et de gradients par un mélange de gaussiennes multi-dimensionnelles dont la moyenne et la matrice de covariance sont retenues. D'autres approches ont été proposées pour représenter l'attribut couleur tel que les moments de couleur [277] et les ensembles de couleur (*color sets*) [278]. La première approche estime que toute distribution couleur peut être caractérisée par ses moments. Les trois premiers moments (moyenne, variance et *skewness*) sont retenus pour la représentation de la couleur. La distance euclidienne pondérée est utilisée pour calculer la similarité. La seconde approche utilise les ensembles de couleur comme approximation de l'histogramme de couleur. Il s'agit de passer de l'espace RGB à un espace uniforme tel que HSV; ensuite quantifier cet espace en N bins. Un ensemble de couleur est défini par une sélection de couleur de l'espace quantifié. Une nouvelle alternative proposée par [309, 101] introduit la notion des histogrammes flous qui réduisent les effets de la quantification.

2.3.2 Attributs de texture

La texture est le second attribut visuel largement utilisé dans la recherche d'images par le contenu étant donné sa forte liaison à la perception humaine (c'est la première caractéristique perçue par l'œil). Elle reflète l'aspect homogène d'une région d'intérêt indépendamment de la couleur ou l'intensité. Aussi, la texture contient une importante information sur l'organisation des primitives ainsi que les interactions entre eux [116]. Mais malgré ces descriptions, une définition générale de la texture n'existe pas.

Plusieurs travaux se sont penchés pour étudier et analyser la texture. En 1973, Haralick et al. [116] proposent l'utilisation des matrices de co-occurrences pour la représentation de la texture. D'autres travaux se sont succédés afin d'améliorer cette approche. Gotlieb et Kreyszig [100] ont pu montrer expérimentalement que les paramètres *contraste*, *homogénéité* et *entropie* sont fortement discriminatifs.

D'autres travaux utilisent des études psycho-visuelles [290] ou mathématiques tels que les ondelettes, les filtres de Gabor, etc. pour décrire la texture. Une description plus détaillée sur la texture sera donnée dans le chapitre suivant.

2.3.3 Attributs de forme

Les caractéristiques de la forme sont extraits à partir des régions qui ont été segmentées au préalable. Selon le type d'applications, les caractéristiques de forme peuvent être invariants par translation, par rotation, ou autres. Deux grandes familles de représentation de la forme existent: approche basée sur les contours et approche basée sur les régions. Pour la première approche, seule la frontière externe entre objets ou régions est exploitée, tandis que pour la seconde, la forme de toute la région est prise en considération. Les techniques les plus utilisées sont les descripteurs de Fourier et les moments invariants.

2.4 Indexation multi-dimensionnelle

Cette section traite la manière par laquelle les descripteurs (signatures) du contenu visuel des images sont organisés notamment dans le cadre de bases d'images volumineuses.

Dans un système de gestion de bases de données standard, deux problèmes se posent lors de l'indexation: premièrement, le nombre de descripteurs (on parle aussi de la taille de la signature)[10, 107] qui doit être le plus minimale possible, maximum quatre descripteurs, douze pour les systèmes haute-dimension. Par exemple, on peut citer les arbres-X [12] ou bien les arbres-SR [152] (non encore disponible en tant qu'outils standards). Deuxièmement, la distance euclidienne utilisée apparaît inefficace pour discriminer certain contenu visuel. Pour cela, la technique d'indexation doit être capable de supporter d'autres mesures de similarité.

Pour parvenir à résoudre ces problèmes d'indexation, la solution proposée dans un premier temps est d'effectuer une réduction de la dimension de la signature. Ensuite adopter une technique d'indexation multi-dimensionnelle supportant des mesures de similarité autres que la distance euclidienne.

La façon la plus simple d'envisager ce post-traitement consiste à réaliser une Analyse en Composantes Principales (ACP) sur l'ensemble des signatures de la base d'images afin de construire une nouvelle base vectorielle de taille réduite, dans laquelle sont ensuite projetés les descripteurs originaux. Le problème de cette réduction dite aveugle est la perte de toute information originale sur la sémantique des descripteurs.

2.5 Mesures de similarité

La mesure de similarité concerne la recherche des images proprement dite. Lorsqu'un utilisateur lance une requête, le système effectue une mesure entre la requête et les images de la base dans l'espace des attributs. Les images sont considérées similaires (proches) si la distance entre eux est faible. Pour cela, la complexité de calcul de la distance doit être raisonnable puisque la tâche de recherche s'effectue en temps réel.

Le problème du choix d'une mesure de similarité la plus adéquate ayant été traité dans la littérature [102], nous listons dans cette partie, les mesures vectorielles les plus répandues.

Considérons deux images, \mathcal{I} l'image requête et \mathcal{J} l'image cible, indexées par leurs vecteurs respectifs $I = (I_1, \dots, I_n)$ et $J = (J_1, \dots, J_n)$. Le calcul de similarité entre \mathcal{I} et \mathcal{J} revient à calculer la similarité entre I et J .

2.5.1 Distances géométriques

La première catégorie de mesures correspond aux distances géométriques entre vecteurs : distances de Minkowski, cosinus, distance quadratique et distance de Mahalanobis.

Les métriques de Minkowski (ou normes L_p) sont les distances géométriques les plus courantes. Leur forme générale est la suivante:

$$L_p(I, J) = \sqrt[p]{\sum_{i=1}^n (I_i - J_i)^p} \quad (2.7)$$

Deux types de métriques de Minkowski se retrouvent fréquemment dans la littérature :

- la distance de Manhattan ou L_1 :

$$L_1(I, J) = \sum_{i=1}^n |I_i - J_i| \quad (2.8)$$

- la distance Euclidienne ou L_2 :

$$L_2(I, J) = \sqrt{\sum_{i=1}^n (I_i - J_i)^2} \quad (2.9)$$

Une autre mesure fréquente est basée sur le cosinus qui mesure l'angle θ entre la direction de deux vecteurs:

$$\cos(I, J) = 1 - \frac{I^T \cdot J}{\|I\| \cdot \|J\|} = 1 - \cos \theta \quad (2.10)$$

À la différence des distances de Minkowski, la distance quadratique suppose que tous les éléments du vecteur des attributs ne sont pas équitables et donc favorise ceux qui se ressemblent le plus:

$$d_{\text{quad}} = \sqrt{((I - J)^T \cdot A \cdot (I - J))} \quad (2.11)$$

où $A = [a_{i,j}]$ est la matrice de similarité dont chaque élément $a_{i,j}$ représente la distance entre deux éléments des vecteurs I et J .

Une proposition est faite par Hafner et al. [110] pour construire la matrice A comme suit :

$$a_{i,j} = 1 - \frac{d_{i,j}}{\max\{d_{i,j} | \forall i, \forall j\}} \quad (2.12)$$

Enfin, la distance de Mahalanobis prend en compte la corrélation entre vecteurs au lieu des distance dans le cas quadratique. La matrice C est la matrice de covariance entre I et J :

$$d_{\text{Mah}} = \sqrt{(I - J)^T \cdot C^{-1} \cdot (I - J)} \quad (2.13)$$

2.5.2 Distances entre distributions

Si on considère que l'image est une variable aléatoire dont les vecteurs d'attributs des pixels sont des réalisations, le problème de la mesure de similarité se ramène dans ce cas à une mesure entre distributions de probabilités. On parle alors d'une approche statistique de la mesure de similarité. La divergence de Kullback-Leibler [168], issue de la théorie de l'information, permet de mesurer la dissimilarité basée sur l'entropie mutuelle de deux distributions de probabilités :

$$d_{\text{Kull}}(I, J) = \sum_{i=1}^n I_i \log \frac{I_i}{J_i} \quad (2.14)$$

Cette mesure de dissimilarité est applicable pour la recherche d'images. Toutefois, elle n'est pas une distance. Comme suggéré par Puzicha et al. [236], il est plus intéressant de considérer la distance de Jeffrey vérifiant les axiomes des espaces métriques (distances). Cette métrique est numériquement stable et fait preuve de plus de robustesse au bruit:

$$d_{\text{Jeff}}(I, J) = \sum_{i=1}^n I_i \log \frac{I_i}{m_i} + \sum_{i=1}^n J_i \log \frac{J_i}{m_i} \quad (2.15)$$

avec $m_i = \frac{I_i + J_i}{2}$

Les tests d'hypothèses sont également adaptés pour le calcul de similarité entre images dans le cadre statistique. Le test du χ^2 , appliqué sous l'hypothèse gaussienne des distributions, est une alternative présentée comme performante dans la littérature [266]:

$$\chi^2(I, J) = \sum_{i=1}^n \frac{(I_i - J_i)^2}{(I_i + J_i)^2} \quad (2.16)$$

La distance de Bhattacharya est aussi appliquée dans le cas de deux distributions gaussiennes représentées par leur covariance Σ :

$$d_{\text{Batt}} = \frac{1}{8}(\mu_1 - \mu_2)^T \cdot \Sigma^{-1} \cdot (\mu_1 - \mu_2) + \frac{1}{2} \ln \frac{\det \Sigma}{\sqrt{\det \Sigma_1 \det \Sigma_2}} \quad (2.17)$$

avec, $\Sigma = \frac{\Sigma_1 + \Sigma_2}{2}$.

2.5.3 Intersection d'histogrammes

Dans le cas où les images sont caractérisées par des histogrammes, il est plus convenable d'utiliser des mesures de distance propres à cette représentation sans écarter évidemment l'utilisation des distances géométriques.

L'intersection des histogrammes est l'une des distances les plus employées dans ce cas. Elle a été introduite dans le cadre de l'indexation de la couleur par Swain et Ballard [288] :

$$d_{\cap}(I, J) = \frac{\sum_{i=1}^n \min(I_i, J_i)}{\sum_{i=1}^n I_i} \quad (2.18)$$

Or, cette mesure de similarité n'est pas une distance car elle ne vérifie pas la propriété de symétrie. Smith [276] propose une version symétrique :

$$d_{\text{Smith}}(I, J) = 1 - \frac{\sum_{i=1}^n \min(I_i, J_i)}{\min\left(\sum_{i=1}^n I_i, \sum_{i=1}^n J_i\right)} \quad (2.19)$$

2.6 Quelques systèmes de recherche par le contenu

Depuis l'émergence des systèmes de recherche d'images par le contenu, plusieurs communautés gouvernementales [141, 142], industrielles [79, 4, 66] et universitaires [231, 129, 280, 204, 200] s'y sont intéressées. Le standard ISO/IEC a proposé MPEG-7 (*Multimedia Content Description Interface*) comme standard de description du contenu multimédia, à travers lequel plusieurs systèmes de recherche d'images par le contenu commerciaux et académiques se sont développés.

Dans cette section nous présentons quelques systèmes opérationnels les plus répandus dans le domaine. Une synthèse des différents systèmes de recherche ainsi qu'une présentation des attributs utilisés et des mesures de similarité employées se trouvent dans les différents travaux de synthèse [273, 308, 192].

QBIC QBIC [219, 80] est le premier système de recherche d'images commercial développé par la société IBM. Les descripteurs visuels qu'intègrent ce système sont la couleur, la texture et la forme. Pour l'attribut de couleur, sa signature correspond à un vecteur moyen 3D exprimé dans l'un des espaces de couleurs retenus : RGB, YIQ, Lab, etc. Pour la texture, il s'agit d'une version modifiée des descripteurs de Tamura [290]. Pour la forme, elle correspond à un vecteur de quatre paramètres : la circularité de l'objet, l'excentricité, l'orientation principale et un ensemble de moments algébriques invariants. La distance employée pour comparer les images est la distance euclidienne et, pour les histogrammes, QBIC intègre la distance quadratique. L'indexation multidimensionnelle appliquée dans QBIC pour accélérer la recherche est basée sur les arbres-R*.

Photobook Photobook [232] est un système académique développé par le laboratoire MIT qui permet la navigation dans de grandes bases d'images en combinant l'annotation textuelle des images et les descripteurs extraits des images. La caractérisation des textures est basée sur l'approche « *Wold decomposition* » [282]. Les descripteurs retenus pour la texture sont *la périodicité, l'orientation et grossièreté*. Pour la forme, elle est modélisée par des éléments finis dont des attributs pertinents sont extraits. La mesure de similarité employée est la distance euclidienne.

NeTra Le département d'électronique et de génie informatique de l'université de Californie à Santa Barbara a développé le système de recherche d'images NeTra [200] qui utilise une description par région segmentée dont chacune est caractérisée par la couleur, la texture, la forme et une localisation spatiale. Le premier attribut de couleur est représenté par un dictionnaire (*code-book*) avec une quantification sur 256 couleurs. Les filtres de Gabor et les ondelettes caractérisent la texture. Pour la forme, les descripteurs sont extraits des courbures et centroïde de la forme. La correspondance entre images s'effectuent par une distance euclidienne.

Blobworld Blobworld [31], au même titre que Netra, repose sur la caractérisation des régions par la couleur, la forme, la texture et la localisation spatiale. La couleur dans le cas de Blobworld est représentée par un histogramme de 218 cases dans l'espace Lab. Le contraste et l'anisotropie caractérisent la texture. La surface, l'excentricité et l'orientation correspondent aux descripteurs de la forme. La requête d'un utilisateur consiste à choisir une région et à caractériser son importance par deux variables linguistiques (« *somewhat* » et « *very* »). La distance quadratique est employée pour la couleur et la distance euclidienne est utilisée pour la texture et la forme. L'indexation multidimensionnelle utilisée est similaire à celle de QBIC (arbre-R*).

IKONA IKONA [23] est un système de recherche et de navigation interactive dans de grandes bases de données multimédia développé par l'équipe IMEDIA de l'INRIA est basé sur une architecture client-serveur. IKONA procède à l'extraction des descripteurs de la couleur à l'aide de l'histogramme de couleur pondéré [309]. L'extraction de la texture est effectuée en utilisant le spectre de Fourier. Pour la forme, il utilise l'histogramme d'orientation des contours.

2.7 Conclusion

Ce chapitre a fait l'état de l'art sans exhaustivité des différentes signatures des attributs visuels pouvant être utilisés pour la recherche d'images par le contenu ainsi que les approches correspondantes. Aussi, nous avons dressé une liste de quelques systèmes CBIR académiques et industriels parmi ceux les plus connus avec leurs principaux choix d'indexation du contenu et de calcul de similarités.

Le chapitre suivant se focalisera à une étude détaillée de la texture, les méthodes d'analyse de texture, les techniques de sélection des paramètres pertinents ainsi que quelques applications de caractérisation de la texture. Le choix d'un meilleur ensemble de descripteurs de texture promet une bonne caractérisation du concept de texture.

Processus de description de la texture

3.1 Introduction

La texture est une caractéristique fondamentale comme la couleur pour l'analyse des différents types d'images telles que les images naturelles, aériennes, médicales, etc. puisqu'elle reflète un aspect fortement lié à la perception humaine. Cependant, la notion de texture est l'une des moins bien définies comparée aux autres attributs visuels. Il semble permis de dire que le problème majeur réside en la définition elle-même de la texture. Reste que cette notion est incontournable dans toute analyse d'images en témoin l'abondance de littérature en la matière surtout que toute analyse d'images appelle un présupposé conceptuel sur la notion de texture.

Plusieurs définitions ont été proposées, tout dépend de l'application visée. Parmi ces définitions littéraires, nous trouvons les suivantes :

- Dans le langage didactique, la texture signifie la disposition, l'entrelacement des parties qui composent un corps. « *Par cette simple exposition de la texture du bois, on voit que la cohérence longitudinale doit être bien plus considérable que l'union transversale.* » [Buffon, Histoire naturelle générale et particulière]
- Pour le dictionnaire Larousse, la texture est la répétition spatiale d'un motif de base (fils de tissage ou étoffe) dans différentes directions de l'espace.
- Pour l'*Oxford dictionary*, « *texture is the way that something feels when you touch it: material with a silky – smooth, soft and shiny – texture.* »

De ces définitions, limitatives, on constate que la texture fait référence à la nature structurée de la surface considérée. Les adeptes d'une telle définition s'orientent généralement vers une approche spectrale, structurelle ou syntaxique de la texture.

Plus généralement, dans le domaine multimédia et particulièrement le traitement d'images, la texture pourra être considérée comme une région d'intérêt (ROI pour *region of interest*) de l'image qui présente un aspect homogène pour un observateur humain. Elle peut représenter un même matériau : bois, tissu, ciel, eau, etc. qui comporte un nombre limité de motifs qui sont répétés suivant des positions plus ou moins régulières ou aléatoires.

De nombreuses études ont été proposées dans la littérature afin de généraliser la notion de texture. Cependant, il semble y avoir aucune méthode ou une approche formelle qui soit utile pour ce large éventail de textures :

- Rosenfeld et Troy [247], Gross [103], et Wu [319] définissent la texture comme un arrangement répétitif d'une zone donnée.
- Dans [115], Haralick et al. se reposent sur les aspects statistiques et structurels de la texture. Ils considèrent la texture comme un phénomène à deux niveaux : le premier concerne les propriétés

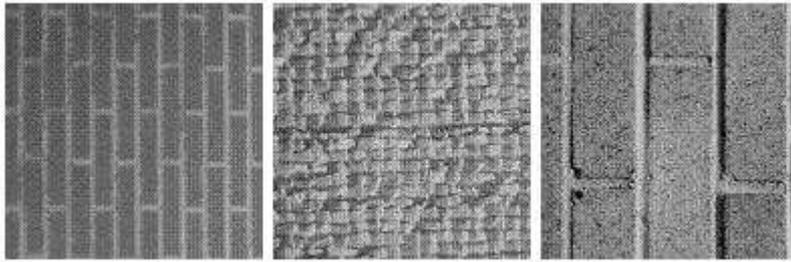


Figure 3.1 – Textures structurales

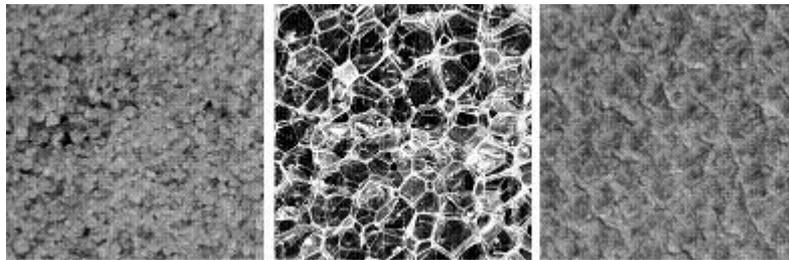


Figure 3.2 – Textures aléatoires

des éléments de base ou « primitives¹ (*texels*) » à partir desquels est formée la texture; le deuxième niveau concerne l'organisation et la dépendance spatiale de ces primitives.

- Tamura et al. [290] se sont reposés dans leur définition de la texture sur des tests psycho-visuels. La texture ainsi est traduite par termes linguistiques proches du langage humain tels que texture
- Une autre définition provient de Unser et Eden [298] où ils déclarent la texture comme une structure disposant de certaines propriétés spatiales homogènes et invariantes par translation.
- En 1991, Haralick et Shapiro [117] ont défini la texture par l'uniformité, la densité, la grossièreté, la rugosité, la régularité, l'intensité et la direction des primitives et de leurs relations spatiales.

Bien qu'il n'existe pas de définition générale, satisfaisante pour tous, de la texture; chacun essaie d'appréhender cette notion en fonction de son centre d'intérêt; certains éléments communs apparaissent dans les définitions trouvées dans la littérature qui sont des primitives et/ou propriétés ainsi que des relations statistiques et/ou structurales entre ces primitives. Ainsi, en pratique, on peut définir les textures en deux grandes classes qui correspondent à deux niveaux de la perception humaine :

- les textures structurales ou *macro textures* qui représentent l'aspect répétitif dans un intervalle régulier d'une primitive selon une règle (ex. : mur de brique, carrelage, etc.) (figure 3.1) ;
- les textures aléatoires ou *micro textures* représentant un aspect fin aléatoire (ex. : sable, herbe, etc.) (figure 3.2) ne comportant ni primitive isolable ni fréquence de répétition.

L'objectif de ce chapitre est de dresser le processus de description de l'information de texture en se basant sur le procédé d'extraction de connaissances à partir des données (ECD) [122] dans un but de classification, de recherche d'images et de navigation dans une base d'images. La figure 3.3 illustre le processus adopté dans notre thèse pour la caractérisation des textures.

1. Sélection des bancs d'essais : Vu la diversité des images texturées provenant de différents do-

¹Une primitive est un ensemble connexe d'un certain nombre de pixels de niveaux de gris à peu près semblables : en fait, on parle d'un motif de base.

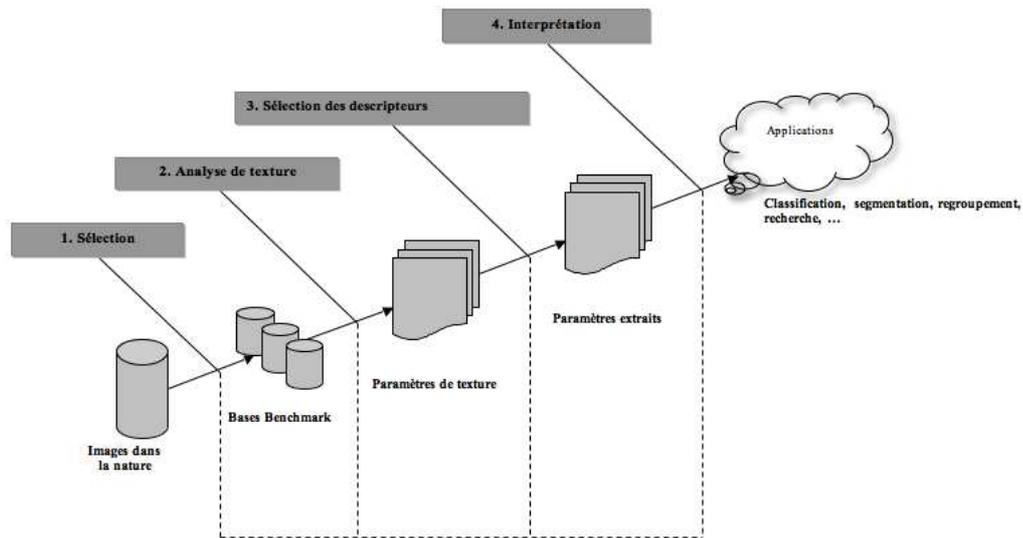


Figure 3.3 – Processus d’ECD pour la description des textures

maines et par différents moyens d’acquisition et de numérisation, il est fortement recommandé de choisir une base de cette population d’images qui soit la plus représentative. Cette base est dite « Benchmark ».

2. **Analyse de texture** : Pour toute application visant l’utilisation des textures pour la classification, la segmentation, la recherche d’images ou autres, elle nécessite une phase d’extraction des paramètres de textures. Il s’agit dans cette étape de transformer les images de la base en paramètres numériques ayant une signification physique. L’extraction des descripteurs de texture permet de projeter les images texturées dans un espace ayant une dimension importante (nombre de paramètres) faisant appel aux techniques de traitement d’images notamment l’analyse de texture. Le nombre de paramètres de textures obtenu est important et complexe à exploiter par la suite. De ce fait, une réduction de l’espace de paramètres se voit utile.
3. **Sélection des descripteurs de texture** : La sélection des descripteurs de texture consiste à réduire l’espace de représentation des images en un espace de dimension inférieure et à ne garder que les descripteurs les plus pertinents caractérisant au mieux la texture sans perdre leur signification physique du départ. La réduction de dimension peut aussi s’effectuer par l’extraction des descripteurs de texture qui consistent à créer de nouveaux paramètres par des combinaisons linéaires ou non-linéaires des paramètres initiaux. Dans ce cas, la signification physique du départ est perdue.
4. **Applications** : Lorsque l’ensemble pertinent de descripteurs de texture est identifié, différentes applications peuvent être envisagées. Parmi ces applications, nous citons la classification supervisée et non-supervisée, la recherche d’images, la segmentation, la reconnaissance d’objets, la navigation, etc.

Classe	# images	Classe	# images
bark	10	brick	10
bubbles	10	cloth	10
grass	12	leather	10
pigskin	10	raffia	10
sand	14	straw	10
wall	10	water	10
weave	10	wood	10

Table 3.1 – Classification de textures dans la base Brodatz 1

3.2 Choix des bancs d’essais

Le but de notre travail est d’effectuer en premier lieu la recherche d’images par contenu dans un cadre général. Autrement dit, nous ne focaliserons pas dans un domaine spécifique où les caractéristiques des applications et des textures traitées sont connues au préalable pour optimiser la recherche. La mise en œuvre et l’évaluation des méthodes de recherche qui sont relativement général et s’appliquent à de nombreux contextes se voient très difficile car elles exigeraient l’utilisation des différentes bases de texture avec différentes techniques d’analyse de texture. Ainsi, il est nécessaire de disposer d’une vérité-terrain sur laquelle il s’agit d’automatiser le processus d’extraction des descripteurs et obtenir une référence statique du modèle texture pour pouvoir l’évaluer par la suite sur des bases de test.

La solution alternative est d’utiliser des bases d’images de texture pour lesquelles la vérité terrain est disponible; cette vérité terrain correspond généralement à un ensemble de classes de textures mutuellement exclusives. Cependant, pour couvrir un large contexte d’images texturées, il est très important d’utiliser plusieurs vérité terrain qui ont des caractéristiques qui diffèrent non seulement entre ces bases d’images (éviter le chevauchement des classes qui peut biaiser l’évaluation des résultats) mais aussi entre les classes de chaque base de textures.

Avec ces contraintes, nous avons choisi, pour l’évaluation des descripteurs de texture, des bases de texture largement utilisées par différentes applications nécessitant l’utilisation des textures :

- **Brodatz 1** : L’album de Brodatz [28] est le premier standard pour l’évaluation des algorithmes de texture tels que la classification, la segmentation, la reconnaissance d’objets, etc. La base « Brodatz 1 » est un sous-ensemble de l’album composée de 14 classes dont chacune comprend en moyenne 10 textures prises sur différents angles d’orientation et de contraste. Le tableau 3.1 donne un résumé sur les différentes classes de cette base. La taille des textures est de 512×512 et 1024×1024 pixels en format TIFF.²
- **Brodatz 2** : La seconde base de l’album de Brodatz [28] comprend 111 textures dont aucune connaissance *a priori* sur les différentes classes de texture n’est fournie ni les classes pouvant exister. La taille des textures est de 512×512 pixels.³
- **Meastex** : La base Meastex fournit un ensemble de textures homogènes comprenant 5 classes. Chaque texture est de 512×512 pixels au format PGM. Le tableau 3.2 donne un récapitulatif sur ces différentes classes.⁴

²Cette base peut être téléchargée sur le site <http://sipi.usc.edu/database/>.

³Elle est disponible à l’adresse <http://www.ux.uis.no/~tranden/brodatz.html>.

⁴La base est disponible à l’adresse http://www.texturesynthesis.com/meastex/www/for_images.html.

Classe	# images	Classe	# images
asphalt	4	concrete	12
grass	10	misc	10
rock	25		

Table 3.2 – Différentes classes de la base Meastex

3.3 Extraction d'attributs de texture

Les définitions de la texture que nous venons d'évoquer se restreignent souvent à des cas particuliers. L'une se réfère au tissage, une autre à la foresterie, etc. Cependant on retrouve toujours un critère important qui est la notion d'organisation spatiale. Aussi pour reconnaître la texture d'un objet, l'homme dispose de deux sens (définition *Oxford*) qui n'apportent pas toujours la même information : la vue et le toucher. Prenons l'exemple d'une zone de sable granuleuse que l'on photographie. Si on insère cette photographie dans son emplacement dans le sable réel, les informations issues de la vue restent identiques tandis que celles issues du toucher deviennent différentes. Dans ce travail, nous nous intéressons évidemment qu'à la partie visuelle.

Une image est une représentation à deux dimensions d'une scène généralement dans un espace tridimensionnel. Elle est créée afin de fournir une entité observable par l'oeil humain. Pour cela, l'information de texture est communément donnée en niveaux de gris.

Le processus d'extraction des attributs de texture se déroule en deux étapes : premièrement, définir le type de la texture en question, ensuite, appliquer la technique appropriée pour extraire les attributs caractéristiques de cette texture.

3.3.1 Classification des images texturées

Malgré la difficulté de générer une définition précise et générale pour tous les types de texture, il est possible de classer l'ensemble des textures en différentes catégories : structurales et aléatoires.

3.3.1.1 Textures structurales

Les textures structurales sont considérées comme étant la répétition d'un motif élémentaire. La répartition spatiale de ces motifs de base est effectuée suivant des règles de direction et de placement. Par exemple, l'image de gauche dans la figure 3.1 est composée d'un ensemble de briques (constituent les primitives de la texture) disposées régulièrement (déplacement) de manière verticale (direction).

La caractérisation de ce type de texture s'appuie donc sur la description des éléments primitives de la texture suivie d'une définition de la structure; c'est à dire un ensemble de règles (syntaxiques ou autres) sur ces primitives en faisant appel aux techniques structurales.

Dans d'autres cas d'images textures naturelles, il n'est pas toujours possible de caractériser la texture par des primitives puisqu'elles n'existent pas. Ceci introduit la classe de texture suivante.

3.3.1.2 Textures aléatoires

À l'inverse des textures structurales, les textures aléatoires ont un aspect totalement désordonné tout en apparaissant globalement homogènes (aucune primitive n'est discernable par l'oeil humaine). L'effet textuel dans ce cas est alors associé à l'impression visuelle faite par l'observateur.

La caractérisation dans ce cas s'effectue en considérant l'image texturée comme étant associée à un processus aléatoire bidimensionnel. L'analyse de ce type de texture fait appel aux approches probabilistes.

3.3.2 Analyse de texture

L'analyse de texture est un domaine important du traitement d'images, de vision par ordinateur et de la reconnaissance de formes. L'analyse de texture a fait l'objet de nombreux travaux de recherche qui ont engendré une multitude de méthodes d'analyse [313, 115, 51, 139, 274]. Les méthodes d'analyse de texture reposent essentiellement sur l'extraction d'un certain nombre de paramètres dits descripteurs. Ces descripteurs permettent de décrire, de caractériser, de classifier, ..., de segmenter les textures en question.

L'analyse de texture peut s'effectuer en plusieurs niveaux que nous allons introduire dans le paragraphe suivant. Dans ce travail, l'accent sera mis sur les méthodes statistiques, en particulier sur les matrices de co-occurrences et la technique de Tamura, puisqu'elles répondent à la majorité de nos contraintes imposées pour l'extraction des descripteurs de texture.

Approche structurale L'approche structurale tient compte de l'information structurelle et contextuelle de la texture et est particulièrement bien adaptée aux macro textures. Les étapes d'analyse sont d'abord l'identification des éléments primitifs puis la caractérisation des règles de déplacement entre ces primitives. Autrement dit, dans un premier temps on effectue une croissance de région sur un ou plusieurs attributs (intensité lumineuse, gradient, etc.) de façon à identifier les primitives qu'on caractérise par leur forme et la valeur moyenne de l'attribut utilisé. Dans un deuxième temps on calcule les histogrammes du premier ordre ou du second ordre de ces paramètres afin de caractériser leur répartition dans l'image. Les deux structures les plus importantes sont les structures de graphe et les structures syntaxiques [295, 324, 193, 156, 198].

Approche statistique L'approche statistique traite la texture comme un processus stochastique déterministe. Elle modélise les notions qualitatives usuelles de texture : « granularité, contraste, homogénéité, répétitivité, fragmentation, orientation, etc. » :

- (i) La granularité est le motif dominant de la texture et parfois même, par abus de langage, synonyme de texture. Le grain qui donne la granularité est lui-même constitué de pixels voisins possédant le même niveau de gris. La taille et la densité des grains déterminent le niveau de finesse de la texture.
- (ii) Le contraste est basé sur le nombre de niveaux de gris et leur taux de variation. Changer le contraste entraîne une modification dans la qualité de l'image mais pas sa structure.
- (iii) L'orientation est une propriété globale pour une région et traduit la direction générale prise par les motifs ou grains d'une texture.

Des mesures statistiques sont estimées sur une région ou un voisinage autour d'un pixel de l'image. Il est à noter que ces paramètres dépendent fortement de la taille de la région d'analyse qui doit contenir au moins une primitive de base. Le choix de la taille de la région d'analyse est conditionnée par plusieurs facteurs tels que : la résolution et la granularité de la texture, la dynamique des niveaux de gris, etc. Par exemple, plus la taille de la région en question est réduite, plus il existe un risque de perdre l'information granularité de la texture (voir figure 3.4).

D'après son travail ??, Julesz montre que l'œil humain peut distinguer entre textures ayant les mêmes statistiques du second ordre.

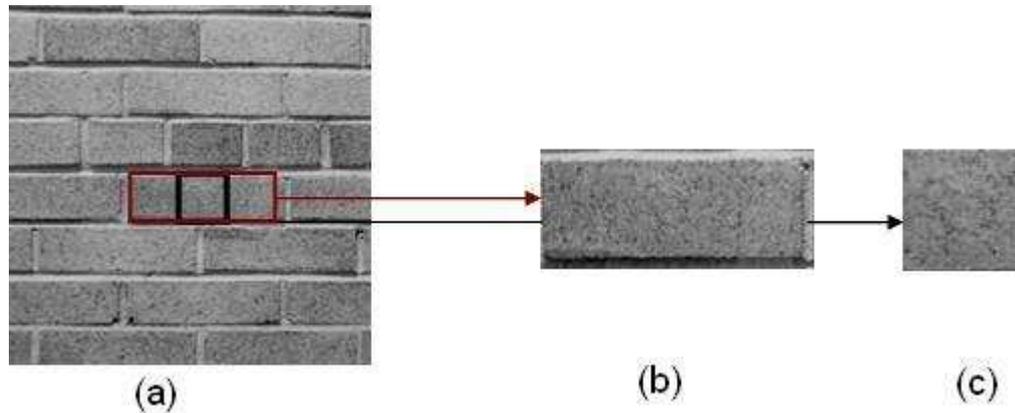


Figure 3.4 – Taille d’analyse et granularité.(a): texture brique, (b): primitive mesurant la granularité de la texture, (c): région d’analyse

Statistiques du premier ordre L’analyse par les statistiques du premier ordre consiste à construire une distribution des pixels individuels, par exemple l’histogramme des niveaux de gris de la texture. Certaines applications utilisent cette distribution comme descripteur caractérisant la texture. Dans d’autres, des paramètres statistiques sont calculés à base de cette distribution comme la moyenne, la variance, le Kurtosis et l’aplatissement (*skewness*). Bien que ces mesures montrent une efficacité de discrimination dans des cas simples, elles restent limitées comparées aux statistiques d’ordre supérieur.

Statistiques du second ordre L’analyse par statistiques du second ordre tient en compte des paires de pixels tout en incluant la dépendance spatiale entre eux, chose qui n’est pas vérifiée par le premier ordre. Parmi ces méthodes, on trouve les matrices de co-occurrences (MCO) [225, 272] et la technique de Tamura [290]. Ces deux méthodes ont fait l’objet de plusieurs travaux et s’avèrent souvent efficace. Dans ce qui suit, nous introduirons ces deux techniques.

3.3.2.1 Matrices de co-occurrences

Dans la mesure où l’on recherche certaines formes de régularité au sein des images et où une image est constituée d’un très grand nombre de pixels, l’analyse statistique semble *a priori* intéressante. En effet, l’utilisation des matrices de co-occurrences des niveaux de gris des pixels d’une image est la technique statistique la plus populaire [38, 205, 163, 157, 300] pour l’extraction des descripteurs de texture pour différents types d’images, ainsi pour l’analyse, la segmentation et la classification des images de types variés, tels que les images médicales, aériennes, astronomiques, etc.

Définitions Cette approche est basée sur la probabilité jointe de la distribution des pixels dans l’image [109, 116]. L’élément $p_{d,\theta}(i, j)$ de la matrice de co-occurrences définit la fréquence d’apparition des couples de niveaux de gris i et j pour les couples de pixels séparés par une distance d selon la direction θ .

Plus précisément, la formule définissant une matrice de co-occurrences non normalisée est la suivante :

$$P_{d,\theta}(i, j) = |\{(i, j) | f(n, m) = i, f(n \pm d \cos \theta, m \pm d \sin \theta) = j\}| \quad (3.1)$$

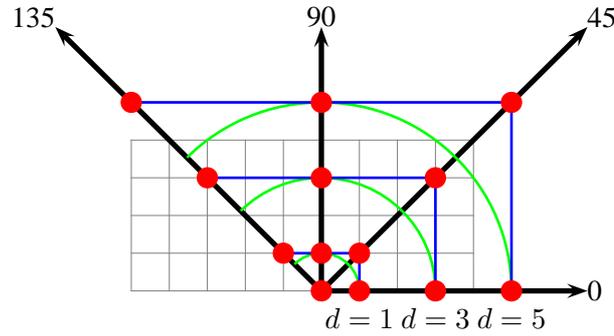


Figure 3.5 – Voisinage (partie supérieure) d'un pixel selon quatre directions principales et trois distances pour l'extraction des matrices de co-occurrences

1	1	2	3	3
3	2	2	2	1
3	1	1	1	2
3	3	2	2	2

Figure 3.6 – Une « image » 4×5 avec trois niveaux de gris

Cette première matrice est ensuite normalisée :

$$p_{d,\theta}(i, j) = \frac{P_{d,\theta}(i, j)}{N \times M} \quad (3.2)$$

pour une image de taille $N \times M$.

En pratique, cette formulation générale est particularisée pour quelques directions principales de θ , à savoir 0° , 45° , 90° et 135° :

$$P_{0,d}(i, j) = |\{(n, m), (k, l) : n = k, |m - l| = d\}| \quad (3.3)$$

$$P_{45,d}(i, j) = |\{(n, m), (k, l) : (n - k = d, m - l = -d) \vee (k - n = -d, l - m = d)\}| \quad (3.4)$$

$$P_{90,d}(i, j) = |\{(n, m), (k, l) : |k - n| = d, l = m\}| \quad (3.5)$$

$$P_{135,d}(i, j) = |\{(n, m), (k, l) : (k - n = d, l - m = d) \vee (k - n = -d, l - m = -d)\}| \quad (3.6)$$

où, dans les quatre définitions, le pixel en position (k, l) doit posséder un niveau de gris égal à i tandis que le pixel situé aux coordonnées (n, m) a la valeur de j .

Une autre variante des matrices de co-occurrences, carré plutôt que circulaire, peut être retenue. La figure 3.5 illustre les « erreurs » commises par ces quatre approximations pour des valeurs de d égales à 1, 3 et 5.

Afin d'illustrer les calculs, la figure 3.6 présente la matrice d'une pseudo-image à partir de laquelle sont extraites quatre matrices de co-occurrences (cf. tableau 3.3). En pratique, les matrices sont bien plus grandes et peuvent être visualisées sous la forme d'images en niveaux de gris, les pixels les plus lumineux représentant les valeurs élevées de la distribution jointe (cf. figures 3.7 et 3.8).

Les figures 3.9 et 3.10 représentent respectivement l'image réelle *Brick* et ses quatre matrices de co-occurrences et l'image *Plasbubbles* suivie des quatre matrices de co-occurrences. Les matrices de co-occurrences sont calculés pour la distance $d = 1$ et sur les quatre orientations principales.

	(a) $\theta = 0$	(b) $\theta = 45$	(c) $\theta = 90$	(d) $\theta = 135$																																																																
	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td></td><td>1</td><td>2</td><td>3</td></tr> <tr><td>1</td><td>6</td><td>3</td><td>1</td></tr> <tr><td>2</td><td>3</td><td>8</td><td>3</td></tr> <tr><td>3</td><td>1</td><td>3</td><td>4</td></tr> </table>		1	2	3	1	6	3	1	2	3	8	3	3	1	3	4	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td></td><td>1</td><td>2</td><td>3</td></tr> <tr><td>1</td><td>2</td><td>3</td><td>3</td></tr> <tr><td>2</td><td>3</td><td>4</td><td>3</td></tr> <tr><td>3</td><td>3</td><td>3</td><td>0</td></tr> </table>		1	2	3	1	2	3	3	2	3	4	3	3	3	3	0	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td></td><td>1</td><td>2</td><td>3</td></tr> <tr><td>1</td><td>0</td><td>7</td><td>3</td></tr> <tr><td>2</td><td>7</td><td>4</td><td>1</td></tr> <tr><td>3</td><td>3</td><td>1</td><td>4</td></tr> </table>		1	2	3	1	0	7	3	2	7	4	1	3	3	1	4	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td></td><td>1</td><td>2</td><td>3</td></tr> <tr><td>1</td><td>0</td><td>7</td><td>2</td></tr> <tr><td>2</td><td>7</td><td>4</td><td>0</td></tr> <tr><td>3</td><td>2</td><td>0</td><td>2</td></tr> </table>		1	2	3	1	0	7	2	2	7	4	0	3	2	0	2
	1	2	3																																																																	
1	6	3	1																																																																	
2	3	8	3																																																																	
3	1	3	4																																																																	
	1	2	3																																																																	
1	2	3	3																																																																	
2	3	4	3																																																																	
3	3	3	0																																																																	
	1	2	3																																																																	
1	0	7	3																																																																	
2	7	4	1																																																																	
3	3	1	4																																																																	
	1	2	3																																																																	
1	0	7	2																																																																	
2	7	4	0																																																																	
3	2	0	2																																																																	

Table 3.3 – Matrices de co-occurrences pour « l'image » de la figure 3.6 avec $d = 1$ et différentes orientations

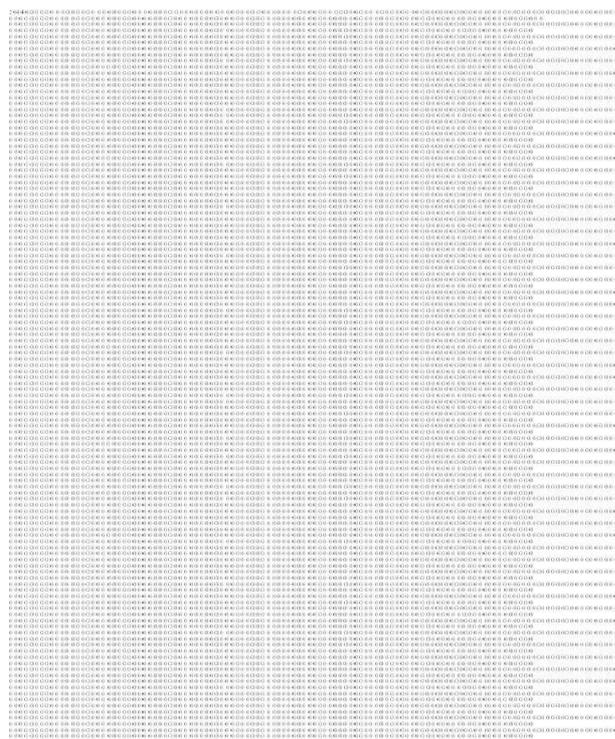


Figure 3.7 – Illustration d'une matrice de co-occurrences pour l'image *Raffia*

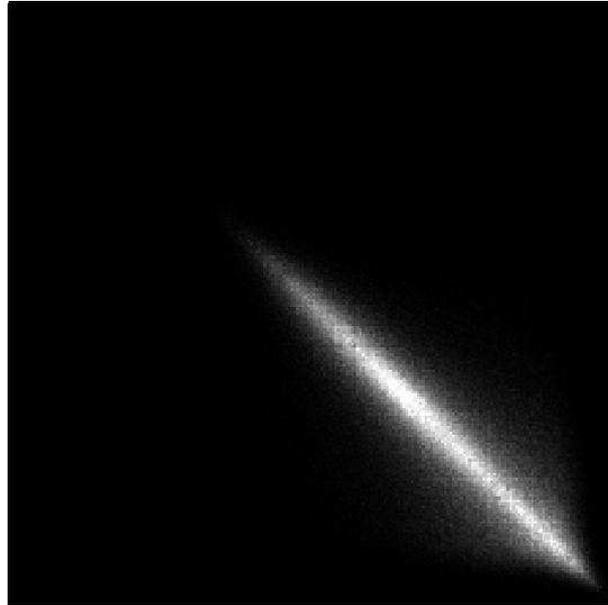


Figure 3.8 – Illustration de sa distribution en niveau de gris



Figure 3.9 – L'image *Brick* et ses matrices de co-occurrences



Figure 3.10 – L'image *Plasbubbles* et ses matrices de co-occurrences

Contraintes À partir de ces statistiques, il est possible de repérer *certaines* textures. Il faut pour cela qu'un motif :

- se répète à la même échelle ;
- sous les mêmes conditions d'éclairage.

Les matrices de co-occurrences contiennent une masse importante d'informations difficilement manipulable. C'est pour cela qu'elle n'est pas utilisée directement mais à travers des mesures dites *paramètres d'Haralick*.

Paramètres associés aux MCO En 1973, Haralick et al. [116] ont proposé quatorze descripteurs calculés à partir des MCO. Bien que corrélés entre eux, ces descripteurs réduisent l'information contenue dans les MCO et permettent une meilleure discrimination entre différents types de textures.

Le choix du pas de déplacement et la taille de la fenêtre d'analyse sur laquelle s'effectue les mesures sont les paramètres clés pour la réussite de la méthode.

Au fur et à mesure de notre lecture approfondie dans le domaine de traitement de texture, il s'est avéré que peu d'entre ces descripteurs sont utilisés pour la description et l'analyse des textures. Dans ce qui suit, nous allons introduire les quatorze différents paramètres d'Haralick mais nous ne retenons pour nos expériences que les huit premiers.

Énergie L'énergie d'une matrice de co-occurrences se définit comme la somme des carrés de ses fréquences :

$$eng = \sum_{i=0}^n \sum_{j=0}^n p_{d,\theta}(i, j)^2 \quad (3.7)$$

Elle est perçue comme un moyen de mesurer l'*homogénéité globale* de l'image. En effet, l'énergie a une valeur d'autant plus faible qu'il y a de nombreuses zones homogènes. Autrement dit, il existe peu de transitions de niveaux de gris. Bien sûr, cette formule s'entend pour une matrice particulière, c'est-à-dire modulo la distance et l'orientation.

Ainsi, une image pourrait présenter une énergie très faible, voire nulle, selon une orientation particulière, par exemple des lignes verticales alternant blanc et noir. Cette même image pourrait même présenter une énergie nulle selon l'orientation horizontale si la distance choisie vaut un et que les largeurs des lignes vaut également un. Mais, en dehors de cas semblables, construits explicitement pour mettre en difficulté la mesure, il est quasiment impossible pour une texture « fine », c'est-à-dire peu homogène, de présenter une énergie faible.

Entropie De manière générale, l'entropie est une mesure de la dispersion d'une distribution. La formule, classique est :

$$ent = - \sum_{i=0}^n \sum_{j=0}^n p_{d,\theta}(i, j) \log p_{d,\theta}(i, j) \quad (3.8)$$

avec $p_{d,\theta}(i, j) \log p_{d,\theta}(i, j) = 0$ quand $p_{d,\theta}(i, j) = 0$.

Ici, l'entropie devient un indicateur du désordre de la texture. Elle est élevée lorsque les valeurs de la matrice sont presque toutes égales. L'entropie permet aussi de caractériser le degré de *granulation* de l'image. Plus l'entropie est élevée plus la texture est dite « granuleuse ou grossière ».

La figure 3.11 illustre le comportement de l'entropie vis-à-vis de celui de l'énergie. Il n'y a donc pas de corrélation entre ces deux mesures... à une exception près. Si toutes les valeurs sont proches de zéro, ce qui sera inévitablement le cas général puisque la matrice de co-occurrences est normalisée, alors

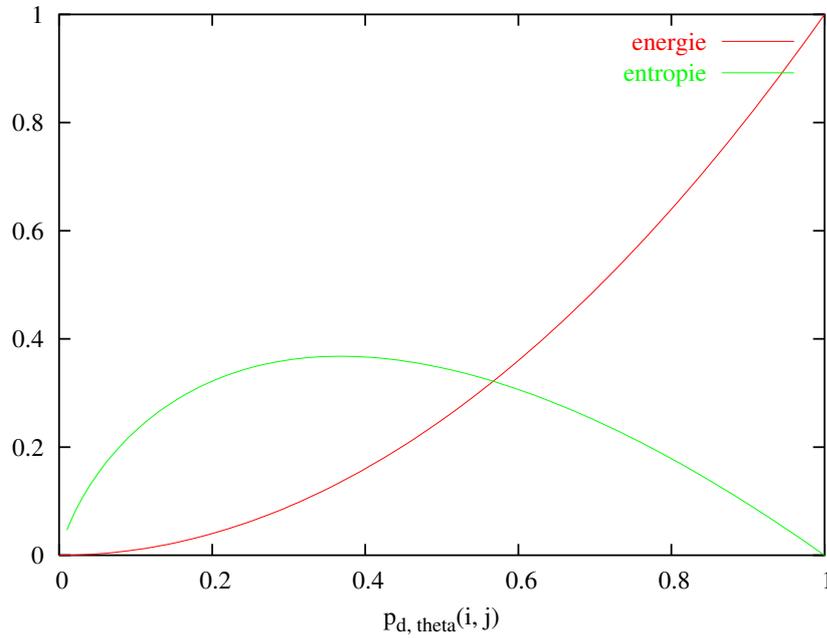


Figure 3.11 – Comportements comparés des valeurs prises par le terme de l'énergie, $p_{d,\theta}(i, j)^2$, et de l'entropie, $-p_{d,\theta}(i, j) \log p_{d,\theta}(i, j)$

l'énergie et l'entropie présentent une corrélation forte, plus particulièrement marquée dans l'intervalle $[0; 0, 1]$ où les courbes peuvent être facilement approximées par des droites. Il n'est donc pas évident que l'entropie soit utile si elle est utilisée telle quelle.

Homogénéité locale Plus le paramètre d'homogénéité locale est élevé plus la texture possède des régions homogènes ou des périodicités dans le sens de la translation :

$$homloc = \sum_{i=0}^n \sum_{j=0}^n \frac{1}{1 + (i - j)^2} p_{d,\theta}(i, j) \quad (3.9)$$

Uniformité L'uniformité d'une texture peut être traduite par l'absence de différences de niveaux de gris. Une formule est alors :

$$unif = \sum_{i=0}^n p_{d,\theta}(i, i)^2 \quad (3.10)$$

Ce paramètre d'uniformité est d'autant plus élevé que les différences de niveaux de gris sont faibles.

Contraste Ce paramètre a un comportement « inverse » de l'homogénéité locale, car il mesure les variations locales des niveaux de gris. Sa valeur est d'autant plus importante que la texture possède moins de régions homogènes. Ce paramètre permet aussi de caractériser la dispersion des valeurs de la matrice par rapport à sa diagonale principale :

$$cont = \sum_{i=0}^n \sum_{j=0}^n (i - j)^2 p_{d,\theta}(i, j) \quad (3.11)$$

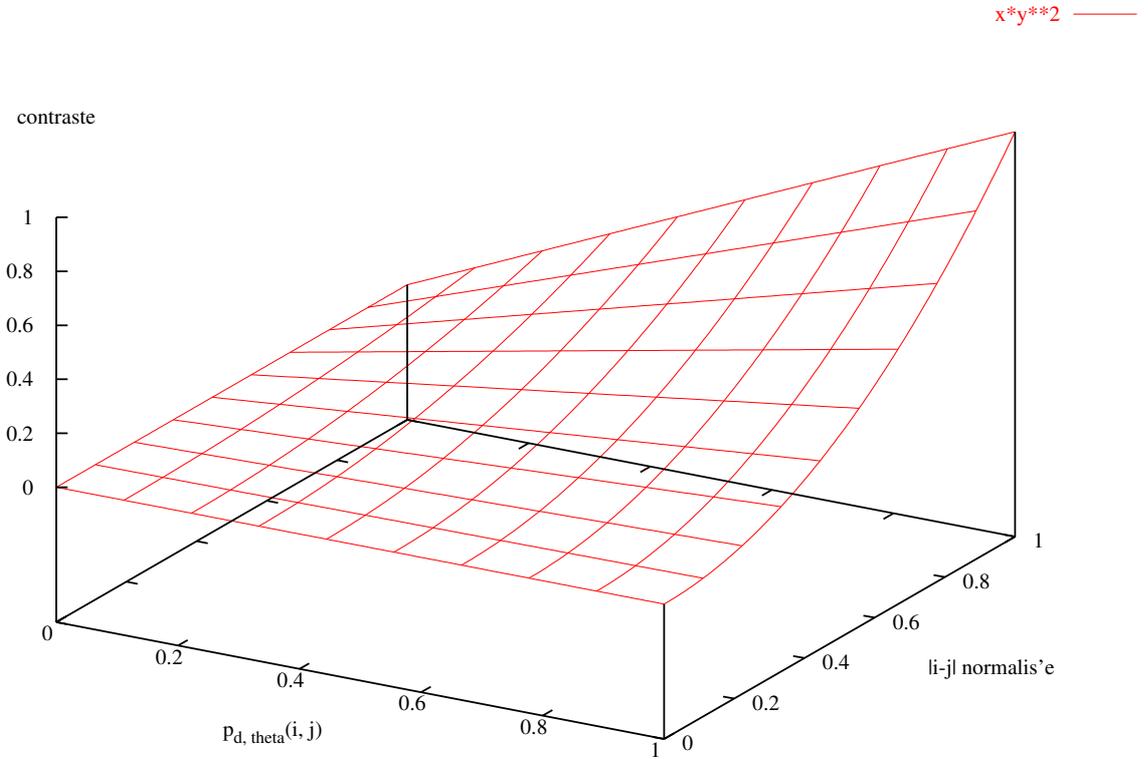


Figure 3.12 – Comportement de la mesure du contraste

La figure 3.12 fournit une illustration du comportement de cette mesure. Une valeur élevée n’est obtenue que par une fréquence élevée jointe à une grande différence entre niveaux de gris. En particulier, les fréquences de la diagonale principale n’interviennent pas dans cette mesure.

La figure 3.13 compare le comportement de cette mesure avec celui de l’homogénéité locale.

Direction Une façon de définir la direction d’une texture est de la caractériser par la présence importante de couples de pixels *de même valeur* dans le sens de la translation. Dans le cas le plus marqué il s’agit d’une ligne d’intensité uniforme. Dans le cas général, il s’agit d’un « pointillé » particulier puisque la direction retenue pour le calcul des matrices est arbitraire et ne correspond pas l’angle effectif de la texture. En se basant sur cette interprétation, la formule est :

$$dir = \sum_{i=0}^n p_{d,\theta}(i, i) \tag{3.12}$$

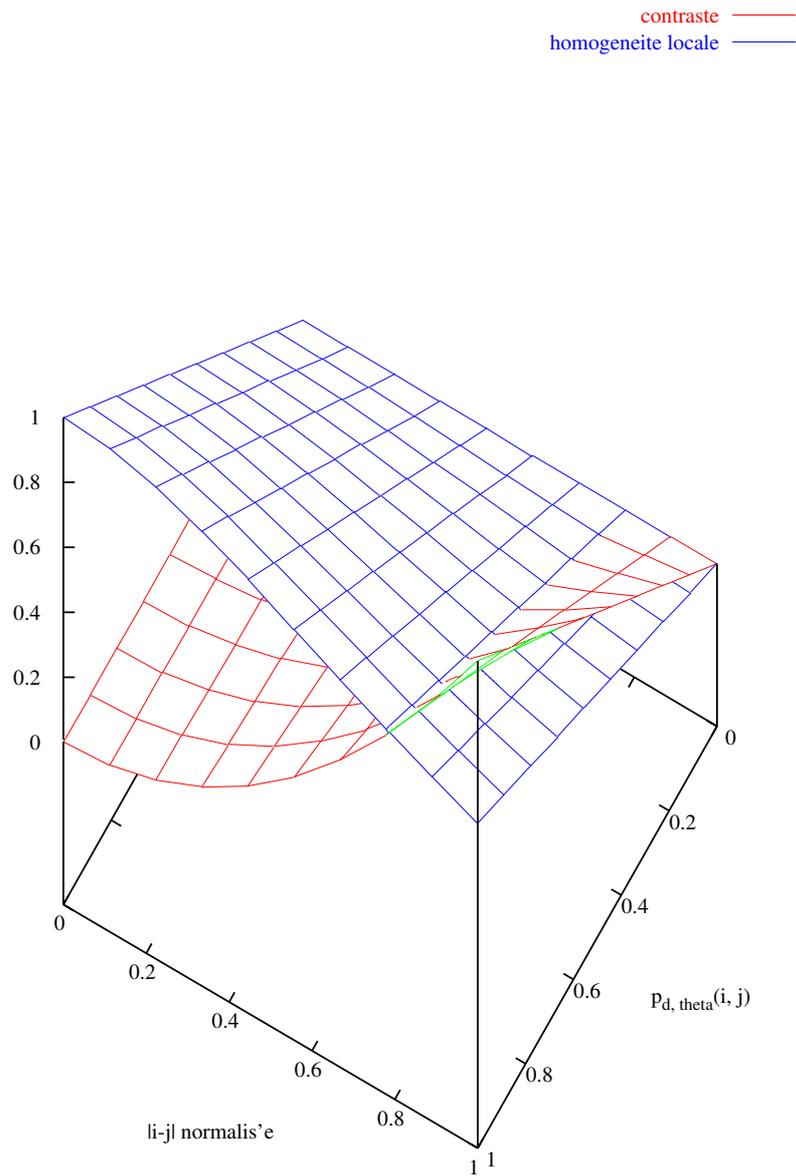


Figure 3.13 – Comportements comparés de la mesure du contraste et de celle de l'homogénéité locale

Corrélation La corrélation mesure l'uniformité des distributions des niveaux de gris :

$$corr = \frac{\sum_{i=0}^n \sum_{j=0}^n ij p_{d,\theta}(i, j) - \mu_i \mu_j}{\sigma_i \sigma_j} \quad (3.13)$$

Autres paramètres Les mesures suivantes sont rapidement introduites par souci d'exhaustivité mais leur interprétation n'est pas simple, leur valeurs sont *a priori* assez redondantes avec les précédentes et certaines sont coûteuses à calculer :

- moyenne des sommes :

$$ms = \sum_{k=2}^{2n-1} k m_{x+y}(k) \quad (3.14)$$

- variance des sommes :

$$vs = \sum_{k=2}^{2n-1} (k - ms)^2 m_{x+y}(k) \quad (3.15)$$

- entropie de la somme :

$$ents = - \sum_{k=2}^{2n-1} m_{x+y}(k) \log(m_{x+y}(k)) \quad (3.16)$$

- variance des différences :

$$vdf = \sum_{k=0}^n (k - \mu_{x-y})^2 m_{x-y}(k) \quad (3.17)$$

- entropie des différences :

$$entdf = - \sum_{k=0}^n m_{x-y}(k) \log(m_{x-y}(k)) \quad (3.18)$$

- corrélation de l'information 1 :

$$corr f1 = \frac{ent - H_{1XY}}{\max(H_X, H_Y)} \quad (3.19)$$

- corrélation de l'information 2 :

$$corr f2 = \sqrt{1 - e^{-2(H_{2XY} - ent)}} \quad (3.20)$$

- corrélation maximum :

$$corrmax = \sqrt{\lambda_2} \quad (3.21)$$

où:

- $m_x(i)$ est la somme de la ligne i ;
- $m_y(j)$ est la somme de la ligne j ;
- $m_{x+y}(i)$ est la somme de la i^e diagonale secondaire ;
- $m_{x-y}(i)$ est la somme de la i^e diagonale principale ;
- $\mu_i = \frac{1}{n} \sum_{x=0}^n m(i, x)$;

- $\mu_j = \frac{1}{n} \sum_{y=0}^n m(y, j)$;
- $\mu_{x-y} = \frac{1}{n} \sum_{y=0}^n m_{x-y}(y)$;
- $\sigma_i = \frac{1}{n} \sum_{x=0}^n (m(i, x) - \mu_i)^2$;
- $\sigma_j = \frac{1}{n} \sum_{x=0}^n (m(y, j) - \mu_j)^2$;
- $H_X = - \sum_{k=0}^n m_x(k) \log(m_x(k))$;
- $H_Y = - \sum_{k=0}^n m_y(k) \log(m_y(k))$;
- $H_{1XY} = - \sum_{i=0}^n \sum_{j=0}^n m(i, j) \log(m_x(i)m_y(j))$;
- $H_{2XY} = - \sum_{i=0}^n \sum_{j=0}^n m_x(i)m_y(j) \log(m_x(i)m_y(j))$;
- $Q = \{q(i, j) | i = 0, \dots, n; j = 0, \dots, n\}$ avec $q(i, j) = \sum_{k=0}^n \frac{m(i, k)m(k, j)}{m_x(i)m_y(j)}$;
- λ_2 est la deuxième des plus grandes valeurs propres de Q .

Après une longue recherche dans le domaine d'analyse de texture et notamment l'extraction de descripteurs de texture à l'aide des matrices de co-occurrences, il nous est apparu que les quatorze paramètres proposés par Haralick et al. ne sont pas tous utilisés, seul un ensemble restreint est appliqué pour tels types de texture et pour tel domaine (les combinaisons des paramètres peuvent changer selon la nature de la texture et le domaine d'application). Notre recherche d'extraction donc, s'est focalisée sur le regroupement de tous les paramètres fréquemment utilisés pour cette fin. Les huit premiers paramètres parmi ceux qui sont cités ci-dessus s'avèrent *a priori* les plus pertinents en terme de reconnaissance et description de texture. Les six derniers sont exclus de notre travail pour trois raisons. La première est qu'ils sont rarement utilisés pour décrire la texture. La seconde raison est qu'ils sont coûteux en temps de calcul comparés aux huit premiers paramètres et dernièrement, aucune définition claire et précise n'est fournie concernant ces paramètres; information qui est nécessaire pour le passage au niveau sémantique.

3.3.2.2 Technique de Tamura

Tamura et al. [290] adopte une approche basée sur des études psychologiques sur la perception visuelle humaine. Les auteurs proposent six descripteurs :

- grossièreté (*coarseness*) ;
- contraste (*contrast*) ;
- direction (*direction*) ;
- présence de lignes (*line-likeness*) ;
- régularité (*regularity*) ;
- rugosité (*roughness*).

La forte liaison de ces descripteurs, notamment les trois premiers, avec la perception humaine fait de cette représentation des textures un champ intéressant et exploitable pour la recherche par le contenu des images. Il semblerait que l'œil l'humain soit le plus sensible à la grossièreté de la texture, puis à son contraste et enfin à la direction.

Grossièreté La grossièreté est une mesure de la granularité de l'image [291]. Le calcul de ce paramètre est assez complexe et surtout très coûteux (*grosso modo* en $O(N \times M \times 2^{k_{\max}})$ où k_{\max} est une constante pouvant être relativement importante.

Il se décompose en quatre étapes :

1. Pour chaque valeur x du pixel positionné en (i, j) , c'est-à-dire que $x = \text{valeur}(i, j)$:

$$A_k(x, y) = \frac{1}{2^{2k}} \sum_{i=x-2^{k-1}}^{x+2^{k-1}-1} \sum_{j=y-2^{k-1}}^{y+2^{k-1}-1} x(i, j) \quad (3.22)$$

2. Ensuite, calculer respectivement les différences horizontales et verticales :

$$E_k^H(x, y) = |A_k(x + 2^k, y) - A_k(x - 2^k, y)| \quad (3.23)$$

$$E_k^V(x, y) = |A_k(x, y + 2^k) - A_k(x, y - 2^k)| \quad (3.24)$$

3. Puis, pour chaque pixel, choisir la valeur k qui maximise les différences horizontales et verticales :

$$\hat{k}(x, y) = \operatorname{argmax}_k \{E_k^H(x, y), E_k^V(x, y)\} \quad (3.25)$$

Cette valeur est censée représenter la taille d'un « granule » centré sur le pixel (x, y) .

4. Enfin, la valeur du paramètre de grossièreté est obtenue comme la moyenne des tailles des meilleurs granules trouvés sur l'ensemble de l'image :

$$f_{crs} = \frac{1}{N \times M} \sum_{x=1}^N \sum_{y=1}^M \hat{k}(x, y) \quad (3.26)$$

Contraste Le contraste mesure la variation des niveaux de gris dans l'image [291, 126]. Il est mesuré en fonction de l'écart-type σ et du kurtosis α_4 :

$$f_{con} = \frac{\sigma}{(\alpha_4)^\gamma} \quad (3.27)$$

où γ est une constante positive. Selon les expériences faites par Tamura et al., $\gamma = \frac{1}{4}$ donne les meilleurs résultats.

Direction Le paramètre de direction mesure l'orientation totale dans l'image, non les différentes directions qui peuvent éventuellement exister dans l'image [291, 126]. Le calcul de ce paramètre s'effectue en quatre étapes :

1. Calculer pour chaque pixel $p(i, j)$ de l'image, le gradient horizontal et vertical. Le gradient calculé sur un voisinage de pixels, reflète la direction des variations des niveaux de gris sur ce voisinage. Cela permet aussi de détecter les contours existants dans l'image.

Tamura a calculé le gradient horizontal Δ_h en faisant la différence entre les trois niveaux de gris du côté gauche du pixel $p(i, j)$ et les trois niveaux de gris du côté droit :

$$\Delta_h(i, j) = \sum_{k \in \{-1, 0, 1\}} (p(i + 1, j + k) - p(i - 1, j + k)) \quad (3.28)$$

Le gradient vertical Δ_v est la différence entre les trois niveaux de gris du dessus et du bas du pixel $p(i, j)$:

$$\Delta_v(i, j) = \sum_{k \in \{-1, 0, 1\}} (p(i+k, j+1) - p(i+k, j-1)) \quad (3.29)$$

2. Calculer la longueur e et l'angle d'orientation a sur chaque pixel comme suit :

$$e(i, j) = \frac{|\Delta_h(i, j)| + |\Delta_v(i, j)|}{2} \quad (3.30)$$

$$a(i, j) = \tan^{-1}\left(\frac{|\Delta_v(i, j)|}{|\Delta_h(i, j)|}\right) + \frac{\pi}{2} \quad (3.31)$$

3. Construire l'histogramme H en sommant tous les points dont la longueur e est supérieure à un certain seuil α et en tenant compte de la valeur de l'angle a du contour.

Plus précisément, $H(k)$ est égal au nombre de pixels tels que : $\frac{2k-1}{2n} < \frac{a}{\pi} \leq \frac{2k+1}{2n} \pmod{1}$ et $|e| > \alpha$ avec $k \in \{-1, 0, 1\}$.

4. Une fois l'histogramme construit, la variance des valeurs des pics est calculée. La direction de l'image est calculée en sommant ces variances.

Les trois descripteurs qui viennent sont fortement corrélés à ceux introduits précédemment.

Présence de lignes Ce paramètre détecte si l'image comprend des lignes ou non. Il est défini comme suit :

$$f_{lin} = \frac{\sum_{i=1}^M \sum_{j=1}^M P(i, j) \cos \frac{2\pi(i-j)}{M}}{\sum_{i=1}^M \sum_{j=1}^M P(i, j)} \quad (3.32)$$

où P est la matrice de co-occurrences calculée pour l'image. Il s'agit de la même notion introduite dans la section 3.3.2.1.

Régularité La régularité est donnée par :

$$f_{reg} = 1 - \alpha(\sigma_{crs} + \sigma_{con} + \sigma_{dir} + \sigma_{lin}) \quad (3.33)$$

où α est un facteur de normalisation.

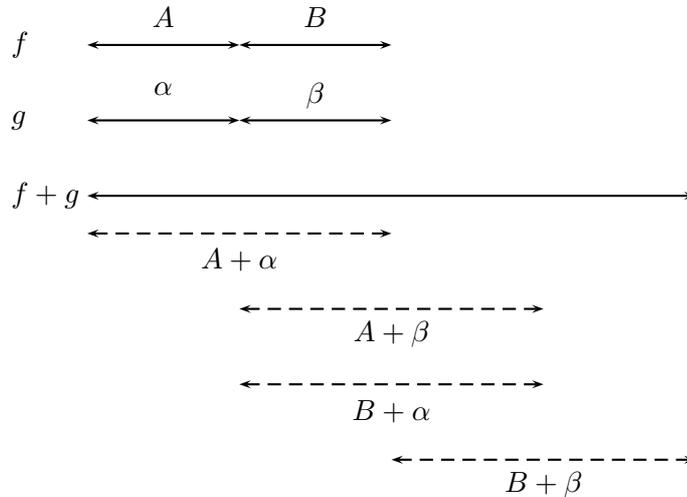
Cette définition additive s'exclut *de facto* de notre analyse. En effet, le découpage en variables linguistiques des descripteurs de base est perdu dans la forme additive. La figure 3.14 illustre cette perte d'information. Soit un descripteur f dont les valeurs sont traduites par les termes A et B et un descripteur g décrit par les termes α et β . Alors la somme de ces deux descripteurs n'apportera aucune information. Bien au contraire, elle va créer une importante zone d'incertitude. Ce phénomène doit être interprété comme la version discrète du théorème central-limite.

Rugosité Le paramètre de rugosité est simplement obtenu en sommant les descripteurs de grossièreté et de contraste :

$$f_{rgh} = f_{crs} + f_{con} \quad (3.34)$$

Ce paramètre est exclu pour la même raison que précédemment.

Après l'étape d'extraction des descripteurs de texture, vient la phase d'analyse de ces attributs afin d'obtenir un vecteur signature unique pour tout type de textures. Ce point sera détaillé dans la section suivante.

Figure 3.14 – Perte de précision *via* l’addition de propriétés discrétisées

3.4 Sélection des descripteurs de texture

Après la phase d’extraction des descripteurs de texture, la chaîne d’analyse des textures est poursuivie par une phase de réduction de dimension de ces descripteurs supposés de grande dimension et/ou redondants. Théoriquement, l’augmentation du nombre des descripteurs pourra améliorer les performances du système. Néanmoins, en pratique, l’utilisation d’un grand nombre de descripteurs, en plus du problème de complexité engendré par la dimension élevée de l’espace de représentation des données, peut en fait aboutir à une baisse des performances [68]. Ce phénomène est connu sous l’expression de « la malédiction de la dimensionalité » (en anglais. *curse of dimensionality*).

Malgré que le nombre de descripteurs choisis préliminairement pour l’analyse de textures est faible, de l’ordre de 8, la réduction de dimension de l’espace des descripteurs dans notre cas est indispensable pour deux raisons conjointes :

1. enlever la redondance entre les descripteurs ;
2. faciliter l’implémentation de ces descripteurs dans un système de gestion de bases de données (*SGBD*) standard dont les performances se dégradent une fois la dimension des variables excède 5 paramètres [120].

Les méthodes de réduction de dimension des données utilisées peuvent être divisées en deux classes :

- les méthodes d’extraction (*feature extraction*) ;
- les méthodes de sélection (*feature selection*).

Méthodes d’extraction La réduction de dimension par extraction consiste à générer de nouveaux paramètres à partir des paramètres initiaux avec une dimension plus petite. Il s’agit de projeter l’ensemble des données d’un espace de dimension \mathfrak{R}^p dans un espace de dimension \mathfrak{R}^q ($q \leq p$). Plusieurs techniques sont proposées dans ce cadre et sont subdivisées selon deux aspects : linéaire et non-linéaire. Parmi les méthodes linéaires [25, 26], on trouve :

- l’analyse en composantes principales (ACP) ;
- l’analyse linéaire discriminante (ALD) ;

- l'analyse factorielle des correspondances (AFC) ;
- etc.

Pour les méthodes non-linéaires, on trouve :

- l'analyse en composantes curvilinéaires (ACC) [60, 61] ;
- *Kernel PCA* [261] ;
- *Data-driven High Dimensional Scaling* (DD-HDS) [179] ;
- etc.

Méthodes de sélection des attributs Les méthodes de sélection sont la seconde option des méthodes de réduction de la dimension des descripteurs. Ces approches cherchent le sous-ensemble optimal (discriminant) des descripteurs pertinents. La pertinence de ce sous-ensemble est évaluée en fonction de l'objectif de l'application.

Formellement, soit Ω_x l'espace de représentation de D descripteurs $X_i, i = 1, \dots, D$. Il s'agit de trouver un sous-ensemble S_x de dimension d de l'espace Ω_x tel que $d < D$ tout en répondant aux critères suivants posés dans notre travail :

1. un taux important de reconnaissance ou prédiction dans la cas de la classification;
2. meilleurs taux de rappel et de précision dans le cas de la recherche d'images par contenu.

Les méthodes de réduction de dimension par sélection sont en plein essor depuis les années soixante et sont largement utilisées pour la catégorisation des textes [326, 81, 181], la classification [58, 187], en *data-mining* [57, 114, 216] et pour toutes les approches automatiques [113, 22]. Des introductions aux techniques de sélection sont données dans [140, 108, 216, 187].

Étant donné que le but principal de ces méthodes est de trouver un sous-ensemble optimal, une recherche exhaustive des différentes possibilités (s'il y a n paramètres initiaux, alors on aura 2^n possibilités à évaluer; par exemple pour $n = 14$, il lui correspond 32768 sous-ensembles) s'avère coûteuse en terme du temps de calcul et d'évaluation. Pour cela, plusieurs stratégies ont été proposées *Backward selection* (BS), *Forward selection* (FS), *Floating forward selection* (FFS) et *Floating backward selection* (FBS):

Backward selection Les méthodes de *backward selection* utilisent à la première itération, toutes les variables; retirent celle qui est la moins discriminante en terme du critère donné. Ensuite, on réitère sur le reste des variables et ainsi de suite, on élimine la variable la plus faible. La recherche s'arrête lorsque toutes les variables dans le modèle soient significatives. Cette approche a pour inconvénient est qu'elle ne permet pas de réévaluer la variable éliminée. Elle est efficace pour un nombre important de variables.

Forward selection À l'inverse des premières, cette approche commence par un ensemble vide de variables, puis à chaque itération, la variable qui donne un taux de discrimination maximal couplée avec celles déjà incluses est sélectionnée. Son inconvénient est qu'elle ne permet pas d'éliminer une variable qui devient inefficace après l'ajout d'autres variables. Par contre, elle est efficace pour un nombre petit de paramètres.

Floating forward & backward selection [235] Pour la méthode FFS, au début, le sous-ensemble optimal est initialisé à l'ensemble vide; et à chaque itération, une nouvelle variable est ajoutée. Après chaque insertion, on vérifie si on peut enlever des variables une après l'autre tout en conservant un bon taux de discrimination. Cette approche est une combinaison des deux premières. Pour la méthode FBS, elle est similaire à la méthode FFS sauf qu'elle débute avec l'ensemble complet de variables.

Une fois la méthode de recherche appropriée à l'application visée est choisie, on doit exécuter un algorithme de sélection des variables pour extraire le meilleur sous-ensemble. Deux catégories d'approches existent : les algorithmes dits *filtres* [57, 327] et les algorithmes *wrappers* [160].

Approches à base de filtres Ces algorithmes mesurent la pertinence du sous-ensemble à sélectionner indépendamment du classifieur ou un critère donné. Il s'agit d'ordonner les différentes variables selon la pertinence du score qui leur a été attribué durant l'évaluation des sous-ensembles [158, 124]. Plusieurs algorithmes sont proposés, nous avons choisi d'étudier l'algorithme CFS (Correlation-based Feature Selection) [113, 327] basés sur l'incertitude symétrique, une autre version du CFS basée sur la corrélation de Pearson (PCFS) [133] et l'information mutuelle (MIFS) [9, 170, 169, 230, 78, 165, 220]. Les détails de ces algorithmes sont présentés dans la suite.

Approches à base de wrappers Au contraire de la première approche, les algorithmes *wrappers* définissent la pertinence des attributs par l'intermédiaire d'une prédiction de la performance du système final (classification par exemple). Nous avons choisi dans ce cas quatre algorithmes : *ReliefF* [158, 246], *Fisher* ou analyse discriminantes [254, 333], les arbres de décision [311] et les algorithmes génétiques [229, 325].

3.4.1 Algorithme CFS

Dans sa thèse, Hall [112] a introduit une méthode heuristique de sélection des variables basée sur la notion de mérite. Cette heuristique tient en compte la capacité de chaque variable à prédire la classe d'appartenance selon leur niveau d'intercorrélation. L'hypothèse sur laquelle est basée cette heuristique provient de cette constatation : « *Good feature subsets contain features highly correlated with (predictive of) the class, yet uncorrelated with (not predictive of) each other.* »

La mesure de mérite d'un sous-ensemble S utilisée dans le CFS est une version empruntée de la théorie des mesures [96] donnée par la formule :

$$\text{Merit}_S = \frac{k\bar{r}_{cf}}{\sqrt{k + k(k-1)\bar{r}_{ff}}} \quad (3.35)$$

avec

- k le nombre de variables dans S ;
- \bar{r}_{cf} la corrélation moyenne entre le descripteur $f \in S$ et la classe d'appartenance c ;
- \bar{r}_{ff} la moyenne de l'inter-corrélation de f .

Le numérateur donne une indication sur la prédiction d'une classe par l'ensemble S de descripteurs, tandis que le dénominateur mesure la redondance des descripteurs au sein du même groupe.

L'algorithme CFS (*Correlation-based Feature Selection*) se déroule comme suit :

1. discrétisation des variables

Dans le cas où les variables à traiter sont continues, l'algorithme CFS procède à une discrétisation⁵ selon la méthode de Fayyad et Irani [73].

2. calcul des corrélations

⁵La discrétisation consiste à transformer une variable continue en une variable discrète par découpe de sa dynamique. Cette notion sera plus détaillée dans le chapitre 6.

Pour l'algorithme CFS [113, 327], les corrélations \bar{r}_{cf} et \bar{r}_{ff} sont mesurées en fonction d'une nouvelle entité dite *incertitude symétrique*. Elle est basée sur l'entropie conditionnelle. Sa formule est donnée par l'équation 3.36 :

$$\text{incertitude symétrique} = 2 \times \frac{\text{gain}}{H(Y) + H(X)} \quad (3.36)$$

où :

- *gain* est le gain d'information mesurant la diminution de l'incertitude sur Y lorsque X est réalisé :

$$\text{gain} = H(Y) - H(Y|X) \quad (3.37)$$

$$= H(Y) + H(X) - H(X, Y) \quad (3.38)$$

- $H(Y)$ et $H(Y|X)$ mesurent respectivement l'entropie de Y avant et après observation de X :

$$H(Y) = - \sum_{y \in Y} p(y) \log_2(p(y)) \quad (3.39)$$

$$H(Y|X) = - \sum_{x \in X} p(x) \sum_{y \in Y} p(y|x) \log_2(p(y|x)) \quad (3.40)$$

3. évaluation de la valeur de la fonction mérite

L'incertitude est remplacée dans l'équation du mérite. L'ensemble qui possède la plus grande valeur est sélectionné.

3.4.2 Algorithme MIFS

La limite des algorithmes d'ordonnement tels que CFS et PCFS est qu'ils traitent chaque paramètre individuellement sans tenir en compte l'information qui peut être apportée en interaction avec d'autres paramètres. Une solution alternative a été proposée par l'algorithme PaCFS.

Une autre alternative est donnée par l'algorithme MIFS (*Mutual Information Feature Selection*) proposé par Battiti [9]. L'algorithme MIFS consiste à chercher le sous-ensemble S , de dimension d inférieure à celle de l'ensemble total des descripteurs initiaux, qui maximise l'information mutuelle $IM(C, S)$ entre cet ensemble et la variable C des classes d'appartenance.

L'information mutuelle est largement utilisée pour la sélection des attributs [9, 230, 165]. En général, elle mesure la quantité d'information d'une variable contenue dans une seconde. Ainsi, lorsque cette valeur est maximale, ces deux variables sont dites « identiques ». Sélectionner le descripteur étant le plus lié à la classe C peut donc se faire en maximisant leur information mutuelle. Généralement, cette information est basée sur la notion d'entropie. Néanmoins, on trouve 3 définitions équivalentes dont chacune permet d'expliquer différemment cette information.

Entropie L'entropie est la quantité d'information contenue dans une source d'information. Cette source peut être un système physique, une image ou autre. Ici, on considère comme source d'information l'ensemble des réalisations d'un descripteur $X = (x_1, \dots, x_n)$ ayant tous une probabilité p_i d'être réalisée. Plus les p_i sont équiprobables, plus son entropie $H(X)$ est grande. Shannon [267] propose une définition de l'entropie telle que :

$$H(X) = - \sum_i p_i \log(p_i) \quad (3.41)$$

Cette expression montre bien que plus un élément est rare, plus il a de signification. Par exemple, dans le cas d'analyse d'images, si une image est constituée de pixels de plusieurs niveaux de gris, alors l'image contient une information plus importante qu'une image d'un seul niveau de gris. La définition de Shannon de l'entropie indique l'information moyenne que l'on peut tirer de chaque élément de l'image.

Information mutuelle Soit X une variable à n réalisations ($X = x_1, \dots, x_n$) et C variable discrète à k classes ($C = c_1, \dots, c_k$).

La première définition de l'information mutuelle $IM(C, X)$ utilise la différence de l'entropie de X et de l'entropie de la même variable X sachant la classe C :

$$IM(C, X) = H(X) - H(X|C) = H(C) - H(C|X) \quad (3.42)$$

$H(X|C)$ mesure la quantité d'information contenu dans X lorsque la classe C est fournie. L'information mutuelle correspond donc à la quantité d'information que C possède sur X , ou inversement, la quantité d'information que X possède sur C .

La seconde définition évoque la distance de Kullback-Leibler qui mesure la distance entre deux distributions :

$$IM(X, C) = \sum_{x \in X} \sum_{c \in C} p_{xc} \log \frac{p_{xc}}{p_x p_c} \quad (3.43)$$

Cette définition de l'information mutuelle mesure donc la dépendance entre X et C . L'attribut X sera sélectionné s'il est fortement corrélé à C .

La troisième définition de l'information mutuelle est une combinaison des entropies séparées et jointes des deux variables :

$$IM(C, X) = H(X) + H(C) - H(X, C) \quad (3.44)$$

L'entropie jointe $H(X, C)$ mesure la quantité d'information que les deux variables apportent en même temps. S'ils sont proches, on dira qu'une variable explique bien la seconde et l'entropie jointe est minimale.

Limitations Dans le cas où la dimension de l'espace des attributs est assez importante $D \gg$, le calcul de l'information mutuelle IM entre tous les sous-ensembles candidats et la classe C s'annonce onéreux en temps de calcul et d'itérations. Si la taille du sous-ensemble à choisir est k , alors on aura 2^{k+1} probabilités à estimer [78]; par exemple, prenons $D = 1000$ et $k = 19$, alors $2^{20} = 1\,048\,576$ probabilités à calculer.

Pour pallier à cette limitation, d'autres alternatives sont proposés utilisant l'information mutuelle conditionnelle [9, 169, 170, 78, 220].

L'information mutuelle conditionnelle IMC donnée par Battiti [9] :

$$IMC_{Battiti}(C, X_i|S) = IM(C, X_i) - \beta \sum_{X_s \in S} IM(X_s, X_i) \quad (3.45)$$

Dans [169, 170], Kwak et Choi ont proposé l'algorithme $MIFS - U$ tel que l'information mutuelle conditionnelle est donnée par la formule suivante :

$$IMC_{Kwak}(C, X_i|S) = IM(C, X_i) - \beta \sum_{X_s \in S} \frac{IM(C, X_s)}{H(X_s)} IM(X_s, X_i) \quad (3.46)$$

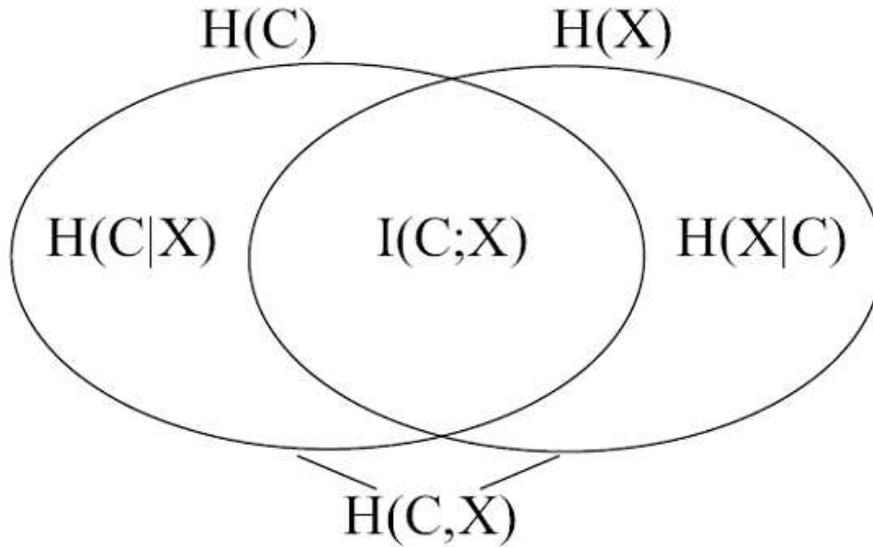


Figure 3.15 – Diagramme de Venn de l'information mutuelle et l'entropie conditionnelle pour une classe C et une variable X [106]

Le second terme dans 3.45 et 3.46 mesure la redondance entre le descripteur candidat X_i et les descripteurs sélectionnés X_s suivant la concordance avec C . Le paramètre β est un facteur de contrôle de la redondance. Peng et al [230] ont montré que pour maximiser $IM(C, S)$ par une recherche séquentielle en amont (SFS), la valeur adéquate de β est $\frac{1}{|S|}$ où $|S|$ est le nombre de descripteurs sélectionnés.

3.4.3 Algorithme ReliefF

L'algorithme ReliefF est une version améliorée de l'algorithme Relief introduit par Kira et Rendell [158] et qui est limité à la classification binaire (deux classes). L'algorithme ReliefF est étendu au cas multi-classes par Kononenko [246].

L'algorithme ReliefF, de plus de sa capacité de détecter et d'éliminer les attributs redondants, permet de définir un critère de pertinence basé sur le calcul des poids (scores). Ce critère mesure la capacité de chaque attribut à regrouper les données provenant de la même classe c et discriminer celles ayant des classes différentes (variables $hits_c$ et $miss_c$, lignes 4 et 5). Les poids de tous les attributs sont mis à jour en tenant compte des valeurs de D_j , $hits_k$ et $miss_{kc}$. Si la donnée D_j et la moyenne des $hits_k$ ont des valeurs différents pour l'attribut X_i alors ce dernier n'est pas sélectionné car il sépare deux instances d'une même classe. Autrement, si l'instance D_j a une valeur différente de $miss_{kc}$ alors l'attribut X_i est pris puisqu'il sépare deux instances de classes différentes. L'algorithme est décrit dans le tableau 1 avec $P(c)$ est la probabilité *a priori* d'avoir la classe c et $(1 - P(\text{classe}(D_j)))$ représente la somme des probabilités des classes non détectées.

La fonction $diff(D_j, X_i, D_k)$ calcule la différence entre l'attribut X_i et deux instances D_j et D_k . Sa formule est comme suit :

$$diff(D_j, X_i, D_k) = \frac{|X_{i,j} - X_{i,k}|}{\max(X_i) - \min(X_i)} \quad (3.47)$$

Un poids positif indique une corrélation forte entre l'attribut et sa classe d'appartenance. Autrement

Algorithme 1 Algorithme ReliefF**Entrées :**

- N : vecteurs descripteurs ;
- $\Omega : (X_1, \dots, X_N)$;
- C : classe d'appartenance ;
- M : nombre de données ;
- m : nombre d'itérations imposé par l'utilisateur.

1. $\omega_i \leftarrow 0, \forall i \in \{1, \dots, N\}$ // Initialiser les poids
2. **pour** $j = 1$ à m avec $m < M$ **faire**
3. Tirer aléatoirement une donnée D_j
4. Trouver les K plus proches voisins de D_j appartenant à la même classe (*hits*)
5. Dans chaque classe $c \neq \text{classe}(D_j)$, trouver les K plus proches voisins de D_j appartenant à des classes différentes (*miss_{kc}*)
6. **Pour** $i \leftarrow 1$ à N **faire** // Mettre à jour les poids
7.
$$\omega_i \leftarrow \omega_i - \sum_{k=1}^K \frac{\text{diff}(D_j, X_i, \text{hits}_k)}{mK} + \sum_{c \neq \text{classe}(D_j)} \frac{P(c)}{1 - P(\text{classe}(D_j))} \sum_{k=1}^K \frac{\text{diff}(D_j, X_i, \text{miss}_{ck})}{mK}$$
8. **fin pour**
9. **fin pour**
- 10.

dit, il est d'autant plus grand que les données issues de la même classe ont des valeurs proches et que les données issues de classes différentes sont bien séparées. Les scores attribués aux différents paramètres permet de les ordonner et donc sélectionner le meilleur sous-ensemble. Cet algorithme est rapide, sa complexité est $O(m.N.M)$.

3.4.4 Algorithme Fisher

L'algorithme Fisher repose sur l'analyse discriminante linéaire de Fisher (AFD) [88]. De façon générale, un problème de discrimination linéaire, à deux classes, revient à séparer l'espace des données en deux espaces grâce à un hyperplan. Choisir la classe (à valeur dans -1 et 1) d'une donnée consiste alors à déterminer de quel côté de l'hyperplan elle se situe, ce qui se traduit par la formule suivante :

$$g(D_j) = \text{sign}(\omega \cdot D_j + b) \quad (3.48)$$

où $D_j \in \mathbb{R}^N, j = 1 \dots M, \omega \in \mathbb{R}^N$. b est un biais d'estimation et $g(D_j)$ la classe prédite pour la donnée D_j .

L'objectif de l'AFD est de trouver la droite sur laquelle les données projetées sont mieux séparées. Cela dit, celle qui maximise le rapport des variances inter et intra classes des projections. Lorsque l'on fait l'hypothèse que les attributs sont non corrélés et suivent sur chaque classe (+, -) une distribution gaussienne $N(\mu^+, \sigma^+)$ et μ^-, σ^- , l'AFD permet de distinguer les deux classes en mesurant le chevauchement de leurs fonctions de densité de probabilité. Ainsi, les scores des attributs seront estimés par :

$$\omega_i = \frac{(\mu_i^+ - \mu_i^-)}{(\sigma_i^+)^2 + (\sigma_i^-)^2} \quad (3.49)$$

Le formule peut être étendue au cas multi-classe en considérant qu'il s'agit d'un ensemble de problèmes à deux classes du type un contre tous (*one-all*).

3.4.5 Algorithmes génétiques

Les algorithmes génétiques, proposés par Holland [123], sont des algorithmes d'optimisation s'appuyant sur des techniques dérivées de la génétique et de l'évolution naturelle : croisements, mutations, sélection, etc. Actuellement, ces algorithmes sont utilisés pour la sélection et l'extraction des paramètres [229, 325, 243]. La figure 3.16 illustre le principe du fonctionnement des algorithmes génétiques. L'opération consiste à sélectionner aléatoirement un sous-ensemble de paramètres, calculer sa fonction de fitness en l'injectant dans un classifieur. L'ensemble ayant le plus grand taux de classification est sélectionné.

Une utilisation directe des algorithmes génétiques pour la sélection des paramètres est faite par Siedlecki et Sklanski [270]. Ils ont utilisé les algorithmes génétiques pour chercher un vecteur binaire optimal où chaque bit est associé à un paramètre. Si le i^e bit de ce vecteur est égal à 1, alors le i^e paramètre est utilisé dans la classification; sinon si ce bit est égal à 0, alors le paramètre correspondant est éliminé de la classification. Ensuite, chaque sous-ensemble obtenu est évalué selon le taux de classification estimé par le classifieur.

Par analogie, dans ce travail, nous utilisons les algorithmes génétiques pour la sélection des paramètres les plus pertinents en terme de classification des textures. Dans ce cas, chaque paramètre est associé à un gène du chromosome de représentation; un *chromosome* est un vecteur de bit de longueur égale au nombre de paramètres; un nombre de chromosomes constitue une *population*. Ainsi, une valeur nulle du gène signifie que le paramètre associé ne fera pas partie des paramètres utilisés dans l'étape de classification comme illustré par la figure 3.17.

L'objectif ici est de sélectionner le chromosome ou le sous-ensemble de paramètres le plus pertinent de taille minimal tout en gardant un bon taux de classification. L'évaluation d'une population et donc un sous-ensemble de paramètres est effectuée suivant une fonction de qualité dite fonction d'adaptation ou *fitness*. Cette fonction est objective, dépendant de l'application envisagée qu'il faudra la maximiser ou minimiser. Dans notre cas, la fonction *fitness* dépend du taux de classification à maximiser, par la suite, nous avons choisi le modèle suivant comme fonction d'évaluation :

$$\text{fitness}_p = (\tau_p + f_p^0) \times 10^2 \quad (3.50)$$

où τ_p est le taux de classification trouvé pour un chromosome de taille p paramètres et f_p^0 est la fréquence des paramètres nuls non utilisés lors de la classification. Si le taux de classification atteint des valeurs maximales, cela implique que le nombre des paramètres nuls est faible et par la suite la valeur de fitness_p est grande. Dans le cas inverse, s'il s'agit d'une mauvaise classification (taux faible), c'est-à-dire que les paramètres utilisés pour la classification sont insuffisants augmentant ainsi le nombre des paramètres nuls dans le chromosome et par la suite, la valeur de la fitness_p s'incrémente avec la fréquence f_p^0 .

Ensuite, on réitère cette évaluation avec la nouvelle population créée à partir de trois opérations essentielles : *sélection*, *croisement* et *mutation* (cf. figure 3.18).

Sélection La première étape d'une recherche par AG consiste à sélectionner une sous-population (nombre de chromosomes) à partir de la population parent. La probabilité qu'un chromosome fera partie de la nouvelle population est directement liée à son efficacité au sein de la population. Il existe plusieurs méthodes de reproduction de population. La méthode la plus courante est celle initiée par Holland [123] : la roulette de loterie biaisée (*roulette wheel selection*). Cette méthode de sélection est proportionnelle

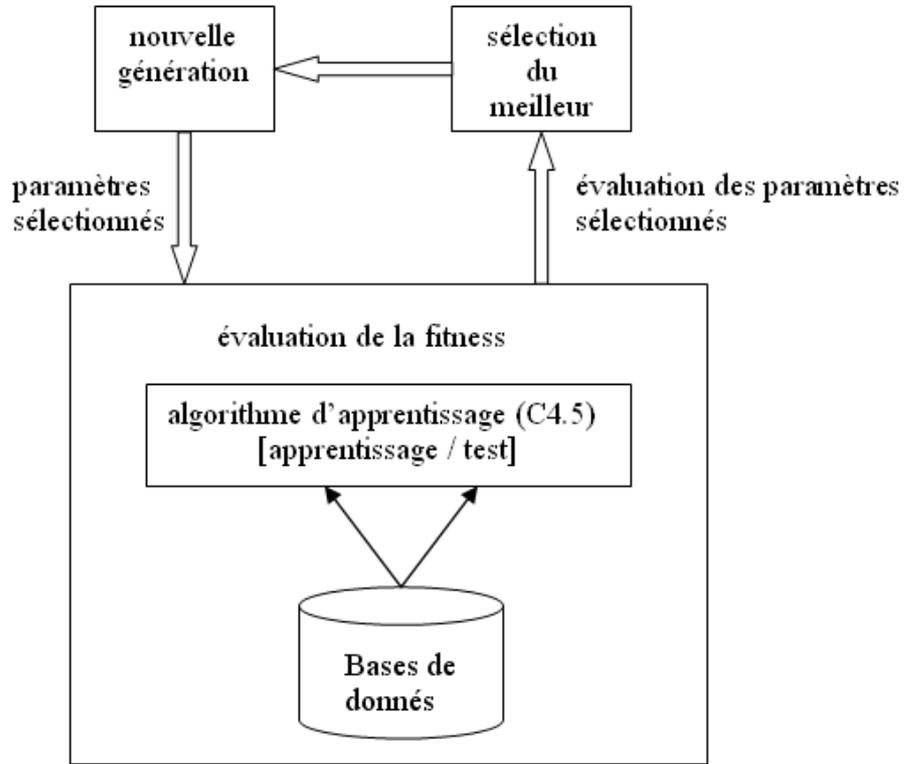


Figure 3.16 – Sélection des paramètres par les AGs [143]

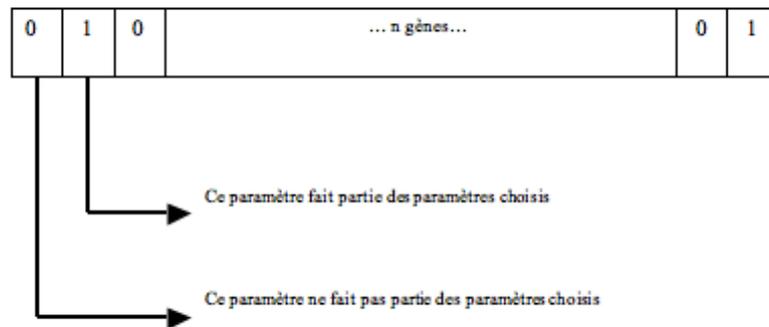


Figure 3.17 – Chromosome pour la sélection de paramètres

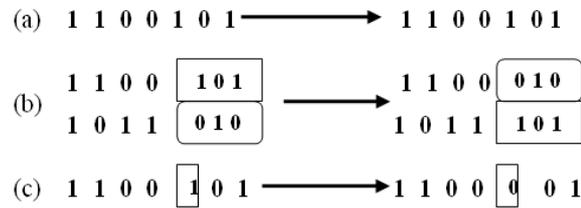


Figure 3.18 – Les trois opérations de base dans les GA : (a) sélection, (b) croisement et (c) mutation

au niveau de *fitness* des chromosomes. Le nombre de fois qu'un chromosome sera sélectionné est égal à sa *fitness* divisé par la moyenne du *fitness* de la population totale (plus exactement, la partie entière représente le nombre de fois qu'il sera sélectionné, et la partie flottante la probabilité qu'il aura d'être sélectionné à nouveau). Cette fonction est déterminante dans un algorithme génétique. De nombreuses autres méthodes de sélection, bien plus complexes sont disponibles : le sigma (*scaling*), la sélection à la Boltzman, la sélection par rang, la sélection par tournois, etc.

Croisement L'opération de croisement (*cross-over*) est un moyen majeur qui permet la création de nouveaux individus à partir des anciens. Le croisement consiste dans un premier temps à tirer aléatoirement des chromosomes deux à deux. Ensuite, choisir aléatoirement le point de croisement (*locus*) (chiffre entre 1 et $p-1$ où p est la taille du chromosome). Dans la figure 3.18(b), le point de croisement est 4. Puis, échanger autour de ce point les gènes des deux chromosomes parents pour obtenir deux nouveaux chromosomes enfants. C'est le principe du *single-point cross-over*. Dans un cadre général, l'échange peut s'effectuer suivant une probabilité p_{cross} .

Mutation L'opération de mutation est d'importance secondaire, mais permet d'éviter une convergence prématurée vers un maximum local en maintenant une diversité de solution. Pour l'appliquer, choisir au hasard un bit du chromosome et modifier sa valeur. La mutation ne doit pas être appliquée systématiquement, mais en fonction d'une probabilité p_{mut} , paramètre de la simulation. Dans le cas du codage binaire, chaque bit est remplacé par son inverse (cf. figure 3.18 (c)). Généralement, les probabilités de l'ordre de 0,01 à 0,03 sont choisies.

Les méthodes de sélection présentées dans cette partie ont pour but de trouver le meilleur sous-ensemble de paramètres de texture à partir des matrices de co-occurrences et les paramètres de Tamura. La pertinence de cet ensemble est évaluée en fonction des taux de classification et de discrimination des textures.

3.4.6 Nos propositions

Les différentes techniques que nous avons présenté en haut pour la réduction de la dimension de l'espace de représentation des attributs de texture et la sélection des plus pertinents sont des techniques utilisées dans le cas supervisé où nous disposons *a priori* de l'information sur les classes d'appartenance des images. Or, cette information n'est pas toujours disponible notamment dans le cas des bases généralistes provenant de diverses sources, ce qui est le cas de notre cadre de travail puisque les bases de textures utilisées n'appartiennent pas à un domaine spécifique mais proviennent de différentes sources (architecture, nature, art, etc).

Algorithme 2 Algorithme génétique canonique (AGC)**Entrées :**

- N : vecteurs descripteurs ;
 - $\Omega : (X_1, \dots, X_N)$;
 - C : classe d'appartenance ;
 - M : nombre de données ;
 - m : nombre d'itérations imposé par l'utilisateur.
1. Codage des paramètres (passage du réel au binaire)
 2. Génération aléatoire de la population initiale de chromosomes
 3. Calculer la fonction de *fitness* pour ces chromosomes
 4. **répéter**
 5. Sélection
 6. Croisement
 7. Mutation
 8. Recalcul de la fonction de *fitness* pour les nouveaux chromosomes
 9. **jusqu'à** atteindre un critère d'arrêt

Liée à l'information d'appartenance qu'exigent ces techniques de sélection, ces dernières mesurent la pertinence apportée par les paramètres à choisir en fonction de leurs forte corrélation avec les classes d'appartenance, ce qui peut entraîner à une possibilité d'avoir un ensemble de paramètres redondants qui peut biaiser les résultats de la recherche.

Dans le paragraphe qui suit, nous allons introduire deux métriques qui permettent de sélectionner un sous-ensemble optimal pour la description de la texture tout en palliant les contraintes rencontrées par les techniques citées en haut à savoir l'information d'appartenance et la redondance des paramètres.

3.4.6.1 Algorithmes PCFS et PaCFS

La première métrique que nous proposons est le PCFS [133] (Pearson Correlation-based Feature Selection) est une variante du CFS qui utilise le coefficient de corrélation Pearson proposé par Galton [89] au lieu de l'incertitude. Ce paramètre largement utilisé en statistique et analyse de données permet de juger de la qualité de dépendance entre deux variables sans la nécessité d'une information supplémentaire telle que les classes d'appartenance, ce qui est le cas du CFS. Pour cette raison et puisque nous travaillons dans un cadre générale de description de textures quelconques, la plupart du temps, cette information sur les étiquettes des classes n'est pas toujours fournie, nous avons opté pour ce paramètre. De plus, ce paramètre est applicable à tous types de variables, donc aucune nécessité d'effectuer une discrétisation des paramètres.

La formule standard du coefficient de corrélation est donnée comme suit :

$$R_{XY} = \frac{Cov(X, Y)}{\sigma_X \sigma_Y} \quad (3.51)$$

avec :

- $Cov(X, Y)$ est la covariance entre X et Y ;
- σ_X, σ_Y respectivement l'écart-type de X et Y .

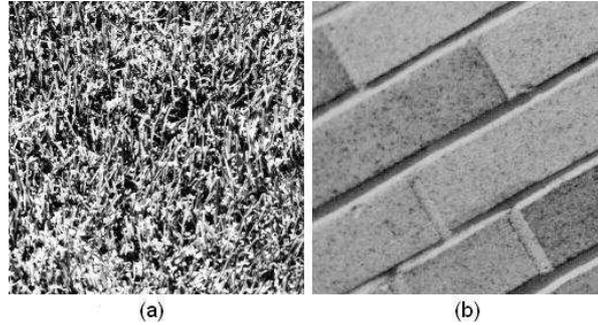


Figure 3.19 – Corrélation entre contraste et direction ?

Le calcul du coefficient de corrélation permet de détecter les couples de descripteurs dépendants (X, Y) et donc redondants. La sélection du meilleur descripteur dans un couple est décidée suivant la meilleure valeur du taux (τ) fourni par l'un des attributs.

Un dernier avantage de cette approche par rapport à la CFS est la rapidité d'exécution puisque après identification de couples redondants, chaque paramètre est évalué seul en fonction de sa valeur τ au lieu d'examiner toutes les possibilités de S jusqu'à obtention de la meilleure valeur de $Merits_S$ pour l'algorithme CFS.

Algorithme 3 PFCS

Entrées :

- N : vecteurs descripteurs ;
- $\Omega : (X_1, \dots, X_N)$.

1. $S \leftarrow \emptyset$
 2. $\tilde{R}_{XY} \leftarrow \{R_{XY}, \forall X, Y \in \Omega\}$ // Calcul de la matrice de corrélation
 3. **pour** $X \in \Omega$ **faire**
 4. $Y \leftarrow \operatorname{argmax}_Y(R_{XY})$ // Recherche
 5. $X^* \leftarrow \operatorname{argmax}_{X,Y}(\tau_X, \tau_Y)$ // Calcul
 6. $S \leftarrow S \cup \{X^*\}$ // Ajout d'attributs
 7. **fin pour**
 8. **retourner** S
-

La limitation de cet algorithme provient du fait qu'une valeur importante du coefficient de corrélation R_{XY} ne signifie pas forcément une dépendance logique entre descripteurs; par exemple, sur la base de Brodatz 1 et avec les paramètres MCO, nous trouvons toujours une forte corrélation entre le descripteur direction et le contraste. Autrement dit, selon le PCFS, si la texture est directionnelle, alors forcément elle doit être contrastée et l'inverse; alors que réellement, ce n'est pas le cas. La figure 3.19 illustre ce cas de figure. D'où la nécessité de faire intervenir une troisième variable explicative.

L'alternative que nous proposons actuellement PaCFS, dans le cas supervisé où nous disposons *a priori* d'une connaissance de la vérité terrain et les différentes classes existantes dans le cas par exemple de la base Brodatz 1 et Meastex, est d'utiliser la corrélation partielle qui prend en compte cette information supplémentaire.

Par sa définition standard, la corrélation partielle permet d'évaluer la corrélation entre deux variables X et Y tout en maintenant en constance une troisième variable Z . Dans le cas supervisé, cette dernière variable ne sera autre que la variable des classes d'appartenance C . La formule de la corrélation partielle est comme suit :

$$R_{XY,C} = \frac{R_{XY} - R_{XC}R_{YC}}{\sqrt{(1 - R_{XC}^2)}\sqrt{(1 - R_{YC}^2)}} \quad (3.52)$$

Par analogie au PCFS, le PaCFS détecte aussi les couples de redondance (ligne 6 du PaCFS). Dans ce cas, l'élimination d'un des deux attributs est évaluée en fonction de sa corrélation avec la classe d'étiquetage, si elle fournit une forte valeur, l'attribut est conservé sinon il sera éliminé (les lignes 7 et 8 du PaCFS). L'algorithme s'arrête jusqu'à aucune corrélation entre descripteurs n'est plus significative (inférieure au seuil fourni).

Algorithme 4 PaCFS

Entrées :

- N : vecteurs descripteurs ;
 - $X : (X_1, \dots, X_N)$;
 - C : classe d'étiquetage ;
 - α : seuil de corrélation (condition d'arrêt).
1. $S \leftarrow X$
 2. $\max \leftarrow 1$
 3. **tant que** $\max > \alpha$ **faire**
 4. $\tilde{R}_{XY,C} = \{R_{XY,C}, \forall X, Y \in \Omega\}$ // Calculer la matrice de corrélation partielle
 5. $\tilde{R}_{XC} = \{R_{XC}, \forall X\}$ // Calculer la matrice de corrélation de Pearson
 6. **pour** $X \in \Omega$ **faire**
 7. $XY_{\max} = \operatorname{argmax}_{XY}(R_{XY,C})$ // Rechercher
 8. $X_{\min} = \operatorname{argmin}_{XY}(R_{XC})$ // Évaluer
 9. $S = S \setminus \{X_{\min}\}$ // Éliminer l'attribut
 10. **fin pour**
 11. **fin tant que**
 12. **retourner** S
-

La complexité de ces deux algorithmes est de

L'évaluation des attributs caractéristiques de la texture peut faire l'objet de plusieurs applications. Dans notre thèse, nous proposons deux applications à savoir : la recherche d'images par contenu et la classification des images texturées.

3.4.7 Recherche d'images similaires sur *ClickImAGE*

Au sein de l'équipe Atlas-Grim du laboratoire d'informatique de Nantes Atlantique (LINA), le système de recherche d'images par contenu *ClickImAGE* a été développé sous la supervision de José Martinez par Erwan Loisant [210], fonctionnant uniquement avec la couleur. Notre travail s'inscrit donc dans le cadre d'une extension de ce système en y ajoutant des attributs de texture (cf. figure 3.20).

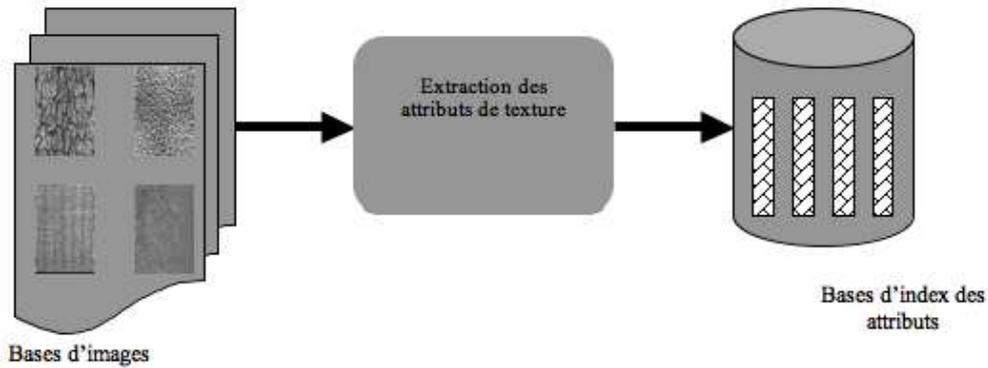


Figure 3.20 – Module d'extraction de texture dans ClickImAGE

Le type de requête de recherche d'images par contenu que supporte *ClickImAGE* est la recherche par image exemple. Le principe de la recherche par exemple est basé sur le calcul d'une mesure de similarité (ou dissimilarité) entre l'image fournie par l'utilisateur et les images stockées dans le système. Le calcul de la similarité est réellement effectué sur les attributs de texture extraits pour l'image requête et les images de la base. Cela, peut être formulé comme suit.

Soit BI la base d'images disponible, ($BI = \{I_m | m = 1, \dots, M\}$) dont chaque image $I_m \in \mathbb{R}^N$ est représentée par un vecteur d'attributs $X_m = \{X_{m,n} | n = 1, \dots, N\}$ et la requête Q indexée par $X_q = \{X_{q,n} | n = 1, \dots, N\}$. La mesure de similarité $D(I_q, I_m)$ entre la i^e image et la requête Q est calculée comme suit :

$$D(I_q, I_m) = \sum_{n=1}^N \omega_n d_n(X_{qn}, X_{mn}) \quad (3.53)$$

où d est une distance ou mesure de dissimilarité et ω est un poids qui permet d'attribuer à chacun des descripteurs un poids selon son importance dans la recherche. Par exemple, si l'on cherche des images « coucher du soleil »; dans ce cas, l'attribut couleur sera pondéré fortement; sinon, dans le cas de la recherche des images de panthères, la texture se voit plus importante.

Deux cas de figures se présentent lors de la recherche d'images que nous verrons en détail dans le chapitre 5 :

1. image appartenant à la base d'images; il s'agit dans ce cas de chercher l'index des attributs lui correspondant dans la base stockée, ensuite effectuer une similarité comme mentionné en haut.
2. image externe à la base d'images; dans ce cas, une étape de calcul des attributs de l'image requête est nécessaire suivie d'une mesure de similarité.

3.4.8 Classification des images texturées

Dans ce travail, l'objectif d'utiliser la classification est de vérifier la performance des descripteurs de texture choisis (cf. section 3.3) et sélectionnés pour le modèle de texture à trouver les classes appropriées aux images fournies au système.

La tâche de classification des images consiste à assigner des étiquettes de classes aux images. La classification est largement utilisée dans divers domaines tels que pour la reconnaissance optique de caractères, la reconnaissance des visages, etc. Disposant d'un ensemble de données étiquetées, le système de classification assigne à chaque donnée une classe $c \in \{1, \dots, C\}$. Pour classifier une image I , il s'agit de trouver une fonction ou règle de décision r telle que :

$$r : \begin{array}{l} \mathcal{R}^N \rightarrow \{1, \dots, C\} \\ I \mapsto c \end{array} \quad (3.54)$$

Une manière de définir une règle de décision r est de déterminer et stocker la classe de chaque image. Ainsi, transformer le problème de classification en un problème de recherche. Cette proposition est impossible puisque le nombre d'images disponibles augmentent d'une manière considérable.

Une autre proposition faisable pour construire la règle de décision r est d'utiliser une base d'apprentissage, représentée dans notre cas par les attributs de texture $\{X_1, \dots, X_N\}$, répartie en K classes, notées C (C_1, \dots, C_K). La règle de décision dans ce cas vérifie une fonction de discrimination $f(X_i, C_k)$. Dans ce travail, la fonction $f(X_i, C_k)$ correspond aux taux de classification estimé. Ce taux doit être important dans le cas d'apprentissage et de test :

$$r(X_i) = \operatorname{argmax}_k \{f(X_i, C_k)\} \quad (3.55)$$

Nous détaillerons dans le chapitre 4, quelques algorithmes de classification utilisés dans le cadre de cette thèse à savoir les k plus proches voisins (k -ppv), les arbres de décision et particulièrement l'algorithme C4.5, les réseaux bayesiens et les machines à vecteurs de support (SVM).

3.5 Conclusion

Dans ce chapitre, nous avons tracé les différentes étapes à suivre pour une caractérisation optimale des textures. Après extraction des différents paramètres de texture par les deux approches proposées de la littérature à savoir les matrices de co-occurrence et les attributs de Tamura, nous procéderons dans une seconde phase à la sélection d'un sous-ensemble pertinent répondant aux contraintes imposées par notre cadre de travail. Plusieurs techniques existent dans la littérature basées sur la notion de classe d'appartenance induisant à une redondance dans l'ensemble des paramètres. Pour pallier à ces inconvénients, nous avons proposé deux autres techniques basées respectivement sur la corrélation de Pearson et la corrélation partielle afin de détecter toute redondance entre paramètres et mesurer ceux les plus pertinents.

L'évaluation du modèle de texture ainsi déduit est une phase primordiale et difficile à mettre en œuvre. Dans ce travail, nous avons proposé deux applications permettant de valider la pertinence du modèle de texture généré en terme de satisfaction des utilisateurs et en mesurant le meilleur taux de classification.

CHAPITRE 4

Applications

Ce chapitre donne une vue plus détaillée sur les différentes applications et techniques utilisées dans ce travail. Le but de cette première partie de la thèse étant de définir et d'évaluer un modèle de texture pertinent répondant à certaines contraintes imposées dans notre cadre de travail, nous avons proposé plusieurs applications permettant d'atteindre ce but.

Le modèle de texture est décrit par les attributs de texture introduits dans le chapitre 3. Ainsi l'évaluation de ce modèle est réalisée dans un premier temps par une mesure de sa performance à satisfaire la requête utilisateur : « *trouver les images visuellement similaires à celle fournie par l'utilisateur* » (cf. section 4.1). Dans le cas où une connaissance de la classe d'appartenance est fournie *a priori*, notre dernière application (cf. section 4.2) vise à tester la capacité du modèle texture à assigner la classe appropriée aux nouvelles images ne figurant pas dans la base d'apprentissage.

4.1 Recherche d'images par la texture

Lorsque l'utilisateur envoie sa requête image au système, ce dernier calcule une distance entre la signature de la requête et celles des images de la base. Ensuite, il ordonne les mesures calculées dans l'ordre croissant; cet ordre reflète le degré de similarité (rapprochement visuel) entre une image de la base et la requête. L'ordre le plus petit correspond à une image visuellement proche de la requête, sinon l'image retournée est visuellement différente de la requête. Par la suite, le système retourne les k premières images estimées être les plus proches visuellement de la requête envoyée. Plusieurs mesures de distance peuvent être utilisées, la plus courante est la distance euclidienne. Néanmoins, cette distance est limitative car elle favorise les attributs ayant des valeurs importantes à ceux ayant une dynamique faible. Une normalisation des attributs est donc nécessaire si on veut bien utiliser cette distance. Dans le chapitre 5, nous donnerons une comparaison entre différentes mesures de distance utilisées.

L'évaluation du système pour la recherche d'images peut s'effectuer d'une manière visuelle, autrement dit, c'est l'utilisateur qui justifie l'intérêt et l'efficacité de la recherche. Cette manière d'évaluation reste insuffisante notamment dans le cas d'une automatisation du processus de la recherche. Dans ce travail, nous présentons une mesure numérique basée sur les scores pour une évaluation plus intéressante.

Pour chaque image I de la base appartenant à une classe donnée C , il existe n^+ images texturées positives correspondant à l'image I et n^- textures négatives (celles qui ne lui correspondent pas visuellement). Des scores peuvent être calculés à partir des exemples positifs et négatifs à la requête afin d'évaluer la performance du système. Le score de chaque image de la base et le score final du système sont donnés respectivement par les formules suivantes :

$$\text{score}_I = \begin{cases} 0 & \text{si } n^+ = n^- \\ e^{\frac{(n^+ - n^-)}{|C|}} & \text{si } n^+ > n^- \\ -e^{\frac{(n^+ - n^-)}{|C|}} & \text{sinon} \end{cases} \quad (4.1)$$

indice image	n^+	n^-	score	s_1	s_2	s_3
I_1	2	2	1			
I_2	3	1	1.6487			
I_3	1	3	0.6065			
I_4	1	3	0.6065	5.7426	1.7426	-2.2574

Table 4.1 – Exemples de scores pour quatre textures

$$\text{score}_{BI} = \sum_{I \in BI} |C| \times \text{score}_I \quad (4.2)$$

Le score est choisi nul dans le cas où le nombre des textures négatives retournées est égal à celui des textures positives, afin de ne pas pénaliser l'évaluation de la performance du système. Supposons par exemple, que pour une base de quatre textures, on trouve les valeurs reportées dans le tableau 4.1.

4.2 Classification des images texturées

La tâche de la classification supervisée¹ consiste à assigner à chaque nouvelle image texture une classe appropriée à partir des connaissances *a priori* fournies. Plusieurs algorithmes existent dans la littérature. Dans ce travail, nous avons choisi quatre parmi eux, à savoir les k plus proches, l'arbre de décision C4.5, les classificateurs bayésiens et les machines à vecteurs de support (SVM), que nous décrivons succinctement et détaillons dans ce qui suit. Les mesures d'évaluation de ces classificateurs seront détaillées dans le chapitre 5.

4.2.1 k -plus proches voisins

La méthode des k plus proches voisins (en anglais *k nearest neighbor*) introduite par Cover et Hart [53] est une méthode de classification géométrique très utilisée, en raison de sa simplicité et de sa robustesse. Les descripteurs sont exploités dans un espace métrique de représentation, généralement \mathcal{R}^n muni d'une distance, particulièrement la distance euclidienne. La phase d'apprentissage consiste à stocker les images étiquetées. L'affectation de nouvelles images s'opèrent en calculant une distance entre les attributs de l'image à classer et ceux de chaque image de la base. Les k images les plus proches sont sélectionnées et l'image requête est assignée à la classe majoritaire (celle qui a le plus grand nombre d'individus).

Cette technique facile à appliquer donne de très bons résultats mais pose un problème pour le choix de la valeur de k qui peut se faire aléatoirement ou bien expérimentalement. Dans notre travail, le choix est évalué expérimentalement en fonction du taux de bonne classification (l'erreur la plus minimale). Une autre principale difficulté consiste à sa dépendance à une fonction de distance pertinente. Nous utiliserons dans ce travail, deux distances à savoir la distance euclidienne et la distance HEOM (Heterogeneous Euclidean-Overlap Metric) [316].

4.2.2 Arbres de décision

Les arbres de décision ou méthodes de segmentation sont parmi les méthodes les plus populaires des méthodes d'apprentissage symbolique utilisés à des fins de classification. Il s'agit plus d'une représenta-

¹Nous noterons la classification supervisée dans tout le manuscrit juste par le terme de « classification ».

tion graphique d'une tâche de classification. Les algorithmes connus sont CART [27], ID3 [238] qui fut amélioré par C4.5 [239]. Les arbres de décision ont connus une large expansion pour la classification des documents [182, 3, 240].

L'analyse d'un ensemble d'individus préalablement classifiés permet de générer une structure arborescente optimale pour la classification des autres individus disponibles. Ceci est fait en partitionnant successivement la population initiale des enregistrements de sorte à isoler au mieux les individus d'une même classe. Ce sont pour la plupart des algorithmes légers, rapides, performants, et la forme arborescente des résultats permet une grande lisibilité et interprétabilité.

Ainsi, un arbre de décision est une structure qui permet de déduire un résultat à partir de décisions successives. Pour parcourir un arbre de décision et trouver une solution il faut partir de la racine. Chaque nœud est une décision atomique (cf. figure 4.1). Chaque réponse possible est prise en compte et permet de se diriger vers un des fils du nœud. De proche en proche, on descend dans l'arbre jusqu'à tomber sur une feuille. La feuille représente la réponse qu'apporte l'arbre au cas que l'on vient de tester.

Très souvent on considère qu'un nœud pose une question sur une variable (la valeur de cette variable pour l'individu présent est supérieure ou inférieure à la valeur de décision?), la valeur de cette variable permet de savoir sur quels fils descendre (gauche ou droit) comme illustré sur la figure 4.1. Pour cet exemple, on mesure le paramètre « Température » pour un individu donné; si la valeur correspondante est supérieur à la « valeur de décision » (ou de découpage), la décision est donc « malade »; dans le cas échéant, cet individu est irrité et ainsi de suite jusqu'à atteindre un nœud feuille correspondant à une décision finale. Pour les variables énumérées il est parfois possible d'avoir un fils par valeur, on peut aussi décider que plusieurs variables différentes mènent au même sous arbre. Pour les variables continues il n'est pas imaginable de créer un nœud qui aurait potentiellement un nombre de fils infini, on doit discrétiser le domaine continu (arrondis, approximation), donc décider de segmenter le domaine en sous ensembles.

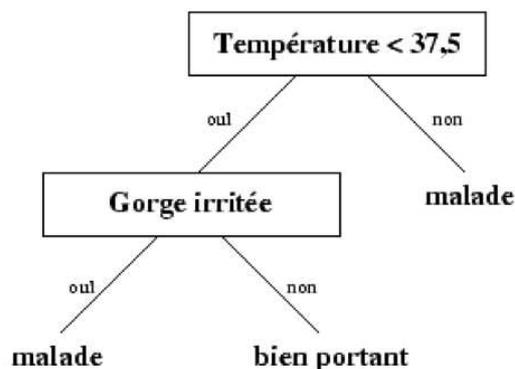


Figure 4.1 – Exemple d'un arbre de décision pour un paramètre de température²

Pour construire un arbre de décision, il faut trouver l'attribut à tester sur chaque nœud. C'est un processus récursif. Pour déterminer quel attribut tester à chaque étape, des calculs statistiques sont utilisés pour déterminer la mesure de cet attribut qui sépare bien les individus (oui/non). On crée alors un nœud contenant ce test, et on crée autant de descendants que de valeurs possibles pour ce test. Par exemple : si on teste la présence d'un mot, les valeurs possibles sont « présent / absent ». A chaque fois, on aura donc deux descendants (fils) pour chaque nœud. Ce processus est répété en associant à chaque descendant le

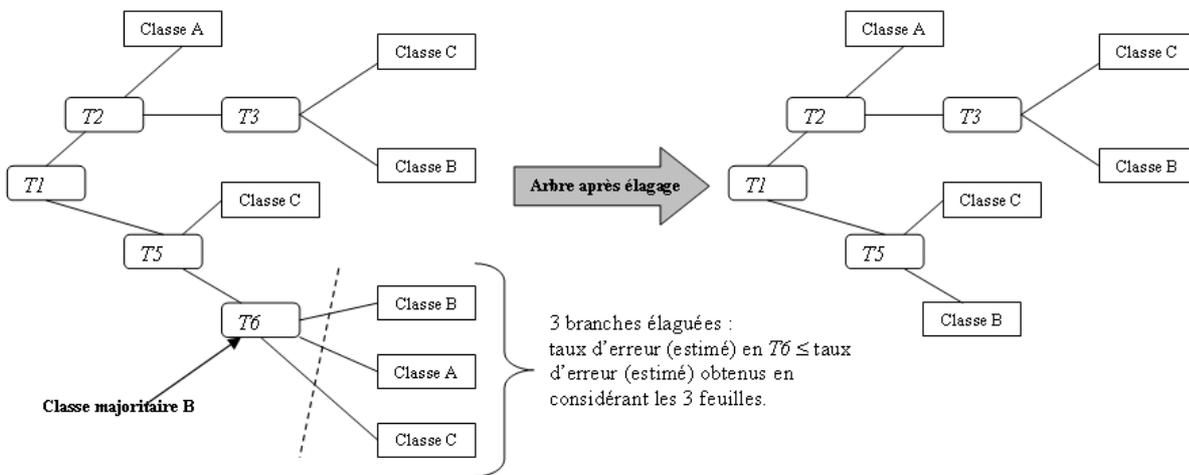


Figure 4.2 – Exemple d'un arbre de décision avant et après élagage⁵

reste des individus qui satisfont le test du prédécesseur. Ainsi, la classification d'un nouveau individu est réalisée en posant une suite de questions relatives à certaines propriétés du cas considéré.

Deux phases nécessaires pour la création d'un arbre de décision : la construction et l'élagage. La construction est la phase la plus importante. Elle consiste en un processus de division récursif sur chaque nœud en produisant dans un premier temps une série de tests relatifs à chacune des variables (qualitatives ou quantitatives) et dans un deuxième temps de sélectionner le meilleur test, i.e. le meilleur partitionnement, d'après un critère d'évaluation de l'hétérogénéité (ou impureté) des classes dont l'objectif est de diminuer le plus possible le mélange des classes au sein de chaque sous-ensemble créé par les différentes alternatives du test. Différentes mesures statistiques sont suggérées pour le sélection à savoir l'entropie et l'information mutuelle (ID3 et C4.5) (cf. section 3.4), l'indice de Gini³(CART) et l'indice d'erreur⁴.

Ensuite, la seconde étape après avoir construit l'arbre est l'élagage (en anglais *pruning*). L'objectif de l'élagage est de supprimer les parties de l'arbre peu représentatives pour garder de bonnes performances de prédiction des nouveaux individus en les remplaçant par un nœud terminal associé à la classe majoritaire. Pour cela, il faudra un critère pour désigner les branches à élaguer. Généralement, ce critère est basé sur une estimation du taux d'erreur de classification; un arbre est élagué à un certain nœud si le taux d'erreur estimé à ce nœud (en y allouant la classe majoritaire) est inférieur au taux d'erreur obtenu en considérant les sous-arbres terminaux. Après élagage, les nouvelles feuilles sont étiquetées par la classe majoritaire (cf. figure 4.2)

Parmi les algorithmes de génération des arbres de décision, nous avons choisi l'algorithme C4.5 [239]. Cet algorithme est une extension de l'algorithme ID3 proposé par Quinlan [238] reposant sur le même algorithme donné dans algorithme 5.

Ainsi, l'algorithme ID3 permet de créer un arbre de décision simple et interprétable facilement pour

³L'indice de Gini sur le nœud n est calculé par :

$$\text{Gini}(n) = 1 - \sum_{\forall j} P^2(\omega_j)$$

avec $P(\omega_j)$ la probabilité *a priori* de la classe ω_j .

⁴L'indice d'erreur mesure la probabilité d'erreur de classification sur le nœud n par rapport à la classe majoritaire donné par $\text{Er}(n) = 1 - \max_j P(\omega_j)$.

Algorithme 5 ID3 (C4.5)

Entrées :

- X : un ensemble d'attributs non cibles ;
- Y : un attribut cible ;
- B : la base d'apprentissage.

Sortie : un arbre de décision.

1. **si** B est vide **alors**
 2. **retourner** un simple nœud avec la valeur « Echec »
 3. **sinon si** B est constituée uniquement de valeurs identiques pour la cible **alors**
 4. **retourner** un simple nœud de cette valeur
 5. **sinon si** X est vide **alors**
 6. **retourner** un simple nœud ayant comme valeur celle la plus fréquente des valeurs de l'attribut cible trouvées dans B
 7. **sinon**
 8. $D \leftarrow$ l'attribut qui a le plus grand $gain(D, B)$ parmi tous les attributs de X
 9. $\{d_j | j = 1, \dots, m\} \leftarrow$ les valeurs des attributs de D
 10. $\{B_j | j = 1, \dots, m\} \leftarrow$ les sous-ensembles de B constitués respectivement des données de valeur d_j pour l'attribut D
 11. **retourner** un arbre dont la racine est D et les arcs sont étiquetés par $d_j, j = 1, \dots, m$ et allant vers les sous arbres $ID3(X_D, Y, B_j), i = 1, \dots, m$
 12. **fin si**
-

des opérations de classification supervisée. Néanmoins, cet algorithme pose quelques problèmes surtout lorsque les données sont incomplètes et dans les cas où les attributs sont continus, ce qui est le cas réellement de plusieurs attributs, et enfin l'arbre produit peut comporter des sous arbres dans lesquels on ne va presque jamais. L'algorithme C4.5 [239] est donc proposé afin d'apporter des solutions à ces limitations.

Attributs de valeurs manquantes Lors de la construction d'un arbre de décision, tenir compte des données pour lesquelles certains attributs ont une valeur inconnue peut s'effectuer en évaluant le gain⁶ où le gain ratio⁷ pour un tel attribut en considérant seulement les données pour lesquelles cet attribut est défini.

La classification de nouvelles données qui ont des valeurs inconnues peut se faire en estimant la probabilité des différents résultats possibles.

Attributs continus Avec l'algorithme C4.5 la gestion des attributs continus est faisable. Soit D l'attribut a un intervalle continu de valeurs. L'algorithme procède par examiner les valeurs de cet attribut sur la base d'apprentissage en ordonnant ces valeurs d'une manière croissante d_1, \dots, d_n . Ensuite pour chacune de ces valeurs, on partitionne les données entre celles qui ont des valeurs de D inférieures ou égales à d_i et celles qui ont des valeurs supérieures. Pour chacune de ces partitions on calcule le gain, ou le gain ratio et on choisit la partition qui maximise le gain. La valeur d_i est dite valeur de découpage (segmentation) pour l'attribut D . On notera quand même que cette méthode nécessite un nombre conséquent d'opérations.

Réduction de l'arbre de décision L'arbre de décision crée en utilisant l'ensemble d'apprentissage traite correctement la plupart des données de l'apprentissage. L'élagage de l'arbre de décision, en est une solution, s'effectue en remplaçant un sous arbre entier par une feuille (classe finale). Cette substitution a lieu si une règle de décision établit que le taux d'erreur attendu dans le sous arbre est supérieur que celui d'une simple feuille.

Pour conclure avec les arbres de décision, d'autres versions améliorées ont vu le jour, les efforts concernent notamment la vitesse de calcul, la quantité de mémoire utilisée, et la taille des arbres générés. Pourtant, nous utiliserons que l'algorithme C4.5 qui a montré une grande performance et efficacité pour divers applications de prédiction et de classification.

4.2.3 Réseaux bayésiens

Un réseau bayésien est un formalisme probabiliste introduit par [144, 285]. Une définition informelle est : *un réseau bayésien est un graphe dans lequel les nœuds représentent des variables aléatoires, et les arcs représentent des relations entre variables qui sont soit déterministes, soit probabilistes. Le graphe est dit acyclique : il ne contient pas de boucle.*

Formellement, $\mathcal{B} = (\mathcal{G}, \Theta)$ est un réseau bayésien (cf. figure 4.3) si \mathcal{G} est un graphe acyclique orienté (en anglais DAG pour *Directed Acyclic Graph*) dont les sommets représentent un ensemble de variables

⁶Le gain de deux variables X et Y exprime le gain d'information après que la valeur de l'attribut Y ait été obtenue : $gain(Y, X) = H(X) - H(Y, X)$ où H est la mesure d'entropie.

⁷Le gain ration récompense le cas où les attributs de fortes valeurs sont favorisés avec $gain : gainratio(X, Y) = \frac{gain(X, Y)}{splitH(X, Y)}$ avec $splitH(X, Y) = H(\frac{|X_1|}{X}, \dots, \frac{|X_n|}{X})$. Les $\{\frac{|X_i|}{X}\}$ sont les partitions de X induites par la valeur de Y .

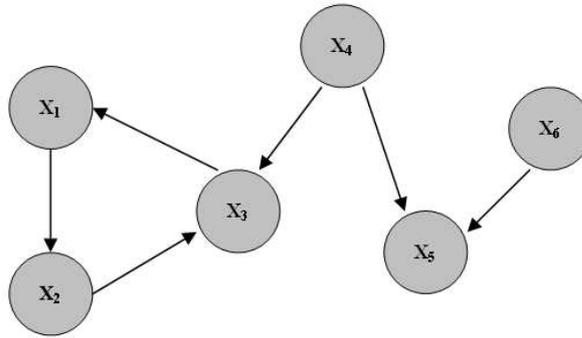


Figure 4.3 – Exemple d'un réseau bayésien [171]

aléatoires $X = \{X_1, \dots, X_N\}$ et chaque $\Theta_i = \mathbb{P}(X_i | X_{\text{parents}(X_i)})$ constitue une matrice de probabilités conditionnelles de la variable X_i ayant une connaissance de l'état de ses parents $\text{parents}(X_i)$ dans \mathcal{G} . Ainsi, Un réseau bayésien \mathcal{B} représente donc une distribution de probabilité sur X qui admet la loi jointe suivante [226] :

$$\mathbb{P}(X_1, \dots, X_N) = \prod_{i=1}^N \Theta_i \quad (4.3)$$

Prenons l'exemple d'un simple réseau bayésien présenté sur la figure 4.3. La probabilité jointe dans ce cas sera :

$$\mathbb{P}(X_1, \dots, X_6) = \mathbb{P}(X_1 | X_3) \mathbb{P}(X_2 | X_1, X_3) \mathbb{P}(X_3 | X_4) \mathbb{P}(X_4) \mathbb{P}(X_5 | X_4, X_6) \mathbb{P}(X_6) \quad (4.4)$$

Cette décomposition de la loi jointe permet d'avoir des algorithmes d'inférence puissants qui font des réseaux bayésiens des outils de modélisation et de raisonnement très pratiques lorsque les situations sont incertaines ou les données incomplètes. Ils sont alors utiles pour les problèmes de classification dans le cas où les interactions entre différents descripteurs peuvent être modélisés par des relations de probabilités conditionnelles.

Lorsque la structure du réseau bayésien n'est pas fournie a priori par un expert, il est possible d'en faire l'apprentissage à partir d'une base de données créée à l'avance. La recherche d'une structure d'un réseau bayésien n'est pas une tâche simple, surtout lorsque la taille de l'espace de recherche s'accroît exponentiellement avec le nombre de variables [245]. Robinson a montré que le nombre de différentes structures de graphes pour réseau bayésien possédant p nœuds est donné par la formule de récurrence de l'équation 4.5 :

$$u_p = \sum_{i=1}^p (-1)^{i+1} \binom{p}{i} 2^{i(p-1)} u_{p-i} \quad (4.5)$$

Les réseaux bayésiens sont utilisés dans divers domaines tels que la classification qui nous intéresse. Le classifieur bayésien ainsi apprend les probabilités conditionnelles de chaque descripteur X_i sachant la classe C à partir d'une base d'apprentissage existante. La classification d'une nouvelle image est faite en appliquant la règle de Bayes pour calculer la probabilité de C sachant les instances des X_1, \dots, X_N . Enfin, prédire la classe d'appartenance C qui est la valeur maximale de la probabilité *a posteriori*.

Dans le cas de la classification, une variable précise correspond à la classe qu'il faut prédire à partir des autres attributs. Plusieurs méthodes d'apprentissage existent proposant des structures pour ce nœud central [43, 44].

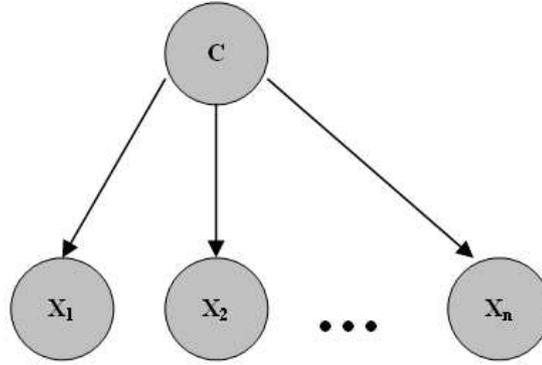


Figure 4.4 – Structure d'un classifieur bayésien naïf

Le classifieur bayésien naïf correspond à la structure la plus simple qui soit, en posant l'hypothèse que les attributs X_1, \dots, X_N sont indépendants conditionnellement⁸ à la classe C . Un exemple simplifié d'un réseau bayésien est donné à la figure 4.4. Cette structure naïve pourtant simple donne de très bons résultats pour divers applications [196, 150].

L'indépendance conditionnelle n'est pas toujours facile à estimer, pour cela, il a été suggéré d'augmenter la structure du réseau bayésien naïf en ajoutant des liens entre les paramètres [68]. Parmi les méthodes avancées pour augmenter la performance de ce réseau, nous citons à titre d'exemple l'arbre-NB (*naive Bayes tree* ou NB-tree) [161] qui combine les arbres de décision et le classifieur bayésien naïf. D'autres existent évidemment tels que le TAN (*Tree Augmented Naive Bayes*) [86] mais nous nous limitons aux arbres-NB.

Pour une application donnée faisant appel aux réseaux bayésiens, deux problèmes se posent. Le premier concerne l'apprentissage des paramètres ou la recherche des distributions de probabilité à partir d'une base de données disponibles [121]. Le second problème réside dans l'apprentissage de la structure du réseau bayésien qui représente le mieux l'application traitée [271]. Ces deux problèmes seront examinés dans ce qui suit.

Le problème de la classification est d'assigner la vraie classe à un ensemble de paramètres. L'erreur de classification (ou erreur de prédiction) est la mesure de qualité (ou évaluation) de la classification la plus courante. Soit $\Omega_{app} = \{X, C\}$ la base d'apprentissage de M individus, $\Omega_{ts} = \{F_j | j = 1, \dots, K\}$ l'ensemble de test où $X = \{X_1, \dots, X_M\}$ et $C = \{c_1, \dots, c_M\}$ sont respectivement l'ensemble des attributs (dont chaque valeur est de la forme $X_i = \{X_i^1, \dots, X_i^N\}$) et l'ensemble des classes. Ainsi, l'erreur de classification sera exprimée comme suit :

$$Err_m = \frac{1}{K} \sum_{k=1}^K (\tilde{c}_k - c_k) \quad (4.6)$$

où \tilde{c}_k et c_k représentent la classe prédite et la vraie classe.

Une bonne classification signifie un taux d'erreur minimal. Ainsi, l'opération de minimisation de l'erreur de classification revient à maximiser la probabilité *a posteriori* de la variable classe $\mathbb{P}(c' | X', X, C)$. Elle correspond à la moyenne sur toutes les structures possibles de la probabilité *a posteriori* :

$$\mathbb{P}(c' | X', X, C) = \frac{\sum_{S^h} \int_{\Theta_S} \mathbb{P}(c' | X', S^h, \Theta_S) \mathbb{P}(S^h, \Theta_S | X, C) d\Theta_S}{\sum_{S^h} \mathbb{P}(S^h | X, C) \mathbf{E}_{\Theta_S | S^h, X, C}(\mathbb{P}(c' | X', S^h, \Theta_S))} \quad (4.7)$$

⁸X et Y sont dits indépendants conditionnellement si $\mathbb{P}(X|Y) = \mathbb{P}(X)$

avec :

$$\mathbb{P}(S^h|X, C) = \int_{\Theta_S} \mathbb{P}(c|X, S^h, \Theta_S) \mathbb{P}(S^h, \Theta_S) d\Theta_S \quad (4.8)$$

et :

$$\mathbf{E}_{\Theta_S|S^h, X, C}(\mathbb{P}(c'|X', S^h, \Theta_S)) = \mathbb{P}(c'|X', S^h, \Theta_S) \mathbb{P}(\Theta_S|S^h, X, C) \quad (4.9)$$

L'équation 4.7 donne la distribution marginale de la classe c' sachant les valeurs des attributs $X' = (X'_1, \dots, X'_N)$ et la base d'apprentissage Ω_{app} . L'utilisation directe de cette équation est difficile vue le nombre de structures à considérer. La solution est de trouver une méthode pour rechercher des structures les plus appropriées pour approcher cette équation.

4.2.3.1 Apprentissage des paramètres

L'estimation des distributions de probabilité à partir des données est un sujet vaste et complexe. Ici, on se limite à présenter les méthodes les plus utilisés dans le cas où les données sont complètes et la structure du réseau bayésien est connue [121, 148, 164].

La méthode simple la plus répandue est l'estimation statistique qui consiste à estimer les probabilités d'un événement par sa fréquence d'apparition dans la base de données. Dans ce cas, la probabilité \mathbb{P} devient :

$$\tilde{\mathbb{P}}(X_i = X_i^n | \mathbf{parents}_{X_i} = X_i^p) = \frac{N_{i,p,n}}{\sum_{\forall n} N_{i,p,n}} \quad (4.10)$$

où $N_{i,p,n}$ est le nombre d'événements dans la base de données pour lesquels la variable X_i soit dans l'état X_i^n et ses parents soient dans la configuration X_i^p .

Une autre approche pour l'apprentissage des paramètres consiste à utiliser une estimation bayésienne. Il s'agit de trouver les paramètres Θ les plus probables en utilisant les probabilités *a priori* des paramètres. L'approche de maximum *a posteriori* (MAP) donne dans ce cas :

$$\tilde{\mathbb{P}}(X_i = X_i^n | \mathbf{parents}_{X_i} = X_i^p) = \frac{N_{i,p,n} + \alpha_{i,p,n} - 1}{\sum_n (N_{i,p,n} + \alpha_{i,p,n} - 1)} \quad (4.11)$$

où $\alpha_{i,p,n}$ représentent les paramètres de la distribution de Dirichlet associés à la loi *a priori* $\mathbb{P}(X_i = X_i^n | \mathbf{parents}_{X_i})$.

Une dernière approche qu'on présente ici consiste à calculer l'espérance *a priori* des paramètres $\Theta_{i,p,n}$ au lieu de chercher le maximum. Ainsi, l'espérance *a priori* (EAP) [244] est :

$$\tilde{\mathbb{P}}(X_i = X_i^n | \mathbf{parents}_{X_i} = X_i^p) = \frac{N_{i,p,n} + \alpha_{i,p,n}}{\sum_n (N_{i,p,n} + \alpha_{i,p,n})} \quad (4.12)$$

4.2.3.2 Apprentissage des structures

Les probabilités données dans la section précédente supposent que la structure du réseau bayésien est connue. La question qui se pose donc est comment trouver la structure du réseau qui répond au mieux au problème de la classification.

Le réseau bayésien naïf se voit le plus simple, le plus répandu pour les problèmes de classification de tous types de données, efficace en temps de calcul et robuste vis-à-vis le bruit et les attributs non-pertinents. Le problème majeur qui en réside est qu'il suppose l'indépendance entre paramètres, ce qui n'est pas vérifié dans le cas réel.

Pour alléger et tenir compte de la dépendance entre paramètres, Kohavi [161] a proposé une méthode hybride de construction de réseau bayésien nommée l'arbre-NB. Cette approche hybride exploite les avantages des arbres de décision en terme de segmentation de la base de données et ceux des classifieurs bayésiens naïfs en tenant en compte que les attributs pertinents.

Méthodes fondées sur le test d'indépendance La première méthode d'apprentissage de structure est fondée sur le test d'indépendance. Différents algorithmes existent : l'algorithme PC [281] (pour Peter et Clark, les prénoms des inventeurs de la méthode) de recherche de causalité, l'algorithme IC [228, 227] (pour *Inductive Causation*) et les algorithmes BN-PC-A et BN-PC-B [42, 45] (pour *Bayes Net Power Constructor*). Ces algorithmes fonctionnent correctement si les tests d'indépendance sont totalement fiables; néanmoins, ils ne sont pas tous appropriés à la classification. Seuls les algorithmes BN-CP sont plus applicables à la classification étant donné qu'ils maintiennent un nombre petit d'arcs dans les réseaux bayésiens.

Méthodes gloutonnes d'optimisation de score Au lieu d'effectuer des tests d'indépendances, cette famille de méthodes cherche la structure qui maximise un score. La vraisemblance des données d'apprentissage est la clé de la plupart des méthodes d'apprentissage de structure basées sur un score. Pour éviter le sur-apprentissage du modèles aux données, la vraisemblance est compensée par un terme de pénalité de la complexité, telle que la longueur minimum de description (MDL, en anglais *Minimum Description Length*), le critère bayésien de l'information (BIC, en anglais *Bayesian Information Criterion*) ou autres. Plusieurs algorithmes existent : les algorithmes génétiques [174], la méthode MWST [49], l'algorithme GS (Greedy Search) [212], etc. Une comparaison détaillée des différentes approches est donnée dans [301].

Méthodes stochastiques d'optimisation pour la classification Dans un cadre général, les deux approches tentent de trouver la structure correcte du réseau bayésien, une tâche qui peut échouer vu l'insuffisance des données d'apprentissage pour avoir des tests d'indépendance fiables ou pour avoir bon score.

Dans ce travail, nous nous intéressons à la classification, autrement dit, on cherche la structure qui donne des meilleurs taux de classification. Pour cela, il est plus logique de concevoir des algorithmes qui utilisent le taux d'erreur de classification comme un critère de recherche de la meilleure structure [50, 262]. Dans cas de figure, nous présentons dans ce qui suit, deux approches pour la classification à savoir le classifieur de Bayes naïf et une de ses variantes à base d'arbre.

4.2.3.3 Classifieur de Bayes naïf

Le classifieur de Bayes naïf (NB pour *naive Bayes*), illustré sur la figure 4.4, est un classifieur probabiliste où les paramètres sont supposés indépendants entre eux. Malgré cette condition qui n'est pas toujours vérifiée dans un cas général, NB reconnaît un succès pour plusieurs problèmes de classification. La raison de ce succès est le nombre réduit de paramètres requis pour son apprentissage. En 2001, Garg et Roth [91] ont montré en utilisant des arguments théoriques fondés sur l'information mutuelle, des raisons additionnelles pour le succès de NB.

Supposons que les paramètres de Ω_{app} sont supposés indépendants sachant la classe c , la classification d'un individu $X_0 = \{X_0^i, i = 1, \dots, N\}$ est donnée par l'équation 4.13. Par la suite, il faut estimer la probabilité $\mathbb{P}(X_0^n | c, \Theta)$ du paramètre X_0^i sachant la classe c . Dans la pratique, il est supposé que les

distributions suivent une loi gaussienne et donc le maximum de vraisemblance peut être utilisé pour estimer les paramètres (la moyenne et la variance). Cependant, cette condition n'est pas toujours vraie. Une solution alternative utilisant la distribution de la Cauchy a été proposée par Sebe et Lew [263] mais qui connaît aussi une difficulté dans l'estimation des paramètres de cette distribution.

$$\tilde{c} = \operatorname{argmax}_c \prod_{n=1}^N \mathbb{P}(X_0^n | c, \Theta) \quad (4.13)$$

où Θ est la structure du réseau de Bayes naïf.

Le classifieur NB donne de bonnes performances malgré :

- (i) le nombre insuffisant de données pour son apprentissage ;
- (ii) l'hypothèse d'indépendance qui n'est pas vérifiée dans certains cas.

Dans le cas où le nombre de données est assez important pour l'apprentissage, NB dégrade de performance, pour cela, il est fortement souhaitable de tenir compte de la dépendance entre les paramètres. Cela pourra améliorer considérablement la qualité de la classification.

4.2.3.4 Classifieur par arbre-NB

Kohavi [161] a proposé le modèle de l'arbre-NB afin d'améliorer les performances de classification dans le cas de grandes bases de données. Dans certaines applications, le classifieur par arbre-NB a montré une grande performance par rapport à l'algorithme C4.5 et NB [161, 315].

Le classifieur par arbre-NB est une structure hybride des arbres de décision sur les nœuds et les branches avec des classifieurs bayésiens naïfs sur les feuilles.

La méthode de construction d'un arbre-NB est donnée par l'algorithme 6. L'algorithme de l'arbre-NB est similaire aux algorithmes de partitionnement récursif sauf que les feuilles de l'arbre créées représentent des structures bayésiennes naïves au lieu que ça soient des nœuds de prédiction d'une seule classe. Sur chaque nœud de l'arbre ayant un ensemble d'instances, l'algorithme évalue l'utilité, (utility) (u), du partitionnement pour chaque attribut. Si la valeur maximale de l'utilité u sur tous les attributs est nettement supérieure de l'utilité de nœud courant, les instances de ce nœud sont partitionnées selon cet attribut. Dans le cas où les attributs sont continus, le seuil du partitionnement utilise l'entropie minimale. Dans le cas discret, les attributs sont découpés selon leurs valeurs. Le partitionnement s'arrête lorsque l'utilité obtenue ne s'améliore plus; dans ce cas, le classifieur NB est créé sur le nœud courant. L'utilité d'un nœud est calculée en discrétisant les données et en utilisant la validation croisée à cinq blocs pour estimer le taux de classification de NB. L'utilité par la suite du partitionnement est la somme pondérée de toutes les utilités des nœuds. Les poids sont proportionnels au nombre d'instances dans chaque nœud. Pour éviter les petits découpages, le partitionnement est considéré *significatif* si l'erreur relative (on ne considère pas l'erreur absolue) de réduction est supérieure de 5% et qu'il existe au moins 30 instances dans le nœud.

La complexité de sélection des attributs dans le cas discret est de $O(M.N^2.K)$, où M est le nombre des instances, N est le nombre des attributs et K le nombre des classes. Dans la pratique, le nombre des attributs est inférieur à $O(\log M)$ et aussi le nombre de classes K est petit; dans ce cas le temps de sélection des attributs en utilisant la validation croisée est très inférieur au temps considéré par le tri des instances pour chaque attribut. Avec cette propriété, l'arbre-NB promet un passage à l'échelle pour de grandes bases de données.

Algorithme 6 arbre-NB

Entrées : un ensemble T de données étiquetées.

Sorties : arbre de décision avec classifieurs bayésiens naïfs sur les feuilles.

1. **pour** $i = 1, \dots, N$ **faire**
2. évaluer l'utilité $\Pi(X_i)$. Pour les attributs continus un seuil de découpage est calculé.
3. **fin pour**
4. $j \leftarrow \operatorname{argmax}_i \Pi(X_i)$ // Trouver l'attribut d'utilité maximale
5. Créer un classifieur de Bayes naïf
6. **si** $\Pi(X_i)$ est non significatif alors
7. Arrêt
8. **fin si**
9. Partitionner T selon le test sur X_j . Si X_j est continu, le seuil de découpage (étape 1) est utilisé; si l'attribut est discret, un multi-découpage est effectué
10. Pour chaque nœud fils, répéter l'algorithme récursivement sur la proportion de T qui a conduit à ce fils.

4.2.4 Machines à vecteurs de supports

Les machines à vecteurs de supports (SVM) sont issues récemment d'une formulation de la théorie de l'apprentissage statistique due en grande partie à l'ouvrage de Vapnik en 1995 intitulé *the nature of learning statistical theory* [305]. Les SVM ont pour but de surmonter le problème de traitement de données de très grandes dimensions. Sans passage par une sélection d'une partie des attributs des données (dans la plupart des techniques d'apprentissage) pour réduire la dimension de l'espace d'entrée, les SVM imposent que le nombre des paramètres soit linéairement lié au nombre des données d'apprentissage. Bien que récemment proposées, les SVM ont fait l'objet d'un nombre important de publications scientifiques.

Les SVM, conçues au départ pour une séparation des données en deux classes, constituent un modèle discriminant qui tente de minimiser les erreurs d'apprentissage tout en maximisant la distance entre les deux classes. Il s'agit en fait de trouver un classifieur linéaire dit hyperplan qui va séparer au mieux les données (cf. figure 4.5). Les points les plus proches (points entourés dans la figure 4.5) sont appelés les vecteurs de support. Ces seuls points sont utilisés pour déterminer l'hyperplan.

Vu la multitude des hyperplans valides, séparant les deux classes, les SVM ont la propriété particulière de chercher l'hyperplan optimal (H dans la figure 4.5). C'est hyperplan correspond à celui qui passe « au milieu » des points des deux classes. Plus formellement, cela revient donc à chercher l'hyperplan dont la distance minimale aux données d'apprentissage est maximale. Autrement dit, celui qui se trouve le plus loin des points. La distance entre l'hyperplan et le point lui est proche (vecteur de support) est appelée « marge ». Pourquoi chercher une marge maximale? La réponse est simple: supposons qu'un point n'a pas été décrit parfaitement, une petite variation ne changera pas sa classification si sa distance à l'hyperplan est grande. Dans le cas inverse où la marge est faible, la classification d'un nouvel élément risque d'être erronée. Donc une marge maximale garantit plus de sécurité lorsqu'on veut classer un nouvel élément.

De nombreux travaux ont démontré la supériorité des SVM par rapport aux méthodes discriminantes

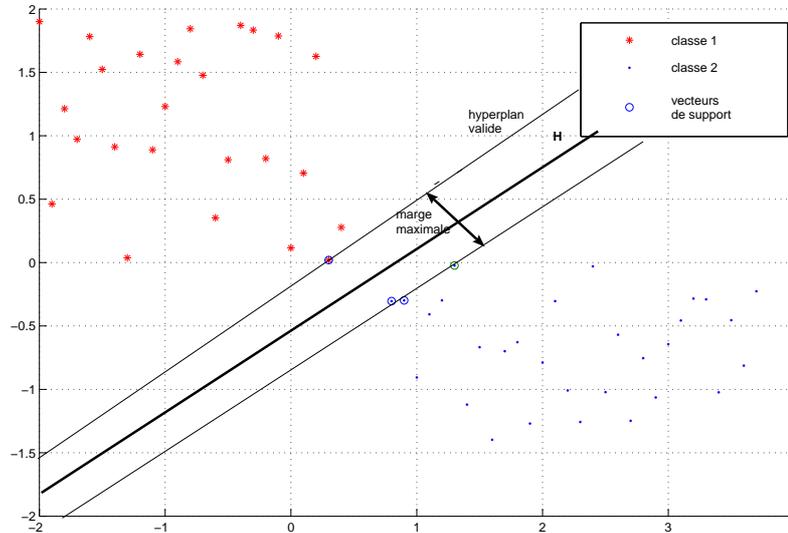


Figure 4.5 – Schéma de l’hyperplan séparant les deux classes

classiques. Leur robustesse vis-à-vis de la dimensionnalité des données et leur pouvoir accru de généralisation, font que les SVM sont nettement plus avantageuses. Le succès de la méthode SVM est justifié par les solides bases théoriques qui la fondent.

Parmi les modèles des SVM, il existe les cas linéairement séparable et les cas non linéairement séparable. Les premiers sont les plus simple des SVM car ils permettent de trouver facilement le classificateur linéaire. Dans la plupart des problèmes réels il n’y a pas de séparation linéaire possible entre les données, le classificateur de marge maximale ne peut pas être utilisé car il fonctionne seulement si les classes de données d’apprentissage sont linéairement séparables. La solution dans ce cas est de détourner le problème en représentant les données dans un autre espace plus grand en terme de dimension puis chercher l’hyperplan linéaire optimal dans cet espace.

4.2.4.1 Cas des données linéairement séparables

Supposons que nous disposons d’une base d’apprentissage $\{x_i | x_i \in \mathbb{R}^p, i = 1, \dots, N\}$ étiquetée dans $y_i \in \{-1, +1\}$. Les points x de l’hyperplan optimal H sont décrits par l’équation $w \cdot x + b = 0$ où w est la normale de l’hyperplan et $\frac{|b|}{|w|}$ est la distance entre l’hyperplan et l’origine (cf. figure 4.6). La marge est définie dans ce cas par $d^+ + d^-$ avec d^+ et d^- correspondent respectivement aux plus proches positif et négatif vecteur de support de l’hyperplan. L’hyperplan optimale est donc celui qui maximise la marge $\frac{2}{|w|}$; cette valeur est obtenue en considérant les équations suivantes :

$$w \cdot x_i + b = 0, \text{ pour les vecteurs de support} \tag{4.14}$$

$$w \cdot x_i + b \geq +1 \quad \text{si } y_i = +1 \tag{4.15}$$

$$w \cdot x_i + b \leq -1 \quad \text{si } y_i = -1 \tag{4.16}$$

résumées par :

$$y_i(w \cdot x_i + b) - 1 \geq 0, \forall i = 1, \dots, N \tag{4.17}$$

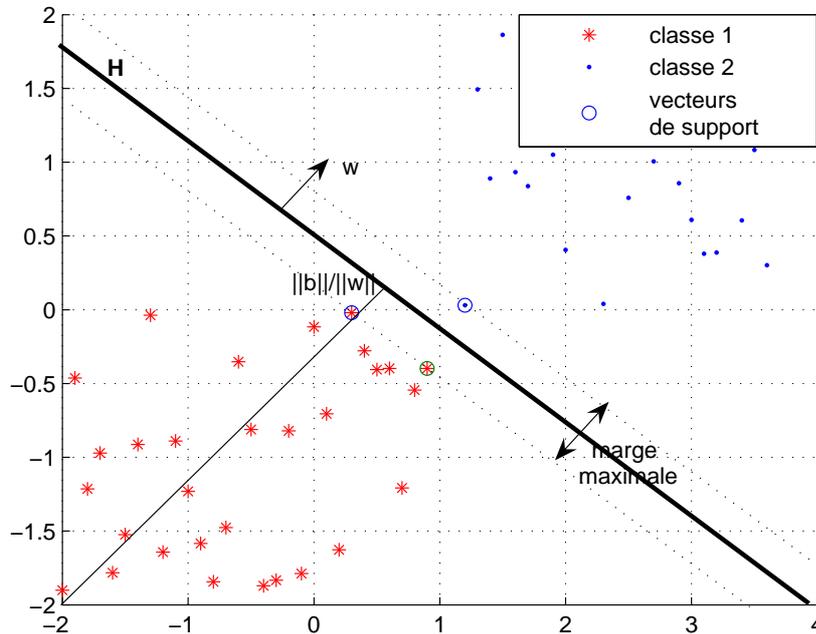


Figure 4.6 – Schéma d'hyperplan pour des données linéairement séparables

Les points de l'équation 4.15 sont reliés à l'hyperplan (H1) dont la distance par rapport à l'origine vaut $\frac{|1-b|}{|w|}$. De même pour l'équation 4.16, elle lie les points dont la distance de l'hyperplan à l'origine vaut $\frac{|-1-b|}{|w|}$. Ainsi, $d^+ = d^- = \frac{1}{|w|}$ d'où la marge vaut $\frac{2}{|w|}$.

La maximisation de $\frac{2}{|w|}$ se traduit sous ces contraintes (eq. 4.17) à une minimisation de $J(w) = \frac{|w|^2}{2}$. Il s'agit de chercher une solution du système non-linéaire suivant :

$$\begin{cases} \min_w J(w) = |w|^2/2 \\ \varphi_i(w, b) \leq 0 \end{cases} \quad (4.18)$$

où $\varphi_i(w, b) = 1 - y_i(w.x_i + b), \forall i = 1, \dots, N$.

La résolution de ce problème d'optimisation est donnée par la méthode de Lagrange en introduisant des multiplicateurs de Lagrange α_i . Autrement dit, il faut trouver le point selle du Lagrangien :

$$L_P(w, b, \alpha) = \frac{1}{2}|w|^2 - \sum_{i=1}^N \alpha_i [y_i(w.x_i + b) - 1] \quad (4.19)$$

On trouve comme solution :

$$w^* = \sum_{i=1}^N \alpha_i^* y_i x_i \quad (4.20)$$

où les α_i^* sont des solutions des conditions de Karush-Kuhn-Tucker (KKT) [77] suivantes, très utilisées

dans le domaine de l'optimisation avec contraintes :

$$\begin{cases} \frac{\partial L_P}{\partial w_v} = w_v - \sum_{i=1}^N \alpha_i y_i x_i v = 0, \forall v = 1, \dots, p \\ \frac{\partial L_P}{\partial b} = - \sum_{i=1}^N \alpha_i y_i = 0 \\ y_i(w \cdot x_i + b) - 1 \geq 0, \forall i = 1, \dots, N \\ \alpha_i \geq 0, \forall i = 1, \dots, N \\ \alpha_i [y_i(w \cdot x_i + b) - 1] = 0, \forall i = 1, \dots, N \end{cases} \quad (4.21)$$

La résolution des α_i^* permet par la suite de classer un nouveau individu z . Pour cela, il suffit d'évaluer le signe de la quantité :

$$w^* \cdot x_i + b^* = \sum_{i=1}^N \alpha_i^* y_i \cdot z \cdot x_i + b^* \quad (4.22)$$

On peut remarquer de l'équation 4.21 que les vecteurs de support (points appartenant à l'hyperplan) correspond aux points x_i tels que $\alpha_i > 0$. Ainsi, l'équation 4.22 devient :

$$f_z = \sum_{x_i \in VS} \alpha_i^* y_i \cdot z \cdot x_i + b^* \quad (4.23)$$

où VS est l'ensemble des vecteurs de support.

4.2.4.2 Cas de données non linéairement séparables

Dans le cas où les données ne peuvent pas être séparées par un hyperplan, les contraintes sont donc relâchées et qui deviennent :

$$\begin{cases} w \cdot x_i + b \geq 1 - \xi_i & \text{si } y_i = +1 \\ w \cdot x_i + b \leq -1 + \xi_i & \text{si } y_i = -1 \end{cases} \quad (4.24)$$

ce qui revient à :

$$\begin{cases} y_i(w \cdot x_i + b) \geq 1 - \xi_i \\ \xi_i \geq 0, \forall i = 1, \dots, N \end{cases} \quad (4.25)$$

Le classifieur attribuera une classe fautive à un élément x_i si ξ_i correspondant est supérieur à 1 ; ainsi $\sum_{i=1}^N \xi_i$ représente la borne supérieure du nombre d'erreurs.

Dans ce cas non-linéaire, la fonction J à minimiser devient :

$$J(w, \xi) = \frac{1}{2} |w|^2 + C \sum_{i=1}^N \xi_i \quad (4.26)$$

où C est une constante, dénommée « capacité », choisie par l'utilisateur qui permet de donner plus ou moins d'importance aux erreurs. Une grande valeur de C traduit une grande pénalisation des erreurs.

Le Lagrangien dans ce cas est donné par :

$$L_P(w, b, \xi, \alpha, \mu) = J(w, \xi) + \sum_{i=1}^N \alpha_i [1 - \xi_i - y_i(w \cdot x_i + b)] - \sum_{i=1}^N \mu_i \xi_i \quad (4.27)$$

La résolution du problème d'optimisation est similaire que le premier cas, il faut trouver de même une solution α^* aux conditions de KKT. La classification d'un nouvel élément suit la même équation 4.23.

4.2.4.3 Cas SVM non-linéaire

Dans ce cas, les données sont séparables avec une fonction de décision non-linéaire. L'idée pour rechercher l'hyperplan optimale est dans un premier temps changer l'espace de données initiales en un autre espace de dimension plus grande, muni d'un produit scalaire (espace de Hilbert). On parle d'un « espace de re-description \mathcal{H} ». La transformation de l'espace s'effectue à l'aide d'une fonction noyau K tel que :

$$\Phi : \mathbb{R}^p \rightarrow \mathcal{H} \quad (4.28)$$

et :

$$K : \begin{array}{ccc} \mathbb{R}^p \times \mathbb{R}^p & \rightarrow & \mathbb{R} \\ (x, x') & \mapsto & \langle \Phi(x), \Phi(x') \rangle_{\mathcal{H}} \end{array} \quad (4.29)$$

Ensuite, déterminer l'hyperplan qui permet une séparation linéaire sur ce nouveau espace \mathcal{H} en considérant les notations nouvelles $(\Phi(x), y)$ au lieu de (x, y) et en reprenant les formules du cas linéairement séparables.

La classification d'un nouveau individu z est donnée par :

$$f(z) = \sum_{\xi \in VS} \alpha_i^* y_i K(x_i, z) + b^* \quad (4.30)$$

En pratique, quelques familles de fonctions noyau paramétrables sont connues et il revient à l'utilisateur de SVM d'effectuer des tests pour déterminer celle qui convient le mieux pour son application. On peut citer les noyaux les plus utilisés :

- noyau polynomial :

$$K(x, y) = (x \cdot y + 1)^d, d \in \mathbb{R} \quad (4.31)$$

- noyau gaussien :

$$K(x, y) = e^{-\frac{|x-y|^2}{2\gamma^2}}, \gamma \in \mathbb{R} \quad (4.32)$$

- noyau sigmoïdal :

$$K(x, y) = \tanh(\gamma x \cdot y - \zeta) \quad (4.33)$$

Parmi les inconvénients du SVM est le choix des paramètres des fonctions noyaux ainsi la constante C . Cela peut être résolu par une validation croisée. John Platt [233] propose un algorithme d'optimisation séquentiel connu par SMO (Sequential Minimal Optimisation). Une version améliorée du SMO est proposée par Keerthi et al. [154]. Dans un travail récent, Laanaya et al. [172] proposent d'utiliser les algorithmes génétiques pour l'automatisation du choix de C et γ .

Nous donnerons une comparaison de performance du SVM et SMO dans le chapitre 5.

4.2.4.4 Classification multi-classe

La méthode SVM, destinée au départ à la classification binaire, peut être étendue au cas multi-classe qui reflète la réalité de plusieurs applications. Dans ce cas, l'optimisation est résolue de différentes manières [127]: en utilisant l'approche « un contre reste », l'approche « un contre un » et la correction d'erreur ECOC (*Error Correcting Output Coding*).

Approche « un contre reste » C'est l'approche la plus simple et la plus ancienne des méthodes de décomposition. Elle considère le problème comme un problème binaire et donc apprend N_c fonctions de décision $\{f_i\}, \forall i = 1, \dots, N_c$ qui permet de discriminer une classe parmi les autres. Pour classer un individu, on le présente donc à tous les classifieurs et la décision s'obtient en appliquant le principe "winners-takes-all" : la classe retenue est celle associée au classifieur ayant renvoyé la valeur la plus élevée.

Si la fonction $f_i(x)$ est maximale alors le nouveau individu x sera affecté à la classe C_i [260] avec :

$$i = \operatorname{argmax}\{f_i(x) | i = 1, \dots, N_c\} \quad (4.34)$$

L'approche « un contre un » Pour cette approche [119], il s'agit de discriminer une classe d'une autre et donc, $N_c \frac{N_c-1}{2}$ fonctions de décision seront apprises et chacune d'entre elles effectue un vote pour l'affectation d'un nouveau individu x . La classe de cet individu x devient ensuite la classe majoritaire après le vote. Cette approche est dite associée en général à une opération de vote majoritaire pour la fusion des décisions des sous-classifieurs pour la classification des nouveaux éléments [85].

L'approche ECOC Cette approche combine les résultats obtenus à partir des classifieurs binaires. La classe d'un individu x est celle qui pour les différents classifieurs binaires donne le moins d'erreurs [63]. Elle est basée sur la matrice de codes $(M_{i,j}, i = 1, \dots, N_c, j = 1, \dots, n_c)$.

La matrice de codes est une matrice de taille $N_c \times n_c$ où N_c est le nombre de classes et en colonne n_c classifieurs. Elle représente la contribution de chaque classifieur au résultat final de la classification en se basant sur le résultat des erreurs commises par les différents classifieurs. La matrice de codes sert à tester les données dans différents cas (base de test connue et inconnue) :

- Dans le cas où la base de test est connue, chaque classifieur S_j donne la distance de chaque point de la base de test par rapport à l'hyperplan du classifieur. Ainsi, la classe d'un individu x est donnée par :

$$\text{classe}(x) = \operatorname{argmin}_i \left(\sum_{j=1}^{n_c} (1 - d_j(x) M_{i,j}) \right) \quad (4.35)$$

où $d_j(x)$ est la distance entre x et l'hyperplan du classifieur S_j .

- Dans le cas où la base de test est inconnue, chaque classifieur S_j prédit la classe d'un individu soit positive (+1) ou négative (-1). Le résultat est donc un vecteur de taille n_c . Ce vecteur est comparé par la suite à chaque ligne de la matrice des codes en utilisant la distance de Hamming. La ligne garantissant la plus petite distance est la classe assignée à l'individu x . Cette technique est dite de *décodage avec la distance de Hamming*.

4.3 Conclusion

Nous avons présenté dans ce chapitre les différentes applications pour l'évaluation du modèle de texture extraits par les techniques citées dans le chapitre 3. La première application concerne la recherche d'images texturées dans une base de d'images. Un paramètre est dit pertinent, dans ce cas, s'il montre une grande capacité à reconnaître les textures de même famille.

La seconde application concerne la classification de nouvelles textures. Un paramètre sera pertinent s'il montre un meilleur de taux de classification. Différentes approches de classification existent dans la littérature. Nous nous sommes restreints à celles les plus répandues et utilisées, à savoir les k -plus

proches voisins, les arbres de décision et particulièrement l'algorithme C4.5, les réseaux bayésiens et les SVM.

Le chapitre suivant dresse les différentes expérimentations menées dans cette thèse.

Expérimentations

L'objectif de ce chapitre est d'expérimenter et d'évaluer les approches d'extraction des descripteurs de texture (MCO et Tamura) en terme de classification et de recherche d'images par contenu particulièrement par la texture dans un cadre général. Pour ce chapitre, nous nous contentons que des bases d'images texturées.

Les expérimentations menées dans cette partie visent à modéliser la texture en proposant un modèle standard basé sur les descripteurs visuels de bas niveau les plus performants et répondants aux contraintes imposées dans le cadre de notre travail.

Nous rappelons brièvement les hypothèses générales conditionnant notre contexte de travail. Les descripteurs du modèle de texture à proposer doivent être :

1. invariants par translation et changement d'intensité ;
2. invariants par rotation sauf pour « la direction » où la préservation de l'orientation des textures est indispensable (les requêtes pouvant être du type : « *Donnez-moi toutes les images orientées horizontalement, verticalement, etc.* ») ;
3. discriminants visuellement et sémantiquement entre différentes catégories de textures ;
4. interprétable en langage naturel (cette transcription en langage humain permettant de traduire *a minima* la sémantique de l'image) ;
5. de taille minimale pour optimiser le temps de calcul lors de l'intégration dans un système de gestion de bases de données standard (SGBD) sans altérer la pertinence des paramètres extraits.

Le paragraphe suivant présente les différentes mesures d'évaluation adoptées pour la recherche d'images et la classification.

5.1 Mesures d'évaluation

Nous devons évaluer les différents modèles de description de textures en terme de la recherche d'images par l'exemple et de la classification.

5.1.1 Recherche par exemple

L'approche classique d'évaluation de la recherche d'images par l'exemple est les critères de rappel-précision détaillés comme suit :

Diagrammes rappel-précision Nous nous limiterons à l'utilisation des critères classiques d'évaluation habituellement utilisées pour la recherche d'information [289, 302] et précisons à la recherche

documentaire [256, 255, 275]. Depuis plusieurs années, les mesures *rappel* (ρ) et *précision* (π) sont largement utilisées pour l'évaluation et pour la comparaison des différents descripteurs appliqués dans les systèmes CBIR [1].

Les paramètres *rappel* et *précision* mesurent la pertinence d'un système de recherche d'information. Autrement dit, ils mesurent la concordance des informations retournées relatives à la requête. L'information dans notre cas est le contenu des images. Les mesures sont calculées sur la base vérité-terrain où nous disposons *a priori* d'une connaissance sur les caractéristiques et l'effectif des différentes classes de la base. Le système effectue une recherche par similarité (qui peut être visuelle et ou sémantique dépendante des objectifs de l'application). Lorsqu'une image I est fournie au système, ce dernier retourne les n plus proches images à cette requête en fonction de la similarité employée que nous notons $R(I)$. L'ensemble des images $R(I)$ contient les images pertinentes et non-pertinentes. Une image est dite pertinente ou correcte lorsqu'elle satisfait la requête utilisateur, dans le cas échéant elle est dite non-pertinente. Nous notons $R^{per}(I)$ l'ensemble des images pertinentes dans $R(I)$ et $\Omega^{per}(I)$ l'ensemble des images pertinentes dans la base de données Ω .

Précision.

La précision est la proportion des images pertinentes parmi la collection des images retournées par le système. Elle représente la capacité du système à ne retrouver que les images pertinentes pour la requête soumise:

$$precision(\pi(I)) = \frac{|R^{per}(I)|}{|R(I)|} = \frac{\text{Nombre d'images pertinentes trouvées}}{\text{Nombre d'images trouvées}}$$

Cette notion est associée au **bruit** qui correspond aux images non-pertinentes retournées par le système:

$$bruit = 1 - precision.$$

valeur entre 0 et 1. Une valeur importante de précision est atteinte lorsque le système fournit de meilleurs résultats.

Rappel.

Le rappel mesure la proportion des images pertinentes retrouvées par le système comparée à la totalité des images pertinentes existantes dans la base Ω . En d'autres termes, il s'agit de mesurer la capacité du système à retrouver toutes les images pertinentes correspondantes aux requêtes fournies.

$$rappel(\rho(I)) = \frac{|R^{per}(I)|}{|\Omega^{per}(I)|} = \frac{\text{Nombre d'images pertinentes trouvées}}{\text{Nombre d'images pertinentes dans la base}}$$

Cette notion est liée au **silence** qui correspond aux images pertinentes non détectées par le système :

$$silence = 1 - rappel.$$

Les deux métriques *rappel* et *précision* s'utilisent conjointement pour l'évaluation des performances des systèmes de recherche d'information et varient inversement: lorsque la précision diminue, le rappel augmente et réciproquement. Les valeurs de ces deux métriques reflètent le point de vue de l'utilisateur:

- si le rappel est faible, une partie de l'information pertinente ne lui sera pas accessible;
- si la précision est faible, l'utilisateur ne sera pas satisfait à cause de la forte concentration des informations non-pertinentes fournies dans les résultats.

Dans les deux cas, le système ne répond pas aux attentes des utilisateurs à retourner l'information utile et pertinente et par la suite, il est non-performant. Le cas idéal est d'avoir la valeur de précision et rappel respectivement égale à un, chose qui n'est pas atteinte en réalité.

Une des manières de tenir compte à la fois du rappel et de la précision d'un système est d'exprimer les valeurs de précision en fonction des différents niveaux de rappel selon la courbe rappel/précision (courbe-*RP*).

Courbe rappel/précision. La courbe rappel/précision permet de suivre la qualité du résultat en fonction du nombre d'images retournées par le système en réponse d'une requête q . D'après la description de Schauble dans [258], les courbes-*RP* se créent en plusieurs étapes: (1)-pour chaque requête q et pour chaque rang r ($r = 1 \dots N$), la précision $\pi_r(q)$ et le rappel $\rho_r(q)$ de la liste triée des r premiers documents résultats sont calculés. Des courbes en dents de scie résultent à cette étape puisque le passage du rang i au rang suivant $i + 1$ correspond à un document rajouté dans la liste résultat. Selon la pertinence du nouveau document ajouté, la précision calculée peut croître ou décroître. Les courbes-*RP* obtenus à ce stade ne reflètent qu'une recherche locale du système qui est fortement sensible à la requête, la base d'images et la réponse système. Pour pallier à cette sensibilité, l'évaluation de la performance globale du système se fera sur la précision moyenne obtenue sur l'ensemble des requêtes pour un rappel donné. Dans le cadre de notre travail, les courbes-*RP* sont toujours évaluées pour 12 valeurs de rappel: $\rho = 0.0, 0.05, 0.1, 0.2, \dots, 1.0$. Toutefois, il est nécessaire de connaître au préalable le résultat attendu des requêtes pour calculer les paramètres π et ρ et évaluer le système de recherche. Malheureusement cette connaissance est souvent fastidieuse à obtenir. Tout d'abord il faut annoter toutes les images de la base et énumérer les requêtes possibles, tâche qui demande un effort considérable lorsque le volume de base des images s'accroît.

5.1.2 Classification

L'évaluation de la classification peut se faire dans un premier temps visuellement en comparant les résultats obtenus avec les images de référence. Cependant, ce moyen devient lourd lorsque la taille de la base augmente considérablement (plus qu'une centaine d'images), de plus il est judicieux de comparer différents algorithmes de classification par des moyens automatiques. La majorité des travaux de classification sont comparés en terme de taux d'erreur ou de taux de bonne classification. Dans ce travail, nous proposons d'utiliser les matrices de confusion ainsi que les différentes mesures qui en découlent.

Une matrice de confusion (MC), aussi connue sous les termes matrice d'erreur, tableau de contingence ou matrice d'erreur de classification, est une matrice carrée qui mesure la répartition des images classées dans les différentes classes par rapport aux classes de référence. Les lignes et les colonnes de cette matrice représentent respectivement les classes vraies (étiquetées manuellement) et les classes prédites (trouvées) par l'algorithme de classification (l'inverse est aussi possible). Donc, chaque élément MC_{ij} de cette matrice correspond au nombre d'images appartenant réellement à la classe C_i qui sont classifiées en classe C'_j (eq 5.1). Pour avoir une interprétation en terme des taux, il suffit de normaliser cette matrice par la somme marginal de chaque ligne qui n'est autre que le nombre total des éléments de la classe C_i (eq. 5.2) :

$$MC_{ij} = \text{card}(C_i \cap C'_j) \quad (5.1)$$

$$MCN_{ij} = \frac{MC_{ij}}{\sum_{k=1}^{N_c} MC_{ik}} \quad (5.2)$$

De cette matrice découle plusieurs mesures statistiques qui permettent d'évaluer la classification en terme d'erreurs et de bonnes classification. Nous utiliserons ici, les mesures suivantes : le taux de classification T_c , la F-mesure F , la valeur Kappa K et la probabilité d'erreur P_e qui sont décrites dans la suite.

Taux de classification est la moyenne des taux de bonnes classification de chaque classe C_i :

$$T_c = \frac{\sum_{i=1}^{N_c} MCN_{ii}}{N_c} \quad (5.3)$$

La F-mesure introduite par Van Rijsbergen's [302] est basée sur l'idée que chaque classe vraie C_i est le résultat connu *a priori* de documents recherchés vis-à-vis d'une requête et que chaque classe prédite C'_j est le résultat retrouvé pour la requête. Ainsi, elle fait appel aux termes de rappel et précision définis respectivement par $REC(i, j)$ et $PREC(i, j)$:

$$REC(i, j) = \frac{card(C_i \cap C'_j)}{card(C_i)}$$

et

$$PREC(i, j) = \frac{card(C_i \cap C'_j)}{card(C'_j)}$$

La matrice de la F-mesure est donnée donc par :

$$F(i, j) = \frac{2 \cdot REC(i, j) \cdot PREC(i, j)}{REC(i, j) + PREC(i, j)} \quad (5.4)$$

La F-mesure globale d'une classification par rapport à la référence est donnée par l'équation 5.5 dont sa valeur appartient à l'intervalle $[0, 1]$:

$$F = \sum_{i=1}^{N_c} \frac{N_{ci}}{N} \cdot max_j \{F(i, j)\}_{j=1, \dots, N_c} \quad (5.5)$$

où N_{ci} est le cardinal de la classe C_i et N le nombre totale des images de la base.

La Mesure Kappa détermine la qualité de la classification en mesurant la concordance entre les jugements de référence et ceux obtenus par les algorithmes de classification. Sa valeur appartient à l'intervalle $[-1, 1]$. Cette valeur est d'autant plus grande que la matrice de confusion MC est diagonale :

$$K = \frac{N \sum_{i=1}^{N_c} MC_{ii} + \sum_{k=1}^{N_c} (\sum_{i=1}^{N_c} MC_{ik} \cdot \sum_{j=1}^{N_c} MC_{kj})}{N^2 - \sum_{k=1}^{N_c} (\sum_{i=1}^{N_c} MC_{ik} \cdot \sum_{j=1}^{N_c} MC_{kj})} \quad (5.6)$$

Landis et Koch [173] ont proposé une échelle de classement du degré d'accord en fonction de la valeur de Kappa présentée dans le tableau suivant :

Accord	Kappa
Excellent	> 0.81
Bon	0.80 - 0.61
Modéré	0.60 - 0.41
Médiocre	0.40 - 0.21
Mauvais	0.20 - 0.0
Très mauvais	< 0.0

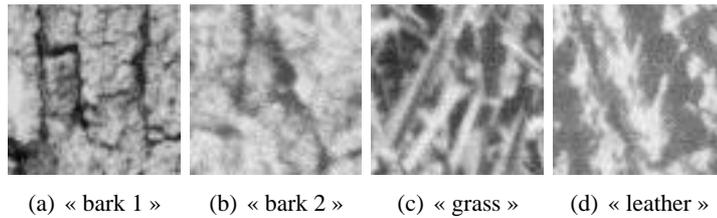


Figure 5.1 – Difficulté de différenciation entre textures extraites à l'échelle 64×64

Erreur de classification Pour mesurer l'erreur de classification, nous utiliserons l'erreur moyenne pour chaque classe (eq. 5.7) ainsi que l'erreur totale sur la base entière (eq. 5.8). Les équations sont données comme suit :

$$E_i = \frac{\sum_{j \neq i}^{N_c} MC_{ij}}{N_{ci}} \quad (5.7)$$

$$E_t = \frac{\sum_{i=1}^{N_c} E_i}{N_c} \quad (5.8)$$

5.2 Extraction des attributs de texture

Dans ce travail, plusieurs facteurs entre en jeu conditionnant la pertinence du modèle de texture extrait à savoir: la taille des textures à étudier (notion de taille d'analyse introduit au début dans le chapitre 3), le choix de la distance d dans le cas des matrices de co-occurrences et la mesure de similarité adéquate pour la recherche d'images similaires.

5.2.1 Choix de la taille des textures de la base

La majorité des auteurs des travaux d'analyse, de classification, de segmentation et de traitement de textures adoptent un découpage aléatoire de la base ; autrement dit, chaque image de la base est décomposée en k régions de taille $n_k \times n_k$. Le paramètre k dépend fortement de la base de texture utilisée et l'application visée. Ainsi, le choix de la taille des textures est un paramètre primordial influençant les résultats. Une petite taille (ex. : 64×64) entraîne une difficulté de l'expert ou de l'utilisateur à différencier entre textures d'une même classe ou de classes qui diffèrent.

Cela est illustré sur la figure 5.1 : les textures 5.1(a) et 5.1(b) appartiennent à la même classe « bark » alors que visuellement, on jugera qu'elles sont différentes ; *a contrario* pour 5.1(c) et 5.1(d), les textures appartiennent respectivement à la classe « grass » et « leather » et apparaissent similaires visuellement.

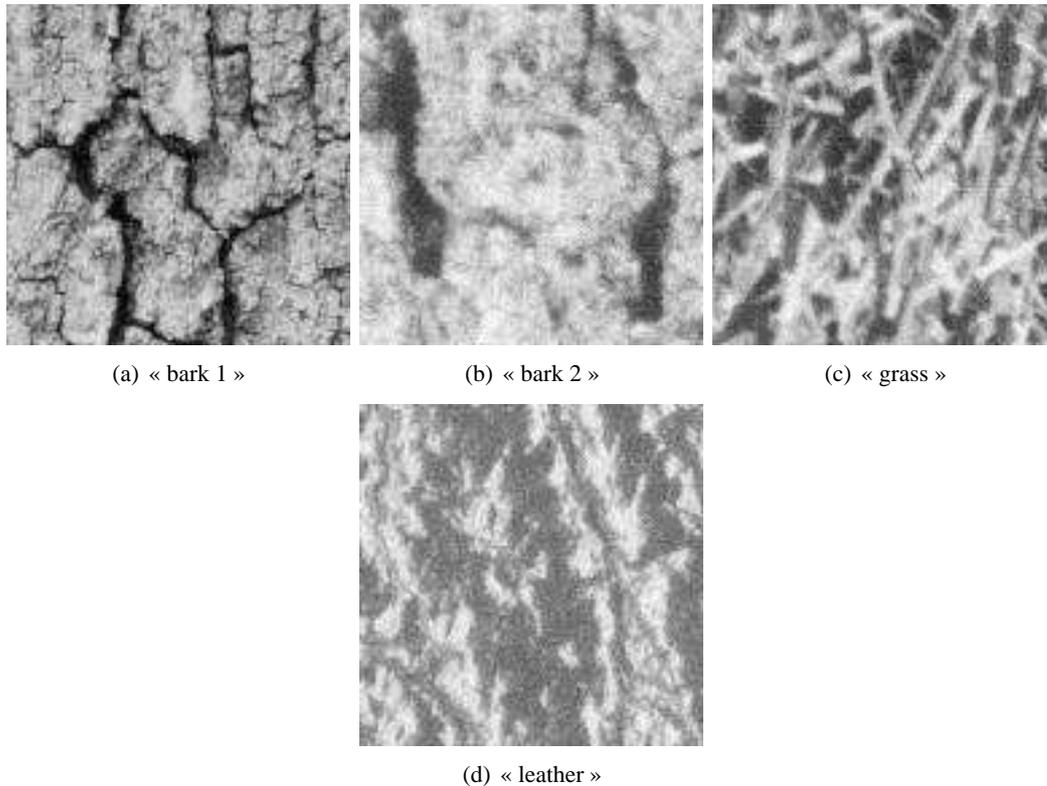


Figure 5.2 – Primitives de textures davantage visibles à l'échelle 128×128

Pour les mêmes textures mais de taille 128×128 (cf. figure 5.2), il y a plus de visibilité sur les primitives de la texture, néanmoins les mêmes remarques que précédemment s'appliquent encore.

Il faut des tailles plus grandes (cf. figures 5.3 et 5.4 pour 256×256) pour constater visuellement une nette différence entre les différentes textures.

Reste maintenant à connaître l'influence de la taille sur les performances d'une reconnaissance et classification automatique. Le but ici est donc de trouver un compromis entre une taille acceptable qui permet à l'expert et ou utilisateur une bonne visibilité afin de pouvoir différencier visuellement entre les textures et de bonnes performances en terme de classification et de recherche d'images similaires. Pour cela, nous testons 3 tailles : 128×128 , 256×256 et 512×512 .

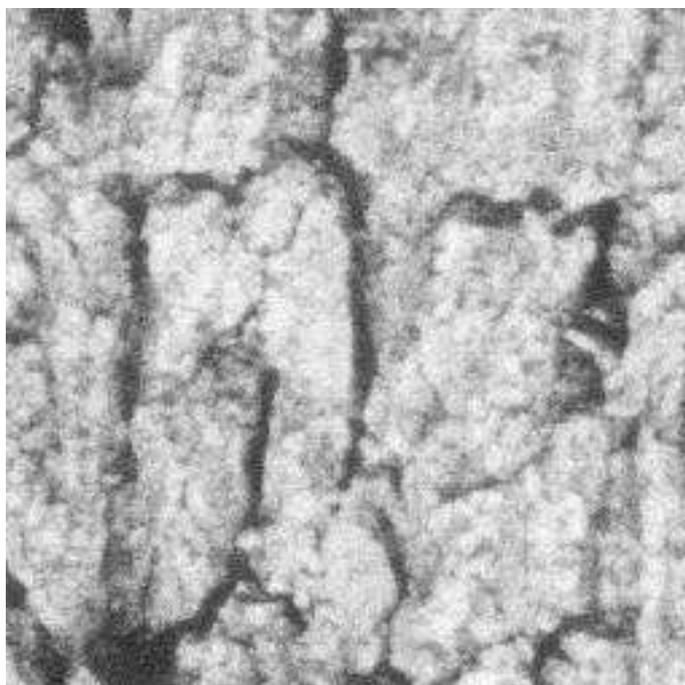
La figure 5.5 trace les taux de classification obtenus pour différents classifieurs comparés au classifieur de référence SVM appliqués à la base Brodatz1 et pour les attributs extraits des matrices de co-occurrences. Nous constatons que pour ces différents algorithmes, la taille correspondant à 256×256 pixels fournit un meilleur taux de classification. De même, dans le cas de la recherche d'images par contenu, avec les courbes de rappel et précision, nous obtiendrons la même conclusion (cf. figure 5.6).

5.2.2 Descripteurs des matrices de co-occurrences

Rappelons que l'élément $p(i, j)$ de la matrice de co-occurrences (MCO) exprime la probabilité d'apparition des niveaux de gris i et j séparés par un pas d suivant une orientation θ . Ainsi, les paramètres d et θ constituent la clé des MCO. Dans ce travail, les huit paramètres introduits dans la section 3.3.2.1, à savoir: l'énergie, l'entropie, la variance, l'uniformité, le contraste, la corrélation, l'homogénéité et la

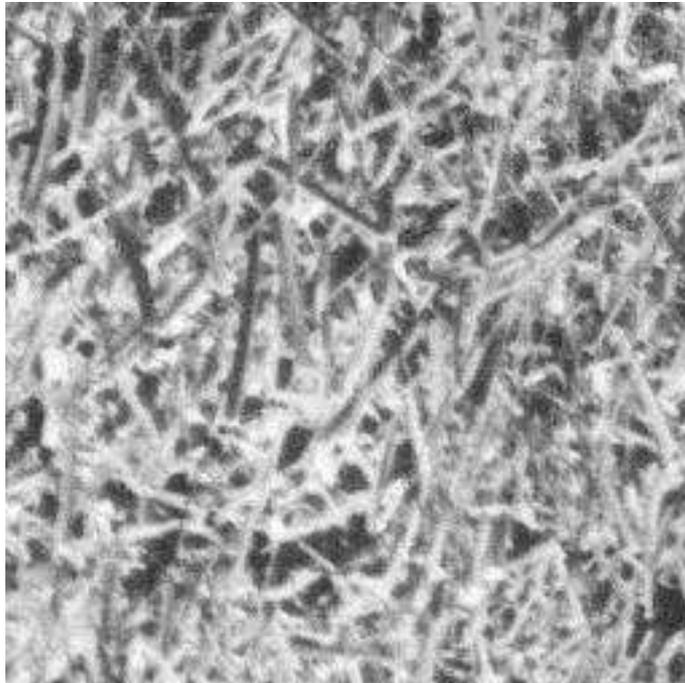


(a) « bark 1 »

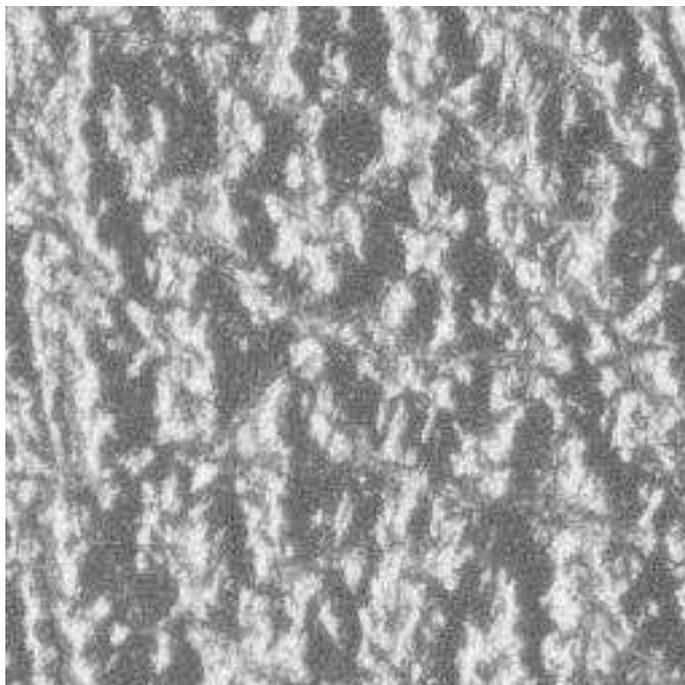


(b) « bark 2 »

Figure 5.3 – Textures extraites à l'échelle 256×256 . Les primitives sont nettement claires et donc pas de difficulté de discrimination visuelle entre textures.



(a) « grass »



(b) « leather »

Figure 5.4 – Textures extraites à l'échelle 256×256 (suite et fin)

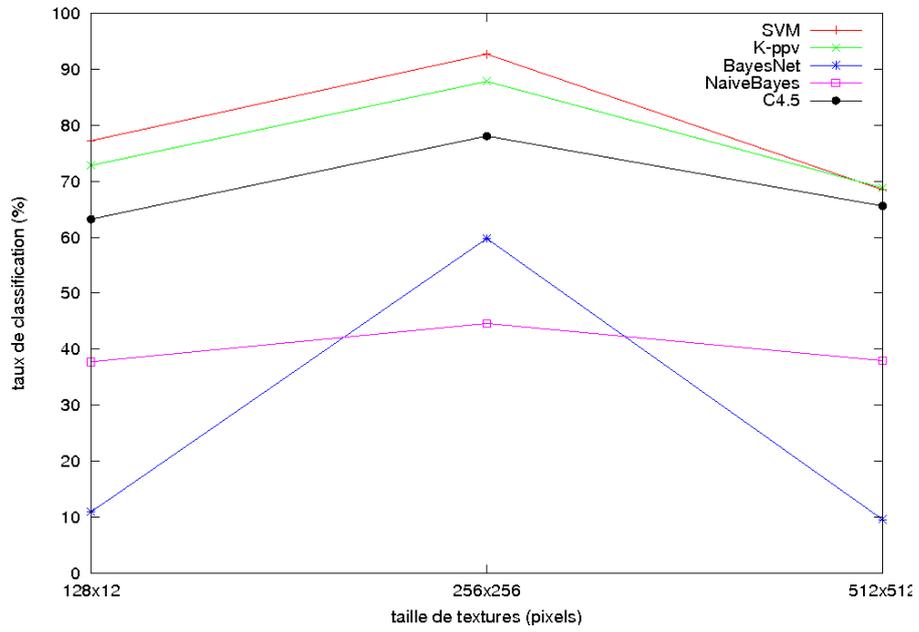


Figure 5.5 – Taux de bonne classification pour différentes tailles de texture

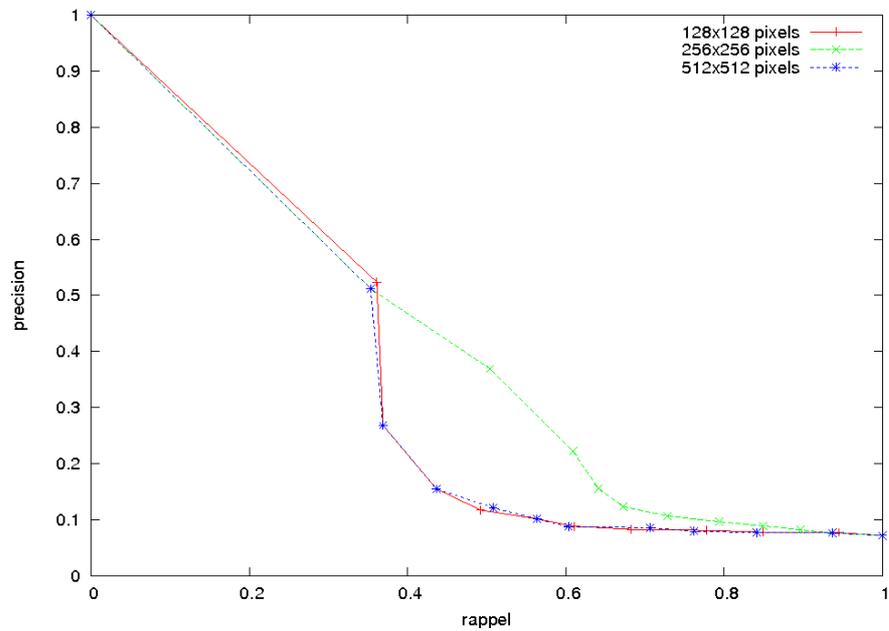


Figure 5.6 – Courbes de rappel/précision pour différentes tailles de texture

direction sont calculés sur les quatre directions 0° , 45° , 90° et 135° . Pour la distance d , le choix est fait expérimentalement. La distance qui fournira de meilleures performances sera choisie. Cinq distances ont été testées : 1, 2, 3, 5 et 7.

Les huit paramètres d'Haralick [116] retenus pour la caractérisation de textures par les MCO sont : l'énergie, l'entropie, la corrélation, le contraste, l'uniformité, la variance, l'homogénéité et la direction.

Nous représentons sur les figures 5.7 et 5.8 les taux de classifications pour différents classifieurs introduits dans le chapitre 4 et les courbes rappel/précision en fonction du pas d des matrices de co-occurrences respectivement pour les bases de Brodatz et Meastex.

Dans le cas de la base de Brodatz (figure 5.7), nous constatons que pour les classifieurs k -ppv, C4.5, NaiveBayes et arbre-NB, les taux de classification maximaux sont obtenus pour $d = 5$. Pour le classifieur SVM et BayesNet, les taux maximaux sont donnés respectivement par $d = 3$ et $d = 7$. Pour les courbes rappel/précision, les pas 5 et 7 donnent des performances proches.

Pour la base Meastex (figure 5.8), le taux de classification maximale est obtenu pour un pas égal à 2 tandis que les classifieurs k -ppv, BayesNet et arbre-NB atteignent un taux maximal pour $d = 5$. Le taux maximale atteint par C4.5 correspond à $d = 7$ et pour le NaiveBayes, il correspond à $d = 3$. Les courbes rappel/précision montrent la similarité des performances pour les distances 5 et 7, néanmoins la distance 1 donne meilleures performances pour les premières 40 % de la base qui diminuent rapidement par la suite.

Nous constatons que les trois distances (3, 5 et 7) sont candidates. Pour résoudre ce choix, nous avons effectué un test sur un sous-ensemble de textures de la base de Brodatz1 constitué de 136 textures orientées. L'idée de ce test est de mesurer un score (cf. chap:ApplTexture) pour les différentes distances. L'ensemble des réponses positives dans ce cas correspond aux orientations détectées correctement et pour les réponses négatives, il s'agit de fausses détections. La distance qui fournit le plus grand score reflète une bonne préservation de l'information direction est celle qui sera retenue. Nous avons trouvé les scores suivants $score_3 = 1,0606$, $score_5 = 1,3224$ et $score_7 = 1,3358$ pour les trois distances respectivement. Étant donné la différence entre le $score_3$ et le $score_7$ n'est pas importante, nous suggérons d'utiliser la distance 5 pour la suite des évaluations.

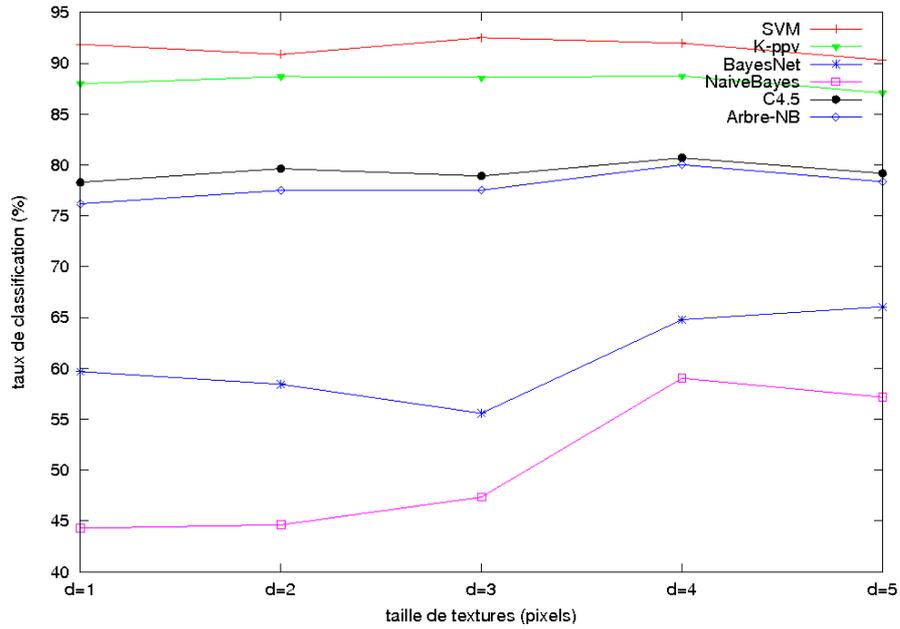
5.2.2.1 Analyse des attributs MCO

Dans cette section, nous présentons le comportement des descripteurs de texture extraits des MCO pour les deux classes Brodatz1 et Meastex. Cette analyse nous guidera dans le choix des descripteurs caractérisant au mieux nos différentes bases.

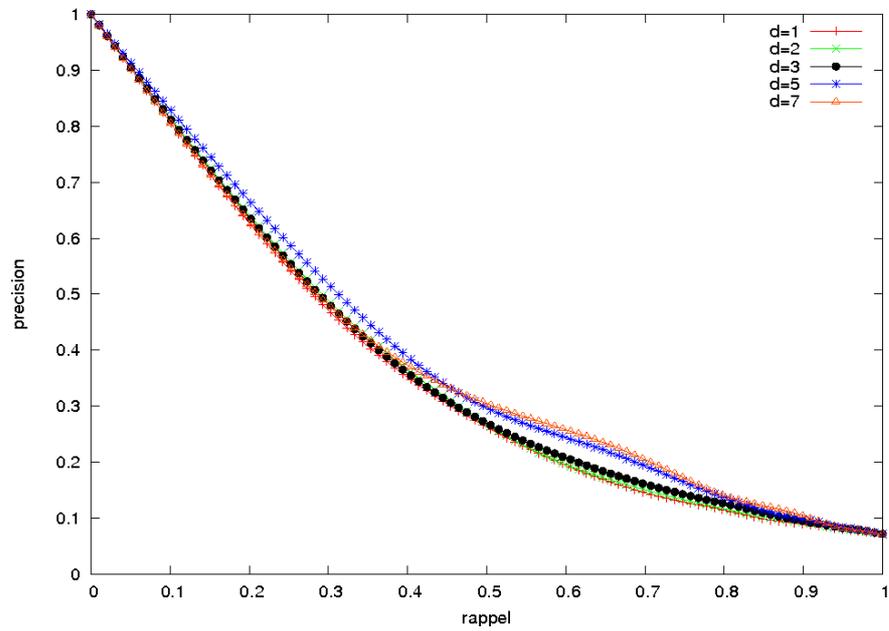
Base de Brodatz1 La figure 5.9 illustre la distribution des densités de probabilités des différents descripteurs MCO pour les différentes classes de la base de Brodatz1. Nous constatons que pour les différents descripteurs, il y a un fort recouvrement entre les treize classes sauf pour la classe « wall » qui se distingue fortement du reste des classes notamment pour les descripteurs : énergie, entropie, uniformité, homogénéité, contraste et direction.

Une interprétation de ce phénomène peut être donnée comme suit. En explorant visuellement la base de Brodatz1, nous avons remarqué que la classe « wall » est une classe particulière du fait de l'aspect fort homogène qui domine cette base. Les différentes textures de la classe « wall » partagent les mêmes propriétés de texture¹ pour les différents descripteurs ainsi elles ont toutes des valeurs numériques proches; ce qui est traduit par une courbe de densité de probabilités presque uniforme pour les différents

¹Nous faisons ici référence à l'information de texture sémantique.

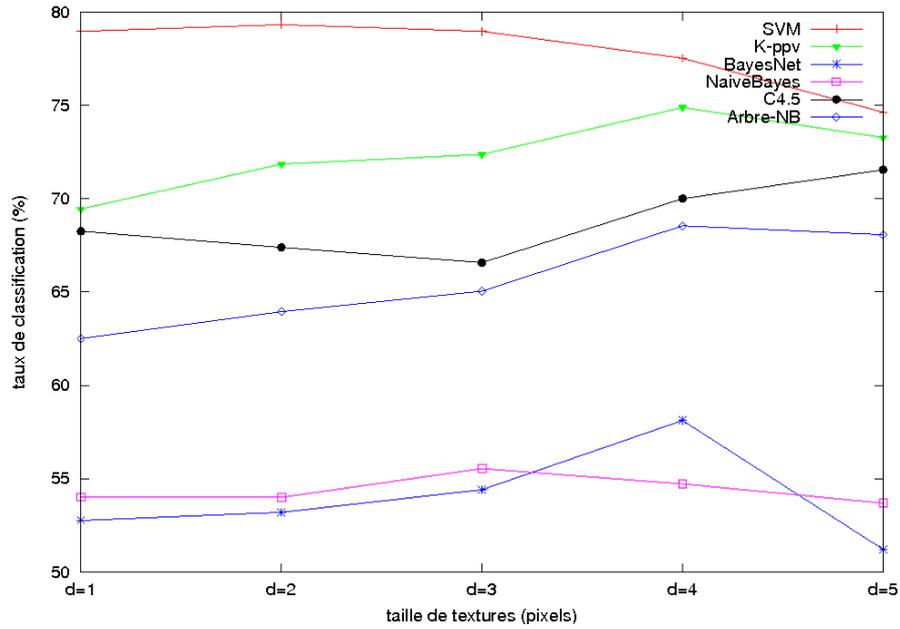


(a) taux de classification

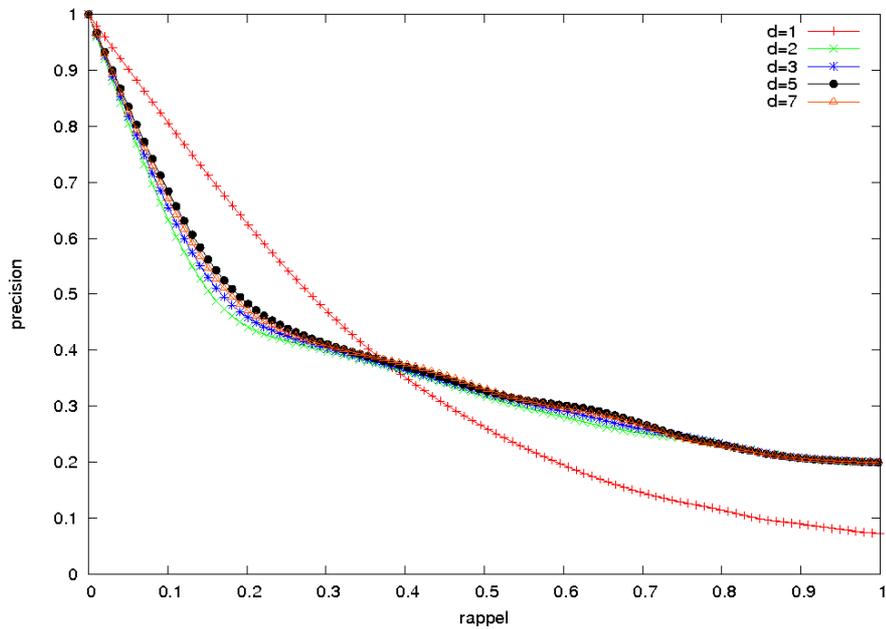


(b) courbes rappel/précision

Figure 5.7 – Base de Brodatz testée avec différents classifieurs



(a) taux de classification



(b) courbes rappel/précision

Figure 5.8 – Base de Meastex testée avec différents classifieurs

paramètres de texture. Pour les autres classes de la base Brodatz1, elles ont le même comportement. Les différentes textures appartenant à une même classe donnée ont des valeurs différentes pour une propriété de texture, ce qui se traduit par des courbes de densités de probabilités multimodales qui se chevauchent.

Base de Meastex Pour la base Meastex, l'aspect uniforme et homogène des différentes classes est vérifiée par les courbes de densité de probabilités données par la figure 5.10. Il y a un fort recouvrement entre les cinq classes de la base. Ce qui montre la difficulté de classifier les différentes textures. Cette remarque est illustrée par le taux de classification par SVM qui atteint 91,96 % pour la base de Brodatz1 tandis qu'il ne dépasse pas les 78 % pour la base Meastex.

Nous concluons de cette première analyse que les descripteurs de texture extraits des MCO sont des indicateurs prometteurs pour la caractérisation primaire et sémantique de textures.

5.2.3 Descripteurs de Tamura

Quatre paramètres parmi les six introduits par Tamura [290] sont retenus pour la caractérisation de textures par la méthode Tamura, à savoir : la grossièreté (*coarseness*), le contraste, le *line-likeness* et la directivité.

Le calcul du premier descripteur (grossièreté) dépend du paramètre k . Pour notre analyse, nous avons opté pour la valeur 3. Ce choix est justifié par les présentes figures 5.11 et 5.12 qui représentent les taux de classification et les courbes rappel/précision estimés respectivement sur la base Brodatz1 et Meastex.

5.2.3.1 Analyse des attributs de Tamura

Dans cette section, nous présentons le comportement des descripteurs de texture extraits par la technique de Tamura pour les deux classes Brodatz1 et Meastex. Cette analyse nous permettra d'avoir une idée sur les descripteurs caractérisant au mieux les différentes bases ainsi que leurs classes.

Nous constatons à partir des figures 5.13 et 5.14 qu'il existe un fort recouvrement entre les différentes classes de textures pour les deux bases d'analyse. Ainsi, ces figures montrent la difficulté de la tâche de classification. Les taux de classification pour les deux bases Brodatz1 et Meastex atteignent respectivement 76,65 % et 77,89 %. Ces taux assez proches peuvent mener à une confusion pour le comportement des deux bases.

5.2.4 Comparaison entre les méthodes d'extraction de texture

Nous comparons dans cette partie le comportement des différentes méthodes d'extraction de textures ainsi que l'impact de la combinaison de ces méthodes. La figure 5.15 représente les courbes rappel/précision. Pour la base Brodatz1, la figure montre que les descripteurs extraits des MCO sont meilleurs pour la recherche d'images similaires que les descripteurs de Tamura. Néanmoins, leurs capacité de détection se dégrade pour la base Meastex et ce sont les attributs de Tamura ainsi que le regroupement des deux méthodes qui améliorent la performance de la recherche.

Un constat immédiat important tiré de ces courbes est que les attributs extraits des MCO sont meilleurs que ceux de Tamura puisqu'ils permettent de discerner entre textures ayant des caractéristiques différentes (Brodatz1) et celles qui sont proches visuellement (Meastex) au contraire des attributs de Tamura qui sont insuffisants dans le cas d'une grande variabilité de l'information de texture.

Les tableaux 5.1 et 5.2 donnent les matrices de confusion, les taux de classification moyens, la F-mesure, la mesure Kappa et les taux d'erreurs moyens estimés pour la base Brodatz et Meastex respectivement. Dans les deux cas de figures, nous remarquons que le taux de classification du regroupement est

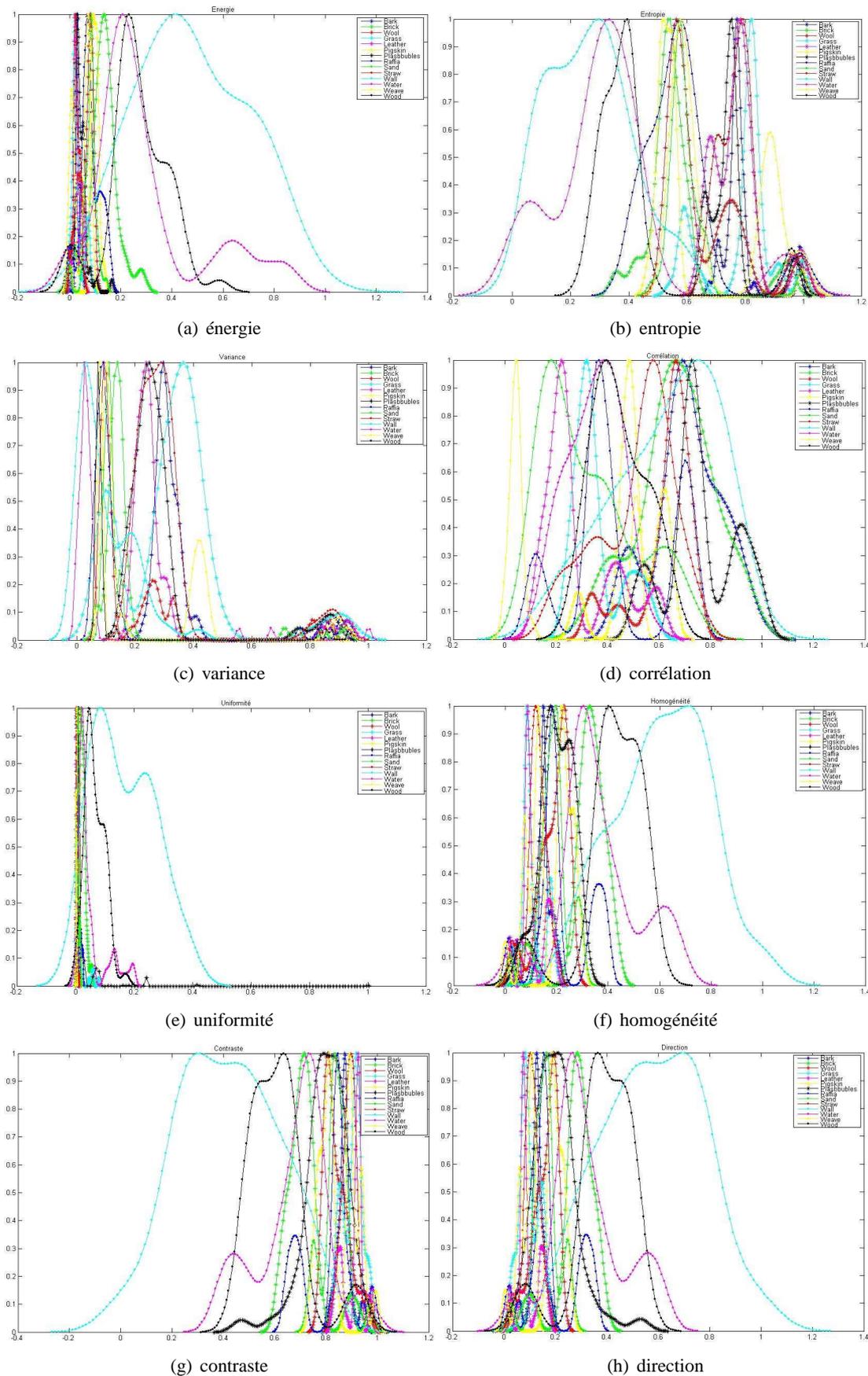


Figure 5.9 – Les densités de probabilités estimées pour les descripteurs MCO calculés sur la base de Brodatz1

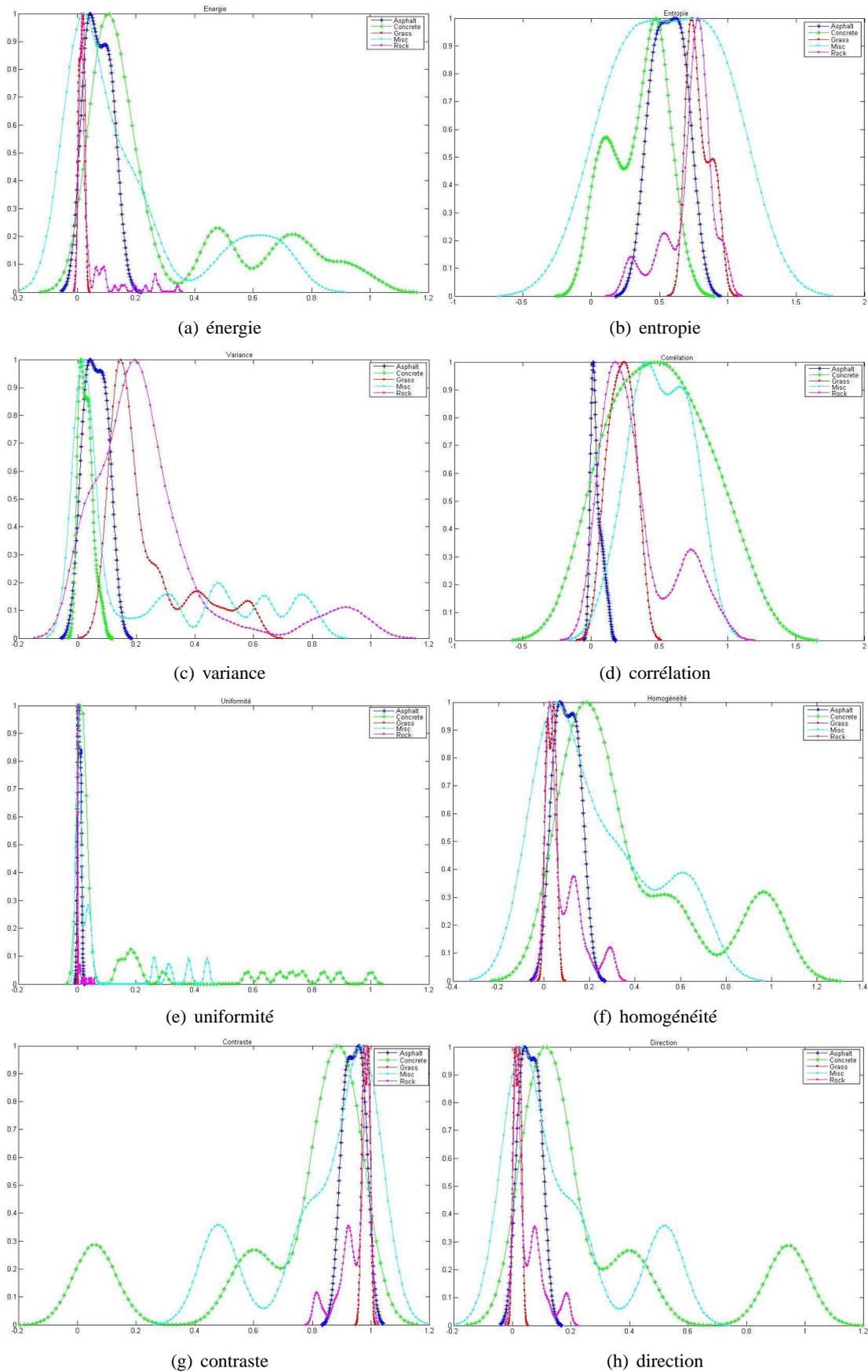


Figure 5.10 – Les densités de probabilités estimées pour les descripteurs MCO calculés sur la base Meastex

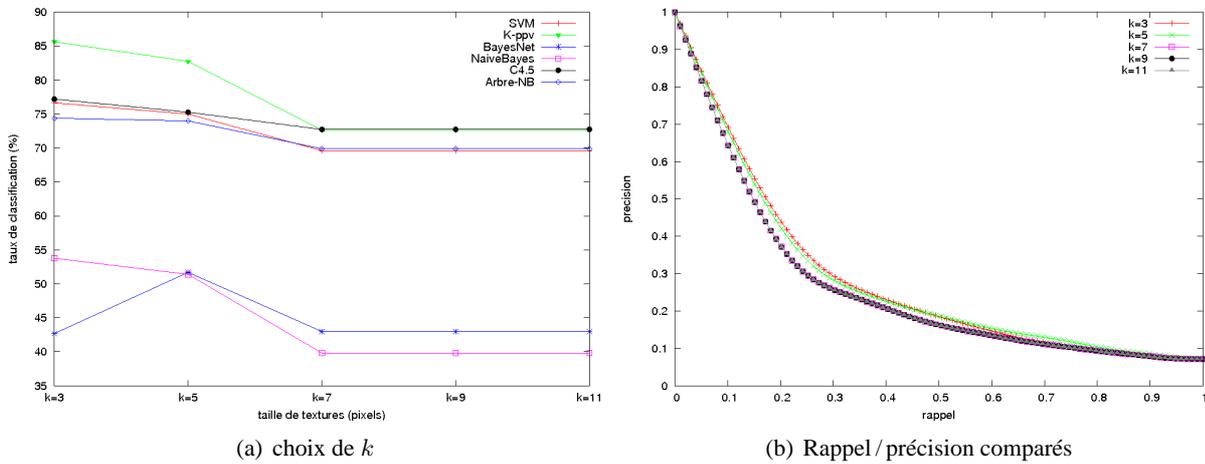


Figure 5.11 – Taux de classification pour différents classifieurs et courbes de rappel/précision pour la base Brodatz

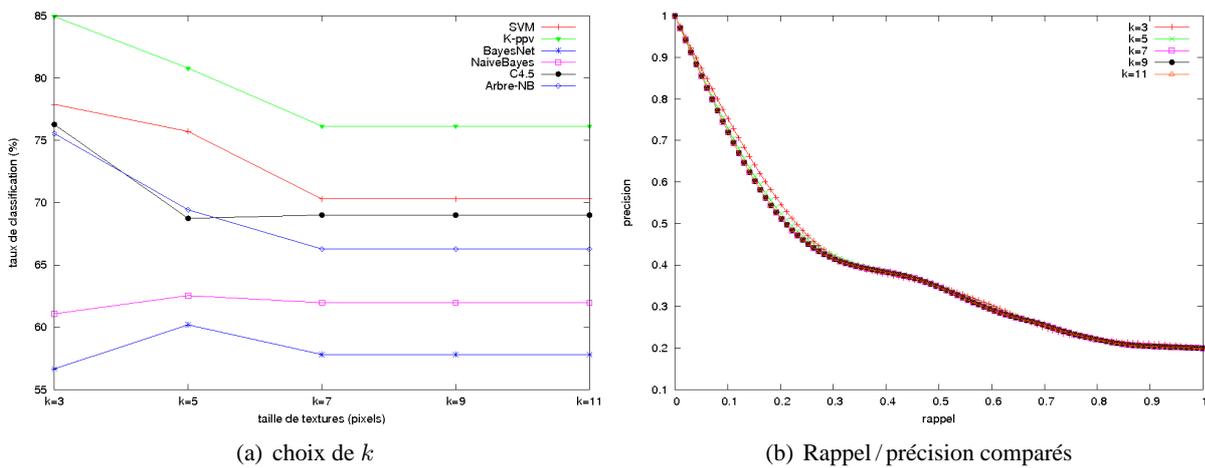


Figure 5.12 – Taux de classification pour différents classifieurs et courbes de rappel/précision pour la base Meastex

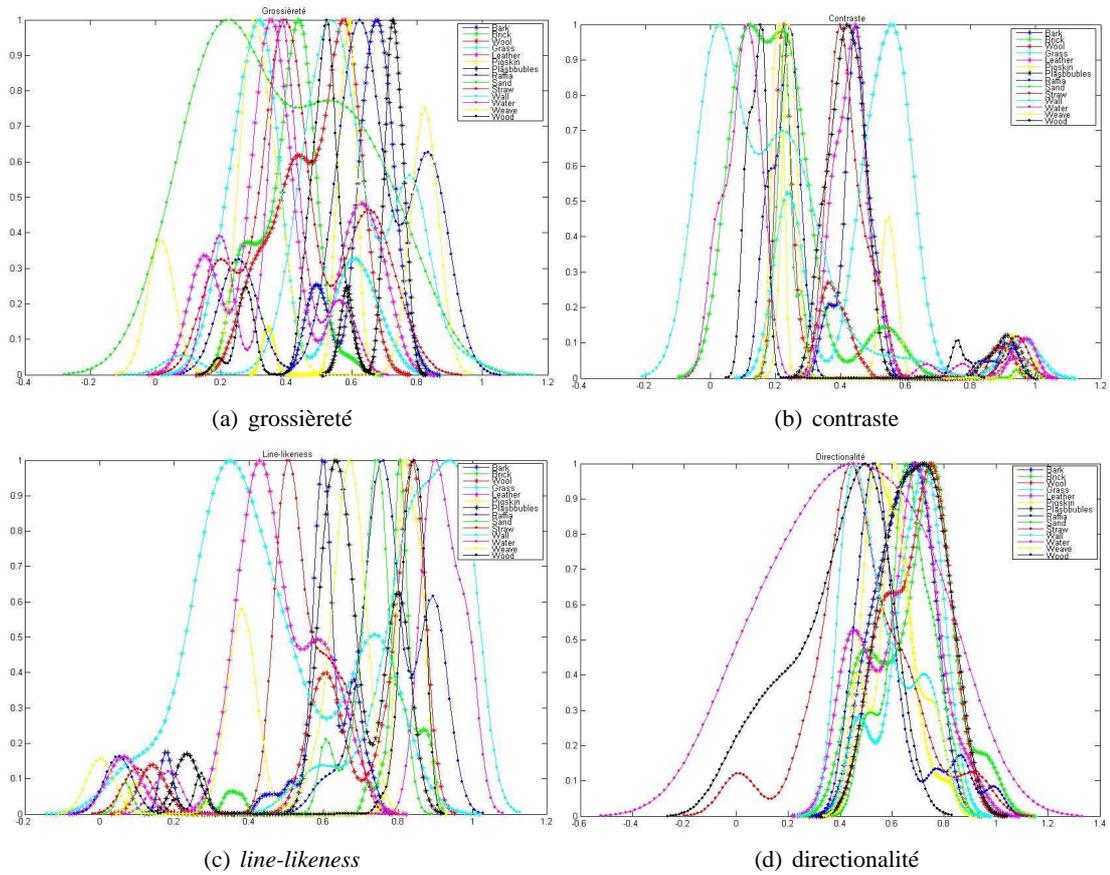


Figure 5.13 – Les densités de probabilités estimées pour les descripteurs de Tamura pour la base de Brodatz 1

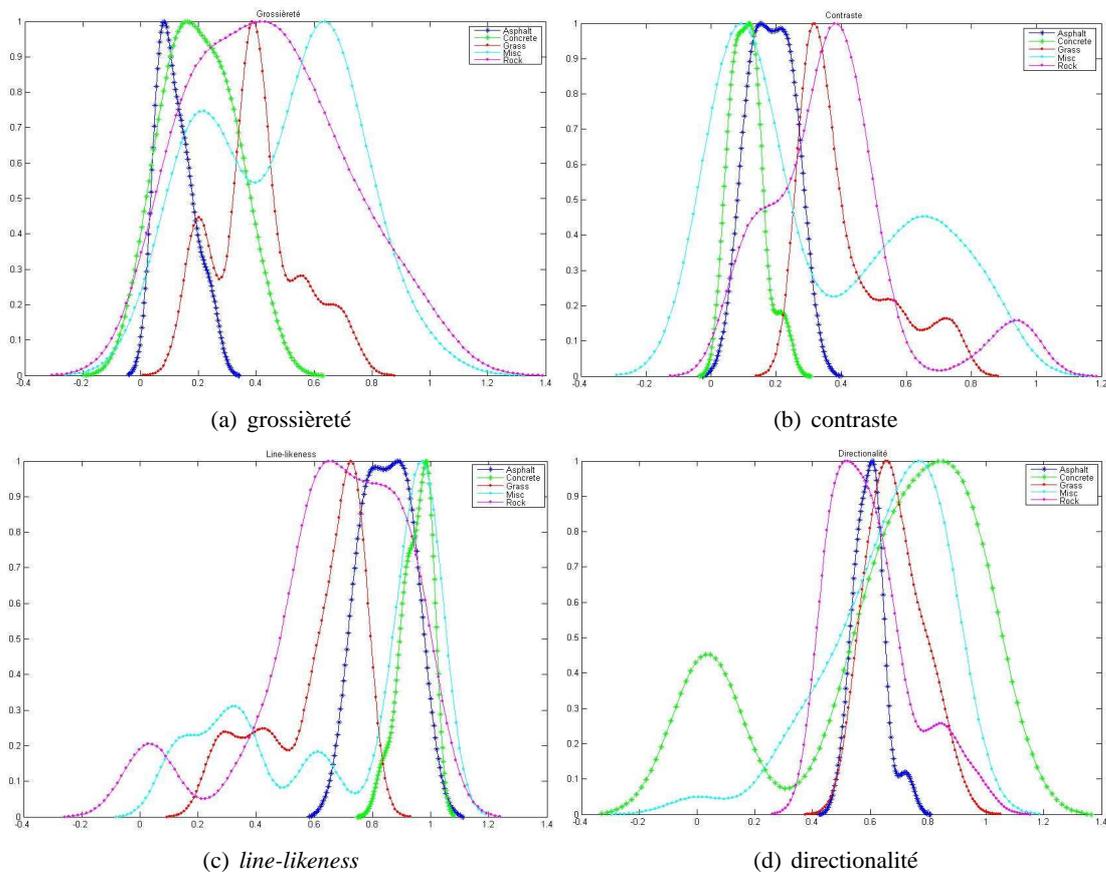
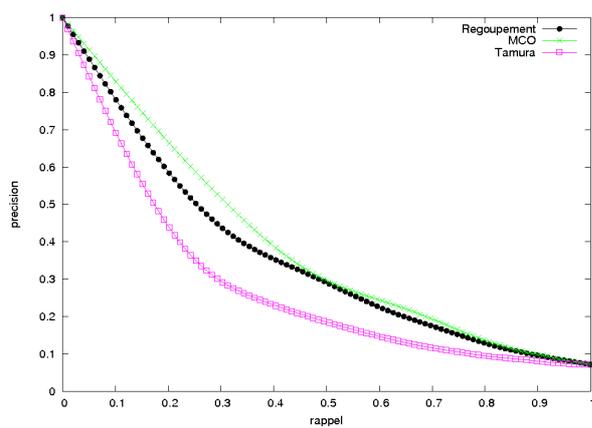
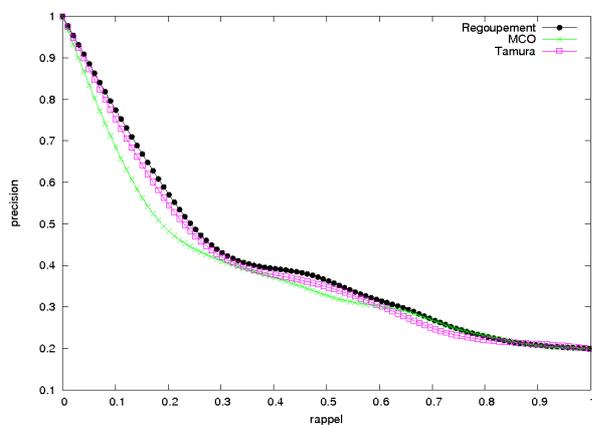


Figure 5.14 – Les densités de probabilités estimées pour les descripteurs de Tamura pour la base Meastex



(a) Brodatz 1



(b) Meastex

Figure 5.15 – Taux de classification pour les différentes méthodes d'extraction de texture

supérieur des deux approches utilisées. Le regroupement donc améliore la capacité du système à classer les textures. Néanmoins, pour la base Meastex, les taux moyens restent très modestes et sont assez proches ne dépassant pas les 90 % contre 95 % pour la base de Brodatz.

Nous déduisons aussi de ces tableaux que les différentes classes de la base Brodatz1 sont bien caractérisées par les attributs extraits des MCO sauf pour la classe « straw » et « wood » où le taux de bonne classification atteint respectivement 61,61 % et 84,82 %. Pour la base Meastex, seule la classe « Grass » est fortement détectée (94 %). Pour les attributs de Tamura, en général, ils manifestent une modeste classification des différentes classes des deux bases. Cette déduction peut être aussi faite en analysant les valeurs de la F-mesure F et celles de Kappa K .

En ce qui concerne les performances de la recherche d'images similaires, l'évaluation faite à partir de la mesure du score proposée rejoint les résultats obtenus par les courbes de rappel / précision. La classification par les attributs regroupés et ceux des MCO permettent une bonne détection de la base de Brodatz. Une détection modeste pour la base Meastex est donnée par les attributs Tamura. En conséquence, cette analyse nous a permis de conclure que les attributs MCO sont plus performants que les attributs Tamura en terme de recherche d'images similaires (courbes de rappel / précision) et pour la classification des textures.

5.3 Réduction de l'espace des attributs de texture

La phase de réduction de l'espace de représentation des textures en terme des attributs de texture est primordiale pour toute application de classification, segmentation, recherche, etc. Il s'agit en fait de réduire l'espace initiale de grandes dimensions en un nombre réduit tout en gardant le maximum de l'information utile.

Dans notre cas, nous avons douze descripteurs à partir du regroupement des attributs MCO et Tamura. Malgré que ce nombre apparaît modeste, or d'après les analyses faites précédemment, nous avons constaté qu'il y a des redondances entre descripteurs avec une éventualité d'avoir des descripteurs non pertinents, ce qui pourra altérer les résultats des différentes applications. Pour cela, il nous est apparu judicieux d'effectuer une réduction du nombre de paramètres en éliminant ceux redondants et non pertinents. La réduction peut donc se faire comme nous l'avons vu dans le chapitre 3 de deux manières : soit en calculant de nouveaux paramètres à partir de ceux initiaux ou bien en faisant une sélection d'un sous-ensemble sur ces paramètres. L'inconvénient de la première approche est la perte de la signification physique des paramètres, information importante dans notre cadre de travail puisque la description de la texture en niveau sémantique fait fortement appel à cette notion. Par la suite, nous avons opté pour la seconde approche de sélection.

Les différentes approches de réduction de dimension par sélection des paramètres introduits dans la section 3.4 vont être étudiées et analysées expérimentalement dans ce qui suit.

5.3.1 Normalisation des paramètres

Le tableau 5.3 donne quelques statistiques des différents attributs de texture extraits des MCO et Tamura. On remarque que la plage des valeurs que peut prendre un descripteur varie fortement d'un descripteur à un autre. Par exemple pour la base Brodatz1, l'intervalle de variabilité de l'énergie est faible de $[0, 0,0005]$ tandis que pour la variance, elle est de l'ordre de quelques milliers, etc. Cette hétérogénéité des valeurs brutes peut affecter le comportement des techniques d'extraction des descripteurs pertinents et par la suite, biaiser les résultats de la classification et de la recherche des images: les descripteurs ayant des valeurs élevées risquent d'avoir plus de poids que ceux atteignant des valeurs plus faibles.

	MC (%)														E (%)	T _c (%)	F _m	K	E _t (%)	S					
MCO	96,43	0,00	1,78	0,00	0,00	0,00	0,00	0,00	0,00	1,78	0,00	0,00	0,00	0,00	3,57	91,13	0,908	0,903	8,86	0,256					
	0,00	92,56	0	0,00	0,00	0,00	0,00	0	1,78	0,00	0,00	0,00	0,00	5,65	7,44										
	3,78	0,00	90,18	0,00	0,00	1,78	0,00	0	2,08	2,08	0,00	0,00	0,00	0,00	9,82										
	0,00	0,00	1,78	92,56	0,00	0,00	0,00	0,00	0,00	5,65	0,00	0,00	0,00	0,00	7,44										
	0,00	0,00	4,16	0,00	91,96	0,00	0,00	3,78	0,00	0,00	0,00	0,00	0,00	0,00	8,09										
	0,00	0,00	3,87	2,08	0,00	94,05	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	5,95										
	3,78	1,78	0,00	1,78	0	0,00	90,47	0,00	0,00	0,00	2,08	0,00	0,00	0,00	9,52										
	0,00	0,00	0,00	0	3,78	0,00	0,00	96,13	0,00	0,00	0,00	0,00	0,00	0,00	3,87										
	0,00	0,00	1,78	0,00	0,00	0,00	0,00	0,00	98,21	0,00	0,00	0,00	0,00	0,00	1,78										
	5,65	0,00	2,08	19,05	5,95	1,78	2,08	0,00	1,78	61,61	0,00	0,00	0,00	0,00	38,39										
	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	98,21	0,00	0,00	1,78	1,78										
	0,00	1,78	0,00	3,78	0,00	0,00	0,00	0,00	0	2,08	90,47	0,00	1,78	0,00	9,52										
	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,78	0,00	0,00	0,00	0,00	98,21	0,00	1,78										
	0,00	7,73	0,00	0,00	0,00	0,00	1,78	0,00	0,00	0,00	5,65	0,00	0,00	84,82	15,18										
Tamura	64,58	0,00	0,00	1,78	9,82	0,21	7,62	0,00	0,00	2,08	0,00	0,00	0,00	0,00	35,42						81,23	0,815	0,797	18,77	0,532
	0,00	86,31	2,08	0,00	0,00	0,00	0,00	2,08	2,08	0,00	2,08	3,57	0,00	1,78	13,69										
	0,00	0,00	90,47	0,00	0,00	5,65	0,00	0,00	2,08	0,00	1,78	0	0	0,00	9,52										
	5,36	0,00	0,00	81,55	9,82	0	0,00	0,00	3,57	0,00	0	0,00	0,00	0,00	18,45										
	11,90	0,00	2,08	3,57	72,92	0,00	0,00	0,00	1,78	7,74	0	0,00	0,00	0,00	27,08										
	0,00	2,08	3,87	0,00	0,00	84,52	0,00	0,00	5,95	0,00	1,78	0,00	0,00	0,00	15,47										
	28,27	0,00	0,00	0,00	0,00	0,00	66,37	0,00	0,00	3,57	1,78	0,00	0,00	0,00	33,63										
	0,00	0,00	0,00	0	0,00	5,95	0,00	79,17	0,00	3,57	0,00	0,00	0,00	0,00	20,83										
	0,00	1,78	0,00	0,00	0,00	0,00	0,00	3,87	90,77	0,00	3,57	0,00	0,00	0,00	9,22										
	3,87	3,57	0,00	0,00	9,52	0,00	0,00	0,00	4,17	72,62	4,16	2,98	0,00	0,00	27,38										
	2,08	0,00	3,57	0,00	0,00	3,87	0,00	5,65	0,00	0,00	3,87	83,03	0,00	0,00	16,96										
	0,00	5,95	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	3,87	86,61	0,00	3,57	13,39										
	0,00	0,00	0,00	0,00	0,00	0,00	2,08	0,00	3,87	10,12	0,00	0,00	94,05	0,00	5,94										
	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,78	0,00	0,00	3,87	10,12	0,00	84,22	15,77										
Regroupement	100	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	95,94	0,958	0,955	4,06	0,149					
	0,00	100	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00										
	0,00	0,00	96,13	3,87	0,00	0,00	0,00	0,00	0,00	0,00	0	0	0,00	0,00	3,87										
	0,00	0,00	0,00	92,86	1,78	0,00	0,00	0,00	1,78	0,00	3,57	0,00	0,00	0,00	7,14										
	0,00	0,00	0,00	0,00	100	0,00	0,00	0,00	0,00	0	0,00	0,00	0,00	0,00	0,00										
	0,00	0,00	0,00	0,00	0,00	95,83	0,00	0,00	4,17	0,00	0,00	0,00	0,00	0,00	4,16										
	1,78	0,00	0,00	0,00	0,00	0,00	94,05	0,00	0,00	2,08	2,08	0,00	0,00	0,00	5,95										
	0	0,00	0,00	0,00	0,00	1,78	0,00	98,2	0,00	0,00	0,00	0,00	0,00	0,00	1,87										
	0,00	0,00	2,08	1,78	0,00	0,00	0,00	0,00	96,13	0,00	0,00	0,00	0,00	0,00	3,87										
	4,17	0	2,08	11,31	0,00	0,00	1,78	1,78	0,00	77,08	0,00	1,78	0,00	0,00	22,92										
	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0	100	0,00	0,00	0,00	0,00	0,00										
	0,00	0,00	0,00	0,00	0,00	1,78	0,00	0,00	0,00	0,00	1,78	94,64	0,00	1,78	5,36										
	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0	0,00	0,00	0,00	100	0,00	0,00										
	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0	1,78	0,00	98,21	0,00	1,78										

Table 5.1 – Résultats de classification par SVM pour les différentes approches d'extraction de texture appliquées à la base de Brodatz

	MC (%)	E (%)	T_c (%)	F_m	K	E_t (%)	S
MCO	$\begin{pmatrix} \mathbf{84,00} & 4,00 & 0,00 & 10,00 & 2,00 \\ 0,00 & \mathbf{80,67} & 0,00 & 6,00 & 13,33 \\ 0,00 & 0,00 & \mathbf{94,00} & 2,00 & 4,00 \\ 10,00 & 10,00 & 6,00 & \mathbf{55,33} & 18,67 \\ 0,00 & 5,70 & 12,86 & 2,66 & \mathbf{78,57} \end{pmatrix}$	$\begin{pmatrix} 16,00 \\ 19,33 \\ 6,00 \\ 44,67 \\ 21,43 \end{pmatrix}$	78,51	0,781	0,72	21,48	0,285
Tamura	$\begin{pmatrix} \mathbf{86,00} & 0,00 & 0,00 & 10,00 & 4,00 \\ 2,00 & \mathbf{88,67} & 0,00 & 0,00 & 9,33 \\ 0,00 & 0,00 & \mathbf{94,00} & 4,00 & 2,00 \\ 10,00 & 6,00 & 6,00 & \mathbf{74,00} & 4,00 \\ 0,00 & 7,14 & 5,71 & 2,86 & \mathbf{84,28} \end{pmatrix}$	$\begin{pmatrix} 14,00 \\ 11,33 \\ 6,00 \\ 26,00 \\ 15,71 \end{pmatrix}$	85,39	0,854	0,816	14,60	0,177
Regroupement	$\begin{pmatrix} \mathbf{88,00} & 0,00 & 0,00 & 10,00 & 2,00 \\ 2,00 & \mathbf{86,67} & 0,00 & 2,00 & 9,33 \\ 0,00 & 0,00 & \mathbf{96,00} & 0,00 & 4,00 \\ 10,00 & 6,00 & 4,00 & \mathbf{76,00} & 4,00 \\ 0,00 & 7,14 & 5,71 & 1,42 & \mathbf{85,71} \end{pmatrix}$	$\begin{pmatrix} 12,00 \\ 13,33 \\ 4,00 \\ 24,00 \\ 14,28 \end{pmatrix}$	86,74	0,870	0,830	13,52	0,165

Table 5.2 – Résultats de classification par SVM pour les différentes approches d'extraction de texture appliquées à la base Meastex

Descripteur	Moyenne	Écart-type
énergie	2,54e-004	2,8154e-004
entropie	13,177	1,223
homogénéité	0,044	0,026
uniformité	4,0989e-006	7,585e-006
contraste	9,1381e+018	5,4376e+016
variance	1,27e+003	1,1946e+003
corrélation	0,287	0,181
direction	0,009	0,005
grossièreté	5,031	0,681
contraste 2	23,690	12,423
<i>line-likeness</i>	0,659	0,217
directivité	3,631	0,990

Table 5.3 – Statistiques du 1^{er} ordre pour l'ensemble des attributs de Texture

	T_c (%)	F_m	K	E_t (%)
CFS	93,17	0,929	0,926	6,824

Table 5.4 – Mesures de classification obtenues par l'algorithme CFS

Plusieurs techniques de normalisation existent pour enlever ce biais. Nous utilisons ici la technique de normalisation la plus répandue dite normalisation « min-max » (cf. formule 5.9). Les valeurs de chaque descripteur sont ramenées à l'intervalle $[0, 1]$:

$$X_n = \frac{X - \min}{\max - \min} \quad (5.9)$$

Dans ce qui suit, nous présenterons les différents modèles de texture obtenus par les différents algorithmes introduits dans la section 3.4 dans le cas de la classification des textures de la base Brodatz1.

5.3.2 Algorithme CFS

L'algorithme CFS présenté dans la section 3.4 utilise la mesure de mérite basée sur l'incertitude symétrique pour l'évaluation de la pertinence des attributs de texture. L'attribut qui présente une valeur importante est sélectionné.

En appliquant l'algorithme CFS sur les douze paramètres, le sous ensemble sélectionné qui donne le meilleur taux de classification est de taille 7 et constitué de : l'entropie, la variance, l'uniformité, l'homogénéité, la corrélation, le contraste*² et la grossièreté. Les performances de classification par le modèle CFS sont données dans le tableau 5.3.2 ainsi que les valeurs de la mesure mérite pour les différents descripteurs choisis sont dressées dans le tableau 5.3.2.

5.3.3 Algorithme MIFS

L'algorithme MIFS introduit dans la section 3.4 est un algorithme de sélection des attributs qui tient en compte de l'interaction entre les attributs et les classes d'étiquetage en se basant sur l'information

²calculé par Tamura

Descripteur	Merit
variance	0,6664
uniformité	0,6529
grossièreté	0,6236
homogénéité	0,660
corrélation	0,5900
contraste 2	0,5393
entropie	0,477

Table 5.5 – Mesures de *Merit* par l’algorithme CFS

Descripteur	IM
grossièreté	1,656
homogénéité	0,415
corrélation	0,396
directivité	0,067

Table 5.6 – Mesures de l’information mutuelle par l’algorithme MIFS

mutuelle. L’attribut qui maximise cette information est retenu par l’algorithme.

Le tableau 5.3.3 dresse les valeurs de l’information mutuelle IM pour les différents descripteurs retenus par l’algorithme MIFS qui sont au nombre de quatre : l’homogénéité, la grossièreté, la corrélation et la directivité. Le taux de classification obtenu par le modèle SVM est de 83,65 %. Les différentes mesures d’évaluation de la classification par le modèle MIFS sont données sur le tableau 5.7.

5.3.4 Algorithme ReliefF

L’algorithme ReliefF introduit dans la section 3.4 mesure la pertinence d’un attribut en terme de score qui lui est attribué. Ce score n’est autre que le poids estimé par le classifieur. Pour cet algorithme, le classifieur en question est le k plus proches voisins k -ppv.

Le modèle sélectionné par l’algorithme Relief comprend cinq descripteurs qui sont : l’entropie, l’homogénéité, la corrélation, le contraste et la direction. Les scores correspondants aux attributs choisis sont donnés dans le tableau 5.8. Le modèle ReliefF obtenu donne un taux de 84,35 % pour la classification des textures. Les différentes performances correspondantes au modèle ReliefF sont récapitulées dans le tableau 5.9.

5.3.5 Algorithme Fisher

L’algorithme Fisher rejoint le même principe de l’algorithme précédent pour l’évaluation de la pertinence des attributs en se basant sur les scores. Ce score mesure le degré de séparation entre les classes. L’attribut à sélectionner est celui qui maximise ce critère.

	T_c (%)	F_m	K	E_t
MIFS	83,65	0,821	0,832	16,35

Table 5.7 – Performances de classification par l’algorithme MIFS

Descripteur	Score_ReliefF
corrélation	0,146
entropie	0,134
homogénéité	0,114
direction	0,108
contraste	0,108

Table 5.8 – Mesures des scores par l’algorithme ReliefF

	T_c %	F_m	K	E_t
ReliefF	84,35	0,846	0,830	15,64

Table 5.9 – Performances de classification par l’algorithme ReliefF

Le modèle obtenu par Fisher est composé de cinq descripteurs : l’énergie, l’entropie, l’homogénéité, le contraste et la direction. Le tableau 5.3.5 donne les scores obtenus par l’algorithme Fisher pour les différents descripteurs. Ce modèle donne un taux de classification de 81,13 %. Les autres mesures de performance sont illustrées dans le tableau

5.3.6 Algorithmes génétiques

La mise en œuvre sélection des descripteurs pertinents par les algorithmes génétiques (AG) nécessite le choix d’un ensemble de paramètres tels que la taille de la population, les probabilités de mutation et de croisement (cf. section 3.4). Ici, nous avons choisi une population de 10 chromosomes pour 20 itérations, une probabilité de mutation de 0,01 et une probabilité de croisement de 0,8. Une valeur petite pour la probabilité de mutation évite à l’algorithme de converger vers une solution non-optimale et une valeur grande pour la probabilité de croisement afin d’avoir une diversité dans la population. A chaque itération de génération, nous allons sélectionner le chromosome qui donne le meilleur taux de classification ce qui nous ressort à la fin 20 meilleurs chromosomes candidats. Pour choisir un parmi eux, nous cherchons les paramètres qui ont été fréquemment sélectionnés suivant différents seuils.

La figure 5.16 illustre le nombre d’occurrences de chaque paramètre sélectionné par les algorithmes génétiques pour la classification. A titre d’exemple, le descripteur numéro 7 (contraste pour les MCO) a été choisi 6 fois alors que le 11^e paramètre (le *line-likeness* de Tamura) a été sélectionné dans toutes les itérations (20 fois). Le modèle que nous avons choisi est composé de paramètres ayant été sélectionné plus que 10 fois : l’énergie, l’entropie, la variance, l’uniformité, la direction, la grossièreté, le *line-likeness* et la directionnalité.

Le tableau 5.12 donne les différentes performances de classification pour le modèle génétique. En

Descripteur	Score_Fisher
homogénéité	208,94
contraste	198,75
direction	198,75
énergie	159,26
entropie	154,37

Table 5.10 – Mesures du critère de Fisher

	T_c (%)	F_m	K	E_t
ReliefF	81,13	0,811	0,794	18,86

Table 5.11 – Performances de classification par l’algorithme Fisher

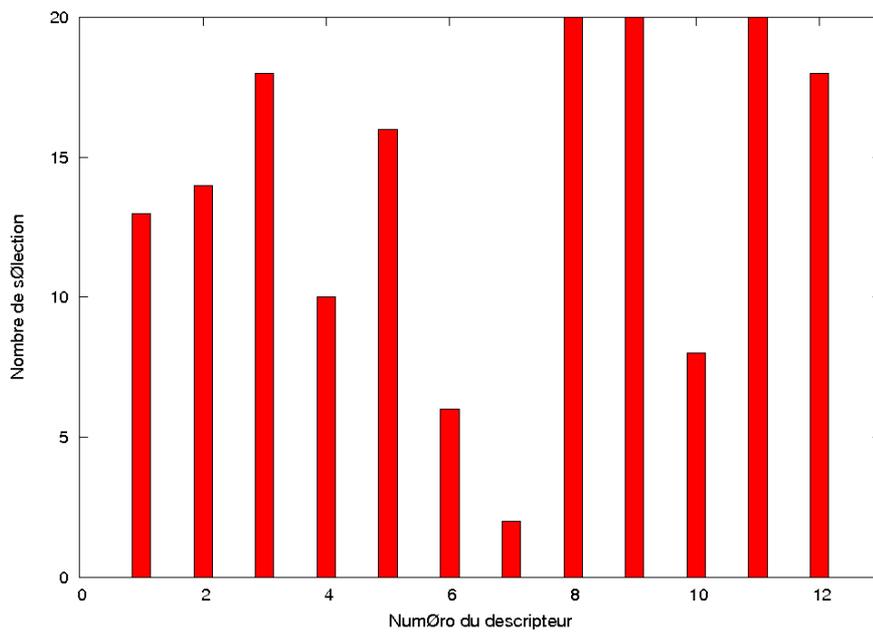


Figure 5.16 – Nombre d’occurrences des paramØtres sØlectionnØs par les AG pour 20 itØrations

	T_c (%)	F_m	K	E_t
GA	92,98	0,926	0,922	7,01

Table 5.12 – Performances de classification par les algorithmes génétiques

	T_c (%)	F_m	K	E_t
PCFS	96,07	0,959	0,959	3,93

Table 5.13 – Performances de classification par l’algorithme PCFS

comparant le taux de classification obtenu par les algorithmes génétiques et celui du modèle global (cf. tableau 5.1), nous remarquons que les algorithmes génétiques n’améliorent pas les résultats de classification en sélectionnant les meilleurs paramètres. Pourtant, elles fournissent un taux de classification supérieur aux autres approches de sélection étudiées en haut.

5.3.7 Nos propositions

Les algorithmes expérimentés en haut mesure la pertinence d’un attribut soit en terme de sa capacité à regrouper les images de même classe et séparer celles qui sont étiquetées différemment exprimée par des scores (CFS, MIFS, ReliefF), soit en terme de poids estimé par un classifieur (Fisher et AG). Donc toute éventuelle corrélation entre les attributs n’est pas pris en compte. Dans cette optique, nous avons proposé dans un premier temps l’algorithme PCFS de type *wrapper* qui tient compte de la corrélation des attributs et ainsi élimine celui qui fournit un taux de discrimination faible. La corrélation est calculé par le paramètre Pearson et l’évaluation du taux de classification est mesurée par le classifieur SVM.

5.3.7.1 Algorithme PCFS

Pour les douze descripteurs, le PCFS a détecté une corrélation négative entre l’homogénéité et le contraste. L’élimination de l’un des deux est mesurée par le taux de classification qui est de 33,48 % et 26,10 % pour le contraste et l’homogénéité respectivement. Donc le modèle obtenu est composé de onze descripteurs : l’énergie, l’entropie, la variance, la corrélation, l’uniformité, le contraste, la direction, la grossièreté, le contraste*, le *line-likeness* et la directivité.

Les différentes mesures de performance de la classification sont récapitulées dans le tableau 5.13. Nous remarquons que le modèle PCFS fournit un taux de classification supérieur et de performances améliorées en comparaison avec le modèle global et les autres algorithmes de sélection.

5.3.7.2 Algorithme PaCFS

Une alternative du PCFS est proposée dans ce paragraphe consiste à tenir compte à la fois et la corrélation entre les descripteurs (PCFS) et leurs corrélation avec la classe d’étiquettes (CFS, MIFS,...). Dans un ensemble corrélé, le descripteur à éliminer est celui qui montre une faible corrélation avec la classe d’étiquettes. Le modèle ainsi obtenu pour nos expérimentations comprend neuf descripteurs qui sont : l’énergie, la variance, la corrélation, l’uniformité, la direction, la grossièreté, le contraste, le *line-likeness* et la directivité.

Le tableau 5.14 donne les différentes mesures de performance de la classification obtenues par le modèle PaCFS. Malgré que le taux de classification obtenu par le modèle PaCFS est assez important,

	T_c (%)	F_m	K	E_t
PaCFS	94,91	0,947	0,945	5,08

Table 5.14 – Performances de classification par l’algorithme PaCFS

Modèle	Paramètres	T_c (%)	F_m	K	E_t (%)	S
Regroupement	$f_1 f_2 f_3 f_4 f_5 f_6 f_7 f_8 f_9 f_{10} f_{11} f_{12}$	95,94	0,958	0,955	4,06	0,149
CFS	$f_2 f_3 f_4 f_5 f_6 f_9 f_{10}$	93,18	0,929	0,926	6,82	0,039
MIFS	$f_4 f_6 f_9 f_{12}$	83,65	0,821	0,832	16,35	0,140
Fisher	$f_1 f_2 f_6 f_7 f_8$	81,13	0,811	0,794	18,86	0,117
ReliefF	$f_2 f_4 f_6 f_7 f_8$	84,35	0,846	0,830	15,64	0,217
AG	$f_1 f_2 f_3 f_5 f_8 f_9 f_{11} f_{12}$	92,98	0,926	0,922	7,01	0,023
PCFS	$f_1 f_2 f_3 f_4 f_5 f_7 f_8 f_9 f_{10} f_{11} f_{12}$	96,07	0,959	0,957	3,93	0,097
PaCFS	$f_1 f_3 f_4 f_5 f_8 f_9 f_{10} f_{11} f_{12}$	94,91	0,947	0,945	5,08	0,032

Table 5.15 – Tableau récapitulatif des performances de classification des différents algorithmes de sélection

il reste légèrement en dessous de celui obtenu par le modèle global. Ainsi, la sélection par l’algorithme PaCFS dans ce cas n’améliore pas forcément la classification de texture.

5.3.8 Comparaison des modèles de sélection

Nous récapitulons les différentes performances de la classification des différents algorithmes de sélection dans le tableau 5.15. Nous constatons que pour la majorité des modèles de sélection (sauf MIFS et ReliefF) il y a une redondance sémantique entre les descripteurs. Par exemple, les descripteurs « énergie » (f_1) et « homogénéité » (f_6) traduisent la même propriété sémantique qui est l’homogénéité des textures. Ainsi, le fait d’avoir ces deux descripteurs dans un même modèle peut entraîner le problème d’effet de poids des descripteurs, ce qui peut aussi être traduit par les forts taux de classification estimés.

Pour résoudre ce problème, nous avons procédé par évaluer le pouvoir discriminatif de chaque descripteur redondant sémantiquement. L’évaluation est basée sur leur pouvoir à distinguer entre classes en terme de taux de classification T_c associé à un taux d’erreur E_t et en terme de recherches d’images similaires évalué par les courbes rappel/précision (cf. figure 5.17) et la mesure du score proposé S . Un ordonnancement par la suite est effectué pour choisir les descripteurs ayant des performances importantes (cf. tableau 5.16).

Ainsi, les meilleurs descripteurs sélectionnés sont : f_2, f_3, f_5, f_6, f_8 . Le descripteur f_3 a été choisi malgré que f_{10} donne un taux de classification légèrement grand du fait que ce dernier donne un score S plus grand, ce qui signifie une mauvaise recherche des images similaires vérifiée aussi par les courbes rappel/précision qui est au dessous de celle de f_3 .

Le tableau 5.17 donne les résultats finaux de chaque modèle après élimination des redondances sémantiques. Le meilleur modèle de sélection est celui qui donne un taux de classification important et sera utilisé dans la suite des expérimentations.

Descripteur	T_c	E_t	S
f_1	25,29	74,70	0,163
f_6	25,61	74,38	0,127
f_2	28,44	71,55	0,301
f_9	22,81	77,20	1,001
f_3	29,40	70,60	0,611
f_7	22,53	77,47	0,205
f_{10}	29,76	70,24	1,100
f_4	19,78	80,21	1,220
f_5	21,53	78,46	0,324
f_8	22,66	77,33	0,205
f_{12}	12,75	87,24	1,018

Table 5.16 – Mesures de performance pour les différents descripteurs de texture

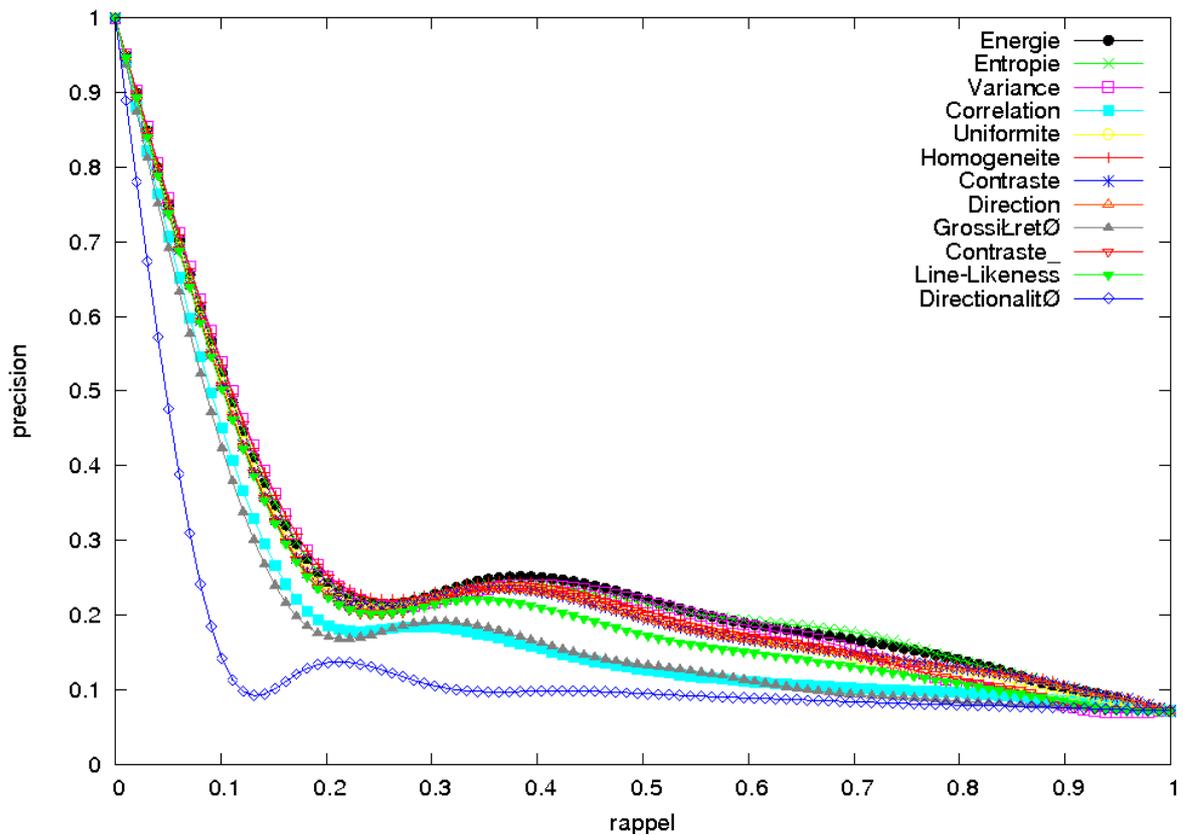


Figure 5.17 – Courbes rappel/précision pour les différents descripteurs de texture

Modèle	Paramètres	T_c (%)	F_m	K	E_t (%)	S
CFS	$f_2 f_3 f_5 f_6$	82,32	0,816	0,806	17,67	0,0960
MIFS	$f_4 f_6 f_9 f_{12}$	83,65	0,821	0,832	16,35	0,140
Fisher	$f_2 f_6 f_7 f_8$	68,12	0,690	0,653	31,87	0,118
ReliefF	$f_2 f_4 f_6 f_7 f_8$	84,35	0,846	0,830	15,64	0,217
AG	$f_1 f_2 f_3 f_5 f_8 f_{11}$	90,51	0,901	0,896	9,48	0,304
PCFS	$f_1 f_2 f_3 f_5 f_8 f_{11}$	90,51	0,901	0,896	9,48	0,304
PaCFS	$f_1 f_3 f_5 f_8 f_9 f_{11}$	92,56	0,922	0,913	7,44	0,005

Table 5.17 – Tableau récapitulatif des performances de classification des différents algorithmes de sélection

	T_c (%)	F_m	K	E_t
k -ppv	88,76	0,871	0,876	11,24

Table 5.18 – Performances de classification par les k plus proches voisins

5.4 Classification de texture

Dans cette section, nous allons aborder la tâche de la classification en étudiant les différentes approches de classification introduites dans la section 4.2 à savoir les k plus proches voisins, les arbres de décision (C4.5), les réseaux bayesiens et les machines à vecteurs de support. Chaque classifieur sera étudié à part puis nous allons faire une comparaison entre les quatre classifieurs avec leurs meilleures performances.

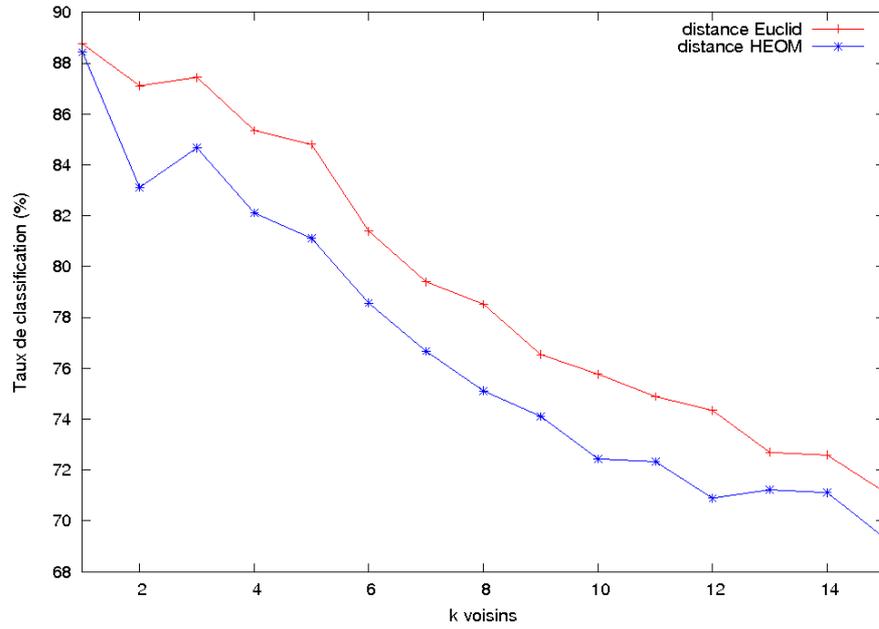
Le modèle de texture sélectionné d'après l'analyse ci-dessus (modèle PaCFS après élimination de redondances) sera utilisé pour la classification des textures. Le modèle de classification qui fournit de meilleurs performances sera utilisé par la suite pour classifier d'autres bases de textures.

5.4.1 k -plus proches voisins

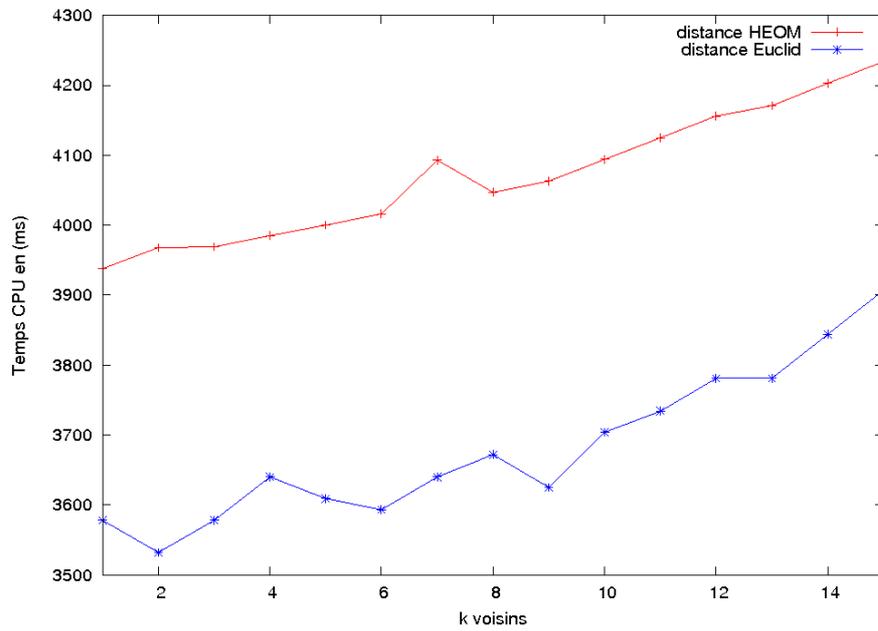
Comme nous l'avons introduit dans la section 4.2.1, le classifieur k plus proche voisins dépend du choix du paramètre k ainsi que la distance utilisée pour l'affectation de nouveaux individus. Dans cette section, nous allons étudier le comportement de la classification par k -ppv pour différentes valeurs de k et suivant deux distances : euclidienne et HEOM. Pour l'évaluation de la classification, nous utiliserons la validation croisée à dix blocs.

La figure 5.18(a)(a) trace le taux de classification du k -ppv pour différentes valeurs de k ainsi que le temps CPU correspondant (cf. figure 5.18(a)(b)).

Nous remarquons d'après les figures que les deux distances ont un comportement similaire pour la classification et le temps CPU. Néanmoins, la distance euclidienne est performante que la distance HEOM puisque le temps CPU de classification correspondant est toujours inférieur à celui de HEOM pour les différentes valeurs de k ainsi qu'une courbe de taux de classification est toujours au dessus. Le meilleur taux de classification (88,76%) est obtenu pour $k = 1$ avec un temps d'exécution de 3,578 s. Le tableau 5.18 donne quelques mesures de performance de classification du k -ppv pour $k = 1$ et une distance euclidienne.



(a) Taux de classification par k -ppv pour différents k



(b) temps CPU correspondant

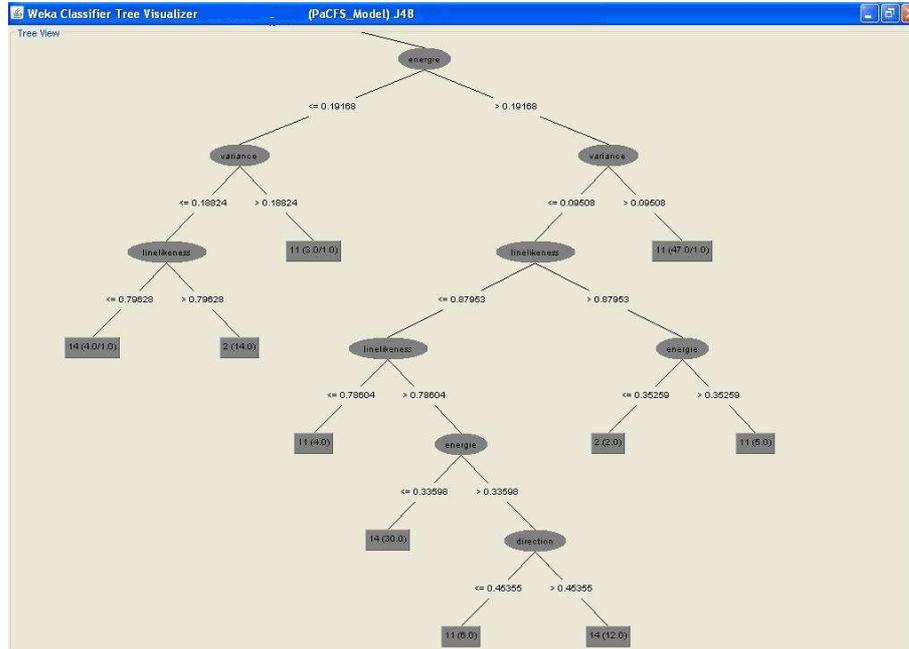


Figure 5.19 – Zoom sur la partie droite de l’arbre de décision obtenue par C4.5

	T_c (%)	F_m	K	E_t
NB	50,77	0,435	0,464	49,22
arbre-NB	79,95	0,777	0,780	20,04

Table 5.20 – Performances de classification par les réseaux bayesiens: NB et arbre-NB

Cette propriété peut être négligée dans le cas où le nombre de données est faible. Cependant, dans un cadre plus large, il faut tenir en compte l’éventuelle dépendance entre paramètres. La structure du réseau bayésien arbre-NB en est une des solutions alternatives qui vérifient cette indépendance sur les feuilles grâce à la décomposition obtenue par l’arbre de décision appliqué sur les nœuds.

Le tableau 5.20 donne les différentes mesures de performance de la classification par le Bayes naïf et l’arbre-NB. Nous constatons une amélioration des résultats de classification par l’arbre-NB comparés à ceux obtenus avec le Bayes naïf.

La figure 5.20⁴ montre la structure obtenue en appliquant l’algorithme de l’arbre-NB sur les différents attributs de texture. Les nœuds en ellipse représentent les différents descripteurs sur lesquels différentes règles de décision sont exécutées (arbre de décision) et les nœuds rectangulaires correspondent aux feuilles qui représentent ici la structure Bayes naïf.

Le tableau 5.21 dresse une comparaison entre le classifieur C4.5 et l’arbre-NB en terme de tailles (nombre de nœuds) et le temps CPU pour la construction. Nous constatons que l’arbre-NB réduit considérablement la taille de l’arbre avec un coût supérieur de construction comparé à l’algorithme C4.5. Pourtant, les performances du arbre-NB et le Bayes naïf restent moins bonne comparées à celles trouvées par les k plus proches voisins et l’arbre de décision C4.5.

⁴figure obtenue par le logiciel de fouille de données Weka.

Noyau	Linéaire	Polynomial	Gaussien	Sigmoïdal
Paramètres	$C = 2^{14}$	$C = 2^{15}, d = 3$	$C = 2^{13}, \gamma = 1$	$C = 2^{-1}, \gamma = 2^{-3}$
T_c	75,43	87,71	93,32	13,28
F_m	0,764	0,873	0,931	0,233
K	0,733	0,866	0,927	0,065
E_t	24,57	12,28	6,67	86,71

Table 5.22 – Performances de classification pour les quatre noyaux SVM avec leurs meilleurs paramètres

5.4.4 Machines à vecteurs de support

Les machines à vecteurs de support sont parmi les classifieurs qui ont montré une grande robustesse et performance. La plus grande difficulté à mettre en œuvre cette approche est le choix du noyau approprié à l'application ainsi que les paramètres associés tels que : le paramètre C , γ et d . D'une manière générale, les données ne sont pas toujours linéairement séparables, ce qui impose de tester les différents noyaux d'une manière empirique. Dans ce travail, nous comparons les quatre noyaux implémentés dans le logiciel *libSVM* [34], à savoir le noyau linéaire, polynomial, gaussien et le noyau sigmoïdal.

Pour choisir les paramètres C , γ et d donnant les meilleures performances, nous avons effectué une recherche par validation croisée en mettant en place pour chaque noyau une grille de couples (C, γ) pour les noyaux gaussien et sigmoïdal et une grille de couples (C, d) pour le noyau polynomial. A chaque itération de la grille, on calcule le taux de classification correspondant au couple (C, γ) et (C, d) . Le couple qui donne le meilleur taux de classification pour le noyau correspondant définira les paramètres du noyau SVM. Pour effectuer cette recherche, nous avons fixé les intervalles de valeurs pour chaque paramètre. Nous avons considéré l'intervalle $[2^{-15}, 2^3]$ pour γ , l'intervalle $[3, 10]$ pour d et pour C l'intervalle $[2^{-5}, 2^{15}]$.

Les différentes mesures de performance de la classification associées aux meilleurs paramètres donnant des taux de classification importants sont présentés dans le tableau 5.22. Une première constatation de ce tableau est que le noyau sigmoïdal donne de mauvaises performances par rapport aux autres noyaux. Le meilleur taux de classification est obtenu pour le noyau gaussien avec $C = 2^{13}$ et $\gamma = 1$. Ce taux de 93,32 % est supérieur de celui trouvé par $C = 10^6$ et $\gamma = 0,75$ (92,56%, cf. tableau 5.17).

5.4.4.1 Comparaison entre les classifieurs

Dans ce paragraphe, nous allons donner un récapitulatif et une comparaison entre les différents classifieurs étudiés en haut (k plus proches voisins, l'arbre de décision C4.5, arbre-NB et SVM) en tenant compte que de leurs meilleurs paramètres :

1. les k plus proches voisins avec $k = 1$;
2. l'arbre de décision C4.5, ne nécessite aucun paramètre ;
3. l'arbre-NB pour les réseaux bayesiens ;
4. les SVM avec $C = 2^{13}$ et $\gamma = 1$ pour le noyau gaussien.

Le tableau 5.23 dresse les différentes mesures de performance de la classification pour les quatre classifieurs choisis. Nous remarquons que les SVM à noyau gaussien donnent le meilleur taux de classification (93,32%) que ceux obtenus par k plus proches voisins (88,76%), l'arbre de décision C4.5 (81,50%) et les réseaux bayesiens avec l'arbre-NB (79,95%).

	T_c (%)	F_m	K	E_t
k -ppv	88,76	0,871	0,876	11,24
C4.5	81,50	0,798	0,797	18,41
arbre-NB	79,95	0,777	0,780	20,04
SVM	93,32	0,931	0,927	6,67

Table 5.23 – Comparaison entre les différents classifieurs avec leurs meilleurs paramètres

Classifieur	k -ppv	C4.5	arbre-NB	SVM
Temps CPU (s)	0,02	0,19	4,3	3,5

Table 5.24 – Temps d'exécution pour les différents classifieurs

Ainsi, nous concluons que les SVM donnent une bonne classification des différentes classes de textures avec une faible valeur de confusion entre les classes suivies des k plus proches voisins qui donnent un taux de classification acceptable avec un léger taux d'erreur. Pour le C4.5 et l'arbre-NB leurs performances sont similaires et reflètent une moyenne détection avec quelques confusions entre classes.

Le tableau 5.24 donne le temps CPU d'exécution de chaque classifieur. On constate que le k plus proche voisin est le plus faible en temps de calcul puisqu'il ne nécessite pas de phase d'apprentissage (recherche directe sur la base des k plus proches) contrairement aux autres classifieurs. l'arbre-NB nécessite plus de temps pour la construction de l'arbre que le C4.5 malgré qu'il soit plus réduit (cf. tableau 5.21). Pour les SVM, elles viennent en second lieu après l'arbre-NB avec un temps de calcul de 3,5 secondes.

5.5 Applications

Dans cette partie, nous allons valider le modèle de texture que nous avons proposé dans la section 5.3.8 sur de nouvelles bases de texture en évaluant les résultats en terme de la recherche d'images similaires et en terme de classification par les quatre classifieurs choisis en haut (les k plus proches voisins, le C4.5, l'arbre-NB et les SVM). Nous présenterons d'abord les différentes bases utilisées, ensuite nous donnerons les différents résultats obtenus.

5.5.1 Bases de texture

Nous avons utilisé deux bases de texture pour les tests expérimentaux suivants. La première base est un échantillon de 113 textures de la base de texture de l'université d'Oulu⁵. Chaque image monochrome est de taille 256×256 pixels avec une illumination incandescente constante et une résolution de 100dpi (cf. figure 5.23). Cette base comprend trois classes que nous avons pu identifier visuellement. Le tableau 5.25 donne le nombre de textures dans chaque classe trouvée. La seconde base est la base Vistex⁶ composée de 167 textures en couleurs de taille 128×128 en format ppm. D'après le tableau 5.26, on a constaté qu'il existe des textures déjà étudiées dans la phase d'apprentissage (bark, brick, grass, misc, sand, water, wood) et aussi des classes contenant un nombre faible de textures ou bien des images composées de plusieurs textures qui n'entrent pas de le cadre de cette étude puisqu'elles nécessitent des techniques de segmentation pour pouvoir identifier les différentes textures (cf. figure 5.22). Ces textures seront éliminées de nos expérimentations.

⁵La base est disponible sur le site http://195.221.38.30/LVR/SIV/interpretation/evaluation/images_segmentation_f

⁶Cette base peut être téléchargée à partir du site <http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.ht>

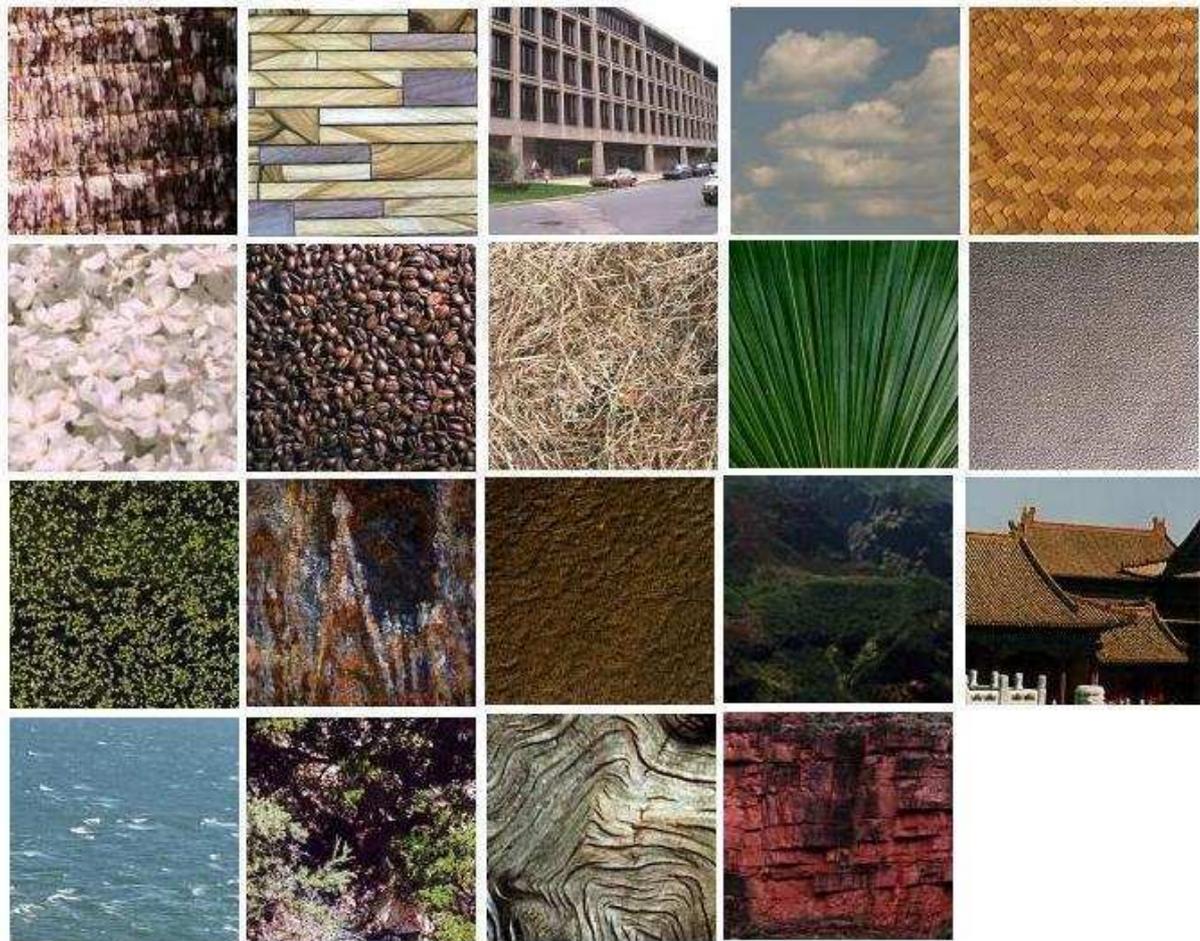


Figure 5.22 – Quelques exemples de textures de la base Vistex

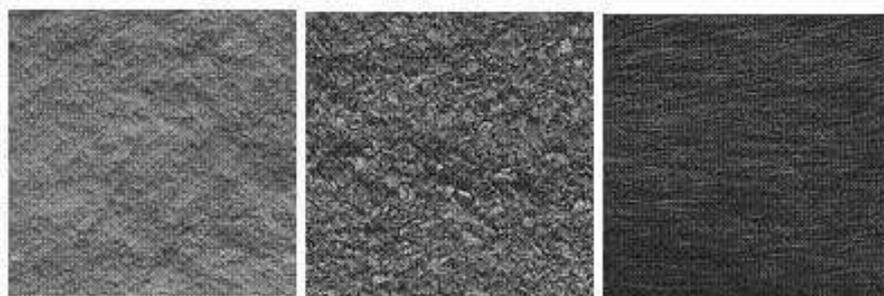


Figure 5.23 – Exemples de textures des trois classes la base Oulu

Classe	# textures
Classe 1	45
Classe 2	35
Classe 3	33

Table 5.25 – Classification de textures dans la base Oulu

Classe	# textures	Classe	# textures
Bark	13	Brick	9
Buildings	11	Clouds	2
Fabrics	20	Flowers	8
Food	12	Grass	3
Leaves	17	Metal	6
Misc	4	Paintings	13
Sand	7	Stone	6
Terrain	11	Tile	11
Water	8	Waldo	3
Wood	3		

Table 5.26 – Classification de textures dans la base Vistex

Après cette présentation des deux bases de texture, nous allons procéder aux tests, premièrement par la recherche d'images similaires (section 5.5.2), ensuite par la classification (section 5.5.3) en se basant sur le modèle de texture sélectionné dans la section 5.3.8 composé de : l'énergie, la variance, l'uniformité, la direction, la grossièreté et le *line-likeness*.

5.5.2 Recherche d'images similaires

la recherche d'images consiste à parcourir chaque image de la base et chercher celles qui sont les plus proches visuellement. L'évaluation de la recherche est faite par les courbes de rappel/précision et le score introduits dans la section 4.1.

La figure 5.24 trace les courbes de rappel/précision obtenues pour la base Vistex et Oulu. Nous remarquons que la courbe correspondante à la base Vistex est au dessus de celle de la base Oulu notamment pour les 50 % premières de la recherche et par la suite elle se dégrade. Cela montre que le système arrive à trouver les images les plus similaires pour la base Vistex dans les toutes premières recherches. A l'inverse, pour la base Oulu, il est difficile de trouver les images similaires qu'après avoir pris en compte un nombre important de la base, ce qui reflète une incapacité de détection de cette base. Ces conclusions sont vérifiées par les valeurs des scores que nous avons trouvé : -0,34 pour la base Oulu contre 2,19 pour la base Vistex. Une valeur négative du score signifie de fausses détection lors de la recherche.

5.5.3 Classification

Pour la classification, les tests seront effectués sur les quatre classifieurs :

1. les k plus proches voisins avec $k = 1$;
2. l'arbre de décision C4.5;
3. le réseau bayésien avec l'arbre-NB;
4. les SVM à noyau gaussien avec $C = 2^13$ et $\gamma = 1$.

Les résultats de la classification pour les deux bases sont donnés respectivement dans le tableau 5.27 et 5.28.

Concernant les résultats obtenus pour la classe Oulu, nous avons obtenu un meilleur taux de classification de 99,11 % avec les k plus proches voisins où les trois classes de la base sont parfaitement classifiées. L'arbre de décision C4.5 donne de performances satisfaisantes (77,86 %) proches de celles

Classifieur	MC (%)	E_m (%)	T_c (%)	F_m	K	E_t (%)	Temps CPU (s)
k -ppv	$\begin{pmatrix} 100,00 & 0,00 & 0,00 \\ 0,00 & 100,00 & 0,00 \\ 0,00 & 3,03 & 96,97 \end{pmatrix}$	$\begin{pmatrix} 100,00 \\ 100,00 \\ 3,03 \end{pmatrix}$	99,11	0,991	0,987	1,01	0
C4.5	$\begin{pmatrix} 97,78 & 0,00 & 2,22 \\ 45,71 & 40,00 & 14,29 \\ 9,09 & 0,00 & 90,91 \end{pmatrix}$	$\begin{pmatrix} 2,22 \\ 60,00 \\ 9,09 \end{pmatrix}$	77,86	0,755	0,658	23,77	0,08
arbre-NB	$\begin{pmatrix} 80,00 & 17,78 & 2,22 \\ 14,28 & 74,28 & 11,42 \\ 9,09 & 12,12 & 78,79 \end{pmatrix}$	$\begin{pmatrix} 20,00 \\ 25,71 \\ 21,21 \end{pmatrix}$	77,69	0,780	0,665	22,31	0,36
SVM	$\begin{pmatrix} 86,67 & 6,67 & 6,67 \\ 37,14 & 48,57 & 14,28 \\ 3,03 & 63,64 & 33,33 \end{pmatrix}$	$\begin{pmatrix} 13,33 \\ 51,43 \\ 66,67 \end{pmatrix}$	59,29	0,621	0,375	43,81	0,31

Table 5.27 – Performances de classification pour la base Oulu

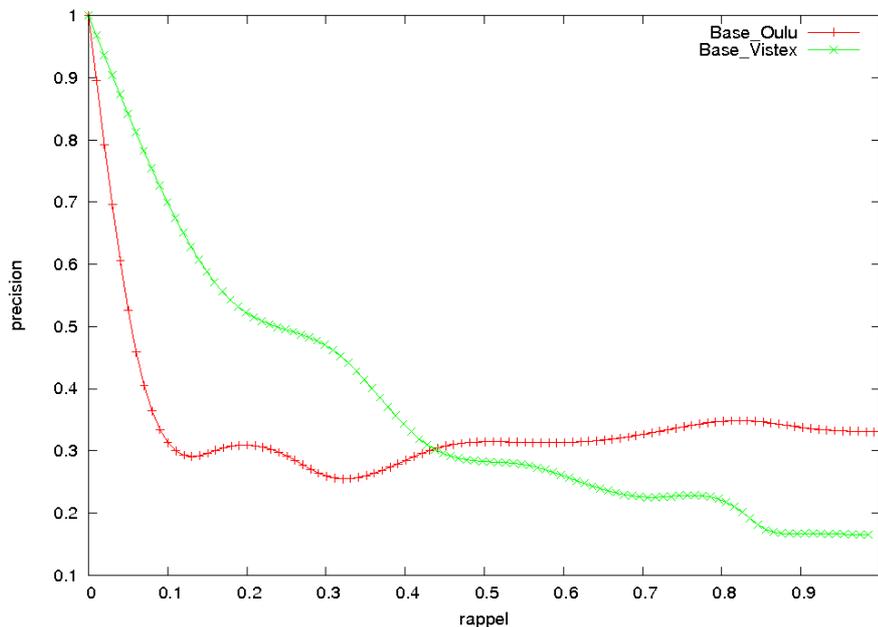


Figure 5.24 – Courbes de rappel/précision pour la base Vistex et Oulu

obtenues par l'arbre-NB (77,69 %) malgré la confusion de la classe 2 avec la classe 1. L'arbre-NB améliore la détection de la classe 2 mais se dégrade légèrement pour la classe 1 et la classe 3. Pour les SVM, elles donnent ici de modestes performances (59,29 %) pour la classification des textures. Seule la classe 1 est bien classifiée, pour les deux autres, elles sont confondues avec toutes les classes. Cette mauvaise détection peut être justifiée par le nombre réduit de données appris par les SVM qui demandent un nombre important de données.

Pour la base Vistex, il s'agit des mêmes remarques faites pour la base Oulu :

- Le classifieur k plus proche voisin donne une parfaite performance de 100 %.
- L'arbre de décision C4.5 aussi donne de bonnes performances : 86,42 %. Seule la classe « paintings » est moyennement détectée (69,23 %).
- Pour l'arbre-NB et les SVM, ils donnent de modestes performances : 69,13 % et 62,96 % respectivement. Les classes « fabrics » et « terrain » sont parfaitement identifiées par l'arbre-NB et les SVM. La classe « flowers » est mal classifiée par les SVM contrairement à l'arbre-NB. Une légère amélioration de la classe « paintings » est trouvée par les SVM. Pour le reste des classes sont confondues entre elles.

En comparant les deux bases de texture Oulu et Vistex simultanément, nous constatons que les performances de la base Vistex sont toujours supérieures à celles obtenues pour la base Oulu. Cela provient de la diversité de la base Vistex par rapport à la base Oulu.

5.6 Conclusion

Dans ce chapitre, notre but est d'identifier un modèle de texture pertinent qui caractérise au mieux différentes classes de textures. Pour cela, nous avons établi un processus de description se déroulant en plusieurs étapes. En premier lieu, il s'agit de la phase d'extraction de texture par les matrices de co-

Classifieur	MC (%)	E_m (%)	T_c (%)	F_m	K	E_t (%)	Temps CPU (s)
k -ppv	$\begin{pmatrix} 100,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 \\ 0,00 & 100,00 & 0,00 & 0,00 & 0,00 & 0,00 \\ 0,00 & 0,00 & 100,00 & 0,00 & 0,00 & 0,00 \\ 0,00 & 0,00 & 0,00 & 100,00 & 0,00 & 0,00 \\ 0,00 & 0,00 & 0,00 & 0,00 & 100,00 & 0,00 \\ 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 100,00 \end{pmatrix}$	$\begin{pmatrix} 0,00 \\ 0,00 \\ 0,00 \\ 0,00 \\ 0,00 \\ 0,00 \end{pmatrix}$	100,00	1	1	0,00	0
C4.5	$\begin{pmatrix} 100,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 \\ 12,50 & 87,50 & 0,00 & 0,00 & 0,00 & 0,00 \\ 8,30 & 0,00 & 83,33 & 8,3 & 0,00 & 0,00 \\ 0,00 & 5,88 & 5,88 & 82,35 & 0,00 & 5,88 \\ 0,00 & 15,38 & 0,00 & 15,38 & 69,23 & 0,00 \\ 9,09 & 0,00 & 0,00 & 0,00 & 0,00 & 90,09 \end{pmatrix}$	$\begin{pmatrix} 0,00 \\ 12,50 \\ 16,67 \\ 17,65 \\ 30,77 \\ 9,09 \end{pmatrix}$	86,42	0,863	0,834	14,45	0,02
arbre-NB	$\begin{pmatrix} 100,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 \\ 0,00 & 100,00 & 0,00 & 0,00 & 0,00 & 0,00 \\ 83,33 & 0,00 & 16,67 & 0,00 & 0,00 & 0,00 \\ 52,94 & 5,88 & 0,00 & 35,29 & 0,00 & 5,88 \\ 15,38 & 15,38 & 0,00 & 0,00 & 69,23 & 0,00 \\ 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 100,00 \end{pmatrix}$	$\begin{pmatrix} 0,00 \\ 0,00 \\ 83,33 \\ 64,71 \\ 30,77 \\ 0,00 \end{pmatrix}$	69,13	0,617	0,616	29,80	0,3
SVM	$\begin{pmatrix} 95,00 & 0,00 & 0,00 & 5,00 & 0,00 & 0,00 \\ 25,00 & 0,00 & 0,00 & 0,00 & 75,00 & 0,00 \\ 33,00 & 0,00 & 16,67 & 41,67 & 8,33 & 0,00 \\ 35,29 & 0,00 & 0,00 & 47,06 & 11,67 & 5,88 \\ 7,69 & 0,00 & 0,00 & 7,69 & 84,61 & 0,00 \\ 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 100,00 \end{pmatrix}$	$\begin{pmatrix} 5,00 \\ 100,00 \\ 83,00 \\ 52,94 \\ 15,29 \\ 0,00 \end{pmatrix}$	62,96	0,619	0,537	42,77	0,3

Table 5.28 – Performances de classification pour la base Vistex

occurrences et la technique de Tamura. Ensuite, une phase d'exploration des différents techniques de sélection à la fois pour la réduction de l'espace de caractérisation de texture et la sélection des attributs pertinents.

L'identification du modèle pertinent est analysée en terme de la recherche d'images similaires et en terme de la classification. Les résultats obtenus à cette étape ont révélé la forte liaison des attributs de texture choisis avec la perception humaine. Pour de bases fortement homogènes, de fortes confusions apparaissent entre les classes. Contrairement aux bases diversifiées, les résultats sont très satisfaisants.

Pour la phase de validation, les résultats obtenus sur des bases externes sont encourageants. Cependant, il faut disposer d'une certaine diversité des textures étudiées.

PARTIE II

Navigation sémantique dans des bases d'images

Interprétation sémantique des textures

Le domaine de la recherche d'images par contenu basée sur les descripteurs de bas niveau s'accroît de jour en jour au vu du large volume d'images nécessitant de mettre à jour les différentes approches de recherche et d'analyse d'images. Néanmoins, depuis une décennie, ces systèmes de recherche affrontent leurs incapacités et leurs insuffisances d'exprimer la sémantique de la requête des utilisateurs. Cette insuffisance est aperçue par le décalage entre cette dernière et les résultats retournés. Ces échecs sont liés d'une part, à la difficulté d'extraction et de sélection des attributs de bas niveau pertinents pour la description des images (*cf.* chapitre 5) et d'autre part à la difficulté à traduire en terme de ces descripteurs de bas niveau le contenu sémantique des images, propriété subjective de la requête. Ce décalage entre les descripteurs de bas niveau et les concepts de haut niveau exprimés par la requête utilisateur est connu sous le nom *fossé sémantique* (*semantic gap*).

Pour les bases d'images liées à un domaine spécifique, par exemple les bases de visages, d'images médicales, etc., les connaissances *a priori* du domaine restreint la subjectivité de la requête. Il est donc possible d'extraire et de modéliser l'information sémantique pertinente pour l'application. Ainsi, le fossé sémantique est réduit.

Dans le cadre des d'images généralistes, ce qui est le cas pour notre thèse, aucune connaissance *a priori* n'est fournie et la recherche dépend fortement des utilisateurs, ainsi la subjectivité des requêtes est maximale. De plus, les attributs de bas niveau de description des images, ici les attributs de texture, demeurent génériques. Sous ces contraintes, le fossé sémantique se montre important. Cependant, une modélisation de ces attributs de bas niveau en attributs de haut niveau sémantiques apparaît comme une solution pour réduire ce fossé.

La suite du chapitre est organisée comme suit. Après une introduction générale des différentes approches de réduction du fossé sémantique (section 6.1), nous allons présenter dans la section 6.2 les travaux les plus récents sur la texture en particulier. Notre démarche sera détaillée dans la section 6.3.

6.1 Introduction

Nous avons présenté les différents niveaux d'abstraction des requêtes des utilisateurs ainsi que la notion du fossé sémantique qui constitue un défi majeur pour les systèmes de recherche d'images dans l'introduction (*cf.* section 1.2). Dans cette partie, nous allons dresser d'une manière non-exhaustive les différentes approches ayant pour but la réduction de ce fossé. Ces approches se regroupent en cinq grandes catégories [189] :

1. les ontologies pour définir les concepts de haut niveau à partir des connaissances du domaine [203, 202, 47, 284, 213, 242] ;

2. la recherche interactive [294, 99, 147, 213, 328] qui tient en compte l'intention de l'utilisateur d'une manière continue ;
3. les techniques d'apprentissage qui associent les attributs de bas niveau aux concepts traduits par les requêtes [188, 75, 39, 265, 299, 293] ;
4. les modèles sémantiques pour une recherche d'images à haut niveau [329, 36, 334, 279] ;
5. la fusion des informations obtenues à partir du contenu visuel des images et les annotations textuelles provenant du Web [297, 24, 128, 76, 30, 334].

Nous allons traiter dans la suite que les trois premières catégories.

6.1.1 Définition des concepts par les ontologies

« Une ontologie est une spécification formelle et explicite d'une conceptualisation partagée d'un domaine de connaissances » [131, 203] :

- *conceptualisation* : une certaine vue du monde par rapport à un domaine, souvent conçue comme un ensemble de concepts, leur définition, leurs inter-relations. En général, c'est le choix quand à la manière de décrire un domaine.
- *explicite* : définition explicite des types de concepts utilisés et des contraintes sur leur usage.
- *formelle* : compréhensible par la machine.
- *partagée* : consensus, connaissances acceptées par un groupe.

Autrement dit, une ontologie est un ensemble de descripteurs intermédiaires formant un vocabulaire spécifique pour un domaine donné qui peut inclure des définitions de classes, des relations, des fonctions et d'autres objets [213].

Dans une représentation par ontologies, le passage d'une description d'images de bas niveau au niveau sémantique plus haut se fait par quantification des descripteurs de bas niveau en différents intervalles où chaque intervalle correspond à un descripteur intermédiaire formant un objet ontologie. Ce descripteur intermédiaire fournit la description sémantique des concepts. Par exemple, pour l'objet « ciel » dans une image, il peut être défini d'une manière naturelle par une « région homogène, supérieure et bleue ». En utilisant les ontologies pour la définition de concepts, le ciel sera exprimé par les termes suivants : « texture homogène », « couleur bleu clair » et « zone supérieure » (location spatiale).

Un exemple typique d'une ontologie est donné par Mezaris et *al.* dans [213]. Chaque image du système est décrite par sa couleur moyenne calculée dans l'espace de couleur Lab, par sa position dans l'espace (x,y) (x, axe horizontal et y, axe vertical), par sa taille et par sa forme (*cf.* figure 6.1).

La phase de quantification est une phase primordiale dans les systèmes supportant les ontologies. Par exemple, la méthode de quantification du descripteur « couleur » largement utilisée est le « nommage de la couleur »¹. L'approche consiste à relier les valeurs numériques de l'espace de représentation de la couleur aux termes sémantiques définis dans le langage naturel. Bien qu'un système informatique peut définir des millions de couleur, un utilisateur ne peut définir qu'un nombre limité entre 10 et 20 termes [13, 14, 52]. Le système le plus connu pour le nommage de la couleur est le système « CNS » (Color Naming System) proposé par Berk et *al.* dans [13, 14]. Le système CNS quantifie l'espace couleur TSL (Teinte, Saturation et Luminance) en 627 couleurs distinctes. L'idée de base est de quantifier l'espace teinte en couleurs de base. L'espace saturation et luminance seront quantifiés en termes de la richesse et la luminosité de la couleur. L'ensemble généré par le système CNS est de 10 termes : *orange, rouge, bleu, jaune, vert, marron, pourpre, noir, gris et blanc.*

¹en anglais, color naming.

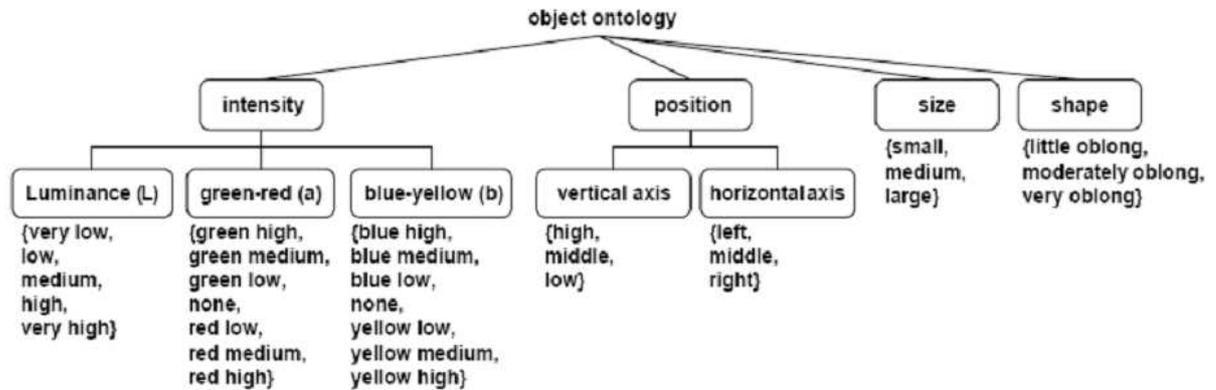


Figure 6.1 – Description d’un objet ontologie [213]

Dans [242], les auteurs constatent qu’un ensemble réduit de couleur est suffisant pour caractériser la couleur d’une région. Ils définissent huit couleurs exprimées dans l’espace RGB : *rouge*, *bleu*, *vert*, *jaune*, *magenta*, *cyan*, *noir* et *blanc*. Les couleurs sont ensuite associées aux objets des images, par exemple, le blanc correspond à la neige, les nuages, etc. De même dans [190], les auteurs utilisent la couleur dominante d’une région exprimée dans l’espace TSL qui est convertie en 35 sémantiques termes tels que : rouge, bleu, vert, etc. Chaque terme sémantique par la suite est attribué à un objet dans une image naturelle. Le système ainsi peut effectuer des requêtes de type « trouver les images avec ciel bleu ». De cette manière, le système supporte des requêtes de haut niveau basées sur les mots-clés ainsi réduisant le fossé sémantique entre les descripteurs de bas niveau de la couleur et la sémantique des requêtes.

Dans la même voie, les auteurs ont essayé de définir un système similaire à CNS pour la texture (TNS) afin de normaliser sa description et sa représentation [241]. Or, cette tâche s’est avérée plus difficile que la couleur et aucun système n’a pu être généré. Dans [241, 290], les auteurs tentent d’identifier les attributs les plus importants utilisés pour exprimer une texture. La *répétition*, la *direction* et la *complexité* sont les trois attributs les plus importants que les auteurs ont pu identifier [241]. Les problèmes récurrents concerne l’extraction des attributs de textures et leurs traduction dans un espace sémantique [178, 241]. Cela provient de la difficulté de définir et de modéliser la texture elle-même (*cf.* chapitre 3). Une solution alternative consiste à quantifier les descripteurs de texture en intervalles de perception différents. Dans [166, 46, 47], les descripteurs de Tamura sont quantifiés en utilisant un algorithme de regroupement flou en cinq intervalles différents. Par exemple, pour le contraste, il est exprimé en : trop faible, faible, moyen, fort et très fort. Pour la grossièreté, elle est : trop lisse, lisse, moyennement grossière, grossière et très grossière, etc. Les requêtes dans ce cas sont exprimées par des combinaisons logiques (et/ou) des différentes descriptions sémantiques : « Trouver les textures lisses et faiblement contrastées. »

Ce type de descriptions sémantique basée sur les ontologies fonctionnent correctement pour des bases spécifiques. Cependant, leurs usage est limité dans un cadre général avec de larges bases d’images et divers contenus sémantiques ce qui nécessite d’introduire des outils d’apprentissage pour exprimer la sémantique.

6.1.2 Apprentissage

Lors d’un processus d’identification de la sémantique, la question qui se pose est : comment connecter la sémantique exprimée par les concepts haut niveau aux caractéristiques de l’image traduites par les attributs de bas niveau. La majorité des travaux se reposent sur les techniques d’apprentissage su-

pervisé et non-supervisé [188, 75, 39, 64, 312]. Dans le cas supervisé, l'information résultat à prédire (classes sémantiques) est connue *a priori*. Dans le cas non-supervisé, aucune information n'est connue. Il s'agit donc d'identifier les classes sémantiques à partir l'étude de l'organisation des données en sortie [135, 39, 118].

6.1.2.1 Apprentissage supervisé

Pour connecter les concepts sémantiques aux caractéristiques des images, l'apprentissage supervisé est une tendance populaire et efficace [188, 64]. Son principe est d'apprendre les concepts sémantiques à partir des attributs de bas niveau constituant les entrées de l'algorithme de classification supervisé. Parmi les classifieurs les plus utilisés, on trouve les machines à vecteurs de support [269, 75, 304], les réseaux bayésiens [197, 84, 146, 285, 307] (*cf.* section 4.2) et les réseaux de neurones [185, 293]. Par exemple, dans [269], les SVMs sont utilisées dans le cadre de l'annotation des images. Lors de la phase d'apprentissage, le SVM binaire est conçu pour apprendre 23 concepts choisis. Par la suite, différentes régions non-étiquetées sont intégrés dans tous les modèles. Le concept, lié à un modèle, qui fournit la plus grande valeur est associée à la région.

Un autre exemple utilisant le classifieur bayésien est donné dans [299, 195]. Les auteurs utilisent le modèle bayésien binaire pour identifier les concepts sémantiques des scènes naturelles à partir des descripteurs de bas niveau. En conséquence, la base d'images est classifiée automatiquement en différents concepts généraux tels que scènes d'intérieur et d'extérieur² qui sont eux même classifiés en d'autres sous-concepts ; par exemple, une image d'extérieur peut correspondre à une ville, un paysage, etc.

Dans [293], les auteurs définissent 11 catégories correspondant aux concepts de haut niveau : *brique, nuage, fourrure, herbe, glace, route, roche, sable, peau, arbres* et l'*eau*. Ensuite, un ensemble de descripteurs de bas niveau, extraits des régions segmentées, sont intégrés dans les réseaux de neurones pour établir un lien entre ces descripteurs et leurs sémantiques. Les inconvénients de cette approche résident dans le volume de données nécessaire pour l'apprentissage, la structure pertinente des réseaux de neurones à choisir et le temps de calcul.

Malgré le grand succès que connaît les méthodes d'apprentissage supervisé pour la réduction du fossé sémantique entre les descripteurs de bas niveau et les concepts sémantiques, deux grands problèmes [75] restent récurrents à savoir : la taille et le choix de la base d'exemples pour apprendre les caractéristiques. Un nombre réduit de données est insuffisant pour apprendre une large variabilité de caractéristiques et un nombre assez important sera fastidieux dans la phase d'étiquetage. Pour le choix de la base, il est fixé pour un domaine spécifique, si ce dernier change, de nouveaux concepts apparaissent et doivent être appris. Une solution basée sur l'approche bootstrap³ est apportée dans [75], les auteurs commencent par apprendre un ensemble réduit de concepts. Ensuite, en utilisant une approche coopérative d'apprentissage effectuée par deux classifieurs indépendants statistiquement, un pour l'apprentissage des exemples non-étiquetés et l'autre pour leurs annotation, l'algorithme d'une manière itérative annote un ensemble de plus en plus large des exemples non-étiquetés. Les expérimentations montrent une amélioration du taux de recherche de 10% comparé au SVM (400 exemples étiquetés sont utilisés pour l'apprentissage) avec seulement 20 exemples étiquetés.

²en anglais, les termes sont connus par *outdoor/indoor scenes*

³Technique permettant d'effectuer de l'inférence statistique sur un nombre limité d'observations en procédant pour chaque échantillon à un tirage aléatoire de N éléments avec remise.

6.1.2.2 Apprentissage non-supervisé

À la différence du cas supervisé, l'apprentissage non-supervisé ne possède aucune information *a priori* sur les différentes classes sémantiques. Son objectif est de regrouper les images partageant certaines caractéristiques, en termes d'une mesure de similarité, en différentes classes (ou clusters) qui vont définir par la suite les différents concepts sémantiques.

Les algorithmes les plus répandus et largement utilisés pour les applications de classification non-supervisée sont les algorithmes de la famille des nuées dynamiques (*k-means* et ses variantes) [135].

Dans [283], l'algorithme *k-means* est utilisé pour la classification de la couleur. En se basant sur des statistiques mesurant le degré de variation dans chaque cluster, un ensemble de descripteurs intermédiaires est identifié entre les attributs de bas niveau et la caractérisation textuelle optimale des classes. Dans [146], pour l'annotation automatique de la base d'images, le système procède dans un premier temps à classifier les images de la base par une variante du *k-means* appelée *pair-wise constraints (PCK)* *k-means* [21]. Le nombre de classes est fixé empiriquement à 300. Ensuite, pour chaque région la probabilité *à posteriori* de chaque concept (59 concepts sont identifiés pour la base d'images utilisée) est estimée en utilisant le classifieur bayésien. Une nouvelle image est annotée par les concepts ayant une probabilité importante. De cette manière, une recherche sémantique est fort possible.

Dans [41, 40], les auteurs définissent une nouvelle méthode « CLUE » pour la réduction du fossé sémantique basée sur l'approche spectrale NCut⁴ [268] de classification non-supervisé. Le système proposé commence par chercher l'ensemble des images les plus proches à l'image requête. Ensuite, l'algorithme NCut est appliqué sur cet ensemble afin d'identifier les différentes classes sémantiques qui seront affichés à l'utilisateur. Le système supporte un module de bouclage qui permet d'ajuster les similarités pour affiner la recherche. L'inconvénient de cette approche est que l'algorithme NCut ne permet pas d'avoir une fonction explicite du passage du bas niveau aux concepts sémantiques, ce qui rend la tâche de classification de nouvelles images plus difficile. Pour pallier à cet inconvénient, Zheng et *al.* proposent dans [330] la méthode LPC (Locality Preserving Clustering) qui permet de définir explicitement une fonction de *mapping*⁵. Les expérimentations montrent de comparables performances pour le LPC et NCut. Pourtant le LPC reste efficace en calcul.

6.1.3 Recherche interactive

En considérant une seule requête, le système ne peut pas comprendre les objectifs et les intentions de l'utilisateur parce que la sémantique de l'image change en changeant le contexte de la recherche. Dans [257], Santini et *al.* donnent un bon exemple de cela, la même image (peinture d'un visage) utilisée par le même utilisateur dans deux contextes différents possède deux sémantiques différentes : l'une est une peinture visant à chercher d'autres peintures (avec et sans visages), l'autre est un visage visant à chercher d'autres visages (peints ou non). Pour pallier à ce problème, quelques systèmes intègrent des interfaces où les utilisateurs peuvent interagir avec le système. Par ces interactions, le système apprend en fonction des requêtes les intentions de l'utilisateur et lui rend des résultats qui peuvent le satisfaire [145, 251, 253]. Cette recherche est dite recherche interactive ou recherche par rétroaction ou aussi recherche par boucle de pertinence (en anglais *relevance feedback RF*).

Le scénario standard d'une recherche interactive (RI) est comme suit[332] :

1. le système retourne à l'utilisateur les premiers résultats correspondants à sa requête ;

⁴Normalized cut

⁵le passage du bas niveau de caractéristiques des images au niveau haut des concepts.

2. l'utilisateur ensuite juge de la pertinence ou non des images retournées en fonction de sa satisfaction ;
3. l'algorithme d'apprentissage est ensuite appliqué afin d'apprendre l'interaction de l'utilisateur et retourne à (2).

Les étapes (2) et (3) sont répétées jusqu'à ce que l'utilisateur soit parfaitement satisfait des résultats retournés. À l'étape (3), les poids des différents descripteurs de bas niveau sont ajustés en fonction des jugements de l'utilisateur [147, 253]. Avec ces ajustements, les concepts sémantiques sont définis. Plusieurs algorithmes d'apprentissage vus dans la section 6.1.2 peuvent être intégrés à l'étape 3 de la recherche interactive pour l'ajustement des poids.

Parmi les systèmes basés sur la RI, le système « iFind » [194] qui effectue la rétroaction pour les attributs de bas niveau et aussi le contenu sémantique des images exprimé par des mots-clés. Premièrement, un réseau sémantique est construit au-dessus de la base d'images. Ensuite, un algorithme d'apprentissage est appliqué pour améliorer ce réseau au fur et à mesure des itérations. Par conséquent, l'ajustement est effectué au niveau des attributs de bas niveau et de la sémantique en même temps. Les expérimentations sur des images réelles montre l'efficacité et l'importance de cette approche.

6.2 Recherche sémantique de la texture

Pour la réduction du fossé sémantique dans le cas de recherche d'images basée sur la texture seule, la majorité des travaux se focalisent sur l'utilisation des attributs de Tamura [29, 290] pour ces différentes caractéristiques proches de la perception humaine (*cf.* section 3.3.2) couplés avec l'algorithme de classification non-supervisée flou FCM (fuzzy C-Means). Liu *et al.*, Kulkarni et Verma respectivement dans [186, 167, 166] proposent un système de recherche d'images supportant des requêtes utilisateurs floues. Un schéma simplifié du fonctionnement de ce système est donné par la figure 6.2. Après l'extraction des attributs de texture par la méthode Tamura, l'algorithme FCM est appliqué. Il permet de traduire chaque descripteur de texture en cinq termes linguistiques ; par exemple, l'ensemble de termes linguistiques associé à l'attribut contraste sont : trop faible, faible, contraste moyen, contraste normale et fort contraste. De la même manière les autres attributs sont traduits. Par conséquent, le fossé sémantique est réduit en permettant aux utilisateurs d'effectuer des requêtes en langage naturel. Une requête est une combinaison logique des différents termes sémantiques exprimée par les opérateurs ET (\wedge), OU (\vee) et NON (\neg) ; par exemple : « trouver les textures lisses ou régulières et fort contrastées ».

Li *et al.* dans [184] proposent une autre approche d'extraction de la sémantique à partir des descripteurs de texture basée sur la programmation génétique [234, 7]. Les auteurs introduisent la notion de *variable linguistique* pour une description globale de la sémantique des textures en se basant toujours sur le modèle Tamura. Ensuite, pour établir les liens entre la description de bas niveau et sa sémantique, les auteurs appliquent la programmation génétique pour simuler la perception humaine et par la suite déduire les différentes règles de correspondance sémantiques. Les requêtes sont par la suite effectuées par des combinaisons logiques en langage naturel.

Définition 1. Une variable linguistique est un quintuplet $(x, T(x), U, G, M)$ où :

- x est le nom de la variable, par exemple « contraste », « grossièreté », etc. ;
- $T(x)$ est l'ensemble de termes linguistiques associés à x (ex. : pour la grossièreté, $T(x)$ sera formé de {grossière, lisse}) ;
- U est l'univers du discours ou espace de référence de la variable x ;
- G est un ensemble flou de termes reflétant un certain degré de perception applicable à chaque terme de $T(x)$ tels que fort, peu, etc. ;

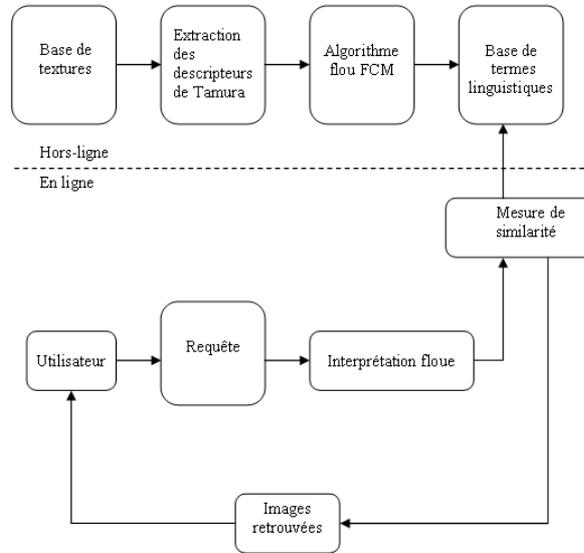


Figure 6.2 – Schéma du système de requêtes floues basées sur la texture [166]

- M désigne les règles sémantiques déduites par la programmation génétique.

Dans un récent travail [185], Li *et al.*, dans le cadre de la réduction du fossé sémantique basée sur la description en haut, montrent que l'application des réseaux de neurones améliore considérablement la sémantique des requêtes utilisateurs en montrant leurs robustesse aux imprécisions éventuelles des interprétations humaines et leurs capacité de modéliser plusieurs termes linguistiques en même temps au contraire de la programmation génétique qui procède un par un.

6.3 Notre démarche

Disposons d'une matrice de valeurs numériques $X_{ij}, i = 1, \dots, n, j = 1, \dots, m$, avec n est le nombre de textures dans la base et m est le nombre de descripteurs de bas niveau caractérisant les différentes textures. La question que nous nous sommes posés est comment passer de l'espace numérique des attributs à un espace linguistique traduisant une sémantique perçue. Plus formellement, il s'agit de trouver une fonction f tel que :

$$f : \begin{array}{l} \mathcal{I} \rightarrow \{G\} \times \{T_j\} \\ X_{ij} \mapsto \{faible, moyen, fort\} \wedge t_j \end{array} \quad (6.1)$$

où G correspond à l'ensemble de termes linguistiques $\{faible, moyen, fort\}$ correspondants aux différents degrés de perception éventuels d'une caractéristique sémantique t_j du j^e descripteur. T_j est l'ensemble de caractéristiques sémantiques que peut avoir le j^e descripteur. Le tableau 6.1 dresse les différentes propriétés sémantiques des différents descripteurs retenus. Dans ce travail, nous nous limitons à une seule propriété par descripteur (celle en italique). Ainsi, une texture pourra être exprimée par : *faible contraste, forte direction, moyennement grossière*, etc.

En s'inspirant du formalisme des ontologies (cf. section 6.1.1), nous avons ramené le problème d'interprétation sémantique des textures en termes linguistiques à un problème de discrétisation de variables numériques (descripteurs de texture) en variables nominales à k intervalles dont chaque intervalle

Descripteur	Sémantique
Énergie	<i>homogénéité</i> , luminosité
Variance	<i>contraste</i> , dispersion, sombre
Uniformité	<i>uniforme</i>
Direction	<i>direction</i> , orientation
Grossièreté	<i>grossière</i> , granuleuse
<i>Line-likeness</i>	<i>linéarité</i> , présence de lignes

Table 6.1 – Propriétés sémantiques

c , ($c = 1, \dots, k$) correspondra à une valeur de G suivant leurs points de découpage (en anglais, *cutting points* ou *break points*). On parle dans ce cas d'une règle d'inférence que nous notons $sem(f_j)$ et qui permet d'affecter chaque descripteur f_j à une classe de G . Dans notre cas, puisque G comprend que trois termes, nous aurons à définir deux points de découpage qu'on note respectivement par inf_j et sup_j :

$$sem(f_j) : \begin{array}{l} \mathcal{I} \rightarrow \{faible, moyen, fort\} \\ I \mapsto \begin{cases} faible & \text{si } f_j(I) < inf_j \\ moyen & \text{si } f_j(I) < sup_j \\ fort & \text{sinon} \end{cases} \end{array} \quad (6.2)$$

Une attention particulière sera donnée à l'attribut « direction » où l'ensemble flou G des termes linguistiques sera décrit par les termes suivants $G_{dir} = \{horizontale, verticale, droite, gauche\}$. Ainsi, on aura à discrétiser l'attribut « direction » en quatre intervalles au lieu de trois, ce qui signifie trois bornes de découpage à identifier.

Dans ce qui suit, nous présentons les différentes approches de discrétisation que nous avons adopté dans ce travail en sélectionnant celle qui donne le meilleur regroupement.

6.3.1 Méthodes de discrétisation

La majorité des algorithmes d'apprentissage ou de classification utilisent des variables nominales [159] alors que pour plusieurs applications réelles, les variables en question appartiennent au domaine continu. Pour pouvoir adapter ces cas aux algorithmes existants, une phase de discrétisation se montre indispensable [83]. Plusieurs méthodes de discrétisation existent dans la littérature [104]. Dougherty et al. [65] classifient les techniques de discrétisation en trois catégories :

1. *supervisé* versus *non-supervisé* : dans le premier cas, l'information correspondant aux étiquettes des instances est utilisé par le processus de discrétisation. À l'inverse, la discrétisation non-supervisée est aveugle à cette information.
2. *globale* versus *locale* : la discrétisation globale [132] s'effectue sur l'espace des instances alors que la discrétisation locale se fait région par région en évaluant un certain critère (entropie, inertie, etc.) ; la région qui donne la meilleure valeur de ce critère et retenue.
3. *statique* versus *dynamique* : pour les méthodes statiques, chaque variable est discrétisée en k intervalles indépendamment des autres variables (le k varie d'une variable à l'autre). Dans le cas dynamique, le k est choisi en tenant compte de toutes les variables. Dans cette thèse, nous nous sommes intéressés aux méthodes statiques.

Dans ce qui suit, nous présentons les différentes méthodes de discrétisation retenues dans ce travail : l'algorithme 1R [125], la méthode de minimisation d'entropie [74, 32], l'algorithme *k-means* [183, 151],

l'algorithme *k-medoids* [153], l'algorithme FCM [16, 70] et l'algorithme GG [93], ainsi notre proposition la ϵ -distance_{max} [134] basée sur la distance maximale.

Nous avons réparti l'ensemble de ces méthodes en deux catégories : discrétisation stricte (en anglais, *crisp* ou *hard*) et floue (*fuzzy*).

6.3.1.1 Discrétisation stricte

La discrétisation stricte signifie que les différents intervalles obtenus sont disjoints. Chaque instance n'appartient qu'à un seul intervalle. Plus formellement, soit X la matrice de n instances $\{x_i \in \mathbb{R}^m\}$ calculés pour m variables :

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix} \quad (6.3)$$

En se basant sur la théorie des ensembles classiques, une partition stricte est un ensemble d'intervalles $\{I_c | 1 \leq c \leq k\}$ tel que :

$$\begin{aligned} \bigcup_{c=1}^k I_c &= X \\ I_c \cap I_{c'} &= \emptyset \quad \forall c \neq c' \\ \emptyset &\subset I_c \subset X \quad \forall 1 \leq c \leq k \end{aligned} \quad (6.4)$$

Algorithme 1R L'algorithme 1R connu en anglais sous le nom de « One-Rule Discretizer » (1RD) introduit par Holte [125] est parmi les méthodes simples de discrétisation supervisée. Il est considéré comme un cas particulier des arbres de décision puisqu'il permet de définir qu'une seule règle de décision. Prenons l'exemple cité dans la section des arbres de décision (section ??) qui mesure la température. Avec les arbres de décision, deux décisions sont induites (malade ou bien portant) selon la valeur mesurée ; dans le cas du 1R, une seule décision sera induite : si la valeur mesurée est supérieure à une certaine valeur α (point du découpage), le patient est malade.

L'algorithme 1R se déroule comme suit :

1. tri

L'ensemble des valeurs numériques est trié en ascendant ;

2. segmentation

L'ensemble trié est partitionné en différents intervalles tel que chaque intervalle ne contient que les instances provenant de la même classe.

3. élimination des intervalles de taille faible

Les intervalles obtenus par (2) de taille fortement faible sont fusionnés ; Holte a fixé la taille minimale des intervalles à six instances.

4. règle de décision finale

Après regroupement des intervalles, l'algorithme procède par calculer la classe majoritaire sur chaque intervalle ; ensuite, les intervalles adjacents de même classe sont fusionnés. Par conséquent, la règle de décision correspond à la classe majoritaire sur l'ensemble des intervalles.

Méthode récursive de minimisation de l'entropie est une méthode de discrétisation supervisée basée sur la mesure de l'entropie sur les différents intervalles [65, 74]. Le partitionnement qui donne la plus faible valeur de l'entropie est retenu. L'algorithme se déroule comme suit :

1. discrétisation binaire

L'algorithme procède, dans un premier temps, par découper l'ensemble des instances en deux intervalles en calculant l'entropie donnée par l'équation 6.5.

$$E(X_j, T; I) = \frac{|I_1|}{|I|} Ent(I_1) + \frac{|I_2|}{|I|} Ent(I_2) \quad (6.5)$$

où X_j est le j^e descripteur à discrétiser, T est un point de découpage et I l'ensemble des instances estimés pour le descripteur X_j ($I = \{x_{1j}, \dots, x_{nj}\}$).

2. point du découpage optimal

Le point du découpage optimal T_{min} correspond à la valeur minimale de l'entropie calculée dans (1) :

$$T_{min} = \operatorname{argmin}_T \{E(X_j, T; I)\}$$

3. recherche du nombre k optimal

Les étapes (1) et (2) sont réitérées récursivement sur les différentes partitions obtenues jusqu'à vérification d'un critère d'arrêt. Fayyad et Irani [74] utilisent le principe de description de longueur minimale (en anglais, Minimal Description Length Principle (MDLP)) pour définir un critère d'arrêt. Une partition induite par un point de découpage T est acceptée si :

$$Gain(X_j, T; I) > \frac{\log_2(n-1)}{n} + \frac{\Delta(X_j, T; I)}{n}$$

avec :

- $Gain(X_j, T; I) = Ent(I) - \Delta(X_j, T; I)$;
 - $\Delta(X_j, T; I) = \log_2(3^k - 2) - [k \cdot Ent(I) - k_1 \cdot Ent(I_1) - k_2 \cdot Ent(I_2)]$;
- où k_i est le nombre de classes présentes dans l'intervalle I_i

Algorithme k -means La méthode des k -means (ou k -moyennes) introduite par J. B. MacQueen [201] en 1967 est l'une des méthodes de classification non-supervisée largement utilisée dans divers domaines tels que classification, segmentation et aussi pour la discrétisation des variables continues [222, 303, 98].

Dans le cadre de la discrétisation, on cherche généralement à partitionner l'espace des données en classes homogènes et isolées les unes des autres. Dans cette optique, l'algorithme k -means vise à minimiser la variance intra-classes (classes homogènes), ce qui se traduit par la minimisation de la fonction objective \mathbf{J} , qui n'est autre que la mesure d'erreur :

$$\mathbf{J} = \sum_{c \in C} \sum_{i=1}^n \|x_i^{(c)} - m_c\|^2 \quad (6.6)$$

où $\|x_i^{(c)} - m_c\|^2$ est une distance entre le point $x_i^{(c)}$ (l'instance x_i affectée à la classe c) et le centre m_c de la classe c . C est l'espace de toutes les classes.

L'algorithme k -means choisit, un premier temps, c valeurs distinctes constituant les centres des c classes. Ensuite, chaque instance est affectée à la classe dont le centre est le plus proche. Après affectation de toutes les instances, les centres des classes sont réévalués. Les processus d'affectation et de réévaluation des nouveaux centres sont réitérés jusqu'à stabilisation (les centres des classes ne changent pas). L'algorithme décrit d'une manière formelle est donné comme suit :

Algorithme 7 *k-means*

Soit $v_c = \frac{1}{N_c} \sum_{i=1}^n x_i^{(c)}$ le centre de gravité de la classe c .

1. **pour** $c \in C$
2. $m_c \leftarrow x_i | x_i \in X$ // Initialisation aléatoire des centres
3. $c \leftarrow \{x_i \in X | \min_{q \in C} \|x_i - m_q\|^2 = \|x_i - m_c\|^2\}$ // Affectation des instances
4. **fin pour**
5. **tant que** $\exists c \in C | m_c \neq v_c$
6. **pour** $c \in C$
7. $m_c \leftarrow v_c$ // Calcul des nouveaux centres
8. $c \leftarrow \emptyset$
9. **fin pour**
10. **pour** $x_i \in X$
11. $\hat{c} \leftarrow \operatorname{argmin}_{c \in C} \|x_i - m_c\|^2$
12. $\hat{c} \leftarrow \hat{c} \cup x_i$ // Réaffectation des instances
13. **fin pour**
14. **fin tant que**
15. **retourner** $hatc \in C$

Algorithme *k-medoids* L'algorithme *k-medoids* [222, 118] est un algorithme de classification non-supervisé basé sur le même principe que les *k-means* pour le partitionnement des données. La seule différence qui existe est que les centres des classes pour l'algorithme *k-medoids* représentent des objets ou instances les plus représentatifs des classes. D'où vient le terme *medoid*. L'algorithme est donné ci-dessous en considérant les mêmes annotations du *k-means* :

Notre proposition: méthode ϵ -distance_{max} La méthode que nous proposons ici combine les deux aspects supervisé et non-supervisé [134]. L'idée sur laquelle nous nous sommes reposés découle de nos observations expérimentales : la différence entre textures proches sémantiquement est faible alors que pour deux textures différentes sémantiquement, pouvant provenir de la même classe visuelle, la différence est significative. Cela est illustré par la figure 6.3. Les textures 6.3(a) et 6.3(b) appartiennent à deux classes différentes (« raffia » et « brick ») mais à la même classe sémantique (faible contraste), leurs différence est de l'ordre de 0,088 alors que pour les deux textures 6.3(b) et 6.3(c) provenant de la même classe visuelle (« brick ») mais appartenant à deux classes sémantiques différentes (faible contraste et fort contraste respectivement) leurs différence est de 0,697. La différence est nette entre les classes.

De ces constats, pour répartir nos textures en classes sémantiques disjointes, nous proposons d'utiliser la distance maximale comme méthode discrétisation de nos attributs de texture. Les points de découpage optimaux correspondent aux plus grandes différences entre deux valeurs consécutives. L'utilisation de l'algorithme à ce stade est non-supervisé entraînant deux inconvénients. Le premier concerne la multiplicité d'intervalles obtenus de taille faible éventuelle et le second est lié au découpage, deux textures appartenant à la même classe sémantique peuvent être séparées en deux classes différentes (les textures

Algorithme 8 *k-medoids*

-
1. **pour** $c \in C$
 2. $m_c \leftarrow x_i | x_i \in X$ // Initialisation aléatoire des centres
 3. $c \leftarrow \{x_i \in X | \min_{q \in C} \|x_i - m_q\|^2 = \|x_i - m_c\|^2\}$ // Affectation des instances
 4. **fin pour**
 5. **tant que** $\exists c \in C | m_c \neq v_c$
 6. **pour** $x_i \in X$
 7. $d_{ic}^2 \leftarrow \|x_i - v_c\|^2$
 8. $x_i^* \leftarrow \operatorname{argmin}_i d_{ic}^2$ // Détection des *medoids*
 9. $v_c \leftarrow x_i^*$ // Affectation des *medoids* aux centres
 10. **fin pour**
 11. **fin tant que**
-

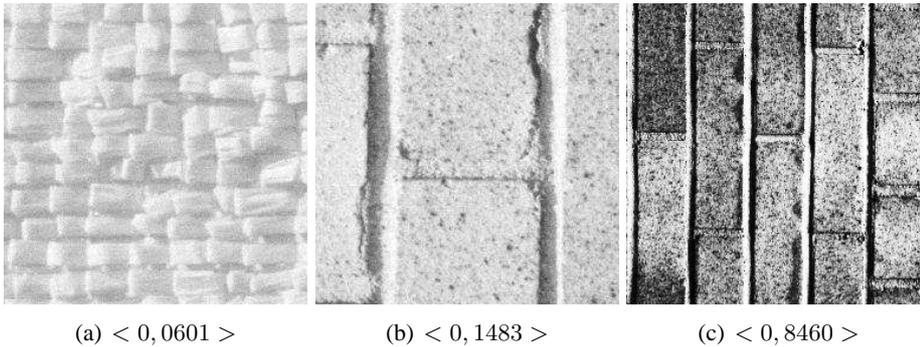


Figure 6.3 – Textures à différents contrastes

des bornes notamment). Pour pallier à ces inconvénients, nous avons rajouté deux contraintes : la taille minimale des classes doit être supérieure à 10% de la base afin d'éliminer les petits intervalles non-significatifs. La deuxième contrainte liée à un mauvais découpage fait introduire le jugement de l'expert. Après l'identification des classes de taille significative reflétant d'une certaine manière la répartition des textures, l'expert intervient pour valider ou pas les classes obtenues. Son intervention conduit au réglage des bornes du découpage initiales, il s'agit du terme ϵ , qui peuvent être réduites ($-\epsilon$) ou avancées ($+\epsilon$).

Nous résumons l'algorithme comme suit :

1. tri des données

les valeurs correspondantes à chaque descripteur sont triées en ascendant.

2. initialisation des bornes

les différences d entre valeurs consécutives ainsi que les premières bornes de découpage α sont estimées par :

$$\forall i \in [1, n] \quad d_i = \|x_{i+1} - x_i\|$$

et :

$$\alpha_c = \operatorname{argmax}_{i \in c} \{d_i\}$$

3. élimination des intervalles faibles

à ce stade, on calcule le nombre d'instances dans chaque partition obtenue, si le nombre est inférieur à 10% du nombre total dans la base, cette partition sera fusionnée avec la partition adjacente. Les étapes (2) et (3) sont répétées jusqu'à validation de contraintes.

4. identification des bornes de découpage finales

Après identification des pré-partitions par les étapes (2) et (3), l'expert intervient pour valider les bornes finales α_i . Le paramètre ϵ correspond à une valeur subjective que l'expert juge représentative d'une classe sémantique :

$$\alpha$$

$$c = \alpha_c \pm \epsilon$$

6.3.1.2 Discrétisation floue

Du fait des transitions brutales entre les différentes classes dans le cas de la discrétisation stricte qui réellement sont imprécises et peuvent manquer de représentativité sémantique en particulier lorsque le nombre des données dont les valeurs aux frontières (bornes) des classes est important. L'intégration des ensembles flous dans le processus de discrétisation permet donc une gestion graduelle des données en évitant le problème de transitions abruptes entre les classes.

En principe, la discrétisation floue [264] est identique à la discrétisation stricte qui cherchent à regrouper un ensemble de n données en k groupes homogènes. Elle en diffère par le fait que les classes obtenues peuvent se chevaucher et les fonctions d'appartenance ne sont pas exclusives. Chaque classe est considérée comme un ensemble flou et un degré d'appartenance de chaque observation à chacune d'entre elles est calculé [62]. Dans ce cadre, on peut citer les méthodes neuronales qui incluent l'apprentissage des paramètres déterminant les fonctions d'appartenance [132], ou encore les méthodes basées sur la recherche d'une partition floue la plus discriminative en utilisant une mesure d'entropie floue [207]. Dans ce travail, nous nous focaliserons sur les méthodes de regroupement flous à savoir l'algorithme FCM (*Fuzzy C-Means*) et l'algorithme GG (Gath-Geva).

Une partition de n données en k classes peut être décrite par une matrice de partition floue à n lignes et k colonnes de terme général μ_{ik} à valeurs dans $[0, 1]$. Le terme μ_{ik} représente le degré d'appartenance de la donnée x_i à la classe k et vérifie les conditions suivantes :

$$\begin{aligned} \mu_{ik} &\in [0, 1] \quad \forall 1 \leq i \leq N, 1 \leq k \leq c \\ \sum_{k=1}^c \mu_{ik} &= 1 \quad \forall 1 \leq i \leq N \\ 0 < \sum_{i=1}^N \mu_{ik} &< N \quad \forall 1 \leq k \leq c \end{aligned} \quad (6.7)$$

Ces conditions signifient qu'un vecteur appartient à plusieurs classes (à divers degrés) et qu'aucune classe n'est vide.

Algorithme FCM L'algorithme FCM, introduit par Dunn [69] et amélioré par Bezdek [16], généralise la méthode des k -means pour le calcul des centres des différentes classes. Il consiste à minimiser itérativement une fonction objective \mathbf{J} qui utilise comme mesure d'homogénéité de la partition, la somme des distances entre chaque donnée aux différents centres, pondérées par son degré d'appartenance à chacun d'eux :

$$\mathbf{J}_m = \sum_{c \in C} \sum_{i=1}^n (\mu_{ic})^m \|x_i - m_c\|^2 \quad (6.8)$$

où m est un paramètre qui contrôle le degré flou de la partition ($m > 1$) et m_c le centre de la classe k . L'algorithme débute avec une initialisation aléatoire des μ_{ik} , recalcule les centres des classes puis met à jour la partition en recalculant de nouveaux les μ_{ik} . Cette procédure est répétée et s'arrête si un nombre d'itérations maximal est atteint ou si le changement des degrés d'appartenance entre deux itérations successives n'est pas significatif (inférieur à un seuil prédéfini). L'algorithme est détaillé dans 9 :

Algorithme 9 FCM

1. initialisation aléatoire de la matrice de partition $U = [\mu_{ic}], U^{(0)}$
 2. **tant que** $\|U^{(k+1)} - U^{(k)}\| < \epsilon$ **faire**
 3. $m_c = \frac{\sum_{i=1}^n \mu_{ic}^m \cdot x_i}{\sum_{i=1}^n \mu_{ic}^m}$ // Recalcule des centres
 4. $\mu_{ic} = \frac{1}{\sum_{q \in C} \left(\frac{\|x_i - m_c\|}{\|x_i - m_q\|} \right)^{\frac{2}{m}}}$ // Mise à jour des degrés
 5. **fin tant que**
-

L'algorithme FCM nécessite le choix de deux paramètres : le nombre de classes k et le degré de flou m de la partition. Plus m est proche de 1, plus la partition tend à être « classique ». Inversement, lorsque m augmente, la partition devient floue.

Algorithme GG L'algorithme GG introduit par Gath et Geva [93] est un algorithme de partitionnement basé sur l'estimation floue du maximum de vraisemblance (en anglais *fuzzy maximum likelihood*

estimates (FMLE)) [17]. La fonction objective à minimiser est de la forme suivante :

$$\mathbf{J}_m = \sum_{c \in C} \sum_{i=1}^n (\mu_{ic})^m D_{ic}^2 \quad (6.9)$$

avec D_{ic}^2 est la distance entre la donnée x_i et la classe c calculée à partir de leurs covariances.

Les étapes sont décrites dans l'algorithme 10.

Algorithme 10 GG

1. initialisation aléatoire de la matrice de partition $U = [\mu_{ic}], U^{(0)}$
 2. **tant que** $\|U^{(k+1)} - U^{(k)}\| < \epsilon$ **faire**
 3. $m_c^{(k)} = \frac{\sum_{i=1}^n \mu_{ic}^m \cdot x_i}{\sum_{i=1}^n \mu_{ic}^m}$ // Calcule des centres
 4. $F_c^{(k)} = \frac{\sum_{i=1}^n \mu_{ic}^m (x_i - m_c) \cdot (x_i - m_c)^T}{\sum_{i=1}^n \mu_{ic}^m}$ // Calcule des matrices de covariances floues des classes
 5. $D_{ic}^2 = \frac{(2\pi)^{n/2} \sqrt{\det F_c^{(k)}}}{\alpha_c} \exp(1/2(x_i - m_c)^T \cdot F_c^{-1} \cdot (x_i - m_c))$ // Calcule des distances
 6. $\mu_{ic} = \frac{1}{\sum_{q \in C} (\frac{D_{ic}}{D_{iq}})^{\frac{2}{m-1}}}$ // Mise à jour des degrés d'appartenance
 7. **fin tant que**
-

Les paramètres α_c et μ_{ic} représentent respectivement la probabilité *a priori* de sélectionner la classe c et la probabilité *a posteriori* de sélectionner la classe c sachant la donnée x_i .

L'utilisation d'une « distance exponentielle » incluant une matrice de covariance floue F permet d'obtenir des partitions optimales variées en formes, en densités et en taille. Avec cette propriété, l'algorithme GG [222] se montre plus performant que l'algorithme FCM qui ne peut identifier que les classes de même forme et orientation.

Dans un cadre général, l'inconvénient des algorithmes de partitionnement stricts et flous est qu'il faut fixer *a priori* le nombre de classes à identifier, propriété qui n'est pas toujours évidente. Dans notre cas, ce problème n'est pas posé étant donné que nous avons pré-identifier nos trois classes sémantiques, ainsi le paramètre k sera fixé à trois dans toutes les expérimentations.

6.3.2 Mesures d'évaluation

Après identification des partitions par l'étape précédente, la question qui se pose immédiatement : quelle est la « bonne » partition à choisir? La réponse à cette question est incluse sous le domaine de la validation des clusters connu en anglais par *cluster validity assessment*. L'objectif des techniques de validation des clusters est de mesurer le meilleur partitionnement relativement à d'autres obtenus par d'autres algorithmes de partitionnement ou à partir des mêmes algorithmes utilisant des paramètres différents. L'évaluation et la validation de la répartition optimale est basée sur deux critères [15, 67] :

- *compacité* : Elle mesure l'homogénéité et la cohérence des données dans chaque classe. Les données d'une même classe fortement proches engendre une partition compacte. L'évaluation de la

compacité d'une partition dépend de la mesure utilisée. Par exemple, si l'on mesure par la variance, une valeur minimale signifie une grande compacité. À l'inverse, si l'on utilise la similarité moyenne ; plus cette valeur est grande, plus la partition est compacte.

- *séparation* : Un « bon » partitionnement correspond à des classes bien séparées entre elles. Mesurer la séparation entre deux classes peut être effectuée selon trois manières : (i)- en mesurant la distance entre les données les plus proches des deux classes ; (ii)- en mesurant la distance des données les plus éloignées et (iii)- en mesurant la distance entre les centres des classes. Une grande valeur de cette distance entraîne une bonne séparation. Si l'on utilise la similarité entre classes, plus cette valeur est faible, plus les classes sont bien séparées.

Définition 2. *Un partitionnement est optimal et valide si les classes sont compactes et bien séparées.*

Nombreuses sont les mesures proposées dans la littérature pour évaluer la validité des classes obtenues par les différents algorithmes de partitionnement [72, 222]. L'utilisation d'un seul indice est insuffisante pour évaluer la performance du partitionnement. Dans cette thèse, nous proposons d'utiliser six indices pour l'évaluation respective des classes dans le cas strict et flou à savoir : l'indice de partition *SC* (*partition index*) [11], l'indice de séparation *S* (*separation index*) [11], l'indice de Dunn *DI* [296, 19, 70], l'indice *CS* de Chou [56, 48], l'indice *PBM* [177, 214, 223] et l'indice Fukuyama-Sugeno *FS* [37, 2, 306, 87] pour la discrétisation stricte et l'indice *PC* (*partition coefficient*) [37, 2, 111], l'indice d'entropie *PE* (*partition entropy*) [37, 18, 111], l'indice Xie-Beni *XB* [37, 111, 323], l'indice *PBM_m* [177, 214, 223], l'indice *FS_m* [37, 2, 111] et l'indice *PACES* [37, 322] pour la discrétisation floue.

6.3.2.1 Évaluation dans le cas strict

Indice *SC* L'indice de partition *SC* proposé par Bensaid et al. [11] mesure pour chaque classe le rapport de sa compacité et sa séparation. La valeur globale est obtenue en moyennant les différentes valeurs obtenues pour chaque classe. Sa formule est donnée comme suit :

$$SC = \sum_{c \in C} \frac{\sum_{i=1}^n \|x_i^c - v_c\|^2}{N_c \sum_{q \in C} \|v_q - v_c\|^2} \quad (6.10)$$

avec x_i^c est la i^e donnée appartenant à la classe c , v_k le centre de la classe k et N_k nombre de données dans la classe k .

L'indice *SC* est très utile lors de la comparaison des différentes partitions ayant le même nombre de classes. Une faible valeur de *SC* indique une bonne partition.

Indice *S* L'indice de séparation *S* [11] à l'inverse de l'indice *SC* utilise la distance minimale pour mesurer la séparation des partitions. Sa formule est donnée comme suit :

$$S = \frac{\sum_{c \in C} \sum_{i=1}^n \|x_i^c - v_c\|^2}{n \min_{q,c} \|v_q - v_c\|^2} \quad (6.11)$$

De même que l'indice *SC*, une valeur faible de *S* correspond à une bonne partition.

Indice DI L'indice DI introduit par Dunn [70] est le rapport entre la distance intra-classes minimale (diamètre des classes) et la distance inter-classes maximale (dissimilarité entre classes). Sa formule est donnée par l'équation suivante :

$$DI = \min_{q \in C} \left\{ \min_{c \neq q \in C} \left\{ \frac{d(c, q)}{\max_{q' \in C} \{diam(q')\}} \right\} \right\} \quad (6.12)$$

avec :

- $d(c, q) = \min_{x \in c, y \in q} \{d(x, y)\}$;
- $diam(c) = \max_{x, y \in c} \{d(x, y)\}$.

Si les données contiennent des classes bien séparées, la distance entre classes est toujours grande tandis que le diamètre des classes est petit [111]. Ainsi, une valeur grande de l'indice DI reflète une bonne partition. Les seuls inconvénients de cet indice sont : le temps de calcul qui augmente considérablement avec la taille de données et sa sensibilité au bruit (pour des données fortement bruitées, le diamètre des classes sera plus large). Plusieurs variantes se sont proposées qui diffèrent dans la définition des distances et le diamètre des classes [292, 224].

Indice CS Récemment Chou et *al.* [48] proposent une nouvelle mesure de la validité des classes notée CS^6 . Basée sur le même principe que les indices introduits précédemment, l'indice CS représente aussi le rapport des distances intra et inter classes. Sa formule est donnée par l'équation 6.13 :

$$CS = \frac{\sum_{c \in C} \left[\frac{1}{N_c} \sum_{x \in c} \max_{y \in c} \{d(x, y)\} \right]}{\sum_{c \in C} \left[\min_{q \neq c} \{d(v_c, v_q)\} \right]} \quad (6.13)$$

Un partitionnement optimale correspond à une valeur petite de l'indice CS . Chou et *al.* considèrent que cette mesure est efficace pour identifier des classes de densités et de taille différentes. Le seul inconvénient de cet indice est le temps de calcul qui augmente avec la taille des classes et des données.

Indice PBM Pakhira et *al.* [223] proposent un nouveau indice PBM^7 adapté aux deux cas du partitionnement stricte et flou. L'indice PBM est le résultat du produit de trois facteurs d'où sa maximisation assure un partitionnement caractérisé par un nombre réduit de classes compactes et une large séparation entre classes. L'indice PBM est défini comme suit :

$$PBM(C) = \left(\frac{1}{C} \times \frac{E_1}{E_C} \times D_C \right) \quad (6.14)$$

avec C le nombre totale des classes.

E_1 est la somme des distances entre chaque donnée x et le centre globale v . E_1 est donné comme suit :

$$E_1 = \sum_{i=1}^n \|x_i - v\| \quad (6.15)$$

E_C est la somme des distances intra-classes donnée par :

$$E_C = \sum_{c \in C} \sum_{i=1}^{N_c} \|x_i^c - v_c\| \quad (6.16)$$

⁶ C pour Chou et S pour Su

⁷ P pour Pakhira, B pour Bandyopadhyay et M pour Maulik

et la distance entre les centres des classes est donnée par D_C :

$$D_C = \max_{c,q \in C} \|v_c - v_q\| \quad (6.17)$$

Indice FS L'indice FS proposé par Fukuyama et Sugeno [87] pour mesurer la validité d'un partitionnement en se basant sur la mesure de la compacité et la séparation des classes. Sa formule est donnée par l'équation 6.18.

$$FS = \sum_{c \in C} \sum_{i=1}^{N_c} [\|x_i^c - v_c\|^2 - \|v_c - v\|^2] \quad (6.18)$$

Une faible valeur du FS indique des classes compactes avec une large séparation entre elles. Varma et Simon [306] ont trouvé que l'indice FS est performant dans le cas des variables bruitées.

6.3.2.2 Évaluation dans le cas flou

L'évaluation dans le cas du partitionnement flou tient en compte le chevauchement entre classes représenté par les degrés d'appartenance.

Indice PC Bezdek a proposé dans [16] une mesure de validité des classes dans le cas flou basé sur les degrés d'appartenance uniquement. Sa formule est donnée comme suit :

$$PC = \frac{1}{n} \sum_{i=1}^n \sum_{c \in C} \mu_{ic}^2 \quad (6.19)$$

La valeur du PC appartient à l'intervalle $[\frac{1}{C}, 1]$. Si la valeur atteint 1, on est dans le cas d'un partitionnement stricte. Dans le cas où la valeur du PC est égale à $\frac{1}{C}$, cela indique que les données ne peuvent pas être partitionnées ou bien que l'algorithme du partitionnement échoue à trouver la bonne représentation. Un bon partitionnement correspond à une valeur importante du PC .

Indice PE Bezdek et al. [18] ont proposé un indice de mesure de la validité sous la forme de l'entropie des degrés d'appartenance des données. Plus cet indice est petit plus le partitionnement est considéré bon. Sa formule est donnée comme suit :

$$PE = -\frac{1}{n} \sum_{i=1}^n \sum_{c \in C} \mu_{ic} \log_a \mu_{ic} \quad (6.20)$$

La valeur du PE appartient à l'intervalle $[0, \log_a .C]$. Si sa valeur est proche de 0, on est dans le cas d'un partitionnement stricte. De même que l'indice PC , si sa valeur est proche de la borne supérieure, cela entraîne une absence de partitionnement ou bien défaillance de l'algorithme. Un bon partitionnement correspond à une valeur petite du PE .

Indice XB L'indice de Xie et Beni [323] (XB) mesure la compacité et la séparation des classes en se basant sur les degrés d'appartenance et sur les données eux-mêmes. Une valeur petite du XB indique un partitionnement optimale :

$$XB = \frac{\sum_{c \in C} \sum_{i=1}^n \mu_{ic}^m \|x_i - v_c\|^2}{n \cdot \min_{c,q} \|v_c - v_q\|^2} \quad (6.21)$$

Le numérateur représente la compacité mesurée la somme des carrés des distances intra-classes alors que le dénominateur représente la séparation mesurée par la distance minimale inter-classes.

Indice PBM_m Cet indice est une extension de l'indice PBM introduit précédemment. Dans ce cas, la distance intra-classe devient :

$$E_C = \sum_{c \in C} \sum_{i=1}^n \mu_{ic}^m \|x_i - v_c\| \quad (6.22)$$

Indice FS_m Cet indice est une extension de l'indice FS pour le cas flou incluant les degrés d'appartenance et les données elles-mêmes :

$$FS_m = \sum_{c \in C} \sum_{i=1}^{N_c} \mu_{ic}^m [\|x_i - v_c\|^2 - \|v_c - v\|^2] \quad (6.23)$$

Le premier terme entre crochets mesure la compacité des classes et le second terme mesure les distances entre classes.

Indice $PACES$ Wu et Yang [322] ont proposé un nouvel indice pour la validité des partitionnement flous nommé $PACES$ ⁸ :

$$PACES = \sum_{c \in C} \sum_{i=1}^n \frac{\mu_{ic}^2}{\mu_M} - \sum_{c \in C} \exp(-\min_{c \neq q} \{\|v_c - v_q\|^2\} / \beta_T) \quad (6.24)$$

avec :

- $\mu_M = \max_{c \in C} \sum_{i=1}^n \mu_{ic}^2$;
- $\beta_T = \frac{1}{C} \sum_{c \in C} \|v_c - v\|^2$.

L'indice $PACES$ est composé de deux termes. Le premier est l'indice PC normalisé pour mesurer la compacité et le second terme est une mesure de séparation exponentielle des distances entre centres des classes.

6.4 Expérimentations et résultats

Dans cette section, nous allons présenté les différents résultats de discrétisation obtenus par les algorithmes précédemment introduits (*cf.* section 6.3.1). L'évaluation et la validation des différentes partitions est basée sur les indices de validité présentés dans la section 6.3.2. Selon les propriétés de chaque indice, nous soutenons qu'un partitionnement stricte est optimale si : les indices SC , S , CS et FS présentent des valeurs faibles et les indices DI et PBM sont importants. Dans le cas flou, un partitionnement est optimale si les indices PE , XB et FS_m sont petits et les indices PC , PBM_m et $PACES$ sont grands.

Dans cette thèse, la discrétisation est univariée, c'est-à-dire que chaque descripteur est discrétisé seul indépendamment des autres. Le nombre de classes est fixé *a priori* à trois. Nous fixons aussi le paramètre m de contrôle du degré flou de la partition à deux.

⁸partition coefficient and exponential separation

6.4.1 Résultats de la discrétisation stricte

Les tableaux 6.2 à 6.7 présentent les valeurs des indices de validité strictes estimées pour chaque descripteur :

- pour l'énergie (tableau 6.2), les valeurs optimales pour les indices SC , S et FS sont obtenues par l'approche proposée ϵ -distance_{max}. La valeur maximale des indices DI et PBM est obtenue par l'algorithme k -medoids et la valeur minimale de l'indice de Chou CS est donnée par l'algorithme supervisé 1R. Nous constatons alors que les trois algorithmes ϵ -distance_{max}, k -medoids et 1R fournissent des valeurs optimales pour différents indices. Le choix dans ce cas n'est pas si évident. Nous proposons une sélection du partitionnement optimale basée sur l'indice de Dunn DI (l'algorithme qui donne la plus grande valeur sera sélectionné) ; ce qui correspond ici à l'algorithme k -medoids.
- pour la variance (tableau 6.3), les valeurs minimales du SC , S et FS sont obtenues par l'algorithme de Fayyad et Irani $MDLP$. L'approche ϵ -distance_{max} présente les valeurs optimales pour les indices DI et CS . Une valeur maximale pour l'indice PBM est donnée par l'algorithme k -means. La même constatation est faite que précédemment, les valeurs optimales des différents indices sont données par les algorithmes ϵ -distance_{max}, $MDLP$ et k -means. Basé sur l'indice DI , nous optons pour l'approche ϵ -distance_{max}.
- pour l'uniformité (tableau 6.4), les valeurs optimales des indices SC , S et DI sont obtenues par l'approche ϵ -distance_{max}. La valeur minimale de l'indice CS est obtenue par l'algorithme $MDLP$. L'indice PBM atteint son maximum avec l'algorithme k -medoids tandis que l'indice FS atteint sa valeur minimale avec l'algorithme k -means. Basé sur l'indice DI dans le cas d'une multitude d'algorithmes de partitionnement, nous optons pour l'approche ϵ -distance_{max}.
- pour la direction (tableau 6.7), les valeurs optimales des indices SC et S sont obtenues par l'algorithme $MDLP$. La valeur maximale de l'indice de Dunn DI est donnée par l'algorithme k -means. L'algorithme 1R présente des valeurs optimales pour les indices CS et FS et la valeur maximale de l'indice PBM est donnée par l'algorithme k -medoids. Toujours la même constatation et en se basant sur l'indice DI , le partitionnement par l'algorithme k -means sera choisi.
- pour la grossièreté (tableau 6.5), nous constatons que les cinq algorithmes 1R, $MDLP$, k -means, k -medoids et ϵ -distance_{max} donnent des valeurs optimales pour les indices SC et FS , PBM , CS , DI et S respectivement. Nous choisirons l'algorithme k -medoids pour sa valeur maximale pour l'indice DI .
- pour le *line-likeness* (tableau 6.6), les valeurs optimales des indices SC et S sont obtenues par l'algorithme 1R. Les valeurs optimales des indices PBM et DI sont données respectivement par l'algorithme $MDLP$ et k -means. Quant à l'algorithme ϵ -distance_{max}, il donne des valeurs optimales pour les indices CS et FS . De même, nous choisirons l'algorithme k -means pour sa valeur maximale pour l'indice DI .

6.4.2 Résultats de la discrétisation floue

Les tableaux 6.8 à 6.13 présentent les valeurs des indices de validité flous estimées pour chaque descripteur :

- pour l'énergie (tableau 6.8), les valeurs optimales des indices PC , PE , PBM_m et $PACES$ sont obtenues par l'algorithme flou de Gath-Geva GG alors que l'algorithme flou FCM donne les valeurs minimales pour l'indice de Xie-Beni XB et FS_m . Nous constatons que pour les indices PC et PE sont proches respectivement des bornes 1 et 0, ce qui signifie que l'algorithme GG peut converger vers une partition stricte. Nous optons dans ce cas pour l'algorithme FCM.

Algorithme	1R	MDLP	<i>k-means</i>	<i>k-medoids</i>	ϵ -distance _{max}
SC	0,1074	0,0764	0,6285	0,6890	0,0550
S	2,9631e-04	1,8677e-04	0,0011	0,0012	9,2118e-05
DI	0,3207	0,4113	0,5462	0,5928	0,2788
CS	-0,7041	-0,6708	-0,5955	-0,6170	-0,4825
PBM	-2,8200	-6,2533	-0,8804	-0,7779	-11,9262
FS	-38,3109	-59,6204	-25,6561	-19,9806	-86,6541

Table 6.2 – Indices de validité pour la discrétisation stricte de l'énergie

Algorithme	1R	MDLP	<i>k-means</i>	<i>k-medoids</i>	ϵ -distance _{max}
SC	0,0728	0,0480	0,2477	0,2462	0,1311
S	1,8559e-04	1,1284e-04	3,9954e-04	3,9686e-04	3,5529e-04
DI	0,5701	0,5751	0,4503	0,4779	0,6609
CS	-0,7795	-0,7774	-0,4939	-0,4929	-0,7811
PBM	-7,8150	-11,7788	-1,3906	-1,3958	-3,8589
FS	-58,9717	-63,9043	-38,0200	-38,3047	-41,8312

Table 6.3 – Indices de validité pour la discrétisation stricte de la variance

Algorithme	1R	MDLP	<i>k-means</i>	<i>k-medoids</i>	ϵ -distance _{max}
SC	0,0849	0,1481	0,4973	1,3133	0,0661
S	2,3096e-04	3,8607e-04	9,2353e-04	0,0025	1,6379e-04
DI	0,4491	0,3560	0,4778	0,4593	0,4932
CS	-0,6849	-0,7403	-0,2067	-0,5449	-0,6406
PBM	-6,6686	-1,9194	-1,7848	-0,5060	-10,2935
FS	-8,3642	-4,5151	-43,1077	1,3645	-9,3242

Table 6.4 – Indices de validité pour la discrétisation stricte de l'uniformité

Algorithme	1R	MDLP	<i>k-means</i>	<i>k-medoids</i>	ϵ -distance _{max}
SC	0,0841	0,0988	0,9657	0,9607	0,0994
S	1,5768e-04	1,6626e-04	0,0018	0,0018	1,5090e-04
DI	0,0489	0,4461	0,3627	0,4777	0,4458
CS	-0,8676	-0,8418	-1,0880	-1,0849	-0,8092
PBM	-2,5216	1,9326	-0,0448	-0,0461	1,5506
FS	-33,1390	-25,5930	26,7843	26,6492	-21,2897

Table 6.5 – Indices de validité pour la discrétisation stricte de la grossièreté

Algorithme	1R	MDLP	<i>k-means</i>	<i>k-medoids</i>	ϵ -distance _{max}
SC	0,0956	0,1397	0,7823	0,7929	0,1366
S	1,9675e-04	3,5523e-04	0,0015	0,0016	3,3629e-04
CS	-1,0164	-1,3225	-0,9821	-0,9933	-1,3566
DI	0,2290	0,4347	0,6525	0,6406	0,2732
PBM	-0,1252	0,5572	0,3172	0,3048	0,0995
FS	-22,0688	-24,0125	14,3335	14,9704	-24,2548

Table 6.6 – Indices de validité pour la discrétisation stricte du *line-likeness*

Algorithme	1R	MDLP	<i>k-means</i>	<i>k-medoids</i>	ϵ -distance _{max}
SC	0,0634	0,0476	0,4845	0,4850	0,0580
S	1,7346e-04	1,212e-04	8,2272e-04	8,2319e-04	1,7532e-04
DI	0,2727	0,3534	0,4946	0,4439	0,1956
CS	-0,4669	-0,3496	-0,3896	-0,3851	-0,4484
PBM	-2,8141	-1,0617	-0,6023	-0,5991	-2,2492
FS	-37,6546	-28,7384	-19,0315	-18,7322	-34,3064

Table 6.7 – Indices de validité pour la discrétisation stricte de la direction

- pour la variance (tableau 6.9), nous constatons le même comportement que le descripteur « énergie ». Sur le même principe, l'algorithme FCM sera choisi.
- pour l'uniformité (tableau 6.10), seule la valeur optimale de l'indice *PC* est obtenue par l'algorithme GG, sinon pour le reste des indices, l'algorithme a eu une difficulté de converger et aucune valeur n'est pu être estimée. Les valeurs optimales sont donc obtenues par l'algorithme FCM.
- pour la direction (tableau 6.13), les valeurs optimales des indices *PC*, *PE* et *PBM_m* sont obtenues par l'algorithme GG alors que les valeurs optimales des autres indices *XB*, *FS_m* et *PACES* sont données par l'algorithme FCM. En se basant sur l'indice de référence *XB*, nous optons pour l'algorithme FCM.
- pour la grossièreté (tableau 6.11), l'algorithme GG converge avec certitude vers une partition stricte vue les valeurs des indices *pC* qui vaut 1 et *PE* qui tend fortement vers 0. L'indice *XB* obtient sa valeur optimale avec l'algorithme FCM.
- pour le *line-likeness* (tableau 6.12), nous faisons la même constatation que le descripteur « grossièreté », l'algorithme GG converge vers une partition stricte alors que l'algorithme FCM donne les valeurs optimales pour le reste des indices.

D'après l'analyse des différents indices de validité pour le cas de la discrétisation stricte et floue, nous résumons dans le tableau suivant (tableau 6.14) l'algorithme retenu pour chaque descripteur ainsi nous donnons les points de découpage trouvés et qui représentent les règles d'inférence pour l'affectation de chaque attribut de texture à l'un des termes sémantiques.

Pour la discrétisation floue, la matrice de partition *U* des degrés d'appartenance obtenue par l'algorithme FCM jouera le rôle des règles d'inférence. Chaque texture lui sera attribuée un terme sémantique, correspondant à un descripteur donné, selon son degré d'apparence dans l'image.

Pour le descripteur « direction », les bornes de découpage des différentes orientations obtenues par l'algorithme *k-means* sont : 0,079, 0,138 et 0,256 correspondant respectivement à la direction à droite,

Algorithme	FCM	GG
PC	0,8899	0,9488
PE	0,2126	0,0959
XB	0,5288	7,8909
PBM _m	-0,7450	-0,0749
FS _m	-31,8850	404,5882
PACES	0,4370	0,4531

Table 6.8 – Indices de validité pour la discrétisation floue de l'énergie

Algorithme	FCM	GG
PC	0,8874	0,9552
PE	0,2092	0,0820
XB	0,7844	12,2126
PBM _m	-1,1382	-0,0999
FS _m	-39,4405	395,9762
PACES	0,1507	0,2805

Table 6.9 – Indices de validité pour la discrétisation floue de la variance

Algorithme	FCM	GG
PC	0,9230	0,9354
PE	0,1449	–
XB	1,2574	–
PBM _m	-0,4395	–
FS _m	-1,1667	–
PACES	0,1640	-0,1838

Table 6.10 – Indices de validité pour la discrétisation floue de l'uniformité

Algorithme	FCM	GG
PC	0,7867	1
PE	0,3870	5,5023e-18
XB	0,9866	48,3541
PBM _m	-0,0264	0,0056
FS _m	16,3478	537,0338
PACES	1,6128	-0,0641

Table 6.11 – Indices de validité pour la discrétisation floue de la grossièreté

Algorithme	FCM	GG
PC	0,8032	1
PE	0,3565	1,2713e-17
XB	0,9356	68,3234
PBM _m	0,2563	0,0151
FS _m	5,8987	622,4076
PACES	0,9100	-0,5071

Table 6.12 – Indices de validité pour la discrétisation floue du *line-likeness*

Algorithme	FCM	GG
PC	0,8069	0,9579
PE	0,3714	0,0726
XB	1,2619	23,1472
PBM _m	-0,4355	-0,0413
FS _m	-19,1018	452,711
PACES	-0,2712	-0,5430

Table 6.13 – Indices de validité pour la discrétisation floue de la direction

Descripteur	modèle	borne inf	borne sup
énergie	<i>k-medoids</i>	0,202	0,539
variance	ϵ -distance _{max}	0,225	0,378
uniformité	ϵ -distance _{max}	0,058	0,143
grossièreté	<i>k-means</i>	0,404	0,636
<i>line-likeness</i>	<i>k-means</i>	0,441	0,731

Table 6.14 – Bornes de découpage des différents descripteurs de texture

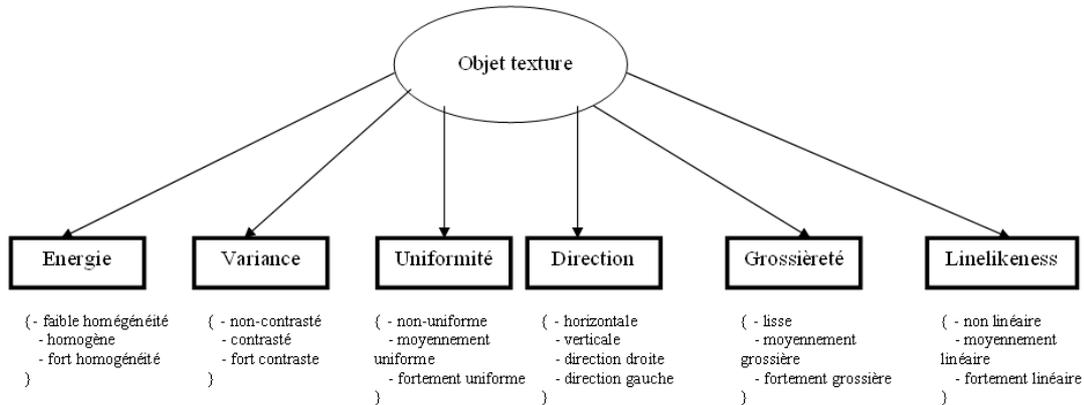


Figure 6.4 – Notre description sémantique d'une texture

direction horizontale, direction à gauche et direction verticale.

La figure 6.4 résume et définit notre description sémantique d'une texture. Un objet texture est défini par six attributs numériques à savoir : l'énergie, la variance, l'uniformité, la direction, la grossièreté et le *line-likeness*. Ces attributs sont ensuite convertis en termes linguistiques sémantiques à l'aide des algorithmes de discrétisation proposés en haut et cela afin de réduire le fossé sémantique entre les descripteurs numériques de bas niveau et les concepts sémantiques de haut niveau lié à la perception et la conception des requêtes des utilisateurs.

La figure 6.5 illustre un exemple d'interprétation sémantique de nos textures extraites de la base de Brodatz. Suivant la valeur de l'attribut comparé à la borne *inf* ou *sup*, le terme qui lui est associé est faible, moyen ou fort. Par exemple, pour la texture « bark » à gauche, la valeur de sa variance est de 0,2873 inférieure à la borne *sup*, ainsi la texture reflète un aspect contrasté. Si on prend la valeur de sa direction, elle est de 0,7494 supérieure à la borne *sup* et donc la texture est orientée verticalement et ainsi de suite. De même pour la texture « straw » à droite, la valeur de sa variance est de 0,0173 inférieure à sa borne *inf* et donc la texture est non contrastée, chose qui est bien aperçue dans la texture réellement. Pour la valeur de sa direction, elle vaut 0,0714 et donc inférieure à la borne *inf*, ce qui est interprété par une texture orientée à droite (propriété fortement corrélée à la direction perçue visuellement).

6.5 Interprétation sémantique des images réelles

Dans les sections précédentes, nous avons étudié les images standards issues d'une collection de texture, i.e, chaque image contient une seule texture. Dans cette partie, nous nous focalisons sur les images réelles. L'extension dans ce cas va introduire une multiplicité de textures. Ainsi, un processus de « segmentation » devient indispensable car la caractérisation globale de l'image entière perd les informations locales (régions textures) en produisant une moyenne grossière du contenu de cette image.

Deux approches différentes peuvent être utilisées pour caractériser une image réelle. La première approche consiste à diviser l'image en une grille et les caractéristiques numériques et sémantiques sont calculés pour chaque case de la grille. La seconde approche consiste à segmenter l'image en zones locales proches des objets ou textures constituant l'image et ensuite calculer les différents attributs numériques et caractéristiques sémantiques pour chaque région extraite. Dans ce travail, nous souhaitons éviter la phase de segmentation à proprement dit, en raison de tous les problèmes pour choisir une "bonne et meilleure"

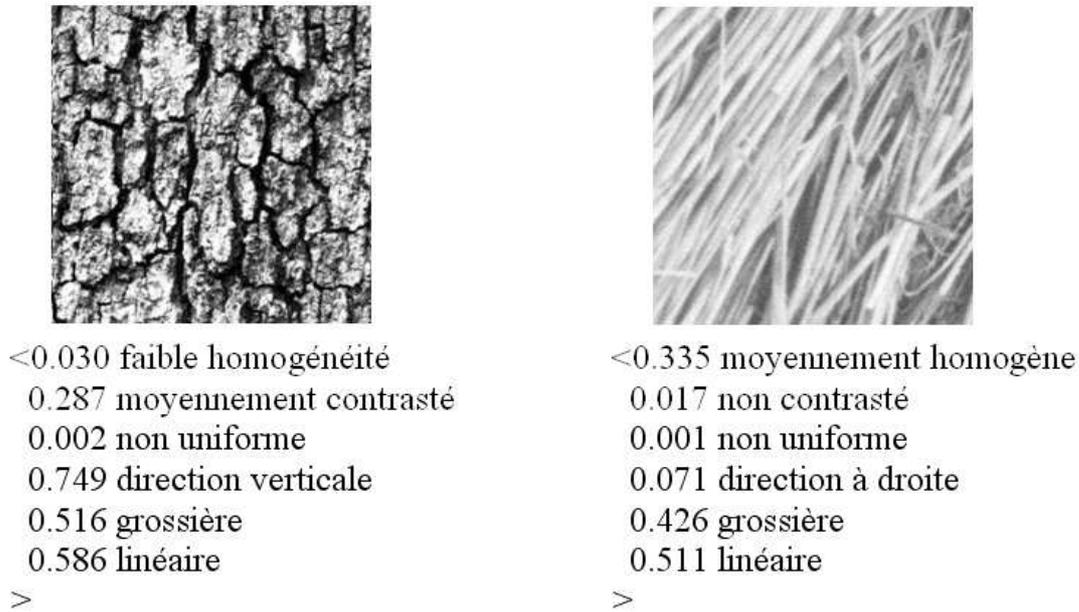


Figure 6.5 – Exemples d’interprétation « sémantique » de texture

méthode de segmentation valide pour toutes les images de la base. On se contente donc d’un découpage des images en grille régulière dont la taille des cases est de $l_p \times l_p$. Nous proposons une taille dynamique qui dépend de la taille de l’image initiale I :

$$l_p = \begin{cases} 19 & \text{si } N, M < 1024 \\ 33 & \text{si } N, M < 5Ko \\ 64 & \text{sinon} \end{cases}$$

avec N et M sont respectivement la hauteur et la largeur de l’image I .

Plus formellement, à partir d’une image I , on construit une matrice de sous-images, les « cases » $K(I)_{i,j}$:

$$K : \begin{array}{l} \mathcal{I} \rightarrow \mathcal{I} \\ I \mapsto \left\{ K(I)_{i,j} : \forall 0 \leq i < \frac{N(i)}{l_p}, 0 \leq j < \frac{M(j)}{l_p} \right\} \end{array} \quad (6.25)$$

et

$$K : \begin{array}{l} \mathcal{I} \rightarrow (\mathcal{I}) \\ I_{i,j} \mapsto \{ I_{i,j} : \forall 0 \leq c < l_p, 0 \leq l < l_p, I_{i,j}(l, c) = I(i \times l_p + l, j \times l_p + c) \} \end{array} \quad (6.26)$$

Notons qu’il en découle que :

$$|K(I)| = \frac{N(i)}{l_p} \times \frac{M(j)}{l_p}$$

Pour chaque case $K(I)_{i,j}$ d’une image I , on calcule ses propriétés sémantiques suivant l’équation 6.2. Ensuite, on s’intéresse aux fréquences d’apparition de certaines sémantiques sur l’ensemble de l’image. Il s’agit donc de dresser l’histogramme normalisé des termes linguistiques :

$$\begin{aligned}
& \mathcal{I} \rightarrow (\{G\}^{n-1} \rightarrow [0, 1]) \cup (\{G_{dir}\} \rightarrow [0, 1]) \\
h : I \mapsto & \left\{ \left(v, \frac{|\{sem(K(I)_{i,j})=v:\forall(i,j)\}|}{\frac{N(i)}{l_p} \times \frac{M(j)}{l_p}} \right) : v \in \{G\}^{n-1} \cup \{G_{dir}\} \right\} \quad (6.27)
\end{aligned}$$

Notre but est donc d'associer à chaque image I le sous-ensemble le plus pertinent (et si possible le plus petit) des textures qui la caractérise. Il suffit donc de considérer les vecteurs des cases dans l'ordre décroissant des fréquences. Une condition d'arrêt peut être que la fréquence cumulée deviennent supérieure ou égale à 50 %. Cela peut s'illustrer visuellement en reconstruisant l'image à partir des seules cases associées aux vecteurs fréquents. Un contrôle visuel par un expert humain permettra de valider cette approche heuristique image par image.

En imaginant que, en moyenne, 3 vecteurs suffisent à décrire au moins 50 % des cases, alors, toujours pour $n = 6$, il peut y avoir jusqu'à $C_{972}^3 = 152\,582\,940$ classes d'images. Même si l'on peut regretter que ces catégories soient très grossières et ne permettent pas de distinguer des images sensiblement différentes, il n'en reste pas moins qu'augmenter leur nombre amènerait à classer chaque image dans sa propre classe singleton. Les expérimentations sur la classification à partir d'un treillis de Galois permettra de valider l'approche sur une base d'images (chapitre 7).

Entrées : image réelle couleur $I(N, M)$, pas de découpage l_p

Sortie : image segmentée $I_s(N, M)$

1. **pour** chaque région de taille $l_p \times l_p$ **faire** :
 2. calculer les descripteurs de textures numériques
 3. calculer les propriétés sémantiques correspondantes
 4. incrémenter le nombre d'apparition des propriétés
 5. calculer les fréquences des propriétés sémantiques sur l'ensemble de l'image
 6. sur l'image de sortie I_s , faire apparaître la case correspondante à chaque propriété retenue $sem_i(I)$ tel que : $\sum_{i=1}^n sem_i(I) \geq \frac{(N \times M)}{2}$.
-

6.6 Conclusion

Dans ce chapitre, nous avons abordé la problématique de la réduction du fossé sémantique entre les descripteurs de texture de bas niveau et les concepts sémantiques de haut niveau liés particulièrement aux interprétations faites par les utilisateurs. Ainsi, la forte subjectivité des interprétations posent un grand problème et freinent d'une certaine manière l'avancement des systèmes de recherche d'images par contenu.

Notre proposition traite ce problème du fossé sous l'angle de la conception d'une ontologie exploitable dans un cadre général. La création d'une sémantique se fait par la discrétisation de l'espace numérique dédié à chaque descripteur de bas niveau. Chaque intervalle obtenu correspond à un terme linguistique faisant référence à une sémantique particulière. L'affectation ou le passage du numérique au sémantique se fait par les règles d'inférence définies à partir des bornes du découpage estimés par les différentes approches de discrétisation.

Notre proposition nous a permis de réduire le fossé sémantique en permettant une interprétation des textures en langage naturel. Le chapitre suivant montrera comment exploiter cette sémantique pour la navigation dans une base d'images.

La navigation dans une base d'images

Dans le chapitre précédent, nous avons proposé un moyen d'interprétation sémantique des textures qui sera mis à profit pour la navigation, utilisée comme alternative pour la recherche d'images dans des bases d'images.

L'interrogation et la recherche d'images peuvent être effectuées selon différents modes apparentés à la recherche d'information [82] :

1. *recherche par mots-clés:*

Ce type de requête est très répandu et proposé en particulier par la plupart des moteurs de recherche tels que Flickr, Google, yahoo, altavista, *etc.* Malgré que les requêtes sont proches du langage des utilisateurs, pourtant ces derniers manifestent une grande insatisfaction vis-à-vis des réponses retournées et qui peuvent être hors contexte de la requête. Cela est dû aux annotations manuelles subjectives associées aux images de la base.

2. *recherche formelle:*

Ce type de requête découle des langages d'interrogation des bases de données reposant sur les modèles traditionnels dont la norme est le langage SQL (Structural Query Language). Le système Cypress [221] (ancien Chabot) est l'un des rares systèmes qui utilisent un système de gestion de bases de données (POSTGRES) basé sur le langage SQL; L'utilisation de SQL a également été proposée pour interroger des bases de données multimédia organisées selon un modèle conceptuel enrichi [287, 206].

L'intérêt d'une telle approche, fondée sur l'utilisation d'un langage formel, est de bénéficier du pouvoir d'expression de la logique des prédicats pour décrire précisément les objets multimédia, et plus encore, de disposer du mécanisme d'appel de fonction récursive, tel qu'il existe dans OQL (Object Query Language) [33] ou SQL-3 [211]. Ainsi, la problématique soulevée concernant la difficulté d'interprétation des requêtes est partiellement résolue dans ce cas. En effet, l'utilisateur interroge le système dans un langage proche du langage machine, ce qui contribuera ainsi à augmenter la satisfaction de l'utilisateur en lien avec son besoin. Plus généralement, une approche reposant sur un SGBD traditionnel offre l'ensemble des fonctionnalités de celui-ci [209, 155], à savoir : la persistance, l'indépendance logique et physique, l'administration des données, un langage de requête, des mécanismes d'indexation, de regroupement, la régénération, le contrôle d'accès concurrents, la sécurité, l'intégrité, la confidentialité, *etc.*

Par contre, le problème majeur de cette approche est la complexité du langage SQL. Il nécessite un long apprentissage de la part de l'utilisateur, tant pour sa syntaxe que pour l'algèbre de Boole sur lesquelles il repose. De plus, pour être capable d'interroger une base de données avec ce type de langage, l'utilisateur doit savoir comment est organisée l'information.

3. *recherche par rétroaction:*

Ce type de requête permet de contourner le problème d'écriture de requêtes complexes en générant automatiquement des requêtes de plus en plus précises obtenues par l'apprentissage des annotations successives des réponses précédentes des utilisateurs [55, 332, 130, 217, 317] (cf. section 6.1.3). Bien que les expériences montrent une amélioration de la qualité des résultats, celle-ci reste limitée. De plus, cette approche, pour être efficace, nécessite un grand nombre d'interactions entre le système et l'utilisateur, propres à décourager ce dernier.

Par exemple, le système Cires [137, 136] en adoptant l'approche de recherche par raffinement de pertinence, propose aussi un certain nombre de paramètres (poids des descripteurs de couleur et texture) à régler selon les besoins des utilisateurs. Pour un utilisateur non-expert, ces informations sont lourdes à manipuler et donc s'ajoutent à la contrainte citée en haut pour décourager encore plus l'usage de tels systèmes.

4. *recherche par navigation:*

L'idée de départ est que l'utilisateur cherche un moyen simple et efficace pour effectuer sa recherche; simple ne nécessitant aucun apprentissage de quelconque langage ni formulation de requêtes notamment dans le cas où l'utilisateur lui-même ne sait pas exactement ce qu'il cherche et efficace en vue des réponses retournées aux utilisateurs qui doivent fortement satisfaire les attentes de ces derniers.

Un processus de navigation peut donc être utilisé comme solution complémentaire de ces approches pour permettre notamment d'exploiter le résultat des requêtes, ou encore une solution indépendante qui permet de parcourir librement une large collection d'images. La navigation s'opère par le biais d'hyperliens entre les images, de telle sorte que l'approche soit intuitive et qu'aucune manipulation complexe ni connaissance a priori des caractéristiques primaires ne soient requises pour l'utilisation de ce genre d'outils. De plus, les images peuvent faire l'objet de comparaisons et classifications très précises car les calculs sont effectués hors ligne.

Du point de vue utilisateur et du notre, la navigation est l'outil le plus facile et efficace pour retrouver les images. Dans la littérature, plusieurs propositions de navigation ont été suggérées [237, 175, 199, 320] et sont divisées en deux catégories :

1. des systèmes qui adoptent la navigation seulement pour l'interface dédiée aux utilisateurs alors que le processus de recherche effectif des images est basé sur les approches standards de recherche d'images similaires.
2. et d'autres qui utilisent la navigation et pour l'interaction avec les utilisateurs et pour la recherche d'images similaires.

La structuration et la visualisation de la base de données sont les principales phases d'établir une technique de navigation. Différentes sont les structures de navigation [8, 138, 92] allant du modèle simple au plus complexe. Le premier modèle simple est le modèle linéaire guidant l'utilisateur dans sa navigation par les liens *précédent* et *suivant*. Pour le second modèle non-linéaire, il est basé sur une structure hiérarchique permettant aux utilisateurs de naviguer de n'importe quel nœud (parent ou fils) correspondant à une catégorie de données vers un autre nœud présentant une autre catégorie. Dans ce travail, nous adoptons l'approche hiérarchique basée sur une structure de treillis de Galois et qui est proposée initialement par J. Martinez et E. Loissant [210]. Cette approche sera présentée dans la section 7.2.

Pour la visualisation, plusieurs travaux proposent d'effectuer une projection des données de la hiérarchie créée dans un espace de $2D$ ou $3D$, principalement par l'analyse en composante principale *ACP* [215], la projection de Sammon « multidimensional scaling(MDS) » [250, 249] ou encore les

cartes de Kohonen [162]. Dans ce travail, nous nous sommes inspirés pour la visualisation des structures des treillis de Galois de textures de la technique proposée dans [208, 210] basée sur les hypertextes. Chaque nœud du treillis correspond à une page XHTML composée de trois frames: haut, bas et centre. Les frames haut et bas contiennent respectivement les nœuds pères et fils du nœud courant. Quant à la frame centrale, elle affiche les images contenues dans ce nœud. Dans cet esprit, l'utilisateur a donc une grande visibilité de la base d'images ainsi que les différentes classes sémantiques (regroupement dans chaque nœud) des nœuds, donc plus besoin de décrire une requête, par de simples cliques sur les images (frames haut et bas), l'utilisateur peut se déplacer jusqu'à atteindre la cible recherchée.

Le reste de ce chapitre est organisé comme suit: la section suivante (7.1) introduit le méta-modèle de texture utilisé pour la navigation. Ensuite, une présentation du formalisme des treillis de Galois sera donnée en section 7.2. L'implémentation cette structure ainsi que les résultats des expérimentations seront présentées respectivement dans les sections 7.4 et 7.5.

7.1 Modèle de texture pour la navigation

Dans cette partie, nous allons présenter le modèle de texture pour la navigation dans une structure de treillis de Galois dans le cas d'images texturées standards et dans le cas des images réelles tout en se basant sur le modèle sémantique proposé dans le chapitre précédent.

7.1.1 Cas des images standards

7.1.1.1 Modèle stricte

Selon l'équation 6.1, chaque descripteur de texture est caractérisé par une seule propriété sémantique, dans ce cas, une image \mathcal{I} sera exprimée par un seul vecteur linguistique unique, qu'on note SSL (Semantic Strict Labels), définit par la conjonction (l'union) des différentes propriétés sémantiques estimées sur chaque descripteur individuel. Formellement, la représentation linguistique d'une texture sera la suivante :

$$SSL(\mathcal{I}) = \left\{ \begin{array}{l} \{faible, moyen, fort\} \times \left(\begin{array}{l} \text{homogeneite} \wedge \text{contraste} \wedge \text{uniformite} \\ \wedge \text{grossierete} \wedge \text{linarite} \end{array} \right) \\ \wedge \text{direction} \times \{verticale, horizontale, droite, gauche\} \end{array} \right.$$

(7.1)

Taille de l'espace de description Dans notre cas, chaque descripteur a trois possibilités $\mathcal{P}(T_i)$ d'être exprimé par un terme linguistique sauf pour la direction qui en génère quatre. Par conséquent, la taille de l'espace de description des textures est donnée par la formule suivante :

$$\prod_{i \in \{\text{homogeneite}, \text{contraste}, \text{uniformite}, \text{grossierete}, \text{linearite}\}} \mathcal{P}(T_i) \times \mathcal{P}(T_{dir}) \quad (7.2)$$

Numériquement, la valeur maximale des différentes possibilités pour exprimer une texture quelconque est de $(3^5) \times 4 = 972$. Ce chiffre, même un peu loin du nombre de textures qui existent dans la

nature, montre la capacité de notre approche à caractériser un nombre important de classes de textures différentes.

7.1.1.2 Modèle flou

Dans le cas flou, chaque terme linguistique flou *SFL* (Semantic Fuzzy Labels) attribué à un descripteur f_j , il lui est associé une fonction d'appartenance μ_j à valeurs dans $[0, 1]$ reflétant le degré d'appartenance de cet attribut dans une texture :

$$\mu_j : \begin{array}{l} \mathcal{I} \rightarrow [0, 1] \\ I \mapsto \mu_j(I) \end{array} \quad (7.3)$$

Par conséquent, la fonction globale d'appartenance d'une texture est alors obtenue par conjonction des fonctions d'appartenance des descriptions linguistiques élémentaires :

$$\mu(I) = \otimes_j \mu_j(I)$$

$$\forall j \in \{\text{homognit, contraste, uniformit, direction, grossiret, linarit}\}$$

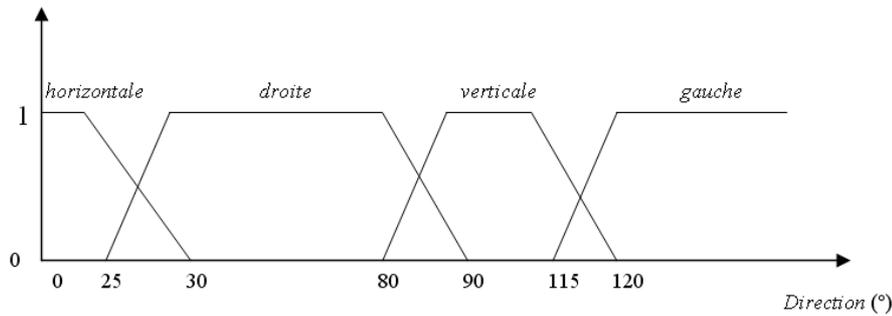


Figure 7.1 – Variables linguistiques définies pour le descripteur *direction*

La figure 7.1 présente un exemple de variables linguistiques floues définies pour le descripteur *direction*. Si la direction d'une texture est de 85, elle sera décrite à la fois par *direction à droite* et *verticale* avec des degrés de satisfaction différents.

La taille de l'espace de description de textures dans la cas flou est donnée par :

$$(A_{np}^1)^{np} \times 2^n$$

Le premier terme calcule le nombre de propriétés possibles (np) pour chaque descripteur et le second donne le nombre de combinaisons possibles pour les n descripteurs. Dans notre cas, pour $n = 6$ et $np = 3$, la taille maximale de l'espace de description flou atteint 1728 classes. Le nombre de description a considérablement augmenté reflétant la dimension importante des textures.

7.2 Navigation sur une structure de treillis de Galois

La représentation sémantique des images étant définie, il s'agit dans cette section de décrire le processus d'organisation et de structuration de la base d'images pour la navigation et la recherche d'images.

J. Martinez et E. Loissant dans [210] ont proposé une méthode pratique, simple et intuitive de navigation au sein d'une collection d'images liées entre elles par des propriétés sémantiques qu'elles peuvent partager ou non. Cette approche, proposée en tant que module du projet $Find_{AGE}^{Im}$ et baptisée $Click_{AGE}^{Im}$, repose sur le schéma conceptuel « treillis de Galois » basé sur l'attribut *couleur* pour organiser les bases d'images. Dans cette thèse, nous étendons ce travail initié à prendre en compte les attributs de texture en se basant sur le modèle sémantique proposé dans le chapitre précédent.

Aussi, une étude récente [218] propose une comparaison de différentes méthodes de classification supervisée basées sur un treillis de Galois, où les expérimentations présentées montrent clairement que le treillis de Galois offre aussi un cadre intéressant en classification malgré une complexité exponentielle dans le pire des cas.

7.2.1 Présentation du formalisme des treillis de Galois

Un treillis de Galois (ou treillis de concepts) est une structure mathématique permettant de représenter les classes non disjointes sous-jacentes à un ensemble d'objets (images) décrits par un ensemble d'attributs (propriétés). Ces classes non disjointes sont aussi appelées concepts formels, hyper-rectangles ou ensembles fermés. Chaque classe correspond à un concept qui donne une idée générale sur les objets de cette classe. Ce concept peut être défini formellement par une *extension* (exemples du concept) ou par une *intention* (propriétés du concept).

Définition 3. Soit \mathcal{R} une relation d'ordre définie sur un ensemble \mathcal{E} :

- $\forall x, y \in \mathcal{E}$, si $x\mathcal{R}y$, alors on dira que x minore y ou y majore x .
- \mathcal{E} est un ensemble ordonné s'il est muni d'une relation d'ordre R .
- Soit $\mathcal{B} \subset \mathcal{E}$, a est un majorant (resp. minorant) de \mathcal{B} , s'il majore (resp. minore) tous les éléments de \mathcal{B} .
- Soit $\mathcal{B} \subset \mathcal{E}$, le plus petit majorant (resp. le plus petit minorant) de \mathcal{B} , s'il existe est appelé borne supérieure notée sup^1 (resp. borne inférieure inf²) de \mathcal{B} .

Définition 4. Contexte. Un contexte au sens de R. Wille [314] est un triplet (O, A, R) où O, A sont des ensembles et $R \subseteq O \times A$ une relation binaire entre O et A .

Définition 5. Correspondance de Galois. Soient E_1 et E_2 deux ensembles ordonnés. Soit p (resp. q) une application monotone décroissante définie sur E_1 dans E_2 (resp. E_2 dans E_1) telle que soit $x \in E_1$ et $y \in E_2$: $y \leq p(x)$ si et seulement si $x \leq q(y)$. Le couple d'applications (p, q) est appelé correspondance de Galois entre E_1 et E_2 .

Définition 6. Applications f et g d'un contexte binaire. Soient f et g deux applications définies comme suit :

- $\forall o \in O$, on associe $f(o) = \{a \in A; (o, a) \in R\}$;
- $\forall a \in A$, on associe $g(a) = \{o \in O; (o, a) \in R\}$;

Proposition 1 f et g sont des applications monotones décroissantes.

¹le nœud *sup* sera le plus grand ensemble de propriétés associé au plus petit ensemble d'images.

²le nœud *inf* sera le plus petit ensemble de propriétés associé au plus grand ensemble d'images.

Remarque 1 f et g sont des étendues respectivement aux parties de O et de A comme suit : soient $O_1 \subseteq O$ et $A_1 \subseteq A$:

$$f(O_1) = \bigcap_{o \in O_1} f(o)$$

et

$$g(A_1) = \bigcap_{a \in A_1} g(a)$$

Proposition 2 Le couple (f, g) est une correspondance de Galois entre les ensembles ordonnés $P(O)$ et $P(A)$ du contexte (O, A, R) .

Définition 7. Concept : Intention/Extension. Soient $O_1 \subseteq O$ et $A_1 \subseteq A$. La paire $\{A_1, O_1\}$ est un concept formel si $f(O_1) = A_1$ et $g(A_1) = O_1$. A_1 (resp. O_1) est appelé l'intention (resp. extension) du concept $\{A_1, O_1\}$.

Définition 8. L'ordre sur les concepts. La relation de sous-concept/sur-concept (dénotée \leq^3) est la relation d'ordre définie entre les concepts de la façon suivante: pour deux concepts $\{A_1, O_1\}$ et $\{A_2, O_2\}$, $\{A_1, O_1\} \leq \{A_2, O_2\} \Leftrightarrow O_1 \subseteq O_2$ ($\Leftrightarrow A_2 \subseteq A_1$).

$\{A_1, O_1\}$ (resp. $\{A_2, O_2\}$) est un sous-concept (resp. sur-concept) de $\{A_2, O_2\}$ (resp. $\{A_1, O_1\}$).

Définition 9. Treillis de Galois. Un treillis de Galois est un ensemble de concepts muni d'une relation d'ordre R (i.e reliés par inclusion).

Dans la définition formelle du treillis de Galois, les ensembles O et A n'ont pas de rôle particulier. Dans ce travail, nous considérons l'ensemble O comme l'ensemble des images de la base et l'ensemble A sera celui des propriétés sémantiques extraites des images. Pour un concept (O_1, A_1) , l'ensemble O_1 définit l'extension du concept, i.e l'ensemble des images qu'il décrit et l'ensemble A_1 en définit l'intention, i.e l'ensemble des propriétés sémantiques partagées par toutes les images de O_1 .

R	a ₁	a ₂	b ₁	b ₂	c ₁	c ₂
1	1	0	1	0	0	1
2	1	0	1	0	0	1
3	1	0	0	1	0	1
4	1	0	0	1	0	1
5	1	0	0	1	0	1
6	0	1	0	1	0	1
7	0	1	0	1	0	1
8	0	1	0	1	0	1
9	0	1	1	0	1	0
10	0	1	0	1	1	0

Table 7.1 – Exemple de relation R binaire

Le tableau 7.1 donne un exemple d'une relation R binaire avec $O = \{1, 2, \dots, 10\}$ et $A = \{a_1, a_2, b_1, b_2, c_1, c_2\}$. La représentation graphique du treillis de Galois correspondant est donnée par la figure 7.2. Les arcs

³à lire « moins générale que »

représentent des relations d'ordre (\leq) directes, i.e qu'un arc est placé entre les sommets $\{A_1, O_1\}$ et $\{A_2, O_2\}$ si $\{A_1, O_1\} \leq \{A_2, O_2\}$ et qu'il n'exsiste pas de concept $\{A_3, O_3\}$ tel que l'on ait $\{A_1, O_1\} \leq \{A_2, O_2\} \leq \{A_3, O_3\}$.

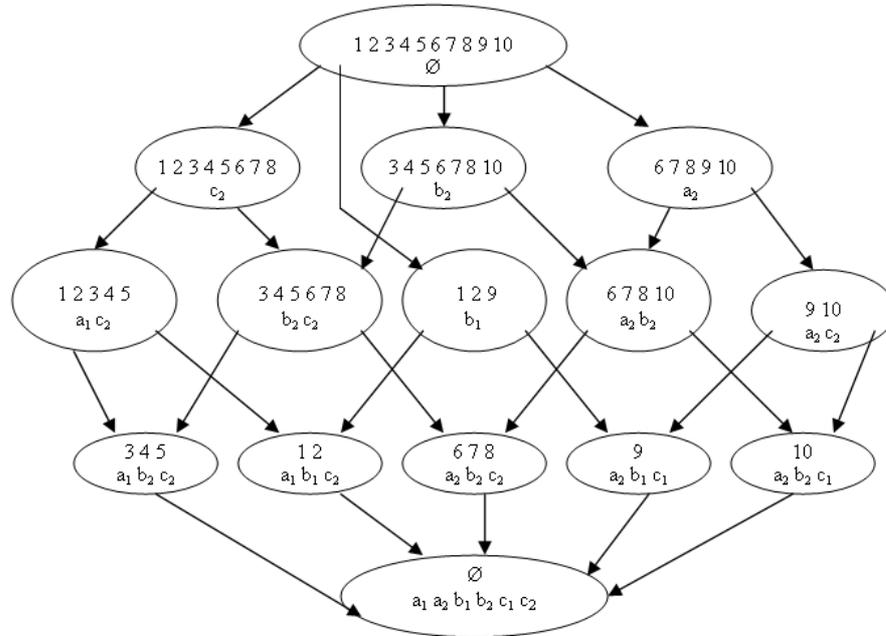


Figure 7.2 – Treillis de Galois pour la relation R

Exemple dans le cas des textures dans le cas des textures par exemple, soit R une relation telle que : $R = (\text{img1}, \text{moyennement homogène}), (\text{img1}, \text{faible contraste}), (\text{img2}, \text{moyennement homogène}), (\text{img3}, \text{faible contraste}), (\text{img3}, \text{lisse}), (\text{img4}, \text{faible contraste}), (\text{img4}, \text{lisse})$.

On dérive f et g : $f(\text{img1}) = \text{moyennement homogène}, \text{faible contraste}$, $f(\text{img2}) = \text{moyennement homogène}$, $f(\text{img3}) = \text{faible contraste}, \text{lisse}$ et $f(\text{img4}) = \text{faible contraste}, \text{lisse}$.
et inversement :

$$g(\text{moyennement homogène}) = \text{img1}, \text{img2}, g(\text{faible contraste}) = \text{img1}, \text{img3}, \text{img4}, g(\text{lisse}) = \text{img4}.$$

Le treillis de Galois correspondant est donnée par la figure 7.3.

D'autres représentations existent, en particulier les treillis d'héritage également appelés diagrammes de Hasse qui permet d'éviter la redondance dans les notations des nœuds du graphe en remarquant simplement que pour un concept $\{A, O\}$, l'ensemble A est inclus dans tous les ascendants de $\{A, O\}$ tandis que O est inclus dans tous les descendants (au sens de la relation d'ordre). Ces diagrammes représentent une notion intuitive de généralisation/spécification entre les concepts où chaque concept courant supérieur généralise un ensemble de concepts inférieurs en étant moins spécifique dans son intention par l'abandon de l'une de ses propriétés, ce qui lui permet de décrire davantage des exemples (images).

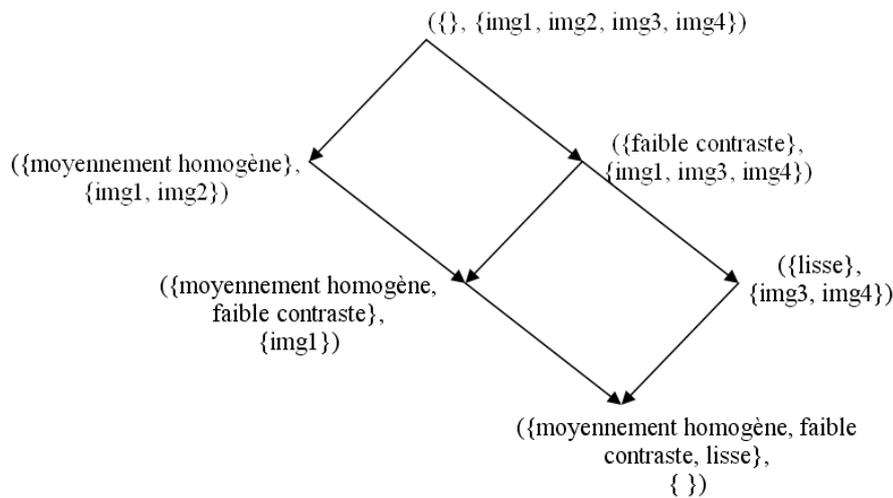


Figure 7.3 – Un exemple de treillis de Galois pour la texture

7.2.2 Construction d'un treillis de Galois

Dépendant d'une relation binaire (ce qui n'est pas toujours le cas des relations dans divers applications), la construction des treillis de Galois est un problème non-triviale et plusieurs recherches s'y intéressent.

L'algorithme non-incrémental *NextClosure* introduit par B. Ganter [90] est considéré comme la référence standard à laquelle différents algorithmes se comparent. D'autres algorithmes incrémentaux existent pour la construction d'un treillis de Galois tel que l'algorithme de Godin et *al.* [97] dont sa complexité algorithmique est de $O(n^2)$. Récemment, Levy et Baklouti [5, 180] proposent un algorithme parallèle nommé ELL. Cet algorithme est utilisé pour la construction de large treillis à l'aide d'un cluster de machines. Cependant, cet algorithme n'est pas adapté à notre cas puisqu'il est non-incrémental. Pour ce travail, nous adoptons l'algorithme de Godin [97] pour la construction des treillis de Galois pour deux raisons suivantes :

1. L'aspect incrémental qui permet de mettre à jour la hiérarchie et ainsi ajouter de nouvelles images;
2. Sa complexité acceptable, même de l'ordre $O(n^2)$, comparée aux autres algorithmes.

7.2.3 Avantages et limites des treillis de Galois

Pour notre cas, la navigation au sein d'une base d'images généralistes, les avantages des treillis de Galois sont :

- Le principale avantage est que la navigation au sein d'un graphe crée à l'avance ne demande ni requêtes formelles ni des aller-retour entre système et utilisateur; ainsi la recherche d'une image cible est très rapide. Si on néglige le temps requis pour charger une image, naviguer d'un nœud à un autre est optimal, soit en $O(1)$.
- Un treillis de Galois est intrinsèquement une technique de classification multi-dimensionnelle. En effet, aucune dimension n'est privilégiée. Cette structure peut donc être vue comme une manière de catégorisation de l'ensemble de la base d'images associées aux sous-ensembles de propriétés.
- Un treillis de Galois est insensible aux corrélations puisqu'il n'y a pas de distance à calculer. Si deux attributs sont corrélés, ils seront regroupés dans le même nœud.

- Un treillis de Galois permet aux utilisateurs de corriger facilement leurs erreurs. Lorsqu'un utilisateur sélectionne un nœud descendant, il ajoute implicitement une propriété et donc une contrainte en réduisant le nombre d'images. S'il se rend compte, après, que ce choix est plutôt mauvais (il ne trouve pas les images cibles), il a simplement à se diriger vers un autre nœud père. Cette opération supprime une contrainte et annule le choix erroné sans avoir à revenir au point de « l'erreur » (ce qui est le cas d'une structure linéaire).
- La structure du treillis de Galois cache facilement les propriétés non souhaitées (ou négligeables) ce qui n'est pas le cas des mesures de similarité. Surfimage [217] est un contre-exemple mais les mesures de similarité sont limitées à la moyenne et l'écart-type des distributions supposées Gaussiennes.

Toutefois, les treillis de Galois ont aussi certaines limites qui résident en la taille exponentielle de l'espace de description suivant le nombre de propriétés et la complexité en temps qui est de $O(n^2)$ avec n est le nombre de nœuds.

7.3 Visualisation

La seconde phase dans le processus de navigation après construction de la hiérarchie est la visualisation de la base d'images. Plusieurs travaux existent reposant sur la projection de la base dans un espace $2D$ ou $3D$ soit par l'analyse en composante principale *ACP* [215], par la projection de Sammon « multidimensional scaling(MDS) » [250, 249] ou par encore les cartes de Kohonen [162].

Dans ce travail et celui de E. Loisant [191], les treillis de Galois ne sont pas utilisés juste pour la structuration de la base d'images mais aussi pour sa visualisation. Un ensemble de pages XHTML est construit à partir des treillis de Galois et stocké dans la mémoire centrale. Cet ensemble est généré dès lors qu'une nouvelle image est insérée dans la hiérarchie. L'utilisateur par la suite peut naviguer à travers ces pages statiques avec un temps de navigation optimal ($O(1)$) (premier avantage cf. section 7.2.3).

Chaque page XHTML de cet ensemble correspond à un nœud du treillis de Galois associé à un ensemble de propriétés (intention du nœud) et un ensemble d'images (extension du nœud) et composée de trois frames : haut, bas et centre. Les frames haut et bas contiennent respectivement les nœuds pères et fils du nœud courant. La représentation des nœuds de ces frames n'est pas globale mais réduite. Cela veut dire que seules les images partageant seulement l'ensemble des propriétés du nœud réduit sont affichées. Pour la page principale (index.html), seules les images ayant les propriétés dominantes de chaque nœud sont affichées (celles ayant la plus grande fréquence d'apparition), cela est très utile lors du chargement d'une base d'images volumineuses. Cette propriété et celle du contenu réduit sont ajoutées dans l'algorithme Godin [97] sans altérer sa complexité. Quant à la frame centrale, elle affiche d'une manière étendue l'ensemble des images contenues dans ce nœud.

Dans cet esprit, l'utilisateur a donc une grande visibilité de la base d'images ainsi que les différentes classes sémantiques(intention des nœuds) des nœuds, donc plus besoin de décrire une requête, par de simples cliques sur les images (frames haut et bas), l'utilisateur peut se déplacer jusqu'à atteindre la cible recherchée.

7.4 Implémentation de la structure

Une présentation simple de l'algorithme de construction des treillis de Galois à partir d'une base d'images basés sur les attributs de texture (*metadata*) est donnée dans l'algorithme 11 :

Algorithme 11 Construction d'un treillis de Galois

1. $structure \leftarrow \emptyset$, $G =$ faible, moyen, fort, $G_{dir} =$ horizontale, verticale, à droite, à gauche
2. $\forall I \in \mathcal{I}$
3. $metadata \leftarrow \emptyset$
4. $metadata.Insert(G + " homogénéité ");$
5. $metadata.Insert(G + " contraste ");$
6. $metadata.Insert(G + " uniformité ");$
7. $metadata.Insert(G + " grossièreté ");$
8. $metadata.Insert(G + " linéarité ");$
9. $metadata.Insert(" direction " + G_{dir});$
10. $structure.Insert(I, metadata)$

7.5 Quelques résultats

Les expérimentations menées ici concernent les bases de textures standards ainsi que des bases d'images réelles. Pour les bases de texture, nous utiliserons toujours la base Brodatz et Meastex (cf. section 3.2). Pour les bases d'images réelles, nous disposons de 3 bases de taille respectivement 50, 50 et 1000 images. Les deux premières bases sont extraites aléatoirement de la toile tandis que la dernière correspond à la base de Wang⁴ qui est un sous-ensemble de la base de Corel⁵. Cette base est créée par le groupe du professeur Wang de l'université de Pennsylvania. Les images sont sélectionnées manuellement afin d'obtenir 10 classes de 100 images de taille 384×256 ou 256×384 pixels. La taille de la base Wang ainsi que l'information disponible sur les étiquettes des différentes classes de la base la rendent plus répandue et plus utilisée particulièrement pour la classification des images. Les 10 classes sont : *africains, mer, monuments, bus, dinosaures, éléphants, fleurs, chevaux, montagnes, alimentation*.

Le tableau 7.2 illustre la taille des treillis de Galois construit pour chaque base d'images en tenant compte des propriétés sémantiques extraites d'après le modèle de discrétisation stricte.

Base	# images	# nœuds
Meastex	60	66
Brodatz 1	158	109
Brodatz 2	111	109
Brodatz 1	60	66
Base réelle 1	50	87
Base réelle 2	50	54
Corel	1000	395

Table 7.2 – Statistiques des différentes bases d'images

La figure 7.4 illustre le nombre de propriétés dominantes pour des images généralistes. Un survol sur l'ensemble des trois bases réels nous a permis de soutenir qu'en maximum une image réelle (plusieurs

⁴Disponible sur le site <http://wang.ist.psu.edu>

⁵Disponible sur le même site.

textures) peut être exprimée par 4 ensembles sémantiques (50 % de l'image est affichée).



Figure 7.4 – Nombre de propriétés par image

Nous laissons le soin à l'utilisateur pour vérifier l'efficacité de l'approche pour une navigation rapide à l'aide des treillis de Galois en se connectant à la page: <http://idrissi-nj.freehosia.com>

CHAPITRE 8

Conclusions

L'extraction des descripteurs visuels et la réduction du fossé sémantique sont des problèmes d'actualité complexes et incontournables pour la recherche d'images notamment dans le cas des bases généralistes.

Dans ce travail, nous nous intéressons d'une part à l'extraction des descripteurs pertinents de la texture et d'autre part à la réduction du fossé sémantique pour une recherche plus efficace et rapide.

Après un état de l'art sur les principes de la recherche d'images par contenu ainsi que les systèmes académiques et commerciaux les plus répandus dans le chapitre 2. Nous dressons dans le chapitre 3, le squelette du processus de description de la texture en s'appuyant sur le processus d'extraction de connaissances à partir des données (ECD).

L'extraction

La texture est l'un des attributs visuels important utilisé largement dans la recherche, la segmentation et la classification des images malgré la difficulté de trouver une définition globale. Plusieurs approches existent dans la littérature pour la caractérisation de la texture. Vu nos contraintes à savoir que les attributs doivent être :

1. invariants par translation et rotation;
2. discriminants entre différentes classes de texture;
3. la dimension de l'espace des attributs doit être minimale;
4. et traduisibles en langage naturel.

Nous avons choisi deux approches qui sont les matrices de cooccurrence et l'approche de Tamura.

Pour l'extraction des descripteurs pertinents, nous avons proposé et dressé un ensemble d'algorithmes de sélection des attributs qui sont évalués en terme des courbes de rappel/précision et une mesure de score proposée pour la recherche d'images et aussi par les taux de classification et matrices de confusion dans le cas de la classification supervisée.

Dans le chapitre 5, les résultats obtenus pour notre modèle (PaCFS) basé sur la corrélation et la mesure de taux de classification pour la sélection du meilleur sous-ensemble sont satisfaisants et encourageants.

Le fossé sémantique

La réduction du fossé sémantique est un problème qui s'accroît de jour en jour vu la difficulté des systèmes bas-niveau à comprendre les attentions des utilisateurs et prendre en compte leurs sémantiques.

Dans ce travail, nous nous sommes focalisés dans le chapitre 6 sur la réduction de ce fossé sous l'angle d'une discrétisation des différents attributs en intervalles dont chaque intervalle représente une classe sémantique et pour l'utilisateur et pour le système. Nous avons proposé l'approche ϵ -distance_{max},

basée sur la distance maximale entre valeurs successives pour une initialisation des bornes de découpage et le jugement de l'expert pour la validation des classes, qui a montré une concurrence à l'approche k -means dans le cas de la discrétisation stricte. Pour une discrétisation floue, nous nous sommes contenté des résultats de l'algorithme *FCM*. Ainsi, en assignant à chaque intervalle un terme linguistique (propriété sémantique), nous avons contribué à la réduction du fossé sémantique en permettant à l'utilisateur d'exprimer sa requête par une conjonction de ces propriétés.

Une approche à laquelle nous nous sommes intéressés dans le chapitre 7 et qui contribue de notre point de vue à réduire considérablement le fossé sémantique est l'approche basée sur la navigation dans une structure hiérarchique des treillis de Galois.

La navigation dans une structure préalablement construite permet une réduction du temps de réponse étant donné qu'aucun calcul ne se fait en ligne; de plus l'utilisateur n'a plus besoin ni de formules ni des mots-clés pour exprimer ses requêtes, il lui suffit de sélectionner un nœud à travers les pages XHTML générées par le treillis pour parcourir toute la base en simple cliques et ainsi atteindre la cible. Ce processus de navigation a aussi un second rôle qui est la classification ou la catégorisation des classes des bases d'images. Chaque ensemble d'images partageant un certain nombre de propriétés constitue une catégorie d'images.

Les limites

Nous avons constaté que dans le cas réel des images généralistes où plusieurs textures partagent la même image, l'approche de segmentation grossière en grille que nous avons proposé fournit des résultats modestes pour la classification et la recherche d'images. Cela peut être résolu par une approche de segmentation plus précise telle la segmentation par régions ou par points d'intérêts.

La seconde limitation concerne la structure de navigation à l'aide des treillis de Galois telle qu'elle a été présentée dans le chapitre 7 qui n'est applicable que dans le cas des bases d'images stables (collections individuelles par exemple). Les bases d'images qui s'incrémentent fréquemment ne sont pas prises en compte.

8.1 Perspectives

Concernant les futurs travaux, plusieurs perspectives sont ouvertes et complémentaires au travail réalisé.

Cas d'images réelles Nous avons constaté que dans le cas d'images de scènes réelles, une segmentation grossière entraîne une grande perte d'information et donc une mauvaise reconnaissance et interprétation sémantique. Un choix plus rigoureux d'une approche de segmentation telle que les points d'intérêts ou approches morphologiques pourrait améliorer considérablement l'identification des zones texturées et offrir une possibilité d'une meilleure d'interprétation des images.

Fusion d'attributs Le deuxième axe consiste à utiliser d'avantage d'autres attributs visuels pour améliorer la caractérisation et l'interprétation des images tels que la couleur et la forme. Ces informations étant hétérogènes, des approches de fusion de données pourront être bien étudiées afin d'optimiser la représentation et la recherche des images.

Cas de larges bases d'images Un autre champ sur lequel nous travaillons actuellement est d'étendre ce travail notamment la navigation dans le cas de larges bases d'images. Le processus de génération des concepts sémantiques par les treillis de Galois sur ClickImAGE s'accroît exponentiellement dans le cas de larges bases d'images. Afin de réduire cette explosion, un processus de création de résumés en premier lieu se voit nécessaire. Il s'agit de créer des résumés à partir de la base initiale d'images. Ensuite, pour chaque résumé de la hiérarchie, identifier les concepts sémantiques lui correspondant en réduisant ainsi le temps de génération de concepts. Les tests que nous avons effectué jusqu'à maintenant sur des bases de taille respectivement 10.000 et 30.000 images montrent l'intérêt de telle approche. Nous utilisons pour la création des résumés le système SaintEtiQ¹ développé par Régis Saint-Paul au sein de l'équipe Altas-GRIM du laboratoire LINA.

... et le texte? Un autre axe de recherche aussi intéressant est la combinaison des approches textuelles et celles proposées pour améliorer l'interprétation sémantique des images. Des travaux provenant du domaine de la recherche documentaire ou TALN pourraient apporter des informations supplémentaires.

D'autres applications et travaux peuvent être envisagés tels que :

Application pair-à-pair Le partage de fichiers et particulièrement les images est l'une des applications les plus répandues sur le web notamment en utilisant la technologie des P2P (pair-à-pair). Cette dernière permet un partage de données entre personnes connectées sur le réseau tout en gardant son anonymat; ce qui fait l'avantage de cette technologie.

Disposant d'une base d'images sur chaque nœud du réseau, nous croyons que les treillis de Galois constituent une bonne approche pour l'organisation et la navigation de cette base. Ainsi, chaque utilisateur d'un nœud du réseau pourra se connecter sur n'importe quel autre nœud et visualiser sa base hiérarchisée. Pour plus de cohérence et maintenance de l'ensemble du réseau à jour, nous suggérons d'appliquer un algorithme de réplication préventive. Cela assure la cohérence des bases après chaque nouvel arrivage des images. Les utilisateurs du réseau auront davantage une vision globale des images des différentes bases au lieu d'avoir une après l'autre. Ce qui réduit considérablement le temps de recherche et le temps de réception des images.

Moteur de recherche basé sur le contenu Les moteurs de recherche actuels tels que Google, altavista, Flickr, *etc.* reposent essentiellement sur les annotations textuelles pour la recherche d'images ce qui rendent leurs utilisations des fois non-souhaitable. Comme perspective, nous proposons de créer un moteur de recherche ou intégrer dans un existant les résultats obtenus dans la section de la recherche d'images par contenu avec l'ajout éventuellement d'autres descripteurs visuels tels que la couleur. L'interface dédiée à l'utilisateur maintient la phase de soumission de mots-clés mais la recherche efficace est basée sur le contenu et non plus les annotations.

Une autre amélioration pourra concerner l'affichage des résultats. Au lieu de l'affichage standard actuel, nous suggérons d'intégrer les treillis de Galois pour la structuration de la base. Ainsi, lors de l'affichage des résultats, le nœud central correspondra à la réponse exacte de la requête et les nœuds supérieurs et inférieurs contiennent d'autres images de la base. L'utilisateur aura la possibilité de changer sa requête en naviguant dans la structure ou bien reprendre par un nouveau mots-clé s'il n'est pas du tout satisfait.

¹<http://www.simulation.fr/seq/>

Bibliographie

- [1] *Exploratory image databases: content-based retrieval*. Academic Press series in communications, networking and multimedia, Academic Press, Duluth, MN, 2001.
- [2] VATHY-FOGARASSY, B. FEIL et J. ABONYI. Minimal spanning tree based fuzzy clustering. In *Proceedings of World Academy of Science, Engineering and Technology*, volume 8, October 2005.
- [3] C. APTE, F. DAMERAU et S. M. WEISS. Text mining with decision trees and decision rules, 1998. Disponible à l'adresse citeseer.ist.psu.edu/apte98text.html.
- [4] Jeffrey R. BACH, Fuller CHARLES, Amarnath GUPTA, Arun HAMPAPUR, Bradpu Horowitz JAIN et Chiao fe SHU. The virage image search engine: An open framework for image management. In *Proceedings SPIE Conference on Visual Communication and Image Proceedings (SPIE96)*, pages 76–87, San Jose, CA, février 1996.
- [5] F. BAKLOUTI, G. LÉVY et R. EMILION. A fast and general algorithm for galois lattices building. *Journal of Symbolic Data Analysis*, 3:19–31, 2005.
- [6] D. H. BALLARD et C. M. BROWN. Principles of animate vision. *CVGIP: Image Underst.*, 56(1):3–21, 1992.
- [7] W. BANZHAF, F. D. FRANCONI, R. E. KELLER et P. NORDIN. *Genetic programming: an introduction: on the automatic evolution of computer programs and its applications*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.
- [8] F. BARBEAU et J. MARTINEZ. About tours in the othy hypermedia design. In *Proceedings of the 5th International Computer Science Conference: Internet Applications (ICSC'99)*, volume 1749, pages 146–155, London, UK, December 13-15 1999. in Lecture Notes in Computer Science (LNCS), Springer-Verlag.
- [9] R. BATTITI. Using the mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5:537–550, 1994.
- [10] Norbert BECKMANN, Hans-Peter KRIEGEL, Ralf SCHNEIDER et Bernhard SEEGER. The R*-tree: an efficient robust access method for points and rectangles. In Hector GARCIA-MOLINA et H. V. JAGADISH, réds., *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data*, pages 322–331, Atlantic City, NJ, mai23-25 1990. ACM Press.
- [11] A. M. BENSaid, L. O. HALL, J. C. BEZDEK, L. P. CLARKE, M. L. SILBINGER, J.Ã. ARRIGTON et R. F. MURTAGH. Validity-guided (re)clustering with applications to image segmentation. *IEEE Transaction on Fuzzy Systems*, 4(2):112–123, 1996.
- [12] S. BERCHTOLD, D. A. KEIM et H.-P. KRIEGEL. The X-tree: An index structure for high-dimensional data. In *22nd International Conference on Very Large Data Bases (VLDB)*, pages 28–39, Mumbai (Bombay), India, septembre 1996.
- [13] T. BERK, L. BROWNSTON et A. KAUFMAN. The cns color naming system. Rapport technique, Florida International University, Mathematical Sciences Department, Technical Report Number FIU MAS-81-1., Miami, FL, 1981.
- [14] T. BERK, L. BROWNSTON et A. KAUFMAN. A new color-naming system for graphics language. *IEEE Comput. Graphics Appl.*, 2(3):37–44, 1982.

- [15] M. J. BERRY et G. LINOFF. *Data Mining Techniques: For Marketing, Sales, and Customer Support*. John Wiley & Sons, Inc., New York, NY, USA, 1997.
- [16] J. C. BEZDEK. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981.
- [17] J. C. BEZDEK et J. C. DUNN. Optimal fuzzy partitions: A heuristic for estimating the parameters in a mixture of normal distributions. *IEEE Trans. Comput.*, 24(8):835–838, 1975.
- [18] J. C. BEZDEK, R. EHRLICH et W. FULL. Fcm: Fuzzy c-means algorithm. *Computers and Geoscience*, 10(2-3):191–203, 1984.
- [19] J. C. BEZDEK et N. R. PAL. Cluster validation with generalized dunn’s indices. In *ANNES '95: Proceedings of the 2nd New Zealand Two-Stream International Conference on Artificial Neural Networks and Expert Systems*, page 190, Washington, DC, USA, 1995. IEEE Computer Society.
- [20] Christophe BIERNACKI et Roger MOHR. Indexing and matching images with a gaussian mixture model of colors. Technical Report RR-3600, INRIA -Rhone-Alpes, 1999.
- [21] M. BILENKO, S. BASU et R. J. MOONEY. Integrating constraints and metric learning in semi-supervised clustering. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, volume 11, pages 81–88, New York, NY, USA, 2004. ACM.
- [22] Avrim L. BLUM et Pat LANGLEY. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271, 1997.
- [23] N. BOUJEMAA, J. FAUQUEUR, M. FERECATU, F. FLEURET, V. GOUET, B. L. SAUX et H. SAHBI. Ikona for interactive specific and generic image retrieval. In *Proceedings of the International workshop on Multimedia Content-Based Indexing and Retrieval (MMCBIR'01)*, Rocquencourt, France, 2001.
- [24] P. BOUQUET, R. BRUNELLI, J.-P. CHANOD, C. NIEDERÉE et H. STOERMER. Bridging the semantic gap in multimedia information retrieval: Top-down and bottom-up approaches. In *3rd European Semantic Web Conference*, Budva, Montenegro, 12 June 2006.
- [25] J.-M. BOUROCHE et G. SAPORTA. *L'analyse des données*. Que sais-je ? Presses Universitaires de France, Paris, France, 1980.
- [26] J.-M. BOUROCHE et G. SAPORTA. *L'analyse des données, pour la science*. Les mathématiques sociales, 1999.
- [27] Leo BREIMAN, J. H. FRIEDMAN, R. A. OLSHEN et C. J. STONE. *Classification and Regression Trees*. Wadsworth, 1984.
- [28] P. BRODATZ. *A Photographic Album for Artists & Designers*. Dover, 1966.
- [29] E. BROEK, P. M. F. KISTERS et L. G. VUUPIJL. Content-based image retrieval benchmarking: Utilizing texture categories and texture distributions. *Journal of Imaging Science and Technology*, 49, 2005.
- [30] D. CAI, X. HE, Z. LI, W.-Y. MA et J.-R. WEN. Hierarchical clustering of www image search results using visual, textual and link information. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 952–959, New York, NY, USA, 2004. ACM.
- [31] Chad CARSON, Megan THOMAS, Serge BELONGIE, Joseph M. HELLERSTEIN et Jitendra MALIK. Blobworld: A system for region-based image indexing and retrieval. In *Proceedings of the 3rd International Conference on Visual Information and Information Systems (VISUAL'99)*, pages 509–516, London, UK, 1999. Springer-Verlag.

- [32] J. CATLETT. On changing continuous attributes into ordered discrete attributes. In *EWSL-91: Proceedings of the European working session on learning on Machine learning*, pages 164–178, New York, NY, USA, 1991. Springer-Verlag New York, Inc.
- [33] R. G. G. CATTELL, D. K. BARRY, D. BARTELS, M. BERLER, J. EASTMAN, S. GAMERMAN, D. JORDAN, A. SPRINGER, H. STRICKLAND et D. WADE. *The object database standard: ODMG 2.0*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.
- [34] Chih-Chung CHANG et Chih-Jen LIN. *LIBSVM: a library for support vector machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>,
- [35] N. S. CHANG et K. S. FU. Picture query languages for pictorial data-base systems. *Computer*, 14(11):23–33, 1981.
- [36] W. CHANG et J. WANG. Metadata for multi-level content-based retrieval. *Third IEEE Meta-Data Conference*, 1999.
- [37] D. CHEN, X. LI et D.-W. CUI. An adaptive cluster validity index for the fuzzy c-means. *International Journal of Computer Science and Network Security (IJCSNS)*, 7(2):146–156, Feb 2007.
- [38] P. C. CHEN et T. PAVLIDIS. Segmentation by texture using a co-occurrence matrix and a split-and-merge algorithm. *Computer Graphics Image Processing (CGIP)*, 10(2):172–182, June 1979.
- [39] Y. CHEN, J. Z. WANG et R. KROVETZ. An unsupervised learning approach to content-based image retrieval. In *IEEE Proc. International Symposium on Signal Processing and Its Applications*, pages 197–200, 2003.
- [40] Y. CHEN, J. Z. WANG et R. KROVETZ. An unsupervised learning approach to content-based image retrieval. In *IEEE Proc. Inter. Symposium on Signal Processing and Its Applications*, pages 197–200, 2003.
- [41] Y. CHEN, J. Z. WANG et R. KROVETZ. Clue: Cluster-based retrieval of images by unsupervised learning. *IEEE Transactions on Image Processing*, 14:2005, 2005.
- [42] Jie CHENG, David A. BELL et Weiru LIU. Learning belief networks from data: an information theory based approach. In *Proceedings of the 6th international conference on Information and knowledge management (CIKM'97)*, pages 325–331, Las Vegas, NV, 1997. ACM Press.
- [43] Jie CHENG et Russell GREINER. Comparing bayesian network classifiers. In *The 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 101–108, San Francisco, CA, 1999. Disponible à l'adresse citeseer.ist.psu.edu/cheng99comparing.html.
- [44] Jie CHENG et Russell GREINER. Learning bayesian belief network classifiers: Algorithms and system. In *Proceedings of the 14th Biennial Conference of the Canadian Society on Computational Studies of Intelligence (AI'01)*, pages 141–151, London, UK, 2001. Springer-Verlag.
- [45] Jie CHENG, Russell GREINER, Jonathan KELLY, David BELL et Weiru LIU. Learning bayesian networks from data: an information-theory based approach. *Artificial Intelligence*, 137(1-2):43–90, 2002.
- [46] Ch. Yi CHIU, H. C. LIN et S. N. YANG. Texture analysis and description in linguistic terms. In *5th Asian Conference on Computer Vision*, pages 205–209, Melbourne, Australia, 2002.
- [47] Ch.-Yi CHIU, H.-Ch. LIN et S.-N. YANG. Texture retrieval with linguistic descriptions. In *PCM '01: Proceedings of the Second IEEE Pacific Rim Conference on Multimedia*, pages 308–315, London, UK, 2001. Springer-Verlag.

- [48] C.-H. CHOU, M.-C. SU et E. LAI. A new cluster validity measure and its application to image compression. *Pattern Analysis Applications*, 7(2):205–220, 2004.
- [49] C. CHOW et C. LIU. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968. Disponible à l'adresse http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1054142.
- [50] Ira COHEN, Nicu SEBE, Fabio Gagliardi COZMAN, Marcelo Cesar CIRELO et Thomas S. HUANG. Learning bayesian network classifiers for facial expression recognition using both labeled and unlabeled data. In *CVPR (1)*, pages 595–604, 2003.
- [51] R. W. CONNERS et C. A. HARLOW. A theoretical comparison of texture algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(3):204–222, 1980.
- [52] D. CONWAY. An experimental comparison of three natural language colour naming models. In *Proc. East-West International Conference on Human-Computer Interactions*, pages 328–339, 1992.
- [53] T. COVER et P. HART. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967. Disponible à l'adresse http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1053964.
- [54] I. J. COX, M. L. MILLER, T. P. MINKA, T. V. PAPATHOMAS et P. N. YIANILOS. The bayesian image retrieval system, pichunter: theory, implementation, and psychophysical experiments. *IEEE Transactions on Image Processing*, 9(1):20–37, 2000.
- [55] M. CRUCIANU, M. FERECATU et N. BOUJEMAA. Relevance feedback for image retrieval: a short review. In *In State of the Art in Audiovisual Content-Based Retrieval, Information Universal Access and Interaction including Datamodels and Languages (DELOS2 Report, 2004)*.
- [56] S. DAS, A. ABRAHAM et A. KONAR. Automatic clustering using an improved differential evolution algorithm. *IEEE Transactions on Systems, Man, and Cybernetics. part A: Systems and Humans*, 38(1):218–237, January 2008.
- [57] M. DASH, K. CHOI, P. SCHEUERMANN et H. LIU. Feature selection for clustering—a filter solution. In *2nd International Conference on Data Mining*, pages 115–122, 2002.
- [58] M. DASH et H. LIU. Feature selection for classification. *Intelligent Data Analysis*, 1(3), 1997. Disponible à l'adresse citeseer.ist.psu.edu/dash97feature.html.
- [59] Alberto DEL BIMBO. *Visual Information Retrieval*. Morgan Kufman Publishers, San Francisco, CA, 1999.
- [60] P. DEMARTINES. *Analyse de données par réseau de neurones auto-organisés*. Thèse de Doctorat, Institut National Polytechnique de Grenoble, Laboratoire TIRF, décembre 1994.
- [61] P. DEMARTINES et J. HÉRAULT. Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks*, 8(1):148–154, janvier 1997. Disponible à l'adresse citeseer.ist.psu.edu/demartines96curvilinear.html.
- [62] E. DIDELET et E. BOUTLEUX. *Détection and Diagnostic de perturbations dans un réseau de télécommunications de processus légers : concepts and exemples*. Diagnostic, intelligence artificielle and reconnaissance des formes, Editions Hermès, 2001.
- [63] T. G. DIETTERICH et G. BAKIRI. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.

- [64] A. DONG et B. BHANU. Active concept learning for image retrieval in dynamic databases. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 90, Washington, DC, USA, 2003. IEEE Computer Society.
- [65] J. DOUGHERTY, R. KOHAVI et M. SAHAMI. Supervised and unsupervised discretization of continuous features. In *Machine Learning: Proceedings of the Twelfth International Conference*, pages 194–202, San Francisco, CA, 1995. Morgan Kaufmann.
- [66] J. DOWE. Content-based retrieval in multimedia imaging. In *Proceedings of the SPIE Conference on Visual Communication and Image Processing*, 1993.
- [67] R. DUBES et A. K. JAIN. Validity studies in clustering methodologies. *Pattern Recognition*, 11:235–254, 1979.
- [68] R. O. DUDA et P. E. HART. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, NY, 1973.
- [69] J. C. DUNN. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3:32–57, 1974.
- [70] J. C. DUNN. Well separated clusters and optimal fuzzy partitions. *Journal of Cybernetica*, 4:95–104, 1974.
- [71] J. P. EAKINS. Automatic image content retrieval: Are we going anywhere? In *In Proc. 3rd international conference on electronic library and visual information research*, pages 123–135, Milton Keynes, UK, 1996.
- [72] M. EL-MELEGY, E. A. ZANATY, W. M. ABD-ELHAFIEZ et A. FARAG. On cluster validity indexes in fuzzy and hard clustering algorithms for image segmentation. *IEEE International Conference on Image Processing (ICIP)*, 6:5–8, September 2007.
- [73] FAYYAD et IRANI. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the International Joint Conference on Uncertainty in AI*, pages 1022–1027, 1993. Disponible à l'adresse <http://www.cs.orst.edu/bulatov/papers/fayyad-discretization.pdf>.
- [74] U. M. FAYYAD et K. B. IRANI. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 1022–1027. Morgan Kaufmann, 1993. Disponible à l'adresse <http://www.citidel.org/>.
- [75] H. FENG et T.-S. CHUA. A bootstrapping approach to annotating large image collection. In *MIR '03: Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, pages 55–62, New York, NY, USA, 2003. ACM.
- [76] H. FENG, R. SHI et T.-S. CHUA. A bootstrapping framework for annotating and retrieving www images. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 960–967, New York, NY, USA, 2004. ACM.
- [77] R. FLETCHER. *Practical Methods of Optimization*, chapter 8.7 : Polynomial time algorithms, pages 183–188. John Wiley & Sons, New York, NY, second édition, 1987.
- [78] François FLEURET. Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, 5:1531–1555, 2004.
- [79] M. FLICKNER, H. SAWHNEY et AL.. Query by image and video content: The qbic system. *IEEE Computer*, 1995.

- [80] M. FLICKNER, H. S. SAWHNEY, J. ASHLEY, Q. HUANG, B. DOM, M. GORKANI, J. HAFNER, D. LEE, D. PETKOVIC, D. STEELE et P. YANKER. Query by image and video content: The qbic system. *IEEE Computer*, 28(9):23–32, 1995.
- [81] Dimitris FRAGOUDIS, Dimitris MERETAKIS et Spiros LIKOTHANASSIS. Integrating feature and instance selection for text classification. In *Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'02)*, pages 501–506, Edmonton, Alberta, Canada, 2002. ACM Press.
- [82] W. B. FRAKES et R. BÄEZA-YATES. *Information Retrieval: Data Structures & Algorithms*. Prentice-Hall, 1992.
- [83] E. FRANK et I. H. WITTEN. Making better use of global discretization. In *In Proceedings of the 17th International Conference on Machine Learning*, pages 115–123. Morgan Kaufmann, 1999.
- [84] Olivier FRANÇOIS et Philippe LERAY. Evaluation d'algorithmes d'apprentissage de structure pour les réseaux bayésiens. In *AFCE/AFRIF-AFIA RFIA - Reconnaissance de Formes et Intelligence Artificielle*, pages 1453–1460, 2004.
- [85] Jerome H. FRIEDMAN. Another approach to polychotomous classification. Rapport technique, Department of Statistics, Stanford University, 1996. Disponible à l'adresse <http://www-stat.stanford.edu/~jhf/ftp/poly.ps.Z>.
- [86] Nir FRIEDMAN, Dan GEIGER et Moises GOLDSZMIDT. Bayesian network classifiers. *Machine Learning*, 29(2-3):131–163, 1997.
- [87] Y. FUKUYAMA et T. SUGENO. A new method of choosing the number of clusters for the fuzzy c-means method. In *In Proceedings of the 5th Fuzzy Systems Symposium*, pages 247–50, 1989.
- [88] T. S. FUREY, N. CRISTIANINI, N. DUFFY, D. W. BEDNARSKI, M. SCHUMMER et D. HAUSLER. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914, octobre 2000.
- [89] F. GALTON. Co-relations and their measurement, chiefly from anthropometric data. *Royal Society of London Proceedings Series I*, 45:135–145, 1888.
- [90] B. GANTER. Two basic algorithms in concept analysis. Rapport technique, Technische Hochschule Darmstadt, 1984.
- [91] Ashutosh GARG et Dan ROTH. Understanding probabilistic classifiers. In *Proceedings of the 12th European Conference on Machine Learning (EMCL'01)*, pages 179–191, London, UK, 2001. Springer-Verlag.
- [92] F. GARZOTTO, P. PAOLINI et D. SCHWABE. Hdm-a model-based approach to hypertext application design. *ACM Transactions on Information Systems (TOIS)*, 11(1):1–26, 1993.
- [93] I. GATH et A. B. GEVA. Unsupervised optimal fuzzy clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(7):773–780, 1989.
- [94] Jan-Mark GEUSEBROEK, Rein van den BOOMGAARD, Arnold W. M. SMEULDERS et Hugo GEERTS. Color invariance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(12):1338–1350, 2001.
- [95] Theo GEVERS et Arnold W. M. SMEULDERS. Color-based object recognition. *Pattern Recognition*, 32(3):453–464, 1999.
- [96] E. E. GHISELLI. *Theory of Psychological Measurement*. McGraw Hill, 1964.

- [97] R. GODIN, R. MISSAOUI et H. ALAOUÏ. Incremental concept formation algorithms based on galois (concept) lattices. *Computational Intelligence*, 11(2):246–267, 1995.
- [98] A. GOH et H. K. MENG. A multi-mechanism rule-extraction pipeline for use on unannotated datasets. In *Trends in Electronics Conference (TENCON 2000)*, volume 1, pages 382–387. IEEE Published 2000/12, 2000.
- [99] I. GONDRA. Learning in region-based image retrieval with generalized support vector machines. In *In Proc. of the Computer Vision and Pattern Recognition*, pages 149–154, 2004.
- [100] C. C. GOTLIEB et H. E. KREYSZIG. Texture descriptors based on co-occurrence matrices. *Computer Vision, Graphics, and Image Processing*, 51:70–86, 1990.
- [101] Horia GRECU et Patrick LAMBERT. Indexation par descripteurs flous: Application à la recherche d’images. In *18e colloque sur le traitement du signal et des images (GRETSI’01)*, volume 14, page 748, 2001.
- [102] P. GROS, G. MCLEAN, R. DELON, R. MOHR, C. SCHMID et G. MISTLER. Utilisation de la couleur pour l’appariement et l’indexation d’images. Technical report RR-3269, INRIA, septembre 1997.
- [103] T. R. GROSS. Code optimization of pipeline constraints. Technical report, Computer Systems Laboratory, Stanford University, décembre 1983.
- [104] J. W. GRZYMALA-BUSSE. Data reduction: discretization of numerical attributes. *Handbook of data mining and knowledge discovery*, pages 218–225, 2002.
- [105] V. N. GUDIVADA et V. V. RAGHAVAN. Content-based image retrieval systems. *Computer*, 28(9):18–22, 1995.
- [106] M. GURBAN et J. THIRAN. An information theoretic perspective on multimodal signal processing. Rapport technique 38, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, 2005.
- [107] A. GUTTMAN. R-trees: A dynamic index structure for spatial searching. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 47–54, 1984.
- [108] Isabelle GUYON et André ELISSEEFF. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.
- [109] J. F. HADDON et J. F. BOYCE. Co-occurrence matrices for image analysis. *IEEE Electronics and Communications Engineering Journal*, 5(2):71–83, 1993.
- [110] J. HAFNER, H. S. SAWHNEY, W. EQUITZ, M. FLICKNER et W. NIBLACK. Efficient color histogram indexing for quadratic form distance functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7):729–736, 1995.
- [111] M. HALKIDI, Y. BATISTAKIS et M. VAZIRGIANNIS. Cluster validity methods: part ii. *SIGMOD Rec.*, 31(3):19–27, 2002.
- [112] M. HALL. *Correlation-Based Feature Selection for Machine Learning*. Thèse de Doctorat, Département de Computer Science, Waikato University, Finland, 1998.
- [113] Mark A. HALL. Correlation-based feature selection for discrete and numeric class machine learning. In *Proceedings of the 17th International Conference on Machine Learning (ICML’00)*, pages 359–366, San Francisco, CA, 2000. Morgan Kaufmann Publishers.

- [114] Mark A. HALL et Geoffrey HOLMES. Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transactions on Knowledge and Data Engineering*, 15(6):1437–1447, 2003.
- [115] R. M. HARALICK. Statistical and structural approach to texture. *Proceedings of the IEEE*, 67, 1979.
- [116] R. M. HARALICK, K. SHANMUGAM et I. DINSTEIN. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610–621, nov 1973.
- [117] Robert M. HARALICK et Linda G. SHAPIRO. Glossary of computer vision terms. *Pattern Recognition*, 24(1):69–93, 1991.
- [118] T. HASTIE, R. TIBSHIRANI et J. FRIEDMAN. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, 2001.
- [119] Trevor HASTIE et Robert TIBSHIRANI. Classification by pairwise coupling. In Michael I. JORDAN, Michael J. KEARNS et Sara A. SOLLA, réds., *Advances in Neural Information Processing Systems*, volume 10. The MIT Press, 1998. Disponible à l'adresse citeseer.ist.psu.edu/hastie98classification.html.
- [120] Ling HE, Ling-Da WU, Yichao CAI et Yuchi LIU. Indexing structures for content-based retrieval of large image databases: A review. In *Second Asia Information Retrieval Symposium, AIRS*, pages 626–634, Jeju Island, Korea, octobre 2005.
- [121] David HECKERMAN. A tutorial on learning with bayesian networks. In *Proceedings of the NATO Advanced Study Institute on Learning in graphical models*, pages 301–354, Erice, Italy, 1998. Kluwer Academic Publishers.
- [122] D. HÉRIN et D. A. ZIGHED, réds. *Extraction et gestion des connaissances*, volume 1 of *Extraction des Connaissances et Apprentissage*, Montpellier, France, 21-23 january 2002. Hermes Science Publications.
- [123] John H. HOLLAND. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press, 1975.
- [124] G. HOLMES et C. G. NEVILL-MANNING. Feature selection via the discovery of simple classification rules. In *Proceedings of the International Symposium on Intelligent Data Analysis (IDA'95)*, 1995.
- [125] R. C. HOLTE. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11(1):63–90, 1993.
- [126] P. HOWARTH et S. RÜGER. Evaluation of texture features for content-based image retrieval. In *Proceedings of the International Conference on Image and Video Retrieval (CIVR'04)*, volume LNCS 3115, pages 326–334, Dublin, Ireland, jul 2004.
- [127] C.-W. HSU et C.-J. LIN. A comparison of methods for multi-class support vector machines. Rapport technique 19, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 2001. Disponible à l'adresse citeseer.ist.psu.edu/hsu01comparison.html.
- [128] Z. HUA, X.-J. WANG, Q. LIU et H. LU. Semantic knowledge extraction and annotation for web images. In *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pages 467–470, New York, NY, USA, 2005. ACM.

- [129] T. S. HUANG, S. MEHROTRA et K. RAMCHANDRAN. Multimedia analysis and retrieval system (mars) project. In *Proceedings of the 33rd Annual Clinic on Library Application of Data Processing-Digital Image Access and Retrieval*, 1996.
- [130] T. S. HUANG et X. S. ZHOU. Relevance feedback in content-based image retrieval: Some recent advances. In *JCIS*, pages 15–18, 2002.
- [131] C. HUDELLOT. *Towards a cognitive vision platform for semantic image interpretation; application to the recognition of biological organisms*. Thèse de Doctorat, University of Nice Sophia Antipolis, France, April 2005.
- [132] R. YICHIKAWA, K. NISHIMURA, M. KUNUGI et K. SHIMADA. Autotuning method of fuzzy membership functions using neural network learning algorithm. In *In: Proc. Second Conf. on Fuzzy Logic & Neural Networks*, pages 345–348, 1992.
- [133] N. IDRISSE, J. MARTINEZ et D. ABOUTAJDINE. Selecting a discriminant subset of co-occurrence matrix features for texture-based image retrieval. In *ISVC*, pages 696–703, 2005.
- [134] N. IDRISSE, J. MARTINEZ et D. ABOUTAJDINE. Semantic texture interpretation for image retrieval and navigation. In *4th International Symposium on Image/Video Communications over fixed and mobile networks (ISIVC 2008)*, Bilbao, Spain, 2008.
- [135] A. ION, L. STANESCU, D. BURDESCU et S. UDRISTOIU. Mapping image low-level descriptors to semantic concepts. *iccg*, 0:154–159, 2008.
- [136] Q. IQBAL et J. K. AGGARWAL. Cires: A system for content-based retrieval in digital image libraries. In *International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 205–210, Singapore, December 2002.
- [137] Q. IQBAL et J. K. AGGARWAL. Feature integration, multi-image queries, and relevance feedback in image retrieval. In *6th International Conference on Visual Information Systems (VISUAL 2003)*, pages 467–474, 24–26 September 2003.
- [138] T. ISAKOWITZ, E. A. STOHR et P. BALASUBRAMANIAN. Rmm: a methodology for structured hypermedia design. *Communication of the ACM*, 38(8):34–44, 1995.
- [139] A. K. JAIN et M. TUCERYAN. *The Handbook of Pattern Recognition and Computer Vision*, chapter 11, Texture analysis, pages 235–276. World Scientific Publishing, 1992.
- [140] Anil JAIN et Douglas ZONGKER. Feature selection: Evaluation, application, and small sample performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):153–158, 1997.
- [141] R. JAIN. Workshop report: Nsf workshop on visual information management systems. In *Proceedings of SPIE Conference on Vision, Communication, and Image Processing*, 1993.
- [142] R. JAIN, A. PENTLAND et D. PETKOVIC, réds. *NSF-ARPA workshop on visual information management systems*, Cambridge, MA, juin 1995.
- [143] J. JARMULAK et S. CRAW. Genetic algorithms for feature selection and weighting. In *In Proceedings of the IJCAI'99 workshop on Automating the Construction of Case Based Reasoners*, 1999. Disponible à l'adresse <http://www.comp.rgu.ac.uk/staff/smc/papers/ijcai99-jarmulak.pdf>.
- [144] Finn Verner JENSEN. *An introduction to Bayesian Networks*. UCL Press, London, England, 1996.
- [145] W. JIANG, K. L. CHAN, M. LI et H. ZHANG. Mapping low-level features to high-level semantic concepts in region-based image retrieval. In *CVPR '05: Proceedings of the 2005 IEEE Computer*

- Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 244–249, Washington, DC, USA, 2005. IEEE Computer Society.
- [146] W. JIN, R. SHI et T.-S. CHUA. A semi-naive bayesian method incorporating clustering with pairwise constraints for auto image annotation. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 336–339, New York, USA, 2004. ACM.
- [147] F. JING, M. LI, L. ZHANG, H.-J. ZHANG et B. ZHANG. Learning in region-based image retrieval. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, pages 206–215, 2003.
- [148] Michael I. JORDAN, réd. *Learning in graphical models*. MIT Press, Cambridge, MA, 1999.
- [149] T. JOSEPH et A. F. CARDENAS. Picquery: A high level query language for pictorial database management. *IEEE Transactions on Software Engineering*, 14(5):630–638, 1988.
- [150] M. J. KANE et A. SAVAKIS. Bayesian network structure learning and inference in indoor vs. outdoor image classification. In *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 2*, pages 479–482, Washington, DC, USA, 2004. IEEE Computer Society.
- [151] T. KANUNGO, D. M. MOUNT, N. S. NETANYAHU, C. D. PIATKO, R. SILVERMAN et A. Y. WU. An efficient k-means clustering algorithm: Analysis and implementation. *PAMI*, 24(7):881–892, July 2002.
- [152] Norio KATAYAMA et Shin'ichi SATOH. The SR-tree: an index structure for high-dimensional nearest neighbor queries. In *ACM International Conference on Management of Data (SIGMOD'97)*, pages 369–380, Tucson, AZ, mai 1997. Disponible à l'adresse citeseer.nj.nec.com/katayama97srtree.html.
- [153] L. KAUFMAN et P. J. ROUSSEEUW. *Finding groups in data. an introduction to cluster analysis*. Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics, New York: Wiley, 1990, 1990.
- [154] S. S. KEERTHI, S. K. SHEVADE, C. BHATTACHARYYA et K. R. K. MURTHY. Improvements to platt's smo algorithm for svm classifier design. *Neural Computing*, 13(3):637–649, 2001.
- [155] S. KHOSHFIAN et A. B. BAKER. *Multimedia and Imaging Databases*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1994.
- [156] Han-Bin KIM et Rae-Hong PARK. Extracting spatial arrangement of structural textures using projection information. *Pattern Recognition*, 25(3):237–245, 1992.
- [157] J. K. KIM et H. W. PARK. Statistical texture features for detection of microcalcifications in digitized mammograms. *IEEE Transactions on Medical Imaging*, 18(3), mars 1999.
- [158] Kenji KIRA et Larry A. RENDELL. A practical approach to feature selection. In *Proceedings of the 9th international workshop on Machine learning (ML'92)*, pages 249–256, Aberdeen, Scotland, UK, 1992. Morgan Kaufmann Publishers.
- [159] R. KOHAVI. Bottom-up induction of oblivious read-once decision graphs: strengths and limitations. In *AAAI '94: Proceedings of the twelfth national conference on Artificial intelligence*, volume 1, pages 613–618, Menlo Park, CA, USA, 1994. American Association for Artificial Intelligence.
- [160] R. KOHAVI et G. JOHN. Wrappers for feature subset selection. *Artificial Intelligence*, pages 273–324, 1997.

- [161] Ron KOHAVI. Scaling up the accuracy of naive-Bayes classifiers: a decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 202–207, 1996. Disponible à l'adresse citeseer.ist.psu.edu/kohavi96scaling.html.
- [162] T. KOHONEN. *Self-Organizing Maps*, volume 30. Springer Verlag, Berlin, San Francisco, CA, USA, 1995.
- [163] Vassili KOVALEV et Maria PETROU. Multidimensional co-occurrence matrices for object recognition and matching. *Graphics Models and Image Processing*, 58(3):187–197, 1996.
- [164] Paul J. KRAUSE. Learning probabilistic networks. *Knowledge Engineering Review*, 13(4):321–351, 1999.
- [165] C. KRIER, D. FRANCOIS, V. WERTZ et M. VERLEYSSEN. Feature scoring by mutual information for classification of mass spectra. In *7th International FLINS Conference on Applied Artificial Intelligence (FLINS'06)*, Genova, Italy, août29-31 2006.
- [166] S. KULKARNI. Interpretation of fuzzy logic for texture queries in cbir. *Vision, Video, and Graphics*, pages 175–179, 2003.
- [167] S. KULKARNI et B. VERMA. Fuzzy logic based texture queries for cbir. In *ICCIMA '03: Proceedings of the 5th International Conference on Computational Intelligence and Multimedia Applications*, page 223, Washington, DC, USA, 2003. IEEE Computer Society.
- [168] S. KULLBACK. *Information Theory and Statistics*. Wiley, New York, NY, 1959.
- [169] N. KWAK et C.-H. CHOI. Input feature selection by mutual information based on parzen window. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(12):1667–1671, décembre 2002.
- [170] N. KWAK et C.-H. CHOI. Input feature selection for classification problems. *IEEE Transactions on Neural Networks*, 13(1):143–159, janvier 2002.
- [171] H. LAANAYA. *Classification en environnement incertain : application à la caractérisation de sédiments marins*. Thèse de Doctorat, Université de Bretagne Occidentale, France, 14 Décembre 2007.
- [172] H. LAANAYA, A. MARTIN, D. ABOUTAJDINE et A. KHENCHAF. Knowledge discovery on database for seabed characterization. In *MCSEAI*, Morocco, 2006.
- [173] J. R. LANDIS et G. G. KOCH. The measurement of observer agreement for categorical data. *Biometrics*, 33, 1977.
- [174] P. LARRAÑAGA et M. POZA. Structure learning of bayesian networks by genetic algorithms: A performance analysis of control parameters. *IEEE Journal on Pattern Analysis and Machine Intelligence*, 18(9):912–926, 1996. Disponible à l'adresse citeseer.ist.psu.edu/larranaga94structure.html.
- [175] G. LAVOUÉ, K. IDRISSE et A. BASKURT. Accès aux bases d'images par navigation visuelle. In *In Proceedings of Colloque CORESA'03*, Lyon, 2003.
- [176] S. LAWRENCE et C. L. GILES. Accessibility of information on the web. *Intelligence*, 11(1):32–39, 2000.
- [177] T.-P. LEE et V. B. KRENG. A nonparametric multi-seed data clustering technique. *Journal of the Chinese Institute of Industrial Engineers*, 25(1):1–10, 2008.

- [178] W.K. LEOW et S. Y. LAI. Scale and orientation-invariant texture matching for image retrieval. In in: M.K. PIETIKAINEN (ED.), réd., *Texture Analysis in Machine Vision*, World Scientific, Singapore, 200.
- [179] S. LESPINATS, M. VERLEYSSEN, A. GIRON et B. FERTIL. Dd-hds: a tool for visualization and exploration of high-dimensional data. *IEEE Transactions on Neural Networks*, 18(5):1264–1279, 2007.
- [180] G. LEVY et F. BAKLOUTI. Parallel algorithms for general galois lattices building. *In workshop on Distributed Data and Structures (WDAS)*, 2003.
- [181] David D. LEWIS. Feature selection and feature extraction for text categorization. In *Proceedings of the Workshop on Speech and Natural Language (HLT'91)*, pages 212–217, Harriman, NY, 1992. Association for Computational Linguistics.
- [182] David D. LEWIS et Marc RINGUETTE. A comparison of two learning algorithms for text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 81–93, Las Vegas, US, 1994. Disponible à l'adresse citeseer.ist.psu.edu/lewis94comparison.html.
- [183] J. LI, X. L. LI et D. C. TAO. Kpca for semantic object extraction in images. *Pattern Recognition*, 41(10):3244–3250, October 2008.
- [184] Q. LI, H. HU et Z. SHI. Semantic feature extraction using genetic programming in image retrieval. In *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 1*, pages 648–651, Washington, DC, USA, 2004. IEEE Computer Society.
- [185] Q. LI, Z. SHI et S. LUO. A neural network approach for bridging the semantic gap in texture image retrieval. In *International Joint Conference on Neural Networks (IJCNN)*, pages 581–585, 2007.
- [186] H.-Ch. LIN, Ch.-Y. CHIU et Sh.-N. YANG. Linstar texture: a fuzzy logic cbir system for textures. In *ACM Multimedia*, pages 499–501, 2001.
- [187] Huan LIU et Lei YU. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17(4):491–502, 2005.
- [188] Y. LIU, D. ZHANG et G. LU. Region-based image retrieval with high-level semantics using decision tree learning. *Pattern Recognition*, 41(8):2554–2570, 2008.
- [189] Y. LIU, D. ZHANG, G. LU et W.-Y. MA. A survey of content-based image retrieval with high-level semantics. *Pattern Recogn.*, 40(1):262–282, 2007.
- [190] Y. LIU, D. S. ZHANG, G. LU et W.-Y. MA. Region-based image retrieval with perceptual colors. In *In: Proceedings of Pacific-Rim Multimedia Conference (PCM)*, pages 931–938, Decembre 2004.
- [191] E. LOISANT. *Browsing a Classification of an Image Collection*. Thèse de Doctorat, UFR Sciences et Techniques, Université de Nantes, laboratoire LINA, 10 December 2005.
- [192] F. LONG, H. ZHANG et D. D. FENG. *Multimedia Information Retrieval and Management*, chapter Fundamentals of content-based image retrieval. Springer, 2003.
- [193] S. Y. LU et K. S. FU. Stochastic tree grammar inference for texture synthesis and discrimination. *Computer Graphics Image Processing (CGIP)*, 9:234–245, 1979.
- [194] Y. LU, C. HU, X. ZHU, H. J. ZHANG et Q. YANG. A unified framework for semantics and feature based relevance feedback in image retrieval systems. In *MULTIMEDIA '00: Proceedings of the eighth ACM international conference on Multimedia*, pages 31–37, New York, NY, USA, 2000. ACM.

- [195] J. LUO et A. E. SAVAKIS. Indoor vs outdoor classification of consumer photographs using low-level and semantic features. In *ICIP01*, pages II: 745–748, 2001.
- [196] J. LUO, A. E. SAVAKIS et A. SINGHAL. A bayesian network-based framework for semantic image understanding. *Pattern Recognition*, 38(6):919–934, 2005.
- [197] J. LUO, A. E. SAVAKIS et A. SINGHAL. A bayesian network-based framework for semantic image understanding. *Pattern Recognition*, 38(6):919–934, June 2005.
- [198] Vilnrotter F. M., Nevatia R. et Price K. E.. Structural analysis of natural textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):76–89, 1986.
- [199] W.-Y. MA et B. S. MANJUNATH. A texture thesaurus for browsing large aerial photographs. *Journal of The American Society for Information Science*, 49(7):633–648, 1998.
- [200] Wei-Ying MA et B. S. MANJUNATH. Netra: A toolbox for navigating large image databases. *Multimedia Systems*, 7(3):184–198, 1999. Disponible à l’adresse citeseer.ist.psu.edu/ma99netra.html.
- [201] J. B. MACQUEEN. Some methods for classification and analysis of multivariate observations. In L. M. Le CAM et J. NEYMAN, réds., *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.
- [202] N. MAILLOT, M. THONNAT et A. BOUCHER. Towards ontology-based cognitive vision. *Mach. Vision Appl.*, 16(1):33–40, 2004.
- [203] N. MAILLOT, M. THONNAT et C. HUDELLOT. Ontology based object learning and recognition : Application to image retrieval. In *Proceedings of 16th IEEE International Conference on Tools For Artificial Intelligence*, Boca Raton, USA, 2004. IEEE Computer Society Press.
- [204] M. K. MANDAL, T. ABOULNASR et S. PANCHANATHAN. Image indexing using moments and wavelets. *IEEE Transactions on Consumer Electronics*, 42:557–565, août 1996.
- [205] D. J. MARCEAU, P. J. HOWARTH, J. M. DUBOIS et D. J. GRATTON. Evaluation of the grey-level co-occurrence matrix method for land-cover classification using spot imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 28:513–519, juillet 1990.
- [206] S. MARCUS et V. S. SUBRAHMANIAN. Foundations of multimedia database systems. *Journal of ACM*, 43(3):474–523, 1996.
- [207] C. MARSALA. *Apprentissage inductif en présence de données imprécises : construction and utilisation d’arbres de décision flous*. Thèse de Doctorat, Thèse de doctorat, Université Pierre and Marie Curie, Paris, France, Janvier 1998.
- [208] J. MARTINEZ. An hypermedia framework for visualising databases contents. In Actes des 17ème Journées Bases de Données Avancées, October 2001.
- [209] J. MARTINEZ et N. MOUADDIB. Multimedia and databases: A survey. *Networking and Information Systems*, 2(1):89–132, 1999.
- [210] José MARTINEZ et Erwan LOISANT. Browsing image databases with galois’ lattices. In *Proceedings of the 2002 ACM symposium on Applied computing (ACM SAC’02)*, pages 791–795, Madrid, Spain, 2002. ACM Press.
- [211] J. MELTON et A. R. SIMON. *Understanding the new SQL: a complete guide*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

- [212] O. J. MENGSHOEL, D. ROTH et D. C. WILKINS. Stochastic greedy search: Efficiently computing a most probable explanation in bayesian networks. Rapport technique, University of Illinois at Urbana-Champaign, 2000.
- [213] V. MEZARIS, I. KOMPATSIARIS et M. G. STRINTZIS. An ontology approach to object-based image retrieval. In *In Proc. IEEE Int. Conf. on Image Processing (ICIP03)*, pages 511–514, 2003.
- [214] M. V. MODENESI, M. C. A. COSTA, A. EVSUKOFF et N. F. F. EBECKEN. Parallel fuzzy c-means cluster analysis. In Michel J. DAYDÉ, José M. L. M. PALMA, Alvaro L. G. A. COUTINHO, Esther PACITTI et João Correia LOPES, réds., *VECPAR*, volume 4395 of *Lecture Notes in Computer Science*, pages 52–65. Springer, 2006. Disponible à l'adresse <http://dblp.uni-trier.de/db/conf/vecpar/vecpar2006.html>.
- [215] B. MOGHADDAM, Q. TIAN et T. S. HUANG. Spatial visualization for content-based image retrieval. In *in: IEEE International Conference on Multimedia and Expo(ICME01)*, Waseda, pages 22–25, 2001.
- [216] Hiroshi MOTODA et Huan LIU. *Handbook of data mining and knowledge discovery*, chapter Data reduction: feature selection, pages 208–213. Oxford University Press, New York, NY, 2002.
- [217] C. NASTAR, M. MITSCHKE, C. MEILHAC et N. BOUJEMAA. Surfimage: a flexible content-based image retrieval system. In *MULTIMEDIA '98: Proceedings of the sixth ACM international conference on Multimedia*, pages 339–44, New York, NY, USA, 1998. ACM.
- [218] E. Mephu NGUIFO et P. NJIWOA. Treillis de concepts et classification supervisée. *Technique et Science Informatiques(TSI)*, 24(4):449–488, 2005.
- [219] W. NIBLACK, W. EQUITZ, M. FLICKNER, PETKOVIC, R. BARBER et Faloutsos C.. Efficient and effective querying by image content. Rapport technique, IBM Research Division, Almaden Research Center, San Jose, CA, 1993.
- [220] J. NOVVIČOVÁ, P. SOMOL, M. HAINDL et P. PUDIL. Conditional mutual information based feature selection for classification task. In *XXX*, volume LNCS 4756, pages 417–426. Springer-Verlag, 2007.
- [221] W. E. OGLE et M. STONEBRAKER. Chabot: Retrieval from a relational database of images. *Computer*, 28(9):40–48, 1995.
- [222] Y. OUYANG, Y. LING et A. D. ZHU. Ehm-based web pages fuzzy clustering algorithm. *International Conference on Multimedia and Ubiquitous Engineering*, pages 561–566, April 2007.
- [223] M. K. PAKHIRA, S. BANDYOPADHYAY et U. MAULIK. Validity index for crisp and fuzzy clusters. *Pattern Recognition*, 37(3):487–501, March 2004. Disponible à l'adresse <http://dx.doi.org/10.1016/>.
- [224] N. R. PAL et J. BISWAS. Cluster validation using graph theoretic concepts. *Pattern Recognition*, 30(4), 1997.
- [225] M. PARTIO, B. CRAMARIUC, M. GABBOUJ et A. VISA. Rock texture retrieval using gray level co-occurrence matrix. In *5th Nordic Signal Processing Symposium (NORSIG'02)*, Trollfjord, Norway, octobre4-7 2002.
- [226] Judea PEARL. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers, San Francisco, CA, 1988.
- [227] Judea PEARL. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, mars 2000.

- [228] Judea PEARL et Tom S. VERMA. A theory of inferred causation. In James F. ALLEN, Richard FIKES et Erik SANDEWALL, réds., *Principles of Knowledge Representation and Reasoning (KR'91)*, pages 441–452, San Mateo, California, 1991. Morgan Kaufmann. Disponible à l'adresse citeseer.ist.psu.edu/pearl91theory.html.
- [229] M. PEI, E. D. GOODMAN et W. F. PUNCH. Feature extraction using genetic algorithms. In *Proceeding of International Symposium on Intelligent Data Engineering and Learning98 (IDEAL98)*, Hong Kong, page 98, 1997.
- [230] Hanchuan PENG, Fuhui LONG et Chris DING. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, août 2005.
- [231] A. PENTLAND, R. PICARD et S. SCLAROFF. Photobook: Content-based manipulation of image databases. *International Journal of Computer Vision*, 1996.
- [232] A. PENTLAND, R. W. PICARD et S. SCLAROFF. Photobook: Content-based manipulation of images databases. Technical Report 255, MIT Media Laboratory Perceptual Computing, novembre 1993.
- [233] John C. PLATT. Fast training of support vector machines using sequential minimal optimization. *Advances in kernel methods: support vector learning*, pages 185–208, 1999.
- [234] R. POLI, W. B. LANGDON et N. F. MCPHEE. *A field guide to genetic programming*. Lulu.com, 2008.
- [235] P. PUDIL, J. NOVOVIČOVÁ et J. KITTLER. Floating search methods in feature selection. *Pattern Recognition Letter*, 15(11):1119–1125, 1994.
- [236] J. PUZICHA, T. HOFFMANN et J. BUHMANN. Non-parametric similarity measures for unsupervised texture segmentation and image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 267–272, juin 1997.
- [237] G. QIU, J. MORRIS et X. FAN. Visual guided navigation for image retrieval. *Pattern Recognition*, 40(6):1711–1721, 2007.
- [238] J. Ross QUINLAN. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.
- [239] J. Ross QUINLAN. *C4.5: Programs for machine learning*. Morgan Kaufmann Publishers, San Francisco, CA, 1993.
- [240] J. Ross QUINLAN. Learning decision tree classifiers. *ACM Computing Surveys*, 28(1):71–72, 1996.
- [241] A. RAO et G. L. LOHSE. Towards a texture naming system: identifying relevant dimensions of texture. In *VIS '93: Proceedings of the 4th conference on Visualization '93*, pages 220–227, 1993.
- [242] J. REN, Y. SHEN et L. GUO. A novel image retrieval based on representative colors. In *Proceedings of the Image and Vision Computing*, pages 102–107, New Zeland, November 2003.
- [243] Christelle REYNÈS, Robert SABATIER, Nicolas MOLINARI et Sylvain LEHMANN. A new genetic algorithm in proteomics: Feature selection for seldi-tof data. *Computational Statistics & Data Analysis*, 52(9):4380–4394, mai 2008. Disponible à l'adresse <http://ideas.repec.org/a/eee/csdana/v52y2008i9p4380-4394.html>.
- [244] C. ROBERT. *The bayesian choice : a decision-theoretic motivation*. Springer, New York, NY, 1994.

- [245] Robert W. ROBINSON. Counting unlabeled acyclic digraphs. In C. H. C. LITTLE, réd., *Combinatorial Mathematics*, volume LNM 622, pages 28–43. Springer-Verlag, 1977.
- [246] Marko ROBNIK-ŠIKONJA et Igor KONONENKO. Theoretical and empirical analysis of relief and relief. *Machine Learning*, 53(1-2):23–69, 2003.
- [247] A. ROSENFELD et E.B. TROY. Visual texture analysis. In *Conference Record for Symposium on Feature Extraction and Selection in Pattern Recognition*, pages 115–124, 1970.
- [248] N. ROUSSOPOULOS, C. FALOUTSOS et T. SELLIS. An efficient pictorial database system for psql. *IEEE Transactions on Software Engineering*, 14(5):639–650, 1988.
- [249] Y. RUBNER. *Perceptual metrics for image database navigation*. Thèse de Doctorat, Ph.D. dissertation, Stanford University, 1999.
- [250] Y. RUBNER et C. TOMASI. *Perceptual Metrics for Image Database Navigation*. Kluwer Academic Publishers, Norwell, MA, USA, 2001.
- [251] Y. RUI et T. HUANG. Optimizing learning in image retrieval. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 236–243, Hilton Head Island, SC, USA, June 2000.
- [252] Y. RUI, T. S. HUANG et S. F. CHANG. Image retrieval : Current techniques, promising directions and open issues. *Journal of Visual Communication and Image Representation*, 10(4):39–62, 1999.
- [253] Y. RUI, T. S. HUANG, M. ORTEGA et S. MEHROTRA. Relevance feedback: a power tool for interactive content-based image retrieval. *Circuits and Systems for Video Technology, IEEE Transactions on*, 8(5):644–655, 1998. Disponible à l'adresse http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=718510.
- [254] B. SAHINER, H.-P. CHAN, N. PETRICK, R. F. WAGNER et L. M. HADJIISKI. Stepwise linear discriminant analysis in computer-aided diagnosis: the effect of finite sample size. In Kenneth M. HANSON, réd., *Proceedings of SPIE Medical Imaging: Image Processing*, volume 3661 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 499–510, mai 1999.
- [255] G. SALTON. *The SMART Retrieval System- Experiments in Automatic Document Processing*. Englewood Cliffs, Prentice-Hall, New Jersey, 1971.
- [256] G. SALTON et M. J. LESK. Computer evaluation of indexing and text-processing. *Journal of the ACM*, 15(1):8–36, 1968.
- [257] S. SANTINI, A. GUPTA et R. JAIN. Emergent semantics through interaction in image databases. *IEEE Transactions on Knowledge and Data Engineering*, 13:337–351, 2001.
- [258] Peter SCHAÜBLE. *Proceedings of the international workshop on multimedia information retrieval (mir'97)*. 1997.
- [259] R. SCHETTINI, G. CIOCCA et S. ZUFFI. *Color Imaging Science: Exploiting Digital Media*, chapter A Survey on methods for colour image indexing and retrieval in image databases, pages 183–211. J. Wiley, 2002.
- [260] B. SCHOLKOPF, C. BURGESS et V. VAPNIK. Extracting support data for a given task. In *KDD'95*, pages 252–257. AAAI Press, 1995.
- [261] B. SCHÖLKOPF, A. J. SMOLA et K. MÜLLER. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

- [262] N. SEBE, I. COHEN, A. GARG et T. S. HUANG. *Machine Learning in Computer Vision (Computational Imaging and Vision)*. Springer-Verlag, Secaucus, NJ, 2005.
- [263] Nicu SEBE et Michael S. LEW. *Robust Computer Vision: Theory and Applications*. Kluwer Academic Publishers, Norwell, MA, 2003.
- [264] M. SERRURIER et H. PRADE. Les avantages de la discrétisation floue dans l'apprentissage de règles. 14-16 June 2004,
- [265] I. K. SETHI, I. COMAN et D. STAN. Mining association rules between low-level image features and high-level concepts. In *In Proceedings of the SPIE: data mining and knowledge discovery III*, pages 279–290, 2001.
- [266] I. K. SETHI et N. V. PATEL. Statistical approach to scene change detection. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 329–338, 1995.
- [267] C. E. SHANNON. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948. Continued in following volume,
- [268] J. SHI et J. MALIK. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:888–905, 2000.
- [269] R. SHI, H. FENG, T.-S. CHUA et C.-H. LEE. An adaptive image content representation and segmentation approach to automatic image annotation. In *International Conference on Image and Video Retrieval (CIVR)*, page 545554, 2004.
- [270] W. SIEDLECKI et J. SKLANSKY. A note on genetic algorithms for large-scale feature selection. *Handbook of pattern recognition & computer vision*, pages 88–107, 1993.
- [271] M. SINGH et M. VALTORTA. Construction of bayesian network structures from data: a brief survey and an efficient algorithm. *International Journal of Approximate Reasoning*, 12:111–131, 1995. Disponible à l'adresse citeseer.ist.psu.edu/article/singh95construction.html.
- [272] S. SINGH et M. SHARMA. Texture analysis experiments with meastex and vistex benchmarks. In *Proceedings of the International Conference on Advances in Pattern Recognition*, volume LNCS 2013, pages 417–424, Rio, mars11-14 2001. Springer-Verlag.
- [273] A. SMEULDERS, M. WORRING et S. SANTINI. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(12), 2000.
- [274] G. SMITH et I. BURNS. Measuring texture classification algorithms. *Pattern Recognition Letters*, 18:1495–1501, 1997.
- [275] J. SMITH. Image retrieval evaluation. In *IEEE Workshop on Content-based Access of Image and Video Libraries*, pages 112–113, Santa Barbara, CA, jun 1998. Disponible à l'adresse citeseer.ist.psu.edu/smith98image.html.
- [276] J. SMITH et S.-F. CHANG. Safe: A general framework for integrated spatial and feature image search. In *IEEE Workshop on Multimedia Signal Processing*, 1997. Disponible à l'adresse citeseer.ist.psu.edu/smith97safe.html.
- [277] J. R. SMITH et S.-F. CHANG. Single color extraction and image query. In *Proceedings of the IEEE International Conference on Image Processing*, 1995.
- [278] J. R. SMITH et S.-F. CHANG. Tools and techniques for color image retrieval. In *IS & T/SPIE Proceedings, Storage & Retrieval for Image and Video Databases IV*, volume 2670, 1995.

- [279] J. R. SMITH et C.-S. LI. Decoding image semantics using composite region templates. In *CBAIVL '98: Proceedings of the IEEE Workshop on Content - Based Access of Image and Video Libraries*, page 9, Washington, DC, USA, 1998. IEEE Computer Society.
- [280] John R. SMITH et Shih-Fu CHANG. Visualeek: A fully automated content-based image query system. In *ACM Multimedia*, pages 87–98, 1996. Disponible à l'adresse citeseer.ist.psu.edu/smith96visualeek.html.
- [281] P. SPIRITES, C. GLYMOUR et R. SCHEINES. *Causation, Prediction, and Search, Second Edition (Adaptive Computation and Machine Learning)*. The MIT Press, janvier 2001.
- [282] R. SRIRAM, Joseph M. FRANCOIS et William A. PEARLMAN. Texture coding using a wold decomposition model. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, pages 1382–1386, 1994.
- [283] D. STAN et I. K. SETHI. Mapping low-level image features to semantic concepts. In *Proceedings of the SPIE: Storage and Retrieval for Media Databases*, page 172179, 2001.
- [284] P. L. STANCHEV, D. GREEN et B. DIMITROV. High level color similarity retrieval. *International Journal Information Theories & Applications*, 10:283–287, 2003.
- [285] Todd A. STEPHENSON. An introduction to bayesian network theory and usage, 2000. Disponible à l'adresse citeseer.ist.psu.edu/stephenson00introduction.html.
- [286] M. STRICKER et M. ORENKO. Similarity of color images. In *Proceedings of the SPIE Storage and Retrieval for Image and Video Databases*, 1995.
- [287] V. S. SUBRAHMANYAN. *Principles of multimedia database systems*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.
- [288] M. J. SWAIN et D. H. BALLARD. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.
- [289] J. A. SWETS. Information retrieval systems. *Science*, 141:245–250, juillet 1963.
- [290] H. TAMURA, S. MORI et T. YAMAWAKI. Texture features corresponding to visual perception. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-8(6):460–473, 1978.
- [291] P. TECHASITH. Image search engine. Project report, Imperial College, London, UK, juillet 2002.
- [292] S. THEODORIDIS et K. KOUTROUBAS. *Pattern Recognition*. Academic Press, 1999.
- [293] C. P. TOWN et D. SINCLAIR. Content based image retrieval using semantic visual categories. Rapport technique, Society for Manufacturing Engineers, Technical Report MV01-211, 2001.
- [294] A. J. M. TRAINA, J. MARQUES et C. J. TRAINA. Fighting the semantic gap on cbr systems through new relevance feedback techniques. In *CBMS '06: Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems*, pages 881–886, Washington, DC, USA, 2006. IEEE Computer Society.
- [295] W. H. TSAI et K. S. FU. Image segmentation and recognition by texture discrimination: A syntactic approach. In *International Conference on Pattern Recognition (ICPR'78)*, pages 560–564, 1978.
- [296] G. E. TSEKOURAS et H. SARIMVEIS. A new approach for measuring the validity of the fuzzy c-means algorithm. *Adv. Eng. Softw.*, 35(8-9):567–575, 2004.
- [297] V. S. TSENG, J.-H. SU, B.-W. WANG et Y.-M. LIN. Web image annotation by fusing visual features and textual information. In *SAC '07: Proceedings of the 2007 ACM symposium on Applied computing*, pages 1056–1060, New York, NY, USA, 2007. ACM.

- [298] Michael UNSER et Murray EDEN. Multiresolution feature extraction and selection for texture segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):717–728, 1989.
- [299] A. VAILAYA, M. A. T. FIGUEIREDO, A. K. JAIN et H.-J. ZHANG. Image classification for content-based indexing. *IEEE Transactions on Image Processing*, 10:117–130, 2001.
- [300] Kimmo VALKEALAHTI et Erkki OJA. Reduced multidimensional co-occurrence histograms in texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):90–94, 1998.
- [301] Tim VAN ALLEN et Russell GREINER. Model selection criteria for learning belief nets: An empirical comparison. In *Proceedings of the 17th International Conference on Machine Learning (ICML'00)*, pages 1047–1054, San Francisco, CA, 2000. Morgan Kaufmann Publishers.
- [302] C. J. VAN RIJSBERGEN. *Information Retrieval, 2nd edition*. Department of Computer Science, University of Glasgow, 1979. Disponible à l'adresse citeseer.ist.psu.edu/vanrijsbergen79information.html.
- [303] M. VANNUCCI et V. COLLA. Meaningful discretization of continuous features for association rules mining by means of a som. In *12th European Symposium on Artificial Neural Networks (ESANN)*, pages 489–494, Bruges, Belgium, April 2004.
- [304] V. N. VAPNICK. *Statistical Learning Theory*. Wiley-Interscience, September 1998.
- [305] Vladimir N. VAPNIK. *The nature of statistical learning theory*. Springer-Verlag, New York, NY, 1995.
- [306] S. VARMA et R. SIMON. Iterative class discovery and feature selection using minimal spanning trees. *BMC Bioinformatics*, 5:126, 2004.
- [307] N. VASCONCELOS. Library-based coding: A representation for efficient video compression and retrieval. In *In Proceedings on Data Compression Conference (DCC97)*, pages 121–130, March 1997.
- [308] R. C. VELTKAMP et M. TANASE. Content-based image retrieval systems: A survey. Technical Report UU-CS-2000-34, Department of Information and Computing Sciences, Utrecht University, 2000. Disponible à l'adresse citeseer.ist.psu.edu/veltkamp00contentbased.html.
- [309] C. VERTAN et N. BOUJEMAA. Using fuzzy histograms and distances for color image retrieval. In *Proceedings of CIR'2000*, Brighton, UK, mai4-5 2000. Disponible à l'adresse citeseer.ist.psu.edu/vertan00using.html.
- [310] J. WANG, W.-J. YANG et R. ACHAYRA. Color clustering techniques for color-content-based image retrieval from image databases. In *Proceedings of the IEEE Conference on Multimedia Computing and Systems*, 1997.
- [311] Y. Y. WANG et J. LI. Feature-selection ability of the decision-tree algorithm and the impact of feature-selection/extraction on decision-tree results based on hyperspectral data. *International Journal of Remote Sensing*, 29:2993 – 3010, 2008.
- [312] M. WEBER, M. WELLING et P. PERONA. Towards automatic discovery of object categories. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 02:2101, 2000.
- [313] J. S. WESZKA, C. R. DYER et A. ROSENFELD. A comparative study of texture measures for terrain classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 6(269-285), 1976.

- [314] R. WILLE. Restructuring lattice theory: an approach based on hierarchies of concepts. In I. RIVAL, éd., *Symposium on Ordered sets*, pages 445–470, Dordrecht-Boston, 1982.
- [315] Nigel WILLIAMS, Sebastian ZANDER et Grenville ARMITAGE. A preliminary performance comparison of five machine learning algorithms for practical ip traffic flow classification. *SIGCOMM Computational Communication Review*, 36(5):5–16, 2006.
- [316] D. Randall WILSON et Tony R. MARTINEZ. Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research*, 6:1–34, 1997. Disponible à l'adresse citeseer.ist.psu.edu/wilson97improved.html.
- [317] M. E. J. WOOD, B. T. THOMAS et N. W. CAMPBELL. Iterative refinement by relevance feedback in content-based digital image retrieval. In *MULTIMEDIA '98: Proceedings of the sixth ACM international conference on Multimedia*, pages 13–20, New York, NY, USA, 1998. ACM.
- [318] Marcel WORRING et Theo GEVERS. Interactive retrieval of color images. *International Journal of Image and Graphics*, 1(3):387–414, 2001. Disponible à l'adresse citeseer.ist.psu.edu/worring01interactive.html.
- [319] C.-M. WU, Y.-C. CHEN et K.-S. HSIEH. Texture features for classification of ultrasonic liver images. *IEEE transactions on medical imaging*, 11(2):141–152, 1992.
- [320] D. WU, D. AGRAWAL, A. EL ABBADI, A. SINGH et T. R. SMITH. Efficient retrieval for browsing large image databases. In *In Proceedings of Conference on Information and Knowledge Management*, pages 11–18, 1996.
- [321] J.-K. WU, A. D. NARASIMHALU, B. M. Mehtre C.-P. LAM et Y. J. GAO. Core: A content-based retrieval engine for multimedia information systems. *Multimedia Systems*, 3(1):25–41, 1995.
- [322] K.-L. WU et M.-S. YANG. A cluster validity index for fuzzy clustering. *Pattern Recogn. Lett.*, 26(9):1275–1291, 2005.
- [323] X. L. XIE et G. BENI. A validity measure for fuzzy clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(8):841–847, 1991.
- [324] Lu S. Y. et Fu K. S.. A syntactic approach to texture analysis. *Computer Graphics Image Processing (CGIP)*, 7(3):303–330, June 1978.
- [325] J. YANG et V. HONAVAR. Feature subset selection using a genetic algorithm. In *IEEE Intelligent Systems*, pages 380–385, 1998.
- [326] Yiming YANG et Jan O. PEDERSEN. A comparative study on feature selection in text categorization. In *Proceedings of the 14th International Conference on Machine Learning (ICML'97)*, pages 412–420, San Francisco, CA, 1997. Morgan Kaufmann Publishers.
- [327] L. YU et H. LIU. Feature selection for high-dimensional data: A fast correlation based filter solution. In *20th International Conference on Machine Learning*, pages 56–63, 2003.
- [328] L. ZHANG, F. LIU et B. ZHANG. Support vector machine learning for image retrieval. In *International Conference on Image Processing (ICIP)*, page 710, Octobre 2001.
- [329] R. ZHAO et W. I. GROSKY. Narrowing the semantic gap - improved text-based web document retrieval using visual features. *IEEE Transactions on Multimedia*, 4:189–200, 2002.
- [330] X. ZHENG, D. CAI, X. HE, W.-Y. MA et X. LIN. Locality preserving clustering for image database. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 885–891, New York, NY, USA, 2004. ACM.

-
- [331] X. S. ZHOU et T. S. HUANG. Cbir: from low-level features to high-level semantics. In *Proceedings of SPIE Image and Video Communication and Processing*, pages 426–431, San Jose, CA, January 2000.
- [332] X. S. ZHOU et T. S. HUANG. Relevance feedback in image retrieval: A comprehensive review. *Multimedia Systems*, 8(6):536–544, April 2003.
- [333] Ying ZHU et Stuart C. SCHWARTZ. Discriminant analysis and adaptive wavelet feature selection for statistical object detection. In *ICPR (4)*, pages 86–89, 2002.
- [334] Y. ZHUANG, X. LIU et Y. PAN. Apply semantic template to support content-based image retrieval. In M. M. YEUNG, B.-L. YEO et C. A. BOUMAN, réds., *Storage and Retrieval for Media Databases (SPIE 2000)*, volume 3972 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 442–449, Decembre 1999.

